

完全使用 **JavaScript** 开发物联网：  
**Arduino + Johnny-Five +**  
**WebSockets**

Phodal Huang

October 24, 2017

## 目录

步骤 0: 知识准备 .....	3
等等。。。 Arduino 和 Node.js? .....	3
这是 Johnny-Five! .....	3
你会构建什么? .....	4
硬件 .....	4
步骤 1: 为 Johnny-Five 腾出空间 .....	5
步骤 2: 给 Johnny 一盏灯 .....	5
步骤 3: 控制色彩 .....	6
创建 Web 服务器 .....	6
打碎它 .....	8
步骤 4: 添加客户端交互 .....	9
步骤 5: 点火 .....	10
Awesome. What next? .....	11

原文链接:<https://www.wandianshenme.com/play/use-javascript-build-iot-projects-with-arduino-j>

我最近开始玩 **Arduino Starterkit**。

从零开始了解硬件组件、电子元件以及如何编写这个物理世界的代码，我可以坦白地说，我非常惊讶，它是如此的容易，并且触手可及。在十分钟的时间里，我从零开始就完成了我的机器人技术世界（一个眨眼的 **LED** 灯），这种速度相当的惊人，并且没有一点痛苦。

但是，当 **JavaScript** 和 **node.js** 进入电子世界时，真正的乐趣开始了！

步骤 0: 知识准备

等等。。。 **Arduino** 和 **Node.js** ?

您可能会认为，你只能使用 **C/C++** 的子集对 **Arduino** 进行编程，因为它是硬件环境的默认编程语言。大多数 **Arduino** 入门套件，都将为您提供带有代码的示例项目。

虽然这样做没有任何问题（我的启动项目都是 **C** 语言开始的，并从中学习），但是使用 **JavaScript** 的强大功能（并且扩展了 **Web** 的强大功能!）的可能性，使您能够更有创造性，特别是如果你拥有 **Web** 开发背景。

另外，你可以提高一下你的 **node.js** 技能。□

这是 **Johnny-Five!**

**Johnny-Five** 是一个 **JavaScript** 机器人编程框架。它使您可以使用 **JavaScript** 从 **Arduino** 开发板读取和写入数据。它是开源的，并且具有类似于 **jQuery** 的非当简单的 **API**，因此，它看起来是非常熟悉的。

看看下面的代码片段，它是用来改变 **RGB** 的颜色：

```
1 led.color({
2   red: 255,
3   blue: 0,
4   green: 0
5 });
```

即使你以前从来没有编程过，我也会认为这是一个非常简单的方式——来宣布你想要对 **LED** 灯改变的内容。

你会构建什么?

以下是我们将要制作的完整视频: 我在这里使用 iPad 来控制灯光的颜色, 但显然您可以使用手机进行操作:

视频地址: <https://www.youtube.com/embed/nLPQNb3Q3B8>

材料清单 我们所需要的硬件有:

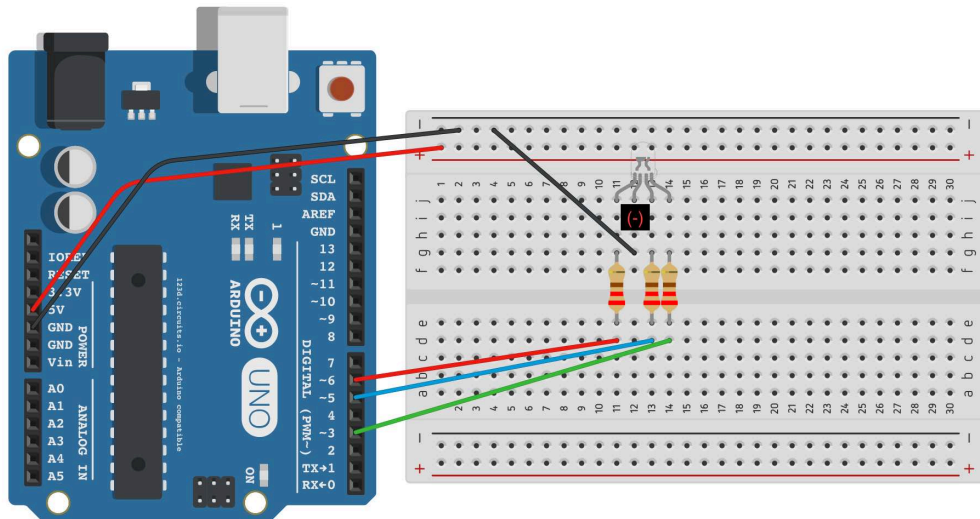
- Arduino 板
- 1 RGB LED
- 1 面包板
- 跳线
- 3 个电阻 (220Ω)

再次, 如果您只想复制并粘贴代码, 它可以在 Github 找到: [Arduino RGB LED](#)

硬件

请注意, 该 RGB LED 有 4 个引脚, 而不是普通 LED 的 2 个。事实上, RGB LED 可以像这个名字本身所暗示的那样, 同时产生 3 种颜色: 红色, 绿色和蓝色。较长的腿被称为阴极, 你将使用它连接到接地 (-)。

以下是它的原理图:



我不会详细说明, 为什么我们将这些连接到 PCW 插槽。因为我确信, 有人会比我更好地解释这个问题。简而言之, PCW 连接器使我们能够处理不仅仅是 0 和 1 (对于开

和关状态有用) 的值, 因为我们希够控制值从 0 到 255 变化, 以此来变幻出每种颜色。

### 步骤 1: 为 **Johnny-Five** 腾出空间

所以, 让我们准备你的 **Arduino** 用于 **JavaScript**! 幸运的是, 你只需要做一次。

1) 将 **Arduino** 与您的计算机连接起来, 就像使用提供的 **Arduino IDE** 编辑器编写代码时一样。

2) 在 **Arduino IDE** 上, 选择 **Tools > Port**, 确保选择了 “**Arduino Uno**”。

3) 仍然在这个 **IDE** 上, 选择 **File > Examples > Firmata > StandardFirmata**, 然后单击 **Upload** 按钮。

4) 就是这样! 你的 **Arduino** 现在可以读取 **johnny-five** 将要调解的信息。现在, 您可以关闭 **Arduino IDE**, 并启动您最喜欢的代码编辑器。

疑难解答: 实际上, 我无法使用运行 **Arduino IDE 1.6.8** 版本上传 **Firmata**。显然这是一个已知的 **bug**, 更新到 **1.6.9** 版本后, 一切顺利。

### 步骤 2: 给 **Johnny** 一盏灯

由于, 我们将使用 **node.js** 与 **Arduino** 进行通信, 因此请务必先将其安装在机器上。然后在您的计算机上的某个地方, 创建一个新文件夹, 并创建一个名为 **app.js** 的文件。同时, 在此文件夹上安装 **johnny-five**, 方法是在您的终端上运行此目录:

```
1 npm install johnny-five
```

稍后, 我们需要安装其他软件包, 如 **express** 用来创建本地服务器, 以便我们可以通过服务器交互。到目录为此, 这真的是您需要与 **Arduino** 本身进行通信的所有工具。

要确保它正常工作, 您可以运行以下脚本:

```
1 var five = require ("johnny-five") ;
2 var myBoard
3
4 myBoard = new five.Board () ;
5
6 myBoard.on ("ready", function () {
7   console.log ('Arduino is ready! ');
8 });
```

在您的终端上, 运行 `node app.js`。你应该看到 “**Arduino is ready**”。

### 步骤 3: 控制色彩

有诸如多种方式可以控制 RGB 灯的各种颜色。Starter Arduino Kit 中的一个项目使用光传感器读取环境光照值，并相应地更改颜色。您还可以根据温度或任何其他外部输入进行操作。

在这种情况下，我们有兴趣构建一个 Web 界面，以便于我们直接操作这些值。这就是为什么我们要使用 web sockets，也就是一个整齐的小库，名为 socket.io。

这样做的好处是实时效果：它可以让您将 Arduino 连接到计算机，但能够控制来自指向服务器的手机或 iPad 的输入。

通过广播所修改的事件，任何连接到服务器的人，也将看到与您所做的相同的值（有点类似于聊天应用）。

#### 创建 Web 服务器

所以我们要构建一个本地的 Web 服务器。由于我们已经在使用 Node.js，所以我们简单地部署一个快速简单的快速服务器，它将托管代码并为我们提供静态客户端文件。

为此我们需要先安装 express 和 socket.io 这两个库：

```
1 npm install express socket.io
```

然后到你的 app.js 文件夹中，输入以下代码：

```
1 'use strict';
2
3 const five = require('johnny-five');
4 const express = require('express');
5 const app = express();
6 const server = require('http').createServer(app);
7 const io = require('socket.io')(server);
8
9 let led = null;
10
11 app.use(express.static(__dirname + '/public'));
12 app.get('/', function(req, res, next) {
13   res.sendFile(__dirname + '/index.html')
14 });
15
```

```
16 five.Board().on('ready', function() {
17   console.log('Arduino is ready.');
```

```
18   // Initial state for the LED light
19   let state = {
20     red: 1, green: 1, blue: 1
21   };
22
23   // Map pins to digital inputs on the board
24   led = new five.Led.RGB({
25     pins: {
26       red: 6,
27       green: 3,
28       blue: 5
29     }
30   });
31
32   // Helper function to set the colors
33   let setStateColor = function(state) {
34     led.color({
35       red: state.red,
36       blue: state.blue,
37       green: state.green
38     });
39   };
40
41   // Listen to the web socket connection
42   io.on('connection', function(client) {
43     client.on('join', function(handshake) {
44       console.log(handshake);
45     });
46
47     // Set initial state
48     setStateColor(state);
49
50     // Every time a 'rgb' event is sent, listen to it and grab its new values
    for each individual colour
```

```
51   client.on('rgb', function(data) {
52     state.red = data.color === 'red' ? data.value : state.red;      state.green
      = data.color === 'green' ? data.value : state.green;
53     state.blue = data.color === 'blue' ? data.value : state.blue;
54
55     // Set the new colors
56     setStateColor(state);
57
58     client.emit('rgb', data);
59     client.broadcast.emit('rgb', data);
60   });
61
62   // Turn on the RGB LED
63   led.on();
64 });
65 });
66
67 const port = process.env.PORT || 3000;
68
69 server.listen(port);
70 console.log(`Server listening on http://localhost:${port}`);
```

打碎它

上面一次给了很多的代码，所以让我们逐一分解它们。第一部分，简单地设置 **express**，并在 **3000** 端口上，提供本地文件（**HTML** 和 **JavaScript** 客户端）的服务。

第二部分是所有的 **Arduino** 魔法。一旦连接到 **Arduino**，所有的代码都包含在 `five.Board().on('ready', function(){...})` 中。如果你拥有 Web 开发背景，你可以认为是 `document.ready`。

## SOCKETS

每当 **socket.io** 接收到一个修改事件时，输入的值（我们将在下一个位置得到）将作为数据参数发送，如下所示：

```
1 client.on('rgb', function(data) {...})
```

事件名，当然可以是你想要的任何东西。我将其命名为 **rgb**，因为，好吧，这就是



我们正在处理的东西。

#### 步骤 4: 添加客户端交互

HTML 非常的简单:

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <title>Arduino</title>
5   </head>
6   <body>
7     <h3>Red:</h3>
8     <input id="red" type="range" min="0" max="255" step="1" value="0">
9
10    <h3>Green:</h3>
11    <input id="green" type="range" min="0" max="255" step="1" value="0">
12
13    <h3>Blue:</h3>
14    <input id="blue" type="range" min="0" max="255" step="1" value="0">
15
16    <!-- include the client socket library-->
17    <script src="/socket.io/socket.io.js"></script>
18
19    <script src="./client.js"></script>
20  </body>
21 </html>
```

我们有 3 个滑块，每个颜色一个。如果你想添加一个 CSS 风格的破折号，只需将它从 Github 项目中复制。请注意一下，我们是如何关联 socket.io 的脚本。

最后，就是所有的客户端交互代码。在每个滑块的每个更改事件上，我们发送一个新事件到 socket.io，其中包含一些数据：哪个颜色已更改，以及其新值。

```
1 // client.js
2
3 (function() {
4   var socket = io.connect(window.location.hostname + ':' + 3000);
5   var red = document.getElementById('red');
```

```
6   var green = document.getElementById('green');
7   var blue = document.getElementById('blue');
8   function emitValue(color, e) {
9       socket.emit('rgb', {
10          color: color,
11          value: e.target.value
12      });
13  }
14
15  red.addEventListener('change', emitValue.bind(null, 'red'));
16  blue.addEventListener('change', emitValue.bind(null, 'blue'));
17  green.addEventListener('change', emitValue.bind(null, 'green'));
18
19  socket.on('connect', function(data) {
20      socket.emit('join', 'Client is connected!');
21  });
22
23  socket.on('rgb', function(data) {
24      var color = data.color;
25      document.getElementById(color).value = data.value;
26  });
27 }());
```

这是我们在 `app.js` 文件中监听的数据。客户端发出数据，`socket.io` 在服务器上接收数据，然后更新 `Arduino` 的值。

## 步骤 5: 点火

所以，重新回顾一下到目前为止，我们做东西：

- `app.js` 文件使用 `express` 作为本地服务器，它使用 `WebSockets` 来监听（和广播）客户端上的修改事件
- 同一个文件使用 `johnny-five`，根据从 `WebSockets` 连接接收到的数据向我们的硬件发送指令
- `index.html` 服务于发送与滑块的交互输出的 `client.js`

剩下的就是启动本地服务器。您可以在终端中键入：

```
1 node app.js
```

如果一切顺利，您应该看到成功的消息，并通过在浏览器打开 <http://localhost:8080>，您应该看到我们的页面！

如果没有，请再一次依据我们的 GitHub: [arduino-rgb-websockets](#) 修改对应的代码。

## Awesome. What next?

真正令我兴奋的是，从其中获取的无限可能性。当然，现在我们直接控制 LED 的值。但是在未来，我们可以很好地根据天气 API，Twitter 帐户，Facebook 消息来修改它们。

实际上，这些代码大部分已经可以重用了。一旦您获取要使用的数据，将其广播到您的主板，并决定如何外部化此数据。也许使用声音，也许使用液晶显示器？唯一限制你的是，亚马逊的传感器部门。

原文链接: <http://blog.ricardofilipe.com/post/getting-started-arduino-johhny-five>

原文链接: <https://www.wandianshenme.com/play/use-javascript-build-iot-projects-with-arduino-j>