



# 小 MU 视觉传感器 III代-microPython开发教程

## MU Vision Sensor III - Development Guide



杭州摩图科技有限公司

版本 V0.0.1-beta

2019.3.28

## 免责声明和版权公告

•本手册中的信息仅适用于摩图科技公司所生产的小 MU 视觉传感器 第 III 代（下称产品）的出厂固件，固件的升级能够提升性能或者引入新功能，敬请留意摩图科技官网 [www.morpx.com](http://www.morpx.com)，以便获取最新版本，版本更新恕不另行通告。

•请仔细阅读和理解本手册中的信息，不正确的使用可能导致产品无法正常工作，检测效果变差，甚至产品损坏。

•未经摩图科技确认及授权，不可私自维修或改装产品上的电子元件，造成损坏的将不予以保修。

•本手册中所提及的技术方案、视觉算法、通讯协议均为摩图科技自主研发，具有知识产权，任何组织或个人不得拷贝、抄袭、剽窃摩图科技的技术成果，对于任何侵权行为，摩图科技将采取法律手段予以维权。

•MORPX 是杭州摩图科技有限公司的注册商标，MU 是小 MU 视觉传感器的注册商标。文本或图片中涉及到的所有商标（名称与图案）归属于其持有者，特此声明。

# 小MU视觉传感器 III代 micropython开发教程

v0.0.1-beta 2019/03/28

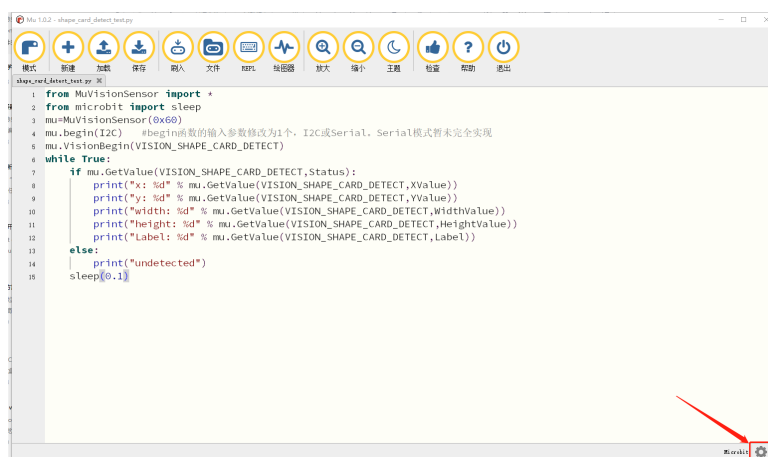
## 1. 配置Mu Editor

Mu Editor是一款具有友好GUI界面的micropython集成开发工具，包含了代码编辑、烧录、REPL终端、串口绘图器等功能。为了使用micro:bit主板控制小Mu视觉传感器，需要使用包含了MuVisionSensor模块的micropython固件，可按以下方式进行设置：

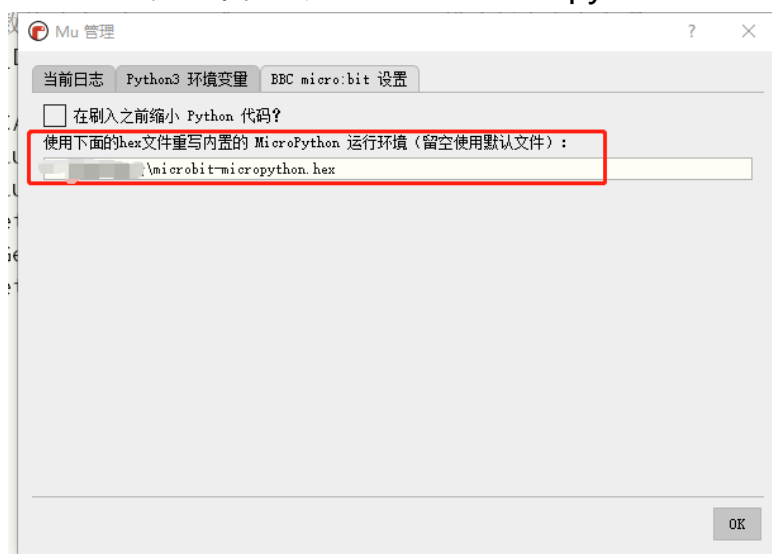
- 下载固件：

<https://github.com/tianli/MoonBot/blob/master/MuVisionSensorIII/microbit-micropython.hex>

- 下载安装mu editor，打开后点击界面右下角的设置图标



- 在micro:bit设置中，添加microbit-micropython.hex文件的路径



## 2. 导入模块

```
1 >>>from MuVisionSensor import *
```

导入模块后即可使用MuVisonSensor类中的所有公开API

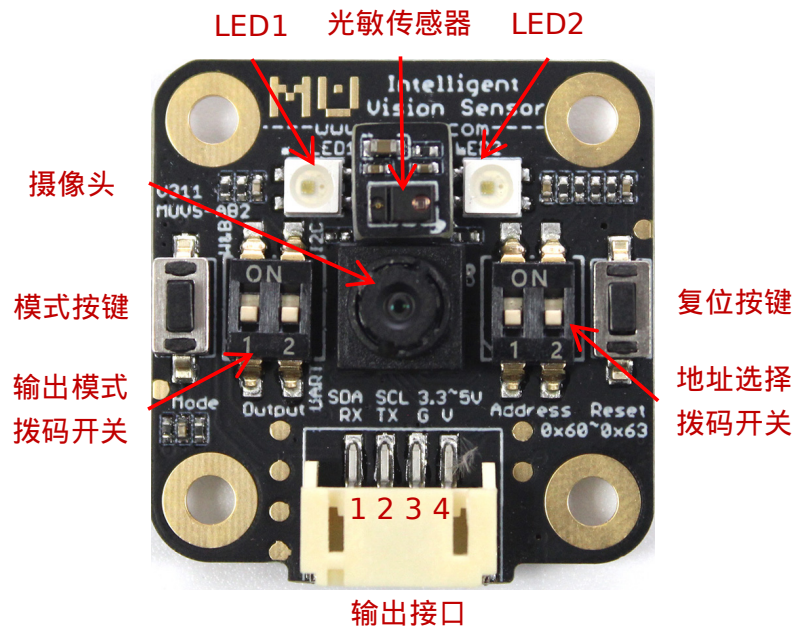
\*MuVisionSensor模块中关键字的自动补全仅在REPL模式下可用

BBC micro:bit REPL

```
MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.
>>>
>>> from MuVisionSensor import *
```

### 3. micro:bit硬件连接

#### a. I2C模式(目前仅支持I2C模式)



1. 将模块左侧输出模式拨码开关1拨至下方，2拨至上方
2. 将模块输出接口SDA (1) 口和SCL (2) 口接至microbit 对应的 SDA 口 (P20) , SCL口 (P19)
3. 将模块的地址选择拨码开关拨至对应位 (默认地址0x60则 1、2都在下方 0, 不推荐修改默认设置)

### 4. API使用说明

#### a. 初始化模块

初始化模块分为两步，首先调用MuVisionSensor(adres)创建一个对象，并指定模块地址

第二步是调用begin()方法，启动模块

```
1 #示例:
2 from MuVisionSensor import *
3 mu = MuVisionSensor(0x60) #创建对象，地址可指定为0x60~0x63
4 mu.begin() #完成初始化
```

#### b. 开启算法

API:MuVisionSensor.VisionBegin(vision\_type)

目前支持的vision\_type有:

- |                              |        |
|------------------------------|--------|
| • VISION_COLOR_DETECT        | 颜色检测   |
| • VISION_COLOR_RECOGNITION   | 颜色识别   |
| • VISION_BALL_DETECT         | 球体检测   |
| • VISION_BODY_DETECT         | 人体检测   |
| • VISION_SHAPE_CARD_DETECT   | 形状卡片检测 |
| • VISION_TRAFFIC_CARD_DETECT | 交通卡片检测 |
| • VISION_NUM_CARD_DETECT     | 数字卡片检测 |
| • VISION_ALL                 | 开启所有算法 |

多个VISION\_TYPE可以同时开启

```

1 #示例:
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 # 开启颜色检测算法
5 mu.VisionBegin(VISION_COLOR_DETECT)
6 # 同时开启形状卡片检测和球体检测算法
7 mu.VisionBegin(VISION_SHAPE_CARD_DETECT | VISION_BALL_DETECT)
```

### c. 设置算法性能

API:MuVisionSensor.VisionSetLevel(vision\_type, level)

可选的vision\_type同上

可选的level有:

- |                 |       |
|-----------------|-------|
| • LevelDefault  | 默认    |
| • LevelSpeed    | 速度优先  |
| • LevelBalance  | 平衡    |
| • LevelAccuracy | 准确性优先 |

```

1 #示例:
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 # 将球体检测算法的性能设置为速度优先
5 mu.VisionSetLevel(VISION_BALL_DETECT, LevelSpeed)
```

### d. 获取算法性能

API:MuVisionSensor.VisionSetLevel(vision\_type)

返回值0~3代表四种算法性能

```

1 #示例:
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 if mu.VisionSetLevel(VISION_BALL_DETECT) is LevelSpeed:
```

```
5 print("当前球体检测算法性能为速度优先模式")
```

## e. 开启摄像头高帧率模式（识别速度增加，同时功耗增加）

API: `MuVisionSensor.CameraSetFPS(mode)`

可选的mode有：

- `FPSNormal`      正常模式
- `FPSHigh`        高帧率模式

```
1 #示例：
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 # 开启高帧率模式
5 mu.CameraSetFPS(FPSNormal)
```

## f. 获取摄像头帧率模式

API: `MuVisionSensor.CameraGetFPS()`

返回值为 0(`FPSNormal`)或1(`FPSHigh`)

## g. 设置摄像头白平衡（调节因为外界光源变化而引起的图像偏色）

API: `MuVisionSensor.CameraSetAwb(mode)`

可选的mode有：

- `AutoWhiteBalance`      自动白平衡
- `LockWhiteBalance`      锁定白平衡
- `WhiteLight`            白光模式
- `YellowLight`            黄光模式

## h. 获取摄像头白平衡模式

API: `MuVisionSensor.CameraGetAwb()`

返回值为 0~3，对应4种白平衡模式

```
1 #示例：
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 if mu.CameraGetAwb() is AutoWhiteBalance:
5     print("当前白平衡模式为自动")
```

## i. 设置摄像头数码变焦

API: `MuVisionSensor.CameraSetZoom(mode)`

可选的mode有：

- ZoomDefault          默认
- Zoom1                  变焦模式1
- Zoom2                  变焦模式2
- Zoom3                  变焦模式3
- Zoom4                  变焦模式4
- Zoom5                  变焦模式5

## j. 获取摄像头变焦模式

API:MuVisionSensor.CameraGetZoom()

返回值为 0~5，对应6种白平衡模式

## k. 恢复模块默认设置（关闭所有算法，重置所有硬件设置）

API:MuVisionSensor.SensorSetDefault()

## l. 重启模块

API:MuVisionSensor.SensorSetRestart()

## m. LED灯光设置

API:MuVisionSensor.LedSetColor(led, detected\_color, undetected\_color, level)

参数说明：

- led：选择LED序号，可选值为
  - LED1
  - LED2
- detected\_color：检测到结果时的颜色，可选值为
  - LedClose                  LED关
  - LedRed                    红色
  - LedGreen                绿色
  - LedYellow               黄色
  - LedBlue                  蓝色
  - LedPurple               紫色
  - LedCyan                  青色
  - LedWhite                白色
- undetected\_color：检测到结果时的颜色，可选值同上
- level：亮度值，可输入0~15的数字，数值越大越亮

## n. 获取算法识别结果

API: MuVisionSensor.GetValue(vision\_type, object\_inf)

vision\_type的可选值同上

object\_inf的可选值为:

- Status                      检测状态，0代表没检测到，1代表检测到
- XValue                    目标的横向坐标
- YValue                    目标的纵向坐标
- WidthValue              目标的宽度
- HeightValue             目标的高度
- Label                    目标的标签
- RValue                   红色通道值（颜色识别模式）
- GValue                   绿色通道值（颜色识别模式）
- BValue                   蓝色通道值（颜色识别模式）

```

1 #示例:
2 from MuVisionSensor import *
3 ... #省略初始化过程
4 # 形状卡片检测
5 while True:
6     if mu.GetValue(VISION_SHAPE_CARD_DETECT, Status):
7         print("x: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT, XValue))
8         print("y: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT, YValue))
9         print("width: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT, WidthValue))
10        print("height: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT, HeightValue))
11        print("Label: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT, Label))
12    else:
13        print("undetected")
14 # 颜色识别
15 while True:
16     if mu.GetValue(VISION_COLOR_RECOGNITION, Status):
17         print("R: %d" % mu.GetValue(VISION_COLOR_RECOGNITION, RValue))
18         print("G: %d" % mu.GetValue(VISION_COLOR_RECOGNITION, GValue))
19         print("B: %d" % mu.GetValue(VISION_COLOR_RECOGNITION, BValue))
20

```