



MU 视觉传感器 3 高级用户手册



杭州摩图科技有限公司
版本 V1.0
2020.06.15

免责声明和版权公告

•本手册中的信息适用于摩图科技公司所生产的 MU 视觉传感器 3（以下简称 MU3），敬
请留意摩图科技官网 <http://www.morpx.com> 以便获取传感器最新版本的固件与库函数，版本
更新不另行通告。

•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。

•私自维修或改装产品上的电子元件造成损坏，摩图科技将不予以保修。

•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 是摩图科技系列产品的注册商标。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。

目 录

免责声明和版权公告	1
1 前言	4
2 MU3 简介	5
2.1 MU3 的工作原理	5
2.2 MU3 的使用步骤	6
2.2.1 设置通讯模式	6
2.2.2 设置 MU3 的地址	6
2.2.3 线路连接	7
2.2.4 编写代码并烧录	8
2.2.5 运行测试	9
3 MU3 的参数设置	10
3.1 摄像头帧率	10
3.2 LED 颜色	10
3.3 数码变焦与视野关系	11
3.4 白平衡的作用	11
3.4.1 白平衡模式	11
3.4.2 白平衡的校准方法	12
3.4.3 锁定模式的使用场景	13
3.5 算法性能设置	14
3.6 光源的影响	15
3.7 物体大小与检测距离的关系	16
4 视觉算法简介	188
4.1 图像坐标系与检测结果	188
4.2 色块检测与颜色识别的区别	20
4.2.1 颜色分类标签的定义	20
4.2.2 色块检测框的作用	20
4.2.3 色块检测的调用方式	21
4.2.4 颜色识别区域的作用	21
4.2.5 颜色识别的调用方式	22
4.3 球体算法与背景干扰	23
4.4 人体算法的有效特征	24
4.5 卡片算法的有效角度	25
5 数据处理方法	27
5.1 判断是否检测到目标物体	27
5.2 分类标签 Label 的使用	27
5.2.1 分类标签的基本调用方式	27
5.2.2 无人驾驶小车	28
5.2.3 数字乐器演奏	29

5.3 同一支撑面上判断物体方位	30
5.4 同一支撑面上判断物体远近	30
5.5 三维空间中判断物体方位	32
5.6 三维空间中判断物体远近	33
5.7 视觉循线-单线	34
5.8 视觉循线-模拟道路	37
5.9 颜色识别算法实现循线	40
5.10 二维空间重分割	41
5.10.1 分割的方法	41
5.10.2 轨迹识别	42
5.10.3 图案扫描	45
5.10.4 图案识别	45
5.11 三维空间坐标系	47
5.11.1 三维空间密码锁	48
5.11.2 三维空间音乐盒	48
5.11.3 三维空间绘画	49
5.12 同时开启多种算法	49
5.12.1 循线加上颜色识别	49
5.12.2 多类卡片同时识别	50
5.12.3 利用颜色识别物体	51
5.13 算法的衍生	52
5.13.1 利用颜色识别来实现加减乘除	53
5.13.2 利用乒乓球实现任意物体跟踪	53
5.13.3 利用形状卡片指示篮球框	53
5.14 数据的特殊处理	54
5.14.1 提升检测结果的可靠性	54
5.14.2 消抖措施	55
5.14.3 快速跟踪	55
5.14.4 施密特触发	56
5.14.5 排除异常数据	57
6 WIFI 功能	58
6.1 WIFI 工作模式	58
6.2 WIFI 传输模式	59
6.3 AT 指令	60
6.4 带 WIFI 遥控的监控小车	61
7 接近检测、环境光亮度检测与运动手势检测	63
服务与支持	65
修订历史	66

1 前言

人工智能 AI (Artificial Intelligence) 对于大多数人而言既熟悉又陌生, 熟悉是因为 AI 已经渗入到我们日常生活, 例如手机拍照的人脸识别, 移动支付的二维码识别, 商场里的服务机器人, 家庭中的智能音箱, 或是无人驾驶汽车等。陌生却是因为 AI 技术太深奥, 似乎只有专家或是学者才关心其技术本身。

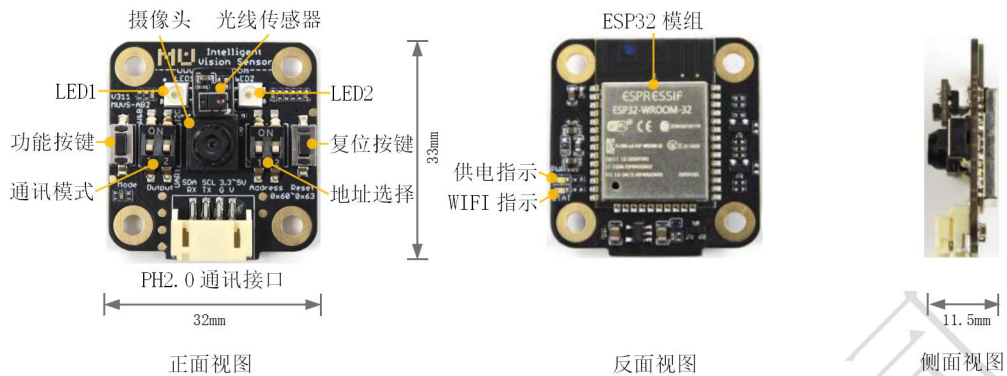
随着时代的变革发展, AI 技术与先进的 STEAM 教育产生融合, 目前市面上各种与 AI 相关的教育用产品和课程开始涌现, 各类赛事也纷纷设置 AI 相关的项目。2018 年与 2019 年教育部先后发文鼓励 AI 教育走进高校以及小学中学校园, 毋庸置疑 AI 教育会成为教育行业另一个新的发展方向。

我司的 MU (目) 系列视觉传感器就是为了满足青少年 AI 教育需求而推出的, 内置多种实用的视觉算法, 不仅识别率高, 稳定性好, 并且我司提供固件与配套驱动库的持续升级。此外, 在使用过程中, 用户无须关注算法内部的处理过程, 更无需进行复杂的参数调整, 可以将更多的精力和时间投入到作品和产品的设计之中。

MU 视觉传感器的本质是“传感器”, 它是通过“看”来感知世界的, 其输出的数据还需要进行一定的“人工处理”。如何灵活应用这些数据设计出有创意的作品, 或者解决生活中的问题, 将会是很有意义的事情。至于视觉算法的基本原理可以有所了解, 但不必“掌握”, 青少年接触算法原理的目的不是为了创造新算法, 而是为了了解不同算法的局限性, 想办法消除其局限性, 让视觉算法更高效稳定地工作, 同时拓展自己的思维, 开阔眼界, 这才应当是青少年 AI 教育的出发点。

本手册将介绍 MU 视觉传感器的一些基本的视觉算法知识、图像处理的知识、数据处理方法以及一些典型的应用案例。相信本手册能够让用户更深入的理解 MU 视觉传感器, 并且解答一些常见的疑惑与问题。

2 MU3 简介

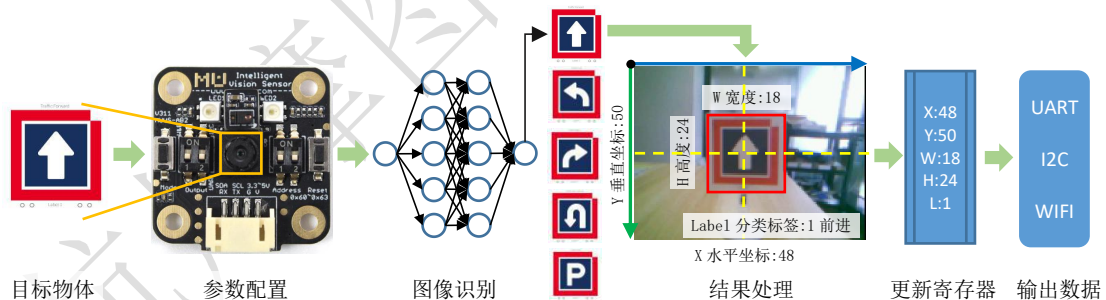


MU 视觉传感器 3 代 (以下简称 MU3) 是一款集成了多种视觉算法及传感功能的边缘计算视觉传感器, 采用 ESP32 双核处理器, 配有一颗 VGA85°无畸变广角镜头, 具有识别速度快、视野广、体积小、功耗低等特点。MU3 的视觉算法全部由摩图科技自主研发, 目前最新的 115 版固件包含了色块检测、颜色识别、球体检测、人体检测, 以及形状、交通、数字卡片检测。摄像头上方是一颗高度集成的红外传感器, 具备环境光强度的检测、红外测距、运动手势识别三种功能。MU3 支持 UART、I2C 以及 WIFI 三种通讯方式, 通过 WIFI 可以传输实时图像以及控制命令。

为了满足青少年 AI 教育的需求, MU3 可与 Arduino、Micro:Bit 以及掌控板配合使用, 支持 C、microPython*以及图形化编程, 支持 Arduino、Makecode、MuEditor、Mixly、Mind+、mPython X 等诸多开发环境。

注: microPython 库函数目前仅支持 Micro:Bit

2.1 MU3 的工作原理



MU3 的视觉识别或者检测功能完全在处理器内部完成, 无须借助云端服务器, 通过简单的寄存器配置即可开启算法。算法开启后, 处理器不间断地处理摄像头所拍摄到的图像, 如果目标物体出现在图像中且其大小、角度、清晰度等可以满足识别的要求, 那么处理器就会将该物体在图像中的相对位置、大小、分类编号等数据写入寄存器, 每处理完一帧图像寄存器就会被更新一次, 通过读取这些寄存器即可获得识别的结果。

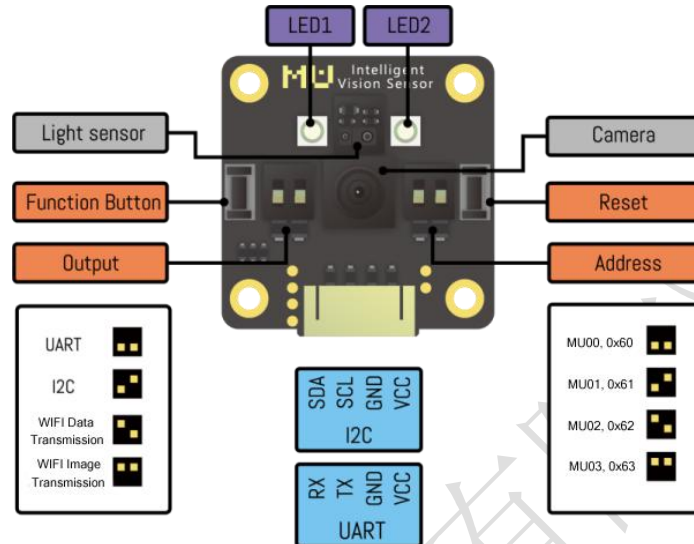
针对 Arduino IDE, 我们提供了完整的 C 语言库函数来实现对 MU3 的操作, Makecode、Mixly、Mind+、mPython X 等 IDE 则提供了可视化编程模块。使用 C 语言编程能更好地发挥 MU3 的性能, 而为了操作简单容易上手, 可视化编程模块对 MU3 的部分功能做了简化处理。

2.2 MU3 的使用步骤

以通过 Micro:Bit 开启人体检测算法为例进行介绍。

2.2.1 设置通讯模式

MU3 支持 4 种通讯模式：UART，I2C，WIFI 数据传输，WIFI 图像传输。根据所需要的通讯方式，拨动 MU3 左侧的 Output 拨码开关。



输出模式	拨码开关	编号	LED 指示	简介
UART		00	闪烁红色	支持 9600 (默认) ~ 921600 波特率*, 8 位数据位, 1 位停止位, 无校验。基于 MU-Protocol 协议进行数据读写, 波特率较低时会降低图像识别帧率
I2C		01	闪烁绿色	最高支持 400Kbps, 直接对寄存器读写操作, 使用较为灵活, 效率高
WIFI 数传		10	闪烁黄色	使用 WIFI-UDP 方式进行数据通讯, 基于 MU-Simple-Protocol 协议以指令的方式来读写数据, 可以用于物联网应用
WIFI 图传		11	闪烁紫色	通过浏览器来查看图传数据以及发送遥控指令, 浏览器中输入地址 192.168.4.1 即可查看图像

备注：因 Micro:Bit 通讯性能限制，Micro:Bit 仅支持 9600 波特率，建议使用 I2C 模式；

选择通讯方式后，程序中的通讯方式应与拨码开关保持一致。编程时应首先配置通讯方式，然后才可以进行其他的参数配置，使用过程中不可更改，每次切换通讯方式，需要重启 MU3。



初始化时配置通讯方式

2.2.2 设置 MU3 地址

MU3 支持 4 个地址：0x60（默认），0x61，0x62，0x63。当 MU3 与其他传感器地址冲

突时需要进行更改。I2C 模式下支持多个不同地址的 MU3 协同工作，可以给 MU3 分配不同的地址。一般情况下保持默认地址即可。

设备地址	拨码开关	编号	设备地址	拨码开关	编号
0x60		00	0x62		10
0x61		01	0x63		11

选择设备地址后，代码中的设备地址应与拨码开关保持一致。编程时应首先配置设备地址，然后才可以进行其他的参数配置，使用过程中不可更改，每次切换设备地址，需要重启 MU3。



程序中的设备地址应保持一致

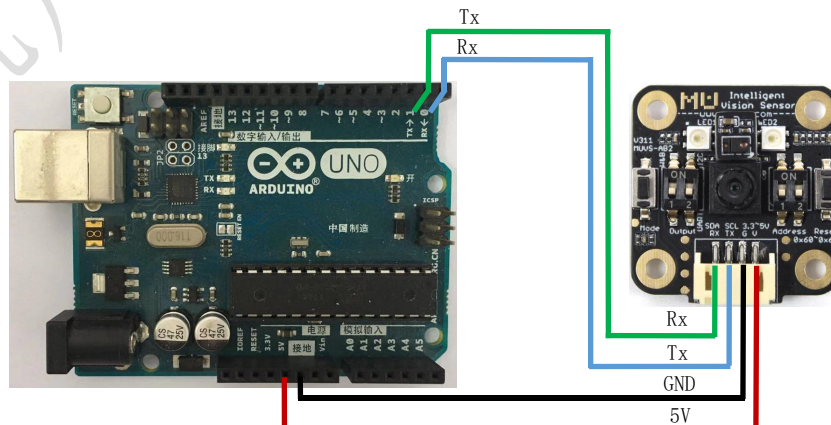
2.2.3 线路连接

线路连接以 Arduino UNO 和 Micro:Bit 为例进行介绍，其他主控板连接方式与示例类似。

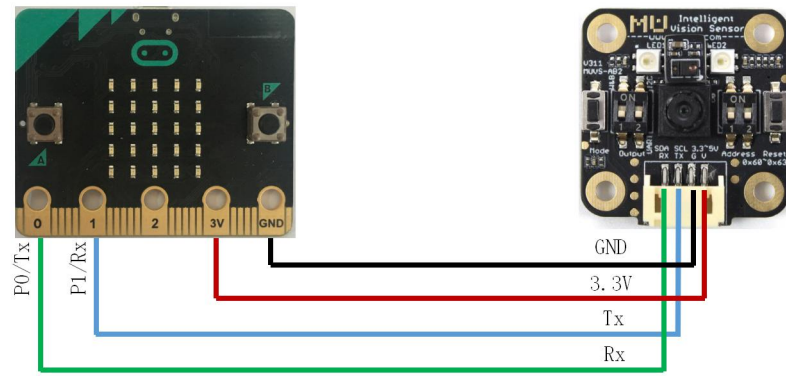
UART 模式：

MU3-Rx 接主控板 Tx，MU3-Tx 接主控板 Rx，（也可使用自定义的软串口连接）

MU3-G 接主控板 GND，MU3-V 接主控板 5V



MU3 与 Arduino 连接，UART

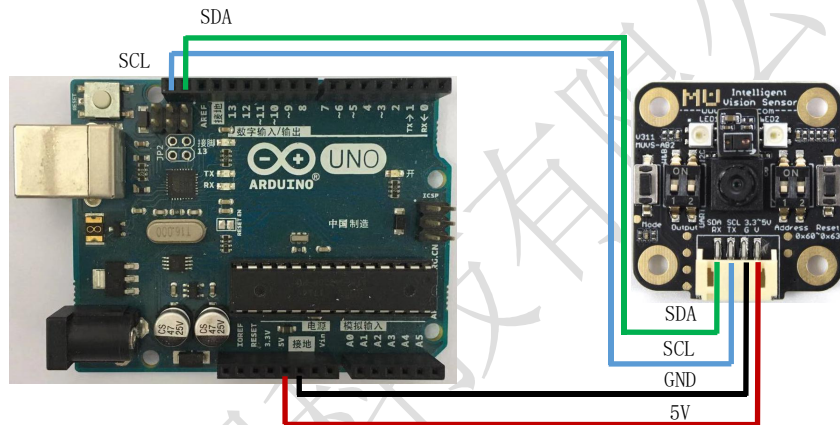


MU3 与 Micro:Bit 连接, UART

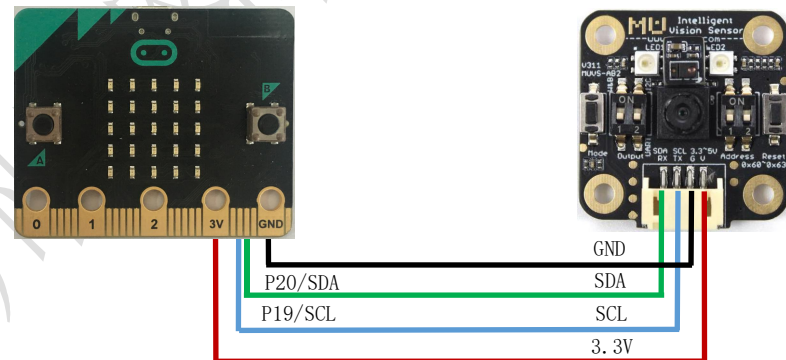
I2C 模式:

MU3-SCL 接主控板 SCL (或 C, CLK) , MU3-SDA 接主控板 SDA (或 D, DATA)

MU3-G 接主控板 GND, MU3-V 接主控板 3.3V 或 5V



MU3 与 Micro:Bit 连接, I2C

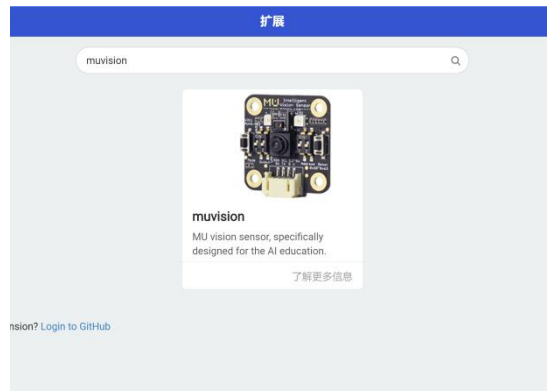


MU3 与 Micro:Bit 连接, I2C

2.2.4 编写代码并烧录

打开浏览器, 进入 Micro:Bit 在线编辑页面 <https://makecode.Micro.Bit.org>

进入左侧边栏>高级>扩展, 搜索关键字 “mu” 或者 “muvision” 或者 “AI” 等, 从列表中选择 muvision, 然后加载库。



Micro:Bit 加载 MU3 库函数

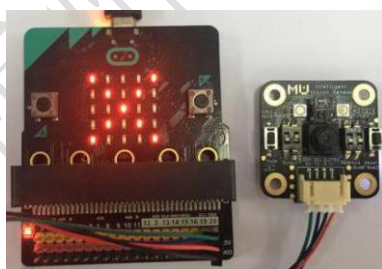
返回编辑页面并编写一个简单的人体检测程序（I2C 模式），代码如下：



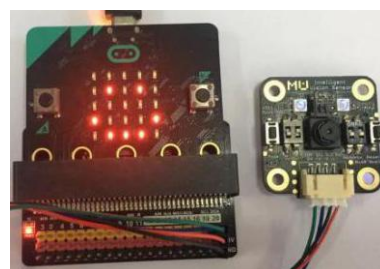
Micro:Bit 的人体检测代码

2.2.5 运行测试

将生成的 Hex 文件烧录至 Micro:Bit 中并运行程序。程序开始后，MU3 板载的两颗 LED 首先会闪烁一下白色，表示算法已开启，然后开始闪烁红色或蓝色，其中红色代表未检测到，蓝色代表检测到。与此同时，如果 MU3 检测到，则 Micro:Bit 屏幕上会显示笑脸，否则显示叉号。



未检测到人体时，显示叉号 X



检测到人体时，显示笑脸：)

3 MU3 参数设置

MU3 提供了简单而实用的参数设置功能：摄像头帧率、LED 的颜色、数码变焦、白平衡、算法性能设置。合理设置这些参数可以提升 MU3 的识别效果。

3.1 摄像头帧率

摄像头帧率指摄像头采集图片的速度，分为高速模式（默认）和标准模式，高速模式为 50fps，标准模式为 25fps。提高摄像头帧率可以提高图像处理的速度，缓解图像拖影问题，但功耗会有一定增加。环境较暗时，标准模式比高速模式成像要亮一些。默认为高帧率模式。



3.2 LED 颜色

MU3 板载 2 颗 RGB-LED，左侧为 LED1，右侧为 LED2，可以用于指示 MU3 的识别状态，在默认配置下，检测到物体显示蓝色，未检测到物体显示红色。

备注：颜色识别算法不会开启灯光指示功能。



C 语言函数库（即 Arduino 库）还提供了 LED 的高级控制功能，包括手动灯光模式、亮度调节、灯光保持等。在手动模式下可以提高 LED 的亮度，用于补光。



未开启 LED，非常暗的环境



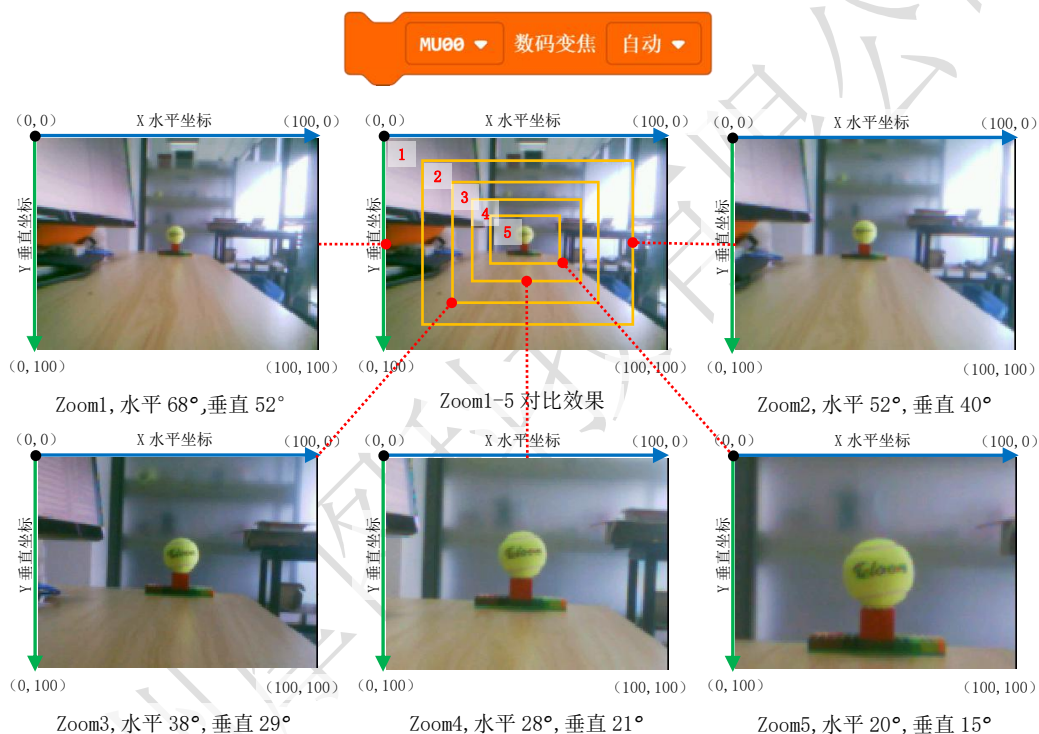
开启高亮白色 LED，非常暗的环境

LED 作为补光灯效果

3.3 数码变焦与视野关系

MU3 提供了数码变焦功能，共有 5 个档位 (Zoom 值) 可以使用，设置该参数可以识别不同距离处的目标。Zoom 值小，则视野广，距离近；Zoom 值大，则视野窄，距离远。

备注：使用 Zoom3 ~ 5 时，图像帧率会比 Zoom1 ~ 2 低。



3.4 白平衡的作用

摄像头在拍摄照片时，由于外界光线或背景的色温不同，图像会产生一定的色偏。为了还原真实的色彩，需要通过白平衡功能调节 R 红、G 绿、B 蓝的增益来纠正这种色偏。但白平衡功能并不总是有利于色彩还原的，合理使用白平衡对于颜色相关的算法是十分关键。

3.4.1 白平衡模式

白平衡的调节总体而言分为自动与固定参数两种方式，对于一张构图颜色均衡的图像，其 RGB 三个通道的值应是分布均匀的，如果出现了失衡，自动白平衡功能会适时的修改各通道的色彩增益值使其重新分布均衡，而固定白平衡功能是不会改变色彩增益的。

MU3 视觉传感器提供了 4 种白平衡模式：自动模式，锁定模式，白光模式，黄光模式。



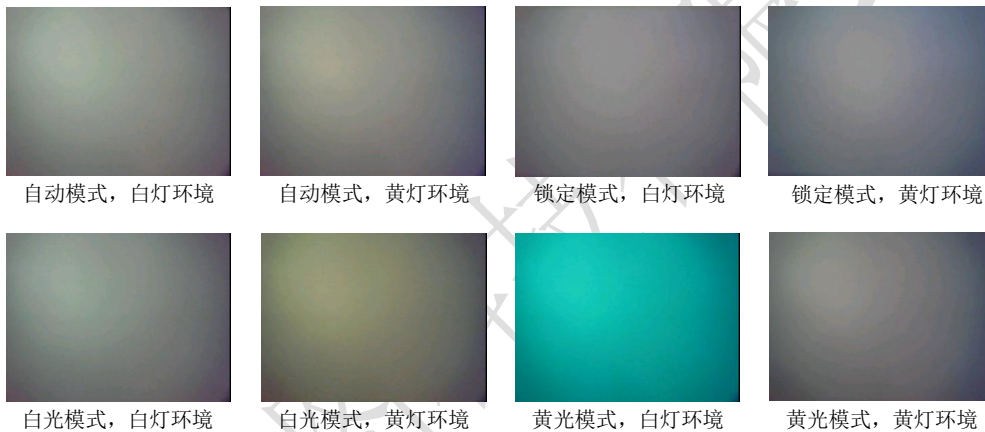
四种白平衡模式

自动模式：适用于光照良好且对颜色要求不高的应用；

锁定模式：适用于颜色还原度要求较高的应用。需要将 MU3 朝向白色平面进行校准，然后锁定白平衡，锁定后颜色不会随环境的改变而变化；

白光模式：适合于白色灯光或者阴天环境中使用，该模式属于自动白平衡模式；

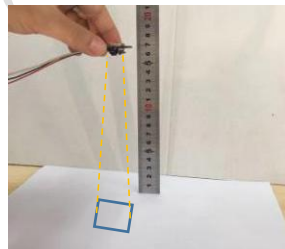
黄光模式：适合于黄色灯光或者阳光环境中使用，该模式属于自动白平衡模式。



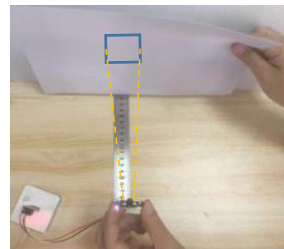
4 种白平衡模式分别在白色灯光和黄色灯光下的测试效果对比

3.4.2 白平衡的校准方法

(1) 将 MU3 面向一张白纸（其他白色面也可）距离 10~20cm，确保摄像头中心区域完全面向白纸，灯光照向白纸，但尽量不要有阴影遮挡；



MU3 垂直向下面向白纸

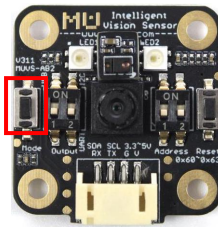


MU3 水平向前面向白纸

(2) 调用锁定白平衡指令，或短按功能按键（MU3 左侧 mode 键）；



方法 1：调用指令方式



方法 2：短按功能按键方式

- (3) MU3 的 LED 会闪烁一下白色灯光，开始白平衡校准；
- (4) 几秒钟后，当 LED 闪烁一下青色灯光后，表明白平衡校准完成；
- (5) 如果 LED 闪烁为紫色灯光，则表明校准失败，需要重新执行一遍。

备注 1：按键方式可以在 MU3 运行的任何时刻使用，而指令模式只适合于开机后的初始化阶段使用；

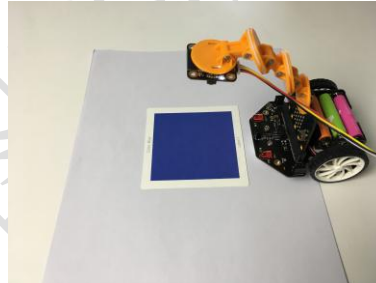
备注 2：如果使用指令校准白平衡，该步骤应是 MU3 初始化的最后一步，并应该设定 2 秒左右的等待时间；

备注 3：如果校准时使用的参照物不是白色，那么即使校准成功了，图像也可能是处于偏色状态的，因为该功能会将其他颜色尽量调节为白色。

3.4.3 锁定模式的使用场景

MU3 的自动模式、白光模式、黄光模式可以满足大部分的应用场景，但是如果图像中存在大面积单色物体时，自动白平衡功能就失去了意义，不仅不能还原色彩，还会将色彩调节失衡。在使用颜色相关的算法时，需要使用锁定模式。

MU3 视野中存在大面积单色物体时：



MU3 视野中有一张蓝色卡片



自动白平衡
大片红色周围呈现淡青色



自动白平衡
大片绿色周围呈现淡紫色



自动白平衡
大片蓝色周围呈现淡黄色



校准并锁定白平衡
大片红色周围呈现白色



校准并锁定白平衡
大片绿色周围呈现白色

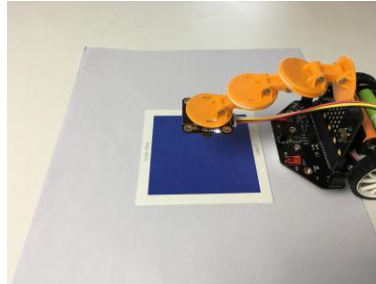


校准并锁定白平衡
大片蓝色周围呈现白色

在上述第一排图像中，当图像中出现大面积红色时，R 通道值明显高于 G 与 B 两个通道的，摄像头为了达到白平衡而会提高 G 与 B 的增益，所以其白框呈现出了淡青色。同理，绿色卡片的白框会呈现淡紫色，而蓝色卡片的白框会呈现淡黄色。

经过校准并锁定白平衡后，三种颜色周围的白色被正确的还原，不存在偏色现象。

MU3 近距离识别颜色时：



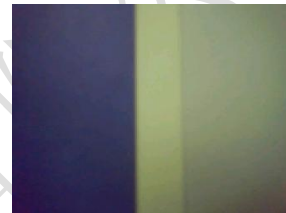
MU3 近距离识别颜色



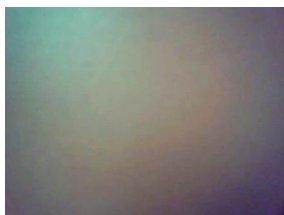
自动白平衡
近距离，图像左半边为红色



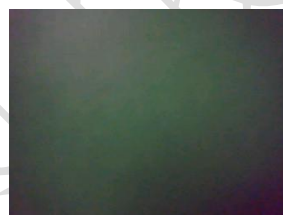
自动白平衡
近距离，图像左半边为绿色



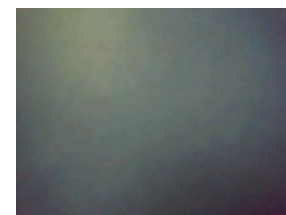
自动白平衡
近距离，图像左半边为蓝色



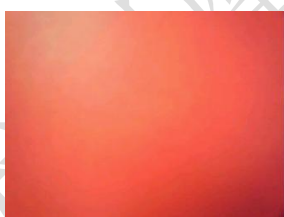
自动白平衡
近距离，图像全部为红色



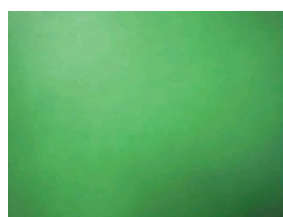
自动白平衡
近距离，图像全部为绿色



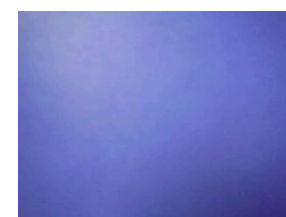
自动白平衡
近距离，图像全部为蓝色



校准并锁定白平衡
近距离，图像全部为红色



校准并锁定白平衡
近距离，图像全部为绿色



校准并锁定白平衡
近距离，图像全部为蓝色

在这一组测试中，摄像头非常接近单色物体，自动白平衡为了还原白色，将原本的红绿蓝卡片调节到完全失衡的状态，此时已经失去了原本的颜色信息，校准并锁定白平衡后，其原本颜色可以得到正确的还原。

3.5 算法性能设置

MU3 视觉算法的运行分为两个阶段：算法处理与结果处理。

算法处理阶段：算法运行前需要设置一些内部参数，参数的不同会直接影响检测效果。另外一个算法是由多个不同的算法单元构成的，这些算法单元分别负责图像预处理、检测、分类、

校验、优化等等。如果这些算法单元全部开启，那么检测结果的准确率是最高的，环境适应性也是最好的，但处理时间会变慢。如果我们放弃一些算法单元，那么算法的整体运行速率可以提高，并且在可控的环境下也能够得到较好的检测结果。

结果处理阶段：算法处理后的数据是原始数据，中间可能夹杂着错误的结果、偶尔的漏报或是误报等问题，所以 MU3 在输出检测结果前，会对数据结果做一定的处理。

根据这两个阶段运行的设置不同，每类算法都有 4 种性能模式选项：自动、速度优先、性能均衡、准确率优先。用户可以根据实际应用需求来选择适合的性能模式：



性能选项	特点	适合场景
自动	自动选择其他三种性能模式	默认的模式
速度优先	通过去掉一定的算法步骤来提高处理速度,在可控环境下依然具有不错的检测效果,数据结果不做任何优化处理,直接输出原始数据	需要高速处理的应用,如实时跟踪某个物体;对某个物体需要做出快速的响应;场地或背景可以控制的环境
准确率优先	以最佳的性能来运行算法,具有最可靠的检测结果,输出结果经过处理,一些异常数据会被优化掉,具有一定平滑性,但有些算法的运行速度会明显降低	需要可靠的检测某个物体的结果;对速率要求不高的应用;场地或背景复杂且不可控的环境
均衡性能	介于速度优先和准确率优先之间的模式	适合一般场景的应用需求

3.6 光源的影响

良好稳定的光源对于视觉识别是一个非常重要的因素，劣质光源、强光直射、暗光、逆光等都不利于识别。



劣质光源的水波纹现象



强光直射造成局部过曝



昏暗环境造成物体颜色失真



逆光造成颜色信息丢失

劣质光源：在某些光源下图像中会出现水波纹，严重的水波纹会影响 MU3 的正常工作。这

一现象与摄像头的曝光时间有关，视野越明亮，曝光时间越短，水波纹也就越明显。可以用手机的相机来检测光源的质量，不合格的光源通常也会导致手机的相机产生水波纹现象；

强光直射：当强光直射到目标物体上时，会造成其局部曝光过度或者反光的现象，此时图像的完整性受到影响，导致无法被正常识别。但通常这个现象只发生在一些特定的角度，有时只需调整摄像头的位置就可解决；

昏暗环境：与人眼不同，摄像头对昏暗环境的适应能力十分有限，因此不能依据人眼来判断是否为昏暗环境。在自动白平衡模式下，摄像头的增益功能可以提高成像的亮度，但同时会引起明显的噪点，另外环境光线微弱，目标物体的颜色无法被正确的还原，两者共同作用会导致识别效果变差，甚至无法识别，因此不建议在昏暗环境中使用，如必须使用，可以使用 LED 补光灯；

逆光环境：当室内的摄像头朝向明亮的窗户时，其就处于明显的逆光环境。逆光会造成颜色信息的严重丢失，影响 MU3 的正常工作。但如果屋内是明亮的，那么逆光问题就会相对较弱，开启 LED 补光灯也可以缓解近距离的逆光问题。

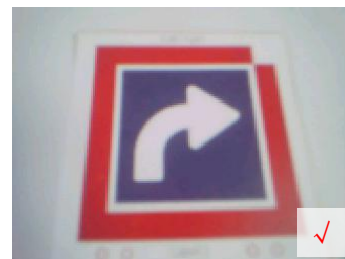
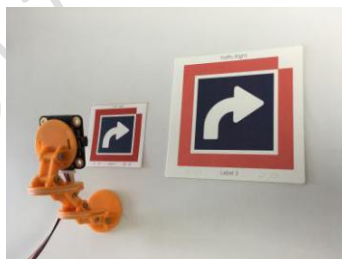
3.7 物体大小与检测距离的关系

目标物体在 MU3 中成像的大小对于其是否能被检测到起到至关重要的作用，我们可以利用数码变焦的高 Zoom 值来检测远距离物体，也可以利用低 Zoom 值来检测近距离物体。但一些特殊情况下如果调节 Zoom 值仍无法满足需求，那么还可以通过放大或缩小目标物体的尺寸来解决。

近距离需要缩小目标物体：当需要近距离检测物体时，如果物体尺寸太大，那么摄像头就可能无法拍摄到完整的图片，自然无法进行识别，此时需要缩小物体的尺寸。



近距离面对大卡片时，无法拍摄到完整的图片



近距离采用小卡片，图像显示正常，可以识别

远距离需要放大目标物体：反之如果被检测物体距离较远，为了能够“看清”远处的物体，则需要适当放大其尺寸。

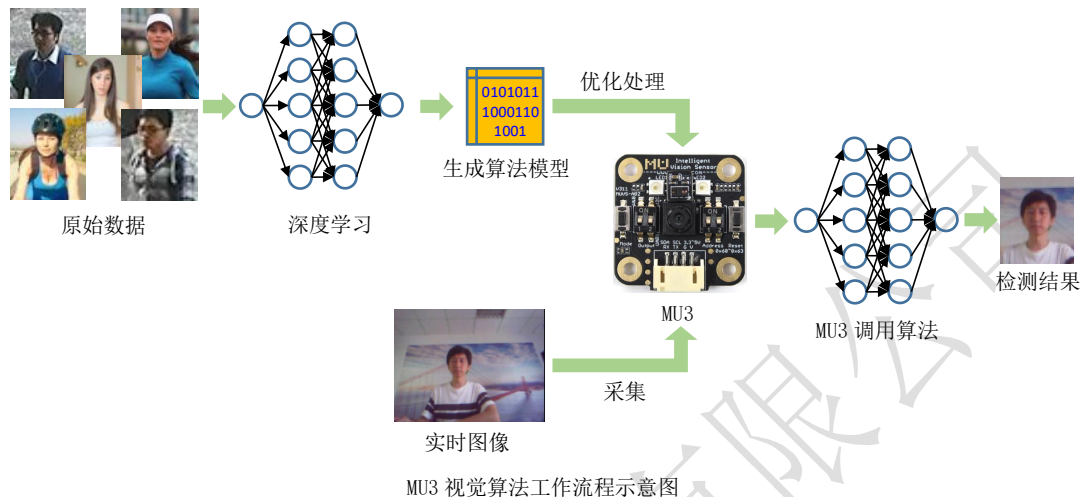


远距离，左侧的小卡片无法被检测到，而右侧的大卡片可以

杭州摩图科技有限公司

4 视觉算法简介

MU3 采用先进的计算机视觉处理技术——深度学习。采用专业的服务器对数十万张图片数据样本进行学习训练，生成适用于边缘计算系统的算法模型，然后经过数轮优化后这些算法模型就可以在 MU3 上快速而稳定的运行。使用时，MU3 会调用这些算法模型实时识别或者检测摄像头采集的图像，然后返回结果。



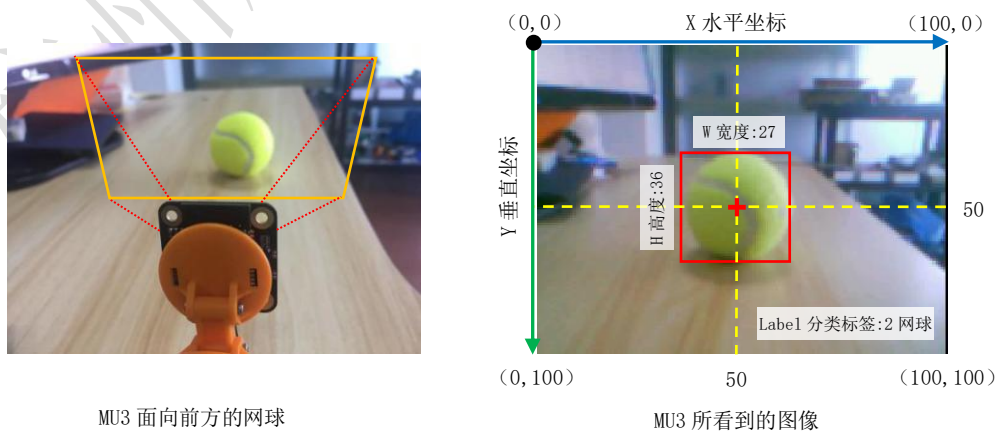
4.1 图像坐标系与检测结果

生活中常用坐标系的有：1 维坐标系（线段），2 维坐标系（平面），3 维坐标系（空间），很显然图像属于 2 维坐标系，将其两个坐标轴分别表示为 X（水平坐标，横向坐标），Y（垂直坐标，纵向坐标）。

与教材中常用的 2 维坐标系有所不同，图像坐标系的左上角为坐标原点，这是因为在显示一幅图像时，像素点总是从左上角开始，从左至右，由上而下逐行的向下显示数据的。接下来以球体算法为例，讲解坐标系的相关概念以及各检测结果的含义。

如下面的两幅图所示，当 MU3 检测到位于其前方的网球时，会返回如下的数据：Status 是否检测到，X-Y 坐标，W-H 宽高，Label 分类标签。

备注：MU3 还有检测结果数量的信息，但目前算法只支持单物体的检测或识别，所以这个值目前始终为 1。



Status 检测状态：当 MU3 检测到目标物体时，该值输出为“真 True”，否则为“假 False”。编程时，首先应该检查是否检测到目标物体，然后再去处理坐标数据或是宽高数据，否则就继续

检测，这样可以提高运行效率。

X-Y 坐标：在 MU3 图像坐标系中，每个点对应的水平坐标与垂直坐标均被量化到了 0~100，表示该点相对于完整图像的百分比位置。例如 X 为 30 时，表示图像中水平方向上该点与原点之间的距离为整个图像宽度的 30%。

W-H 宽高：当 MU3 检测到物体后，还会返回物体的宽度值（W）和高度值（H），同样被量化到了 0~100，W 表示图像中物体宽度与图像宽度的百分比值，H 表示图像中物体高度与图像高度的百分比值。需要注意的是，MU3 摄像头输出图像的宽高比例为 4:3，因此对于正方形物体而言，MU3 输出的宽高值并不相等，存在 3:4 的比例关系。

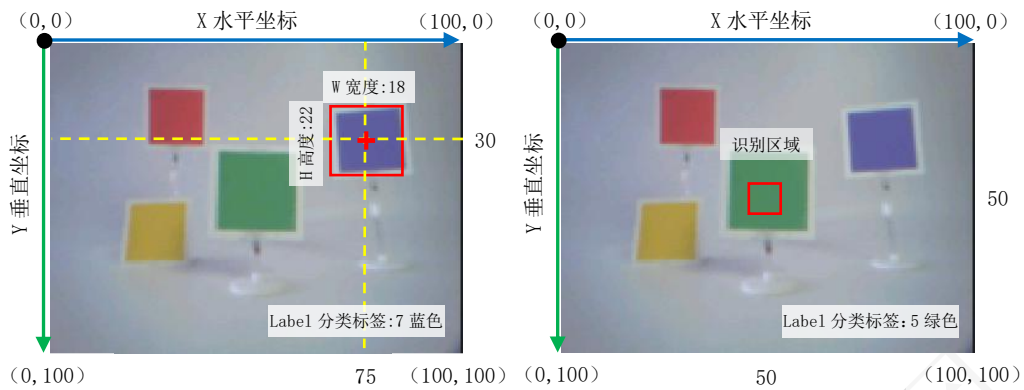
Label 分类标签：具有分类识别功能的算法会根据所检测到的不同物体而输出其对应的分类标签。分类标签是 MU3 内部预设定好的一系列数字，每一个数字对应一个特定的物体，以球体算法为例，当 Label 标签为 1 时，代表检测到了乒乓球，而如果 Label 标签为 2，则说明检测到的是网球。Label 是颜色算法和卡片的算法重要信息。

示意代码：



当检测到球体时，依次输出各个检测结果

4.2 色块检测与颜色识别的区别



色块检测: 当设定检测蓝色块时, MU3 会在图像中寻找满足条件的蓝色色块

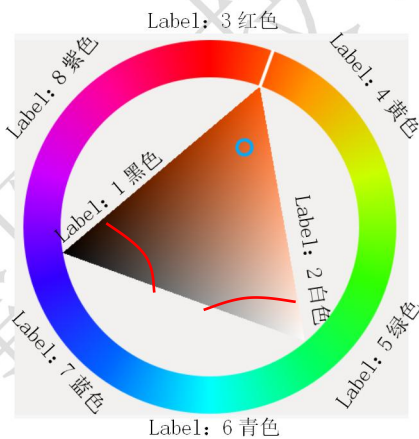
颜色识别: 设定识别区域位于图像中心, 图中识别结果为绿色, Label15

色块检测: 指定某种颜色和色块尺寸的最小值, MU3 在图像中寻找符合条件的一个色块区域, 返回其中心坐标与 W-H 值, 解决“是不是”的问题。

颜色识别: 指定一个检测区域, 然后 MU3 统计该区域中像素的颜色, 并返回该颜色的 Label 值以及 RGB 每个通道的值, 解决“是哪一个”的问题。

4.2.1 颜色分类标签的定义

MU3 共定义了 8 种颜色, 其分布区间如下图所示



颜色 Label 分类标签分布图

颜色算法分类标签表

分类标签	目标物体	图案	分类标签	目标物体	图案
1	黑色 (深灰色)		2	白色 (浅灰色)	
3	红色		4	黄色 (橙色)	
5	绿色		6	青色 (蓝绿色)	
7	蓝色		8	紫色	
0	未知	N/A			

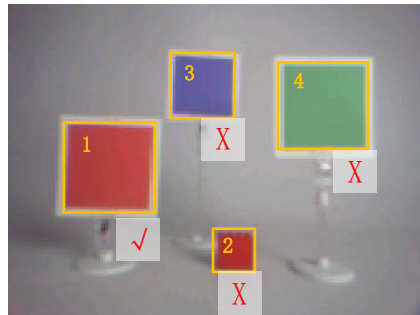
4.2.2 色块检测框的作用

在使用色块检测算法时, 需要设置目标尺寸的最小值, 只有颜色符合并且尺寸不小于设定值的色块才会被检测到。通过合理设置条件可以排除背景中杂色的干扰。

色块检测框有 3 个参数可以设定：检测框最小宽度（默认 10），检测框最小高度（默认 10），颜色标签（默认 3：红色）。



尺寸设置积木



四个色块对比

假设我们要检测宽 15、高 20 的红色色块，那么在上图的 4 个区域中

- 区域 1：可以检测到，满足尺寸大小的需求
- 区域 2：无法检测到，色块尺寸太小
- 区域 3：无法检测到，颜色不符
- 区域 4：无法检测到，颜色不符

4.2.3 色块检测的调用方式



当检测到绿色色块时通过串口输出数据

色块检测在不更改设置的情况下只能检测一种选定的颜色，示例程序指定绿色色块，其他参数保持默认，MU3 会寻找视野中符合条件的色块并且返回最优目标的相关信息，如果视野内存在多个满足条件的目标。MU3 不支持多目标输出。

4.2.4 颜色识别区域的作用

在使用颜色识别算法时，需要指定一个识别区域，算法运行后返回该区域的颜色信息。识别区小，像素少，速度快，容易因个别像素的偏色导致可信度降低；识别区域大，像素多，速度慢，容易受到背景的干扰。识别区域应根据实际需求合理设置。

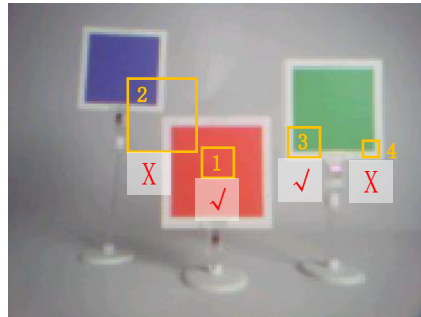
颜色识别区域有 4 个参数可以设置：水平中心坐标（默认 50），垂直中心坐标（默认 50），识别区域宽度（默认 5），识别区域高度（默认 5）



尺寸设置积木



中心设置积木



四个检测区域对比

上图中有四个检测区域：

区域 1：结果可信，识别为红色

区域 2：结果不可信，区域中有多个颜色

区域 3：结果可信，虽然有白色边框干扰，但绿色占据主要部分，仍可识别为绿色

区域 4：结果不可信，识别区域很小，白色边框的干扰不可忽略

4.2.5 颜色识别的调用方式

颜色识别算法输出的结果并不是坐标和宽高，而是颜色的 Label 值以及 RGB 三个通道的值（范围 0~255）。在使用颜色识别时可以只指定识别区域的坐标，其余参数保持默认值。



识别图像中心区域颜色并通过串口输出



识别图像中心区域，判断颜色



当识别不同区域时，需要多次识别以确保结果可靠

4.3 球体算法与背景干扰

MU3 的球体识别算法并不能识别所有球体，仅支持橙色乒乓球 (Label: 1) 和绿色网球 (Label: 2) 的识别。

注：实际上类似的物体也可被检测到，譬如小金桔、青桔等。

球体算法分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	橙色乒乓球		2	绿色网球	

背景对球体识别的影响：任何识别算法都容易受到背景的干扰，球体算法更是如此。在使用该算法时，背景与球体应具有明显的对比度，光源应为白色，并且场地内不应出现与乒乓球和网球类似物体，以免误报。



高对比度，有利于识别



低对比度，不利于识别



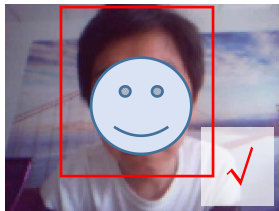
低对比度，不利于识别

球体算法的特殊性：球体算法分为搜索阶段和跟踪阶段，在球体未检测到时 MU3 处于搜索阶段，此时处理速度较慢，检测距离较近。一旦检测到球体后，MU3 就会进入跟踪阶段，处理速度会提升，检测距离变远。一旦跟踪失败，MU3 会再次进入搜索阶段。

4.4 人体算法的有效特征

人体检测实际是检测人的上半身特征—由头部（必须包含人脸）和肩膀所有构成的特征组合。如果图像中出现符合该特征的物体，那么算法就会判断检测到的人。

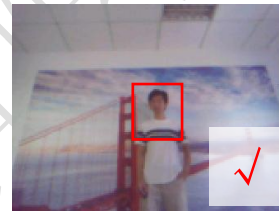
当正对摄像头时具有最佳的检测角度：



近距离，Zoom1，30cm

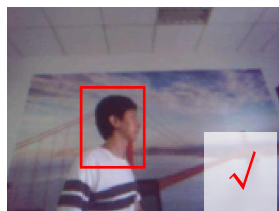


中距离，Zoom1，90cm

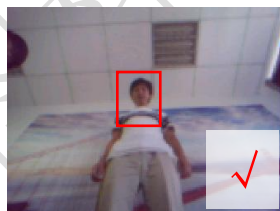


远距离，Zoom1，150cm

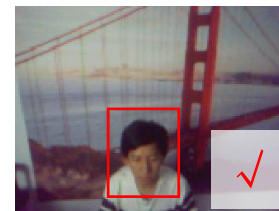
以下几种情况也可以检测，但效果会受影响：



侧视



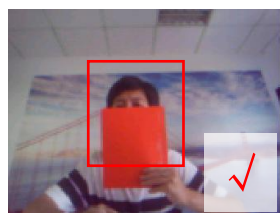
仰视



俯视



遮住衣物

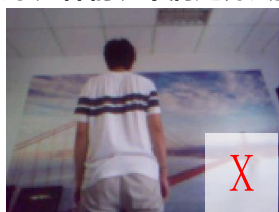


遮住部分脸

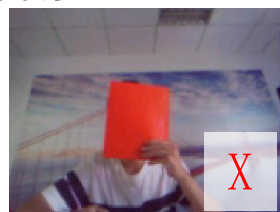


遮住头发

以下几种情况不满足算法的特征要求：



背对摄像头



完全遮住脸



只有身体

4.5 卡片算法的有效角度

MU3 可以识别 3 类定制的卡片：形状卡片、交通卡片与数字卡片，每组卡片都有对应的识别算法。同一时间每个算法只能返回单个目标的检测结果，但这 3 类卡片的识别算法可同时运行。



卡片相关的三个算法

形状卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	对号图案		2	叉号图案	
3	圆形图案		4	方形图案	
5	三角形图案		0	无效	

交通卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	前进		2	左转	
3	右转		4	掉头	
5	停车		0	无效	

数字卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	数字 1		2	数字 2	
3	数字 3		4	数字 4	
5	数字 5		6	数字 6	
7	数字 7		8	数字 8	
9	数字 9		0	数字 0	

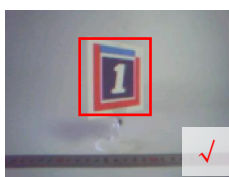
卡片的识别角度：当卡片正对摄像头时具有最佳的检测视角，如果卡片倾斜、旋转、俯仰在一定角度范围内，也是可以识别的，只是在这些情况下，检出率与准确率会下降。



最佳识别角度，正对摄像头



左倾斜，最大 40°



右倾斜，最大 40°



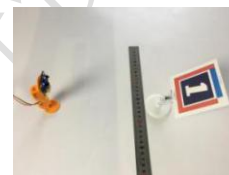
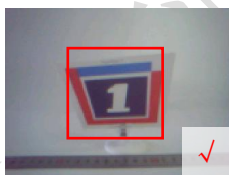
左旋转，最大 20°



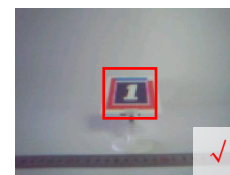
右旋转，最大 20°



前倾，最大 30°



后仰，最大 40°



5 数据处理方法

传感器数据：MU3 直接输出的检测结果我们称之为传感器数据。

传感器数据经过简单的逻辑判断即可满足一般的应用需求，如果这些数据能够被进一步加工，MU3 就可以激活诸多创意应用。本章节将深入介绍一些典型的数据处理方法。

5.1 判断是否检测到目标物体

最简单的 MU3 应用就是判断目标物体是否被检测到，当检测到目标物体时执行某个动作，否则执行另一个动作。

示意代码：



检测到人体时，串口输出“检测到人体”，否则输出“未检测到人体”

特殊技巧：



利用循环方式实现：直到检测到人体为止

利用循环方式实现：直到人体消失为止

5.2 分类标签 Label 的使用

检测到目标物体后，可以通过读取分类标签 Label 来判断具体属于哪一个分类。利用分类标签可以做更多细化的应用。

5.2.1 分类标签的基本调用方式



初始化



直接判断球体 Label 标签，如果为 1 表明是乒乓球，如果 Label 为 2 表明是网球

5.2.2 无人驾驶小车

交通卡片有 5 张，分别是：1 前进，2 左转，3 右转，4 掉头，5 停车。通过调用可视化编程中的卡片模块可以很方便的进行判断，然后控制小车的行进。



利用交通卡片实现无人驾驶

示意代码：



初始化，启动卡片识别算法



根据检测到的卡片，执行相应的动作

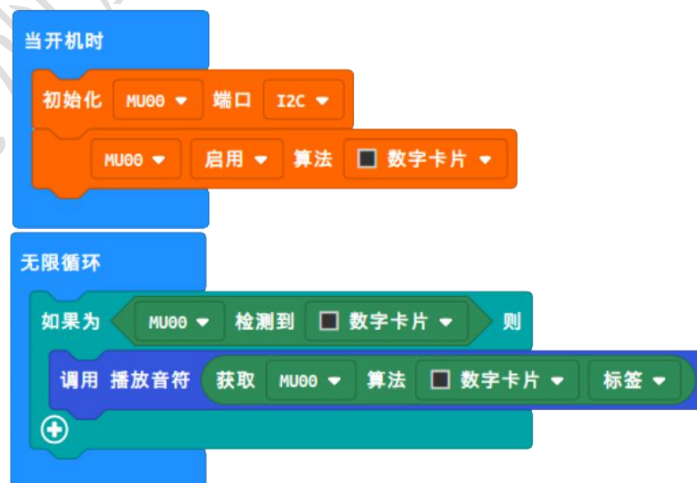
5.2.3 数字乐器演奏

数字 1~7 的卡片分别代表 7 个音符 Do、Re、Mi、Fa、Sol、La、Si，通过 MU3 识别数字卡片，然后主控播放对应的音符。



利用数字卡片实现乐曲演奏

示意代码：

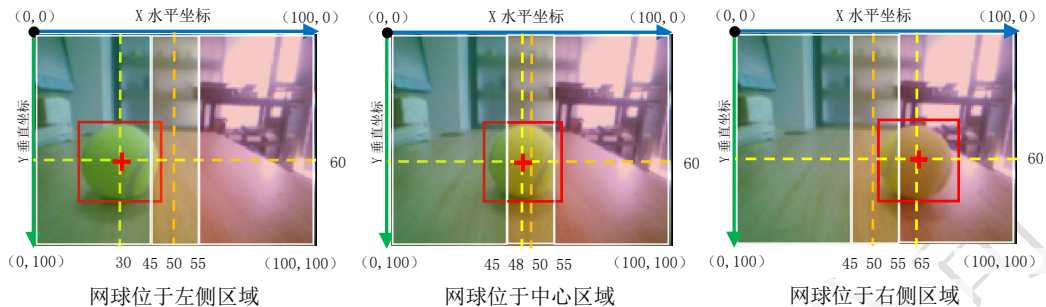


看到卡片，播放对应的音符

5.3 同一支撑面上判断物体方位

同一支撑面：指目标物体与 MU3 位于同一平面上，比如同一桌面，被测物体只在该平面上前后左右的运动，不会脱离该平面。

当目标物体的水平坐标 $X=50$ 时，表示其位于 MU3 的正前方；当 $X<50$ 时，表示其位于 MU3 的左前方；当 $X>50$ 时，表示其位于 MU3 的右前方。

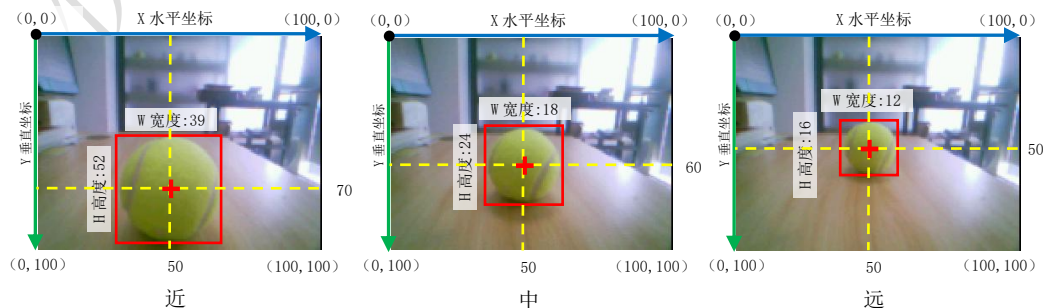


实际中，即使目标物体静止不动，MU3 输出的检测结果也会小幅度变化，即数据抖动，如果简单地将 $X=50$ 作为判断左右的分界点，那么很可能发生物体位置忽左忽右变动的问题，此时需要做消抖处理。可将中心区域的坐标值设定为一个区间，比如 $45 \sim 55$ ，当 X 位于该区间内则认为目标是处于图像水平方向上的中心区域。相对应的左侧区域水平坐标范围变为 $0 \sim 45$ ，右侧区域的变为 $55 \sim 100$ 。



通过判断 X 水平坐标来控制左右转弯

5.4 同一支撑面上判断物体远近



利用 W-H 宽高值判断远近：根据透视原理，视野中的物体存在近大远小的现象，因此可以通过 W-H 宽高值来判断物体的远近。该判断方法容易理解，使用方便，适用于 MU3 直视前方

且目标物体距离较近的情况，因为在此范围内物体的 W-H 宽高值变化比较明显。当物体距离较远时，W-H 宽高值的变化幅度很小，难以准确分辨。

目标物体的远近关系需要根据其实际大小和应用场景而定，对于上图中网球，我们可以定义 $W < 35$ 为远， $W > 50$ 为近， $35 \leq W \leq 50$ 为适中距离。



通过判断 W 宽度来控制前进后退

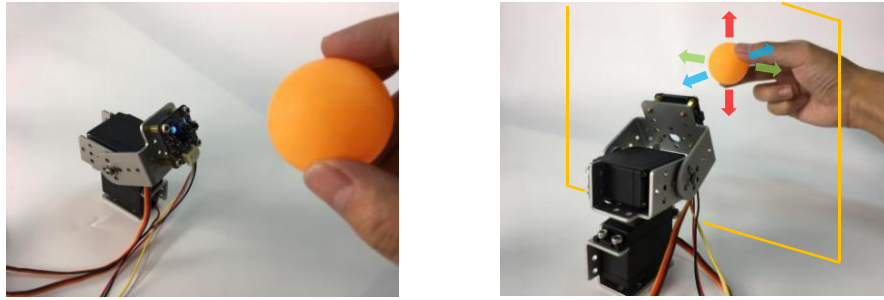
利用 Y 垂直坐标判断远近：观察上面的 3 张图可以发现这样一个规律：当物体接近 MU3 时，其中心位置不断向下移动（Y 值不断增大）；反之当物体远离 MU3 时，其中心位置不断向上移动（Y 值不断减小），该规律适用于 MU3 俯视目标物体。如果 MU3 仰视目标，则情况相反，物体靠近 MU3 时中心位置不断上移（Y 值逐渐变小），反之远离时中心点向下方移动。



俯视角度：通过判断 Y 垂直坐标来控制前进与后退

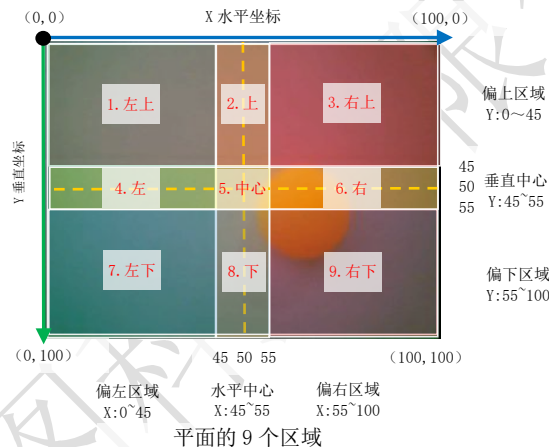
5.5 三维空间中判断物体方位

以最常见的舵机云台为例进行说明。



用小球控制舵机云台上下左右跟随

将图像的水平轴和垂直轴分别划分为 3 个区间，分别是水平方向的左侧区域 (X: 0 ~ 45)，水平中心区域 (X: 45 ~ 55) 与右侧区域 (X: 55 ~ 100)，以及垂直方向的上方区域 (Y: 0 ~ 45)，垂直中心区域 (Y: 45 ~ 55) 以及下方区域 (Y: 55 ~ 100)，这样整幅图像就被分割为以下 9 个区域：



控制一个双轴云台跟踪小球，本质上就是控制水平方向与垂直方向两部舵机的运动，使得图像中球的中心点进入上图中“5.中心”区域。程序思路：球当前位于哪个区域，就控制舵机朝向该区域转动，每个方向每次都分别移动一个小的角度，然后刷新中心点坐标，重复执行直至球的中心点进入“5.中心”区域。

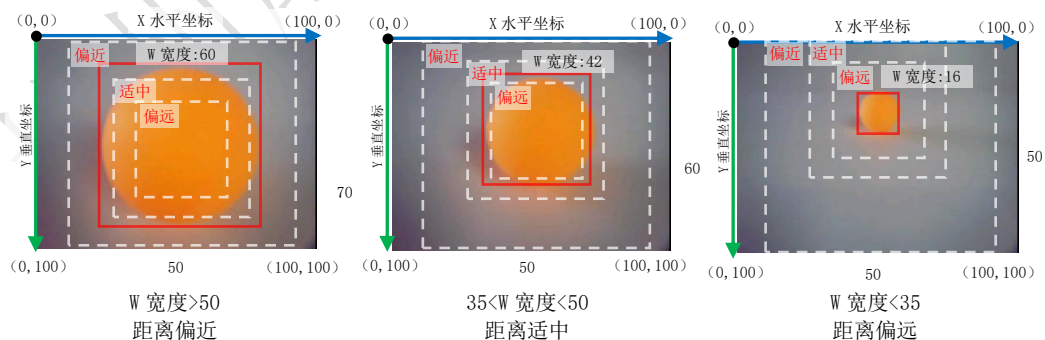


通过判断 X-Y 坐标来控制云台的转动

5.6 三维空间中判断物体远近

根据近大远小的透视原理，可以利用目标物体的宽度 W 或者高度 H 来判断远近，以宽度 W 为例，如果物体距离 $MU3$ 比较近，那么图像中其成像就会较大，即宽度 W 值较大；反之，如果物体距离较远则其成像较小，宽度 W 值较小。

假设一辆装安装了 $MU3$ 的小车，如果希望该辆小车与一只乒乓球之间保持特定的距离，那么首先需要测定在该距离上乒乓球的 W 值，做为参照，然后在车子运动的过程中实时检测乒乓球的 W 值，并与参照值进行比较，如果两者达到一致，那就说明小车与乒乓球之间的距离达到了要求。



为了消除抖动，参照值往往可以设定为一个区间，比如 $35 \sim 50$ ，检测到的 W 值如果在此区间内，则可认为此时车与乒乓球之间的距离满足要求；如果 W 值 < 35，则两者相距较远，需要小车前进；如果 W 值 > 50，则两者相距太近，需要小车后退。



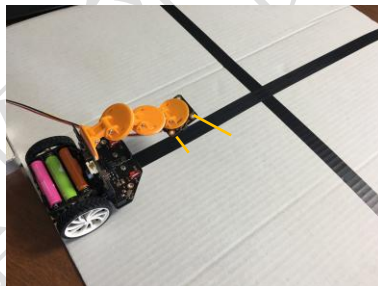
三维空间中，通过判断宽度 W 来控制小车的前进与后退

5.7 视觉循线-单线

利用色块检测算法可以方便地实现循线应用。与传统的红外传感器相比，MU3 不仅可以跟随黑线，还可以跟随彩色线、异形线...下面以跟随单根黑色线为例进行讲解。

(1) **MU3 的视角**：为了实现较好的效果，MU3 的视野中应避免存在干扰因素。如果线条比较细，则需要压低 MU3；如果线条比较粗，则可以适当抬高 MU3。MU3 的视线应当斜向下，不宜垂直于地面，更不应向后看，目的是为了看到前方的路况并做出预判。

注：可将 MU3 切换至 WIFI 图传模式，然后通过手机或者电脑边观察图像边调整角度。



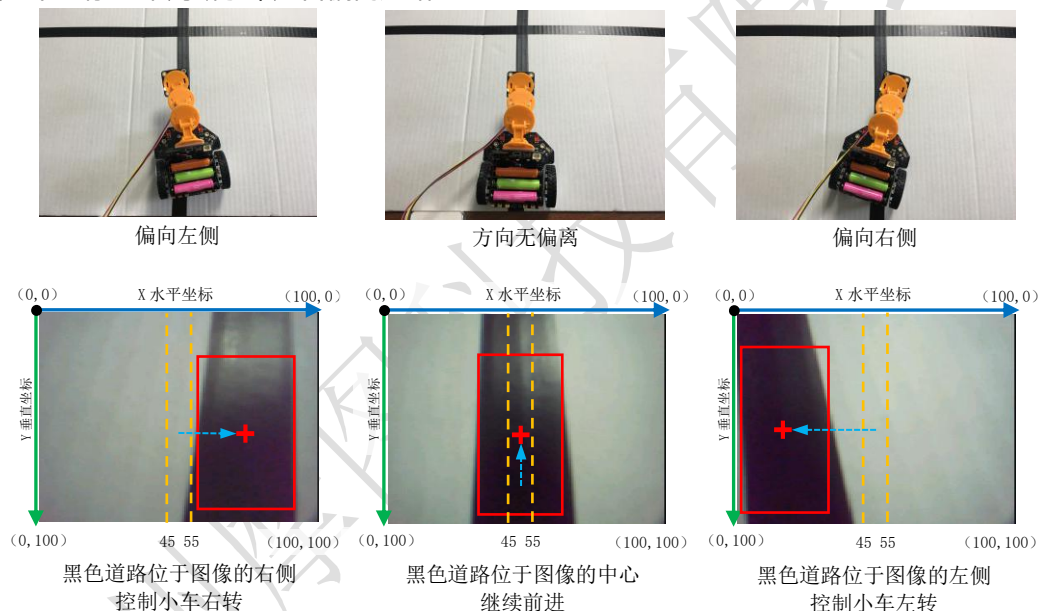
MU3 斜向前看，位置较低

(2) **开启色块检测**：开启色块检测算法时需要校准并锁定白平衡，也可在运行过程中短按 mode 键来校准。另外为了避免 MU3 板载 LED 的灯光产生干扰，需要关闭 LED 显示。为了提高检测帧率，请使用 I2C 通讯。算法性能与高帧率模式保持默认设置。



参考初始化程序

(3) 直线道路识别：当 MU3 视野中出现黑线时，会返回黑色块的坐标和大小，我们利用其水平坐标 X 来判断小车是否偏离道路。



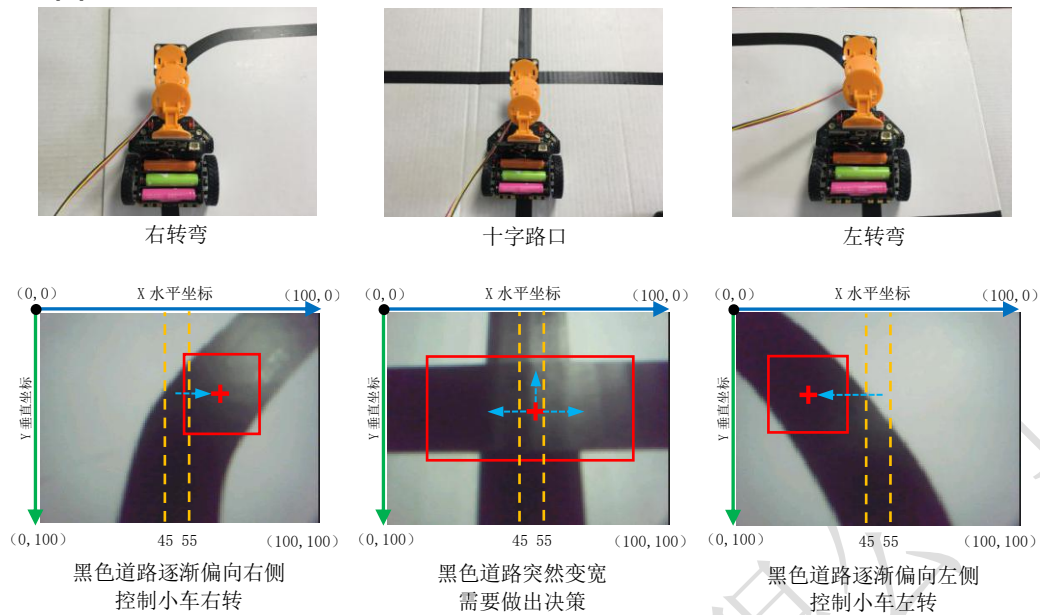
直行：理论上只有当 $X=50$ 时，道路才位于 MU3 视野的中心，此时小车的方向无偏离，可以继续行驶。但实际中为了消除抖动，需要将中点的 X 值设置为一个区间：45 ~ 55，也就是说，只要 MU3 返回的黑色块的 X 值在该区间内，就认为方向无偏离，继续行驶；

偏左：在行驶过程中，一旦小车偏向左侧，黑色块的中心就会偏移到图像的右侧，如果 $X > 55$ ，需要控制小车右转一定角度，直到 X 重返中心区域 45 ~ 55；

偏右：同理当小车偏向右侧时，黑色块的中心就会偏移到图像的左侧，如果 $X < 45$ ，需要控制小车的左转向一定角度，直到 X 重返中心区域 45 ~ 55。

注：为了行驶的平滑性，转动角度不可设置过大，否则容易出现超调的问题，可以结合车速与道路地图多次试验来确定合适的值。想要更好的控制效果，请查阅 PID 算法的相关知识。

(4) 转弯道路识别：同样依据黑色块的水平坐标 X 来判断转弯方向。



右转弯：当小车接近右转弯时，黑色块的中心位于图像的右侧， $X > 55$ ，此时小车应右转一定角度，直到 X 重新回到中心区域 $45 \sim 55$ ；

左转弯：当小车接近左转弯时，黑色块的中心位于图像的左侧， $X < 45$ ，此时小车应左转一定角度，直到 X 重新回到中心区域 $45 \sim 55$ ；

交叉路口：当小车接近十字路口、T 字路口时，某一时刻 MU3 返回的黑色块 W 值会突然增加，这是因为检测到了相交的横向色块，此时程序就可判定遇到了交叉路口。

(5) 示意代码：通过上述的分析，我们可以得出以下的控制方案

- 当检测到黑色块时，控制小车行进，否则寻找道路
- 当 $W < 40$ 时，小车沿单条道路前进（该参数需要根据实际调整）
- 当 $W \geq 40$ 时，遇到路口，需要做出行进决策
- 当 $X < 45$ 时，控制小车左转
- 当 $X > 55$ 时，控制小车右转
- 当 $45 \leq X \leq 55$ 时，控制小车前进



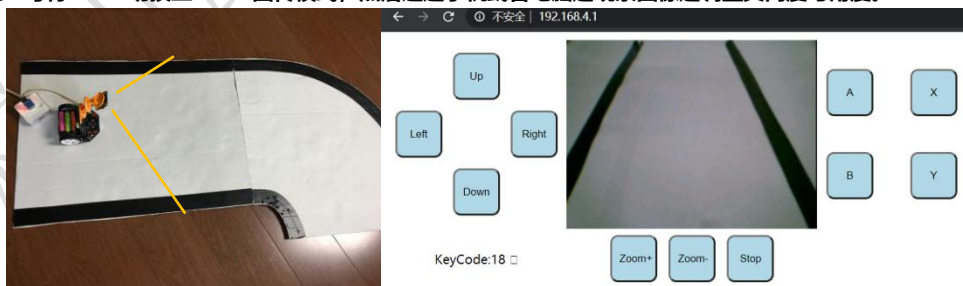
利用色块检测实现单线循线

5.8 视觉循线-模拟道路

利用 MU3 的色块识别功能, 小车也可以在下方左侧图中由两条黑线所包围的白色道路上流畅行驶。以下图为例进行讲解, 使用其他颜色搭配的地图原理相同。

(1) **MU3 的视角:** 与单线循线有所不同, 当小车位于道路中心时, MU3 需要能够看到包括两侧黑色边界在内的道路全景, 所以 MU3 的安装位置要高并且斜向下看, 。

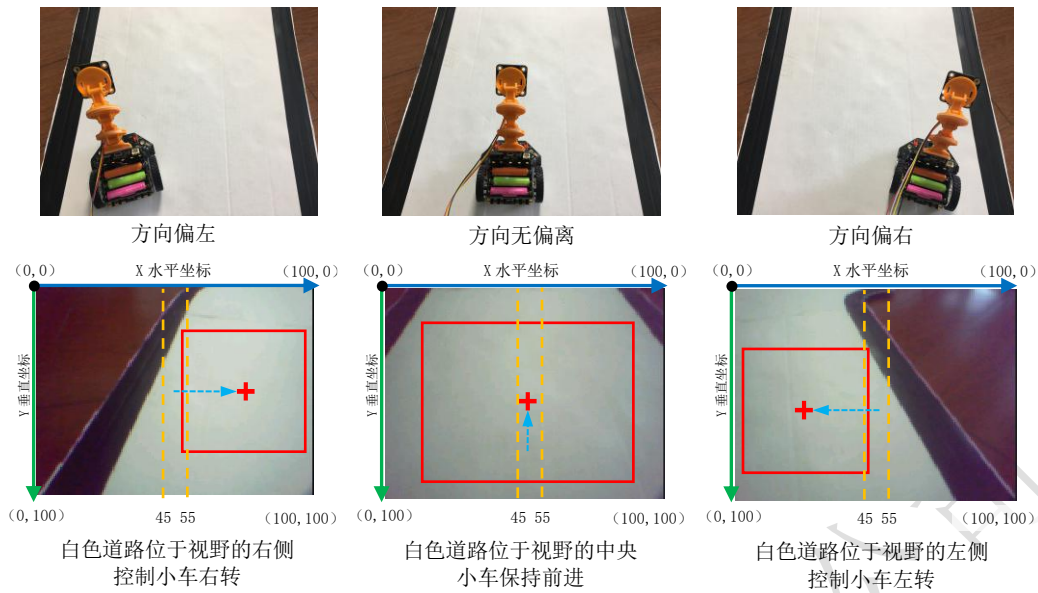
注: 可将 MU3 切换至 WIFI 图传模式, 然后通过手机或者电脑边观察图像边调整其高度与角度。



MU3 斜向下看

(2) **开启色块检测:** MU3 的参数配置与单线循线应用的相同: 校准并锁定白平衡; 为了避免板载 LED 的灯光产生干扰, 需要关闭 LED 显示; 为了提高检测帧率, 采用 I2C 通讯; 算法的性能以及摄像头的高帧率模式保持默认设置。

(3) **道路识别:** 与单线循线应用的原理类似, 区别在于此时需要检测白色色块。程序中设定 MU3 返回白色块的坐标和大小, 然后利用水平坐标 X 来判断小车是否偏离了道路。

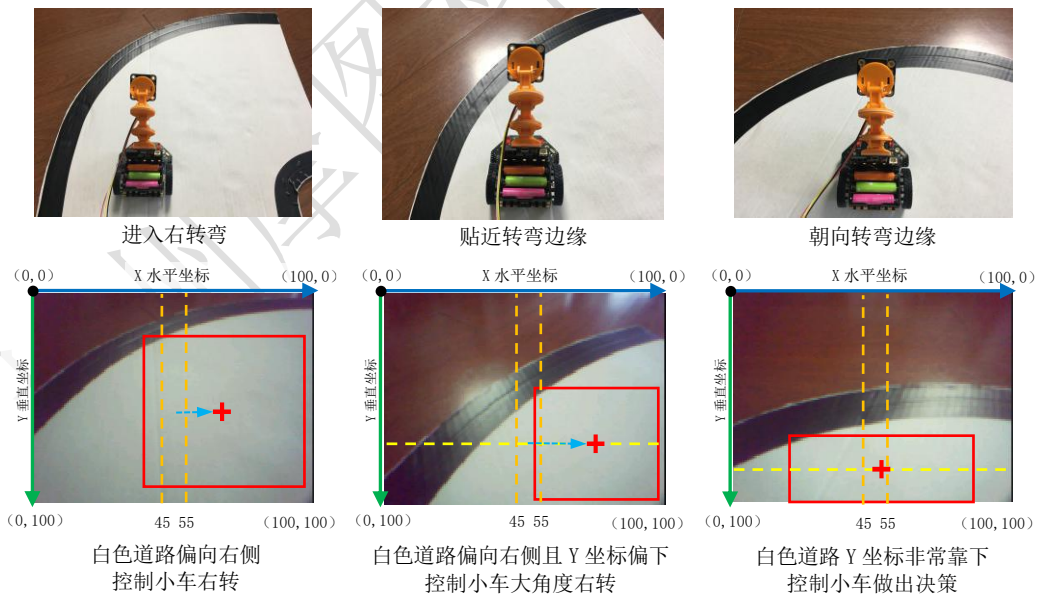


直行：设定中心区域的 X 值为 45 ~ 55，如果白色块的 X 值位于这一区间内，那么就认为道路位于正前方，此时小车可以继续直行；

偏左：行驶过程中，当小车偏向左侧导致返回的 $X > 55$ 时，需要控制小车右转一定角度，直到 X 重新回到中心区域 45 ~ 55；

偏右：当小车偏向右侧时，如果 $X < 45$ ，控制小车左转一定角度，直到 X 重新回到中心区域 45 ~ 55。

(4) 转弯道路识别：依据白色块的水平坐标 X 来判断转弯方向，通过垂直坐标 Y 判断转弯角度的大小。



右转弯：当小车进入右转弯时，图像中白色块的中心会移动到右侧，当 $X > 55$ 时，控制小车右转，直到 MU3 返回的 X 值重新回到中心区域 45 ~ 55；

左转弯：与右转弯处理类似，白色块的中心会移动到左侧，当 $X < 45$ 时，控制小车左转，直到 X 值重新回到中心区域 45 ~ 55；

贴近转弯边缘：当小车贴近右转弯边缘时，白色块的中心会移动到图像的右下方，此时除

了水平坐标 $X > 55$ 外，其垂直坐标 Y 不断增大，表示越来越接近道路边缘了，需要加大转弯角度，迅速做出调整；

朝向转弯边缘: 当小车朝向转弯边缘时, 虽然水平坐标 X 位于 45~55 的范围内, 但是垂直坐标 Y 的值很大, 这是因为边缘处的白色区域都移动到了图像的下方。此时需要做紧急决策。

(5) 示意代码：通过上述的分析，我们可以得出以下的控制方案

当检测到白色块时，控制小车行进，否则寻找道路

当 $X < 45$, $Y < 60$ 时, 控制小车左转

当 $X < 45$, $Y \geq 60$ 时, 控制小车加速左转

当 $X > 55$, $Y < 60$ 时, 控制小车右转

当 $X > 55$, $Y \geq 60$ 时, 控制小车加速右转

当 $45 \leq X \leq 55$ 时, 且 $Y < 60$ 时, 控制小车前进

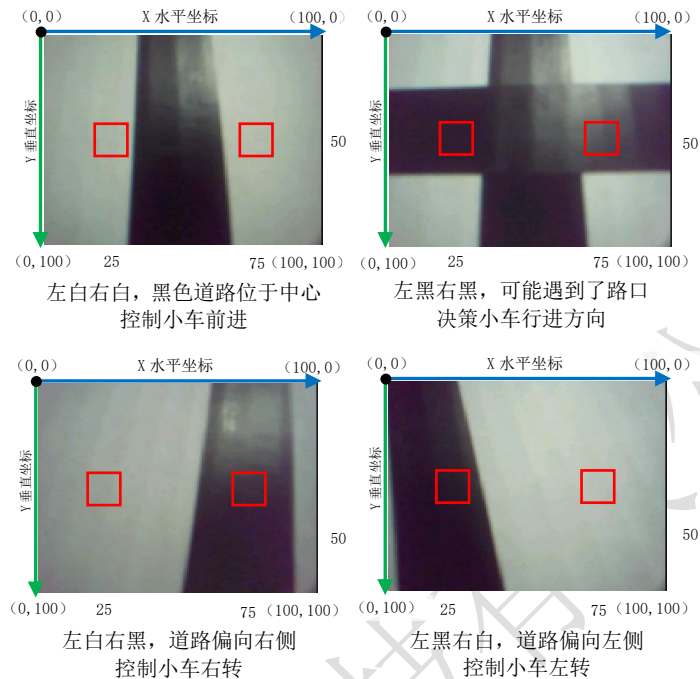
当 $45 \leq X \leq 55$ 时, 且 $Y \geq 60$ 时, 小车朝向转弯边缘, 需要做紧急决策



利用色块检测实现模拟道路循线

5.9 颜色识别算法实现循线

可以利用颜色识别算法进行循线，其原理与利用红外传感器循线一样，识别图像中多个点的颜色来判断是否偏离道路。下面以最简单的双点识别为例来说明其原理，实际中还可以使用更多点位来完成更复杂的道路判断。

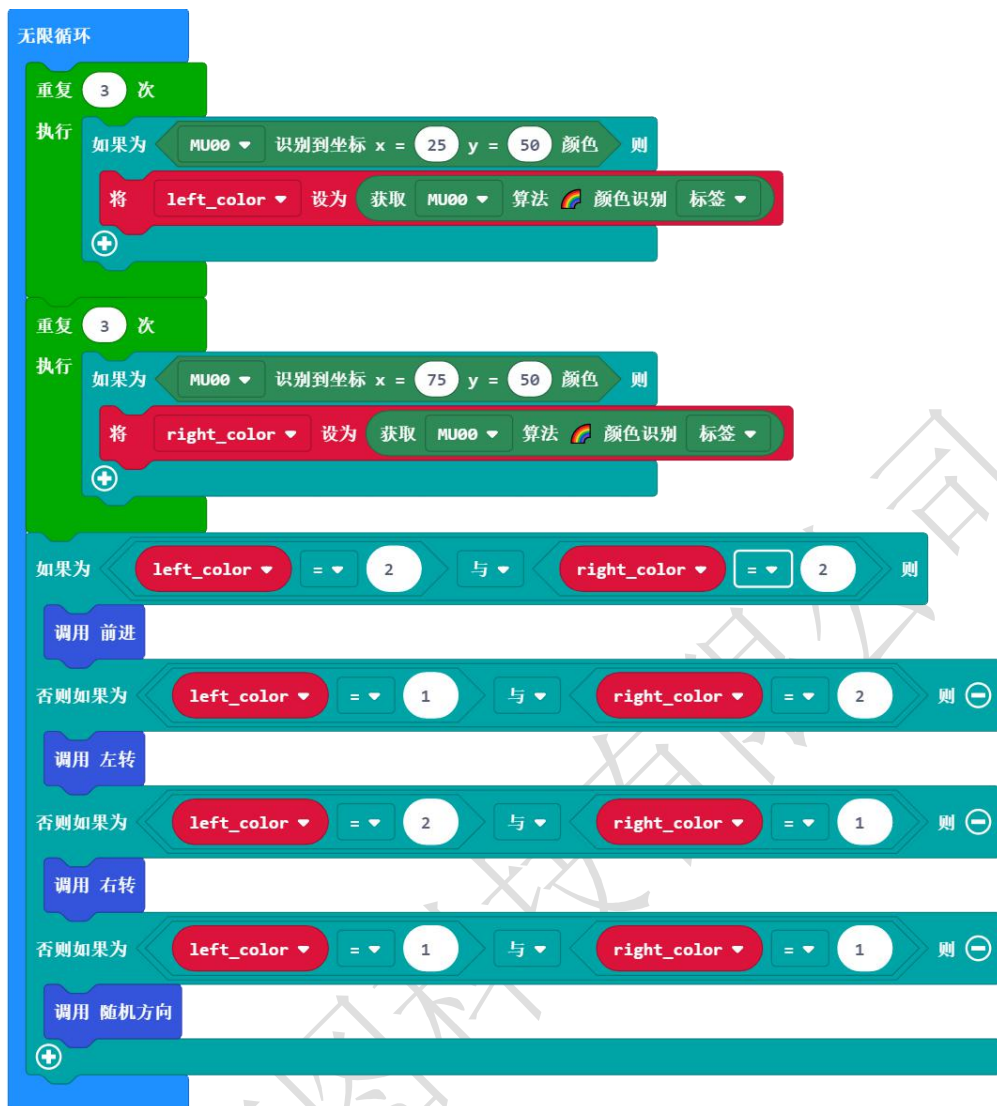


示意代码：

- (1) 根据道路的粗细程度，在图像的左右两侧对称地各选择一个识别点；
- (2) 开启颜色识别算法，并将识别区域的坐标设置到左侧点位，识别颜色并记录；
- (3) 将识别区域设置到右侧点位，识别颜色并记录，完成一次采样过程；
- (4) 如果为左白右白：前进；左黑右黑：路口；左白右黑：右转；左黑右白：左转。



初始化，启动颜色识别，校准并锁定白平衡



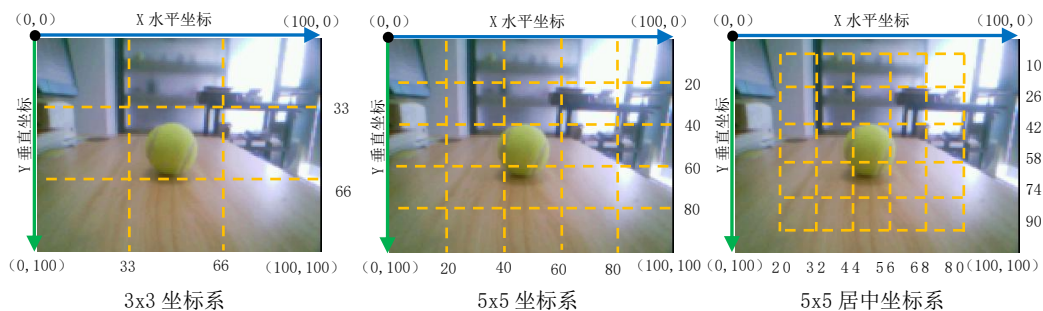
轮流识别两点颜色，执行对应的动作

5.10 二维空间重分割

有些应用不需要用到类似 (12, 41) 的精确坐标，而只要知道大概的趋势（譬如位于图像左侧）即可，此时可以进行二维空间的重分割。

5.10.1 分割的方法

二维空间重分割就是将原本 100x100 的坐标系简化到一个 3x3、5x5 或是其它的低分辨率坐标系中，在计算过程中，数值需要取整数：





3x3 坐标系：将 X-Y 的值分别除以 33 取整即可，坐标为 100 时可以忽略



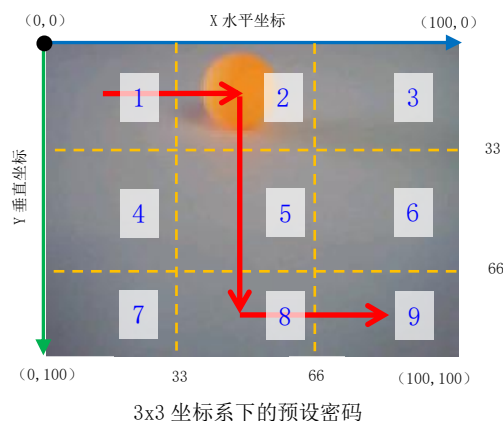
5x5 坐标系：将 X-Y 的值分别除以 20 取整即可



5x5 居中坐标系：将 X 值先减去 20 再除以 12 取整，将 Y 值先减去 10 再除以 16 取整

5.10.2 轨迹识别

轨迹跟踪是二维空间重分割的典型应用。现在以 3x3 分割为例，讲解如何利用视觉算法跟踪轨迹解开虚拟密码锁。类似于智能手机的图案锁，预设密码如下图中红色箭头所示，MU3 需要跟踪乒乓球的移动轨迹，如果与预设密码一致则解锁成功，否则失败。



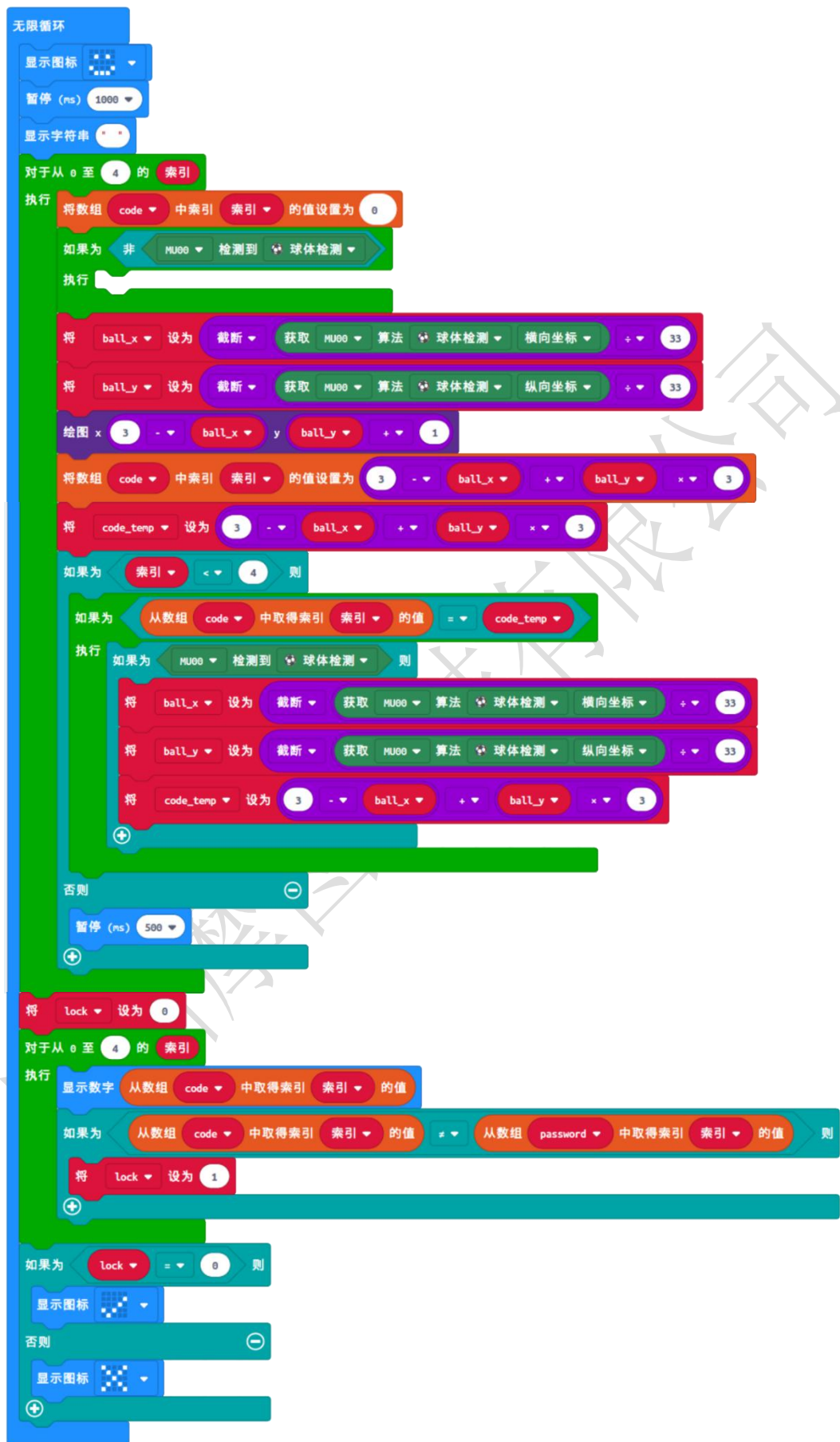
3x3 坐标系下的预设密码

示意代码：

- (1)按 3x3 将图像分割为图中的 9 个区域, 区域 1~9 的坐标分别为 1 (X0,Y0) , 2 (X1,Y0) , 3 (X2,Y0) , 4 (X0,Y1) , 5 (X1,Y1) , 6 (X2,Y1) , 7 (X0,Y2) , 8 (X1,Y2) , 9 (X2,Y2);
- (2) 当首次检测到乒乓球时, 计算并记录其当前所在区域编号, 此编号为起始编号;
- (3) 如果乒乓球始终位于同一个区域, 则不重复记录编号, 直到其所在区域发生变动;
- (4) 所在区域变动后, 则记录第二个区域编号, 依此类推, 直到检测到 5 个编号为止;
- (5) 将这 5 个编号与预设密码 (1, 2, 5, 8, 9) 进行匹配, 如果顺序和编号完全一致则解锁成功, 否则失败。



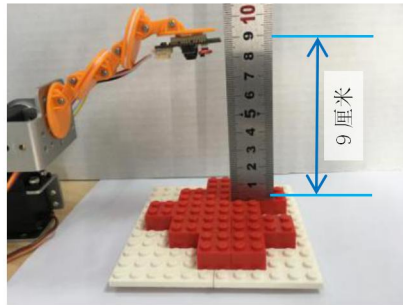
设置正确密码为“12589”



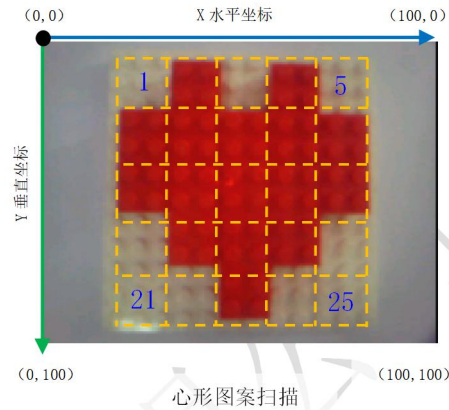
识别轨迹，显示数值，判断是否与密码一致

5.10.3 图案扫描

图案扫描可以利用二维空间重分割来实现，在此以 5x5 居中分割为例，讲解如何扫描心形图案。首先利用积木块拼出一个白底的心形图案，如下方左侧图片所示，白底的作用是参照物；然后将 MU3 固定于图案的上方，测量 MU3 的电路板距离乐高积木块上表面约 9 厘米（可以根据实际检测效果去调整），该图案在 MU3 中的成像如下方右侧图片所示。



MU3 与乐高图案的相对位置



- (1) 开启颜色识别算法，校准并锁定白平衡；
- (2) 从左上角的第一个格子区域 1 开始，设置颜色识别区域的坐标至该区域中心；
- (3) 识别该区域的颜色，为了提高可靠性连续检测 3 次，然后记录下颜色的 Label；
- (4) 设置颜色识别区域的坐标至右侧的区域 2，重复步骤 3 记录结果，依此类推，直到 25 个区域全部统计完毕，最后会得到一个 5x5、关于 Label 值的数组；
- (5) 对于该心形图案，有效数据是 3（红色的 Label=3）。可利用 Micro:Bit 的 LED 矩阵将该图案显示出来，数组中 Label=3 的位置对应的 LED 亮红色，其余位置的 LED 保持熄灭。

注：该方法同样适用于扫描手绘图案，也可以用 10x10 的矩阵去扫描。

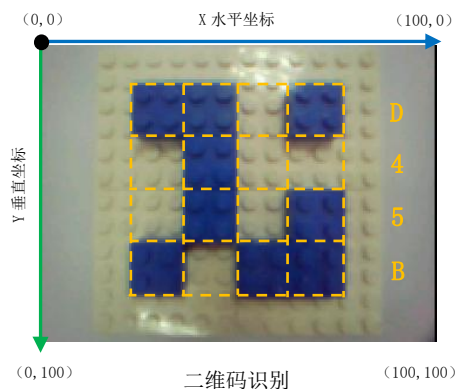


在 5x5 矩阵内检测红色色块，并在 LED 屏幕对应位置上显示

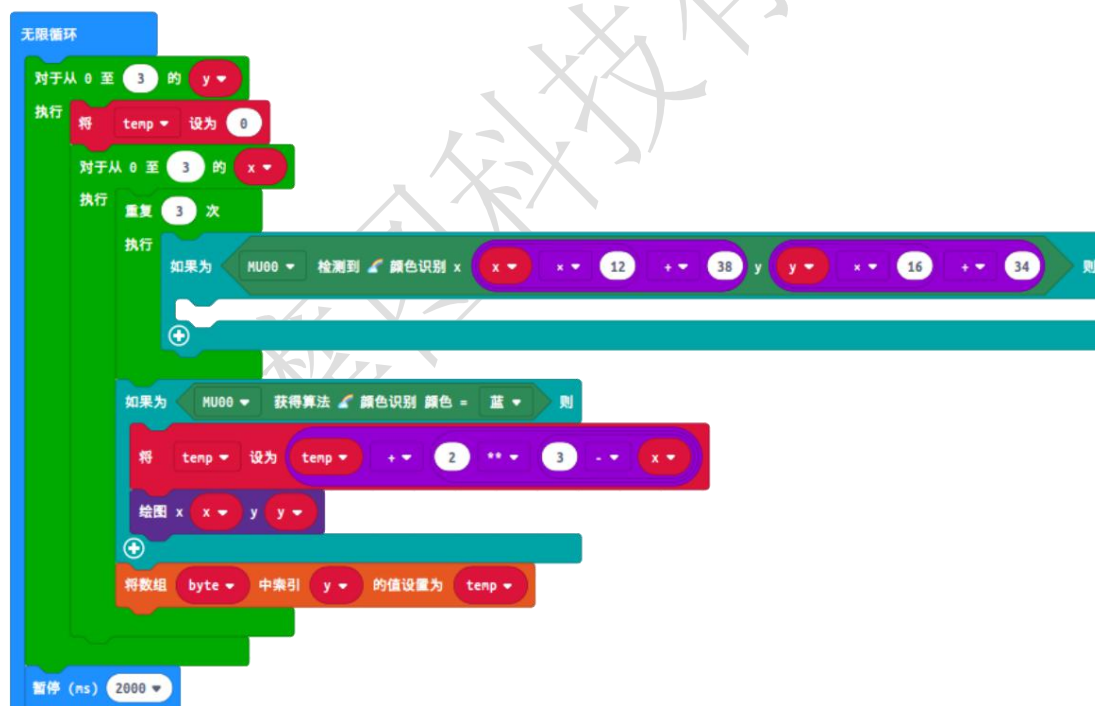
5.10.4 图案识别

图案识别可以利用二维空间重分割来实现，在此以 4x4 居中分割为例，讲解如何识别自定

义的二维码。MU3 的固定方式与上一节一致。

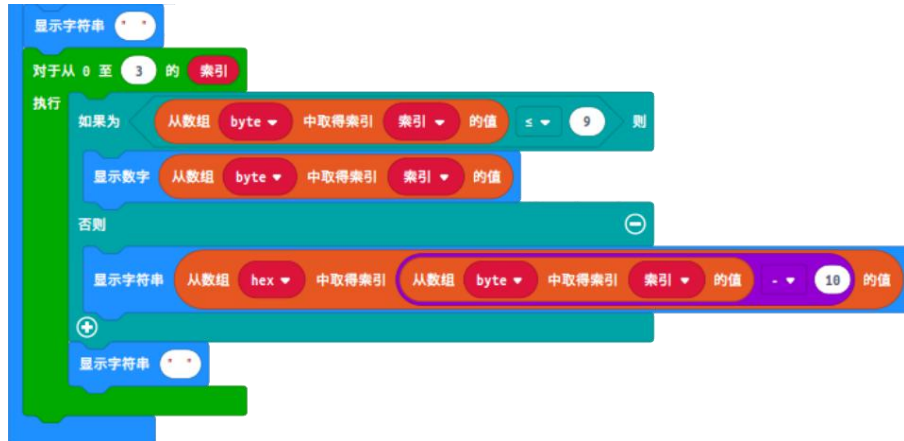


- (1) 开启颜色识别算法，校准并锁定白平衡；
- (2) 从左上角的第一个格子区域 1 开始，设置颜色识别区域的坐标至该区域中心；
- (3) 识别该区域的颜色，如果为蓝色则标记为 1，其他颜色则标记为 0；
- (4) 设置颜色识别区域的坐标至右侧的区域 2，重复步骤 3 记录结果，依此类识别完第一行 4 个区域会得到值为“1101”的数据，即 16 进制的“D”；
- (6) 解码其余的 3 行数据，于是得到该二维码的数据“D45B”；
- (7) 如果将这串字符定义为“开门”，据此就可以设计一个门禁卡片系统。



(下页继续)

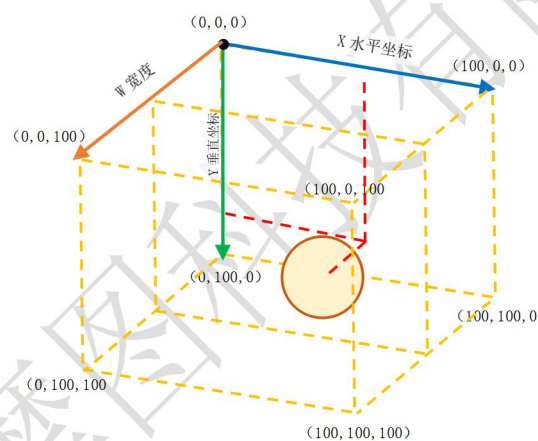
(接上页)



在 4x4 矩阵内检测蓝色色块，每行结果转换为十六进制并显示出来

5.11 三维空间坐标系

结合 X-Y 坐标和宽度 W 可构建一个三维空间坐标系。由于三维空间比二维空间多一个维度的信息，就可以实现更多奇妙的应用，譬如三维空间密码锁、三维空间音乐盒、三维空间绘画等。



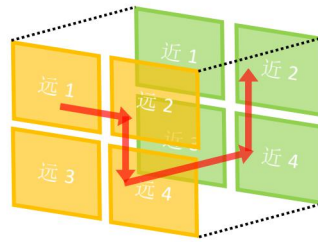
三维空间坐标系

类似于二维空间区域分割的方法，我们可将三维空间分割出不同的区域块，通用的代码如下：



三维空间坐标系的通用示意代码

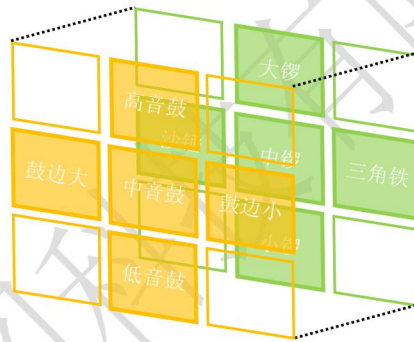
5.11.1 三维空间密码锁



在三维空间中的轨迹密码

- (1) 将 X-Y 坐标系进行 2x2 区域分割，将 W 宽度以 50 为分界点分为远（51~100），近（0~50）两个区间，于是三维空间中存在 8 个象限区间：远 1，远 2，远 3，远 4，近 1，近 2，近 3，近 4；
- (2) 使用球或者卡片，跟踪并记录其在三维空间中移动过 5 个位置；
- (3) 如果移动的轨迹与图中一致（远 1>远 2>远 4>近 4>近 2）则表明密码正确，可以开锁。

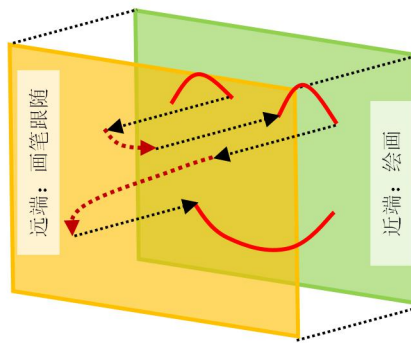
5.11.2 三维空间音乐盒



三维空间乐器的定义

- (1) 将 X-Y 坐标系按照 3x3 再分割，将 W 宽度以 50 为分界点分为远（51~100），近（0~50）两个区间，于是三维空间中存在 18 个区间；
- (2) 远端和近端只使用“上下左右中”五个区间，假定每个区间存储了一种乐器演奏的音乐；
- (3) 使用球或者卡片，当目标出现在十个有效区间的任意一个内，则播放对应的音乐；
- (4) 如果同一区间内目标被持续检测到，则只播放一次音乐；如果其消失后又连续 3 帧被检测到或者位置发生了变化，则播放对应的音乐；
- (5) 还可以利用目标物体 Label 值的不同来传递更多信息，比如利用数字卡片的 Label 值来控制音量的大小或音域的高低。

5.11.3 三维空间绘画



在三维空间中绘画

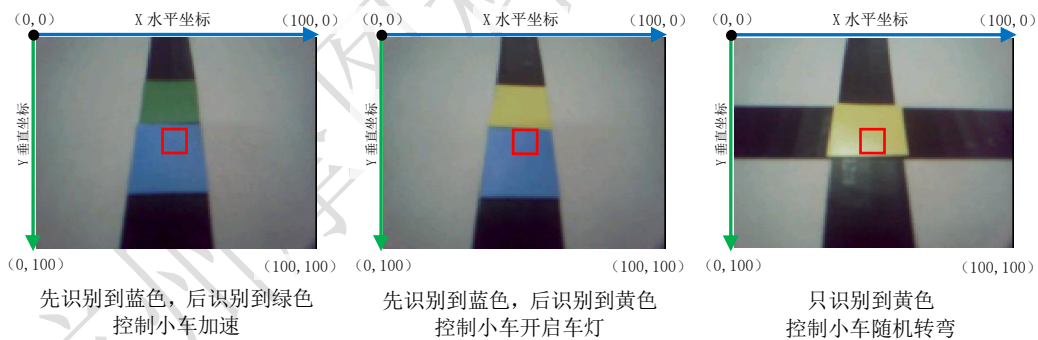
- (1) 将 W 宽度以 50 为分界点分为远 (51 ~ 100) , 近 (0 ~ 50) 两个区间, 于是三维空间中存在远近两个平面;
- (2) 用球作为画笔, 在远处平面上只跟随画笔, 但并不在屏幕上绘画;
- (3) 当球移至近处平面时, 表明画笔被按下, 在屏幕上同步显示坐标轨迹。

5.12 同时开启多种算法

MU3 支持多个算法同时运行, 可以满足较复杂应用的需求。但是与运行单个算法相比, 处理时间会变长, 因此需要灵活地处理算法的开启和关闭。

5.12.1 循线加上颜色识别

如下面三幅图所示, 利用色块检测算法巡线的同时开启颜色识别算法, 当小车前进过程中识别到不同的颜色或颜色组合, 就可以执行对应的命令, 例如:



示意代码：



同时识别两种卡片

5.12.3 利用颜色识别物体

对于具有单一颜色的物体，可以先用颜色识别，后用色块检测的方法来实现对这个物体的识别或跟踪，比如黄色的香蕉，红色的袋子，蓝色的瓶子等等，当然，背景中不能有类似颜色的干扰。

示意代码：



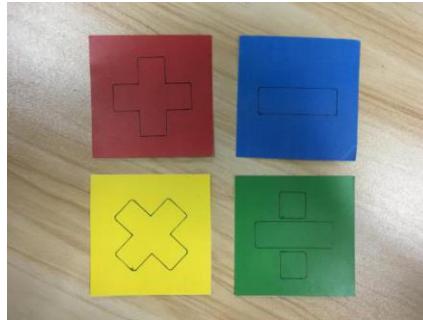
当按下 A 键，则识别颜色，然后在主循环中开始检测相应的颜色物体

5.13 算法的衍生

如果 MU3 现有的算法无法直接满足项目的需求，那么就需要思考能否利用现有算法来创造“新算法”，间接地解决问题。

5.13.1 利用颜色识别来实现加减乘除

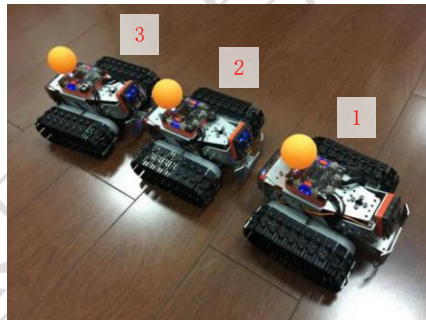
MU3 目前尚无法识别“加减乘除”的运算符号，但是可以利用 4 种颜色来替代。使用时，首先在四张不同颜色卡片上绘制出相应的运算符号图案，然后开启颜色识别算法来判断是哪一个运算符号。下图中，红色代表“加号”，蓝色代表“减号”，黄色代表“乘号”，绿色代表“除号”。类似的，也可以用来表达“石头、剪刀、布”。



利用颜色卡片制造的“加减乘除”运算卡片

5.13.2 利用乒乓球实现任意物体跟踪

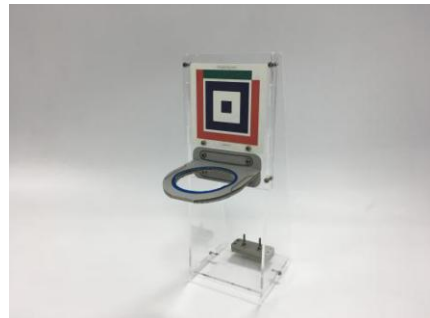
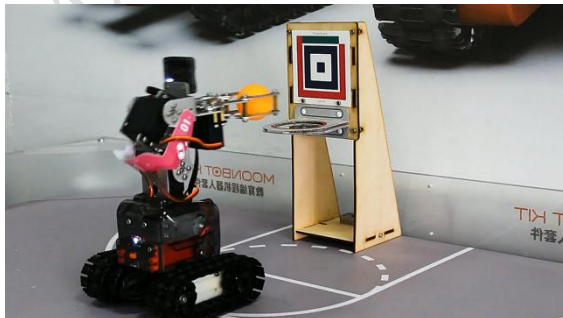
想要使用 MU3 跟踪某一移动的物体，可在该物体上固定一个乒乓球，然后识别和跟踪该乒乓球即可。下图中，1 号车在最前面由遥控或者程序控制行驶，后面的 2 号车通过识别 1 号车上的乒乓球实现跟随，同理 3 号车实现跟随 2 号车，这样三台车就组成了一个自动驾驶的车队。



利用乒乓球作为标识物，实现对任意物体的跟踪

5.13.3 利用形状卡片指示一个篮球框

想要将球准确投进篮筐，但是没有篮筐的识别算法，如何处理？如下图所示，可以利用一张形状卡片来指示篮筐的方位，通过卡片的大小来推断与篮筐之间距离。



利用形状卡片指示篮球框

5.14 数据的特殊处理

MU3 输出的检测结果（位置、宽高等信息）并不是稳定不变的，即使 MU3 和目标物体的相对位置没有发生变化，小幅度的数据抖动会一直存在，并且这是不可避免的。数据在被输出前会经过一定处理，但如果仍然无法满足应用的需求，可以尝试采用下面的一些方法进行二次处理来提高数据的稳定性与可靠性。

5.14.1 提升检测结果的可靠性

为了消除偶尔的误报或是漏报，可以采取对检测结果进行数据统计，当统计结果达到某个阈值时才更新相应的结果。以人体检测为例：

(1) 统计人体的被检测到的次数和未被检测到次数，分别用变量 `detected_count` 和 `undetected_count` 表示；

(2) 如果 `detected_count > 5`，则表明确实检测到了人，此时将检测标志变量 `detect_flag` 置 1 并清空 `undetected_count` 数据；

(3) 反之如果 `undetected_count > 5`，则表明人确实离开了，此时将检测标志变量 `detect_flag` 置 0 并清空 `detected_count` 数据；

(4) 通过 `detect_flag` 来判断串口输出内容。



利用数据统计来提高检测结果可靠性



串口输出相应的结果

5.14.2 消抖措施

数据抖动是无法避免的, 比如水平坐标 X 的期望值为 50, 而实际检测结果可能为: 49, 50, 48, 51, 50, 50, 49, 51.....假设程序简单地以 X 是否等于 50 作为舵机转动的判断条件, 那么舵机会一直转动, 导致寿命急剧降低。针对这种情况应该定义一个阈值, 只有当前后两次结果的差值大于该阈值时, 舵机才会动作, 如果小于这个阈值维持不动。



利用阈值来消除抖动

5.14.3 快速跟踪

进行目标跟踪时, 有时水平坐标 X 会距离中心位置 50 很近, 有时很远, 如果不区分目标的位置而采用相同的跟踪速度, 那么就可能会出现近距离时速度太快而过调, 而远距离时速度不足而丢失目标的问题, 此时应根据位置的不同分别处理。

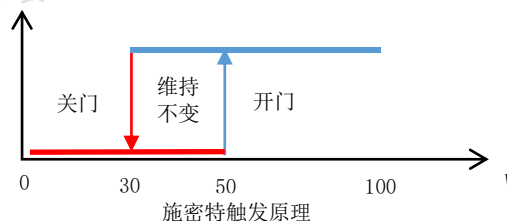
下例中, 当 $X < 30$ 时快速左转, 当 $30 \leq X < 45$ 时正常左转, 而当 $X > 70$ 时快速右转, 当 $70 \leq X < 85$ 时正常右转, 位于 $45 \leq X \leq 70$ 之间则保持不变。



根据位置的不同执行对应的程序

5.14.4 施密特触发

施密特触发具有两个独立的触发条件，一个是开启条件，一个是关闭条件，在这两个参数之间是一个缓冲区间，系统的状态会维持不变。如下图所示将 $W > 50$ 作为开门条件，将 $W < 30$ 设为关门条件，那么 30~50 之间就是缓冲区。如此定义，即使 W 在 50 或是 30 附近有数据抖动，门也不会的频繁开启或关闭，因为抖动区间达不到改变门状态的触发条件。





利用施密特触发控制门的开启与关闭

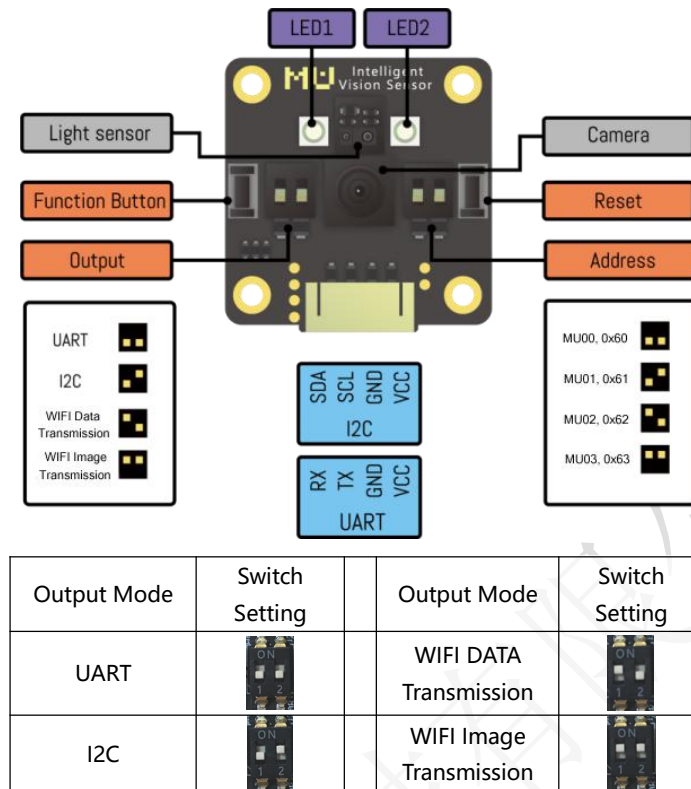
5.14.5 排除异常数据

对于平稳运动的物体而言，理论上与其相关的数据不会存在跳变。倘若新检测结果与上一次结果相比发生了较大的变动，则有可能是一个异常数据，采用下例中的方法可以简单有效的排除异常。如果前后两次检测到的人体水平坐标 X 值相差在 10 以内，则认为新结果是合理的并更新 old_x，反之则视为异常数据，不做处理。



简单的排除异常数据

6 WIFI 功能



注意：WIFI 功能的功耗较高，尤其是开启 WIFI 的瞬间，为了确保 MU3 能够正常工作，请确保电源能够稳定提供 0.3A 的电流。

6.1 WIFI 工作模式

当输出模式开关切换到“10”（WIFI 数传模式）或者“11”（WIFI 图传模式）时，WIFI 功能被激活，此时 AP 模式（热点模式）与 STA 模式（终端模式）同时生效。

AP 模式

WIFI 功能激活后，MU3 会**自动**建立一个 SSID 为“MORPX-MU-AABBBB”格式的热点网络，同时两颗板载的 LED 会分别显示不同的颜色。

“AA”表示两颗板载 LED 的颜色：R 表示红色，G 表示绿色，B 表示蓝色，Y 表示黄色，W 表示白色，P 表示紫色。当存在 SSID 冲突时，可以通过变更右侧的地址开关来改变 SSID。

“BBBB”表示 4 位随机数字。

一旦有设备成功接入，MU3 就会进入 AP 工作模式。

由于输出功率的限制，MU3 热点覆盖的范围有限。当穿墙后，WIFI 信号会变得微弱，很有可能会出现无法接入热点的问题。

STA 模式

MU3 同样可以接入 WIFI 局域网。**一旦成功接入局域网，MU3 就会工作在 STA 模式下**，此时同一局域网内的其他设备就可以访问它。

不同于 AP 模式，需要设置一些参数 MU3 才能够接入到局域网。

示例程序：



"your_ssid" : 局域网的 SSID;

"your_password" : 局域网的密码;

"your_target_ip" : 同一局域网内要与 MU3 建立连接的设备的 IP 地址, 譬如笔记本电脑或者智能手机;

"3333" : MU3 默认接收 UDP 数据的端口, 请勿变更。

6.2 MU3 的 WIFI 传输模式

MU3 的 WIFI 功能激活后 (切换至 WIFI 数传模式或者 WIFI 图传模式), 无论 WIFI 工作在 AP 模式还是 STA 模式, 都可以通过 UDP 数据包与 MU3 或者其配套的控制板进行信息交换。

注: MU3 对接收到的非以 AT 开头的数据会通过 UART 口透传给控制板, 同时将来自控制板非以 AT 开头的数据通过 WIFI 进行转发。

WIFI 数据传输模式

在此工作模式下, MU3 能够正常运行所有的视觉算法, 并且可以将识别结果通过 WIFI 返回给 PC, 也可以通过 UART 发送给控制板。

WIFI 图像传输模式

在此工作模式下, 由于 ESP32 自身硬件资源的限制, MU3 **无法**运行视觉算法, 但是可以传

输实时的图像，并且可以透传数据。实时图像的查看方法如下：

AP 模式

手机或者电脑接入 MU3 的热点，成功接入后 MU3 板载的两颗 LED 会熄灭。

在浏览器（建议采用 chrome 或者 firefox）地址栏输入 192.168.4.1，回车后 MU3 内置的控制页面就会打开。

STA 模式

首先确保电脑或者手机与 MU3 都成功地接入了同一局域网。

在浏览器（建议采用 chrome 或者 firefox）地址栏输入 MU3 在该局域网内的 IP 地址，回车后 MU3 内置的控制页面就会打开。

- 注：1、同一时间只允许一台终端与 MU3 建立连接；
2、其他的浏览器不推荐；
3、在与 MU3 建立连接前请审查下电脑防火墙的设置。



MU3 内置的控制页面

- 注：1、红色框区域为实时图像播放窗口；
2、可以利用此模式查看 MU3 的安装角度是否合适或者检查视野内是否存在干扰因素。调整结束后，可以切换到其他的数据输出模式，配置运行相关的算法。如此操作可以极大地缩短调试时间。

虚拟按键-字符对应表

虚拟按键	字符/ASCALL	虚拟按键	字符/ASCALL	虚拟按键	字符/ASCALL
Up	U / 85	A	A / 65	Zoom+	
Down	D / 68	B	B / 66	Zoom-	
Left	L / 76	X	X / 88	Stop	S / 83
Right	R / 82	Y	Y / 89		

6.3 AT 指令

MU3 拥有自己的 AT 指令集。指令的标准格式如下：

AT+COMMAND=<PARAM1>,<PARAM2>END

“END” 代表如下的字符：“回车” “换行” 或者 “空格”。

可以通过串口或者 UDP 数据包发送 “AT+HELP” 指令获取更详细的信息。

注：I2C 通讯模式下无法使用 AT 指令。

6.4 带 WIFI 遥控的监控小车

为了达到较好的效果，MU3 工作在 STA 模式，即上位机与 MU3 接入同一局域网，这样可以在一个较大的范围内进行遥控。

在示例程序中，通过点击 “Up” “Down” “Left” “Right” “Stop” 五个虚拟按键来控制小车的运动。此外，通过敲击键盘或者其他的虚拟按键可以发送更多的信息。

示例程序

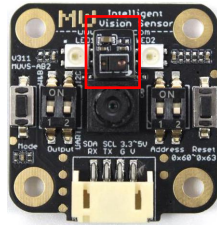


接入局域网，初始化遥控命令



接收遥控命令，执行相应动作

7 接近检测、环境光亮度检测与运动手势检测



功能高度集成的红外传感器

MU3 板载一颗功能高度集成的红外传感器，能够实现接近检测、环境光亮度检测以及运动手势检测。这些是 MU3 所额外具备的。

使用时需注意，运动手势检测功能只可单独开启，无法与其他两个功能并用。

该传感器的功耗是可观的，如果采用 3.3V 供电，开启运动手势检测时需要将其灵敏度调到最高，建议采用 5V 供电。

有六个程序积木块与该传感器相关。



运动手势检测



编程时，首先利用第一个程序块来判断 MU3 是否有检测到手势，如果有再判断是哪一种手势。

接近检测



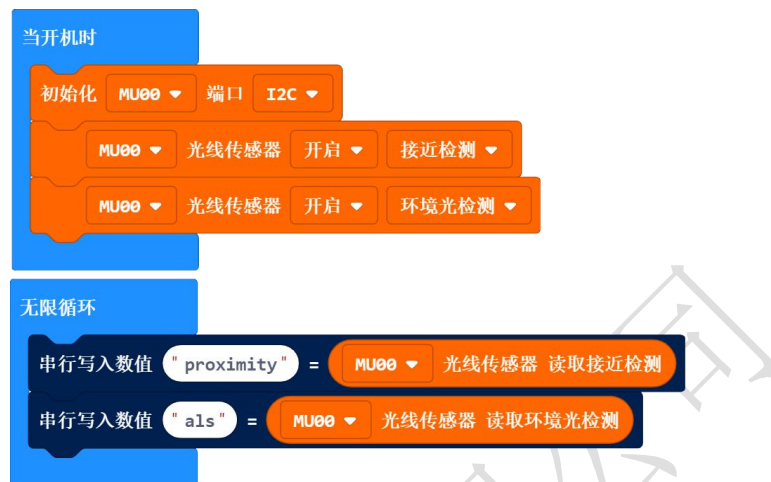
这个程序块会返回一个 0-255 之间的值。距离越近，返回的值越大。

环境光亮度检测



这个程序块会返回一个 0-65535 之间的值。光线越亮，返回值越大。

示例程序



技术支持

感谢您购买使用MU 视觉传感器3，我司会持续更新本产品的固件以及配套的库函数与例程，您可以从如下网址获取最新的技术资料：

GitHub: <https://github.com/mu-opensource/>

官网: <http://mai.morpx.com/page.php?a=sensor-support>

您在使用过程中有遇到技术方面的问题，可以通过电话 0571-81958588 或者email: support@morpx.com 与我司取得联系。

官方技术支持微信号



官方技术支持 QQ 群号



修订历史

日期	版本	发布说明
2019.08.10	V0.1	起草
2020.06.15	V1.0	版本升级; 本手册名称变更为《高级用户手册》，与英文版统一; 增加了两章，分别关于多合一红外传感器以及WIFI 功能的使用说明; 伴随固件升级，部分内容做同步更新。