

Pixy 快速入门指南

1. 教 Pixy 一个物体

教 Pixy 识别一个物体是非常简单的。但让我们先说一下什么要的物体会更好的被 Pixy 所识别。Pixy 使用的是基于色调过滤算法(hue-based color filtering algorithm)来识别物体。因为 Pixy 使用色调(hue)，物体应有明显的色调，下面的物体就很好，因为他们都有比较好、易辨识的色调。



下面有几个不好识别的物体，因为他们没有色调（黑，白或灰）或因为色调不明显。



把这条原则记在心里，挑选一个物体来教 Pixy。（通过电池或 USB 线给 Pixy 供电,如果你还没有。当你给 Pixy 供电时，Pixy 会闪几下 LED。等待 LED 灭掉后再教 Pixy。）用

手按住 Pixy 顶部的按钮，等待大约 1 秒钟，LED 会变亮，先显示白色，然后是红色，然后是别的颜色，当显示红色时就松开按钮。

当你松开按钮后，Pixy 会进入所谓的“light pipe”模式，LED 的颜色就是 Pixy 摄像头所看到的图像中间部分的颜色。

用 LED 的颜色来检查是否你的物体在 Pixy 视角的中心位置。当你感觉 LED 颜色和你的物体颜色很接近时，点击一下按钮（按下并释放按钮，就像单击鼠标一样）。如果 Pixy 感觉你的物体色调足够好（有足够的色彩饱和度[enough color saturation]）的话，LED 会闪烁，你现在已经操作完成。Pixy 现在已经“学会”了你的物体，并且会追踪具有相同颜色的物体。如果物体的色调没有那么好，LED 会关闭（不会闪）并且 Pixy 并没有学到物体。如果 LED 没有闪，你可以换一个有更好色彩饱和度的物体。

多个标记[Signatures]

Pixy 可以学习多达 7 个颜色标记。在上面部分，我们通过松开按钮当 LED 变红色时，已经教了 Pixy 第一个颜色标记。如果我们继续按下按钮，LED 会变为橙、黄等，来指示剩下的颜色标记。下面是标记的顺序：

- 1、红
- 2、橙
- 3、黄
- 4、绿
- 5、青
- 6、蓝
- 7、紫

颜色标记的编号是指你松开按钮时的颜色。松开按钮 LED 颜色是黄色，那么你在教 Pixy 第三个标记。松开按钮后 LED 变蓝色，你教的是第 6 个标记。这些颜色跟实际的物体色调没有任何关系。这些颜色只是用来识别标记编号。举例，标记 1 可能是黄色的物体，虽然标记 1 的提示颜色是红色，标记 2 可能是粉色物体，虽然它的提示颜色是橙色。

关于教物体的其它说明

当你按下并松开按钮来表示 LED 颜色 and 实际物体颜色匹配时，Pixy 会闪 LED，并且闪烁的亮度代表物体的色调的饱和程度。所以高亮度的闪烁是好的。试着教 Pixy 一个颜色鲜艳的物体——Pixy 会通过闪烁 LED 来指示它是不是“高兴”。

如果你偶然发现本打算教 Pixy 第 3 个颜色标记时，它却再学第 2 个。你可以按下按钮直到 LED 关闭后再松开。这将是取消 Pixy 学习模式的方法。你可以通过按住按钮重新进入学习摸索。

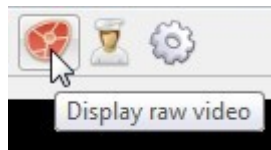
白平衡

某些类型的灯（如白炽灯）有略红的色调，而有些（如荧光灯及常见的节能灯）会有略蓝的色调。这样的光线会影响你的颜色标记。比如，如果你在白炽灯条件下教的物体，转移到荧光灯下，颜色标记可能不会被识别的非常好。你可以重新教标记或调节白平衡。

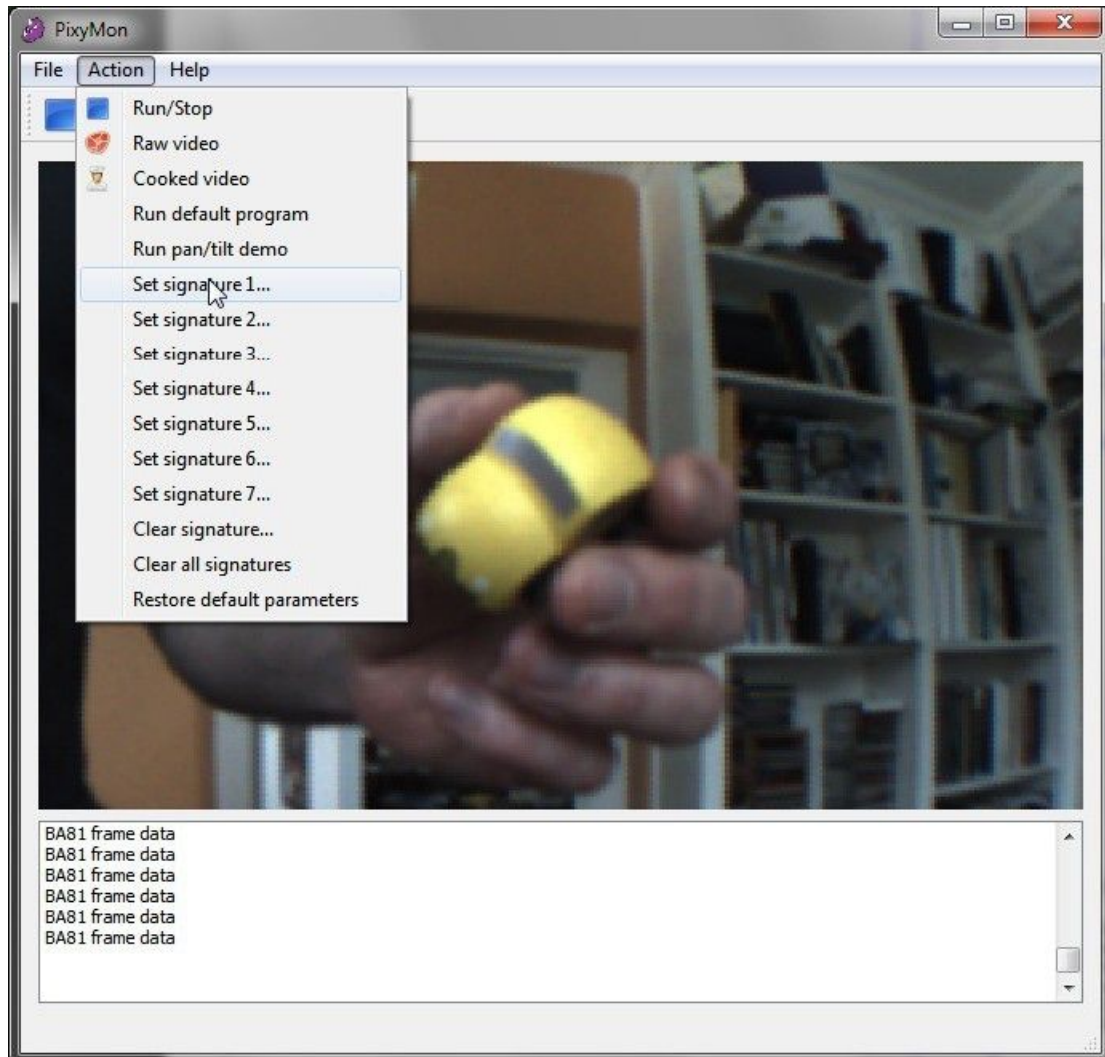
当你第一次上电时，Pixy 会花 5 秒钟的时间来监测需要用哪个合适的白平衡。然后它会关闭自动白平衡的功能。如果你想重新配置白平衡，按下按钮直到 LED 变白后松开，时间会很快，做好松开的准备。Pixy 现在就是在自动白平衡摸索中。你可以拿一张白色的纸在 Pixy 前面，从而让 Pixy 可以进行白平衡调节，这将会花 2~3 秒钟的时间。然后你可以像点鼠标一样按下并松开按钮。LED 会闪烁来表示白平衡调试成功，完成了。

通过 PixyMon 来教 Pixy

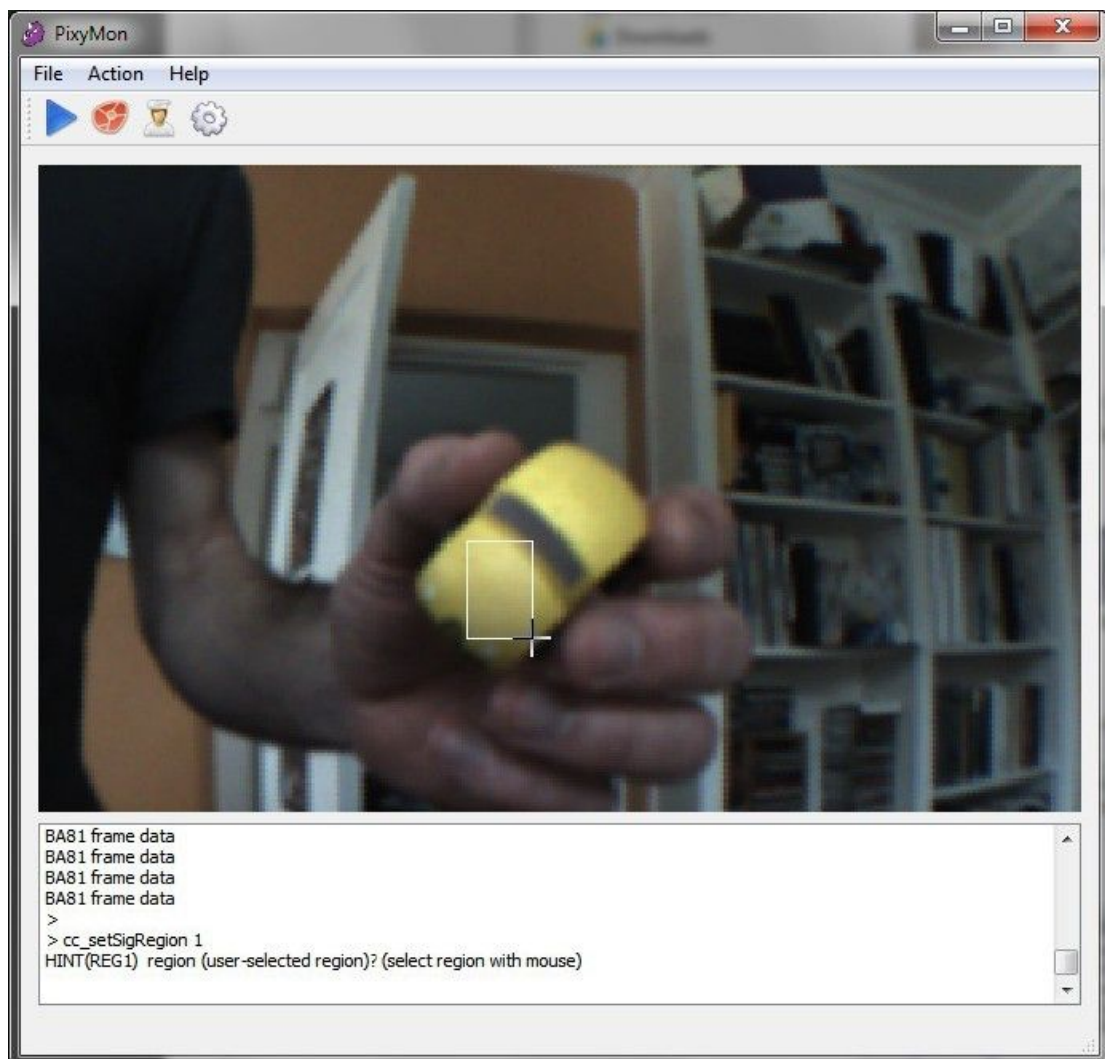
你也可以通过 PixyMon 来教 Pixy。当你的物体很小或者你想更好的控制哪些像素被用到时会很有帮助。通过连接 USB 线并且运行 PixyMon。当 PixyMon 打开后，点击上面的 raw 按钮。



现在拿着要教 Pixy 的物体在它前面，然后选择菜单 Action=》Set signature 1。



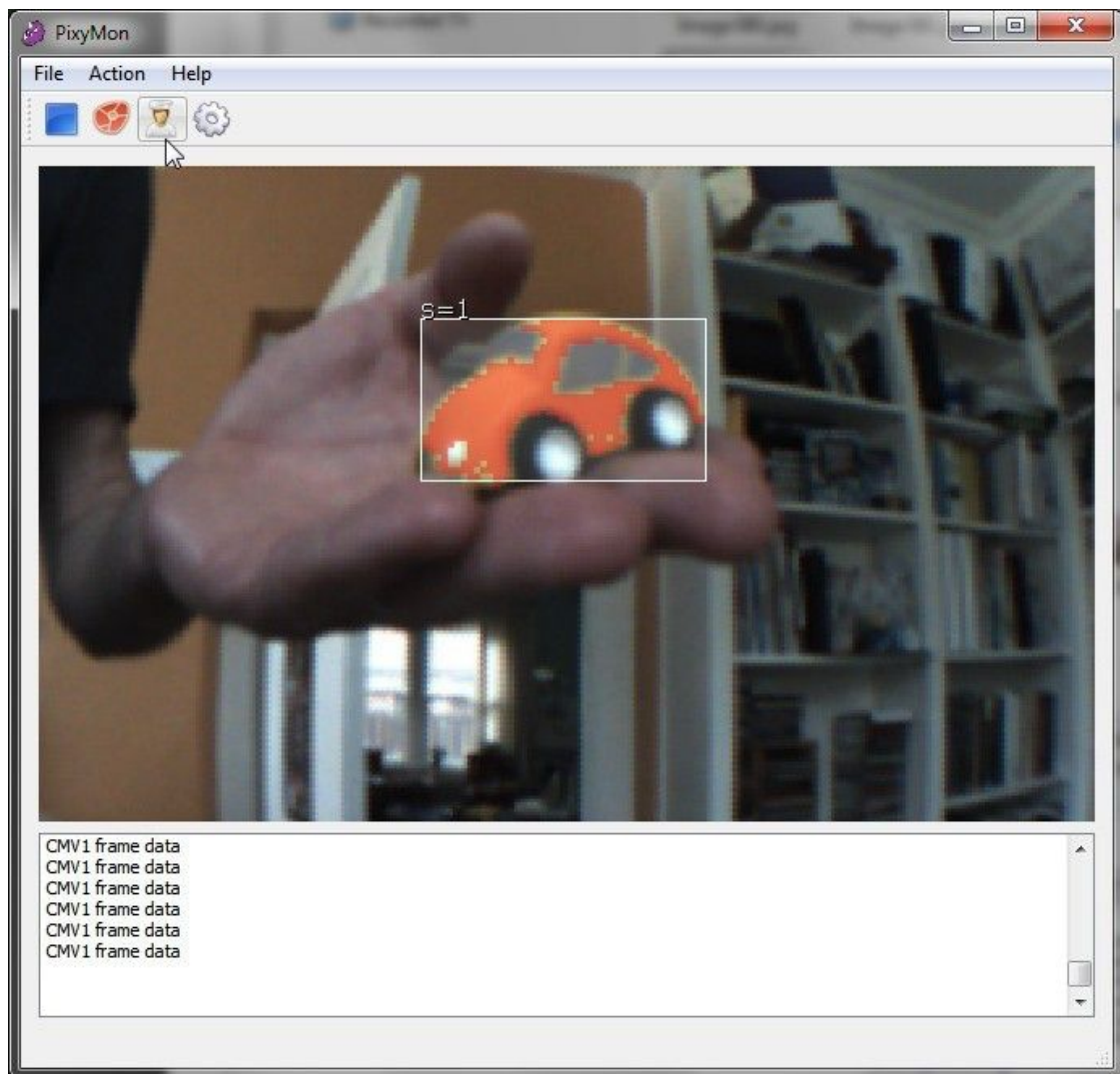
现在选择哪一个区域是你想用的物体。



如果成功，你应该可以在下面的命令区看到“Success!”信息，并且还有一个数字是 0~100 之间来指示色调饱和度。色调饱和度高会有更高的数字。

就是这样了，你现在可以监测颜色标记的效果了，通过点击厨师图标或选择 **Action→Cooked Video**。Cooked Video（处理过的图像）会显示哪些像素与颜色标记一致，并且如何组成识

别的物体：



下面是什么呢？

你可以通过这种方式教 Pixy 识别物体，最多可以有 7 种颜色标记。需要注意的是 Pixy 总是将标记记忆在 flash 里，所以当你下一次上电时，这些标记都是直接可用的。

2. 上传新固件

Pixy 使用保存在 ROM 里的 bootloader 来上传最新的固件到 Flash 里。这使得 Pixy 不会“变砖”，也就是说你总可以从不成功的固件上传或不完美固件版本的升级中恢复你的 Pixy。

第一次上传固件

请仔细按照本文操作。（如果你做错了某件事情或顺序颠倒，也不会有大问题，只是有可能会让你困惑。）OK，开始前做如下准备：

- *Pixy 上没有任何接线，没有电源，没有 usb，没有 IO，没有舵机等；
- *你的计算机没有运行 PixyMon（但是 PixyMon 已经在你的计算机上安装了）；
- *你自己的固件已经准备好了（.hex 文件）；

首先，将 USB 线插在计算机上



然后按下 Pixy 顶部白色的按钮，并且在保持按下按键的同时插上 USB 线。要同时完成，这可能是最麻烦的一个环节了。

Pixy 上的 LED 会变的非常亮（白色）并且保持，这说明你已经进入了固件上传模式。因为你这是第一次上传固件，你的计算机提示你安装固件，因为 Pixy 在固件上传模式下使用单独的驱动，所以计算机之前并没有安装过相应的驱动。这个过程只针对 Windows 用户，如果你用的是 Mac，你可以直接掉过驱动安装部分。

Windows Vista、7 或 8

你应该会看到这样的下弹窗：



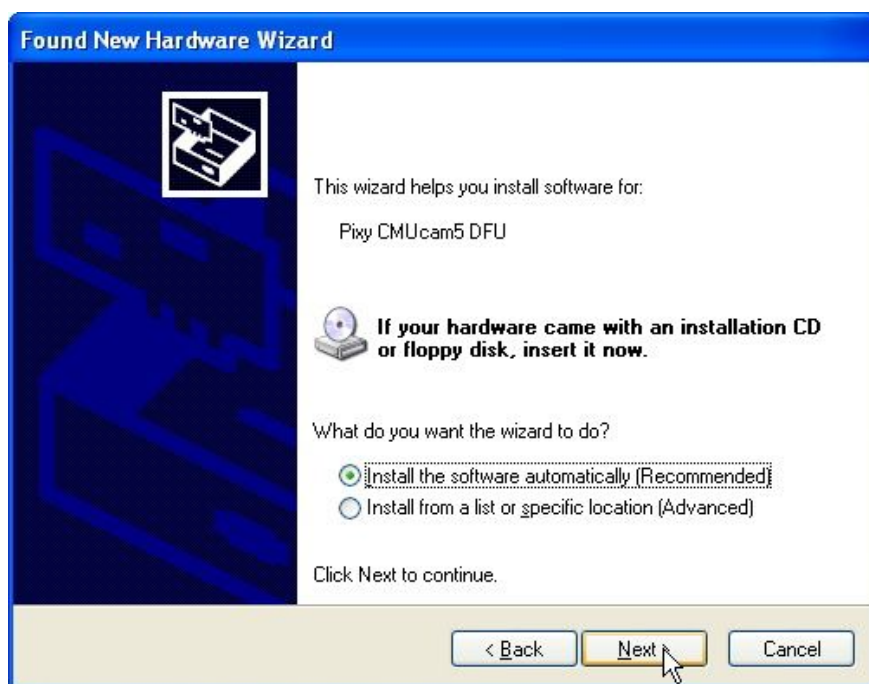
只需要等待驱动安装完成即可，期间你不需要做任何操作，仅仅是等待。

Windows XP

你会看到添加新硬件的对话框



点击，“No,not this time”然后点击“下一步”。然后点击“自动安装驱动”，然后点击“下一步”



驱动将会安装完成，等待完成后继续。

打开 PixyMon

首先打开 PixyMon,当 PixyMon 运行后,Pixy 上的 LED 会关闭,你会看到如下信息“Pixy programming state detected”在 PixyMon 的命令窗口区。如果你没有看到上面信息，有可能是驱动还在安装，再多等一段时间。

选择菜单 **File→Program**，选取文件对话框将会弹出，浏览到你的新固件位置（.hex 文件）然后选择相应文件，然后点击 OK。Pixy 会开始接受新固件，然后将其写入到 flash 中。这是非常快的，一般只需要 2~3 秒钟。当完成后，你会看到“done!”信息，你需要拔掉 USB

线然后重新插上来运行新固件。

任务完成。

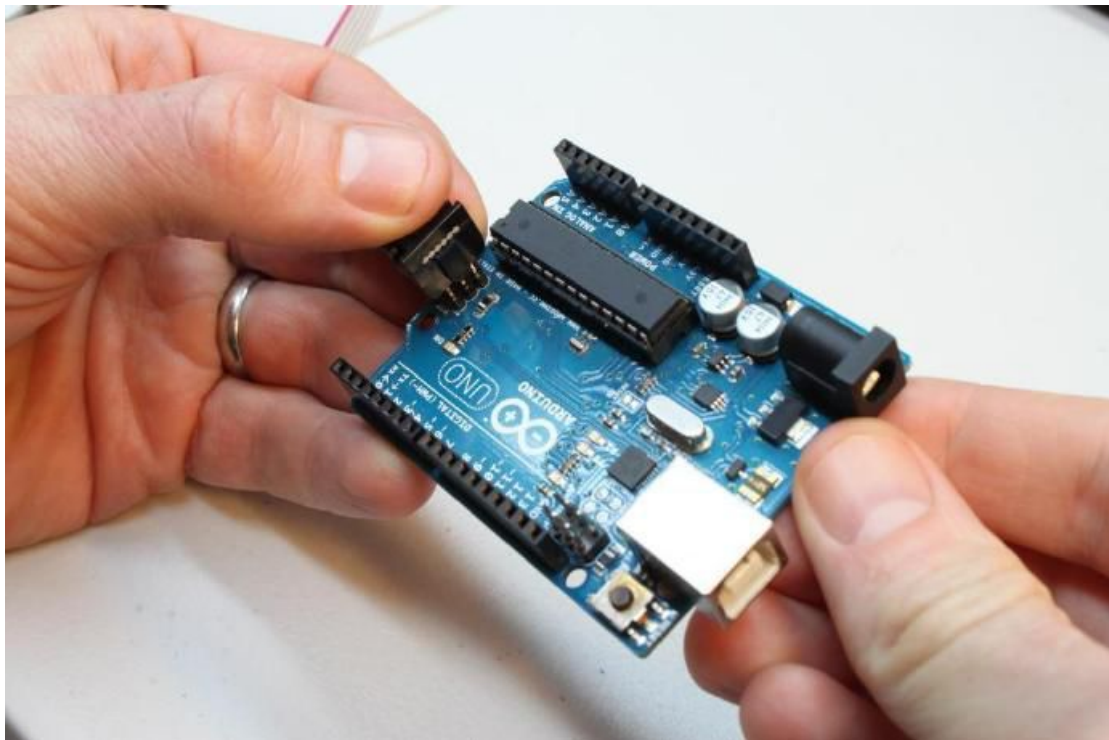
再一次安装固件

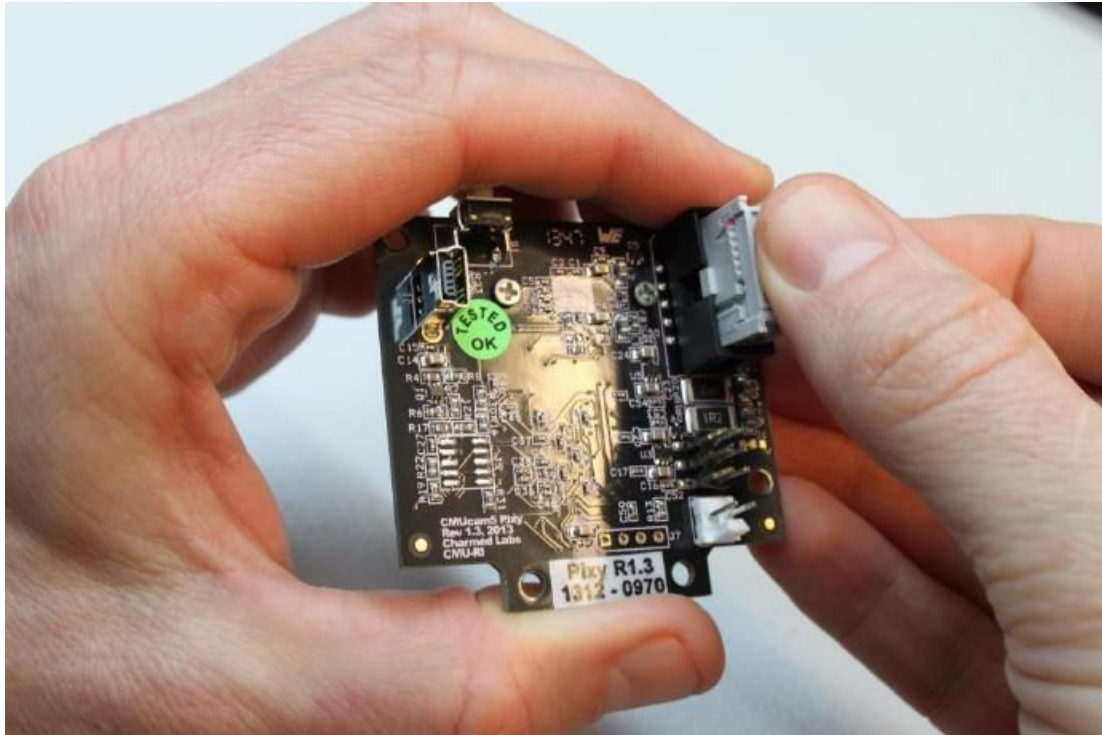
下一次上传固件时，你的驱动已经安装，所以你只需要通过按键然后插上 USB 线进入固件上传模式。并且 PixyMon 可以已经在运行，这并不影响。

3. Pixy 与 Arduino 连接

Pixy 被设计为可以与微控制器进行通讯，并且 Pixy 可以直接与 Arduino 进行通讯。它会以 1Mbps/s 的速度发送块信息给 Arduino，这意味着 Pixy 每秒可以发送超过 6000 个识别的物体或每帧 135 个被识别的物体（Pixy 每秒可以处理 50 帧画面）。

OK，想让 Pixy 与 Arduino 进行通讯，使用 Pixy 自带的数据线连接 Pixy 到 Arduino。





接下来, 下载 Arduino 库在这里: <http://pan.baidu.com/s/1dDpDlvV> , 打开 Arduino IDE 并且通过菜单 Sketch->Import Library 导入 Pixy 库文件, 然后浏览到 Pixy 库文件解压的地方。

然后, 通过菜单 **File→Examples→Pixy** 载入“hello_world”例子。上传代码到 Arduino 并且打开 Serial Monitor (串口监视器)。你应该可以看到一些类似于下面的数据信息:

```
Detected 1: block 0: sig: 1 x: 159 y: 109 width: 61 height: 61Detected 1:
block 0: sig: 1 x: 173 y: 114 width: 60 height: 61Detected 1: block 0: sig:
1 x: 146 y: 111 width: 70 height: 65...
```

注意, 这个例子中, 如果 Pixy 运行了默认的固件, 并且一个物体被识别才会打印这些信息。

Arduino API

在 Arduino 中使用 Pixy 是非常简单的, 你只需要包含 SPI 和 Pixy 的头文件:

```
#include <SPI.h>
#include <pixy.h>
```

并且在 setup()和 loop()函数外面创建一个全局的 Pixy 实例变量:

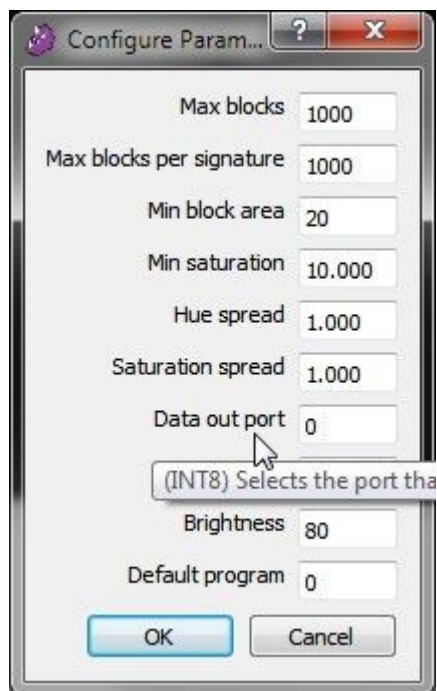
```
Pixy pixy;
```

API 包含一个函数: getBlocks(), 这个函数将返回 Pixy 所识别的物体数量。然后你就可以通过 pixy.blocks[]的数组得到每个识别到的物体数据 (每个数组成员对应一个被识别的物体)。每个成员 (i) 包含下列元素:

pixy.blocks[i].signature: 被识别物体的标记编号;
pixy.blocks[i].x: 被识别物体中心位置在 x 方向的坐标;
pixy.blocks[i].y: 被识别物体中心位置在 y 方向的坐标;
pixy.blocks[i].width: 被识别物体的宽度 (1~320);
pixy.blocks[i].height: 被识别物体的高度 (1~200);
pixy.blocks[i].print(): 一个成员函数用来打印被识别物体的信息到串口上;
所以 Pixy 与 Arduino 通讯很简单。

其它微控制器或设备

Pixy 支持多种方式输出被识别物体的数据。它支持 SPI, I2C, UART 或模拟/数字 I/O 口 (通过 10Pin 接口)。Pixy 还支持 USB2.0。你可以通过 PixyMon 的对话框来配置 Pixy 使用哪种方式。“Data out port”参数决定了输出方式。



如果你将鼠标指针移动到“Data out port”文字上, 会有提示信息显示各个类型的接口对应的文字:

SPI: 这是默认的输出端口, 使用了三根线 (IO 口的第 Pin1、3、4) 并且用于与 Arduino 通讯;

I2C: 两线接口 (IO 口的 Pin5 和 9), 可以支持一个主机与 127 个从机的通讯 (多大 127 个 Pixy);

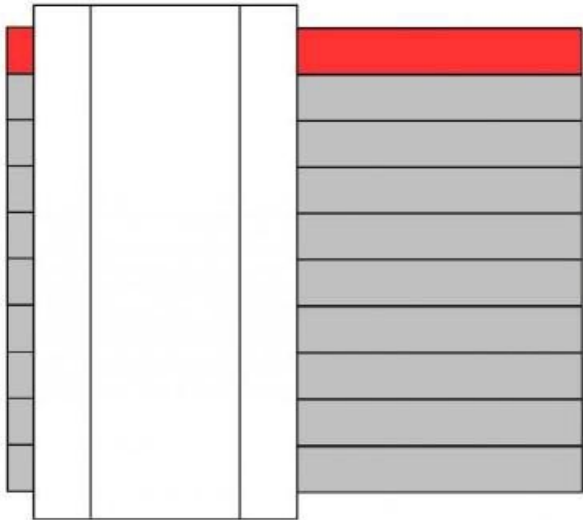
UART: 这是普通的串口 (IO 口的 Pin1 和 4)。Pixy 通过 Pin1 (输入) 接受数据, Pin4 (输出) 发送数据;

模拟/数字 X: 这种方式将在 Pin3 口输出 0~3.3V 的模拟电压, 表示监测到的最大物体的 X 值;

模拟/数字 Y: 这种方式将在 Pin3 口输出 0~3.3V 的模拟电压, 表示监测到的最大物体的 Y

值:

2



1: SPI MISO, UART RX, GPIO0

2: 5V (in or out)

3: SPI SCK, DAC OUT, GPIO1

4: SPI MOSI, UART TX, GPIO2

5: I2C SCL

6: GND

7: SPI SS, ADC IN, GPIO3

8: GND

9: I2C SDA

10: GND