

一、HUSKYLENS（二哈）人工智能摄像头

SKU:SEN0305

1. 产品简介

HuskyLens 二哈识图是一款简单易用的 AI 视觉传感器，内置 8 种功能：人脸识别、人脸检测、物体追踪、物体识别、巡线追踪、颜色识别、标签识别、物体分类。仅需一个按键即可完成 AI 训练，摆脱繁琐的训练和复杂的视觉算法，让你更加专注于项目的构思和实现。

HuskyLens 板载 UART / I2C 接口，可以连接到 Arduino、LattePanda、micro:bit 等主流控制器，实现硬件无缝对接。HuskyLens 直接输出识别结果给控制器，你无需折腾复杂的算法，就能制作非常有创意的项目。



2. 技术规格

处理器：Kendryte K210

图像传感器：OV2640 或 GC0328

供电电压：3.3~5.0V

电流消耗：320mA@3.3V, 230mA@5.0V（电流值为典型值；人脸识别模式；80%背光亮度；补光灯关闭）

连线接口：UART, I2C

显示屏：2.0 寸 IPS，分辨率 320*240

内置功能：人脸识别，物体追踪，物体识别，巡线追踪，颜色识别，标签识别，物体分类

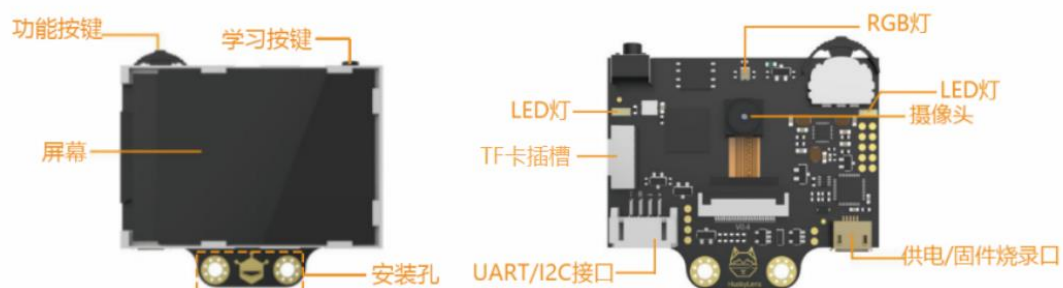
尺寸：52mm*44.5mm

在不同模式下的图像处理的 FPS 会有所不同。

Object Tracking, Face Recognition, Obejct Recognition 中，图像处理 FPS 为 15Hz

Color Recognition, Line Tracking, Tag Recognition 中，图像处理 FPS 为 30Hz

3. 接口与按键说明



3.1 接口

USB 接口：连接到电源，给 HuskyLens 供电；也可以连接到电脑的 USB 接口，给 HuskyLens 升级固件

4pin 接口（UART 模式）

序号	标注	管脚功能	用途
1	T	TX	HuskyLens的串口数据发送管脚
2	R	RX	HuskyLens的串口数据接收管脚
3	-	GND	电源负极
4	+	VCC	电源正极

4pin 接口（I2C 模式）

序号	标注	管脚功能	用途
1	T	SDA	串行数据线
2	R	SCL	串行时钟线
3	-	GND	电源负极
4	+	VCC	电源正极

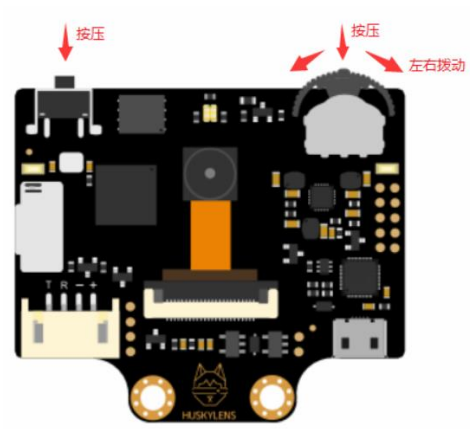
小提示：

- 1、USB 接口与 4pin 接口，任接一个，即可对 HuskyLens 供电。
- 2、由于板载了电源自动切换电路，USB 接口与 4pin 接口可以同时接电源，且优先使用 USB 接口上的电源。
- 3、请确保电源电压与功率足够，防止 HuskyLens 工作异常。

3.2 功能按键与学习按键

功能按键与学习按键的基本操作如下：

- 1、向左或向右拨动“功能按键”，切换到不同的功能
- 2、短按“学习按键”，学习指定的物体；长按“学习按键”，从不同的角度和距离持续学习指定的物体；如果 HuskyLens 之前学习过，则短按“学习按键”，可让 HuskyLens 忘记当前功能下所学的
- 3、长按“功能按键”，进入当前功能的二级菜单参数设置界面，向左、向右拨动或向下短按“功能按键”即可设置相关参数



4. 升级固件

使用本产品之前，强烈建议你升级到最新版本的固件，以获取最新的功能和最稳定的体验。

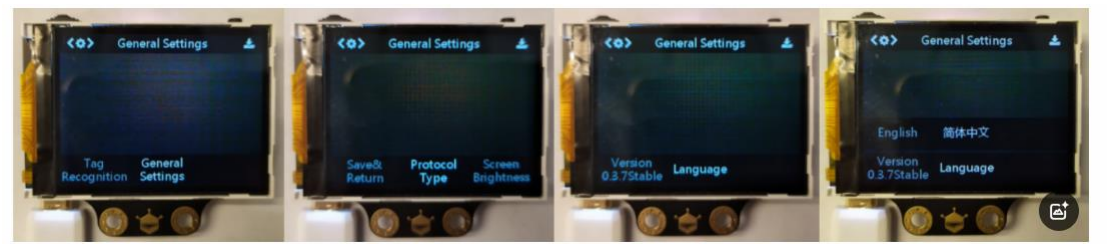
如果你的 HuskyLens 已经是最新固件则无需更新。

5. 常规参数设置

5.1 设置中文语言

菜单界面的默认语言为英文。本教程使用中文语言界面。要将菜单界面的语言设置为中文，请按照如下步骤操作：

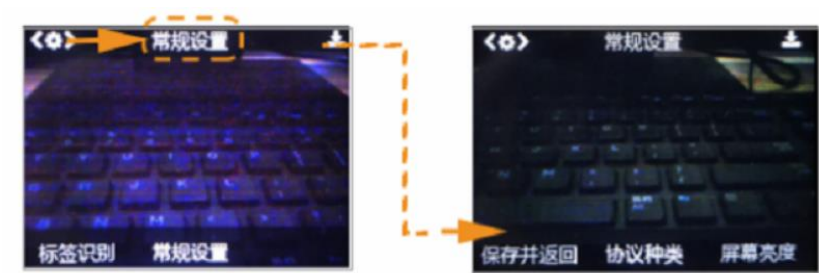
- 1、向右拨动“功能按键”，至屏幕顶端显示“General Settings”。
- 2、向下短按或长按“功能按键”，进入“General Settings”的二级菜单参数设置界面。
- 3、向右拨动“功能按键”，选择“Language”参数（位于最后一个），然后短按“功能按键”，再向右拨动“功能按键”选择“简体中文”，再短按“功能按键”，此时 HuskyLens 会自动切换到中文语言。



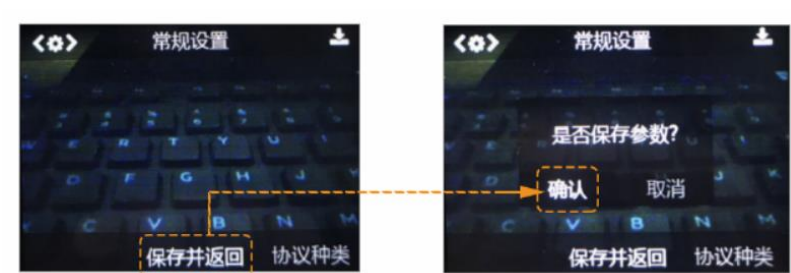
5.2 进入常规设置

进入常规设置的二级菜单并调整各个参数的操作方法如下：

- 1、选择“常规设置”：向右拨动“功能按键”，至屏幕顶端显示“常规设置”。
- 2、进入“常规设置”界面：向下短按或长按“功能按键”，即可进入



- 3、调整各个参数：向左或向右拨动“功能按键”，选择需要设置的参数，然后短按“功能按键”进入所选参数的设置，向左或向右拨动“功能按键”调整所选参数，再短按“功能按键”，确认该参数
- 4、保存并退出：参数调整完毕后，向左拨动“功能按键”，直至选择“保存并返回”，然后短按“功能按键”，屏幕上会显示“是否保存参数”的提示，默认选择“确认”，此时，短按“功能按键”即可保存设置好的参数，并自动退出“常规设置”的二级菜单界面。



小提示：

- 1、常规设置里的各项参数，都是通过左右拨动“功能按键”和按下“功能按键”来进行选择和设置的。
- 2、设置好参数之后，务必选择“保存并返回”，才能将设置好的参数保存下来

5.3 常规设置的参数项

常规设置的二级菜单参数设置界面，包含了 10 个不同的参数。

- 协议种类

HuskyLens 支持三种不同波特率的串口协议（9600,115200,1000000），也支持 I2C 协议。此外，还支持自动识别协议，也就是，HuskyLens 会自动切换成串口或 I2C 协议。建议使用自动识别协议，简单方便。默认设置为自动识别。

- 屏幕亮度

屏幕可以调节亮度，范围为 1~100。默认值为 80。

- 菜单自动隐藏

如果有一段时间不操作 HuskyLens，那么屏幕上的菜单会自动隐藏。你可以设置这个等待时间，范围为 1~100 秒。默认值为 10 秒。

- LED 开关

HuskyLens 板载了 2 个 LED 补光灯。你可以打开或者关闭补光灯。默认值为关。

- LED 亮度

LED 补光灯可以调节亮度，范围为 1~100。默认值为 50。

- RGB 开关

HuskyLens 板载了一颗 RGB 指示灯，可以打开或者关闭 RGB 指示灯。默认值为开。

- RGB 亮度

RGB 指示灯也可以调节亮度，范围为 1~100。默认值为 20。

- 初始化设置

通过初始化设置，把 Huskylens 恢复到出厂状态

- 版本号

当前固件的版本信息

- 语言

目前 HuskyLens 支持中文和英文两种语言。通过语言设置，选择你需要的语言。

6. 颜色与坐标

6.1 颜色指示

各个功能中，屏幕中央的方框、“+”字的颜色定义是统一的，有助于你了解当前 HuskyLens 所处的状态。

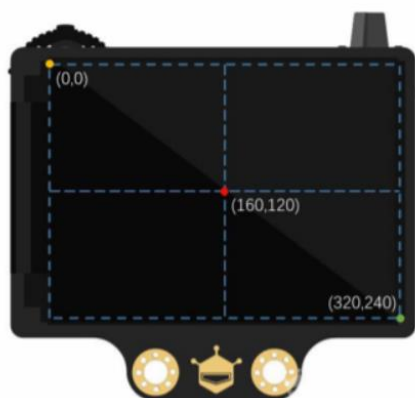
颜色	状态
由橙色渐变到黄色，再由黄色渐变到橙色	未学习过，待学习
黄色	正在学习中
蓝色	已学习过，且识别到学习过的东西

RGB 指示灯可用于人脸识别功能的状态指示，颜色定义如下：

颜色	状态
蓝色	未学习过，侦测到人脸
黄色	正在学习中
绿色	已学习过，且识别到学习过的人脸

6.2 坐标系

当 HuskyLens 识别学过的物体时，在屏幕上会有彩色方框将其框选，你可以读取这些方框的坐标、长宽等数据。HuskyLens 的坐标系如下所示，这有助于你了解方框的坐标值、长宽等数据。



7. 各功能说明

7.1 人脸识别

本功能可以侦测任何脸部轮廓；识别、追踪学习过的人脸。

识别单个人脸

默认设置为学习并识别单个人脸。

操作设置

向左拨动“功能按键”，直至屏幕顶部显示“人脸识别”。

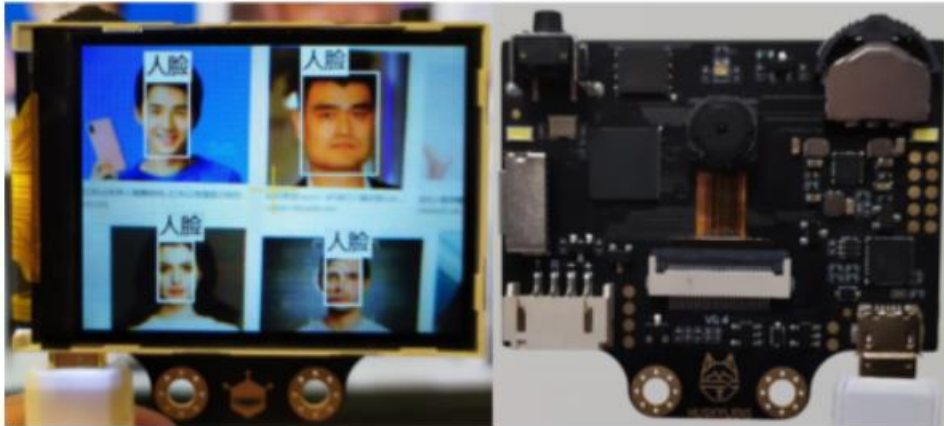


学习与识别

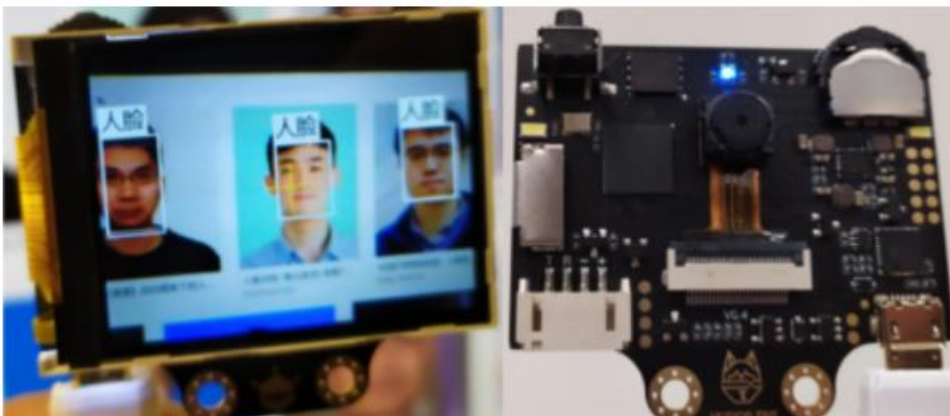
1、侦测人脸

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。

若此时屏幕中央的“+”字没有对准任何人脸框，则另一面的 RGB 灯不亮。



若此时屏幕中央的“+”字对准在任意人脸框内，则另一面的 RGB 灯亮蓝色。



小提示：

如需让 HuskyLens 学习或识别自己的脸，也就是自拍，此时看不到屏幕，那么可以根据 RGB 指示灯的不同颜色来确定状态。

2、学习人脸

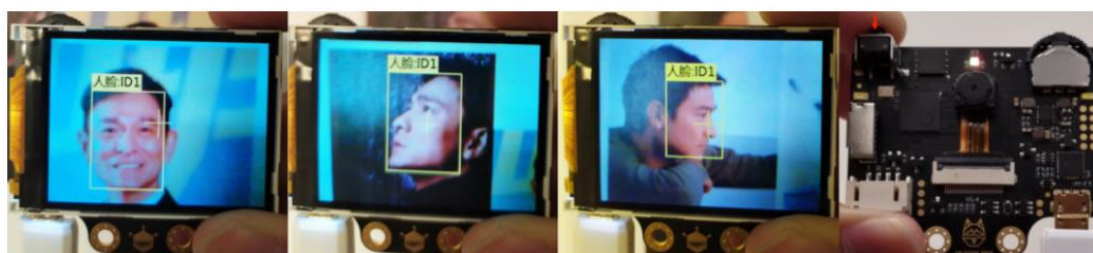
将 HuskyLens 屏幕中央的“+”字对准需要学习的人脸，短按“学习按键”完成学习。如果识别到相同的脸，则屏幕上会出现一个蓝色的框并显示“人脸：ID1”。这说明已经可以进行人脸识别了。



然而，上述操作只是让 HuskyLens 学习了人脸的一个角度(维度)，但实际上人脸是立体的，因此人脸有多个角度。如果人脸角度变化了，比如正脸换成了侧脸，那 HuskyLens 不一定能识别出来。为解决这个问题，HuskyLens 内置有持续学习的功能，能录入人脸的各个角度，让 HuskyLens 越学越准确。

录入各个角度人脸的操作方法如下：(注：开始学习新的人脸之前，先要让 HuskyLens 忘记已经学习过的人脸。操作方法参考“忘记学过的人脸”章节。)

将 HuskyLens 屏幕中央的“+”字对准需要学习的人脸，长按“学习按键”不松开，此时屏幕中会在人脸上显示黄色框并标识“人脸:ID1”，说明 HuskyLens 正在学习人脸。然后，将 HuskyLens 屏幕中央的黄色框依次对准同一个人的脸的不同角度，如正脸、侧脸（也可以是同一个人的多张照片），录入此人脸的各个角度。学习过程中，RGB 灯为黄色。



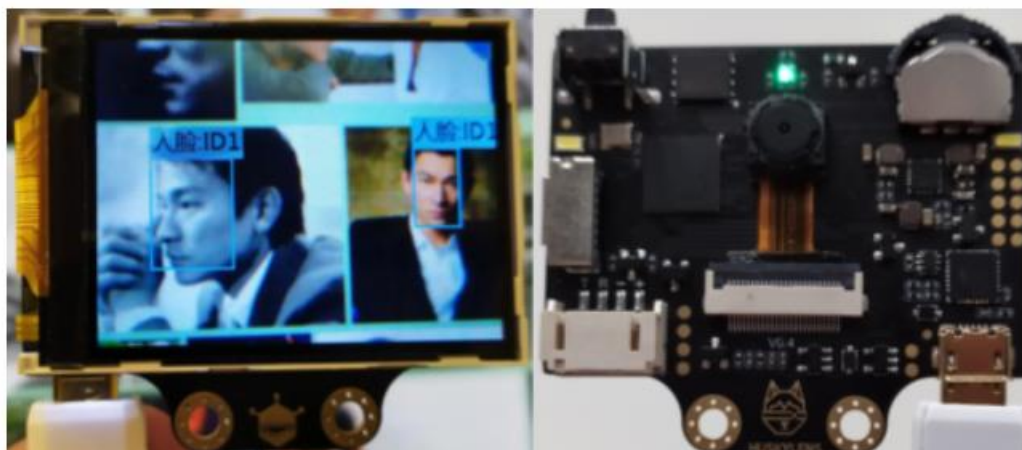
接着，松开“学习按键”，结束学习。如果 HuskyLens 识别到学习过的人脸，则屏幕上，这个人脸会被蓝色的框选中，并显示“人脸：ID1”。

小提示：

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）。如要让 HuskyLens 学习新的人脸，则需要先让 HuskyLens 忘记已学的人脸。

3、识别人脸

HuskyLens 学习过的人脸信息会自动保存起来。后续，当 HuskyLens 检测到学习过的人脸时，会将该人脸用蓝色方框框选出来并标识“人脸:ID1”，边框大小会随着人脸大小而变化，并自动追踪人脸。此时 RGB 指示灯为绿色。

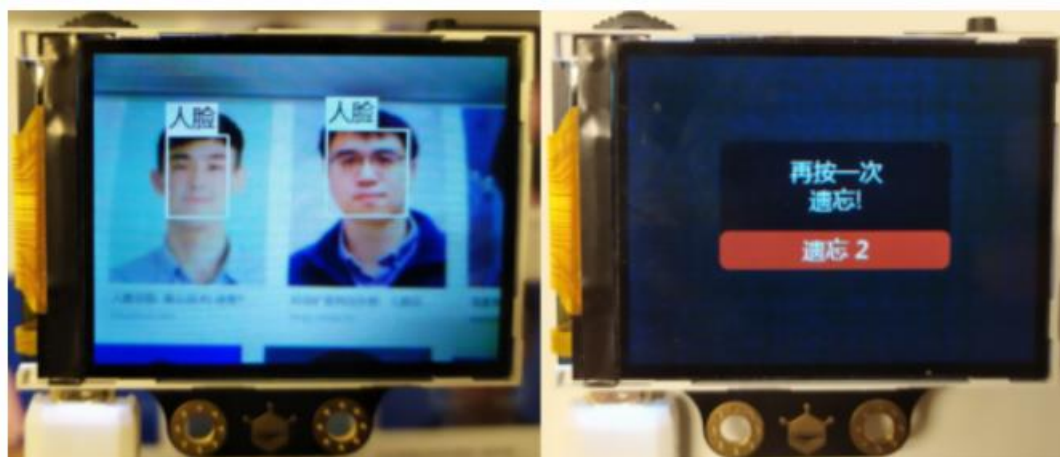


4、忘记学过的人脸

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）。如要让 HuskyLens 学习新的人脸，则需要删除之前学习过的人脸信息，也就是让 HuskyLens 忘记已学的人脸。

删除已学东西的操作方法如下：

在当前功能下，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，屏幕中央显示“+”字，说明 HuskyLens 已经准备好学习新东西了。如果不小心短按了“学习按键”，屏幕已经提示“再按一次遗忘！”，但又不想删除已学习的东西，那么在倒计时结束前，不要有任何操作即可。



注：让 HuskyLens 忘记所学东西的操作方法，在其他功能下，也是完全一样的，后续不再复述。

识别多个人脸

默认设置为学习并识别单个人脸。如要学习并识别多个人脸，则需要人脸识别功能的二级菜单参数设置中打开“学习多个”选项。

操作设置

- 1、向左拨动“功能按键”，至屏幕顶部显示“人脸识别”。
- 2、长按“功能按键”，进入人脸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，并自动返回到人脸识别模式。



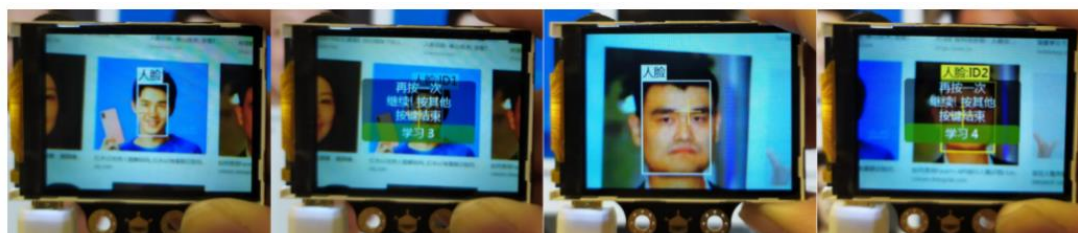
学习与识别

1、学习多个人脸：

将 HuskyLens 屏幕中央的“+”字对准需要学习的人脸，长按“学习按键”完成第一个人脸的学习（各个角度）。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人脸，则在倒计时结束前短按“学习按键”，可以继续学习下一个人脸。如果不再需要学习其他人脸了，则在倒计时结束前短按“功能按键”即可，或者不操作任何按键，等待倒计时结束。

本章节中，我们需要继续学习下一个人脸，因此在倒计时结束前短按“学习按键”。然后将 HuskyLens 屏幕中央的“+”字对准需要学习的下一个人脸，长按“学习按键”完成第二个人的脸的学习。以此类推。

HuskyLens 标注的人脸 ID 与录入人脸的先后顺序是一致的，也就是：学习过的人脸会按顺序依次标注为“人脸：ID1”，“人脸：ID2”，“人脸：ID3”，以此类推，并且不同的人脸 ID 对应的边框颜色也不同。

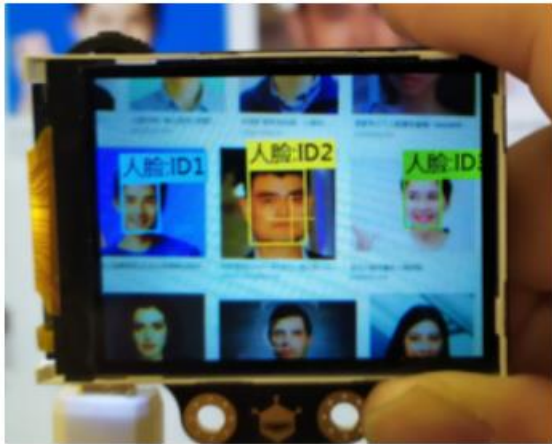


小提示：如果屏幕中央没有“+”字，说明 HuskyLens 在该功能下已经学习过了（已学习状态）。

如要让 HuskyLens 学习新人脸，则需要先让 HuskyLens 忘记已学的。操作方法参考“忘记学过的人脸”章节。

2、识别多个人脸：

HuskyLens 学习过的人脸信息会自动保存起来。后续，当 HuskyLens 检测到学习过的人脸时，会将这些人脸用方框框选出来并标识 ID，第一个学习的人脸标注为“人脸：ID1”，第二个学习的人脸标注为“人脸：ID2”，第三个学习的人脸标注为“人脸：ID3”，以此类推。不同的人脸 ID 对应的边框颜色也不同，边框大小会随着人脸大小而变化，并自动追踪人脸。

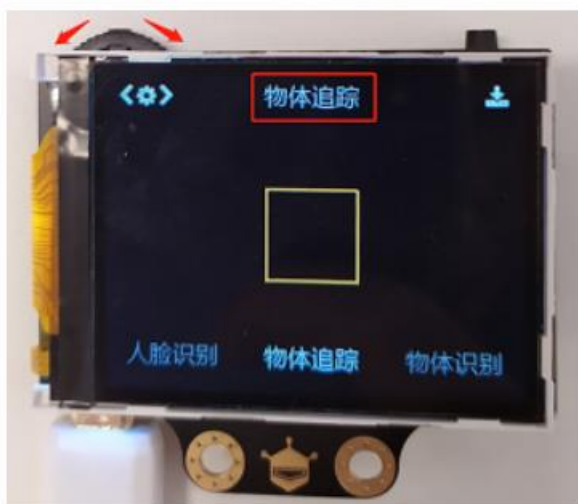


7.2 物体追踪

本功能可以学习并追踪一个指定的物体。只能追踪一个物体，暂不支持追踪多个物体。

操作设置

1、向左或向右拨动“功能按键”，至屏幕顶部显示“物体追踪”。



2、长按“功能按键”，进入物体追踪功能的二级菜单参数设置界面。

3、向右拨动“功能按键”，选中“学习开启”，然后短按“功能按键”，接着向右拨动“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。再短按

“功能按键”，确认该参数。

4、同样的方法，打开“自动保存模型”的开关。



5、也可以设置“学习框长宽比”和“学习框大小”来调整学习框的尺寸，以便于更好的匹配所学物体的尺寸。

6、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，并自动返回到物体追踪模式。

学习与追踪

1、学习物体：

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。如不方便，包含特征鲜明的局部亦可。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”。

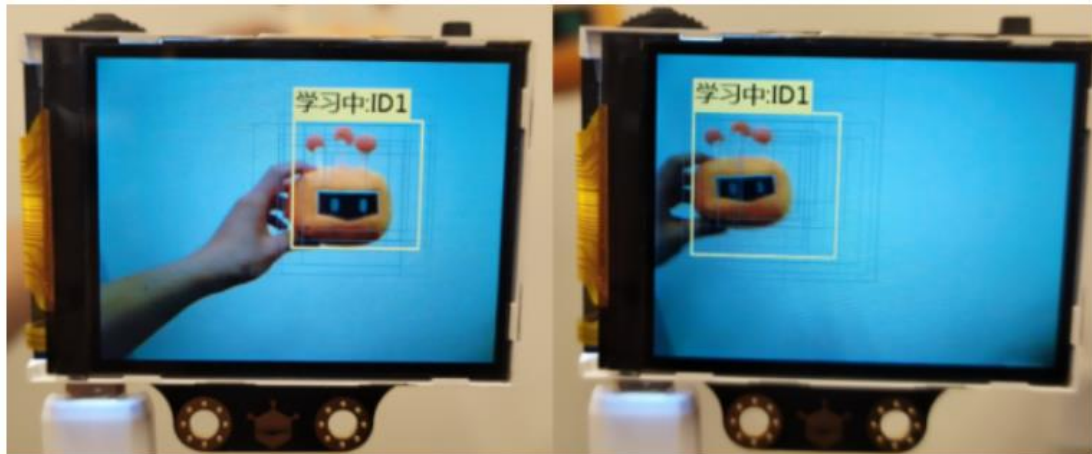


当 HuskyLens 在不同的角度和距离都能追踪到该物体时，就可以松开“学习按键”结束学习了。

小提示：如果屏幕中央没有橙黄色方框，说明 HuskyLens 已经学习过一个物体了。如需让 HuskyLens 学习新物体，先需要让 HuskyLens 忘记已学的，操作方法参照“忘记学过的人脸”章节。

2、追踪物体：

移动 HuskyLens 或者目标物体，屏幕中的框会自动追踪目标物体。追踪物体时，会显示“学习中：ID1”，表示 HuskyLens 一边追踪物体，一边学习，这样设置有助于提高物体追踪的能力。



当识别结果满足要求，达到预期效果，就可关闭一边追踪一边学习的功能，方法是：长按“功能按键”，进入物体追踪功能的二级菜单参数设置，选择“学习开启”，关闭此参数即可（进度条颜色变白，进度条上的方块位于进度条的左边）。

小提示：

追踪物体时，每次只能追踪一个物体，可以是任何有明显轮廓的物体，甚至是各种手势（或脸部，如上下左右）。

7.3 物体识别

本功能可识别这是什么物体，并追踪。目前仅支持 20 种物体，分别为：飞机、自行车、鸟、船、瓶子、巴士、汽车、**猫**、椅子、牛、餐桌、**狗**、马、摩托车、人、盆栽植物、羊、沙发、火车、电视。

20 种物体的英文名称，分别为：aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor.

注：物体识别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，人是一类，但可以区分不同的人脸。

默认设置为只标记并识别一个物体。本章节采用标记并识别多个物体为例进行演示。

操作设置

- 1、向左或向右拨动“功能按键”，直至屏幕顶部显示“物体识别”。
- 2、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。

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，并自动返回到物体识别模式。

标记与识别

1、侦测物体：

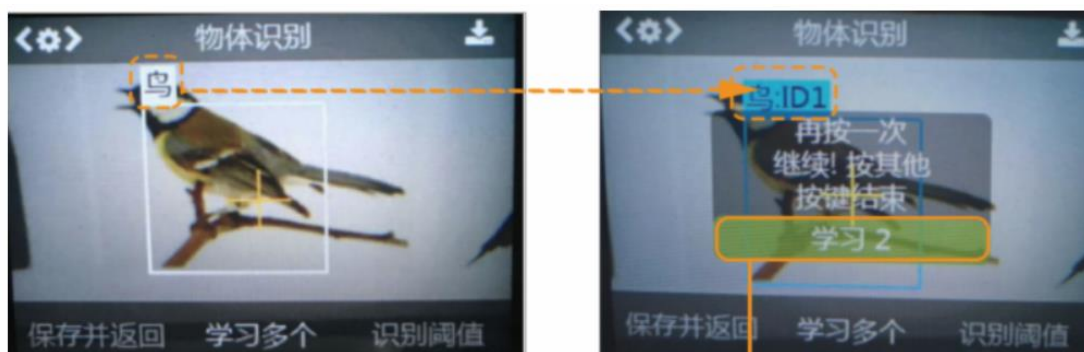
把 HuskyLens 对准目标物体，在屏幕上，会有白色框自动框选出识别到的所有物体，并显示对应的物体名称。目前只能识别并框选 20 种物体，其余物体无法识别和框选。



2、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。



3、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方框框选出这些物体，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，并做追踪。

7.4 巡线追踪

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。本章节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。

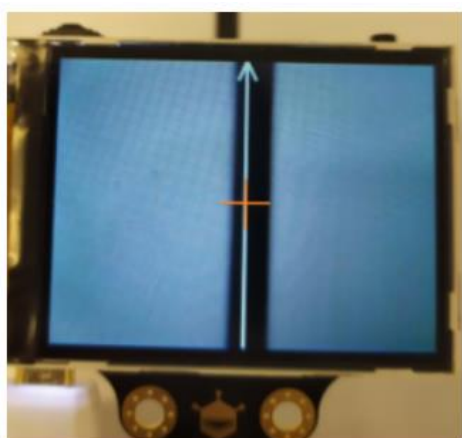


5、向左拨动“功能按键”，选中“保存并返回”，短按“功能按键”，屏幕提示“是否保存参数？”，默认选择“确认”，此时短按“功能按键”，即可保存参数，并自动返回到巡线模式。

学习与追踪

1、学习线条：

将 HuskyLens 屏幕上的“+”字对准目标线条。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。



2、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。



小提示：

- 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。

7.5 颜色识别

本功能可以学习、识别、追踪指定的颜色。

默认设置为学习、识别并追踪一种颜色。本章节采用学习、识别并追踪多种颜色为例进行说明。

小提示：

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。

操作设置

- 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。

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方框颜色也不同。



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。



V0.5.1 版本以下的固件中，当出现多个相同颜色的色块时，相隔的色块不能被同时识别，只能一次识别一个色块。如下图所示：



V0.5.1 版本及其以上的固件中，优化了这个功能，当出现多个相同颜色的色块时，可以同时识别出这些色块，这个功能，可以用来对色块进行计数。

色块数量识别

V0.5.1 及其以上版本的固件中，HUSKYLENS 在颜色识别功能中可以进行色块计数，即：判断 HUSKYLENS 屏幕中色块的数量。下面以识别多个相近颜色的气球为例进行说明。

1、学习并识别颜色

将 HuskyLens 屏幕中央的“+”字对准需要学习的气球，按下“学习按键”学习该气球的颜色，然后松开“学习按键”结束学习，就可以看到已经能识别该气球的颜色了，但对于颜色相近的气球，可能还不一定能识别的出。



2、调节识别框阈值

在识别相近颜色的色块时，通过设置阈值可以调节识别的精度，比如上图中，有的颜色相近的黄色气球没被识别到，可能就是阈值设置的较高。颜色识别功能的二级菜单参数设置界面中，有“识别框阈值”参数，该参数的数值越低，精度越低，但识别的相近色块就越多。如下图，阈值为 20 时，只能识别到一个黄色球，阈值为 0 时，3 个黄色球都可以识别到。请根据实际效果，调节识别框阈值，使得识别精度达到可接受的范围内。



通过此功能，可以方便的实时获得屏幕中黄色气球的数量了。

7.6 标签识别

本功能可以侦测 AprilTag 标签；学习、识别、追踪指定的 AprilTag 标签。 点击[此处](#)下载更多标签图片。

Tips: 本功能只能识别 AprilTag 标签！

默认设置为只学习、识别并追踪一个标签。本章节采用学习、识别并追踪多个标签为例进行说明。

操作设置

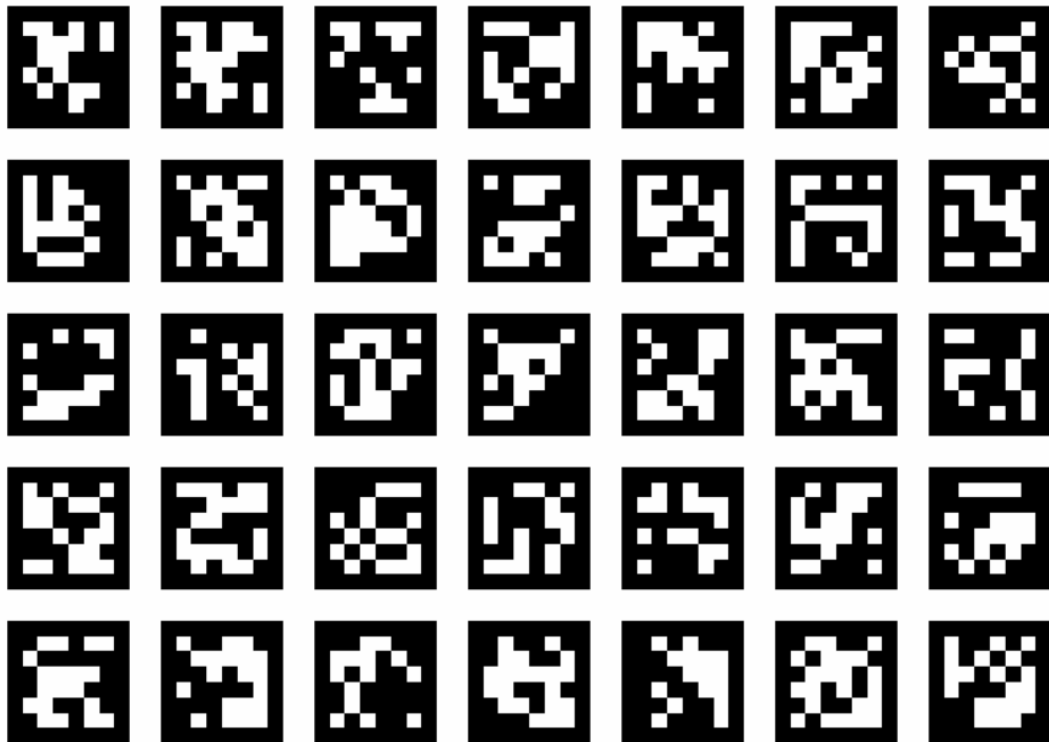
- 1、向左或向右拨动“功能按键”，直至屏幕顶部显示“标签识别”。
- 2、长按“功能按键”，进入标签识别功能的二级菜单参数设置界面。
- 3、向左或向右拨动“功能按键”，选中“学习多个”，然后短按“功能按键”，接着向右拨动“功能按键”打开“学习多个”的开关，即：进度条颜色变蓝，进度条上的方块位于进度条的右边。再短按“功能按键”，确认该参数。



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，并自动返回到标签识别模式。

学习与识别

可用下面的标签来测试。



1、侦测标签

当 HuskyLens 检测到标签时，屏幕上会用白色框自动框选出检测到的所有标签。



2、学习标签

将 HuskyLens 屏幕中央的“+”字对准需要学习的标签，短按或长按“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。



本章节中，需要继续学习下一个标签，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，短按或长按“学习按键”完成第二个标签的学习。以此类推。

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。边框的大小会随着标签的大小进行变化，边框自动追踪这些标签。



7.7 物体分类

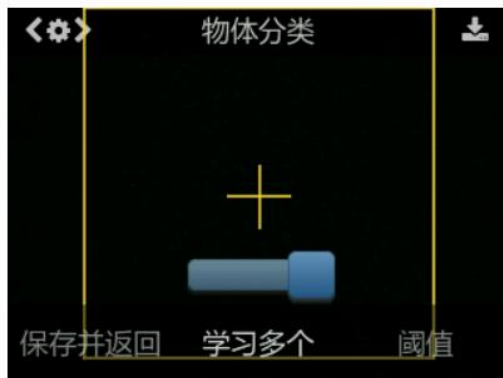
本功能可以学习不同物体的多张照片，然后使用内置的机器学习算法进行训练，训练完成之后，当 HuskyLens 的摄像头画面再次出现学习过的物体时，HuskyLens 就可以识别出它们，并显示其 ID 号。同一个物体的照片，学的越多，识别越精准。

默认设置为学习多个，本章节以识别工人是否佩戴安全帽为例进行说明。

操作设置

- 1、向左或向右拨动“功能按键”，直至屏幕顶部显示“物体分类”。
- 2、长按“功能按键”，进入物体分类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”，查看该开关是否打开。如没打开，则短按“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，并自动返回到物体分类模式。

学习与识别

可用下面的图片来测试。



1、学习物体

将 HuskyLens 屏幕中央的橙黄色大框对准需要学习的物体(上图中左边带安全帽的工人)，长按“学习按键”，从不同的角度和距离学习，完成第一个物体的学习之后，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物体，则在倒计时结束前按下“学习按键”，可以继续学习下一个物体。如果不再需要学习其他物体了，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。

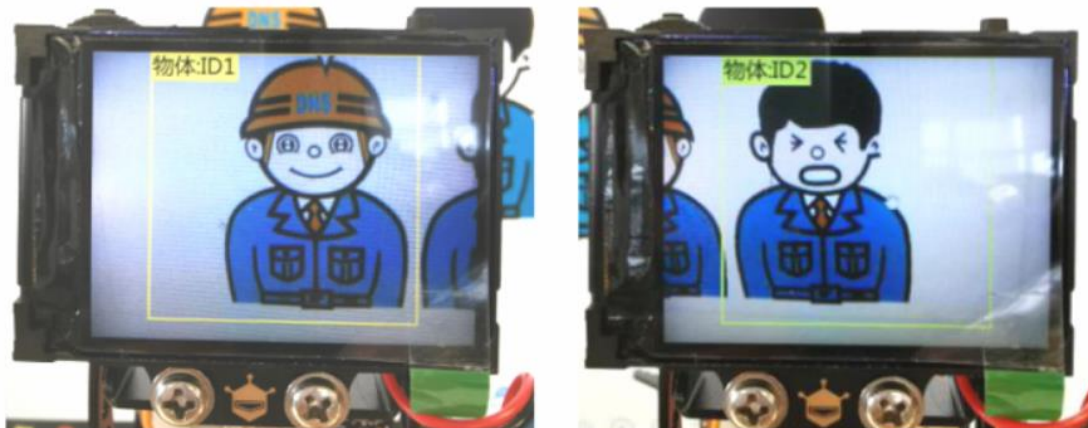


本章节中，需要继续学习下一个物体（上上图中右边不带安全帽的工人），因此在倒计时结束前按下“学习按键”，然后将 HuskyLens 屏幕中央的橙黄色大框对准需要学习的物体，长按“学习按键”完成第二个物体的学习。以此类推。

物体 ID 与录入物体的先后顺序是一致的，也就是：学习过的物体会按顺序依次标注为“物体：ID1”，“物体：ID2”，“物体：ID3”，以此类推，并且不同的物体对应的边框颜色也不同。

2、识别物体

HuskyLens 再次遇到学习过的物体时，在屏幕上会显示其 ID 号。如下图，当 HUSKYLENS 识别到工人带着安全帽，屏幕显示 ID1，没带安全帽显示 ID2。



更多关于物体分类的想法和项目：

<https://mc.dfrobot.com.cn/thread-304439-1-1.html>

常见问题

物体分类算法可以得出物体在 HUSKYLENS 屏幕上的相对位置吗？

答：不能。物体分类算法下，输出方框的位置固定，其在屏幕上的 x、y 中心坐标保持不变，所以 HUSKYLENS 无法得出物体在屏幕上的相对位置。但是可以通过学习物体的不同位置作为不同的 ID，通过 ID 来判断位置，比如无人车中，分别学习道路左侧、中间、右侧为 ID1、2、3，通过判断 ID 就可以知道无人车相对于马路的位置。

物体分类算法下，怎么提高识别的精准度？

答：长按“学习按键”不松开，可以从多个角度和距离录入目标图片，提高识别精准度。

7.8 辅助功能

V0.5.1 及以上版本的固件中，新增了很多辅助功能，能够更好的帮助你使用 HuskyLens 搭建项目。

7.8.1 自定义 ID 名称

你可以更改 ID 对应的名称，比如：把“人脸:ID1”改为“Jack:ID1”，把“颜色:ID1”改为“Red:ID1”。此功能非常适合给人、物体取名字，让识别结果更容易辨识。除了巡线追踪功能外，其余 6 大算法功能均支持自定义 ID 名称。自定义 ID 名称只支持英文名。

演示：更改人脸识别算法 ID1 的名称为 Jack。

mind+ 上传模式示例程序：



makecode 示例程序：



Arduino 示例程序：

```
#include "HUSKYLENS.h"
```

```
HUSKYLENS huskylens;
```

```
void setup()
```

```
{
```

```
    Serial.begin(115200);
```

```
    Wire.begin();
```

```
    while (!huskylens.begin(Wire))
```

```
    {
```

```
        Serial.println(F("Begin failed!"));
```

```
        delay(100);
```

```
    }
```

```
    while (!huskylens.setCustomName("Jack",1)) // bool setCustomName(String  
name,uint8_t id)
```

```
    {
```

```
        Serial.println(F("ID1 customname failed!"));
```

```
        delay(100);
```

```
    }
```

```
}
```

```
void loop()
```

```
{
```

```
}
```

Copy

Mind+python 图形化模式示例程序：



Mind+python 代码模式示例程序：

```
# -*- coding: UTF-8 -*-

# MindPlus
# Python
from pinpong.libs.dfrobot_huskylens import Huskylens
from pinpong.board import Board

Board().begin()
p_huskylens = Huskylens()
p_huskylens.command_request_algorithm("ALGORITHM_FACE_RECOGNITION")
p_huskylens.command_request_customnames(1, "Jack")

while True:
    pass
```

Copy

运行结果：

HuskyLens 学习完第一张人脸 ID1 后，会显示“Jack:ID1”，如下图所示。



7.8.2 屏幕显示自定义文字

HuskyLens 板载了一块屏幕，你可以在屏幕任意位置叠加显示字符，支持显示英文字母、数字及符号，暂不支持显示中文。你可以将识别结果、传感器数据等直接显示在 HuskyLens 的屏幕上。7 大算法功能均支持自定义显示文字。

演示：在屏幕右上角坐标(230,35)处开始显示文字 "Mind+"

mind+上传模式示例程序：



makecode 示例程序：



Arduino 示例程序：

```
#include "HUSKYLENS.h"
```

```
HUSKYLENS huskylens;
```

```
void setup()
```

```
{
```

```
    Serial.begin(115200);
```

```
    Wire.begin();
```

```
    while (!huskylens.begin(Wire))
```

```
    {
```

```
        Serial.println(F("Begin failed!"));
```

```
        delay(100);
```

```
    }
```

```
    while (!huskylens.customText("mind+",230,35)) // bool customText(String text,uint16_t  
x,uint8_t y)
```

```
    {
```

```
        Serial.println(F("custom text failed!"));
```

```
        delay(100);
```

```
    }
```

```
}
```

```
void loop()
```

```
{
```

```
}
```

```
Copy
```

Mind+python 图形化模式示例程序：



Mind+python 代码模式示例程序：

```
# -*- coding: UTF-8 -*-

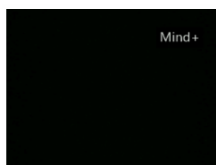
# MindPlus
# Python
from pinpong.libs.dfrobot_huskyLens import HuskyLens
from pinpong.board import Board

Board().begin()
p_huskyLens = HuskyLens()
p_huskyLens.command_request_custom_text("Mind+", 230, 35)

while True:
    pass
```

Copy

运行结果：



7.8.3 拍照、截屏保存到 SD 卡

HuskyLens 可以像一个卡片相机一样，拍照或者截屏摄像头画面，保存到 SD 卡中，HuskyLens 板载一个 SD 卡插槽，将 SD 卡插入即可使用。截屏保存的图片包含屏幕上显示的文字、方框等内容，而拍照出来的图片只包含摄像头画面，没有方框、文字信息。

HuskyLens 的 7 大算法功能均支持该功能。使用此功能时，需要先插入一张 SD 卡。

小提示：

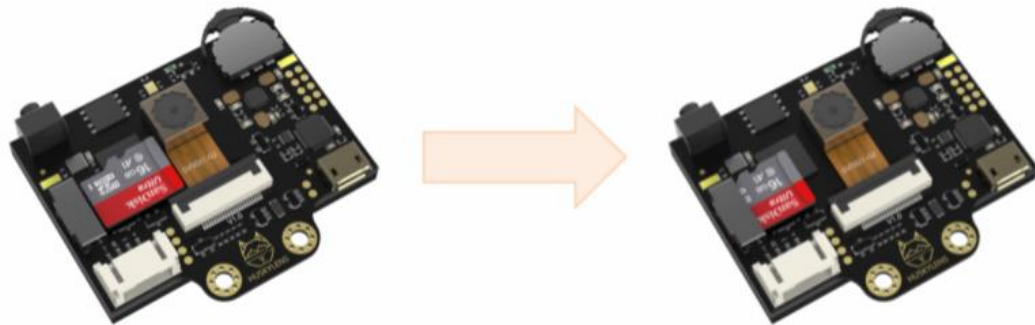
1、推荐一线品牌的 SD 卡，如闪迪、东芝、三星、金士顿等，个别小品牌的 SD 卡，可能存

在兼容性问题。

2、HUSKYLENS 保存的图片，分辨率为 320*240，无法更改。

3、请将 SD 卡格式化为 FAT32。因拍照/截图并保存需要一定的时间，因此建议调用该功能的间隔至少 0.5 秒以上。

4、SD 卡插槽向内，需要从里向外插入，插入 SD 卡的过程如下图所示：



演示：按下 micro:bit 的 A 键或拉低 Arduino 的 A0 管脚，HuskyLens 截屏并保存到 SD 卡。
mind+上传模式示例程序：



makecode 示例程序：



Arduino 示例程序:

```
#include "HUSKYLENS.h"
```

```
HUSKYLENS huskylens;
```

```
void setup()
```

```
{  
    Serial.begin(115200);  
    pinMode(A0,INPUT_PULLUP);  
    Wire.begin();  
    while (!huskylens.begin(Wire))  
    {  
        Serial.println(F("Begin failed!"));  
        delay(100);  
    }  
}
```

```
void loop()
```

```
{  
    if(digitalRead(A0) == 0)  
    {  
        while (!huskylens.saveScreenshotToSDCard()) // bool saveScreenshotToSDCard() or  
bool savePictureToSDCard()  
        {  
            Serial.println(F("save screenshot to SD card failed!"));  
            delay(100);  
        }  
        Serial.println(F("saving screenshot to SD card..."));  
        delay(500); // The interval between calling this function is not less than 0.5 seconds.  
    }  
}
```

Copy

Mind+python 图形化模式示例程序:



Mind+python 代码模式示例程序：

```
# -*- coding: UTF-8 -*-

# MindPlus
# Python
from pinpong.libs.dfrobot_huskyLens import HuskyLens
from pinpong.extension.unihiker import *
from pinpong.board import Board,Pin
from pinpong.board import Board
import time
```

```
Board().begin()
p_huskyLens = HuskyLens()
```

```
while True:
    if (button_a.is_pressed()==True):
        p_huskyLens.command_request_photo()
        time.sleep(0.5)
```

Copy

运行结果：

按下 micro:bit 的 A 键或拉低 Arduino 的 A0 管脚，HuskyLens 截图保存到 SD 卡。将 SD 卡拔出，再插入到电脑上，就可以看到 HuskyLens 截取的图片了。

	<table><thead><tr><th>名称</th><th>大小</th></tr></thead><tbody><tr><td> 1.bmp</td><td>226 KB</td></tr><tr><td> 2.bmp</td><td>226 KB</td></tr></tbody></table>	名称	大小	 1.bmp	226 KB	 2.bmp	226 KB	
名称	大小							
 1.bmp	226 KB							
 2.bmp	226 KB							
HuskyLens屏幕上显示的画面	电脑上读取SD卡里面的图片文件(.bmp文件)	打开图片文件						

7.8.4 SD 卡保存/加载模型

HuskyLens 支持将同一个算法下学习的多个物体作为一个数据模型，通过 SD 卡能保存这个模型，即使删除了当前算法的所有数据，也可以再从 SD 卡中重新加载模型，让 HuskyLens 自动学习模型中的数据。通过保存模型的方法，HuskyLens 可以在一个算法中满足多场景的使用，比如在物体分类算法中，学习剪刀石头布为 1 个模型，学习垃圾分类为 1 个模型，然后通过模型切换，就能快速实现所需功能，避免重复的让 HuskyLens 学习。

使用此功能时，需要先插入一张 SD 卡，SD 卡需要格式化为 FAT32。HUSKYLENS 的 7 大算法功能都支持 SD 卡保存和加载模型，每个算法功能可以保存 5 个模型。

方式一：在各算法功能的二级菜单目录下手动操作

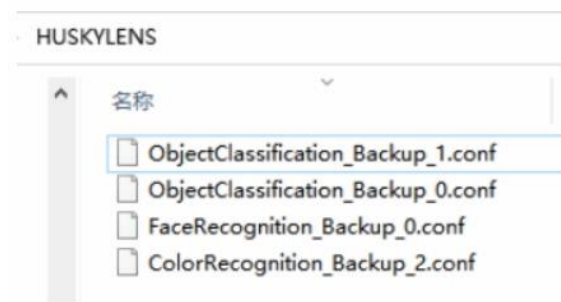
此处以物体分类功能中为例进行演示，其他算法功能中的用法也是一样的。

****学习新物体：****让 HuskyLens 学习完几个新的物体。比如：在物体分类功能中，依次学习多个动物，有不同的 ID 号。

****导出模型：****长按“功能按键”，进入物体分类功能的二级菜单参数设置界面，然后向右拨动“功能按键”，直至选择“导出到 SD 卡”一项（即保存模型到 SD 卡），短按“功能按键”，左右拨动“功能按键”，选择 0~4 任意一个即可（相当于选择保存的位置，方便区分不同的模型），然后短按“功能按键”即可保存模型。



模型导出到 SD 卡后，用电脑读取该 SD 卡，在“HUSKYLENS”文件夹下即可看到后缀为.conf 的模型数据文件，不同算法保存的文件名称不同。如下图所示：



小提示：根据模型数据文件名可以知道导出的模型数据文件属于哪个算法功能；无法查看每个模型中学习的图片。

****导入模型：**当 SD 卡中已经有模型数据文件时，可以把模型直接导入给 HuskyLens。比如：在物体分类功能中，SD 卡中已经有物体分类的模型数据文件了，那么在物体分类功能的二级菜单参数设置界面中，选择“从 SD 卡导入”一项，选择 0~4 其中一个，即可把 SD 卡的模型加载到 HuskyLens 中。

导入模型后，HuskyLens 就可以使用当前模型进行识别了。



小提示：导入模型时，只支持导入相同算法的模型，不能导入不同算法的模型。

方式二：程序触发保存/加载模型

该方式，可以使用 micro:bit、Arduino 等主板，自动触发保存或加载模型文件，使用以下代码块，即可实现。

Mind+ 积木块：



MakeCode 积木块：



Arduino 函数:

```
bool saveModelToSDCard(int fileNum);
```

```
bool loadModelFromSDCard(int fileNum);
```

Mind+python 积木块:



Mind+python 函数:

```
p_huskylens.command_request_save_model_to_SD_card(0)
```

```
p_huskylens.command_request_load_model_from_SD_card(0)
```

7.8.5 程序触发学习功能

除了手动按下“学习按键”学习目标物体外，还能通过程序触发，让 HuskyLens 自动学习。有了这个功能，可以让 micro:bit、Arduino 等主控板自动控制 HuskyLens 的学习。

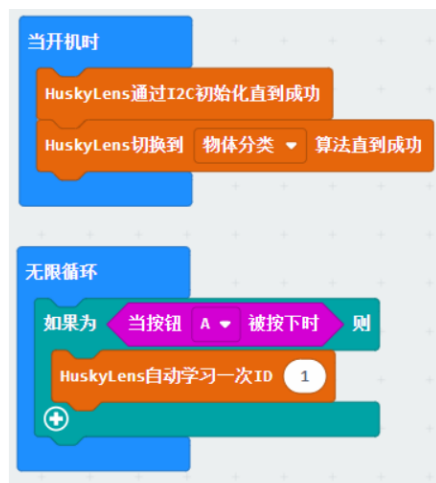
HuskyLens 的 7 大算法功能均支持程序触发学习。

演示：按下 micro:bit 主板的 A 键或拉低 Arduino 的 A0 管脚，让 HuskyLens 自动学习一次目标物体，该物体的 ID 号为 1。

mind+上传模式示例程序:



makecode 示例程序:



Arduino 示例程序:

```
#include "HUSKYLENS.h"
```

```
HUSKYLENS huskylens;
```

```
void setup()
```

```
{
    Serial.begin(115200);
    pinMode(A0,INPUT_PULLUP);
    Wire.begin();
    while (!huskylens.begin(Wire))
    {
        Serial.println(F("Begin failed!"));
        delay(100);
    }
}
```

```
void loop()
```

湖北 荆门 彭先清 技术支持

```

{
    if(digitalRead(A0) == 0)
    {
        while (!huskylens.writeLearn(1)) // bool writeLearn(int ID)
        {
            Serial.println(F("learn object ID1 failed!"));
            delay(100);
        }
        Serial.println(F("learn object ID1 success"));
    }
}

```

Copy

Mind+python 图形化模式示例程序：



Mind+python 代码模式示例程序：

```
# -*- coding: UTF-8 -*-
```

```
# MindPlus
```

```
# Python
```

```
from pinpong.libs.dfrobot_huskylens import Huskylens
```

```
from pinpong.extension.unihiker import *
```

```
from pinpong.board import Board,Pin
```

```
from pinpong.board import Board
```

```
Board().begin()
```

```
p_huskylens = Huskylens()
```

```
p_huskylens.command_request_algorithim("ALGORITHM_OBJECT_CLASSIFICATION")
```

```
while True:
```

```
    if (button_a.is_pressed()==True):
```

p_huskylens.command_request_learn_once(1)

Copy
运行结果：

在物体分类模式下，将 HuskyLens 依次对准下面 3 张图片，按下 micro:bit 的 A 键或拉低 Arduino 的 A0 管脚，HuskyLens 会依次学习 3 张图片作为物体：ID1。当 HuskyLens 再次识别到 3 张图片的任意一张时，会在屏幕上显示物体：ID1，表示这 3 名工人，都带了安全帽。



此时，可以对 ID1 自定义名称，比如改成 safe:ID1；也可以用屏幕自定义显示文字功能，叠加相关信息，说明此状态是带了安全帽的。有了这些信息说明，识别结果简单易懂。

8. micro:bit 教程（略）

9. Arduino 教程（略）

10. 行空板教程

本章使用 Mind+，分别用几个案例演示具体用法。

10.1 行空板案例：人脸识别

本案例演示如何把 HuskyLens 连接到行空板，然后行空板从 HuskyLens 读取人脸识别结果。如果 HuskyLens 侦测到任意人脸，则行空板的屏幕显示微笑，否则显示紧张。

HuskyLens 与行空板的通信接口采用 I2C 接口。

连接图



操作步骤

- 1、下载及安装官网最新软件。下载地址：<https://www.mindplus.cc> 详细教程：Mind+基础wiki 教程-软件下载安装
- 2、切换到“Python 模式”。“扩展”中选择“官方库”中的“行空板”和“pinpong 库”中的“pinpong 初始化”和“HUSKYLENS AI 摄像头”。切换模式和加载库的详细操作链接
- 3、进行编程
- 4、连接行空板，程序点击运行后，可在终端查看数据。行空板官方文档-行空板快速上手教程 (unihiker.com)

python 图形化编程

(1) 任意人脸



(2) 某某人脸



python 代码编程

以 pinpong 库为例，行空板官方文档-行空板快速上手教程 (unihiker.com)

```
# -*- coding: UTF-8 -*-
```

```
# MindPlus
```

```
# Python
```

```
from pinpong.libs.dfrobot_huskylen import Huskylen
```

```
from pinpong.board import Board
```

```
from unihiker import GUI
```

```
import time
```

```
Board().begin()
```

```
u_gui=GUI()
```

```
p_huskylen = Huskylen()
```

```
p_huskylen.command_request_algorithim("ALGORITHM_FACE_RECOGNITION")
```

```
img=u_gui.draw_emoji(emoji="Smile",x=0,y=0,duration=0.2)
```

```
while True:
```

```
    p_huskylen.command_request()
```

```
    if p_huskylen.is_appear_direct("blocks"):
```

```
        img.config(emoji="Smile")
```

```
    else:
```

```
        img.config(emoji="Nerve")
```

```
    time.sleep(0.2)
```

Copy

11. FAQ

问题 1: 怎么使用串口调试获得反馈数据?

回答 1: USB 接口的串口, 不能输入和输出指令的, 只能在转串口模块上 4pin 接口转, 二哈设置协议-自动识别, 二哈的 tx 连接转串口模块的 rx, 二哈的 rx 连接转串口模块的 tx。

图中使用的转串口模块购买链接: <https://www.dfrobot.com.cn/goods-242.html>

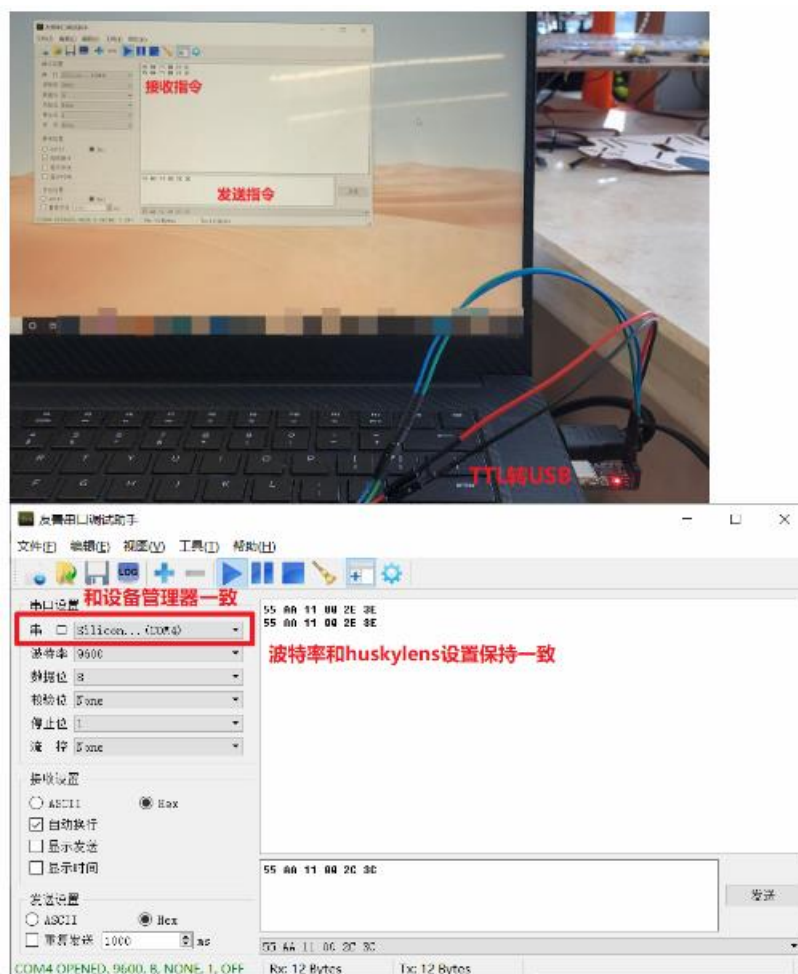
图中使用的杜邦线模块购买链接: <https://www.dfrobot.com.cn/goods-703.html>

使用参考贴: <https://mc.dfrobot.com.cn/thread-313836-1-1.html>

请参考: - HuskyLens 通信协议文档(GitHub)

或 HuskyLens 通信协议文档(PDF)

样例:



问题 2: 二哈的帧率是多少?

回答 2: 在不同模式下的图像处理的 FPS 会有所不同。Object Tracking, Face Recognition, Obejct Recognition 中, 图像处理 FPS 为 15Hz。
Color Recognition, Line Tracking, Tag Recognition 中, 图像处理 FPS 为 30Hz。

问题 3: 二哈能做什么功能?

回答 3: 请看相关代码块就能自己明白的

二哈识图

```
1 HuskyLens 初始化I2C地址为0x32
2 HuskyLens 切换到 人脸识别 算法 直到成功
3 HuskyLens 请求一次数据 存入结果
4 HuskyLens 从结果中获取 已学ID总数
5 do while
6   HuskyLens 从结果中获取 方框 是否在画面中?
7   HuskyLens 从结果中获取靠近中心 方框 的 ID 参数
8   HuskyLens 从结果中获取ID 1 是否已学习?
9   break
10  HuskyLens 从结果中获取ID 1 方框 是否在画面中?
11  HuskyLens 从结果中获取ID 1 方框 的参数 X中心
12 HuskyLens 从结果中获取 方框 总数
13 HuskyLens 从结果中获取第 1 个 方框 的 ID 参数
14 HuskyLens 从结果中获取ID 1 方框 总数
15 HuskyLens 从结果中获取ID 1 第 1 个 方框 的参数 X中心
16 HuskyLens 自动学习一次 ID 1
17 HuskyLens 遗忘当前算法的所有学习数据
18 HuskyLens 设置当前算法ID 1 名字为 "Mind+"
19 HuskyLens 屏幕叠加显示文字 "Mind+" 在x 230 y 35
20 HuskyLens 清除屏幕显示的文字
21 HuskyLens 触发 拍照 保存到SD卡
22 HuskyLens 保存 当前算法数据为SD卡 0 号模型
```

书包



—“物体分类”能完成：

- 1.学生着装（校服、红领巾、校徽识别）
- 2.驾乘车安全（安全带、安全帽识别）
- 3.体育馆球类设备（篮球、足球等识别）
- 4.学生就餐（光盘识别）
- 5.厨房亮灶（厨师帽识别）
- 6.动物保护（流浪猫、流浪狗、珍稀动物识别）
- 7.海洋馆解说（动物识别）
- 8.树木保护（火灾、积雪、冻雨识别）
- 9.姿态检测（倒地、拥挤识别）
- 10.文化传承（甲骨文、金文识别）
- 11.停车收费（车辆类型识别）
- 12.电商图片（自动分类识别）

- 13.智能垃圾桶（垃圾识别）
- 14.工业智造（零件划痕自动识别）
- 15.化工厂加油站（抽烟识别）
- 16.电梯预警（电动车识别）
- 17.公路监控（车辆碰撞识别）
- 18.图书馆管理（书籍错放、书架空位识别）
- 19.景区游客行为（乱丢垃圾、破坏景观识别）
- 20.超市货架管理（商品缺货、摆放混乱识别）
- 21.农田作物监测（病虫害、缺水缺肥识别）
- 22.校园环境清洁（垃圾堆积、卫生死角识别）
- 23.物流仓库管理（货物破损、分类错误识别）
- 24.银行安保（异常人员徘徊、蒙面识别）
- 25.机场安检（违禁物品识别）
- 26.博物馆文物保护（展品位移、受损识别）
- 27.医院病房管理（患者离床、输液完成识别）
- 28.公交站台秩序（插队、推挤识别）
- 29.商场消防通道畅通（堵塞识别）
- 30.养殖厂动物健康（生病症状识别）

.....

二、“人脸识别（或物体识别、物体追踪、物体分类）”能完成：

- 1.人脸（或物体）计数（密度，多少）
- 2.人脸（或物体）中心（位移，上下左右）
- 3.人脸（或物体）识别（是谁，张三）

注：“物体识别”目前仅支持 20 种物体，并追踪。分别为：飞机、自行车、鸟、船、瓶子、巴士、汽车、**猫**、椅子、牛、餐桌、**狗**、马、摩托车、人、盆栽植物、羊、沙发、火车、电视。

.....

三、“巡线、颜色识别、物体追踪、物体分类”能完成：

- 1.智能驾驶
2.

四、“标签识别、二维码识别、条码识别”能完成：

- 1.某某管理
2.

12. 更多文档（略）

二、离线语音识别

SKU:SEN0539

详细教程详见官方网址:

https://wiki.dfrobot.com.cn/SKU_SEN0539_Gravity_Voice_Recognition_I2C_UART

概 述

该模块采用了全新的离线语音识别芯片。内置 150 条常用的固定命令词条，新增命令词

自学习功能，自学习命令词可以不是一段语音，可以是一段口哨、一个响指、一声猫叫等，支持 17 条自学习命令词。采用双麦克风收音使模块有更好的抗噪音能力和更远的识别距离。模块自带一个喇叭和外接喇叭的接口，能实时语音反馈识别结果。模块采用 I2C 和 UART 两种通讯方式，Gravity 接口，兼容 Arduino Uno、Arduino Leonardo、Arduino MEGA、FireBeetle 系列控制器，树莓派，ESP32 等主控。

模块特点

Arduino 兼容控制器：Arduino Uno、Arduino Leonardo、Arduino MEGA、FireBeetle 系列控制器，树莓派，ESP32

通过 Gravity 连接，简化方便，且兼容 3.3V 与 5V

命令词自学习功能：通过语音控制模块学习命令词，可以使用任何音频作为命令

内置 150 条常用的固定命令词条

模块自带一个喇叭和外接喇叭的接口，能实时语音反馈识别结果

采用 I2C 和 UART 两种通讯方式，Gravity 接口。

板载电源指示灯（红）和识别状态指示灯（蓝）

双麦克风收音更好的抗噪音能力和更远的识别距离

产品参数

工作电压：3.3~5V

最大工作电流：≤370 mA(5V)

通信方式：I2C/UART

I2C 地址：0x64

固定命令词条数：135 条

学习命令词条数：17 条

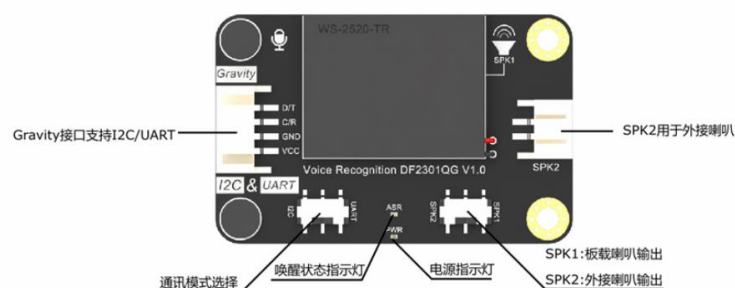
学习唤醒词条数：1 条

板载麦克风灵敏度：-28db

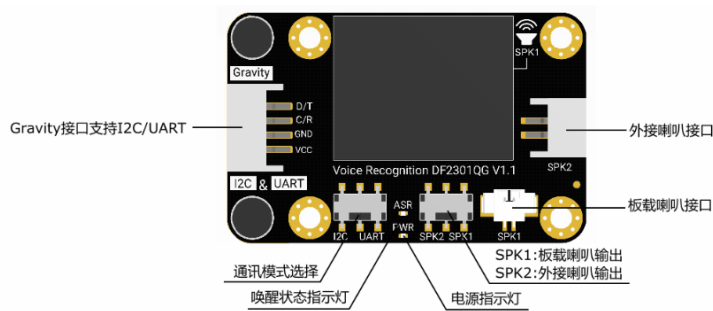
模块尺寸：49 * 32 mm

接口与引脚

V1.0 引脚介绍



V1.1 引脚介绍



学习命令词

唤醒词

唤醒词是指将产品从待机状态切换到工作状态的词语, 是用户与语音互动产品的第一个接触点。

学习唤醒词

首先用默认的唤醒词唤醒语音助手, 然后说出“学习唤醒词”, 根据提示来学习唤醒词 (每次学习命令词前需删除前一次学习的唤醒词, 请参考删除唤醒词和命令词删除)

提示: 学习状态中, 保持安静, 请说出需要学习的唤醒词!

需学习的唤醒词 (以理想同学为例): 理想同学

提示: 学习成功, 请再说一次!

需学习的唤醒词: 理想同学

提示: 学习成功, 请再说一次!

需学习的唤醒词: 理想同学

提示: 学习完成

就可以使用学习过的唤醒词来唤醒语音助手!

固定命令词条

命令词是指用户对语音互动产品发出一定的指令, 以此与其进行沟通的词语。

学习命令词

用唤醒词 (默认或已学习的) 唤醒语音助手, 然后说出“学习命令词”, 根据提示来学习命令词 (每次学习命令词前需删除前一次学习的命令词, 请参考删除唤醒词和命令词删除)

提示: 学习状态中, 保持安静, 请按提示学习命令词! 请说出第一条要学习的指令!

需学习的命令词 (以打开红灯为例): 打开红灯

提示: 学习成功, 请再说一次!

需学习的命令词: 打开红灯

提示: 学习成功, 请再说一次!

需学习的命令词: 打开红灯

提示: 恭喜你第一条指令学习成功, 请说出第二条要学习的指令

………… (继续学习即可)

或者使用“退出学习”来退出当前的学习状态。

学习结束后会自动生成一个 ID 详情见下文命令词/唤醒词 ID 对照表, 通过这个 ID 编写程序来控制即可。

删除唤醒词和命令词

用唤醒词（默认或已学习的）唤醒语音助手，然后说出“我要删除”，根据提示来学习命令词
提示：请问需要删除的是学习的唤醒词还是命令词
删除命令词：删除学习过的命令词
删除唤醒词：删除学习过的唤醒词
全部删除：删除学习过的唤醒词和命令词
退出删除

三、离线语音合成

SKU:DFR0760

详细教程详见官方网址:

https://wiki.dfrobot.com.cn/_SKU_DFR0760_Gravity_Speech_Synthesis_module_V2.0

简介

让声音为你的项目增添一抹特色！连接上语音合成模块，再添加几行简单的代码就可以让您的项目开口说话。无论是中文还是英文对于语音合成模块来说都是 so easy，播报当前时间，播报环境数据统统不在话下，与语音识别模块结合还可实现语音对话！该模块采用 I2C 和 UART 两种通讯方式，Gravity 接口，兼容绝大部分主控。模块上已经自带了一个喇叭，所以您无需再额外的添加喇叭。

注意：语音合成模块与 micro:bit 电机驱动扩展板地址相同，故不能一起使用。

V2.0 功能改动

发音更加自然

多音字发音错误率更小

无发音人选择

特性

支持中文、英文和中英文混合合成

自带喇叭

Gravity I2C/UART 双通讯

支持多种文本控制标识

支持多种主控板，arduino、micro:bit、掌控板等



应用场景

机器人语音

语音播报

语音提示

文本阅读

技术规格

供电电压：3.3V~5V

工作电流：160mA

I2C 地址：0x40

工作温度范围：-40℃~85℃

产品尺寸：42*32mm

引脚说明

序号	丝印	功能描述
1	D/T I2C	数据线/TX
2	C/T I2C	时钟线/RX
3	GND	电源负极
4	VCC	电源正极
5	AOP	音频输出正，可外接 D 类功放后接外置喇叭
6	AON	音频输出负，可外接 D 类功放后接外置喇叭

语音识别与语音合成使用案例

智能垃圾桶（语音识别、语音合成、OLED 显示、舵机开盖）基本教程

一、灵感来源：

垃圾分类，一般是指按一定规定或标准将垃圾分类储存、投放和搬运，从而转变成公共资源的一系列活动的总称。分类的目的是提高垃圾的资源价值和经济价值，力争物尽其用，减少垃圾处理量和处理设备的使用，降低处理成本，具有社会、经济、生态等几方面的效益。当今，垃圾分类是一个新潮的问题，垃圾分类的大潮终将席卷全国。为此，设计出这样一个智能分类垃圾桶，来让我们的生活更快更便捷地适应这样的趋势，也让小朋友更好地了解垃圾分类分为可回收垃圾、不可回收垃圾和（有害垃圾、其他垃圾）等类别。

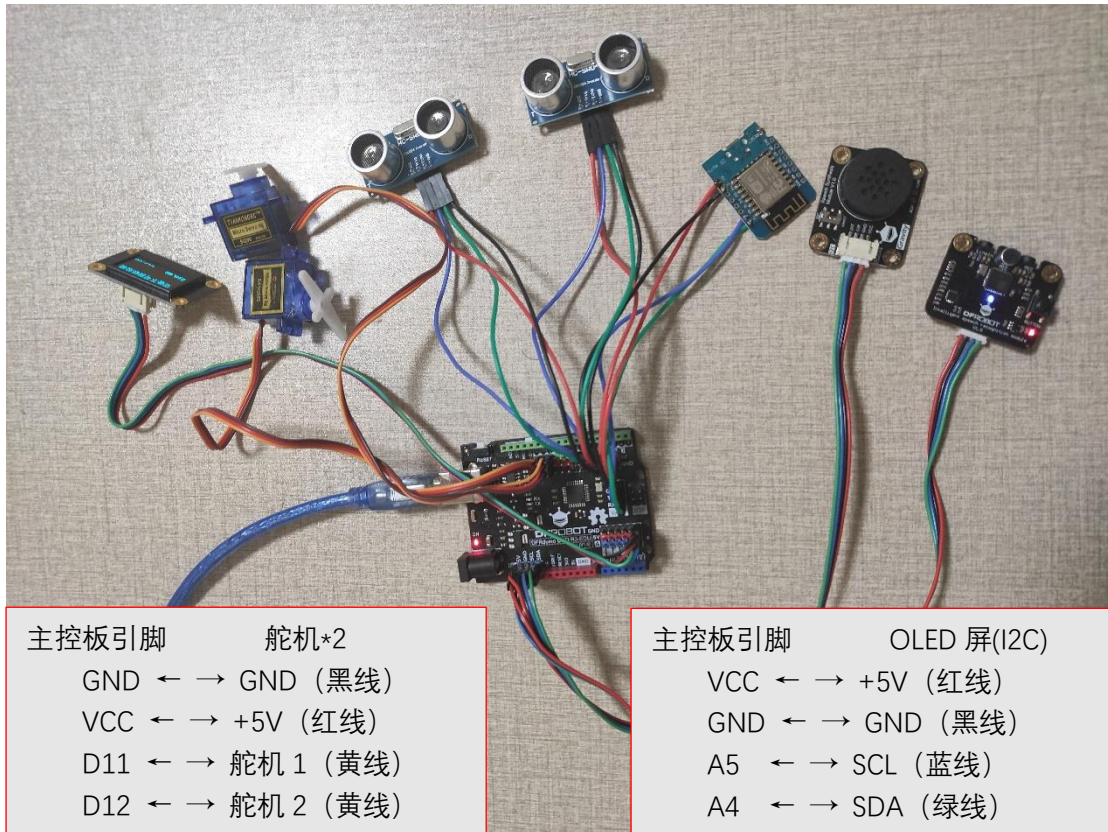
二、作品功能：

小朋友可以通过自然语音告诉垃圾桶需要扔的垃圾的名称，垃圾桶可以自动辨别垃圾的类型并告知你垃圾分类相关知识，同时自动打开对应垃圾桶让你投放，屏幕调试显示垃圾分类。

三、硬件清单：

1. Arduino Uno 主控板*1
2. 语音识别模块*1
3. 语音合成模块*1
4. OLED 显示屏*1
5. 舵机*2
6. 电源

四、硬件连接：



主控板引脚	语音识别(I2C) 语音合成(I2C)
VCC	← → +5V (红线)
GND	← → GND (黑线)
SCL	← → SCL (蓝线)
SDA	← → SDA (绿线)

主控板引脚	超声波(1) 超声波(2)
VCC	← → +5V (红线)
GND	← → GND (黑线)
D6	← → T (蓝线)
D7	← → E (绿线)
D8	← → T (蓝线)
D9	← → E (绿线)

五、垃圾分类：

(一) 可回收垃圾：

1 fei bao zhi 废报纸,2 su liao ping 塑料瓶,3 yi la guan 易拉罐,4 bo li ping 玻璃瓶,5 su liao wan ju 塑料玩具,6 xin feng 信封,7 jiu shu ben 旧书本,8 bu liao 布料,9 fei jiu jin shu 废旧金属

(二) 不可回收垃圾：

11 fei dian chi 废电池,12 guo pi 果皮,13 fei deng guan 废灯管,14 shui yin wen du ji 水银温度计,15 sha chong ji ping 杀虫剂瓶,16 niao bu si 尿不湿,17 can jin zhi 餐巾纸,18 sheng cai sheng fan 剩菜剩饭,19 yan tou 烟头

六、Mind+编程：



七、溢满告知调试：

1. 将上面用 Mind+ 编好的程序，上传到设备（arduino Uno），用纸张（或其他物品）挡在超声波传感器前，就会看到 OLED 屏第四行显示的超声波测得的数据，听到语音合成模块非常

的声音（可回收垃圾桶溢满，快来更换！或不可回收垃圾桶溢满，快来更换！）

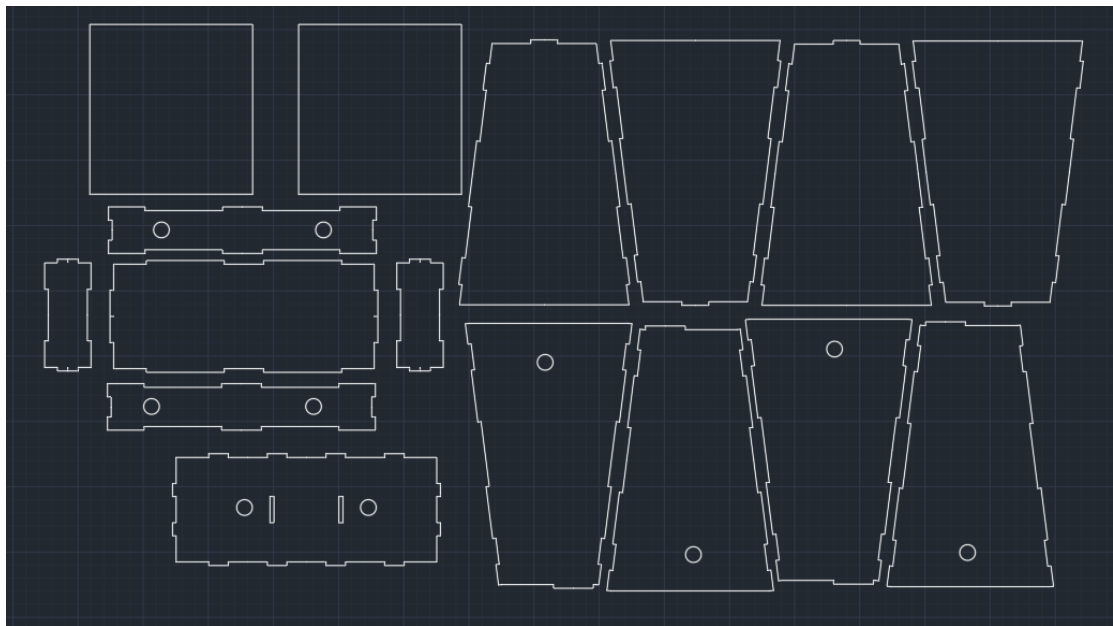
2. 自然语音播报需要扔的“垃圾的名称”（详见六、垃圾分类），垃圾桶可以自动辨别垃圾的类型并告知你垃圾分类相关知识，同时自动打开对应垃圾桶让你投放，屏幕显示垃圾分类。

3.调试视频

http://jmsck.cn/iot_ai/2021-04-21-ljt2.mp4

八、结构件制作及作品多角度照片：

1. 结构件制作



2. 作品多角度照片



