Flasher.js + Espruino 使用 JavaScript 物联网应用

Phodal Huang

October 24, 2017

目录

原文链接:https://www.wandianshenme.com/play/use-flasher-espruino-build-full-javascript-iot-app

去年，我做了几个关于物联网和函数式响应编程的几个文章及会议，我总是演示我的小 Arduino 光电敏感器，它是一个用于演示他们如何协作的很好的例子。到目前为止，我在 IoTReX 的所有工作就是一直使用 johnny-five，它在运行 JavaScript 的计算机和运行 Firmata 固件的设备之间建立了客户端主机关系。我使用一个 Arduino Uno 和一个 Feather Huzzah 来无线读取温度数据。

但是，如果设备可以在 WiFi 上通讯，那么就不需要依赖于在主机上运行 JavaScript，就因会变得更加美好。换句话说，该设备将不再是被指示的被动接收者，它将自行运行一个仅向客户端发送数据的 JavaScript 程序（就像您可以在 C 中执行的那样 - 另一个几十年前的新概念!）。开发板将作为 “后端”，就像使用 johnny-five 的项目一样，但它将独立运行。

我的最终目标是，使用一个运动有 JavaScript 解释器固件的 Espruino 开发板来更换 Firmata，以便于 ESP8266 可以通过 WiFi 将模拟传感器读数，传送到 IoTReX 技术栈（stack）的 rest 部分。然后，前端可以像以前一样使用 RxJS 读取和操作传感器数据。这意味着从应用的顶层一直到底层完成是 JavaScript，甚至在设备本身！这太棒了!

事实证明，这是一个崇高的目标，在尖端（bleeding edge）的技术里蹒跚而行，所以我们将采取一步步的步骤。希望这个逐渐学习的过程，也将让您对当前现代 JavaScript 开发物联网的现状有所了解。

说到这里，这是一个有趣的附注。现在，实际上有一个更大、更糟糕的 ESP8266 表亲，称为 ESP32。

这是不是很漂亮吗? 它内置了蓝牙和 WiFi、大量的内存、大量的输入、几个内置的传感器和更快的处理器。不幸的是，它是如此的新，甚至还没有 Arduino Core 准备好开发，更不用说像 Espruino 这样的东西。所以，除非你使用起 C 语言很舒服，才不担心这个。因此，我们将坚定的使用我们的老朋友 ESP8266，最常见的迭代是 Huzzah。

步骤 0：等等，备份 - 我们该怎么做?

我们的第一步是了解一些所处领域的背景，以及我们所使用的工具。第二步将是，让 Espruino 在设备上运行。

我们将使用 Andrew Chalkley 和 Craig Dennis 在波特兰开发的新的 thingsSDK 工具链。到目前为止，开发一种完全使用 JavaScritp 的光敏电阻，会带来了很多挑战:

- 由于它不是一个正式支持的开发板，手动在 ESP8266 上使用 Espruino 闪烁 LED 是一个非常漫长的过程，特别是如果您不熟练使用硬件，并且对 C. Flasher.js 简

单而优雅地解决了这个问题更如何上手，这就是本文的主要内容。

- ESP8266 只有 4MB 的存储空间，所以任何代码和依赖都需要有选择地入进去。这一点与 johnny-five 不同，在我们的指尖上没有部分完成接触 Node.js 的全宇宙，因为我们实际上运行在设备上，而不是在我们的电脑上。幸运的是，SDK 可以帮助我们解决这个问题（减少冗余代码），其使用了一个名叫 tree-shaking 的算法。我们将在下一篇文章中介绍。
- ESP8266 世界并不像 Arduino 世界那么发达。它仍然具有高风险及古怪，就像开发板本身一样。这可能意味着，不管是在硬件还是在软件层面上，它都会改变我们的计划。例如，我希望像使用 Uno 一样使用光敏电阻，但是如果 ESP8266 ADC 对此比较挑剔，我们可能最终不得不切换到其他类型的传感器。什么是 ADC？ADC 表示 “模拟到数字转换器” - ADC 将模拟输入信号转换为数字输出值。到目前为止，在我的实验及我的好朋友经验中，ESP8266 ADC 在 Espruino 相当的粗糙。这可能会阻碍我们。
- 就像硬件方面，Espruino、thingsSDK 和周边的软件工具，都是流行的新技术和只有少数人接触的领域。它们可能会随时发生变化。这是我为什么写这个东西的一个重要原因，实际上 - 我想向其他开发人员展示这个领域发生了什么，现在就使用这些工具呢。人们使用越多，贡献越多，他们就会变得越来越好。

好吧，让我们来看几个定义，然后开发搭建。

步骤 **0**：定义

什么是 **Espruino**?

Espruino 是由 Gordon Williams 创建的用于解释 JavaScript 的微控制器的固件。它是用 C 写的，专为只有 128kB Flash 和 8kB RAM 的设备而设计！其有正式支持的开发板，如 Espruino Pico，以及非官方的开发板，如 ESP8266 芯片。

使用 Espruino，您可以引入基本的依赖关系，运行脚本，并直接与板上的 REPL 进行交互。Espruino 还包含了用于与板上的引脚进行交互的 API，以及与它们相连的任何连接，这与 Arduino 类似。这是一个非常令人印象深刻的软件，并且是完全开源的。

Espruino 的默认开发环境是一个基于 Web 的 IDE，您可以在 Chrome 中运行它，而这很好，但是并非在所有的情况下都是理想的选择。这就是 SDK 进来的地方。

什么是 **thingsSDK**?

ThingsSDK 是 Andrew Chalkley 和 Craig Dennis 为 JavaScript 构建的一个合理、专业的开发人员工具链的工具。其他人（包括你真正的）也开始投入到工具中。目前的工具链包括：

- Flasher.js，一个 Electron 编写的桌面程序，用于将 Espruino 或其他固件二进制文件烧录到兼容的开发板上。
- thingssdk-cli，用于生成使用 Espruino 的新应用程序的 CLI 工具

CLI 生成的项目有三个不同的部分：

- 中间件（称为 “strategies”，策略），用于转译您要运行的 JavaScript（目前仅适用于 Espruino，但将来会扩展到其它开发板）。Strategies 还使用了一个名为 Rollup 的 JS 工具，其用于打包项目所需的模块，方便节省空间。
- thingssdk-deployer，它通过串口与板对话，并将脚本上传到上面。
- 模块，它们是包装在 ES6 中的外围设备的驱动程序，可以包含在您的 thingsSDK 项目中。Andrew 刚刚发布了第一个，它是 Adafruit HT16K33 矩阵的驱动。

还有其他一些计划未来的事情，如 WiFi 配置工具。你可以在 thingsSDK 的官网上看到：http://thingssdk.com/。

所以，这个想法是，在用 Flasher.js 烧录 Espruino 之后，您可以使用 CLI 生成一个新项目，编写代码以执行所需的操作（如从传感器读取或连接到服务器），然后执行 `npm run deploy` 部署到开发板上。我不知道你的想法是怎样的，但这听起来比现在甚至大多数前端构建过程更容易。

步骤 **1**：下载和安装 **Flasher.js**

因此，我们所需要做的有这么三步：

- 下载和安装 Flasher.js
- 安装驱动
- 烧录到设备

首先，转到 Flasher.js 发行版本页面：https://github.com/thingsSDK/flasher.js/releases，并依据你的操作系统选择最新的操作。Linux，Mac 和 Windows 都提供二进制文件（32 位和 64 位）。你也可以自己编译源码。

在 Windows 上，您将直接下载该二进制文件，因此您可以将其移动到想要的位置。

在 Mac 上，您将下载磁盘映像（.dmg）文件。双击它并将 Flasher.js 图标拖动到应用程序文件夹。

如果您使用的是 Linux，则将下载 Debian 软件包（.deb）。双击它进行安装。

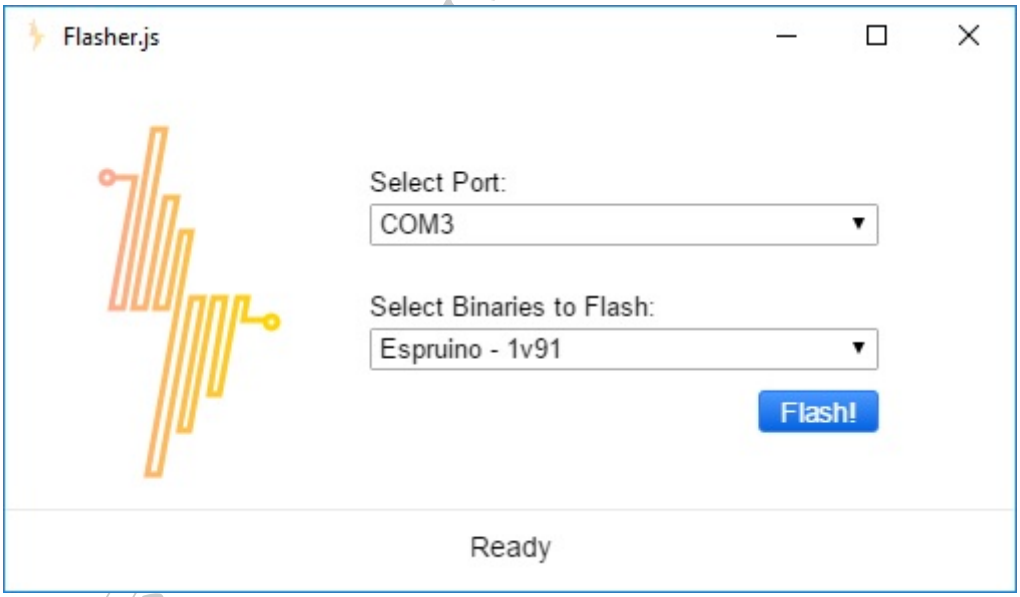
步骤 2：安装驱动

在打开 Flasher.js 之前，你 tqve 有可能需要为您的电路板安装一些驱动程序。如果您使用的是 Huzzah，则需要 Silicon Labs 的这些驱动程序：https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx。如果您使用的是 NodeMCU V3，则需要使用这些适用于 Mac 的 CH340G 驱动程序：https://www.wemos.cc/tutorial/get-started-nodemcu.html。

如果您使用的是 El Capitan，并遇到一些麻烦，请查看此解决方法。我们还将所有这些信息保存在 Flasher.js 指南中的兼容设备图表中。

步骤 3：烧录到设备

好的，这是有趣的部分。插入设备，然后双击新安装的 Flasher.js 二进制文件。你会看到一个这样的屏幕（对你的操作系统有微小的差异）：



Flasher.js 将尝试为您自动选择端口，但您需要仔细检查它。在 Windows 上，这些将看起来像 COM3，而在 Mac 和 Linux 上，它们将更像是 `/dev/ttyUSB1` 或 `/dev/cu.usbmodem150` 。

在第二个 dropdrown 中，Flasher.js 带有许多不同的二进制选项。除了 Espruino，您可以使用此工具来烧录 Mongoose OS（以前的 Smart.js），thingsSwitch 甚至是 MicroPython。为了我们完成我们的目标，您可以将其留在最新版本的 Espruino 上，它是默认的。

现在 - 您一直在等待的那一刻 - 点击 Flash！按钮。Flasher.js 将为您下载适当的二进制文件，准备并闪存您的设备 - 全部在一个步骤！

当它闪烁时，你应该看到你的 LED 闪烁。

这一步应该只需要一两分钟，然后，瞧！您将全部开始使用其他的 SDK 工具链来构建项目。

步骤 4：下一步？

一旦您完成在你的 Espruino 的设备上闪烁 LED，您将有两种不同的选择。

首先，您可以从 Chrome App Store 查找 Espruino Web IDE，并将其设置为与您的设备配合使用。Espruino 网站有关于如何做这个非常有用的教程：http://www.espruino.com/Web+IDE。

其次，您可以继续前进并尝试使用 SDK CLI，从而获得额外的经验。thingsSDK 指南有几个教程可以帮助您在 ES6 中快速启动和运行。我将在下一篇文章中更深入地介绍使用 CLI 创建项目。

无论你采取哪种路径，推荐一下你所创造的项目，所以我们都可以看到，并庆祝它。我们总是在本地的 JSoT 聚会上说，在这个领域，字面上没有一个项目太小了。闪烁的光线始终是令人高兴的原因，但特别是 JavaScript 在 2 美元 ESP8266 芯片上运行，并闪烁时！

原文连接：http://www.samjulien.com/flasherjs/

原文链接：https://www.wandianshenme.com/play/use-flasher-espruino-build-full-javascript-iot-app