

单片机快速入门

作者 Redstone/Designspark

在编写基于系统的单片机时，我承认自己很“老式”：我喜欢 C 语言和嵌入式编程，我还在等待更好的东西说服我去使用。

我也喜欢快速的得到概念验证启动与运行的方法。当我一转身发现，Python 作为最常用的编程语言与其应用在应用程序的现象越来越多，如机器学习与 Intel Movidius Neural Compute Stick。

因此，在本文中，我将展示如何运行一个“Hello World”应用程序，该应用程序用 Python 编写，并在半小时内从头开始运行一个主流的开发工具包。当然，除非你的互联网连接速度很慢，它可能会耗费一些时间。

STM32F401 Nucleo

在我们的开发工具包中，使用 ST Micro Nucleo F401RE 板。这是一个功能强大的嵌入式应用开发板，并为我们提供：

MCU: STM32F401RET6 ARM®Cortex®-M4 32-bit, 84MHz

SRAM: 96 KB

Flash: 512 KB

Digital I/O: 50 pins at 3.3V (in the ubiquitous Arduino format)

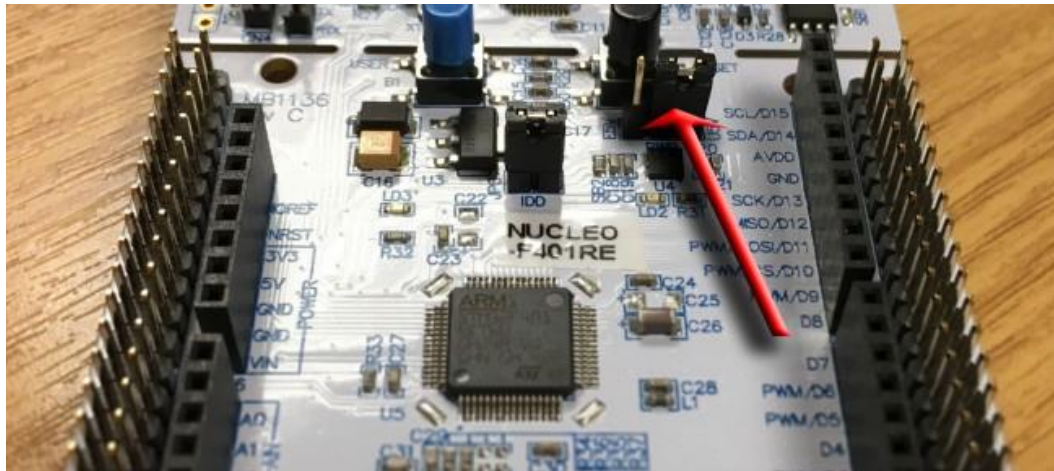
ADC input: 16 pins

... 在许多其他有趣的功能中

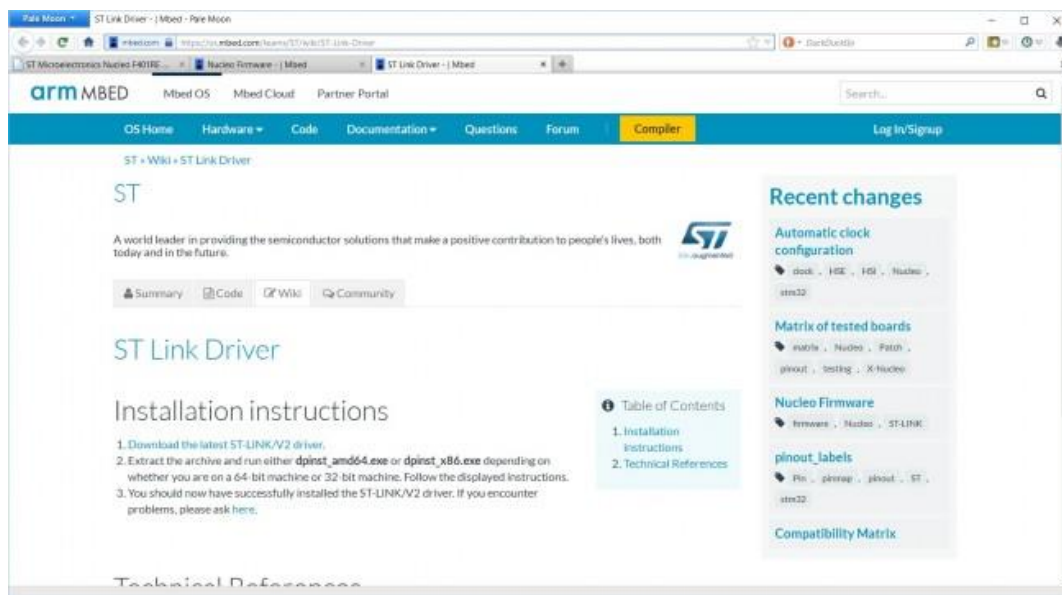
使用 Nucleo 的一个好处是，我们需要启动的另外一个硬件 USB A 到 USB mini 电缆，我们将用它们来供电和编程。

在我们开始之前

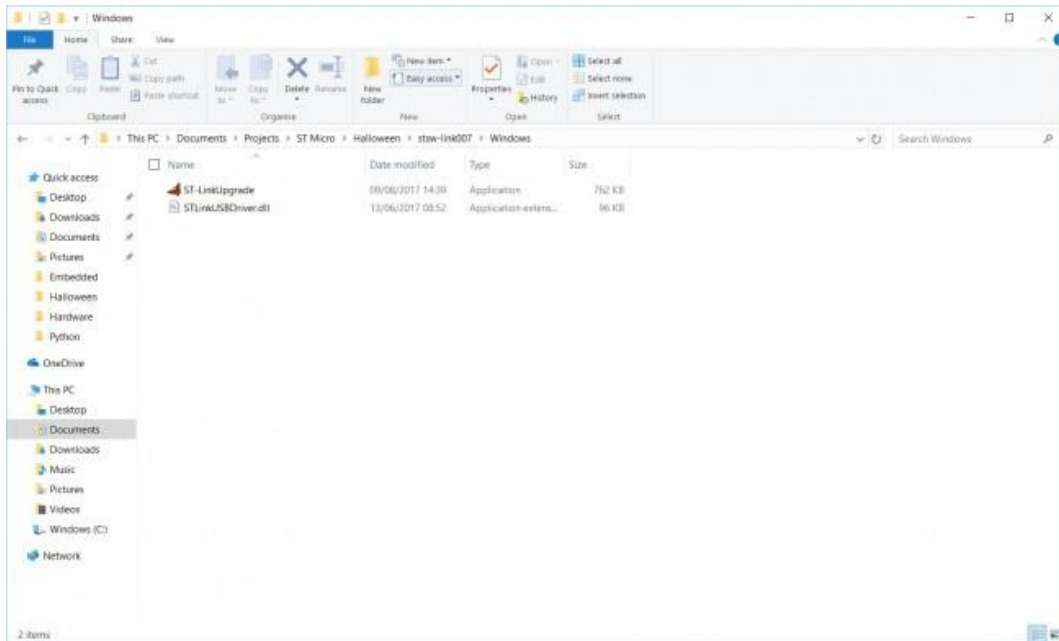
在我们使用 ST Nucleo 板之前，我们需要做一些事情。首先要做的是确保我们的电源跳线在 Nucleo 板上的正确位置：在 JP5 上，跳线应该连接 pin 1 和 pin 2：



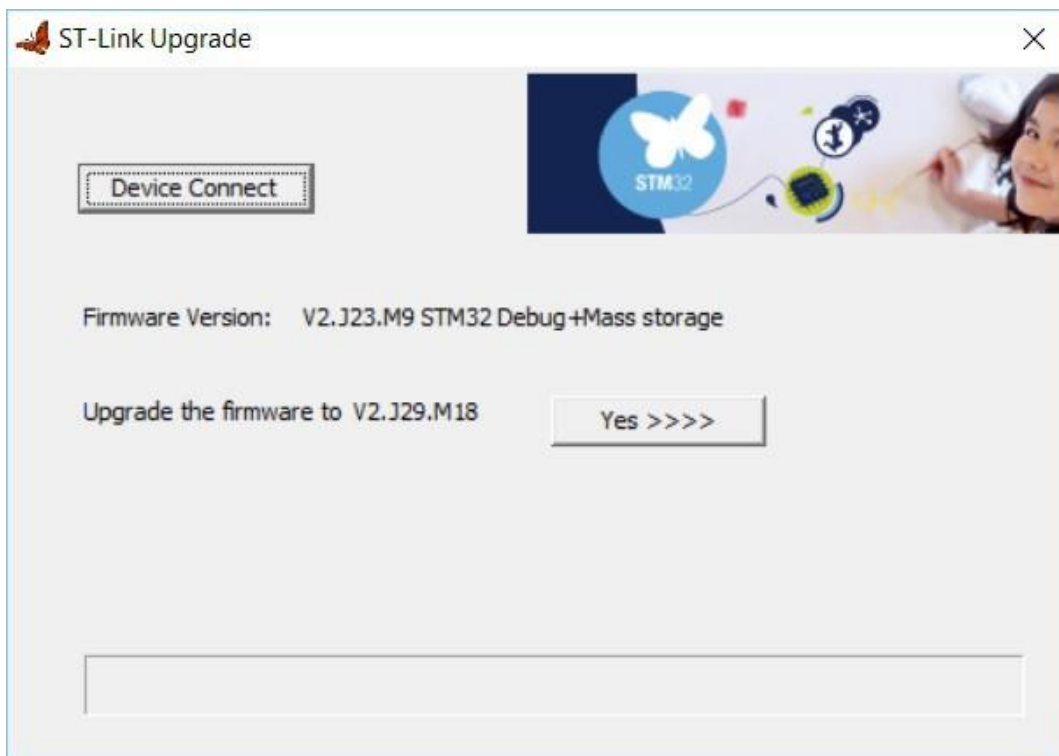
如果您使用的是 Windows，那么下一步就是下载并安装 [ST-Link driver](#)。它使我们能够访问 ST Nucleo 编程端口并在主机上创建一个虚拟的 COM 端口。如果你使用的是 MAC OSX 或 Linux，就可以很好的运行了，因为设备会自动识别。



在使用 Python IDE 之前，我们还需要升级机载固件。升级和安装说明可以在[这里](#)找到。当你下载了 zip 文件并打开它时，Windows 用户可以找到一个可执行文件：



运行该可执行文件，单击[Device Connect]按钮：



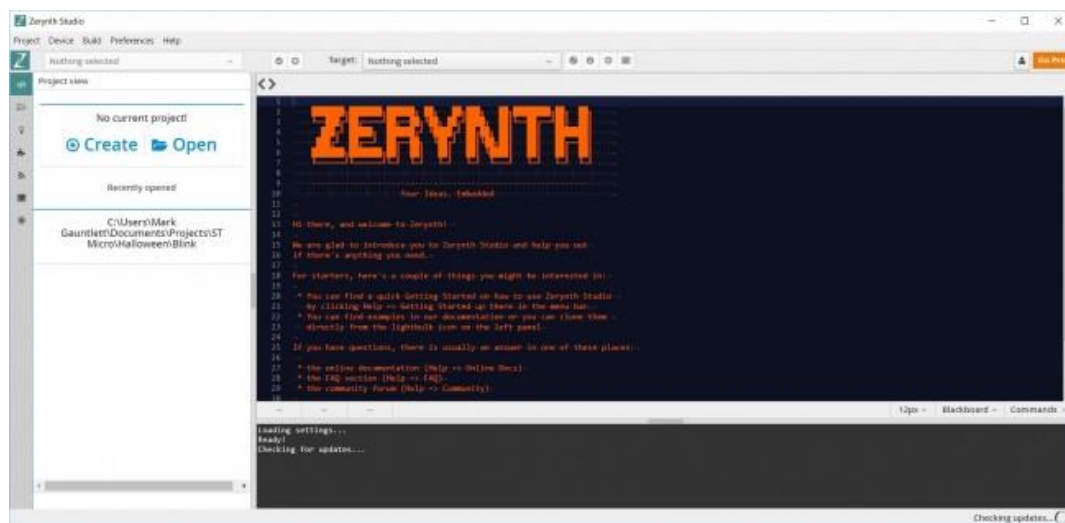
单击[Yes>>>]按钮。完成后，您应该得到一个“Update Successful”消息

Zerynth Studio

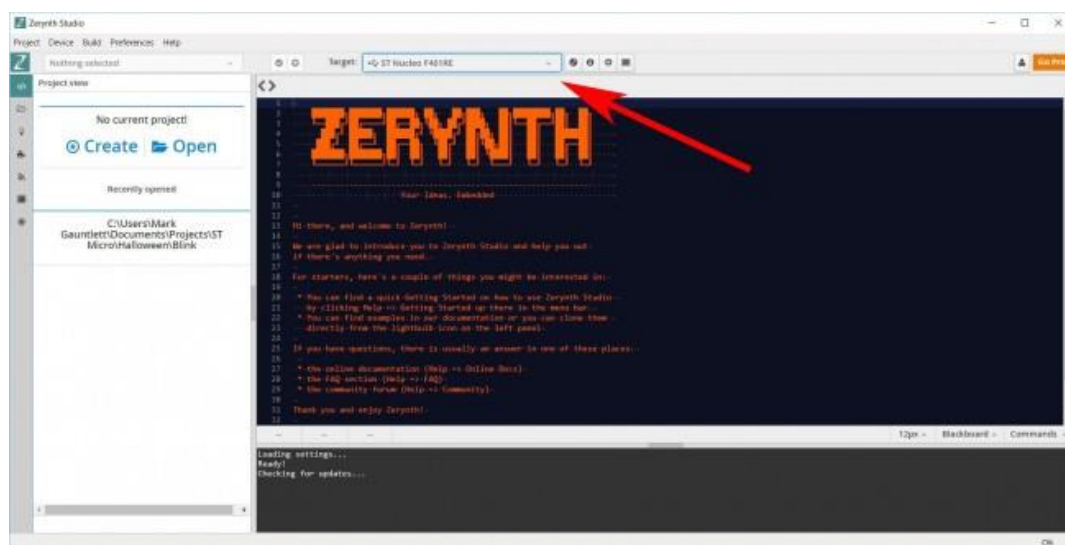
我们将下载并安装 [Zerynth Studio](#) 作为我们的 IDE，用于使用 Python 编程 ST Nucleo 板。虽然有 ‘Pro’ 版本，但免费版给了我们所有我们需要的东西，我们需要在 Arduino, Hexiware, NodeMCU, Particle, RedBear, ST Micro, Sparkfun 和 [其他一些制造商](#) 的支持下，构建相当复杂的应用程序。

[安装过程](#) 非常简单，无论您使用什么操作系统，您都可以为 Linux 下载 Windows 可执行安装程序、MacOS Disk 图像 (.dmg) 文件或 or a ‘./zerynth’ 脚本(在 “tar.xz” 存档)。

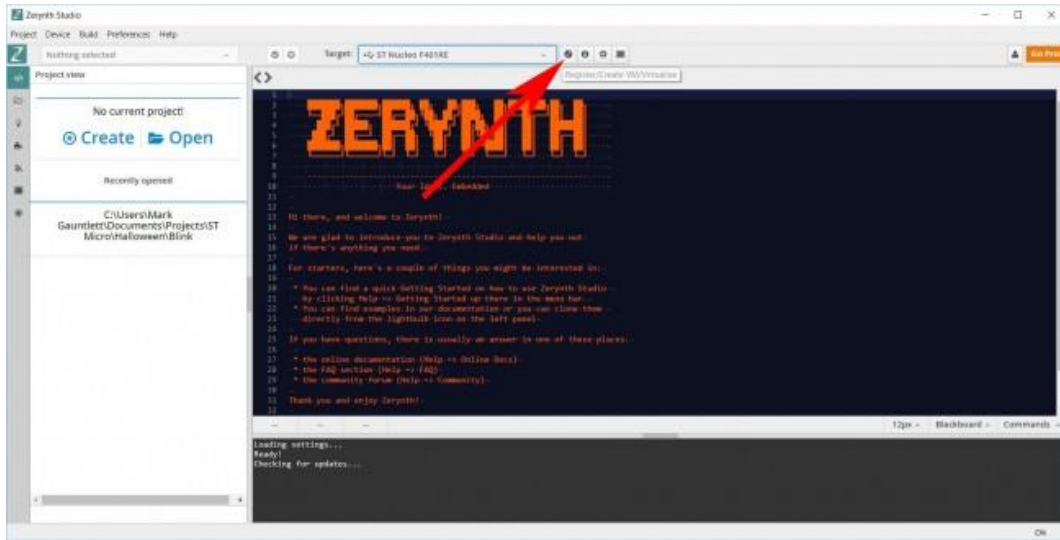
一旦安装并运行，您将会得到一个与此不太类似的接口：



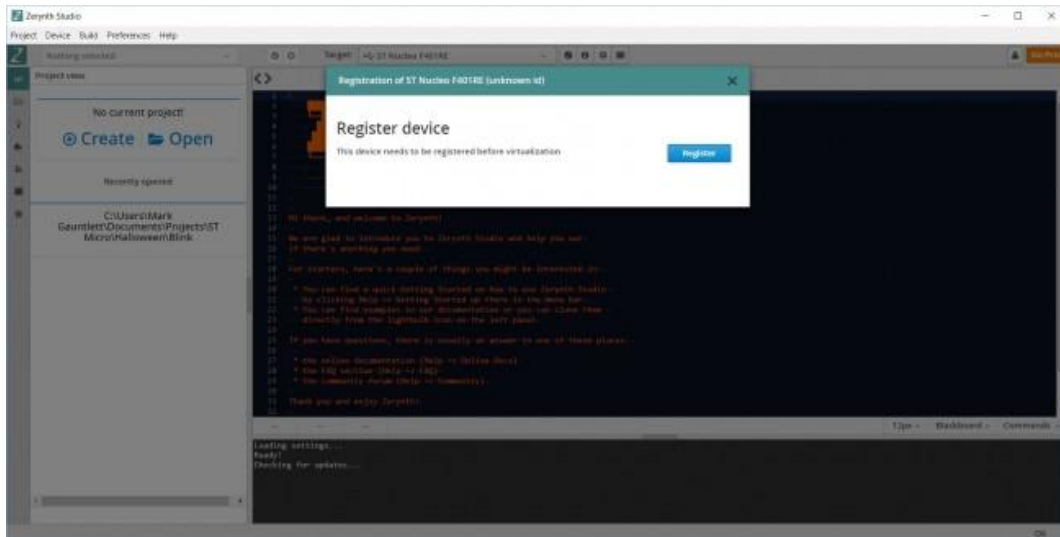
如果你现在将你的 ST Nucleo 连接到电脑上的一个 USB 端口(或者你已经连接了), Zerynth 将会识别 ST Nucleo, 你可以在设备管理工具栏上的 ‘target’ 框中选择它：



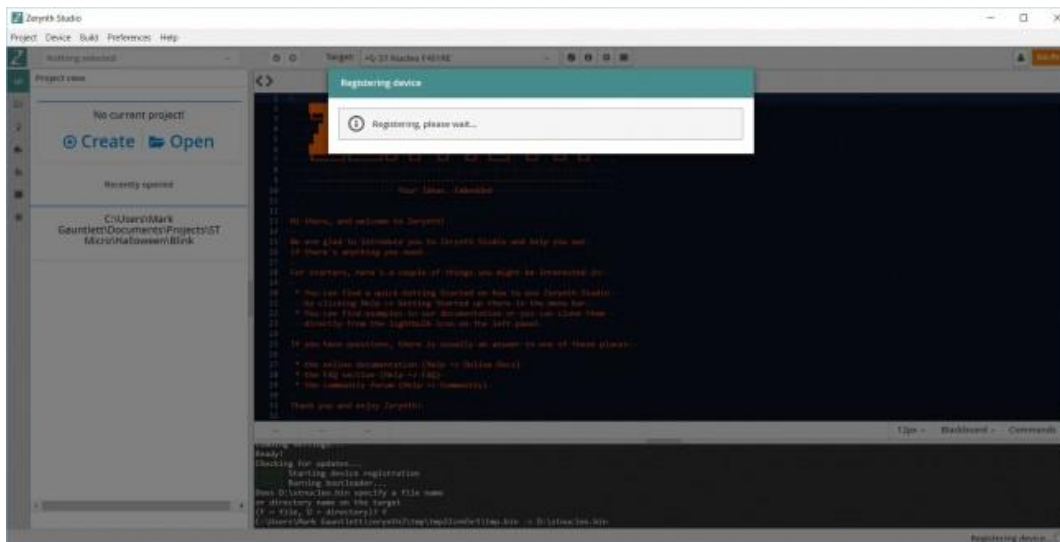
接下来，我们要使用 DMT 上的 [Z] 按钮注册该设备：



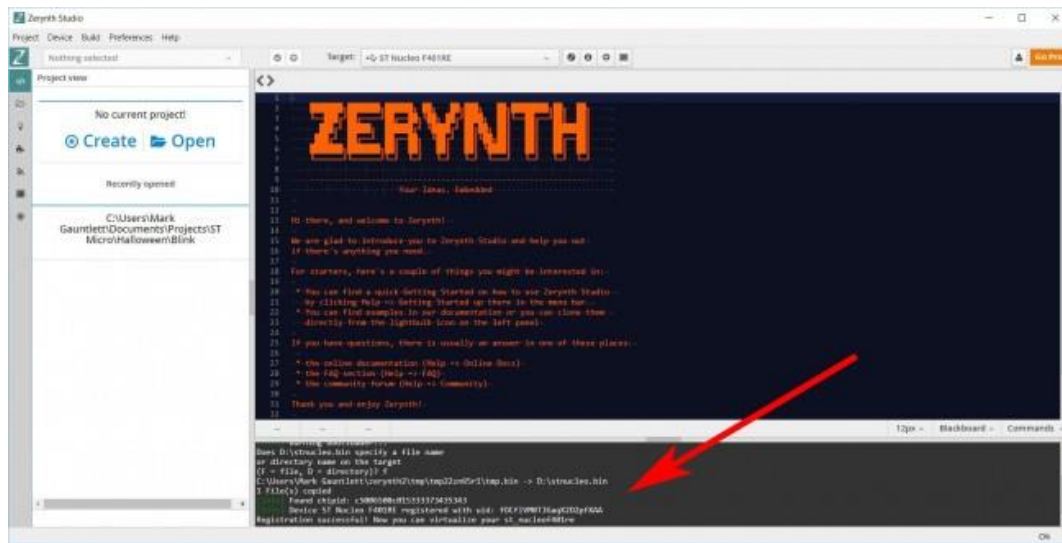
点击[Register] 按钮:



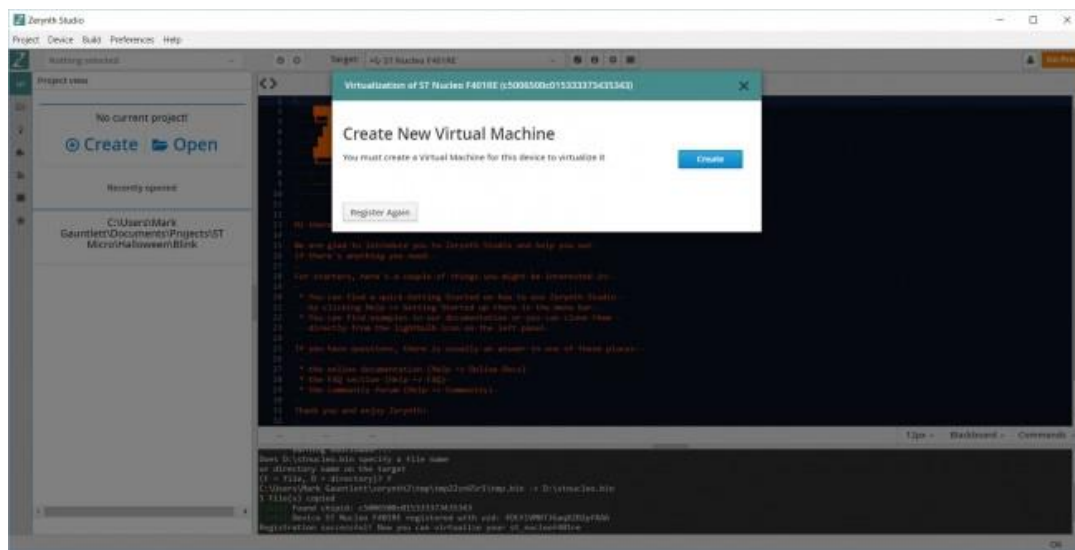
注册设备窗口向我们展示正在注册设备:



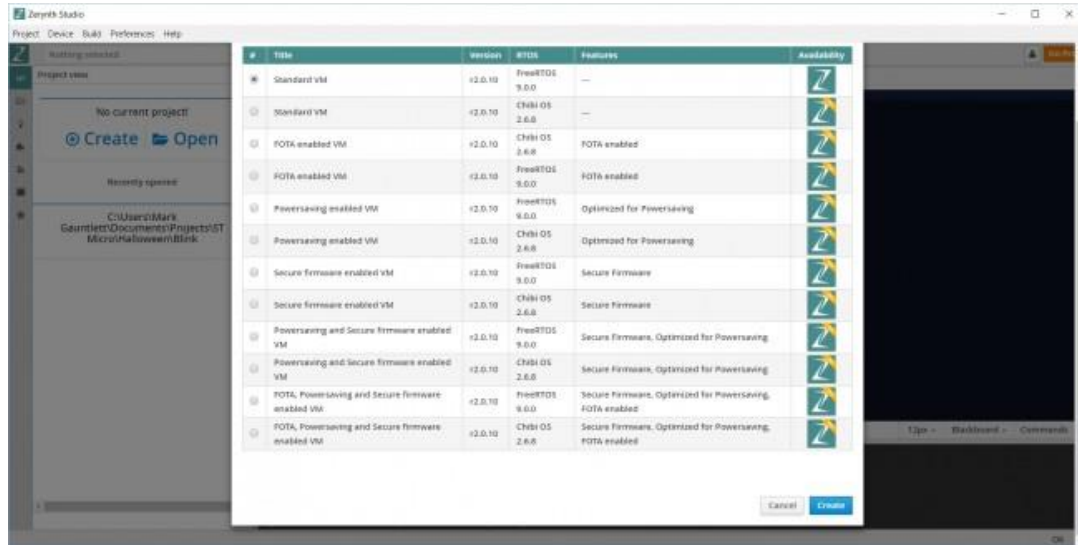
如果成功注册了你的板，你可以看见如下所示的东西：



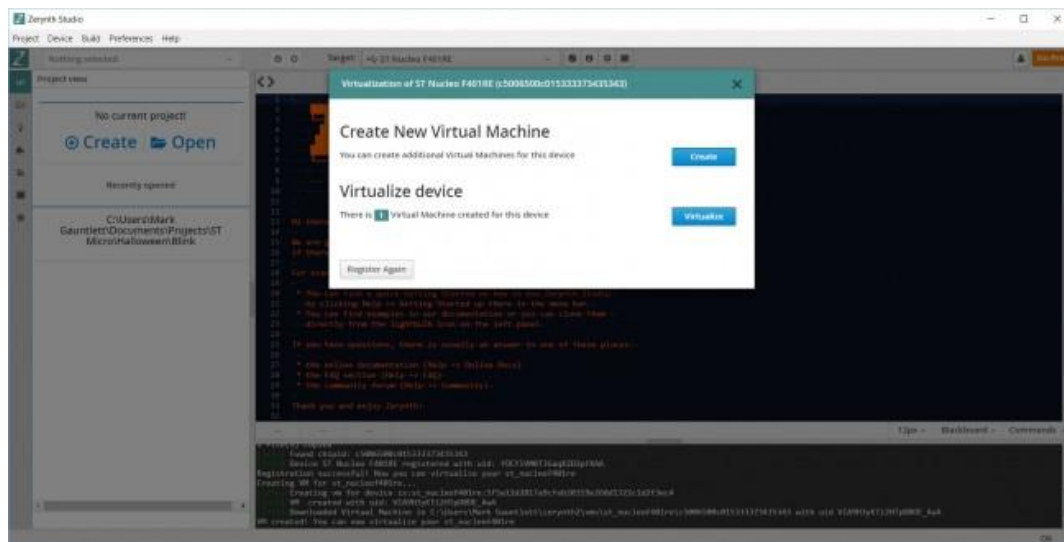
再次点击[Z]按钮来创建为了运行代码的虚拟机器。此时会跳出一个窗口，其上有需要您点击的[Create]按钮：



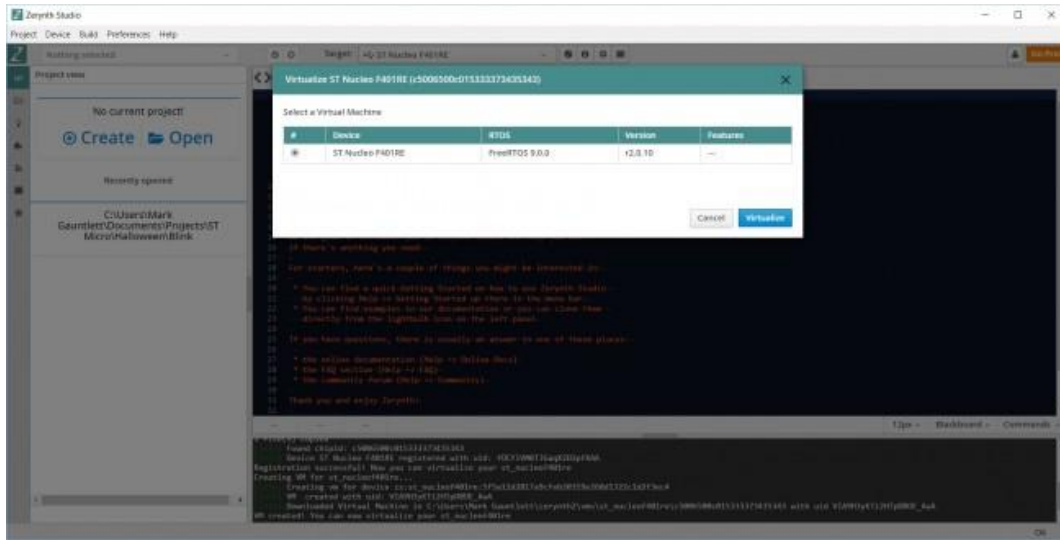
确保选择了‘Standard VM’随机按钮，点击[Create]：



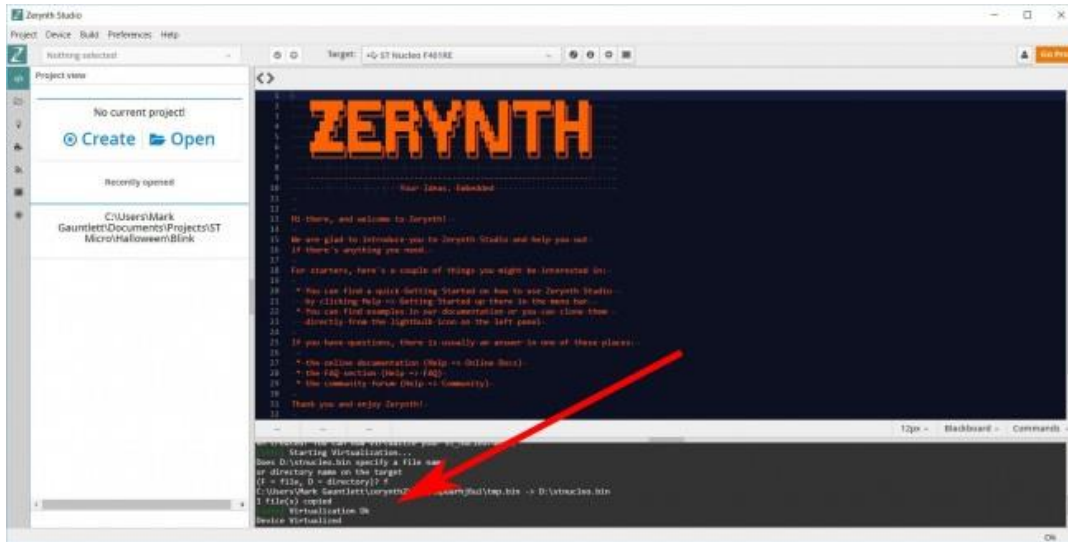
然后点击[Virtualize]按钮:



继续点击[Virtualize]按钮:



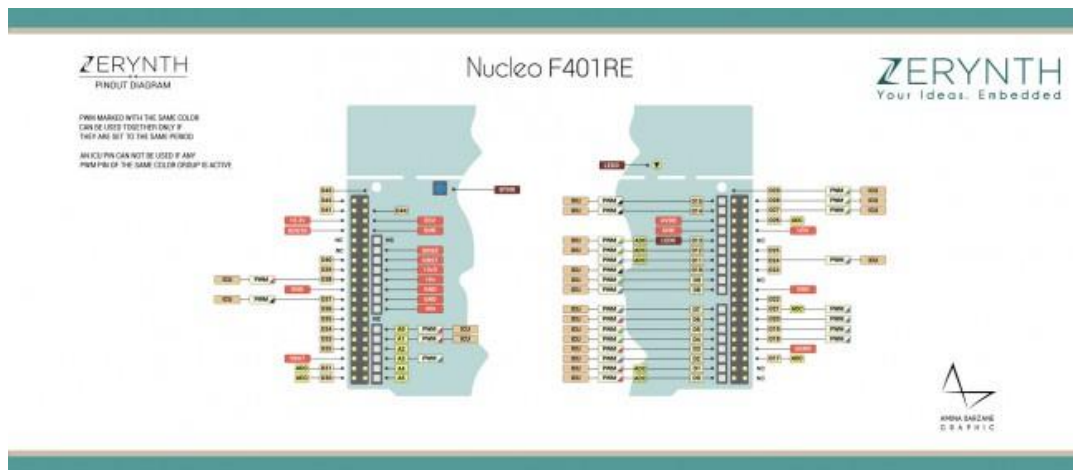
在主机 PC 处理后，你的设备就实现了其虚拟化：



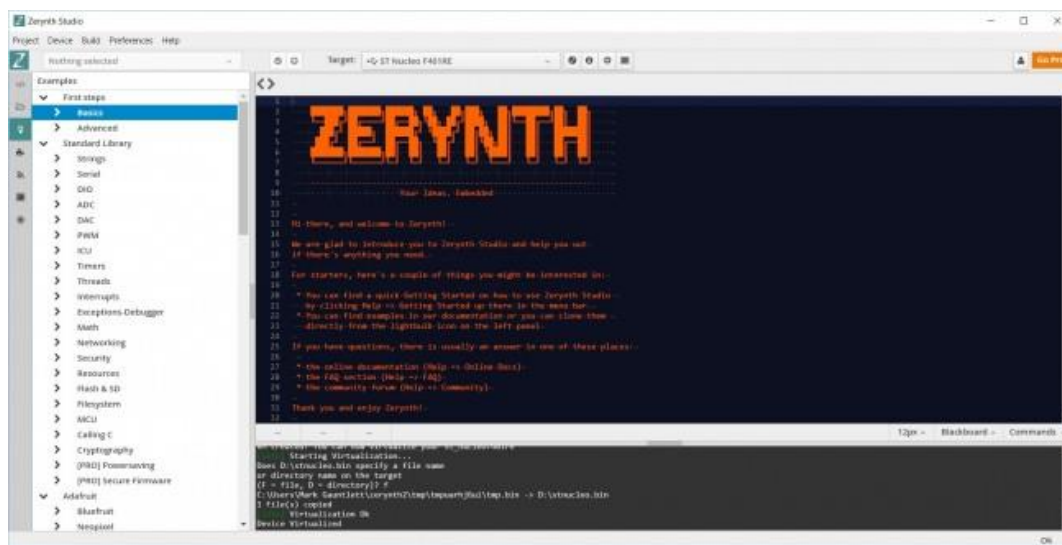
经过这些操作后，我们就能用 Python 编辑我们的板子了

用 Python 编程

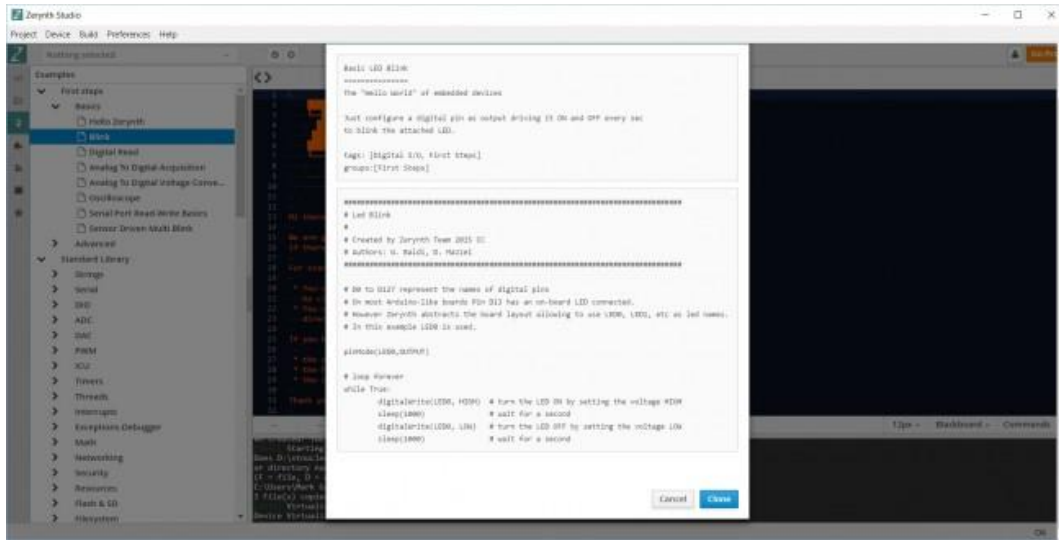
如果您是嵌入式 Python 编程的新手，那么您将会喜欢下一部分，因为 Zerynth 会给您一个巨大的开端。首先，是相当有吸引力的 I / O pin map——在得出我们的程序所需的 pin 名称时总是有用的：



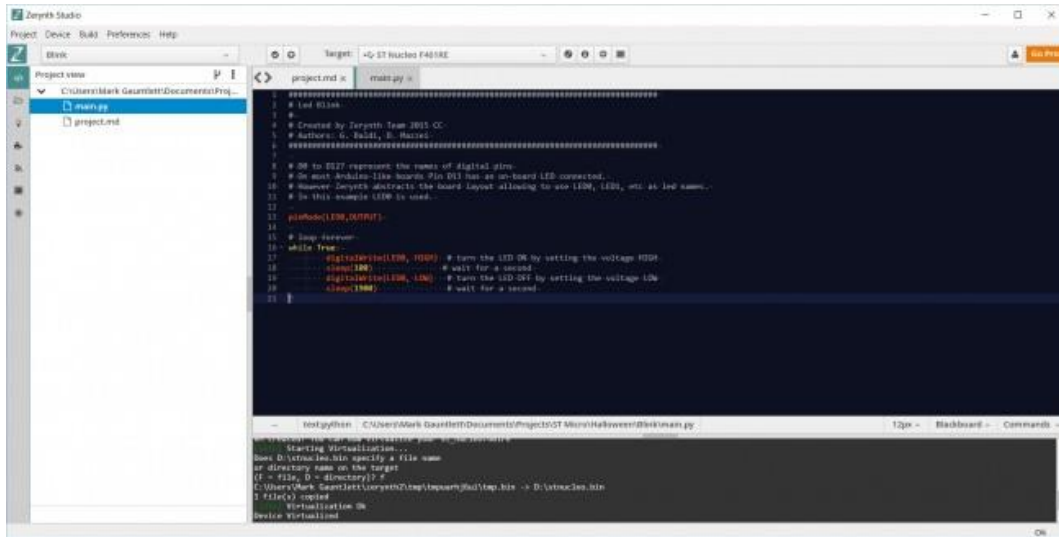
但更重要的是，如果你向下看 IDE 窗口左边的图标，你会看到一个灯泡形状的图标，这是示例按钮。单击这个，您就会看到一个 Python 示例代码的列表，它包含了最初的步骤到使用 Zerynth 库的硬件交互。当我们希望得到基本的“hello world” blink 应用程序并运行时，单击 ‘Basics’ 旁边的箭头来列出介绍的例子：



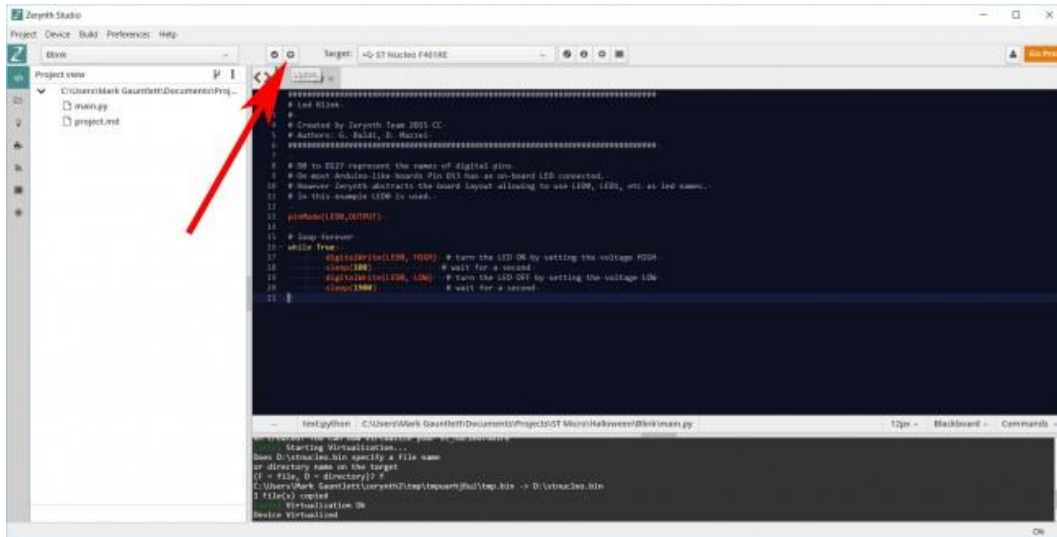
然后单击“Blink”来弹出示例代码框。通过单击[Clone]按钮，我们可以将该代码放入编程窗口中：



因此，我们觉得我们实际上已经在这里做了一些工作，让我们编辑自己的例子，并给我们的 LED 一个短期的任务周期：

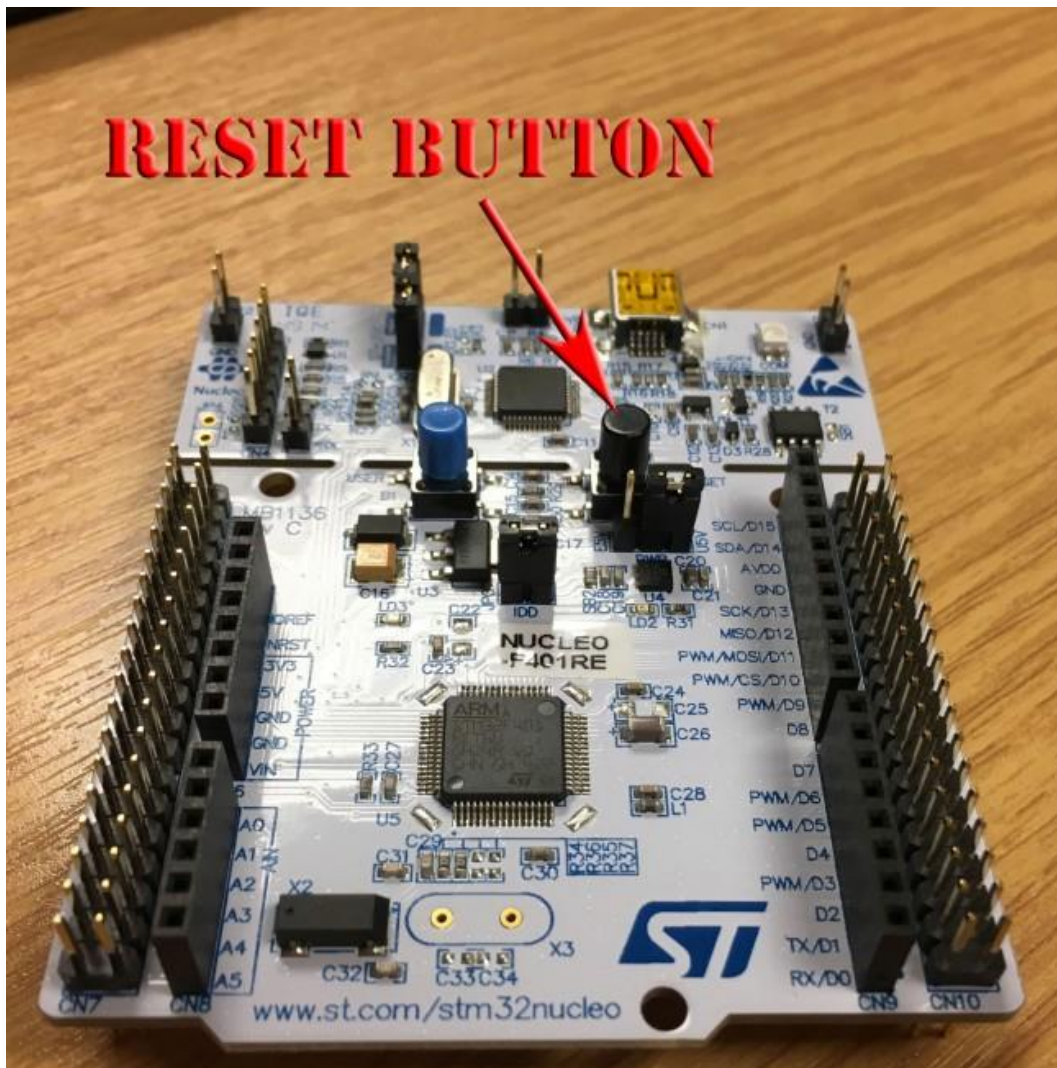


我们用 ‘upload’ 按钮就可以把这段代码放到我们的设备上：

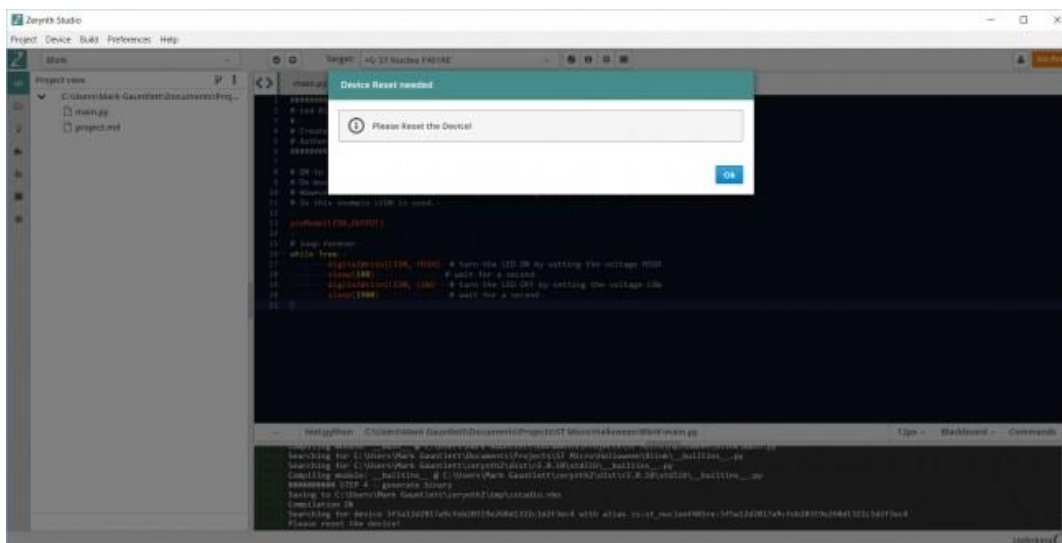


注意，如果你只是想验证你的代码，而不上传它，你可以使用 `uplink` 按钮左边的按钮。

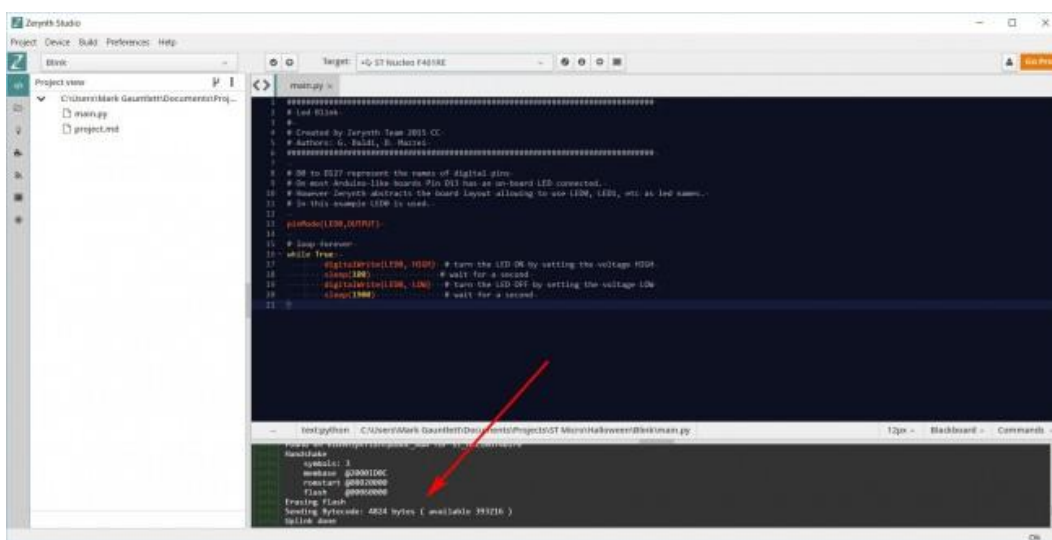
现在, 提个建议: 保持你的手指放在 Nucleo 的复位按钮上...



…由于在 uplink 过程中,你会被要求重置你的板, 在 uplink 超时前你不会拥有大量的时间:



现在, 如果一切都按计划进行, 我们将在消息窗口中得到“uplink done”的消息:



在我们的 Nucleo 板上也应该有一个闪烁的 LED。

总结

ST Micro Nucleo 板是可靠的单片机开发工具包, 它由许多传统的嵌入式工具支持, 包括 IAR EWARM, Keil MDK-ARM, mbed 和 GCC / LLVM。

然而, 我喜欢 Zerynth 在这里做的事情。一个固定的开始, 它获得代码所需的时间是最小的。这使得在开发工具包中尝试和修改变得非常容易。

Zerynth 还在文档、例子和编程指南上投入了大量精力，这样你就可以通过使用它来学习 Python，而不仅仅是通过 web 脚本。从这其中吸取经验，你可以学到一些值得你花时间去学习的新技巧。编码快乐！