

ESP8266 + Firmata + Johnny-Five 制作 NodeBot

Phodal Huang

September 8, 2017

目录

步骤 1: 配料	3
步骤 2: 一些信息	4
Johnny-Five	4
StandardFirmataWiFi	4
EtherPortClient	4
步骤 3: Arduino IDE 配置	4
步骤 4: 做出一些修改	4
步骤 5: 设置 Johnny-Five	5
安装 Johnny-Five	5
编写 JavaScript 客户端代码	6
步骤 6: It works!	7

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/esp8266-firmata-johnny-five-build-nodebot>

ESP8266 是一款支持 WiFi 功能的 SoC, 具有许多种不同的形状和形式。它可以运行使用 lua 脚本的 nodemcu 固件。我们中的大多数人只是用 ESP8266 社区自己提供的 Arduino 核心编程。

而现在, 您可以使用 Javascript 进行编程! 这是得益于 soundanalogous, jacobrosenthal 和 jnsbyr (工作于 Firmata 固件上) 和 Johnny-five 库的 rwaldron 等人的伟大工作。

而本玩法, 这只是一个关于如何使其工作的指南。

除此:

1. ESP8266 将作为 WiFi 服务器。将其配置为 WiFi 客户端的选项将在未来添加。
2. Analog Read 暂时似乎不能工作, 但是正在努力中。
3. 这是全新的, 依赖关系可能会经常更新。

在这里, 我们将使用 **StandardFirmataWiFi** 来将 **ESP8266** 作为服务器, 连接到网络, 并在连接到同一网络的主机上运行 **node.js J-5 (Johnny-Five)** 脚本, 以闪烁 **13** 引脚上的 **LED**。

本玩法, 在 **mac os X** 系统上测试完成的。

步骤 1: 配料

在本文中, 我们所需要的库有:

- 最新的 Arduino IDE (1.6.8)
- ESP8266 Core
- 最新的 Arduino Firmata 库, 能支持 ESP8266
- 最新的 Johnny-five 库
- EtherPortClient

在计算机端, 我们需要的东西有:

- 主机 (计算机, Raspberry Pi 等) 能够运行 Node.js
- 耐心
- 根据您的电路板, 您可能还需要一个 USB 串行转换器

此外, ** 我假设您有使用基于 Arduino IDE 上的 ESP8266 编程经验。

步骤 2: 一些信息

Johnny-Five

JavaScript 机器人和 IoT 编程框架, 由 Bocoup 开发。

StandardFirmataWiFi

StandardFirmataWiFi 是一个 WiFi 服务器应用程序。您将需要一个 Firmata 客户端库用来进行网络传输, 以建立与 StandardFirmataWiFi 的连接。要使用 StandardFirmataWiFi, 您需要具有以下开发板或扩展板中的一个:

- Arduino WiFi Shield
- Arduino WiFi Shield 101
- Arduino MKR1000 板 (内置 WiFi 101)
- ESP8266 WiFi 板 (与 ESP8266 Arduino Core 兼容)

EtherPortClient

是一个用于 Etherport 的客户端虚拟串行端口, 用于实现 firmata 兼容的开发板和网络共享集线器, 以便于远程实体控制电路板。

步骤 3: **Arduino IDE** 配置

1. ** 获取最新 Arduino IDE
2. 安装 ESP8266 Core (参考《使用 ESP8266 制作 WiFi 干扰器》中的步骤 2)
3. 下载最新的 Arduino Firmata 库 (通过 Sketch > Include Library > Manage LibrariesFilter 下载最新)

步骤 4: 做出一些修改

1. 打开 Arduino IDE 示例 (example) 中的 StandardFirmataWiFi 项目。
2. 它将打开 StandardFirmataWiFi.ino 文件, 以及 wifiConfig.h
3. 根据你的硬件情况, 取消注释/注释 (选项 A、B、C)。默认情况下启用选项 A。
4. 它已经为 ESP8266 配置好了。找到步骤 2:

```
1 // STEP 2 [REQUIRED for all boards and shields]
```

```
1 // replace this with your wireless network SSIDchar ssid[] =
   "your_network_name";
```

根据你的情况修改 “your_network_name” 成你的 WiFi 名称。

5. 在这个例子中, 我们使用 DHCP, 所以我们没有配置步骤 3。如果你需要使用静态 IP, 请按照 `wifiConfig.h` 文件提供的说明修改。6. 配置您的家庭网络通行证: 默认值为选项 A:

```
1 /* * OPTION A: WPA / WPA2 * * WPA is the most common network security type.
   A passphrase is required to connect to this type. * * To enable, leave
   #define WIFI_WPA_SECURITY uncommented below, set your wpa_passphrase
   value * appropriately, and do not uncomment the #define values under
   options B and C */
2
3 #define WIFI_WPA_SECURITY
4 #ifndef WIFI_WPA_SECURITY
5
6 char wpa_passphrase[] = "your_wpa_passphrase";
7 #endif //WIFI_WPA_SECURITY
```

7. 使用正确的配置烧录到你的 ESP8266 开发板中

例如, 我使用 NodeMCU 0.9 开发板。我就使用它和 FTDI-TTL 适配器来编程 MCU。您需要选择适当的端口设置。

Arduion IDE 可能会说文件是只读的, 只需保存为另一个副本

步骤 5: 设置 **Johnny-Five**

安装 **Johnny-Five**

安装 **Johnny-Five** 之前, 需要先安装 Node.js 环境 (参考: [Node.js + Rasp-i-o + Johnny-Five 控制 Raspberry Pi](#))

安装 **Johnny Five** 有两种方式

1. 代码 clone

```
1 git clone git://github.com/rwaldron/johnny-five.git && cd johnny-five
2 npm install
```

2.npm 包安装

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

除此, 我们还需要 **Etherport** 客户端库:

```
1 npm install etherport-client
```

编写 **JavaScript** 客户端代码

在 `johnny-five` 目录中创建一个 `js` 文件, 并将其命名。

然后, 粘贴下面的代码:

将第 18 行更新到 **ESP8266** 开发板的地址:

```
1 /*
2  * Update line 18 below to the ESP8266 board address
3  *
4  * Enable Serial debugging by uncommenting //#defin SERIAL_DEBUG in
   StandardFirmataWiFi
5  * (save a copy of StandardFirmataWiFi first)
6  *
7  * On startup (you may have to reset the ESP board because it starts up
   really fast
8  * view the Serial output to see the assigned IP address (if using DHCP)
9  * Or if you want to give the board a static IP (not use DHCP) then
   uncomment the
10 * following lines in wifiConfig.h and update to your chosen IP address:
11 * #define STATIC_IP_ADDRESS 10,0,0,17
12 * #define SUBNET_MASK      255,255,255,0 // REQUIRED for ESP8266_WIFI,
   ignored for others
13 * #define GATEWAY_IP_ADDRESS 0,0,0,0      // REQUIRED for ESP8266_WIFI,
   ignored for others
14 */
15 var Firmata = require("firmata").Board;
16 var EtherPortClient = require("etherport-client").EtherPortClient;
17 var board = new Firmata(new EtherPortClient({
18   host: "192.168.1.103",
19   port: 3030
```

```
20 });
21 board.on("ready", function() { console.log("READY!");
22   console.log(
23     board.firmware.name + "-" +
24     board.firmware.version.major + "." +
25     board.firmware.version.minor
26   );
27   var state = 1;
28   var lastVal = 0;
29   this.pinMode(2, this.MODES.OUTPUT);
30   setInterval(function() {
31     // blinks the blue LED on a HUZZAH ESP8266 board
32     // for other boards, wire an LED to pin 2 or change
33     // the pin number below
34     this.digitalWrite(13, (state ^= 1));
35   }.bind(this), 500);
36   // this does not seem to be working - need to look into it
37   // one other thing is ESP uses a 1V reference for analog so
38   // once this works, it will need scaling
39   this.analogRead(0, function(value) {
40     if (value != lastVal) {
41       console.log(value);
42     }
43   });
44 });
```

步骤 6: *It works!*

所以现在，在使用 `StandardFirmataWifi` 固件闪烁您的 `ESP8266` 开发板后，我们可以运行基本的 `Johnny-five` 脚本。

```
1 node "your_file_name".js
```

根据刚创建的文件名，更改命令行上的文件名。

现在，你就会看到 `ESP8266 13` 引脚上的 `LED` 闪烁。

现在你可以从这里做出令人敬畏的项目。可能性是无止境的。`Johnny-five` 已经有相

当多的例子了。

再一次。任何帮助这个令人敬畏的功能的人，都可以大肆宣传。对不起，如果没有人无意中给任何人。

原文链接: <http://www.instructables.com/id/ESP8266-Firmata-J5-NodeBot/>

原文链接: <https://www.wandianshenme.com/play/esp8266-firmata-johnny-five-build-nodebot>

玩点什么: <https://www.wandianshenme.com>