

基于OpenMV平台的Python编程实战

张秀君

2018.4.17



深圳职业技术学院
SHENZHEN POLYTECHNIC



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟上手

03. 机器视觉基础

04. 例程解析

05. 项目实战



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟上手

03. 机器视觉基础

04. 例程解析

05. 项目实战



OpenMV是什么？

简单的来说，它是一个**可编程的摄像头**，通过**MicroPython语言**，可以实现你的逻辑。而且摄像头本身内置了一些**图像处理算法**，很容易使用。



OpenMV适合做什么？

- DIY相关的项目制作，比如追踪小球的车，云台，或者解魔方的机器人。
- 对成本要求很高的嵌入式工业方案，比如流水线物品的分拣。



OpenMV不适合做什么？

- 高像素的处理任务。OpenMV只有30万的像素，通常使用320×240的分辨率来处理图像，OpenMV3 Cam M7采集图像采集图像分辨率最高可以达到640×480，无法获得更高的像素。》
- 复杂的算法：比如OCR识别，车牌识别，猫狗分类，深度学习之类的。

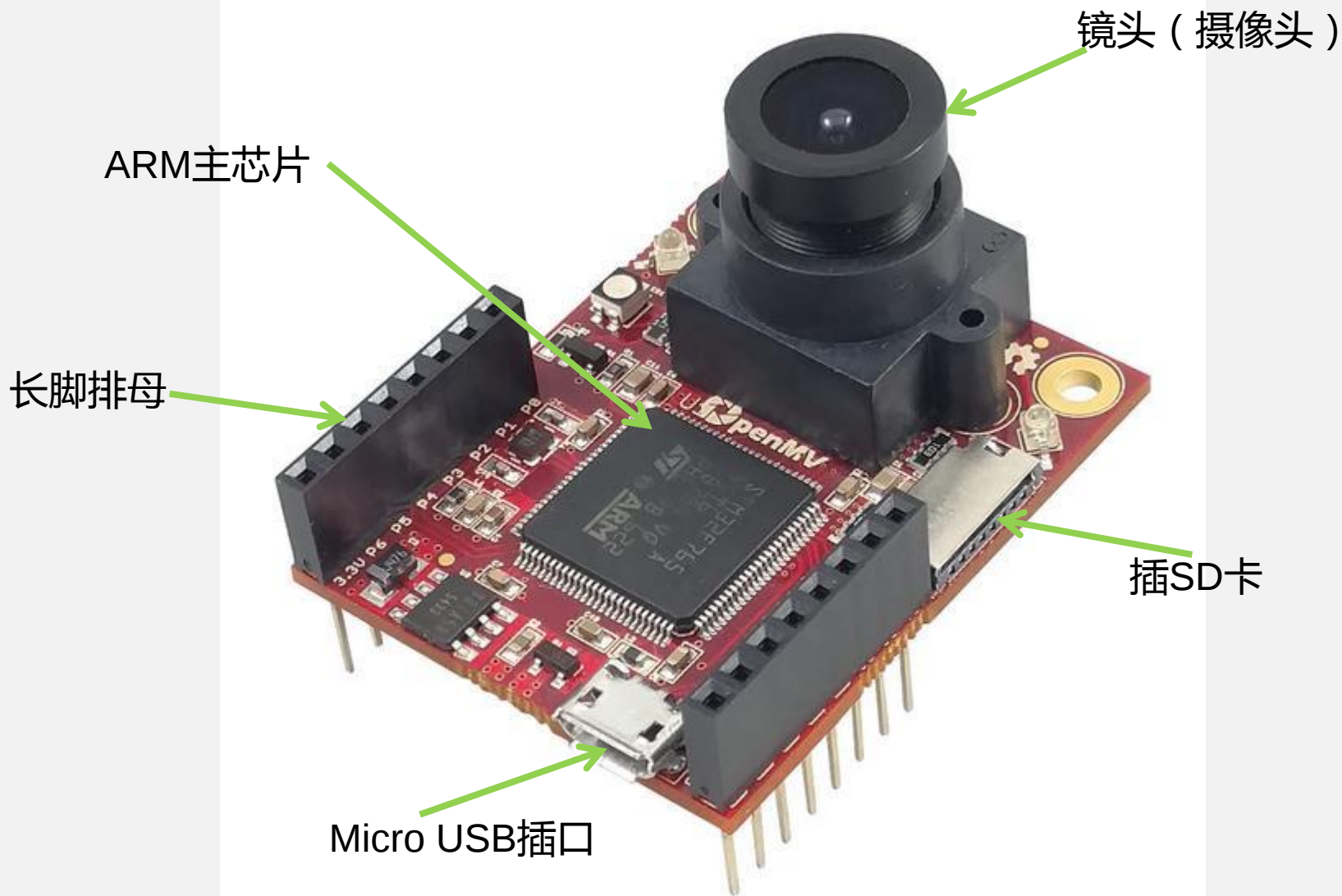
师傅领进门
修行靠个人

OpenMV美国官网 : <https://openmv.io>
OpenMV中国官网 : <http://openmv.cc>

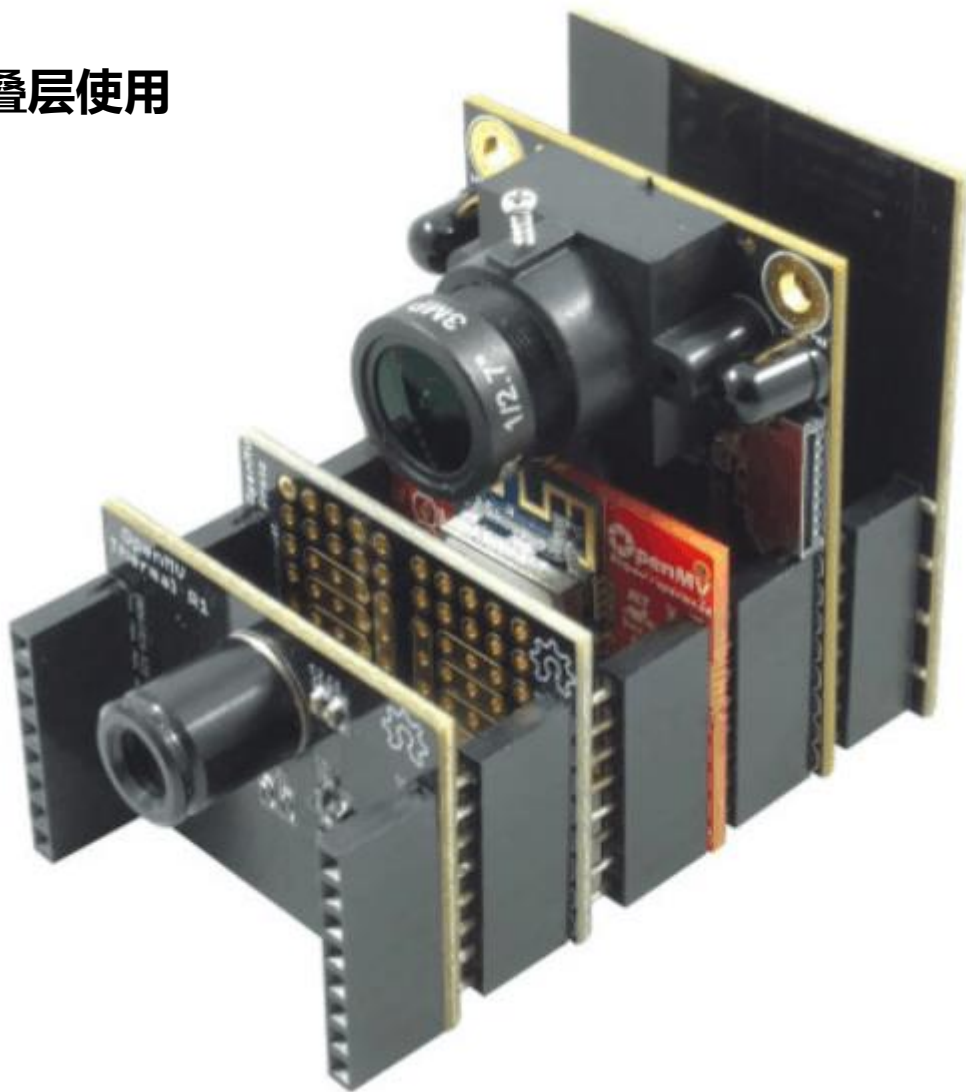
OpenMV论坛 :
<https://forum.singtown.com/category/5/openmv-cam>

各种库和使用,看官网文档 : <http://docs.openmv.io/>
中文文档 : http://doc.singtown.cc/openmv_zh/

MicroPython 文档 : <http://docs.micropython.org/en/latest/pyboard/>
MicroPython 中文文档 :
<https://docs.singtown.com/micropython/zh/latest/pyboard/index.html>



OpenMV配件的叠层使用



插入SD卡



洞洞板



长脚排母



LCD扩展板



WIFI扩展板



电池



视频教程1-OpenMV简介（7分钟）：

<https://www.bilibili.com/video/av16418889/>

观看OpenMV开箱视频（5分钟）：

<https://www.bilibili.com/video/av16790030/>

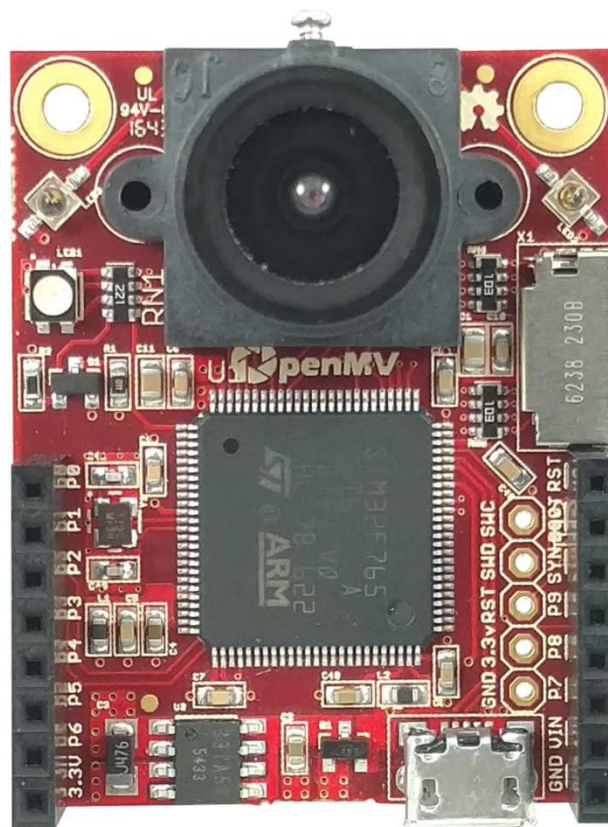
OpenMV详细管脚定义



By: Ibrahim Abdelkader & Kwabena W. Agyeman
https://openmv.io

LED1 – Red
LED2 – Green
LED3 – Blue
LED4 – IR

Peripherals / Timers	CPU Name	Pin Name
UART 1 RX	PB15	P0
UART 1 TX	PB14	P1
CAN2 TX	PB13	P2
CAN2 RX	PB12	P3
TIM2 CH3	PB10	P4
TIM2 CH4	PB11	P5
TIM2 CH1	PA5	P6
3.3V Rail (250 mA supply Max)		



All pins are 5V tolerant¹ with a 3.3V output
All pins can sink or source up to 25 mA²

¹ P6 is not 5V tolerant in ADC or DAC mode
² Up to 120mA in total between all pins

Max current used w/o μ SD card < 150 mA
Max current used w/ μ SD card < 250 mA

Micro SD Slot
SD < 2GB Max
SDHC < 32GB Max

Pin Name	CPU Name	Peripherals / Timers
Reset (Connect to GND to reset)		
BOOT 0 (Connect to 3.3V for DFU mode)		
Frame Sync (use to frame sync cams)		
P9	PD14	Servo 3
P8	PD13	Servo 2
P7	PD12	Servo 1
VIN (3.6V - 5V)		
GND Rail		



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟上手

03. 机器视觉基础

04. 例程解析

05. 项目实战

<http://book.openmv.cc/quick-starter.html>

请自学以上网址的10分钟快速上手教程，思考如下几点问题：

- ❑ 如何连接OpenMV与电脑？
- ❑ 如何安装驱动？
- ❑ 熟悉IDE开发环境界面
- ❑ “连接电脑”键在哪里？“开始运行”键在哪里？图像窗口在哪里
- ❑ 如何新建一个py文件？如何保存？
- ❑ 如何将py文件下载到OpenMV中？
- ❑ 你的OpenMV能脱机运行吗？
- ❑ Hello World运行成功了吗？



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟上手

03. 机器视觉基础

04. 例程解析

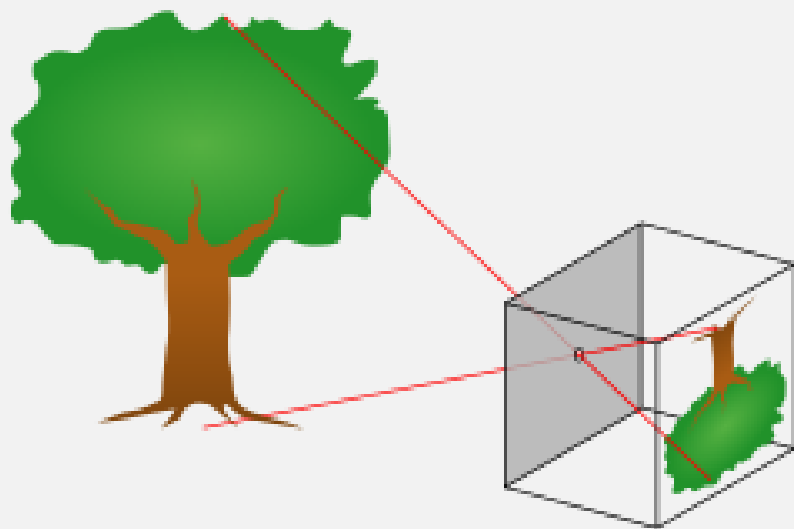
05. 项目实战

什么是摄像头？



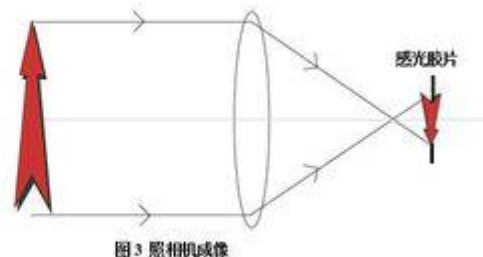
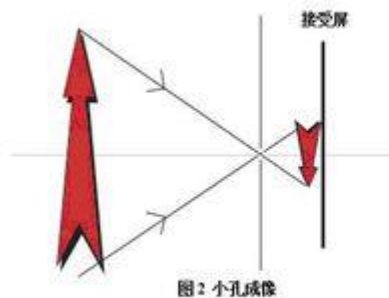
摄像头，是一个将光学信号转变成电信号的一个装置

摄像头成像原理—小孔成像



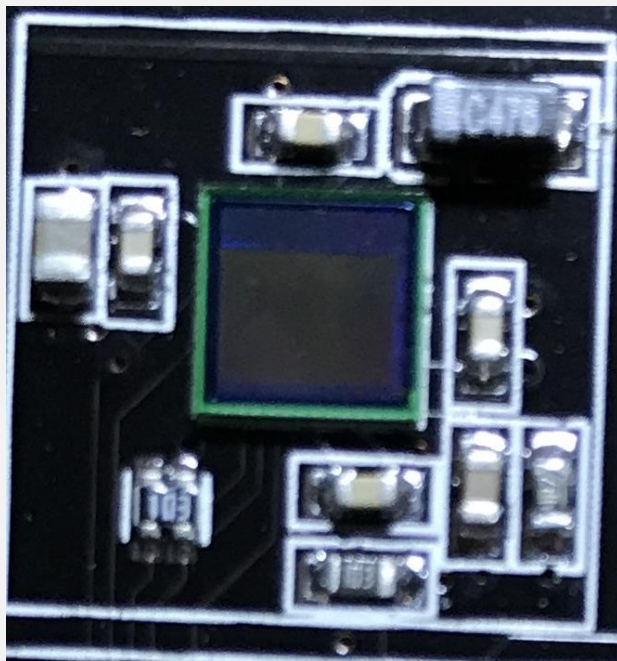
小孔模型是一种理想相机模型，没有考虑实际相机中存在的场曲、畸变等问题。但是在实际使用时，这些问题可以通过在标定的过程中引入畸变参数解决，所以小孔模型仍然是目前最广泛使用的相机模型。

小孔成像与相机成像的区别：



小孔成像原理是光的直线传播。
照相机成像原理是光的折射。
但二者都得到倒立、缩小的实像。

OpenMV的感光元件：

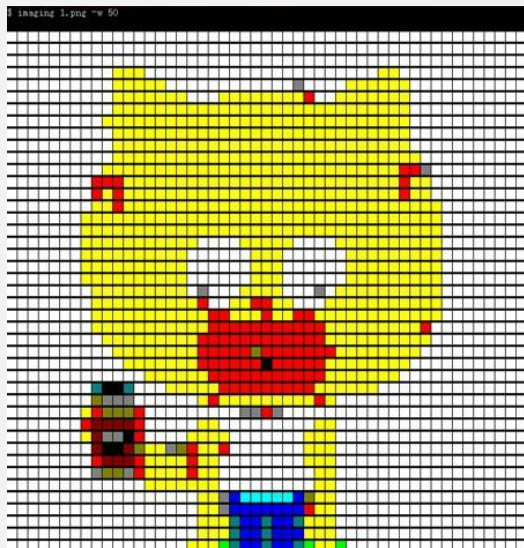


图像透过镜头，照在一个感光芯片上，感光芯片可以把光照的波长和强度等信息转成计算机（数字电路）可以识别的数字信号。

中间的方形元件就是感光元件

请同学们慢慢拧下摄像头上的镜头，观察里面的感光元件。

什么是像素和分辨率？



感光元件是有很多个感光点构成的，比如有 640×480 个点，每个点就是一个像素，把每个点的像素收集整理起来，就是一副图片，那么这张图片的分辨率就是 640×480 。

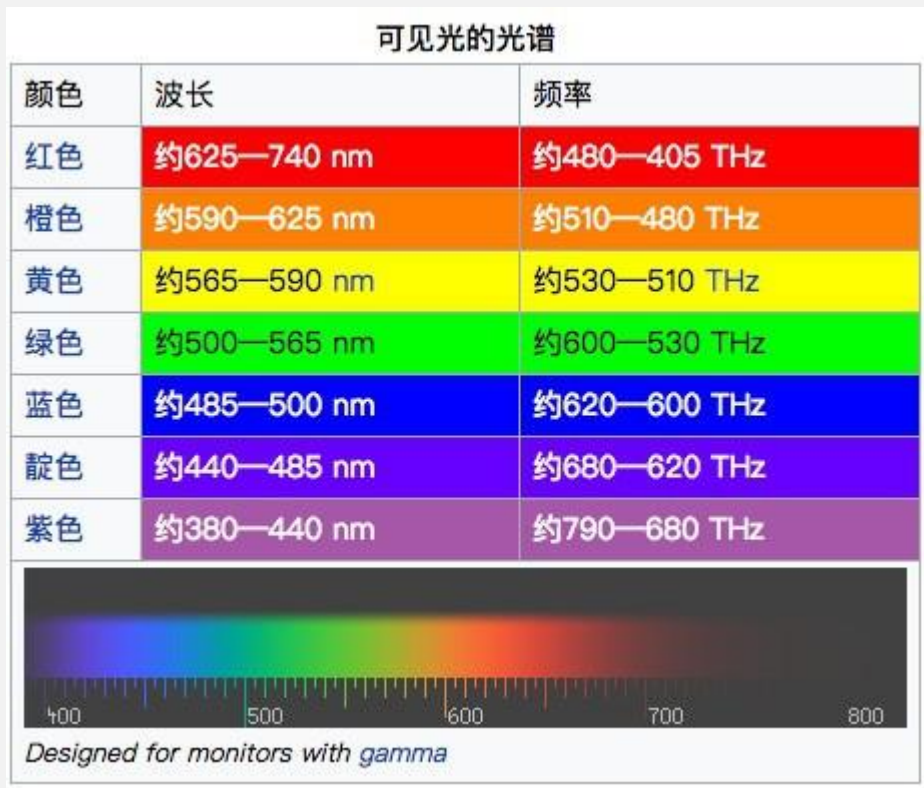
我们常听说的1080P是什么意思？高清的分辨率最低又是多少？

什么是帧率？

帧率（FPS）就是每秒钟处理的图片数量，如果超过20帧，人眼就基本分辨不出卡顿。当然，如果用在机器上，帧率是越高越好的。

- ❑ 电影胶片的播放速度固定为每秒24帧。
- ❑ 欧美、日等电视制式是NTSC，每秒钟29.97帧，也就是常说的30帧每秒。亚洲等电视制式是PAL，每秒25帧。
- ❑ 部分高清的游戏，制作水准达到60帧每秒
- ❑ 本地录像机帧数通常设置为20FPS到25FPS。
- ❑ 远程监控时使用的辅码流可以根据网络情况设置，一般设置范围在10FPS到25FPS之间，低于10FPS会使监控画面出现明显跳帧感。

什么是颜色？



物理上，颜色就是不同波长的电磁波。但是，根据人眼的视觉效果，可以通过RGB，CMYK，HSB，LAB色域，来将可见光的颜色描述出来。

RGB三原色

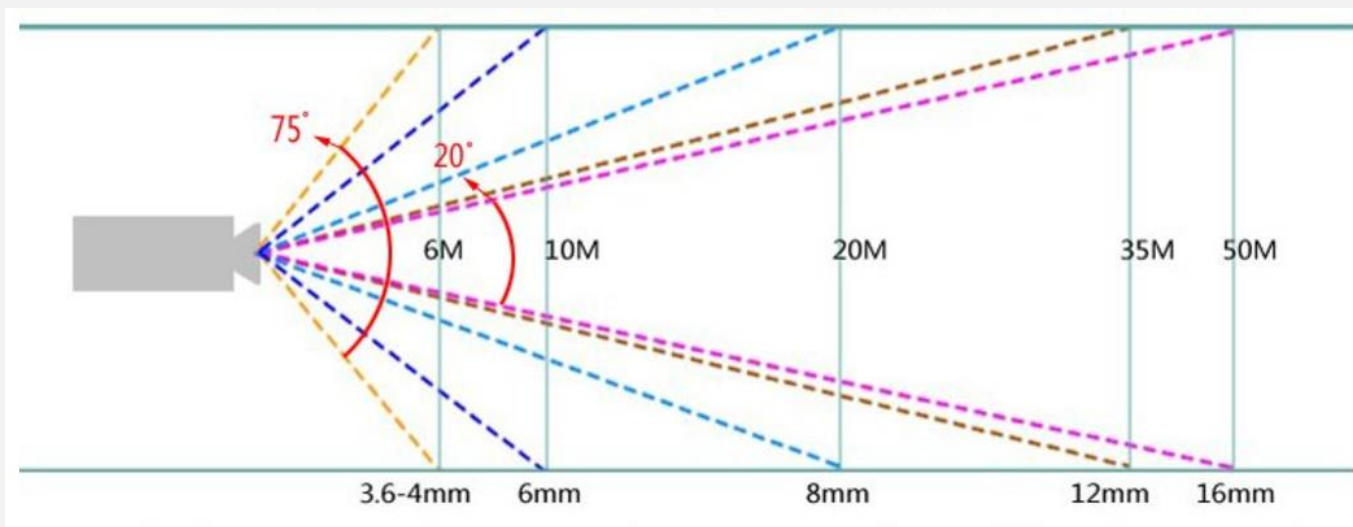


三原色的原理不是物理原因，而是由于人的生理原因造成的。人的眼睛内有几种辨别颜色的锥形感光细胞，分别对黄绿色、绿色和蓝紫色（或称紫罗兰色）的光最敏感（波长分别为564、534和420纳米）。所以RGB经常用于显示器上，用来显示图片。

LAB亮度-对比度

- Lab颜色空间中，L亮度；a的正数代表红色，负端代表绿色；b的正数代表黄色，负端代表蓝色。不像RGB和CMYK色彩空间，Lab颜色被设计来接近人类视觉。
- 因此L分量可以调整亮度对，修改a和b分量的输出色阶来做精确的颜色平衡。
- 注意：在OpenMV的查找色块的算法中，运用的就是这个LAB模式！

镜头的焦距



□ 图像是通过镜头的光学折射，照到感光元件上的。那么镜头就决定了，整个画面的大小和远近。一个最重要的参数就是焦距。

□ 镜头焦距：是指镜头光学后主点到焦点的距离，是镜头的重要性能指标。镜头焦距的长短决定着拍摄的成像大小，视场角大小，景深大小和画面的透视强弱。

□ 当对同一距离远的同一个被摄目标拍摄时，镜头焦距长的所成的象大，镜头焦距短的所成的象小。注意焦距越长，视角越小。

镜头的畸变

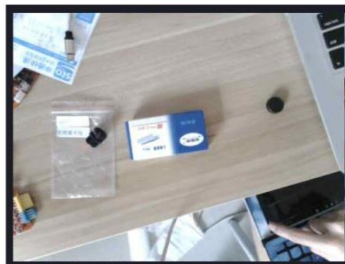
因为光学原理，在感光芯片上不同的位置，与镜头的距离不同的，简单说就是近大远小，所以在边缘会出现**鱼眼效果（桶型畸变）**。为了解决这个问题，可以在代码中使用算法来矫正畸变，注：OpenMV中使用`image.lens_corr(1.8)`来矫正2.8mm焦距的镜头。也可以直接使用无畸变镜头。无畸变镜头加入了额外的矫正透镜部分，价格自然会高不少。



标配镜头



广角镜头



无畸变镜头



长焦镜头



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟上手

03. 机器视觉基础

04. 例程解析

05. 项目实战

序号	例程题目	章节	难度
1	helloworld	Basics	容易
2	让LED亮起来！	Basics	容易
3	在图像里画线	Drawing	容易
4	绘图及写字	Drawing	容易
5	二值化滤波	Image Filter	中等
6	边缘检测	Image Filter	中等
7	腐蚀膨胀	Image Filter	中等
8	均值滤波	Image Filter	中等
9	图像锐化	Image Filter	中等
10	人脸识别拍照	Snapshot	中等
11	拍摄移动物体	Snapshot	中等

序号	例程题目	章节	难度
12	人脸检测	Face Detection	中等
13	人脸识别	Face Detection	中等
14	人脸追踪	Face Detection	中等
15	人眼追踪	Face Detection	中等
16	识别圆形	Feature Detection	中等
17	识别数字	Feature Detection	中等
18	识别矩形	Feature Detection	中等
19	颜色追踪	Color Tracking	中等
20	LCD显示	扩展板	中等
21	Wifi扩展板	扩展板	中等
22	舵机扩展板	扩展板	中等

例程1：Helloworld-你好，世界

思考如下问题：

1、 `import sensor, image, time`

这一句的含义是什么？

2、 图像分辨率在哪里设置的？当前是多少？

3、 当前图像色彩模式什么？改为灰度图像怎么改？

例程2：让LED亮起来！

思考如下问题：

- 1、LED的颜色控制是在哪里实现的？改改试试，看有什么不同。
- 2、闪灯的间隔是在哪里设置？
- 3、如果判断OpenMV是否连接电脑？

例程3：在图像里画线

思考如下问题：

- 1、摄像头初始化应该怎么做？
- 2、如何设置图像的大小、图像的颜色格式？
- 3、程序是如何实现灰、红、绿、蓝循环变色的？

例程4：绘图及写字

思考如下问题：

- 1、画线的函数是哪一个，怎么调用的？
- 2、画矩形的函数是哪一个，怎么调用的？
- 3、画圆的函数是哪一个，怎么调用的？
- 4、写字的函数是哪一个，怎么调用的
- 5、画十字的函数是哪一个，怎么调用的

例程5：二值化滤波

思考如下问题：

- 1、阈值是如何设置的？
- 2、二值化函数是哪一个，怎么调用的？
- 3、如何实现彩色图像的阈值，即三原色中都有阈值？

例程6：边缘检测

思考如下问题：

- 1、调用哪个函数实现的边缘检测，分别有哪些参数，各是什么含义？
- 2、二值化函数在这里起到什么作用？

例程7：腐蚀膨胀

思考如下问题：

- 1、什么叫腐蚀？什么叫膨胀？他们的作用是什么？
- 2、分别调用哪个函数实现的腐蚀和膨胀，各有几个参数？

例程8：均值滤波

思考如下问题：

- 1、什么叫均值滤波？
- 2、调用哪个函数实现的均值滤波，有几个参数？

例程9：图像锐化

思考如下问题：

- 1、什么叫图像锐化？
- 2、他和图像边缘检测有什么区别？代码中是如何体现这种区别的？

例程10：人脸识别拍照

思考如下问题：

- 1、什么叫人脸识别拍照？他的实现原理是什么？
- 2、调整参数，观察不同阈值参数对于识别正确率的影响。

例程11：拍摄移动物体

思考如下问题：

- 1、请百度利用帧差法检测运动信息的原理。
- 2、代码中帧间距设为多少？
- 3、如何判断是有运动的？
- 4、为什么要关掉白平衡？

例程12：人脸检测

思考如下问题：

- 1、请百度利用Haar算子检测人脸的原理。
- 2、代码中调用哪个函数实现的人脸检测？
- 3、代码中调用哪个函数实现的人脸画框功能？

例程13：人脸识别

思考如下问题：

- 1、请百度利用LBP特征识别人脸的原理。
- 2、代码中调用哪个函数实现的人脸识别？

例程14：人脸追踪

思考如下问题：

- 1、关键点检测是如何实现的？
- 2、人脸追踪和人脸识别有什么不同？
- 3、为什么要用关键点检测？

例程15：人眼追踪

思考如下问题：

- 1、如何实现的人眼追踪？
- 2、调整参数，体会不同参数对于追踪成功率的影响？

例程16：识别圆形

思考如下问题：

- 1、自学利用霍夫变换识别圆形的原理。
- 2、镜头的畸变校正是如何实现的？
- 3、熟悉函数的各个参数的含义。

例程17：识别数字

思考如下问题：

- 1、神经网络在哪里？
- 2、短短的代码如何实现的手写数字识别？

例程18：识别矩形

思考如下问题：

- 1、自学四元检测法识别矩形的原理。
- 2、熟悉函数的各个参数的含义。

例程19：颜色跟踪

思考如下问题：

- 1、如何实现的颜色跟踪？
- 2、图像的颜色直方图是什么含义？如何获取的？
- 3、机器人巡线案例代码理解。

例程20 : LCD显示

思考如下问题：

- 1、正确连接LCD扩展板，熟悉各个接口的作用
- 2、利用LCD扩展板，如何在运行时查看OpenMV Cam的帧缓冲区？

例程21：WIFI扩展板

思考如下问题：

- 1、正确连接WIFI扩展板，熟悉各个接口的作用
- 2、如何利用WIFI扩展板实时传送巡线小车摄像头中所摄视频？

例程22：舵机扩展板

思考如下问题：

- 1、正确连接舵机扩展板，熟悉各个接口的作用
- 2、如何利用舵机扩展板控制巡线小车的运动？



课程内容

CONTENTS

01. 初识OpenMV

02. 10分钟快速上手

03. 机器视觉基础

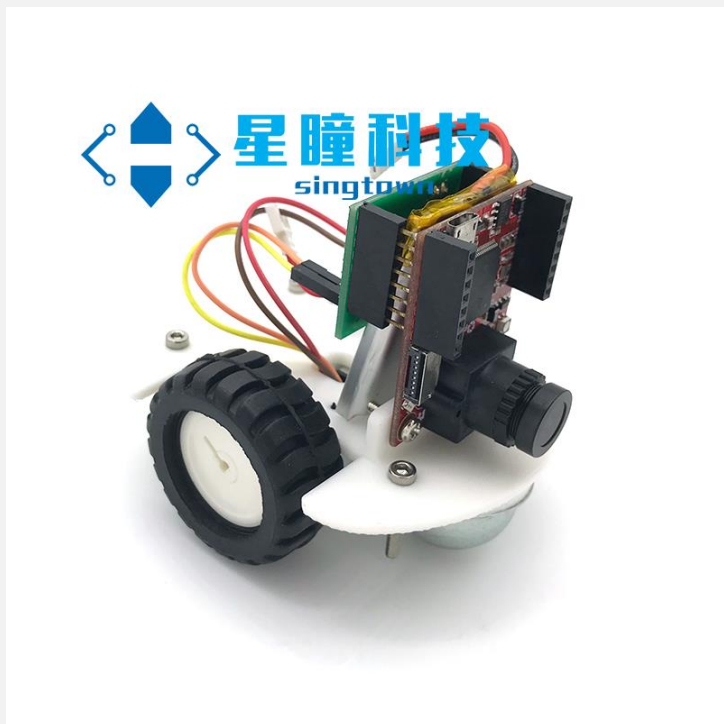
04. 例程解析

05. 项目实战

追小球的小车

准备材料清单：

- 1、OpenMV Cam7 电路板，1个
- 2、3D打印的小车底板，1个
- 3、3.7V锂电池，1个
- 4、TB6612电机驱动板，1个
- 5、牛眼轮，2个
- 6、N20直流电机（含固定座，含轮胎），2个
- 7、M3*20螺丝螺母，2个
- 8、M2*4自攻螺丝，2个



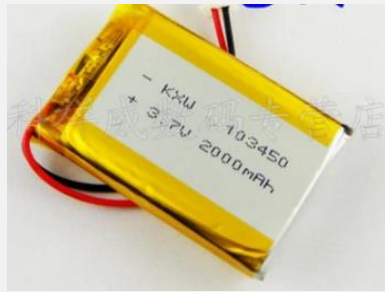
物料图片



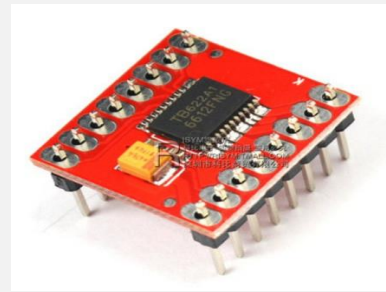
OpenMV Cam7 电路板



3D打印的小车底板



3.7V锂电池



TB6612电机驱动板



牛眼轮



N20直流电机

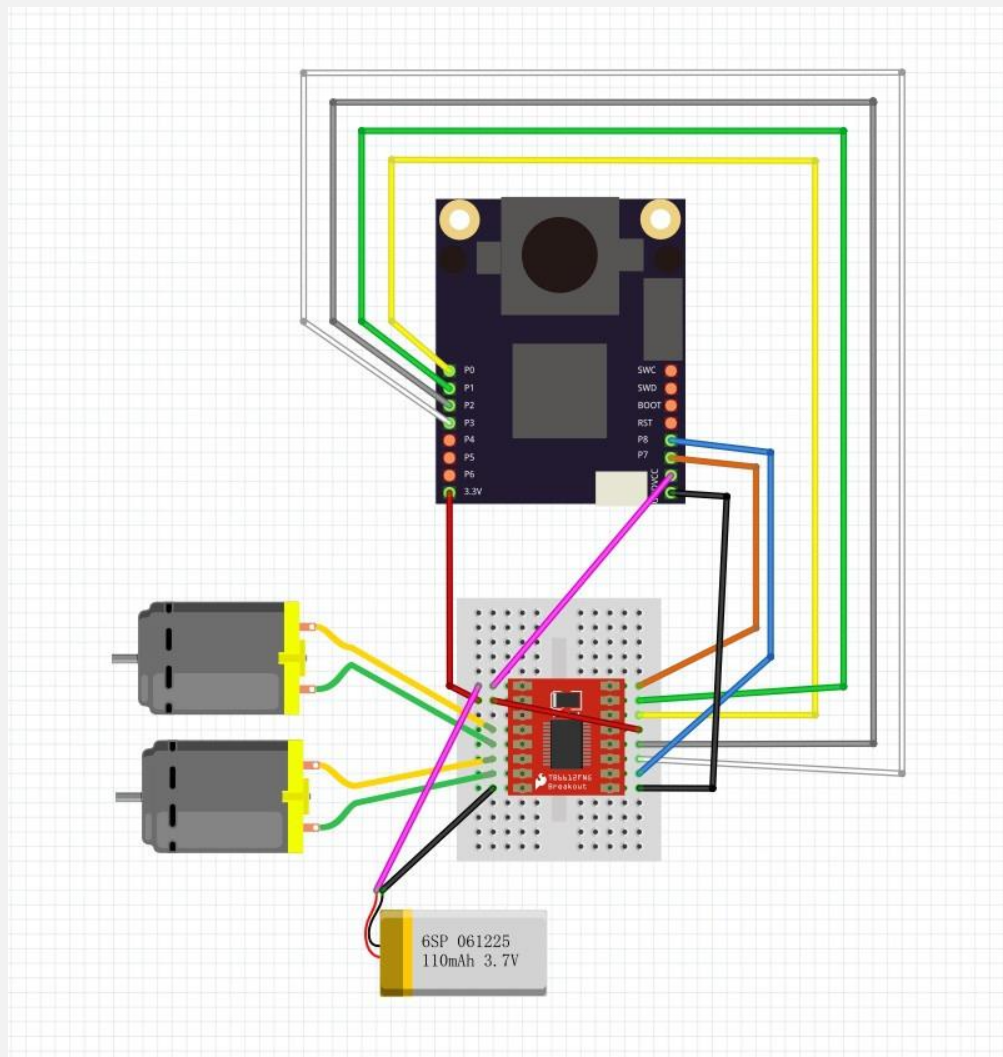


M3*20螺丝螺母



M2*4自攻螺丝

电路连接示意图



软件部分

- 1、编写小车控制模块car.py，并将car.py保存到OpenMV中；
- 2、通过调整参数看一下小车是不是向前走，如果不是，更改第二行和第三行的inverse_left和inverse_right来将左轮子或者右轮子反转，确保小车是正向前进的；
- 3、编写PID算法，并将pid.py保存到OpenMV中；
- 4、调整PID参数，实现小球的跟随

本课程结束

学以致用，勤学苦练！

同学们，再见！