

# Python 使用说明

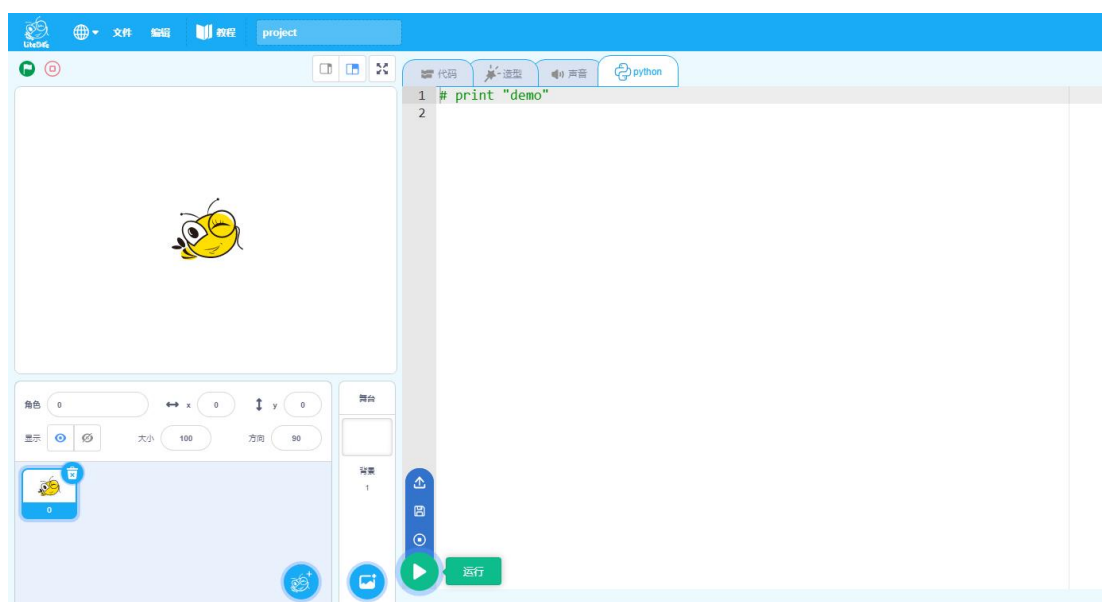
## 一、简介

LiteBee Go 的 Python 功能可以对 LiteBee Wing、Ghost II 和 LiteBee Stars 三款机型进行编程操作，包括了 LiteBee Wing 的单机模式、避障模式、编队模式和巡线模式，Ghost II 的单机模式和避障模式，以及 LiteBee Stars 编队模式。每个 Python 指令均与对应模式下的积木块编程功能一致，且在输入指令后会有相应的注释。指令的详细内容，会在第三点进行介绍。本使用说明适用于 LiteBee Go V0.3.2。

## 二、基本操作及示例

### 1. 基本操作

Python 界面如下图所示，打开软件后点击“python”即可进入 Python 编程界面，在程序编写完成以及连接上设备后，点击“运行”，飞行器即可执行 python 程序。



其中，将鼠标移动至“运行”按钮时，可弹出对程序进行“保存”和“打开”操作的界面。保存文件时，文件的后缀请保持为“.py”，便于下次打开。

### 2. 示例

#### 时间函数 import time

除了编队模式，其它模式下进行 Python 编程时都需要使用 import time 这个时间模块，定义执行任务的时间。示例如下：

```
import time          #调用“import time”函数，首行输入
                    #函数与程序主体隔行，养成程序编写习惯，便于阅读

litebeeWing_calibration() #校准
time.sleep(5)           #延时 5 秒
litebeeWing_takeOff()  #起飞
time.sleep(10)         #延时 10 秒
```

```
litebeeWing_autoland()    #执行降落
```

### 条件和循环语句

在使用“获取\*\*”指令时，需要引入条件控制语句和循环语句，Python 中的条件控制语句有 if-elif-else，循环语句有 for 和 while，示例如下：

```
import time                #定义时间函数

litebeeWing_calibration()  #校准
time.sleep(5)             #延时 5 秒
litebeeWing_takeOff()     #起飞
time.sleep(5)             #延时 5 秒
if float(litebeeWing_getVol()) > 7.5: #判断电压值是否大于 7.5V
    litebeeWing_setHigh([50]) #电压大于 7.5V 则设置高度为 50 厘米
while True:                #执行 while 循环
    if float(litebeeWing_getVol()) < 7.2: #判断电压是否小于 7.2V
        litebeeWing_autoLand() #条件满足则执行降落
        break                  #跳出 while 循环
```

注：float()为转化数据成浮点型。

### 编队示例

编队模式中，需要对每一台飞行器进行配置，以下为 2 台飞行器的程序，多台飞行器的写法类似，示例如下：

```
import time                #定义时间函数，发送指令时需要用到

litebeeWingFormation_flightnum([1]) #配置 1 号飞行器
litebeeWingFormation_calibration([5]) #校准，执行时间 5s
litebeeWingFormation_takeOff([100,5]) #起飞到 100cm 高度，执行时间 5s
litebeeWingFormation_code([10,5]) #移动到二维码 10，执行时间 5s
litebeeWingFormation_RGB([1]) #亮红灯
litebeeWingFormation_autoLand() #执行降落

litebeeWingFormation_flightnum([2]) #配置 2 号飞行器
litebeeWingFormation_calibration([5]) #校准，执行时间 5s
litebeeWingFormation_takeOff([100,5]) #起飞到 100cm 高度，执行时间 5s
litebeeWingFormation_code([20,5]) #移动到二维码 20，执行时间 5s
litebeeWingFormation_RGB([2]) #亮绿灯
litebeeWingFormation_autoLand() #执行降落

litebeeWingFormation_missionSend() #发送以上指令至飞行器
time.sleep(3) #留出指令发送的时间
litebeeWingFormation_missionStart() #执行任务
```

### 三、指令含义

注：p1,p2,p3,p4 为带参数的指令中，依次从左向右的参数。指令后括号内无参数的，

直接使用即可。例如：

`litebeeWing_setDir([0, 10, 1, 20])` 设置飞行方向

此指令的注释为：**#p1:** 0-左 1-右；**#p2:** 左右的速度；**#p3:** 0-前 1-后；**#p4:** 前后的速度。

详细含义为：第一项参数“0”（P1）有两种输入值，“0”为向左飞行，“1”为向右飞行；第二项参数“10”（P2）为第一项飞行方向的速度值，可输入范围为“0~100”（飞行器最大飞行速度为 100cm/s）；第三项参数“0”（P3）有两种输入值，“0”为向前飞行，“1”为向后飞行；第四项参数“20”（P4）为第三项飞行方向的速度值，可输入范围为“0~100”（飞行器最大飞行速度为 100cm/s）。

以下为指令的含义及注释：

### 1. litebeeWing 单机模式

`litebeeWing_autoLand()` 降落

`litebeeWing_calibration()` 校准

`litebeeWing_getAnglePitch()` 获取俯仰角度

`litebeeWing_getAngleRoll()` 获取横滚角度

`litebeeWing_getAngleYaw()` 获取偏航角度

`litebeeWing_getHigh()` 获取高度

`litebeeWing_getVol()` 获取电压

`litebeeWing_setBuzzer(['E4', 0.5])` 拓展蜂鸣器

**#p1:**音阶[

**#'B0', 'C1', 'CS1', 'D1',**

**#'DS1', 'E1', 'F1', 'FS1',**

**#'G1', 'GS1', 'A1', 'AS1',**

**#'B1', 'C2', 'CS2', 'D2',**

**#'DS2', 'E2', 'F2', 'FS2',**

**#'G2', 'GS2', 'A2', 'AS2',**

**#'B2', 'C3', 'CS3', 'D3',**

**#'DS3', 'E3', 'F3', 'FS3',**

**#'G3', 'GS3', 'A3', 'AS3',**

**#'B3', 'C4', 'CS4', 'D4',**

**#'DS4', 'E4', 'F4', 'FS4',**

**#'G4', 'GS4', 'A4', 'AS4',**

**#'B4', 'C5', 'CS5', 'D5',**

**#'DS5', 'E5', 'F5', 'FS5',**

**#'G5', 'GS5', 'A5', 'AS5',**

**#'B5', 'C6', 'CS6', 'D6',**

**#'DS6', 'E6', 'F6', 'FS6',**

**#'G6', 'GS6', 'A6', 'AS6',**

**#'B6', 'C7', 'CS7', 'D7',**

**#'DS7', 'E7', 'F7', 'FS7',**

**#'G7', 'GS7', 'A7', 'AS7',**

**#'B7', 'C8', 'CS8', 'D8', 'DS8']**

**#p2:**节拍

`litebeeWing_setDir([0, 0, 0, 10])` 设置飞行方向

**#p1:** 0-左 1-右

#p2: 左右的速度  
#p3: 0-前 1-后  
#p4: 前后的速度  
#往前以 10 (0.1m/s)的速度飞行  
litebeeWing\_setHigh([110]) 设置高度  
litebeeWing\_setMotor([1, 50]) 测试电机  
#p1:第几个电机  
#p2:转速  
litebeeWing\_setRGBLed([1, 255, 255, 255]) 扩展彩灯  
#p1:第几个灯(1 2 3 4 ... 255-全部)  
#p2:R 所占分量 (0-255)  
#p3:G 所占分量 (0-255)  
#p4:B 所占分量 (0-255)  
litebeeWing\_setYaw([1, 90]) 机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度  
litebeeWing\_takeOff() 起飞

## 2. litebeeWingAvoidance 避障模式

litebeeWingAvoidance\_autoLand() 降落  
litebeeWingAvoidance\_backward() 获取离后方障碍物的距离  
#值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)  
litebeeWingAvoidance\_calibration() 校准  
litebeeWingAvoidance\_center() 获取离下方障碍物的距离  
#值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)  
litebeeWingAvoidance\_forward() 获取离前方障碍物的距离  
#值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)  
litebeeWingAvoidance\_left() 获取离左方障碍物的距离  
#值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)  
litebeeWingAvoidance\_right() 获取离右方障碍物的距离  
#值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)  
litebeeWingAvoidance\_setDir([0, 0, 0, 10]) 设置前后左右飞行的速度 (cm/s)  
#p1: 0-往左飞行 / 1-往右飞行  
#p2: 往左飞行的速度 / 往右飞行的速度  
#p3: 0-往前飞行 / 1-往后飞行  
#p4: 往前飞行的速度 / 往后飞行的速度  
litebeeWingAvoidance\_setHigh([110]) 设置高度  
litebeeWingAvoidance\_setVXYZ([0, 0, 0, 10, 0, 0]) 设置前后左右上下飞行的速度 (cm/s)  
#p1: 0-往左飞行 / 1-往右飞行  
#p2: 往左飞行的速度 / 往右飞行的速度  
#p3: 0-往前飞行 / 1-往后飞行  
#p4: 往前飞行的速度 / 往后飞行的速度  
#p5: 0-往上飞行 / 1-往下飞行  
#p6: 往上飞行的速度 / 往下飞行的速度

litebeeWingAvoidance\_setYaw([1, 90]) 机头旋转

#p1: 1-顺时针 2-逆时针

#p2: 旋转角度

litebeeWingAvoidance\_takeOff() 起飞

litebeeWing\_getVol() 获取电压

litebeeWing\_setBuzzer(['E4', 0.5]) 扩展蜂鸣器

其中，第一项参数音阶，第二项参数为节拍。

#p1: 音阶[

#'B0', 'C1', 'CS1', 'D1', 'DS1', 'E1', 'F1', 'FS1', 'G1', 'GS1', 'A1', 'AS1', 'B1', 'C2', 'CS2', 'D2', 'DS2', 'E2', 'F2', 'FS2', 'G2', 'GS2', 'A2', 'AS2', 'B2', 'C3', 'CS3', 'D3', 'DS3', 'E3', 'F3', 'FS3', 'G3', 'GS3', 'A3', 'AS3', 'B3', 'C4', 'CS4', 'D4', 'DS4', 'E4', 'F4', 'FS4', 'G4', 'GS4', 'A4', 'AS4', 'B4', 'C5', 'CS5', 'D5', 'DS5', 'E5', 'F5', 'FS5', 'G5', 'GS5', 'A5', 'AS5', 'B5', 'C6', 'CS6', 'D6', 'DS6', 'E6', 'F6', 'FS6', 'G6', 'GS6', 'A6', 'AS6', 'B6', 'C7', 'CS7', 'D7', 'DS7', 'E7', 'F7', 'FS7', 'G7', 'GS7', 'A7', 'AS7', 'B7', 'C8', 'CS8', 'D8', 'DS8']

#p2: 节拍

#0.5 ,1 ,1.5 ,2 ,2.5

litebeeWing\_setDir([0, 0, 0, 10]) 设置飞行方向

#p1: 0-左 1-右

#p2: 左右的速度

#p3: 0-前 1-后

#p4: 前后的速度

litebeeWing\_setHigh([110]) 设置高度

litebeeWing\_setMotor([1, 50]) 测试电机

#测试电机在飞行状态时无效，转速建议值为 5~50，输入值过高可能引起飞行器侧翻

#p1: 第几个电机

#p2: 转速

litebeeWing\_setRGBLed([1, 255, 255, 255]) 扩展彩灯

#p1: 第几个灯(1 2 3 4 ... 255-全部)

#p2: R 所占分量 (0-255)

#p3: G 所占分量 (0-255)

#p4: B 所占分量 (0-255)

litebeeWing\_setYaw([1, 90]) 机头旋转

#p1: 1-顺时针 2-逆时针

#p2: 旋转角度

litebeeWing\_takeOff() 起飞

### 3. litebeeWingFormation 编队模式

litebeeWingFormation\_allLand() 紧急降落

#紧急降落可同时对所有飞行器有效，且优先级高于其它指令

litebeeWingFormation\_autoLand() 降落

litebeeWingFormation\_calibration([5]) 校准

#p1: 执行时间(秒)

litebeeWingFormation\_circle([66, 77, 0, 2, 15]) 转圈

#p1:二维码编号 1  
#p2:二维码编号 2  
#p3:0-顺时针, 1-逆时针  
#p4:要旋转几个半圈  
#p5:执行时间(秒)  
#转圈的圆心为二维码号 1 和二维码号 2 的中点, 2 个半圈即回到原点  
litebeeWingFormation\_circleZ([66, 77, 0, 150, 15]) 转半圈以及调整高度  
#p1:二维码编号 1  
#p2:二维码编号 2  
#p3:0-顺时针, 1-逆时针  
#p4:同时高度上升(厘米)  
#p5:执行时间(秒)  
#转圈的圆心为二维码号 1 和二维码号 2 的中点, 旋转时高度同时上升, 只转半圈  
litebeeWingFormation\_clear() 清除之前的配置  
litebeeWingFormation\_code([66, 10]) 移动到二维码  
#p1:要移动到的二维码编号  
#p2:执行时间(秒)  
litebeeWingFormation\_codeS(['12 23 27', 15]) 连续移动  
#p1:要连续移动到的二维码编号(输入字符串, 空格作为间隔)  
#p2:执行时间(秒)  
litebeeWingFormation\_codeZ([66, 120, 10]) 移动到二维码以及调整高度  
#p1:要移动到的二维码编号  
#p2:同时升高的高度(厘米)  
#p3:执行时间(秒)  
litebeeWingFormation\_flightNum([1]) 配置飞行器编号  
#p1:配置第几号飞行器([1-16])  
litebeeWingFormation\_missionSend() 任务发送  
#任务发送应延迟数秒 (time.sleep(3)), 待任务发送完成后再执行任务  
litebeeWingFormation\_missionStart() 任务开始执行  
litebeeWingFormation\_RGB([1]) 亮灯  
#p1:设置灯的颜色(0-黑色 1-红色 2-绿色 3-黄色 4-蓝色 5-紫色 6-青色 7-白色)  
litebeeWingFormation\_setZ([100, 3]) 设置高度  
#p1:高度(厘米)  
#p2:执行时间(秒)  
litebeeWingFormation\_takeOff([100, 5]) 起飞  
#p1:起飞高度(厘米)  
#p2:执行时间(秒)

#### 4. litebeeWingLine 巡线模式

litebeeWingLine\_autoLand() 降落  
litebeeWingLine\_calibration() 校准  
litebeeWingLine\_color() 获取巡线颜色  
#颜色的值为'红'、'绿'、'蓝' (英文版为'R'、'G'、'B')  
#如: if litebeeWingLine\_color() == '红':

litebeeWingLine\_control([1, 15]) 巡线控制  
 #p1:控制方式(0-偏航 1-速度 2-高度)  
 #p2:值 (0-255)

litebeeWingLine\_crossDetrct() 是否在交点上  
 litebeeWingLine\_crossX() 距离交点中心距离坐标 X  
 litebeeWingLine\_crossY() 距离交点中心距离坐标 Y  
 litebeeWingLine\_open() 开启巡线模式  
 litebeeWingLine\_rho() 获取巡线偏心  
 litebeeWingLine\_setRGBLed([1, 255, 255, 255]) 彩灯控制  
 #p1:第几个灯(1 2 255-全部)  
 #p2:R 所占分量 (0-255)  
 #p3:G 所占分量 (0-255)  
 #p4:B 所占分量 (0-255)

litebeeWingLine\_takeOff() 起飞  
 litebeeWingLine\_theta() 获取巡线偏差  
 litebeeWingLine\_type() 获取巡线类型  
 #巡线类型的值对应的中/英文符号及含义:  
 ‘前后左右’ - ‘FBLR’, 路线图示为 “**+**”。  
 ‘前后左’ - ‘FBL’, 路线图示为 “**┌**”。  
 ‘前后右’ - ‘FBR’, 路线图示为 “**┐**”。  
 ‘后左右’ - ‘BLR’, 路线图示为 “**└**”。  
 ‘后左’ - ‘BL’, 路线图示为 “**┌**”。  
 ‘后右’ - ‘BR’, 路线图示为 “**┐**”。  
 ‘前后’ - ‘FB’, 路线图示为 “**|**”。  
 ‘后’ - ‘B’, 路线尽头。  
 #如: if litebeeWingLine\_type() == ‘前后左右’:

## 5. litebeeWingCodeAndLine 位置标签+巡线

litebeeWingCodeAndLine\_autoLand() 降落  
 litebeeWingCodeAndLine\_calibration() 校准  
 litebeeWingCodeAndLine\_closeInt() 关闭电磁铁  
 litebeeWingCodeAndLine\_color() 巡线颜色  
 litebeeWingCodeAndLine\_color1() 获取颜色 1  
 #返回 1: 检测到, 0: 未检测到  
 litebeeWingCodeAndLine\_color2() 获取颜色 2  
 #返回 1: 检测到, 0: 未检测到  
 litebeeWingCodeAndLine\_color3() 获取颜色 3  
 #返回 1: 检测到, 0: 未检测到  
 litebeeWingCodeAndLine\_color4() 获取颜色 4  
 #返回 1: 检测到, 0: 未检测到  
 litebeeWingCodeAndLine\_control([0, 10]) 巡线控制  
 #p1: 0-偏航, 1-速度, 2-高度  
 #p2: p1 对应的值  
 litebeeWingCodeAndLine\_getAnglePitch() 获取俯仰角度

litebeeWingCodeAndLine\_getAngleRoll() 获取横滚角度  
litebeeWingCodeAndLine\_getAngleYaw() 获取偏航角度  
litebeeWingCodeAndLine\_getCodeDeltaCenterX() 偏移标签中心 X  
#偏移标签中心左右差值 X  
litebeeWingCodeAndLine\_getCodeDeltaCenterY() 偏移标签中心 Y  
#偏移标签中心左右差值 Y  
litebeeWingCodeAndLine\_getCodeId() 位置标签  
litebeeWingCodeAndLine\_getFlightMode() 飞机状态  
litebeeWingCodeAndLine\_getHigh() 获取高度  
litebeeWingCodeAndLine\_getVol() 获取电压  
litebeeWingCodeAndLine\_moveColor([0,0]) 移动到颜色上方  
#p1: 移动到颜色正上方偏移中心左右 X  
#p2: 移动到颜色正上方偏移中心前后 Y  
litebeeWingCodeAndLine\_openCode() 开启位置标签模式  
litebeeWingCodeAndLine\_openColor() 开启颜色模式  
litebeeWingCodeAndLine\_openTrack() 开启巡线模式  
litebeeWingCodeAndLine\_posHold() 悬停  
litebeeWingCodeAndLine\_posHoldCode() 悬停在位置标签上方  
litebeeWingCodeAndLine\_setColorLAB([1,0,20,-10,10,-10,10]) 设置颜色 LAB  
#p1: 设置第几号颜色 (1~4)  
#p2、3: 颜色 L 的范围 (0~120)  
#p4、5: 颜色 A 的范围 (-120~120)  
#p6、7: 颜色 B 的范围 (-120~120)  
litebeeWingCodeAndLine\_setDir([0, 0, 0, 10]) 设置前后左右飞行速度  
#p1: 0-左 1-右  
#p2: 左右的速度  
#p3: 0-前 1-后  
#p4: 前后的速度  
#往前以 10 (0.1m/s)的速度飞行  
litebeeWingCodeAndLine\_setHigh([110]) 设置高度  
litebeeWingCodeAndLine\_setRGBLed([1, 255, 255, 255]) 拓展彩灯  
#p1:第几个灯(1 2 3 4 ... 255-全部)  
#p2:R 所占分量 (0-255)  
#p3:G 所占分量 (0-255)  
#p4:B 所占分量 (0-255)  
litebeeWingCodeAndLine\_setYaw([1, 90]) 设置机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度  
litebeeWingCodeAndLine\_takeOff() 起飞  
litebeeWingCodeAndLine\_type() 巡线类型

## 6. litebeeWingCode 位置标签+吸附搬运

litebeeWingCode\_autoLand() 降落  
litebeeWingCode\_calibration() 校准



litebeeWingCode\_closeInt() 关闭电磁铁  
litebeeWingCode\_closeServo() 合起机械臂  
litebeeWingCode\_color1() 获取颜色 1  
    #返回 1: 检测到, 0: 未检测到  
litebeeWingCode\_color2() 获取颜色 2  
    #返回 1: 检测到, 0: 未检测到  
litebeeWingCode\_color3() 获取颜色 3  
    #返回 1: 检测到, 0: 未检测到  
litebeeWingCode\_color4() 获取颜色 4  
    #返回 1: 检测到, 0: 未检测到  
litebeeWingCode\_getAnglePitch() 获取俯仰角度  
litebeeWingCode\_getAngleRoll() 获取横滚角度  
litebeeWingCode\_getAngleYaw() 获取偏航角度  
litebeeWingCode\_getCodeDeltaCenterX() 偏移标签中心 X  
    #偏移标签中心左右差值 X  
litebeeWingCode\_getCodeDeltaCenterY() 偏移标签中心 Y  
    #偏移标签中心左右差值 Y  
litebeeWingCode\_getCodeId() 位置标签  
    # -1: 未检测到标签, 其他: 标签号  
litebeeWingCode\_getHigh() 获取高度  
litebeeWingCode\_getVol() 获取电压  
litebeeWingCode\_moveColor([0,0]) 移动到颜色上方  
    #p1: 移动到颜色正上方偏移中心左右 X  
    #p2: 移动到颜色正上方偏移中心前后 Y  
litebeeWingCode\_openCode() 开启位置标签模式  
litebeeWingCode\_openColor() 开启颜色模式  
litebeeWingCode\_openInt() 开启电磁铁  
litebeeWingCode\_openServo() 张开机械臂  
litebeeWingCode\_posHold() 悬停  
litebeeWingCode\_posHoldCode() 悬停在位置标签上方  
litebeeWingCode\_setColorLAB([1,0,20,-10,10,-10,10]) 设置颜色 LAB  
    #p1: 设置第几号颜色 (1~4)  
    #p2、3: 颜色 L 的范围 (0~120)  
    #p4、5: 颜色 A 的范围 (-120~120)  
    #p6、7: 颜色 B 的范围 (-120~120)  
litebeeWingCode\_setDir([0, 0, 0, 10]) 设置前后左右飞行速度  
    #p1: 0-左 1-右  
    #p2: 左右的速度  
    #p3: 0-前 1-后  
    #p4: 前后的速度  
    #往前以 10 (0.1m/s)的速度飞行  
litebeeWingCode\_setHigh([110]) 设置高度  
litebeeWingCode\_setRGBLed([1, 255, 255, 255]) 拓展彩灯  
    #p1:第几个灯(1 2 3 4 ... 255-全部)

#p2:R 所占分量 (0-255)  
#p3:G 所占分量 (0-255)  
#p4:B 所占分量 (0-255)  
litebeeWingCode\_setYaw([1, 90]) 设置机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度  
litebeeWingCode\_takeOff() 起飞

## 7. litebeeWingClamping 夹取搬运

litebeeWingClamping\_autoLand() 降落  
litebeeWingClamping\_calibration() 校准  
litebeeWingClamping\_color1() 获取颜色 1  
#返回 1: 检测到, 0: 未检测到  
litebeeWingClamping\_color2() 获取颜色 2  
#返回 1: 检测到, 0: 未检测到  
litebeeWingClamping\_color3() 获取颜色 3  
#返回 1: 检测到, 0: 未检测到  
litebeeWingClamping\_color4() 获取颜色 4  
#返回 1: 检测到, 0: 未检测到  
litebeeWingClamping\_control([0, 10]) 巡线控制  
#p1: 0-偏航, 1-速度, 2-高度  
#p2: p1 对应的值  
litebeeWingClamping\_getAnglePitch() 获取俯仰角度  
litebeeWingClamping\_getAngleRoll() 获取横滚角度  
litebeeWingClamping\_getAngleYaw() 获取偏航角度  
litebeeWingClamping\_getCodeDeltaCenterX() 偏移标签中心 X  
#偏移标签中心左右差值 X  
litebeeWingClamping\_getCodeDeltaCenterY() 偏移标签中心 Y  
#偏移标签中心左右差值 Y  
litebeeWingClamping\_getCodeId() 位置标签  
# -1: 未检测到标签, 其他: 标签号  
litebeeWingClamping\_getFlightMode() 飞机状态  
litebeeWingClamping\_getHigh() 获取高度  
litebeeWingClamping\_getVol() 获取电压  
litebeeWingClamping\_moveColor([0,0]) 移动到颜色上方  
#p1: 移动到颜色正上方偏移中心左右 X  
#p2: 移动到颜色正上方偏移中心前后 Y  
litebeeWingClamping\_openCode() 开启位置标签模式  
litebeeWingClamping\_openColor() 开启颜色模式  
litebeeWingClamping\_openTrack() 开启巡线模式  
litebeeWingClamping\_posHold() 悬停  
litebeeWingClamping\_posHoldCode() 悬停在位置标签上方  
litebeeWingClamping\_setColorLAB([1,0,20,-10,10,-10,10]) 设置颜色 LAB  
#p1: 设置第几号颜色 (1~4)

#p2、3: 颜色 L 的范围 (0~120)  
#p4、5: 颜色 A 的范围 (-120~120)  
#p6、7: 颜色 B 的范围 (-120~120)  
litebeeWingClamping\_setDir([0, 0, 0, 10]) 设置前后左右飞行速度  
#p1: 0-左 1-右  
#p2: 左右的速度  
#p3: 0-前 1-后  
#p4: 前后的速度  
#往前以 10 (0.1m/s)的速度飞行  
litebeeWingClamping\_setHigh([110]) 设置高度  
litebeeWingClamping\_setRGBLed([1, 255, 255, 255]) 拓展彩灯  
#p1:第几个灯(1 2 3 4 ... 255-全部)  
#p2:R 所占分量 (0-255)  
#p3:G 所占分量 (0-255)  
#p4:B 所占分量 (0-255)  
litebeeWingClamping\_setServoAngle([90,90]) 设置舵机  
#p1: 舵机 1 旋转角度  
#p2: 舵机 2 旋转角度  
litebeeWingClamping\_setYaw([1, 90]) 设置机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度  
litebeeWingClamping\_takeOff() 起飞  
litebeeWingClamping\_type() 巡线类型

## 8. ghost2 单机模式

ghost2\_autoLand() 降落  
ghost2\_calibration() 校准  
ghost2\_getAnglePitch() 获取俯仰角度  
ghost2\_getAngleRoll() 获取横滚角度  
ghost2\_getAngleYaw() 获取偏航角度  
ghost2\_getHigh() 获取高度  
ghost2\_getVol() 获取电压  
ghost2\_setDir([1, 1]) 设置飞行方向  
#p1: 飞行方向: 1-前 2-后 3-左 4-右 5-上 6-下 7-停  
#p2: 速度: 1-缓慢 2-普通 3-快速  
ghost2\_setHigh([110]) 设置高度  
ghost2\_setMotor([1, 50]) 测试电机  
#测试电机在飞行状态时无效, 转速建议值为 5~50  
#p1:第几个电机  
#p2:转速  
ghost2\_setYaw([1, 90]) 机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度  
ghost2\_takeOff() 起飞

## 9. ghost2Avoidance 避障模式

ghost2Avoidance\_autoLand() 降落

ghost2Avoidance\_backward() 获取离后方障碍物的距离  
# 值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)

ghost2Avoidance\_calibration() 校准

ghost2Avoidance\_center() 获取离下方障碍物的距离  
# 值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)

ghost2Avoidance\_forward() 获取离前方障碍物的距离  
# 值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)

ghost2Avoidance\_left() 获取离左方障碍物的距离  
# 值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)

ghost2Avoidance\_right() 获取离右方障碍物的距离  
# 值为 0 时代表距离无效 (超出范围 240cm 或者检测出错)

ghost2Avoidance\_setDir([0, 0, 0, 10]) 设置前后左右飞行速度 (cm/s)  
#p1: 0-左 1-右  
#p2: 左右的速度  
#p3: 0-前 1-后  
#p4: 前后的速度

ghost2Avoidance\_setHigh([110]) 设置高度

ghost2Avoidance\_setVXYZ([0, 0, 0, 10, 0, 0]) 设置前后左右上下飞行速度 (cm/s)  
#p1: 0-左 1-右  
#p2: 左右的速度  
#p3: 0-前 1-后  
#p4: 前后的速度  
#p5: 0-上 1-下  
#p6: 上下的速度

ghost2Avoidance\_setYaw([1, 90]) 机头旋转  
#p1: 1-顺时针 2-逆时针  
#p2: 旋转角度

ghost2Avoidance\_takeOff() 起飞

## 10. litebeeStars 编队模式

litebeeStars\_allLand() 紧急降落

litebeeStars\_autoLand() 降落

litebeeStars\_calibration([5]) 校准  
#p1:执行时间(秒)

litebeeStars\_circleXY([X, Y, 0, 2, 15]) 转圈  
#p1:圆心 X  
#p2:圆心 Y  
#p3:0-顺时针, 1-逆时针  
#p4:要旋转几个半圈  
#p5:执行时间(秒)  
#转圈的圆心为(X,Y), 2 个半圈即回到原点

litebeeStars\_circleXYZ([X, Y, 0, Z, 15]) 转半圈和高度  
#p1:圆心 X  
#p2:圆心 Y  
#p3:0-顺时针, 1-逆时针  
#p4:同时高度上升(厘米)  
#p5:执行时间(秒)  
#转圈的圆心为(X,Y), 旋转时高度同时上升, 只转半圈

litebeeStars\_clear() 清除之前的配置

litebeeStars\_flightNum([1]) 配置飞行器编号  
#p1:配置第几号飞行器([1-16])

litebeeStars\_missionSend() 任务发送

litebeeStars\_missionStart() 开始执行

litebeeStars\_moveXY([X, Y, 10]) 移动到坐标点  
#p1:要移动到的坐标点 X  
#p2:要移动到的坐标点 Y  
#p3:执行时间(秒)

litebeeStars\_moveXYZ([X, Y, Z, 10]) 移动到坐标点和高度  
#p1:要移动到的坐标点 X  
#p2:要移动到的坐标点 Y  
#p3:同时升高的高度 Z(厘米)  
#p4:执行时间(秒)

litebeeStars\_RGB([1]) 亮灯  
#p1:设置灯的颜色(0-黑色 1-红色 2-绿色 3-黄色 4-蓝色 5-紫色 6-青色 7-白色)

litebeeStars\_setZ([100, 3]) 控制高度  
#p1:高度(厘米)  
#p2:执行时间(秒)

litebeeStars\_takeOff([100, 5]) 起飞  
#p1:起飞高度(厘米)  
#p2:执行时间(秒)