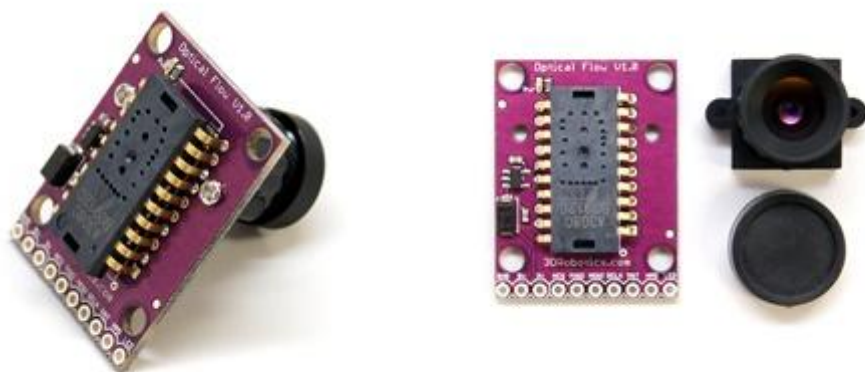
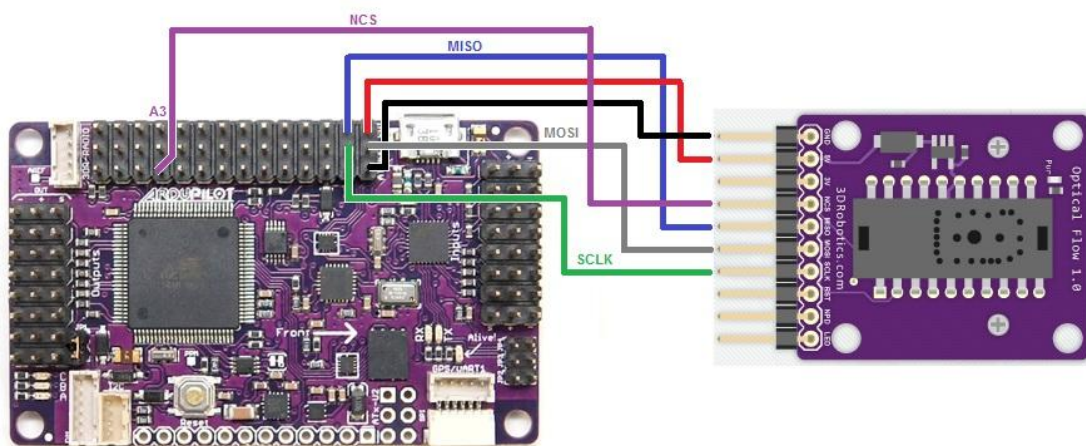


光流传感器使用说明

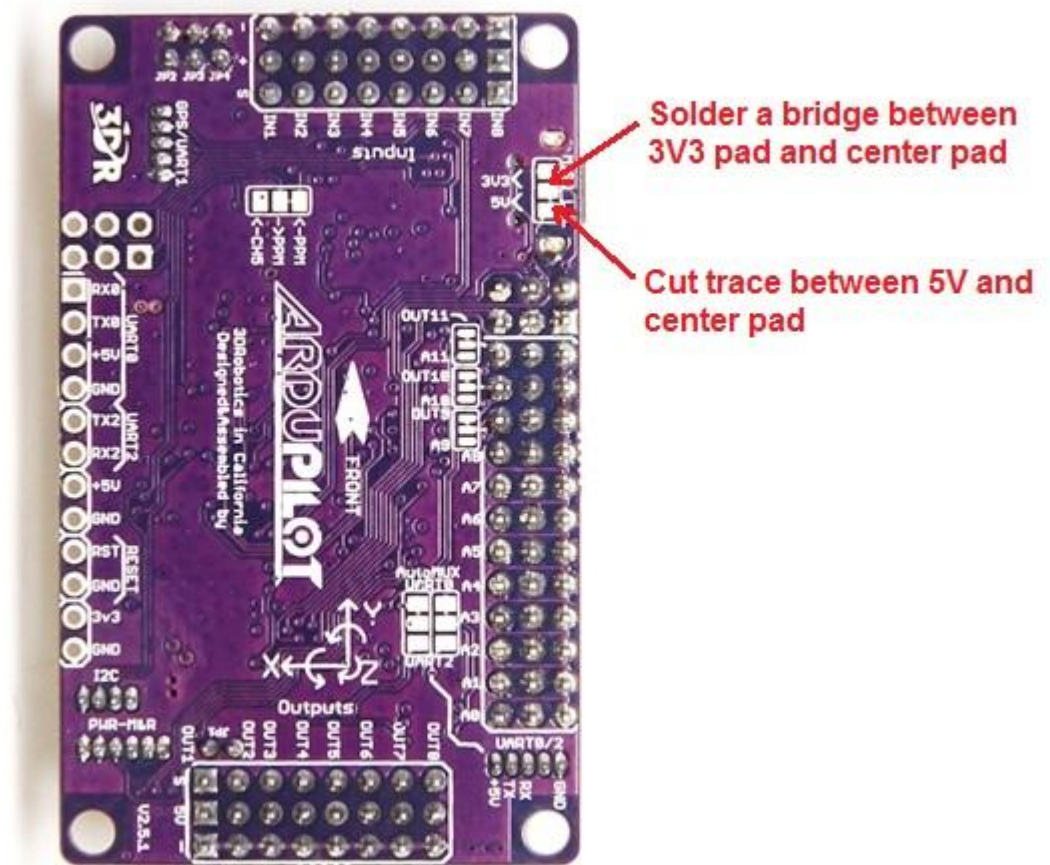


第一步：连接传感器至 APM2.5

- 连接 VCC, GND, MISO, MOSI, SCLK 和 NCS 引脚，如下图所示。



- 断开板子背面 的 MISOLVL 跳线，重新焊接使 MISO 引脚工作在 3.3v。这非常重要，确保光流传感器不会干扰 MPU6000。如下图所示。



第二步：测试传感器

- 把 AP_OpticalFlow_test.pde 项目加载到 APM 中
- 用串口监视器或 AP 任务规划器终端连接 APM
- 输入 'c', 确保传感器相应 APM
- 输入 'm', 把摄像头前后启动, 检查 x,y 值变化。如果没有变化, 左右旋转镜头调整焦距

第三步：从传感器捕获图像

为了检查你的镜头是否对焦合适, 你可以从传感器直接捕捉一张图片, 并用一个 Python 写的查看器显示该图片。

- 1 把 AP_OpticalFlow_test.pde 上传到 APM (参见上方)
- 2 安装 Python 2.7 (或更新的版本) <http://www.python.org/getit/>
- 3 启动 Python IDLE 编辑器
- 4 File, Open ADNS3080ImageGrabber.py
- 5 Run, Run Module - 会先出现 Python 命令行, 然后出现 ADNS3080ImageGrabber 程序
- 6 在 ADNS3080ImageGrabber 界面, 把默认 com 端口改成 APM 用的端口, 点击 Open

7 按 Send 按钮启动/停止捕获图片 (每 2 秒应该出一张新图片)

注: 你可以在 Python 命令行窗口看见 AP_OpticalFlow_ADNS3080 的菜单和任何错误提示

工作原理

鼠标传感器返回它看见的表面特征的平均运动 (x 和 y 方向)。单个像素点的运动不会使传感器输出 "1"。它会返回一个更高的 5 左右的值。这个值在下面成为 **scaler***。在下面的例子中, 返回值大约是 1.6 ((-5+5+5) / 3)

传感器的 x 和 y 值可以根据高度转换为实际距离

为了把传感器输出值转换成实际运动的距离, 我们需要考虑高度。从下面的两张图中可以看到, 如果两个四轴运动了同样距离, 但是一个低, 一个高, 低的一个会看到表面特征运动得更远, 所以光流数值会更大。

我们弥补飞行器侧倾和俯仰变化

飞行器的侧倾和俯仰的变化也将导致传感器返回的 X 和 Y 值的变化。与横向运动的计算不同, 这些都不依赖于可见物体的距离。在下面的图片中, 你可以看到的四轴已经侧倾了10度, 但第一张图中视角中心的两个花都移到了第二张图片的视角边缘。

传感器值预期的变化可以根据下面的公式用侧倾和俯仰的变化算出。我们把这些预期的变化从传感器返回的实际值中减去。

我们得到的 x/y 轴的运动后, 就可以与当前的偏航相结合, 用于估计位置。