

项目1：色彩简谱

音乐和颜色有关系吗？

许多音乐家把音乐与颜色相比拟，美国著名音乐家马利翁曾说过：“声音是听得见的色彩，色彩是看得见的音乐”。在音乐作品中运用不同的音色与在美术作品中运用不同颜色极为相似，音色与颜色同样能给人以明朗、鲜明、温暖、暗淡等感觉。



虽然目前还没有科学理论能证明音乐与色彩的关系，但是我们可以做出好玩的音乐与颜色互动的装置，最简单的是把音阶中七个音与七种颜色联系起来，让每种颜色代表一个音阶，做一个彩色的乐谱，通过色彩发出美妙的音乐！

功能介绍：



本项目利用HUSKYLENS的颜色识别功能，识别不同颜色的色块，通过麦昆plus小车播放不同的音阶，让你的乐谱既好看又好听，拥有绝对美妙的视听效果。

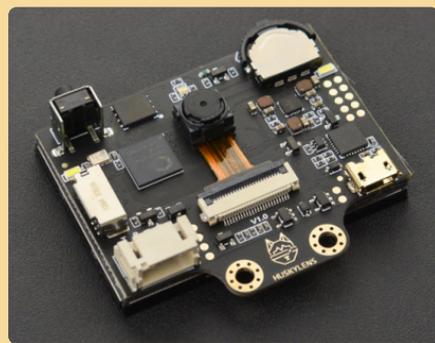
材料清单：



Micro:bit ×1



麦昆plus ×1



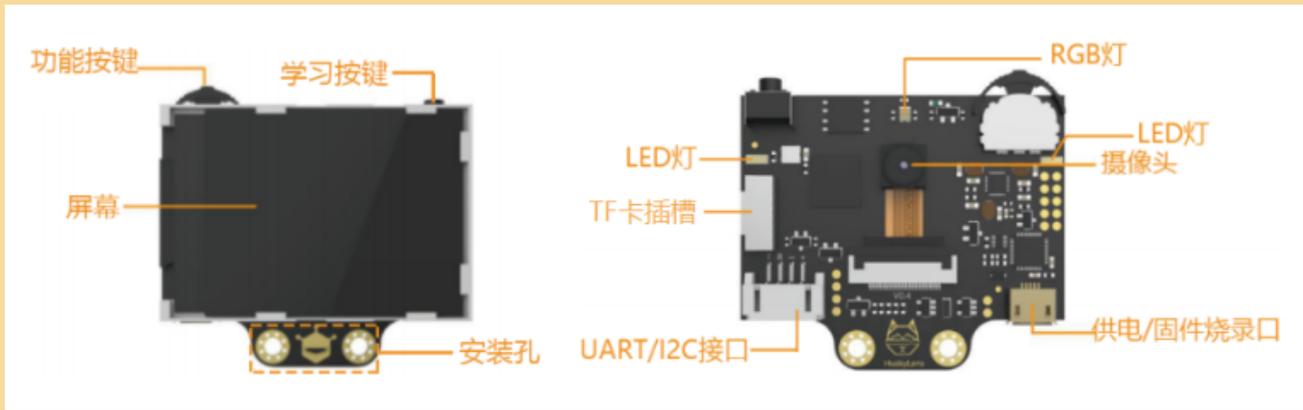
HUSKYLENS ×1

知识园地：



当今社会，自动化生产已经成为了社会的发展趋势，机器视觉作为“机器人”的眼睛，则显得尤为重要。

颜色识别作为其中一个重要的技术方向，已经经历了多代技术的升级。我们这个项目就是借助HUSKYLENS传感器的**颜色识别**功能。

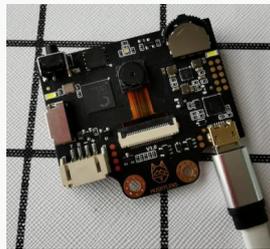


第一次使用摄像头的用户请参照WIKI网址进行固件烧录和语言设置:

http://wiki2.dfrobot.com.cn/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336

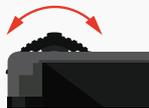
STEP1 上电

HUSKYLENS自带独立USB供电口，连接USB线，即可开机。

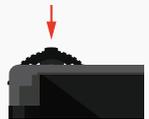


STEP2 操作设置-学习多个

① 向左或向右拨动“功能按键”，直至屏幕顶部显示“颜色识别”。



② 长按“功能按键”，进入颜色识别功能的二级菜单参数设置界面。



③ 向左或向右拨动“功能按键”，选中“学习多个”，然后短按“功能按键”，接着向右拨动“功能按键”打开“学习多个”的开关，即：进度条颜色变蓝，进度条上的方块位于进度条的右边。再短按“功能按键”，确认该参数。



④ 向左拨动“功能按键”，选中“保存并返回”，短按“功能按键”，屏幕提示“是否保存参数？”，默认选择“确认”，此时短按“功能按键”，即可保存参数，并自动返回到颜色识别模式。这样我们就设置好学习多个的功能了。

STEP3 操作设置-学习与识别

① 侦测颜色

将HUSKYLENS屏幕中央的“+”字对准目标颜色块，屏幕上会有一个白色方框，自动框选目标颜色块。调整HUSKYLENS与颜色块的角度和距离，让白色方框尽量框住整个目标色块。



② 学习颜色

侦测到颜色后，按下“学习按键”学习第一种颜色，然后松开“学习按键”结束学习，屏幕上有消息提示：“再按一次继续，按其他按键结束”。

如要继续学习下一种颜色，则在倒计时结束前按下“学习按键”，可以继续学习下一种颜色。如果不再需要学习其他颜色了，则在倒计时结束前按下“功能按键”即可，或者不操作任何按键，等待倒计时结束。

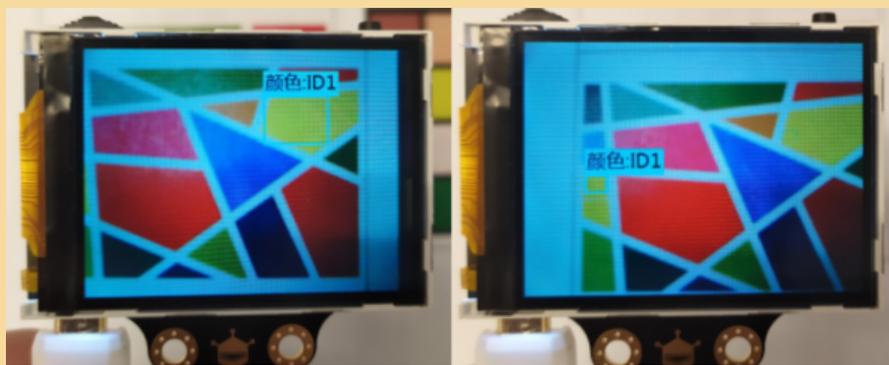


HUSKYLENS显示的颜色ID与学习颜色的先后顺序是一致的，也就是：ID会按顺序依次标注为“ID1”，“ID2”，“ID3”，以此类推，并且不同颜色对应的边框颜色也不同。

③ 识别颜色

如HUSKYLENS遇到相同或近似的颜色，屏幕上会有彩色边框框出色块，并显示该颜色的ID，边框的大小随颜色块的面积一起变化，边框会自动跟踪色块。多种不同的颜色可以同时识别并追踪，不同颜色对应的边框颜色也不同。

当画面中出现多个相同颜色的色块时，只能一次识别一个色块。



*环境光线对颜色识别的影响很大，对于相近的颜色，HUSKYLENS有时会误识别。建议保持环境光线的稳定，在光线适中的环境中使用此功能。

项目实践：▶

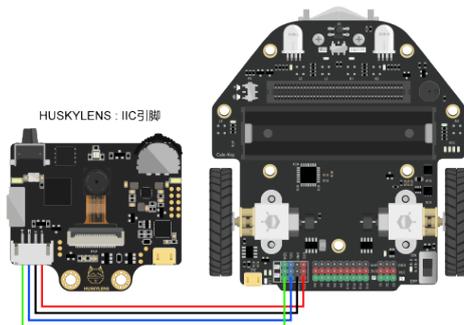
HUSKYLENS的颜色识别是如何使用的呢？如何将颜色与音阶一一对应呢？让我们来分解整个项目为几个小任务，一步一步来完成色彩简谱吧！

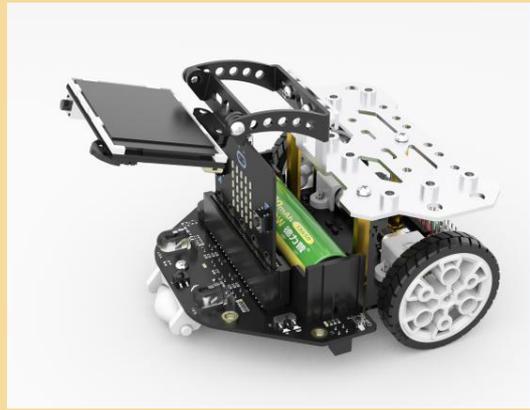
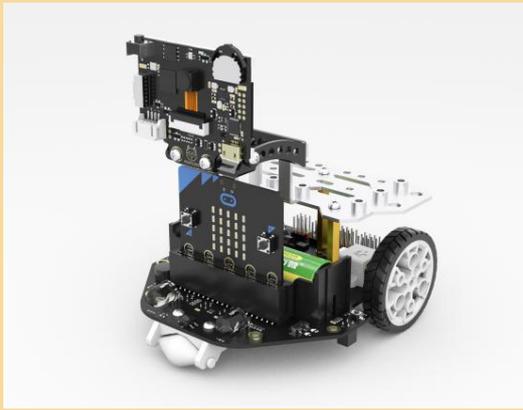
这个项目将分为三步将任务完成，首先我们会学习使用HUSKYLENS的颜色识别功能，并将识别到的颜色ID输出；然后我们就可以根据输出的颜色ID给他们对应的音阶播放；最后完善整个项目。

任务一：初识颜色识别

1、硬件设置

连线图：HUSKYLENS传感器使用的是IIC接口，需要注意线序，不要接错或接反。





组装图：HUSKYLENS传感器自带支架结构件，通过螺丝可以固定到小车上，并且HUSKYLENS可以调整多种角度。

2、程序设计

这里我们以让HUSKYLENS传感器学习3种颜色为例，通过串口输出颜色ID方便实时查看。

*串口：串口是计算机与硬件实时通讯的一种方式。比如这个任务中，通过串口，可以在电脑端实时查看小车上HUSKYLENS的数据。

STEP1 学习与识别

在设计程序之前我们需要让HUSKYLENS传感器学习各个颜色。
(注意：先开启“学习多个”的功能)



STEP2 Mind+软件设置

打开Mind+软件（162或以上版本）：



- ①切换到“上传模式”；
- ②点击“扩展”，在“主控板”下点击加载“micro:bit”；
- ③继续在“扩展板”下点击加载“麦昆Plus国内版”；
- ④最后在“传感器”下点击加载“HUSKYLENS AI摄像头”。

STEP3 指令学习

来认识一下主要用到的几条指令。

①初始化，仅需执行一次，放在主程序开始和循环执行之间，可选择I2C或串口，I2C地址不用变动。

注意：HUSKYLENS端需要在设置中调整“输出协议”与程序中一致，否则读不出数据。



STEP3 指令学习

②切换算法，可以随时切换到其他算法，同时只能存在一个算法。

注意：切换算法需要一些时间。



③主控板向HUSKYLENS请求一次数据存入“结果”（存在主控板的内存变量中，一次请求刷新一次存在内存中的数据），之后可以从“结果”中获取数据，此模块调用之后“结果”中才会获取到最新的数据。



④从请求得到的“结果”中获取当前界面中靠近中心的方框信息，未学习的框id为0，没有则返回-1。



⑤变量：变量指变化的量，方便用来存放变化的数字。
点击“新建数字类型变量”即可新建变量，可以直接设定变量的值，也可以增减变量值。



3、程序示例

程序图



4、运行效果

使用串口时需要保持计算机与主控板通过USB线进行连接，在Mind+中，连接对应COM口。

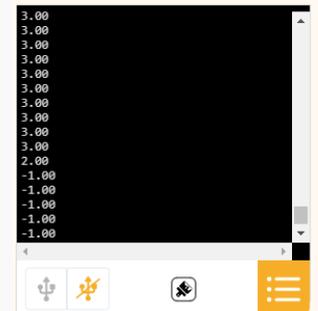
串口操作方法



1、串口波特率默认9600



2、打开串口



运行窗口

当在HUSKYLENS传感器已经学习过3种颜色后，识别到对应颜色的ID，就在串口显示对应的数字。1表示第一种学习的颜色，2表示第二种学习的颜色，3表示第三种学习的颜色，-1表示画面中没有学习过的颜色。

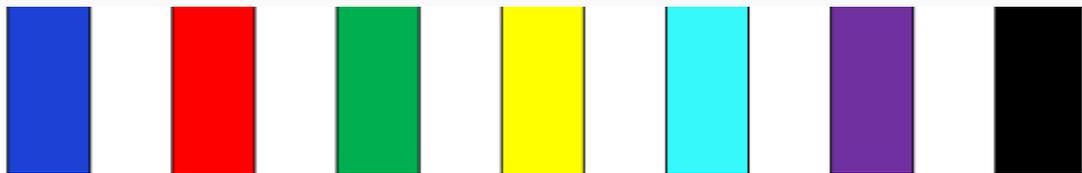
任务二：给每个颜色定义音阶

1、程序设计

STEP1 操作设置-学习与识别

在设计程序之前我们需要让HUSKYLENS传感器学习7种颜色。为了避免误识别，要选用7种色差大的颜色。

这里将ID1-ID7的颜色分别设置为蓝、红、绿、黄、青、紫、黑。请大家按照这个顺序学习颜色，才能与后面的程序对应。学习完7种颜色后，HUSKYLENS停止继续学习。



结合麦昆plus上的车灯，识别一种颜色，让车灯显示对应颜色。

怎么样？视觉效果是不是更加丰富了！

STEP2 指令学习

来认识一下主要用到的几条指令。

①从请求得到的“结果”中获取是否IDx已经进行了学习



②从请求得到的“结果”中获取当前界面中是否有方框或箭头，包含已学习 (id大于0) 和未学习的，有一个及以上则返回1。



STEP2 指令学习

③设置车灯显示不同的颜色，程序提供8种颜色。



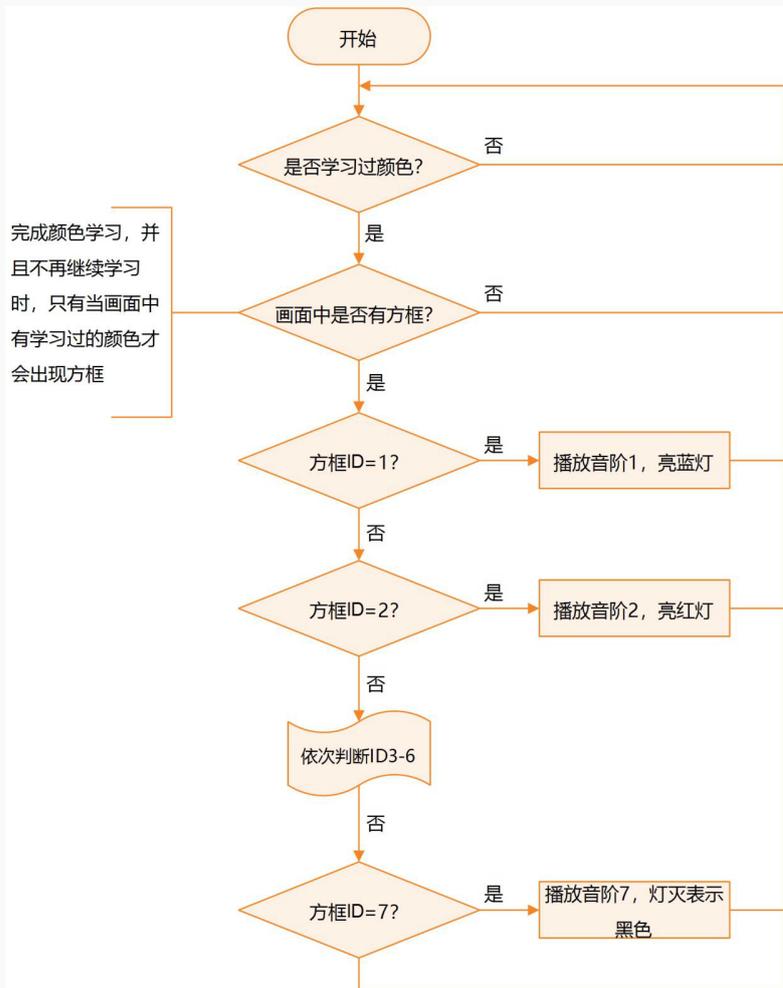
④播放音符指令，分为低中、高音，还有各种节拍。



⑤函数：函数可以理解为执行某种指定功能的程序。点击“自定义模块”新建函数。有时候主程序的指令过多会影响我们理解程序，通过函数可以简化主程序，方便理解整个程序。

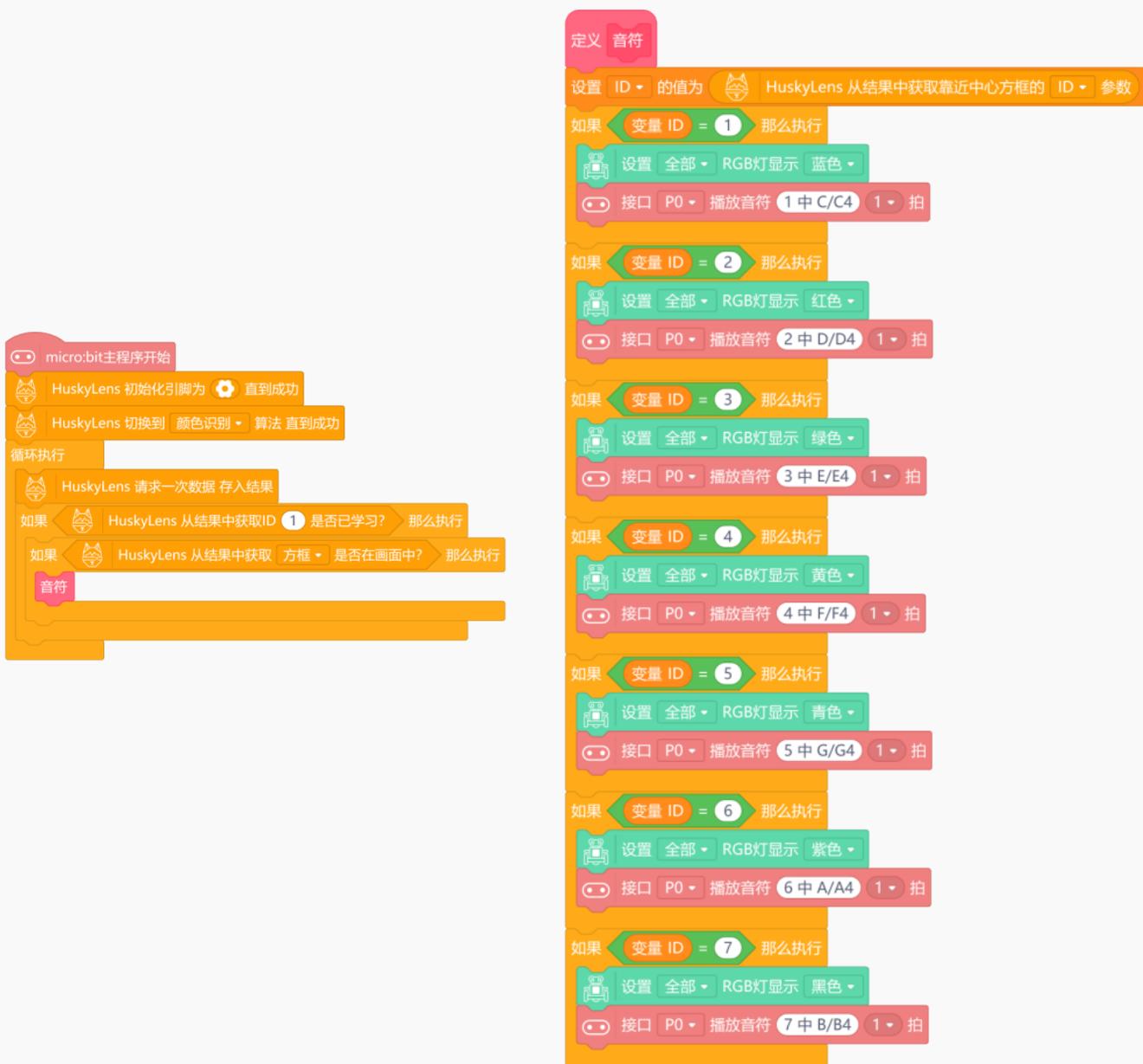


STEP3 流程图分析



2、程序示例

程序图



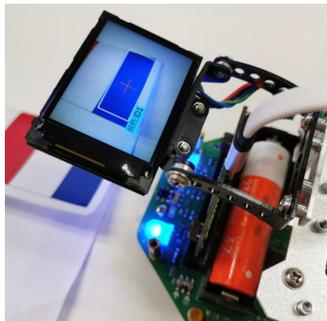
3、运行效果

如果你按照上文中给出的蓝、红、绿、黄、青、紫、黑顺序进行颜色学习，执行程序时，当HUSKYLENS显示屏画面中央为蓝色时，播放音符1，车灯为蓝色；当为红色，播放音符2，车灯成为红色；依次类推。

如果你的程序不能正常运行，请自行检查：

①是否按蓝、红、绿、黄、青、紫顺序学习了颜色；

②是否在学习完上述7种颜色后就停止了学习，如果没有停止，则摄像头屏幕上会一直输出白框，这个白框会影响程序执行。



任务三：去除小色块误判干扰

1、程序设计

STEP1 分析遇到的问题

任务二中的程序基本可以实现项目功能，但是摄像头画面中不小心有多个色块时，一些小色块可能会成为干扰。如何排除这种干扰呢？

一种解决思路是，通过色块在摄像头屏幕上的**面积占比**来判断是否是目标色块，设置一个阈值，在阈值之下的色块自动判断为干扰。

STEP2 指令学习

来认识一下主要用到的几条指令。

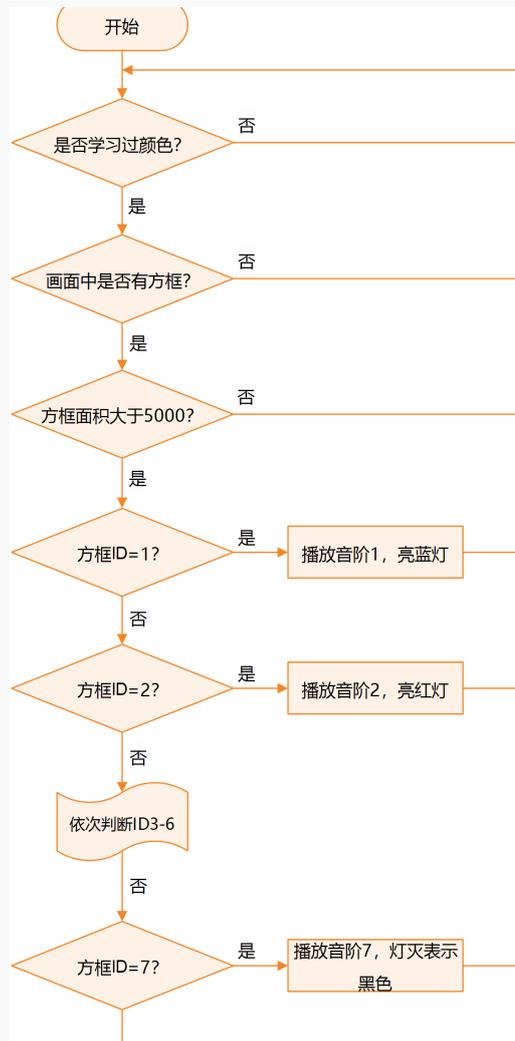
①从请求得到的“结果”中获取当前界面中靠近中心的方框信息，将“宽度”与“高度”相乘，即可得到方框面积。



HuskyLens 从结果中获取靠近中心方框的 宽度 ▾ 参数



STEP3 流程图分析



2、程序示例

程序图

修改任务二中的程序，函数“音符”不变，主程序修改如下。



3、运行效果

当HUSKYLENS显示屏画面中央为蓝色时，播放音符1，车灯为蓝色；当为红色，播放音符2，车灯成为红色；依次类推。

项目小结：

项目回顾

本节课学习了解了颜色识别的工作原理，并通过使用HUSKYLENS传感器学习了传感器的颜色识别功能。

颜色识别在人工智能视觉识别中非常重要的一个功能，在工业中有着广泛的引用。大家想想颜色识别还可以实现什么有趣的功能？

知识点回顾

- 1、了解颜色识别的工作原理；
- 2、学习了HUSKYLENS传感器的颜色识别功能和识别多个的操作方法。

项目拓展：

完成这个项目后，大家可以在网络上搜索一段简谱，用色块拼出简谱，打印出来后就可以让麦昆plus小车播放音乐啦！

但是我们一定会发现一个问题，音阶的数量比较少，不饿实现中、低、高音。如果我们要增加色块，随着颜色的增多，会有很多相近的颜色，就有可能会出现误识别。所以有没有什么办法扩宽音域呢？

(提示，可以利用micro:bit上的AB按键实现升阶和降阶的功能。)

项目2：简易不停车高速收费系统

走高速公路的时候我们可以发现，平时上下高速倒没什么，但要是碰上节假日，收费站可以说是车山人海！

随着ETC的推广，收费站拥堵问题也有所缓解。所谓ETC就是**不停车电子收费系统**，它采用电子收费的方式，ETC车主在经过收费站时，只需要低速通过收费站，就能自动扣费。

麦昆plus小车能否实现ETC功能呢？这个项目中，让我们一起来一个简易的ETC，让麦昆plus小车在上“高速”的时候，也能实现不停车收费！



功能介绍：

本项目利用HUSKYLENS的标签识别功能，用两个标签识别分别代表进、出收费站，通过记录间隔时间，算出里程和过路费，实现麦昆plus的简易ETC功能。

材料清单：



知识园地：

标签识别是机器视觉的一个分支领域，目前在生产生活中都有广泛的应用。

一、什么是标签识别？

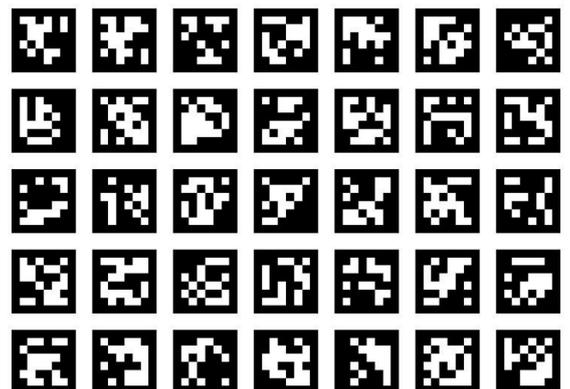
标签识别技术是指对物品进行有效的、标准化的编码与标识的技术手段。在生活中，有多种标签，比如条形码、各类二维码。



标签识别技术是指对物品进行有效的、标准化的编码与标识的技术手段。在生活中，有多种标签，比如条形码、各类二维码。

在HUSKYLENS中采用了AprilTag视觉基准库。AprilTag是通过特定的标志（与二维码相似，但是降低了复杂度以满足实时性要求），可以快速地检测标志，并计算相对位置。

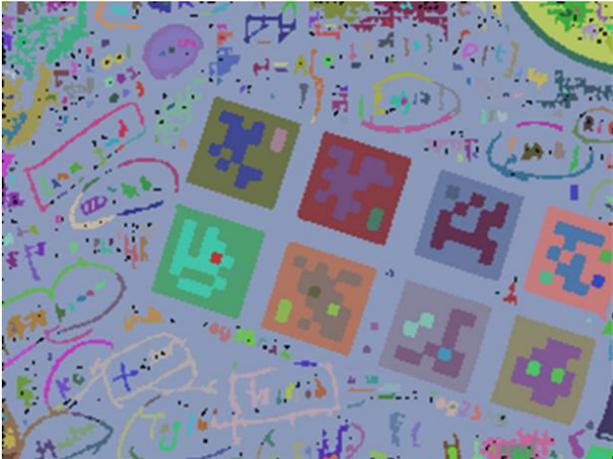
HUSKYLENS只支持识别内置的AprilTag视觉基准库标签，如右图。



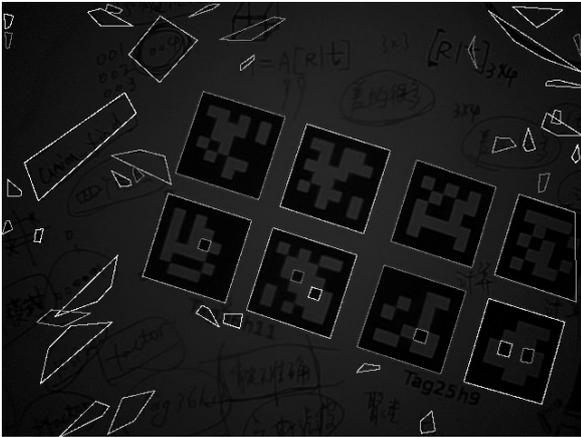
二、标签识别工作原理

AprilTag的算法主要包含如下步骤：

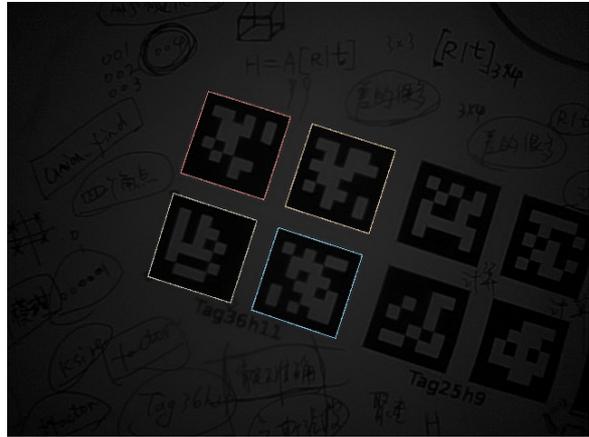
1、边缘检测，寻找图像中的边缘轮廓。



2、四边形检测，找出轮廓中的四边形。



3、解码，对找出的四边形进行匹配、检查。



三、HUSKYLENS标签识别功能演示

1、侦测标签

当HuskyLens检测到二维码标签时，屏幕上会用白色框自动框选出检测到的所有二维码标签。



2、学习标签

将HuskyLens屏幕中央的“+”字对准需要学习的标签，短按“学习按键”完成第一个标签的学习。

松开“学习按键”后，屏幕上会提示：“再按一次按键继续！按其他按键结束”。如要继续学习下一个标签，则在倒计时结束前按下“学习按键”，可以继续学习下一个标签。如果不再需要学习其他标签了，则在倒计时结束前按下“功能按键”即可，或者不操作任何按键，等待倒计时结束。

本项目中，需要学习2个标签，因此在倒计时结束前按下“学习按键”，然后将HuskyLens屏幕中央的“+”字对准需要学习的下一个标签，短按“学习按键”完成第二个标签的学习。



标签ID与录入标签先后顺序是一致的，也就是：学习过的标签会按顺序依次标注为“标签：ID1”，“标签：ID2”，“标签：ID3”，以此类推，并且不同的标签对应的边框颜色也不同。

3、识别标签

HuskyLens再次遇到学习过的标签时，在屏幕上会有彩色的边框框选出这些标签，并显示其ID。边框的大小会随着二维码标签的大小进行变化，边框自动追踪这些二维码标签。



项目实践：

HUSKYLENS的标签识别是如何使用的呢？如何计算识别标签的时间呢？让我们来分解整个项目为几个小任务，一步一步来完成简易ETC吧！

这个项目将分为三步将任务完成，首先我们会学习使用HUSKYLENS的标签识别功能，利用主控板点阵屏显示标签ID；然后将我们将学习如何使用“系统运行时间”计算两次识别标签的时间差；最后完善整个项目，实现简易ETC功能。

任务一：初识标签识别

1、硬件连接

与色彩简谱项目相同，HUSKYLENS通过支架固定在小车上，连线到IIC接口。

后面项目中都是这样接线，不再赘述。

2、程序设计

STEP1 学习与识别

这里可以任意选择两个标签，让HUSKYLENS进行学习。（注意：先开启“学习多个”的功能）



STEP2 Mind+软件设置

与色彩项目一样，打开Mind+软件（162或以上版本），切换到“上传模式”。

点击“扩展”，在“主控板”下点击加载“micro:bit”，在“扩展板”下点击加载“麦昆Plus国内版”，在“传感器”下点击加载“HUSKYLENS AI 摄像头”。

后面项目中的软件设置都是如此，不再赘述。

STEP3 指令学习

来认识一下主要用到的指令。

①从请求得到的“结果”中获取是否IDx在画面中，方框指屏幕上目标为方框的的算法，箭头对应屏幕上目标为箭头的算法，当前仅为巡线算法时选择箭头，其他算法都选择方框。



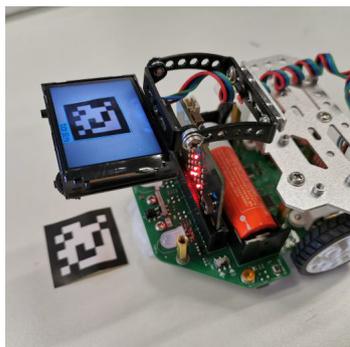
3、程序示例

程序图



4、运行效果

当HUSKYLENS识别到ID1二维码，主控板点阵屏显示1；当HUSKYLENS识别到ID2二维码，主控板点阵屏显示2。



任务二：记录两个二维码识别时间，计算时间差

1、程序设计

STEP1 功能分析

从生活经验来看，当车进入高速收费站开始计费，到从另一个收费站下去则停止计费。

对应到我们的项目中，就是识别到ID1二维码，记为开始时间；识别到ID2二维码，记为结束时间。最后得出时间差，算取里程费。

这里就可能会遇到一个问题：如果不小心识别到两次ID1二维码怎么办呢？

高速公路上一次进就和一次出对应，我们也可以采取这样一一对应的方法。当识别到一次ID1时，记录起始时间，此后只有当识别到ID2时，才会记录结束时间。

STEP2 指令学习

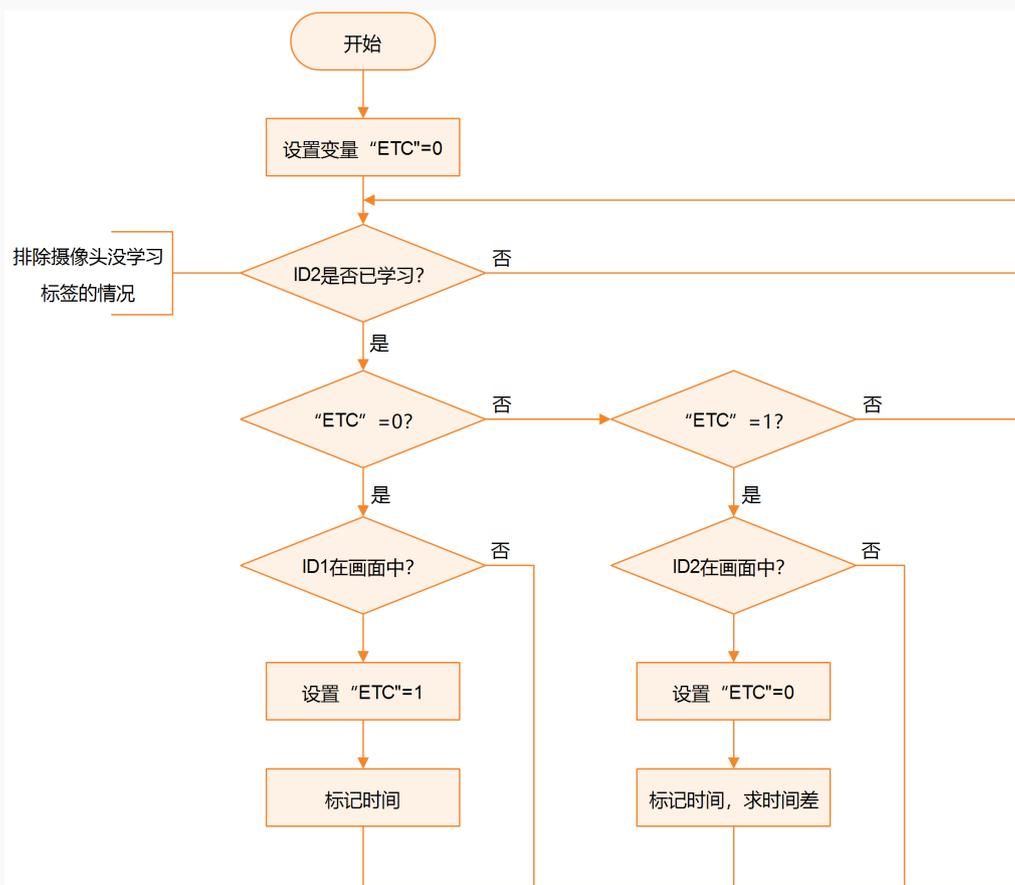
来认识一下主要用到的指令。

①表示程序开始运行后的实时时间。读取的时间单位为毫秒。表示程序开始运行后的实时时间。读取的时间单位为毫秒。



系统运行时间(ms)

STEP3 流程图分析



2、程序示例

程序图



3、运行效果

STEP1 功能分析

当识别到ID1时，串口输出一次系统运行时间，此后只有当识别到ID2时，串口再次输出系统运行时间和时间差。



任务三：启动小车

1、程序设计

STEP1 功能分析

这里为了方便看出效果，我们可以让小车在识别到ID1二维码时启动，直到识别到ID2二维码时停止。

STEP2 指令学习

来认识一下主要用到的几条指令。

①打开PID可以闭环控制轮子转速，让轮子实际转速不受环境干扰，接近设置转速。

注意：PID算法存在一定延时，不建议和灰度巡线一起用。



②设置电机速度和方向。



③设置电机停止



STEP3 速度测试

这里采用PID控制轮子转速有利于准确测试距离。为了计算出小车的实际里程，要先测算一下小车的速度。

测试程序：



测试方法：

小车运行上面程序时，会以50的速度向前跑10秒，记录起点和终点位置，测量距离S（单位：厘米），则S/10为小车的里程速度（单位：厘米/秒）。

2、程序示例

程序图



3、运行效果

当HUSKYLENS识别到ID1二维码，小车启动；此后当HUSKYLENS识别到ID2二维码，小车停止。

项目小结：

项目回顾

本节课学习了解了标签识别的工作原理，并通过使用HUSKYLENS传感器学习了标签识别的功能。

结合麦昆plus本身的PID功能，制作了一个简易ETC，项目的最后并没有计算出里程进而算出高速费用，但是相信看到这里，大家已经可以独立完成这部分数据运算，可以将最后的结果显示在点阵屏上。

知识点回顾

- 1、了解标签识别的工作原理；
- 2、学习了HUSKYLENS传感器的标签识别功能和识别的操作方法；
- 3、学习麦昆plus小车的PID控制。

项目拓展：

完成这个项目后，大家可以在网络上搜索一下自己所在地的高速收费标准，完成项目最后计算高速费用这个部分。

这里请思考一个问题：不同省份，高速收费标准不一样，但是现在有的ETC已经实现了跨省计费，中间不用进出省际高速收费口，这是怎么做到的呢？我们的项目中是否也能模拟实现省际不停车计费呢？

(提示，可以利用更多的二维码表示不同省份，每进入一个省份就识别一次，最后分段计费，算出总费用。)

项目3：AI分拣大师

在上海的洋山港码头，抬眼望去，鲜红色桥机舒展巨臂，塔吊车沿港口一字排开，轨道吊准确锁住集装箱，一个个巨大的集装箱被迅速抓起，自动引导运输车早已等候在下方，集装箱迅速装车拖向指定位置。港区内车流不息，然而繁忙的港口内却几乎空无一人。

自动引导运输车 (Automated Guided Vehicle) 简称AGV，是指具备有电磁或者光学等自动导引装置，能够沿规定的导引路径行驶，是无人驾驶的自动化车辆。

麦昆plus小车能否作为一辆AGV呢？甚至比AGV更加智能呢？比如识别出运输物的种类，根据识别结果，运输到指定位置。让麦昆plus小车成为有“眼睛”的AGV！



功能介绍：



本项目利用HUSKYLENS的**物体识别**功能，让HUSKYLENS识别不同的物品，根据识别结果，将它们运输到不同位置，让麦昆plus小车化身AI分拣大师！

为了实现精准运输，将使用麦昆plus自带的巡线传感器，通过简单的**巡线算法**，实现小车定点运输。

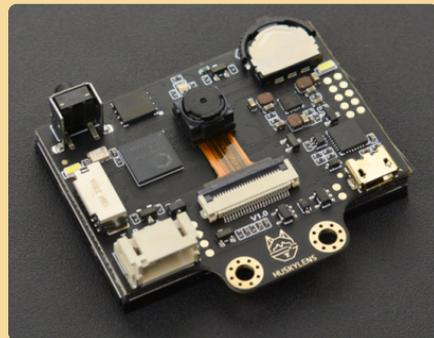
材料清单：



Micro:bit ×1



麦昆plus ×1



HUSKYLENS ×1

知识园地：

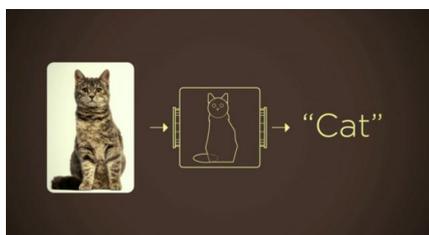
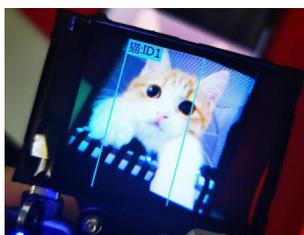


当我们谈起计算机视觉时，首先想到的就是**图像分类（物体识别）**，没错，图像分类是计算机视觉最基本的任务之一。

在这个项目中就应用了物体识别功能来区分不同种类的物体。

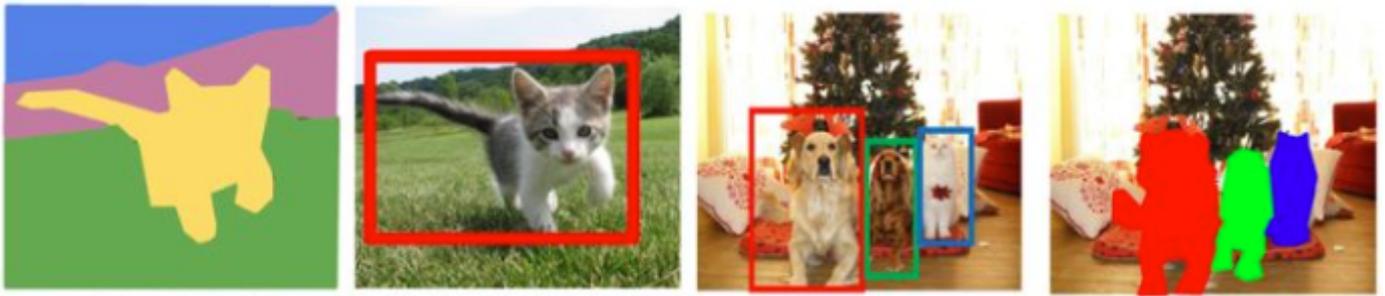
一、什么是物体识别？

物体识别主要指的是对三维世界的客体及环境的感知和认识，属于高级的计算机视觉范畴。



二、物体识别工作原理

在物体识别中，有很多复杂和有意思的任务，比如目标检测，物体定位，图像分割等。其中目标检测可以看成图像分类与定位的结合，比如给定一张图片，目标检测系统要能够识别出图片中的目标并给出其位置。



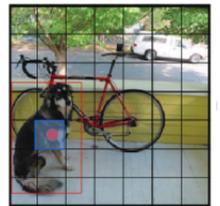
目前比较流行的目标检测算法有两类，一类是R-CNN算法，另一类是Yolo算法。R-CNN算法准确度高一些，但是速度慢；Yolo算法速度快，但是准确性要低一些。下面简单介绍一下Yolo算法。

Yolo算法：

Yolo是 You only look once的缩写，意思是你看一次就可以预测了，灵感就来自于我们人类自己，因为人看一张图片时，扫一眼就可以得知这张图片不同类型目标的位置。

Yolo胜在它的简单与快速。如果把目标检测看做是一个捕鱼的过程，其他算法是拿着渔叉一个一个精准地狙击，那么 Yolo 就粗犷的多，一个渔网撒撒下去，一网打尽。

Yolo的预测是基于整个图片的，并且它会一次性输出所有检测到的目标信息，包括类别和位置。Yolo算法的实现过程是把输入图片分割成 $S \times S$ 个网格，如果一个目标的中心点落在某个网格当中，那么对应的那个网格就负责预测这个目标的大小和类别。如下图，狗的中心点落在了蓝色的网格当中，所以蓝色的网格就负责这个目标的信息预测。



“物体中心落在哪个网格，哪个网格就负责预测这个物体”要分两个阶段来看，包括训练阶段和测试阶段。在训练阶段，如果物体中心落在这个网格，算法就教会这个网格要预测图像中的物体。在测试阶段，因为在训练阶段已经教会了网格去预测中心落在该网格中的物体，那么网格自然会继续这么做。

三、HUSKYLENS物体识别功能演示

“物体中心落在哪个网格，哪个网格就负责预测这个物体”要分两个阶段来看，包括训练阶段和测试阶段。在训练阶段，如果物体中心落在这个网格，算法就教会这个网格要预测图像中的物体。在测试阶段，因为在训练阶段已经教会了网格去预测中心落在该网格中的物体，那么网格自然会继续这么做。

1、操作设置-学习多个

向左或向右拨动“功能按键”，直至屏幕顶部显示“物体识别”。长按“功能按键”，进入物体识别功能的二级菜单参数设置界面。

向左或向右拨动“功能按键”，选中“学习多个”，然后短按“功能按键”，接着向右拨动“功能按键”打开“学习多个”的开关，即：进度条颜色变蓝，进度条上的方块位于进度条的右边。再短按“功能按键”，确认该参数。



向左拨动“功能按键”，选中“保存并返回”，短按“功能按键”，屏幕提示“是否保存参数？”，默认选择“确认”，此时短按“功能按键”，即可保存参数，并自动返回到物体识别模式。

2、侦测物体

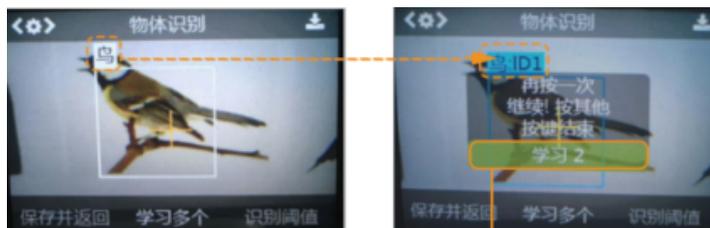
把HUSKYLENS对准目标物体，在屏幕上，会有白色框自动框选出识别到的所有物体，并显示对应的物体名称。目前只能识别并框选20种物体，其余物体无法识别和框选。



3、标记物体

把HUSKYLENS对准目标物体，当屏幕上显示的物体被检测到并显示其名字时，将屏幕中央的“+”字对准该物体的白色框中央，短按“学习按键”进行标记。此时，框体颜色由白色变为蓝色，并显示其名字和ID1，同时有消息提示：“再按一次继续，按其他按键结束”。如要继续标记下一个物体，则在倒计时结束前按下“学习按键”，可以继续标记下一个物体。如果不再需要标记其他物体了，则在倒计时结束前按下“功能按键”即可，或者不操作任何按键，等待倒计时结束。

HUSKYLENS显示的物体ID与标记物体的先后顺序是一致的，也就是：ID会按顺序依次标注为“ID1”，“ID2”，“ID3”，以此类推，并且不同的物体ID对应的边框颜色也不同。



4、识别物体

HUSKYLENS再次遇到标记过的物体时，在屏幕上会有彩色的边框框选出这些物体，并显示物体名称与ID。边框的大小随着物体的大小而变化，自动追踪这些物体。同类物体，有相同颜色的边框、名字和ID。支持同时识别多类物体，比如同时识别出瓶子和鸟。

这个功能，可以作为一个简单的筛选器，从一堆物体中找出你需要的物体，并且做追踪。

*此功能不能区分同类物体间的不同，比如：只能识别出这是猫，但不能识别出这是什么猫。有别于人脸识别，可以区分不同的人脸。



项目实践：

HUSKYLENS的物体识别是如何使用的呢？如何控制小车走指定路径呢？让我们来分解整个项目为几个小任务，一步一步来完成AI分拣大师吧！

这个项目将分为三步将任务完成，首先我们会学习使用HUSKYLENS的物体识别功能，串口输出物体名；然后将学习灰度巡线算法，实现小车定点跑；最后完善整个项目，实现分拣运输场景模拟。

任务一：初识物体识别

1、程序设计

学习与识别：这里选择3种物品，让HUSKYLENS进行学习。（注意：先开启“学习多个”的功能）



ID1: 瓶子



ID2: 自行车



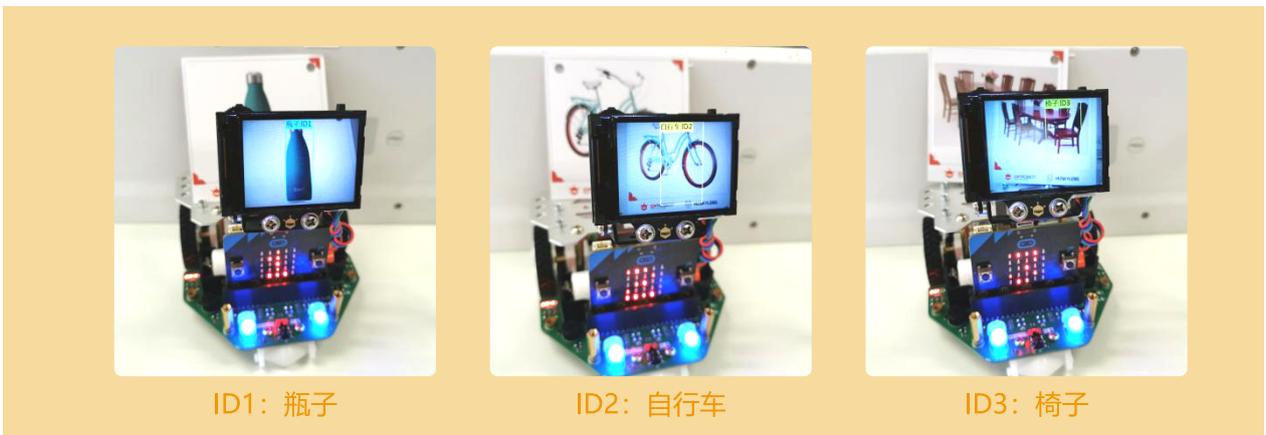
ID3: 椅子

2、程序示例

程序图



3、运行效果



当HUSKYLENS识别到瓶子，主控板点阵屏显示1，串口输出瓶子；当HUSKYLENS识别到自行车，主控板点阵屏显示2，串口输出自行车；当HUSKYLENS识别到椅子，主控板点阵屏显示3，串口输出椅子。

任务二：2路灰度巡线算法

1、程序设计

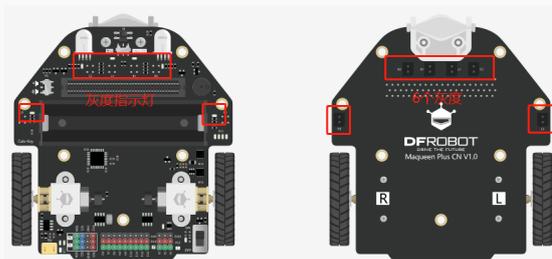
STEP1 功能分析

作为AI分拣大师，麦昆plus通过HUSKYLENS的物体识别来识别物体种类，但还要运输到指定位置，才算完成分拣任务，就像港口的AGV小车一样，完成自动运输。

对于这种点对点的精准运输，可以使用麦昆plus自带的灰度传感器，通过巡线实现。

麦昆plus底部自带6个灰度传感器，可以用来检测黑线。

当灰度面向白色背景，灰度指示灯灭，灰度传感器检测值为0；当灰度面向黑线，灰度指示灯亮，灰度传感器检测值为1。



STEP2 指令学习

来认识一下主要用到的指令。

①读取灰度传感器的值，反馈值为0或1，1表示在黑线上。在下拉框选择灰度，L1、L2、L3、R1、R2、R3与小车底部标志一致。



STEP3 灰度测试

测试结果：

当灰度L1、R1都在黑线上，串口输出1、1；

当只有R1在黑线上，串口输出0、1；

当只有L1在黑线上，串口输出1、0；

两个灰度都不在黑线上，则输出0、0。



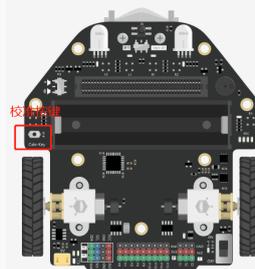
STEP3 灰度测试

如果程序无法正常运行，请排查下面问题：

①打印机打印的黑色可能无法正确识别，黑色胶带、印刷地图可以正常使用。

②环境光线可能会影响灰度传感器，当光线变化比较大，可以重新校准灰度。

灰度校准方法：麦昆plus自带一键校准灰度功能，校准按键如下图，使用时确保所有的巡线传感器都在黑色校准区域内，按下校准按键1秒钟，车前的两个RGB灯闪烁绿色，表示校准完成，松开按键即完成校准。

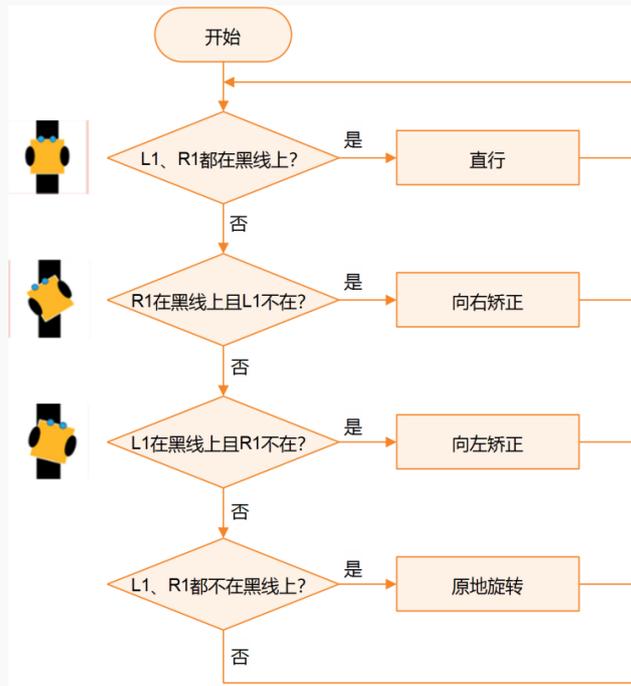


STEP3 灰度测试

* **灰度传感器原理**：每个灰度传感器由一个**红外发射器**和一个**红外接收器**组成。由于其经常被用来控制机器人沿着线走，故也称作巡线传感器。红外发射器不断向地面发射红外光，如果红外光被反射（如遇到白色或其他浅色平面），则接收器收到红外信号，输出数值0，灰度指示灯灭；若红外光被吸收或无法被反射，则接收器收不到红外信号，输出数值1，灰度指示灯亮。

STEP4 流程图分析

这里我们用两个灰度L1、R1来巡线，预设黑线宽度为2cm。当L1、R1都在黑线上时，小车直行。



2、程序示例

程序图



3、运行效果

如果大家手边没有合适的地图，建议用宽度2cm的黑胶带贴出巡线图，比如右图。



运行程序时，小车会自动巡黑线行驶。

任务三：布置场景，完善程序

1、程序设计

STEP1 功能分析

假设需要分拣3种物品，麦昆plus小车在识别出物品种类后，按照一定的巡线轨迹完成运输。

为了精准控制小车，在巡线地图中加入T型路口、左路口、右路口（如右图），对应在程序中加入路口判断程序。



利用麦昆plus上的L3、R3灰度辅助判断路口。

STEP2 路口判断程序

修改任务二中函数“读灰度”，新增函数“路口判断”、“左转”、“右转”。

```
定义 读灰度
设置 L1 的值为 0
设置 R1 的值为 0
设置 L3 的值为 0
设置 R3 的值为 0
如果 读取巡线传感器 L1 = 1 那么执行
  设置 L1 的值为 1
如果 读取巡线传感器 R1 = 1 那么执行
  设置 R1 的值为 1
如果 读取巡线传感器 L3 = 1 那么执行
  设置 L3 的值为 1
如果 读取巡线传感器 R3 = 1 那么执行
  设置 R3 的值为 1
```

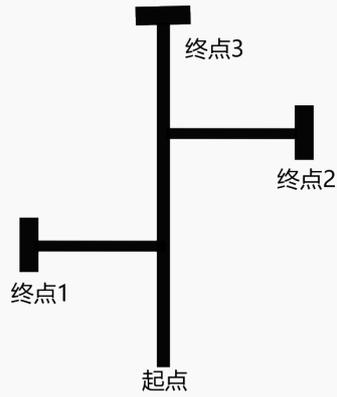
```
定义 路口判断
如果 变量 L3 = 1 与 变量 R3 = 1 那么执行
  设置 路口类型 的值为 T
否则如果 变量 L3 = 1 那么执行
  设置 路口类型 的值为 L
否则如果 变量 R3 = 1 那么执行
  设置 路口类型 的值为 R
否则
  设置 路口类型 的值为 0
显示文字 变量 路口类型
```

建议在调试程序时，单独运行上面的函数“路口判断”、“左转”、“右转”，测试每个函数是否能实现对应功能。

```
定义 左转
电机 左侧 以 60 的速度 后退
电机 右侧 以 60 的速度 前进
等待 0.9 秒
电机 全部 停止
```

```
定义 右转
电机 左侧 以 60 的速度 前进
电机 右侧 以 60 的速度 后退
等待 0.9 秒
电机 全部 停止
```

STEP3 地图示例



STEP4 流程图分析

不同的地图需要做不同的路径规划，以上面的地图为例，从起点到终点1的路径规划是：小车巡线行走直到遇到左侧路口，则小车主转，继续巡线行走找到遇到T字路口，则到达终点1。如果HUSKY-LENS识别到ID1的物体，则按照路径规划走到终点1。终点2、3以此类推。

这里延用任务一中物体识别的顺序。



ID1: 瓶子

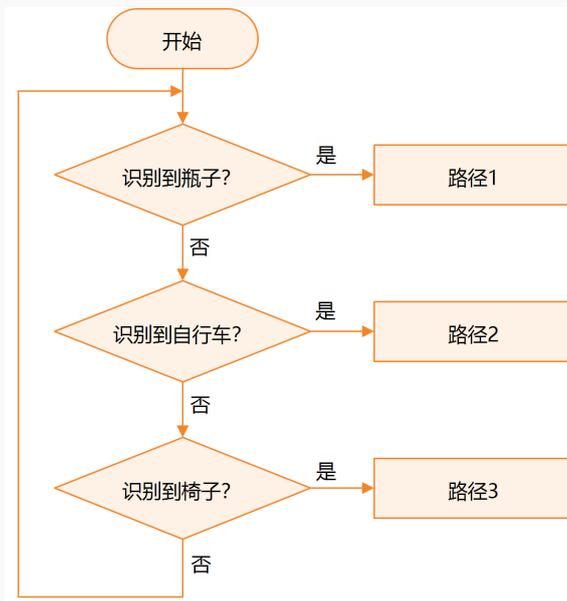


ID2: 自行车



ID3: 椅子

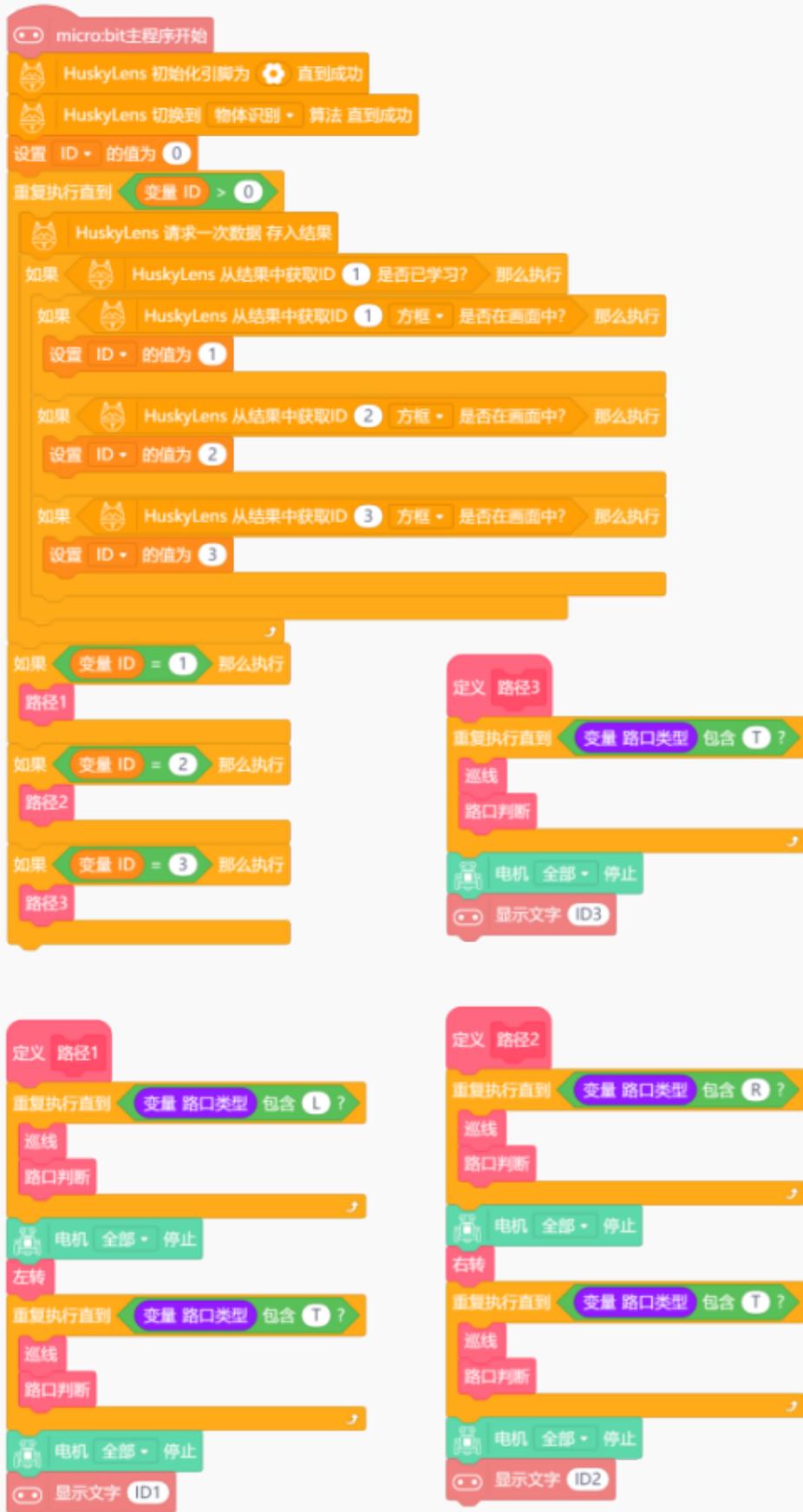
流程图如下：



2、程序示例

修改主程序，添加函数“路径1”、“路径2”、“路径3”。大家在实际编程的时候，参考自己的地图做出路径规划，下面程序使用的是前面的示例地图。

程序图



3、运行效果

HUSKYLENS在物体识别下依此学习瓶子、自行车、椅子。运行程序时，将麦昆plus小车放在起点位置。当HUSKYLENS识别到瓶子，小车会巡线行驶到终点1；当HUSKYLENS识别到自行车，小车会巡线行驶到终点2；当HUSKYLENS识别到椅子，小车会巡线行驶到终点3。

项目小结：

项目回顾

本节课认识了物体识别的工作原理，并通过使用HUSKYLENS传感器学习了物体识别的操作方法。HUSKYLENS结合麦昆plus本身的巡线传感器，让麦昆plus化身为AI分拣大师，能够自动分拣运输。想象一下，如果麦昆plus加上机械臂，能够装货或者卸货，整个场景将更加智能。

知识点回顾

- 1、了解物体识别的工作原理；
- 2、学习了HUSKYLENS传感器的物体识别功能的操作方法；
- 3、学习麦昆plus小车的巡线控制。

项目拓展：

这个项目中，我们实现了分拣运输的功能，但是小车运输到指定地点后，怎么回到起点呢？能够继续巡线回来吗？尝试用程序实现。

项目4：卧底神探

脸识别技术经过多年的发展，已经开始落地运用。比如在火车站进站时，通过摄像头比对身份证信息，还有上班时的人脸打卡、支付宝的人脸支付、智能手机上的人脸解锁等等。

人脸识别技术在公共安全领域也得到了有效的运用，比如抓捕逃犯。近年来在明星演唱会上，就已经多次成功抓获逃犯。利用AI人脸识别技术，当逃犯过安检时，摄像头捕捉到逃犯的面部信息，通过和后端数据库进行比对，确认他和数据库中的逃犯是同一个人，系统就会发出警告信息。

在HUSKYLENS中，就内置了人脸识别功能，麦昆plus能否结合HUSKYLENS去抓坏人呢？在生活中，坏人往往要主动去有人脸识别的地方才有可能被识别、抓到，那么那些没有安装人脸识别摄像头的地方要怎么抓到坏人呢？



功能介绍：



本项目利用HUSKYLENS的人脸识别功能，让麦昆plus小车化身一名神探，卧底在民间。平时能随意移动，具备基本的避障功能。通过人脸识别，将画面中的人脸跟后台数据库对比，一旦发现坏人，就发出警报。

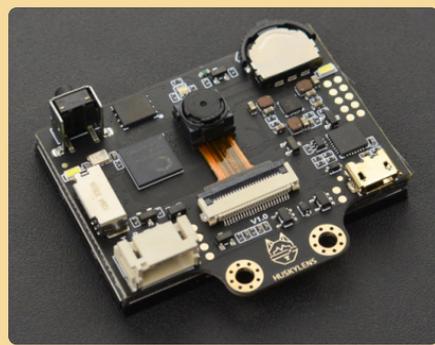
材料清单：



Micro:bit ×1



麦昆plus ×1



HUSKYLENS ×1

知识园地：



人脸与人体的其它生物特征（指纹、虹膜等）一样与生俱来，具有唯一性和不易被复制的特性。人脸图像属于最早被研究的一类图像，也是计算机视觉领域中应用最广泛的一类图像，这个项目就是利用了HUSKYLENS的人脸识别功能。

一、什么是人脸识别？

人脸识别是基于人的面部特征信息进行身份识别的一种生物识别技术。使用摄像头或者摄像机采集含有人脸的图像或视频，自动检测图像信息和跟踪人脸，对检测到的人脸进行脸部的一系列相关分析技术。



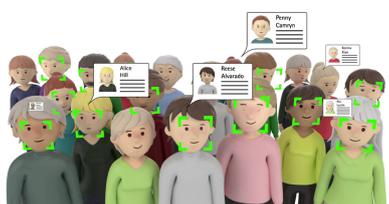
二、人脸识别工作原理

人脸识别的过程中有4个关键的步骤：

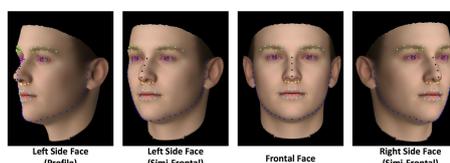


下面简单说明一下这4个步

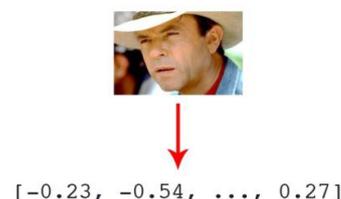
①**人脸检测**：寻找图片中人脸的位置，一般会用方框标出。



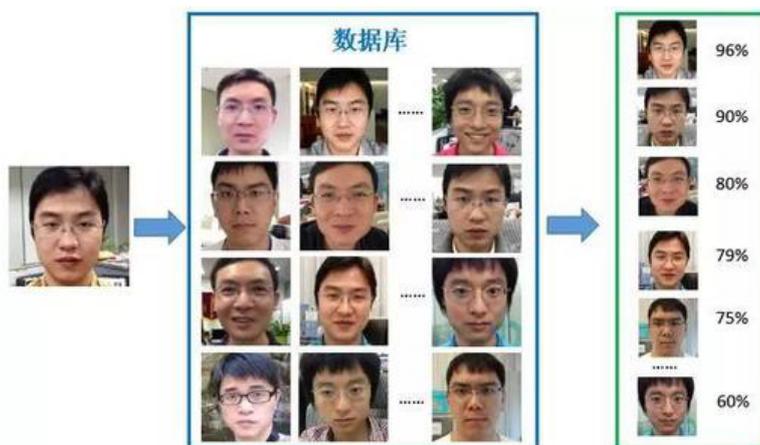
②**人脸对齐**：通过定位人脸上的特征点，识别不同角度的人脸。



③**人脸编码**：可以简单理解为提取人脸信息，转换为计算机可以理解的信息。



④**人脸匹配**：将人脸信息匹配已有的数据库，从而得到一个相似度分数，给出匹配结果。



人脸识别也被认为是生物特征识别领域甚至人工智能领域最困难的研究课题之一。**人脸识别的困难**主要是人脸作为生物特征的特点所带来的。

相似性：不同个体之间的区别不大，所有的人脸的结构都相似，甚至人脸器官的结构外形都很相似。这样的特点对于利用人脸进行定位是有利的，但是对于利用人脸区分人类个体是不利的。

易变性：人脸的外形很不稳定，人可以通过脸部的变化产生很多表情，而在不同观察角度，人脸的视觉图像也相差很大，另外，人脸识别还受光照条件（例如白天和夜晚，室内和室外等）、人脸的很多遮盖物（例如口罩、墨镜、头发、胡须等）、年龄等多方面因素的影响。

三、人脸识别应用场景

门禁系统：受安全保护的地区可以通过脸部辨识辨识试图进入者的身分，比如监狱、看守所、小区、学校等。

摄像监视系统：在例如银行、机场、体育场、商场、超级市场等公共场所对人群进行监视，以达到身分辨识的目的。例如在机场安装监视系统以防止恐怖分子登机。

网络应用：利用脸部辨识辅助信用卡网络支付，以防止非信用卡的拥有者使用信用卡，社保支付防止冒领等。

人脸识别目前在各行各业都有非常广泛的应用，例如学生考勤系统、相机、解锁手机、人证核验一体机等。



四、HUSKYLENS人脸识别功能演示

1、选择“人脸识别”功能

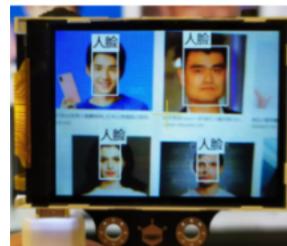
向左拨动“功能按键”，直至屏幕顶部显示“人脸识别”。



2、学习人脸

把HUSKYLENS对准有人脸的区域，屏幕上会用白色框自动框选出检测到的所有人脸，并分别显示“人脸”字样。

将HUSKYLENS屏幕中央的“+”字对准需要学习的人脸，短按“学习按键”完成学习。如果识别到相同的脸，则屏幕上会出现一个蓝色的框并显示“人脸：ID1”。这说明已经可以进行人脸识别了。



长按“学习按键”不松开，可以多角度录入人脸。

* 如果屏幕中央没有“+”字，说明HUSKYLENS在该功能下已经学习过了（已学习状态）。此时短按“学习按键”，屏幕提示“再按一次遗忘！”。在倒计时结束前，再次短按“学习按键”，即可删除上次学习的东西。



项目实践：▶

HUSKYLENS的人脸识别是如何使用的呢？如何实现麦昆plus小车的避障行走呢？如何侦查坏人呢？让我们来分解整个项目为几个小任务，一步一步来完成卧底神探吧！

这个项目将分为三步完成任务，首先学习使用HUSKYLENS的**人脸识别**功能，在主控板的点阵屏上显示对应的人脸ID；然后学习如何让小车**避障行走**；最后完善整个项目，让麦昆plus一边避障行走，一边侦查坏人，如果发现坏人，则**发出警报**。

任务一：初识人脸识别

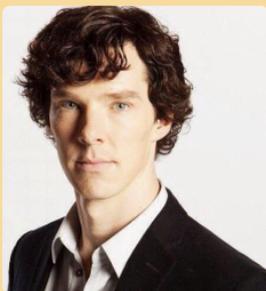
1、程序设计

STEP1 学习与识别

这里可以随意选择几张人脸图片，让HUSKYLENS进行学习。（注意：先开启“学习多个”的功能）



ID1



ID2



ID3

STEP2 指令学习

来认识一下主要用到的指令。

①从请求得到的“结果”中获取当前算法下已经学习了多少个目标，注意HUSKYLENS端长按选择键开启高级设置后可以设置是否学习多个目标。



HuskyLens 从结果中获取 已学ID总数

STEP3 功能分析

假设只有3个人脸信息，程序是很容易实现的。如下图，只需要一个一个判断就好。



但是当学习过的人脸是5个、10个甚至更多，如何用程序实现呢？一个一个判断未免有点啰嗦，从上面的程序中你有没有发现一些规律呢？

STEP3 功能分析

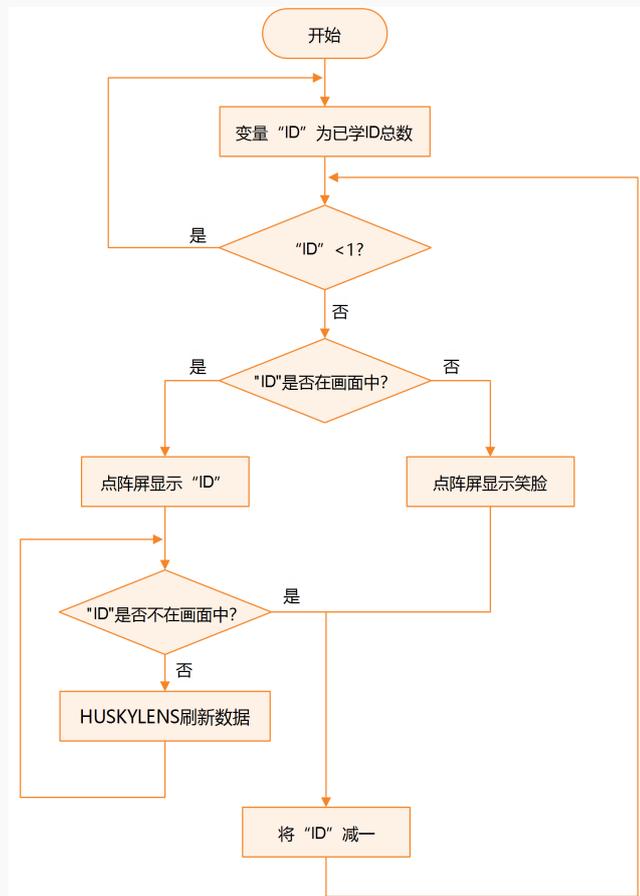
其实只要加入一个变量，就可以将上面程序中重复性高的部分简化，程序如下，这样无论学习过的ID有多少个，都可以一一判断。



STEP4 流程图分析

完善程序功能，当画面中没有学习过的人脸时，点阵屏显示笑脸，当有学习过的人脸时，显示其ID。

程序流程图分析如下，逻辑可能会有点复杂，大家可以利用流程图去理清程序逻辑，也可以通过程序去理解流程图实现功能的方法。

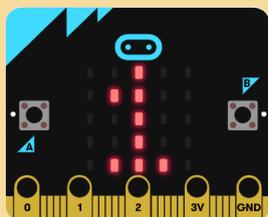


2、程序示例

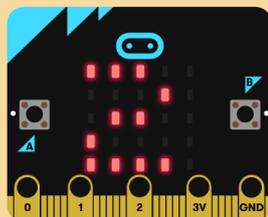
程序图



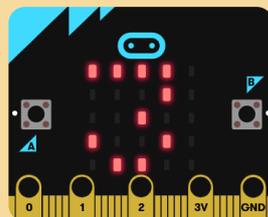
3、运行效果



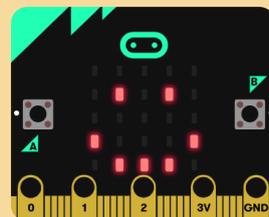
识别到ID1



识别到ID2



识别到ID3



无ID在画面中

任务二：避障行走

1、程序设计

STEP1 功能分析

麦昆plus作为一个卧底在民间的神探，需要具备的基本功能就是自主行走，那遇到障碍物怎么办呢？如何自动避障呢？

你可能会想，摄像头可以用来避障吗？其实基本的避障功能根本用不上摄像头，麦昆plus小车自己就可以搞定。一起来看看吧！

麦昆plus的电机自带编码器，有了编码器就可以实时读取电机的速度。当小车行驶过程中遇到障碍物，小车被迫停下，电机速度接近于0。所以可以通过读取电机速度来判断是否遇到障碍物，如果遇到的话，小车就做一些避障动作，比如先后退再转向。

STEP2 指令学习

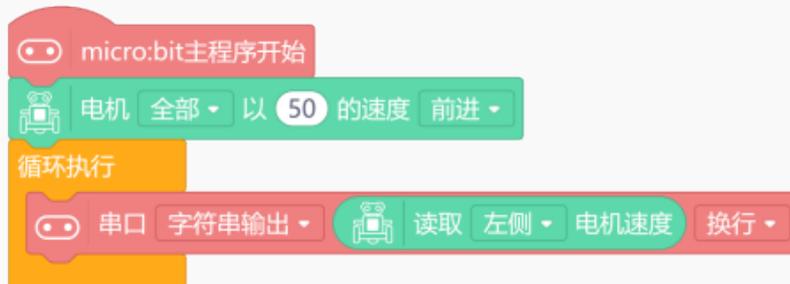
来认识一下主要用到的指令。

①读取电机当前运行速度，可以选择左侧或右侧电机。



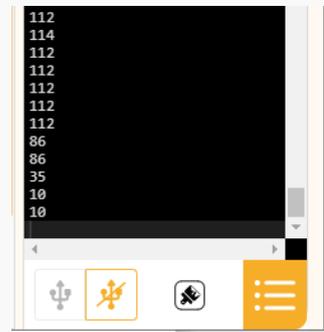
STEP3 测试程序

通过串口输出电机速度，以左侧电机为例。测试程序：



测试结果：串口区显示左侧电机速度，用手给左轮一点阻力，可以明显看到电机速度降低。

*大家在看速度时会发现电机读取的速度与程序设置的速度有一定偏差，这与电池电量、轮子受到的摩擦力等都有关系。还记得上个项目中学习的PID控制吗？其实PID就是通过编码器读取电机速度后，通过算法去实时调节速度，让电机转速尽量接近设置速度。



2、程序示例

程序图



3、运行效果



小车自动避障行走，当遇到障碍物时，小车先后退再转向，自动避开障碍物。

*程序启动时，由于电机速度没有立刻变大，会先执行避障的动作。这只发生在刚刚启动时。

任务三：遇到坏人，发出警报

1、程序设计

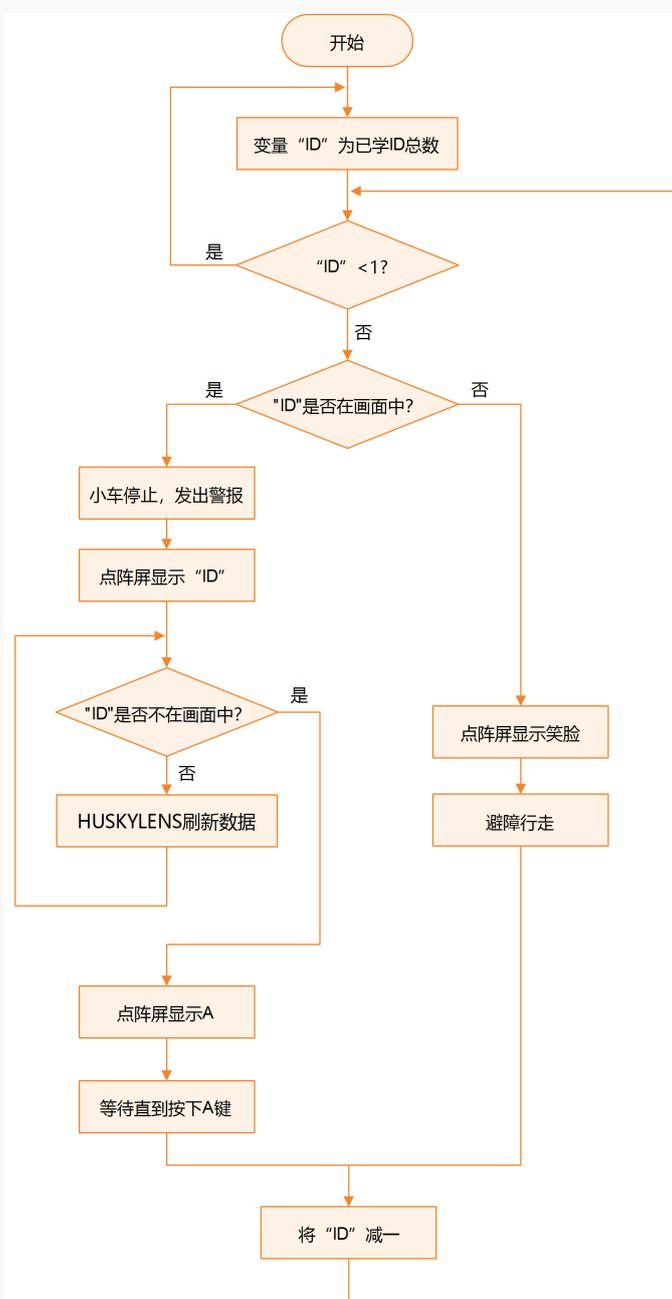
STEP1 功能分析

任务一实现了人脸识别的功能，任务二实现了避障行走的功能，将两个功能结合在一起，就是我们想要的卧底神探！

当HUSKYLENS没有识别到指定人脸时，小车一直避障行走；当识别到指定人脸时，小车停止运动，在原地发出警报，并在主控板点阵屏上显示识别到的人脸ID。

STEP2 流程图分析

为了让程序持续运行，当识别到人脸后，如果人脸不再在画面中，按下主控板上的A键，即可重启程序。流程图如下：



2、程序示例

程序图

函数“避障”不变，修改主程序，完整程序如下。



3、运行效果

程序图

先让HUSKYLENS在人脸识别下依此学习多个人脸。运行程序时，将麦昆plus放在任意位置，小车将**自动避障行走**。

行走过程中如果HUSKYLENS看到已经学习过的人脸，小车停止运动、**发出警报**、主控板点阵屏显示对应人脸ID。此后，当HUSKYLENS画面中没有学习的人脸时，主控板点阵屏显示A，表示**按下A键即可重新启动麦昆plus**，继续搜索坏人！

项目小结：

项目回顾

本节课认识了**人脸识别**的工作原理，并通过使用HUSKYLENS传感器学习了人脸识别的操作方法。

HUSKYLENS结合麦昆plus本身的**电机编码器**，让麦昆plus化身为卧底神探，能够在避障行走的同时，不停的甄别采集到的人脸信息。

知识点回顾

- 1、了解人脸识别的工作原理；
- 2、学习了HUSKYLENS传感器的人脸识别功能的操作方法；
- 3、学习麦昆plus小车的电机编码器的作用。

项目拓展：

这个项目中，我们的神探麦昆plus可以搜寻到隐藏在人群中的坏人，但是抓到坏人以后能不能自动跟随坏人呢？让坏人们在AI人脸识别下无处可逃！

如果你觉得追踪有一定的难度，在下一个项目中就会讲到怎么跟踪哦，一起来继续学习吧。

项目5：宠物小精灵

说起皮卡丘大家肯定都不陌生吧！它是动画《精灵宝可梦》中主人公小智的第一只精灵，在被小智驯化后，就始终陪伴在他左右，和他一起磨炼和成长。

这样的陪伴真是让人十分羡慕，现实生活中虽然没有真实的宝可梦，但是谁不想有这样一只始终陪伴在自己左右的宝可梦呢！

想一想，我们能不能自己动手创建一个宠物小精灵呢？



功能介绍：



本项目利用HUSKYLENS的物体识别功能，让麦昆plus小车变成你的专属宠物小精灵，能够灵活移动，始终跟随在你身后。

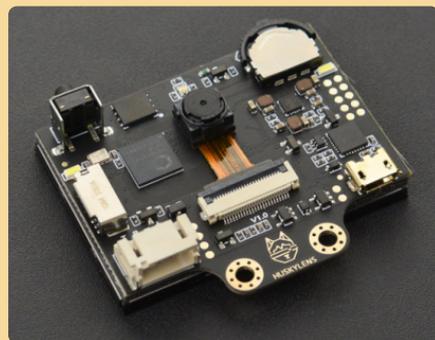
材料清单：



Micro:bit ×1



麦昆plus ×1



HUSKYLENS ×1

知识园地：



当我们需要追踪一个活动的物体，除了人工操作以外就需要使用到视觉物体追踪技术了，这项技术广泛应用在我们的生活中，如视频监控、无人机跟随拍摄等。这个项目就是利用HUSKYLENS的物体追踪功能。

一、什么是物体追踪？

物体追踪是计算机视觉中的一项重要任务，是指对视频序列中的目标状态进行持续推断的过程，简单来说就是识别指定目标并追踪。



二、物体追踪工作原理

通过单摄像头采集图像，将图像信息传入计算机，经过分析处理，计算出运动物体的相对位置，同时控制摄像头转动，对物体进行实时追踪。

物体追踪系统执行追踪功能时主要分为四步：识别物体、追踪物体、预测物体运动、控制摄像头。



识别物体



追踪物体



预测物体运动



控制摄像头

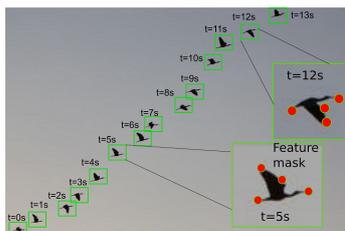
①**识别物体**：在静态背景下,通过一些图像处理算法,得到较为精确的运动物体信息。



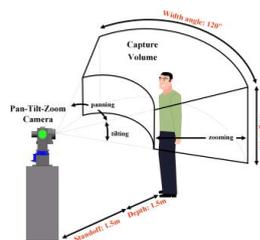
②**追踪物体**：通过上一步得到的运动物体点的位置,根据物体的色彩概率分布,使用算法对后面的图像序列进行跟踪,并且可以在后续的追踪中进行更深入的学习,使追踪越来越精确。



③**预测物体运动**：为了提高效率采用算法进行计算,预测下一帧运动物体图像的位置。可以优化算法并提高效率。



④**控制摄像头**：在采集图像信息的同时移动摄像头,使摄像头跟着物体移动的方向调整方向,一般需要配合云台或其他运动机构来实现。



三、物体追踪应用领域

智能视频监控：基于运动识别（人类识别、自动物体检测等），自动化监测（监视一个场景以检测可疑行为）；交通监视（实时收集交通数据用来指挥交通流动）。



人机交互：传统人机交互是通过计算机键盘和鼠标进行的,为了使计算机具有识别和理解人的姿态、动作、手势等能力,跟踪技术是关键。



虚拟现实：虚拟环境中3D交互和虚拟角色动作模拟直接得益于视频人体运动分析的研究成果,可给参与者更加丰富的交互形式,人体跟踪分析是其关键技术;

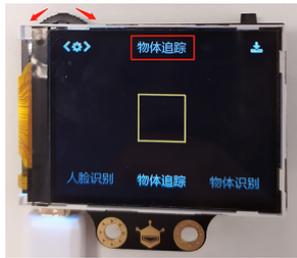


四、HUSKYLENS物体追踪功能演示

HUSKYLENS内置物体追踪功能,通过对物体的特征进行学习而实现对物体的追踪,并将位置信息反馈给主控。与颜色识别或人脸识别等其他功能不同,物体追踪是可以对一个物体(或人)进行完整的学习并进行识别的,颜色识别只针对颜色,而人脸则只是人体的一部分,物体追踪则是针对这个物体的整体特征进行学习从而进行追踪。物体追踪功能只能追踪一个物体,暂不支持追踪多个物体。学习的物体最好有明显的轮廓,这样更容易被识别。

1、选择“物体追踪”功能

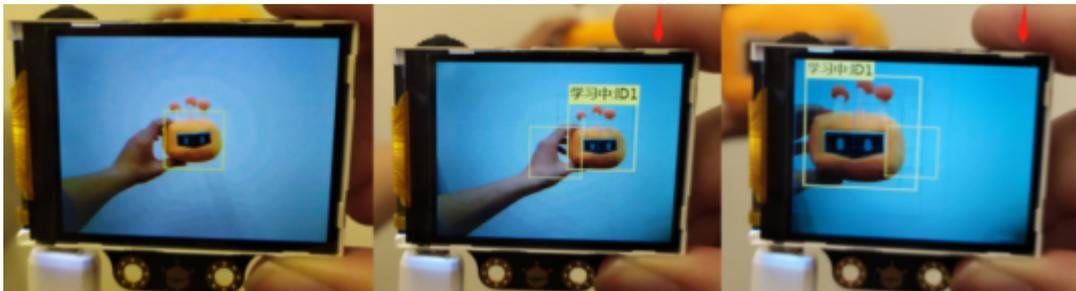
向左拨动“功能按键”，直至屏幕顶部显示“物体追踪”。



2、学习

把HUSKYLENS对准需要追踪的物体，调节物体与HUSKYLENS的距离，将物体包含在屏幕中央的橙黄色方框内，保证方框内包含物体特征鲜明的局部特征亦可。

长按“学习按键”不松开，并调整角度和距离，使得HUSKYLENS从不同的角度和距离学习该物体。学习过程中，屏幕上的黄框会标注“学习中：ID1”。



当HUSKYLENS在不同的角度和距离都能追踪到该物体时，就可以松开“学习按键”结束学习了。

* 如果屏幕中央没有橙黄色方框，说明HUSKYLENS之前学习过一个物体，请选择“忘记学过的物品”后重新学习。

3、保持学习

在物体追踪功能下，可以让HUSKYLENS保持学习，即只要摄像头看到所学物体，就不停的学习物体当前状态，这样有利于捕捉动态的物体。

操作方法：长按“功能按键”，进入物体追踪功能的二级菜单，选中“学习开启”，然后短按“功能按键”，再拨动“功能按键”打开“学习开启”的开关，即：进度条颜色变蓝，进度条上的方块位于进度条的右边。退出时选择“确认”，保存参数。



4、保存模型

当重启HUSKYLENS时，默认不保存上一次学习的物体，可以开启重启摄像头保存物体的功能。

操作方法：与上面相同，进入二级菜单后，打开“保存模型”功能。这样只需要学习一次物体，以后重启摄像头都会保存上一次学习的物体。

项目实践：▶

HUSKYLENS的物体识别是如何使用的呢？如何让麦昆plus小车紧紧跟随主人呢？让我们来分解整个项目为几个小任务，一步一步来完成自己的宠物小精灵吧！

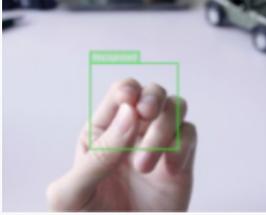
我们将分为三步完成任务，首先学习使用HUSKYLENS的**物体识别**功能，在串口区输出各项参数；然后学习如何让小车跟随指定目标，当目标移动时，可以进行左右调整和前后调整；最后完善整个项目，让麦昆plus能够按照要求跟随指定物体。

任务一：初识物体识别

1、程序设计

STEP1 学习与识别

这里可以选择一个轮廓特征明显的物体，比如手势。



STEP2 指令学习

来认识一下主要用到的指令。

①从请求得到的“结果”中获取IDx的参数，如果此ID在画面中没有或没有学习则会返回-1。



STEP3 坐标分析

HUSKYLENS传感器屏幕分辨率为320*240，如右图。我们通程序获取的物体中心点坐标也就在这个范围之内。比如获取的坐标值为(160, 120)那现在追踪的物体就在屏幕的中心。



方框参数的“X中心”、“Y中心”是指识别框的中心点在屏幕坐标系中的位置。

方框参数的“宽度”、“高度”是指识别框的大小，在物体识别功能下，识别框为正方形，所以宽度与高度相等。

STEP4 功能分析

通过串口查看方框的各个参数。

2、程序示例

程序图



3、运行效果

在Mind+的串口区，打开串口开关，可以看到各个参数。尝试左右移动物体，观察X中心的数值变化；上下移动物体，观察Y中心的数值变化；前后移动物体，观察宽度、高度的数值变化。

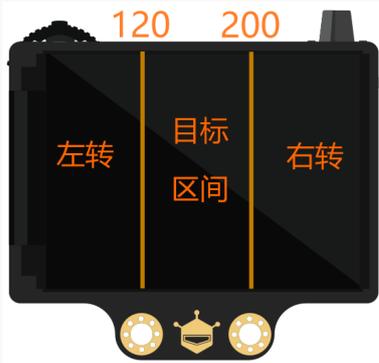


任务二：左右调整

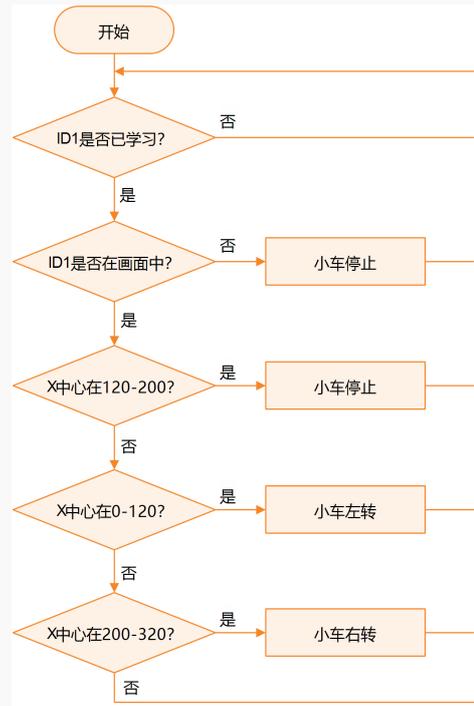
1、程序设计

STEP1 功能分析

如下图，按照摄像头屏幕坐标系的X轴将屏幕划分为3个区间，中间区间就是我们的目标区间。摄像头不停检测画面中目标物体的状态，当其X中心在120-200时，说明目标在视野中央，小车不用调整位置；当其X中心在0-120时，小车右转调整；当其X中心在200-320时，小车左转调整。



STEP1 功能分析



2、程序示例

程序图



任务三：前后调整

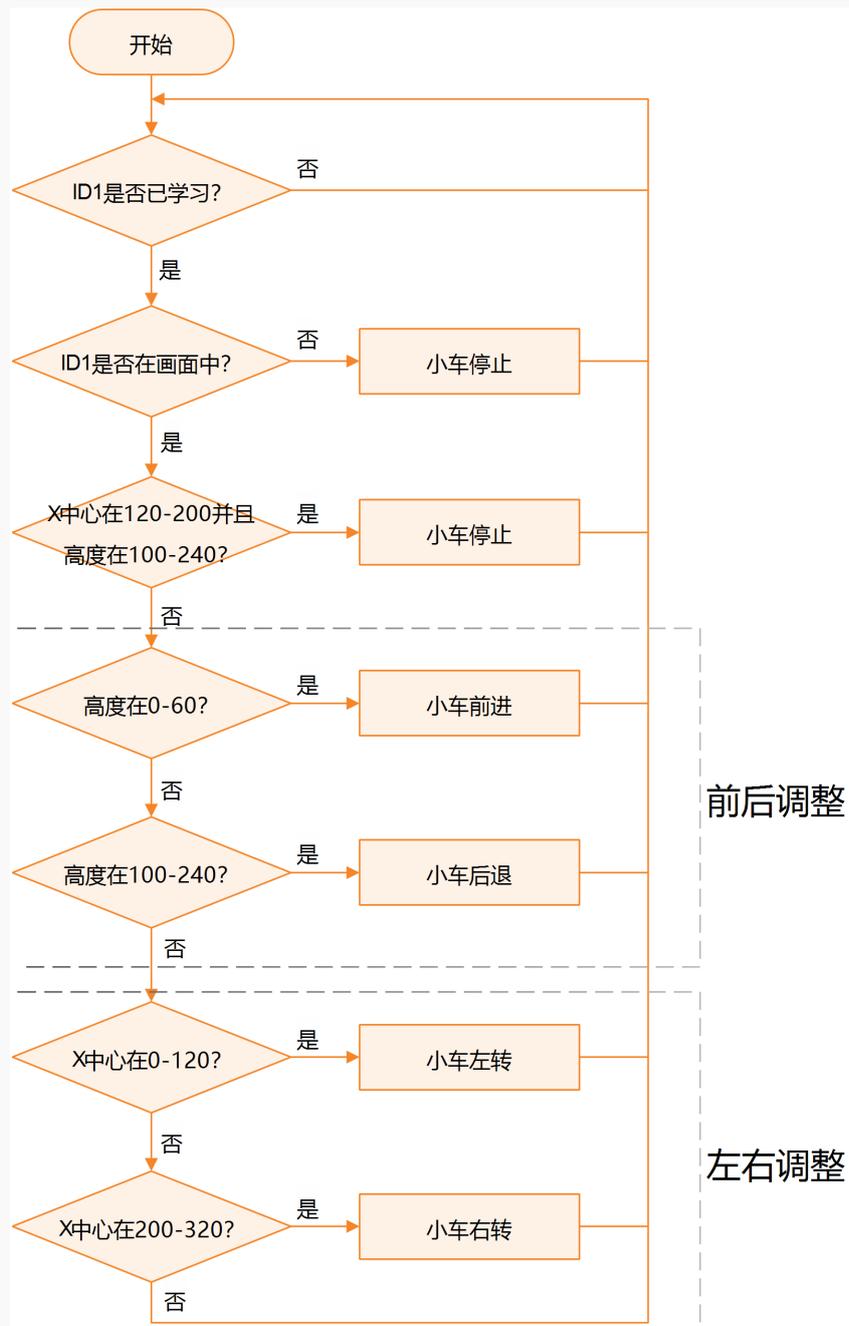
1、程序设计

STEP1 功能分析

作为主人的宠物小精灵，还要能跟随主人一起运动，当距离主人比较远的时候，会自动追过去，当距离太近的时候，又会后退到安全距离。

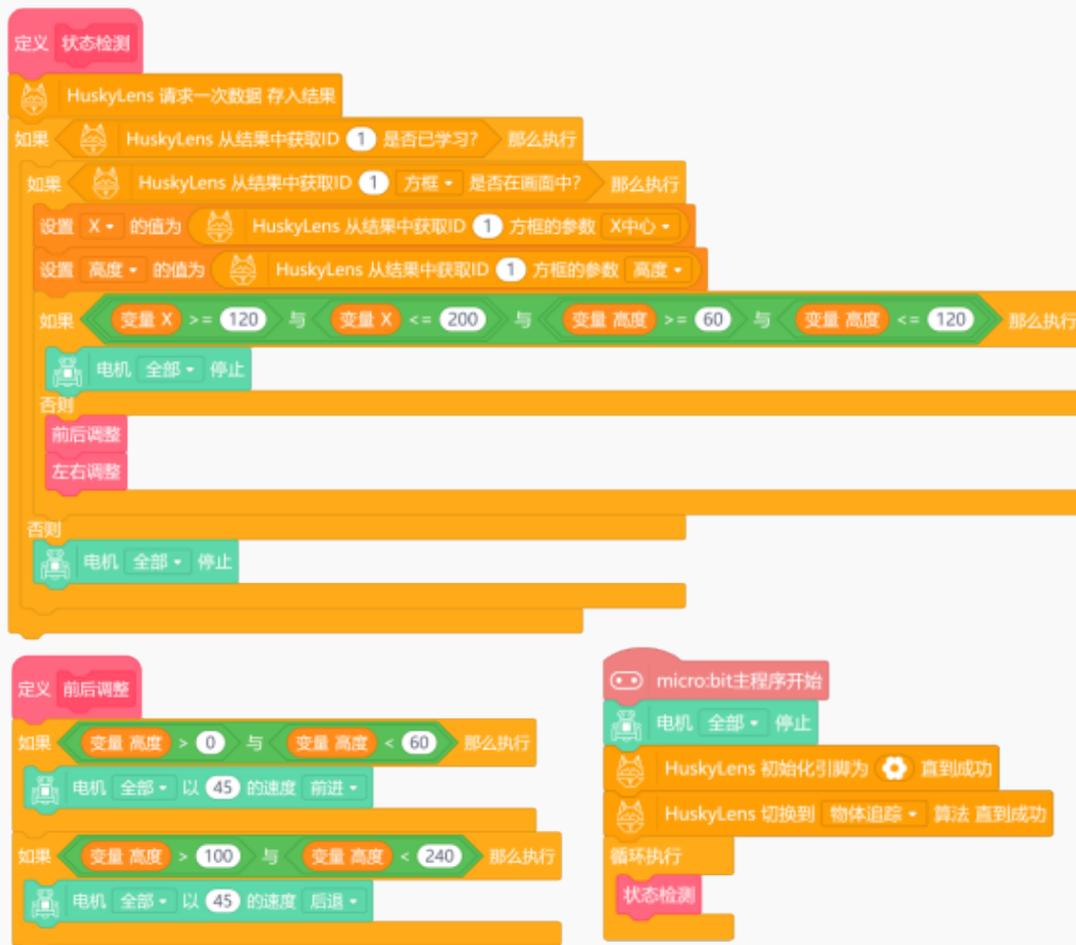
STEP2 流程图分析

在任务二的基础上，通过方框高度，判断小车到目标物体的距离，根据结果做前后调整。



2、程序示例

程序图



3、运行效果

在麦昆plus学习完成一个物体后，会自动跟随这个物体前、后、左、右移动，始终保持物体方框在画面中央，并且保持在一个合适的距离。

项目小结：

项目回顾

本节课认识了物体识别的工作原理，并通过使用HUSKYLENS传感器学习了人脸识别的操作方法。

HUSKYLENS传感器可以输出目标方框的X中心、Y中心、高度和宽度，利用这些参数可以判断出目标物体的相对位置以及距离。

当麦昆plus小车作为一辆跟踪小车时，这些参数就可以辅助定位目标物体，这在HUSKYLENS传感器的其他功能下依然有效，比如你也可以将这个�目变成一个人脸跟随小车，只要看到人脸，就让小车跟随移动。

知识点回顾

- 1、了解物体识别的工作原理；
- 2、学习了HUSKYLENS传感器的物体识别功能的操作方法；
- 3、学习使用HUSKYLENS让麦昆plus小车跟随目标移动。

项目拓展：

就像动画片中，主人公驯服自己的小精灵让它更加强大一样，能不能让你的麦昆plus小车也有更多功能呢？比如加入**手势控制**，当你的手向左画圈，小车就向左打转，当你的手往前推，小车就往前走。你能用程序实现吗？

项目6：循“轨”蹈矩

如今的机器人真是越来越厉害了，它们上天入地，几乎无所不能。你看它们有的在餐厅当送餐员，服务周到；有的在工厂车间当“快递小哥”，任劳任怨；有的在电网枢纽当安全检查员，尽职尽责...



如果大家仔细观察，会发现这些机器人工作的地面上有线条，它们就是循着线条行进的。巡线？它们是怎样做到的？

HUSKYLENS也有巡线功能，想象一下，如果家里贴上一些指引线，麦昆plus是不是也能像机器人餐厅的送餐机器人一样，成为我们的生活小助手呢？

让我们DIY一个循“轨”蹈矩的麦昆plus小车来一探究竟吧！

功能介绍：



本项目利用HUSKYLENS的巡线功能，让麦昆plus机器人按照地面上的线路轨道行进。

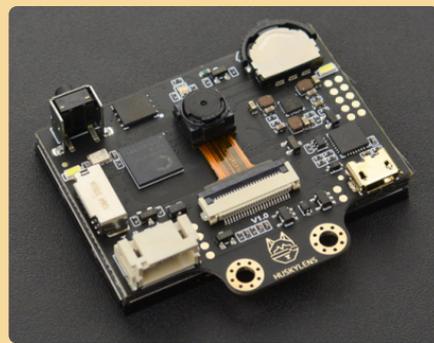
材料清单：



Micro:bit ×1



麦昆plus ×1



HUSKYLENS ×1

知识园地：



如果我们要让小车机器人按照地面上的线条移动，就需要一些传感器来识别这些线条，根据传感器的不同，巡线方式也分几种，我们这个项目是使用视觉传感器——HUSKYLENS的巡线功能来实现巡线效果的。

一、什么是巡线？

巡线是指移动物体按照指定线路行驶的过程。功能完整的巡线机器人是以移动机器人作为载体，以可见光摄像机、红外热成像仪、其它检测仪器作为载荷系统，以机器视觉—电磁场—GPS—GIS的多场信息融合作为机器人自主移动与自主巡检的导航系统，以嵌入式计算机作为控制系统的软硬件开发平台。

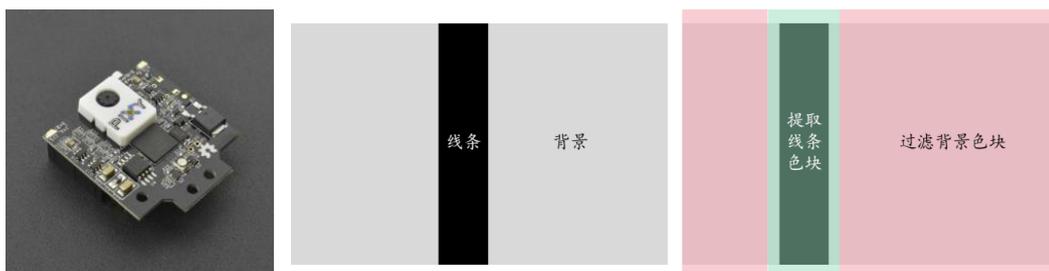
二、两种常用巡线方式的对比

种类 对比项	红外巡线传感器	视觉传感器
成本	低	高
视野	传感器需贴近地面使用，视野范围小	视野范围大，能根据线条变化提前调整运动状态
环境适应	改变使用环境时，需要调节传感器灵敏度，且调节过程繁琐	改变使用环境时，只需要重新学习线条即可，操作简单
地图适应	一般只适合背景线条颜色分明、黑色与白色线条、实线的简单地图	可适应背景线条颜色分明、多种颜色线条、实线与虚线等复杂条件下的地图

三、巡线功能原理

HUSKYLENS的巡线功能是基于卡内基梅隆大学的开源项目pixy实现的。

pixy的算法能对图片进行颜色识别，其基本思想是利用颜色空间来除去所有用户不感兴趣的背景，提取出前景（如线条）。



如何让麦昆plus在白底黑线的巡线地图上循着黑线运动呢？其实我们只需要知道麦昆plus相对黑线是什么位置，分下面三种情况：

- 1、当麦昆plus在黑线的偏右位置时，控制麦昆plus向左转弯；
- 2、当麦昆plus与黑线相对居中时，控制麦昆plus直行；
- 3、当麦昆plus在黑线的偏左位置时，控制麦昆plus向右转弯。

具体应该怎么实现呢？我们将HuskyLens在巡线过程中屏幕上显示的信息剥离出来，抽象成下图的几何数学模型。

HuskyLens屏幕的分辨率是 320×240 ，屏幕左上角的O点为屏幕的坐标原点 $(0, 0)$ ，水平向右方向为 X 轴正方向，竖直向下方向为 Y 轴正方向，因此屏幕右下角的坐标为 $(320, 240)$ 。上图中橙色虚线为屏幕的中轴线，这条线的横坐标值为160。上图屏幕中黑色的线，是HuskyLens摄像头“看到”的巡线地图线条，蓝色箭头为HuskyLens计算出来的线条方向，蓝色箭头的起点坐标为 $(x1, y1)$ ，终点坐标为 $(x2, y2)$ 。

综上，所以我们只需要判断蓝色箭头的起点相对中轴线的位置就能实现巡线了。



四、HUSKYLENS巡线功能演示

1、学习物体

本功能可以追踪指定颜色的线条，做路径预测。默认设置为只追踪一种颜色的线条。本项目以只追踪一种颜色的线条为例进行说明。

操作设置：

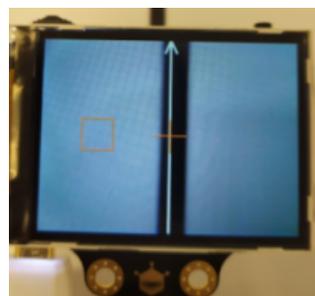
- ①向左或向右拨动“功能按键”，直至屏幕顶部显示“巡线”。
- ②长按“功能按键”，进入巡线功能的二级菜单参数设置界面。
- ③向左或向右拨动“功能按键”，选中“学习多个”，然后短按“功能按键”，查看“学习多个”的开关是否处于关闭状态。如果没关，就向左拨动“功能按键”关闭“学习多个”的开关，即：进度条颜色变白，进度条上的方块位于进度条的左边。再短按“功能按键”，确认该参数。
- ④如果环境光线比较暗，可以**打开补光灯**。参照上述方法，将“LED开关”打开即可。



⑤向左拨动“功能按键”，选中“保存并返回”，短按“功能按键”，屏幕提示“是否保存参数？”，默认选择“确认”，此时短按“功能按键”，即可保存参数，并自动返回到巡线模式。

2、学习与追踪

学习线条：将HUSKYLENS屏幕上的“+”字对准目标线条，将橙黄色的方框对准背景色。建议HUSKYLENS的视野范围内只有需要学习的线条，并且没有交叉线。尽量将HUSKYLENS与目标线条保持平行，然后HUSKYLENS会自动检测线条，并出现白色的箭头。然后短按“学习按键”即可，白色箭头变成了蓝色箭头。



巡线追踪：当HUSKYLENS检测到学习过的线条时（即：同一种颜色的线条），HUSKYLENS的屏幕上会显示蓝色的箭头，箭头的指向表示路径预测的方向。



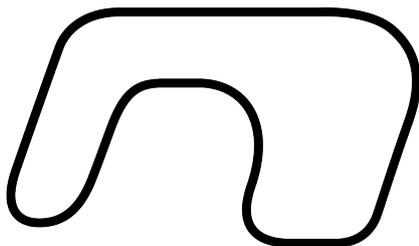
- *1、学习线条时，尽量把HUSKYLENS的位置调节到与线条平行。
- *2、识别的线条可以是任何与背景有明显色差的单色线条，建议线条与背景色有足够的色差，以保证巡线的稳定。
- *3、HUSKYLENS可以根据线条的颜色不同，进行多个线条的学习与巡线，但是线条必须是与背景有明显色差的单色线条。大多数情况下，巡线用的线条，只有一种颜色。为了保证稳定性，建议只对一种颜色的线条进行巡线。
- *4、线条的颜色与环境光线有很大关系，建议巡线的时候，保持环境光线的稳定。

项目实践：

我们将按照巡线逻辑实现项目，分若干步不断优化巡线效果，使麦昆plus能既快又稳地通过地图。首先我们会学习使用HUSKYLENS的巡线功能，读取线条的横坐标数据，编写简单的调整运动状态程序满足巡线要求，然后根据调试效果改进我们的项目程序。

任务一：巡线算法1

下图为参考巡线地图。



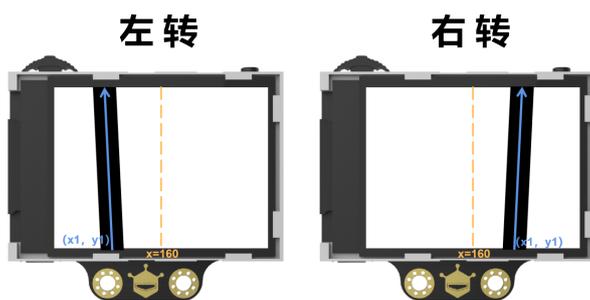
1、结构搭建

使用螺丝固定哈士奇与麦昆plus，需要注意的是，为了巡线，我们需要将摄像头斜向下调节，这样能够看到离麦昆小车更近距离的黑线，巡线效果更佳。



2、程序设计

当HUSKYLENS检测到黑线在屏幕的左边时，即蓝色箭头的起点横坐标值 $x_1 < 160$ ，控制麦昆plus左转；当黑线在屏幕的右边时，此时 $x_1 > 160$ ，控制麦昆plus右转；当黑线在屏幕的中间时，此时 $x_1 = 160$ ，控制麦昆plus直行。



3、程序示例

程序图

4、运行效果

在HUSKYLENS巡线功能下，学习线条和背景，摄像头朝下即可巡线。

麦昆plus能完成基本的巡线任务，但同时也暴露如下几个问题：

- ① 麦昆plus在前进中明显左右晃动，速度变化不连贯，不能稳定直行运动；
- ② 速度不能设置太快，速度太快在转弯处很容易脱线；
- ③ 不同的转弯角度，需要的转弯速度也不一样，当巡线地图中有好几种转弯角度时很容易脱线。

* 如果转弯过程中麦昆plus脱离轨道，就需要通过修改左右侧电机的速度，不断调试。

如何优化呢？一起来看看吧！

任务二：巡线算法2

1、程序设计

任务一中麦昆plus在前进中左右晃动，不能稳定直行运动，这是为什么呢？因为麦昆plus直行的横轴区间只是一条线，而运动过程中带有惯性，所以在 $x=160$ 处直行是很难实现的。

优化思路是将直行的运动区间扩大，如下图所示，我们将区间 $[150, 170]$ 设置为麦昆plus直行区间，当起点坐标值 x_1 在这个区间内时，控制麦昆plus直行；当 $x_1 < 150$ 时，控制麦昆plus左转；当 $x_1 > 170$ 时，控制麦昆plus右转。



2、程序示例

程序图

在任务一程序基础上，主程序不变，函数“巡线”修改如下。



3、运行效果

通过这些调整，我们发现，小车的巡线效果好一些了，遇到弧线时，也能基本不脱线。但是在转弯角度变化比较大的巡线地图中，还是很容易脱线。

任务三：巡线算法3

尽量避免小车入弯时脱线。

1、程序设计

我们继续来调节，既然可以将检测到的黑线位置分成 3 个区间，我们是不是能如法炮制，将区间继续分为 5 个呢？黑线位置越靠两边，转弯速度越大；黑线位置越趋近于中间，越趋向于直行运动。

或者更进一步，直接分为 7 个调速区间呢？



2、程序示例

程序图

在任务二程序基础上，主程序不变，函数“巡线”修改如下。



3、运行效果

小车的巡线速度可以更快了，不管是转弯还是直行，速度变化都顺滑很多了。

项目小结：📝

项目回顾

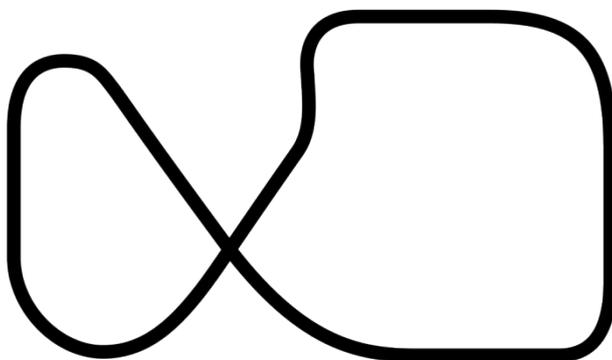
本项目中我们通过由简单到复杂的巡线算法，使巡线的效果越来越快速平稳。

知识点回顾

- 1、学习了巡线的主要实现思路
- 2、学习了HUSKYLENS传感器的巡线功能的操作方法；
- 3、巡线算法优化方法

项目拓展：+

在完成了轨道不交叉巡线后，可能需要处理如下图所示的轨道交叉巡线，可不可以用前面的标签识别让小车在交叉点选择正确的路径呢？



拓展知识：

麦昆巡线时的运动状态或巡线区间可以分为 2 档、3 档、5 档、7 档，那是不是还能继续细分呢？9 档？11 档？.....直到无限细分。调速区间越细分，巡线效果越好。但是这样写程序会越来越长，有什么办法呢？

PID调速算法可以帮助我们解决这个问题，PID全称比例、积分、微分控制，它是一种闭环控制系统，闭环控制是根据控制对象输出的反馈来进行校正的控制方式，它能够根据测量出的实际值与计划值之间的误差Error，按一定的标准来进行纠正。

有兴趣的同学可以在网上搜索相关信息，继续优化我们的巡线效果。