

Arduino Yun 上使用 jQuery + Bootstrap 制作简单的 Web UI 控制

Phodal Huang

September 8, 2017

目录

步骤 1: 介绍项目架构	3
步骤 2: 解释项目	4
步骤 3: 自制 LCD API 命令	5
步骤 4: Do it!	7

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/arduino-yun-bootstrap-jquery-build-simple-web>

Hello! 在我发表上一篇博客之后,最近一直很忙。但是上个周末,我终于有时间和我的 **Arduino Yun** 做了一个小项目。在那之间,它已经堆满了很多灰尘。我想做一个简单的项目来展示 **REST API** 和前端应用程序的概念,以及它们之间如何进行交互。

为什么我选择 **Arduino Yun** 来做这个项目?它是这种项目的理想工具,因为在 **RESTful** 模式下,很容易就能设置 **Arduino** 的开发板引脚(您只需选择 **IDE** 中的 **Bridge** 示例)。并且很容易为前端应用程序 (**HML/JS/CSS**) 提供服务,您只需将该站点放在 `/root/path_to/www/` 文件夹中,然后通过 **URL** 来访问它 (http://arduinoYunIp/path_to/www/index.html)。

步骤 1: 介绍项目架构

我将引导您完成每个步骤,轻松完成此项目。

首先让我们来了解前端应用(用户界面)如何与 **API** 进行交互,看看下面的图片:

Arduino Yun 前端应用架构图

Arduino REST API:

如你所见,前端应用程序作为一个类似 **Apache** 的服务器。要设置 **API**,请将“**Bridge Example**”上传到 **Arduino** 上,如图所示:

Arduino Yun bridge Example

API 的工作原理如下:要打开 **LED 13**,您必须将以下 **GET** 请求发送到 **Arduino**:

- <http://arduinoIp/arduino/mode/13/output>
- <http://arduinoIp/arduino/digital/13/1>

您还可以使用模拟引脚,或创建自己的自定义 **URL** 命令,我们将在后面的内容介绍到。

前端:

前端基本上包含这些文件(在 **Arduino Yun** 的 `www/` 文件夹中提供):

前端项目文件

我使用 **bootstrap** 来美化 **html**,并使用 **jQuery** (**JavaScript** 库)将请求发送到 **API**。您可以使用 **jQuery** 轻松地向 **API** 使用 **AJAX** 请求,示例如下:

main.js

```

1 $('#onButton').click(function() {
2 $.get('http://192.168.0.10/arduino/digital/13/1',
      function(response) { console.log(response);
3   });
4 });

```

index.html

```
1 <button type="button" class="btn btn-secondary" id="onButton">On</button>
```

我不会解释 **bootstrap** 的语法，因为它对于项目来说并不重要，实际上你可以忽略 **CSS**，应用程序也将会正常工作。

步骤 2: 解释项目

一图胜行言，看看照片，布线看起来有点复杂，但它实际上它很简单：有两个 **led**（引脚 **8** 和 **11**）连接它们的电阻，还有一个 **16x2** 的 **LCD** 连接到引脚 **7**、**12**、**2**、**3**、**4** 和 **5**，电位器仅用于液晶显示器的亮度控制。

Arduino Yun 简单 Web UI Demo

因为 **Arduino Yun** 内置的闪存非常有限（**16 MB**），因为在 **Arduino Yun** 的 **microSD** 卡上旋转服务器文件。您可以按照 **Arduino Yun** 的控制台中的这些命令，在 **SD** 中提供网页：

```

1 > mkdir /mnt/sda1
2 > mount /dev/sda /mnt/sda1
3 # and the symlink (may already exist)
4 > ln -s /mnt/sda1/www /sd/www

```

解释一下 **UI**：

- **On/Off Button**，这个按钮连接到引脚 **8**
- **Slider**，用来控制使用 **pwm**（**analogWrite**）引脚 **11** 的亮度强度。
- **Text Input**，可以填写任意的内容，当您按“**Send**”按钮时，这些文本将显示在液晶显示器上。
- **Console Label**，这示 **API** 的输出，每当应用程序向 **API** 发送请求时，它会收到一个响应，这些都会显示在 **Console label** 中。

步骤 3: 自制 **LCD API** 命令

默认的 **bridge** 示例没有控制 **LCD** 的命令, 但添加以下代码后, 将会方便我们发送文本:

main.ino

```
1 void process(BridgeClient client) {
2   // read the command
3   String command = client.readStringUntil('/');
4
5   // is "digital" command?
6   if (command == "digital") {
7     digitalCommand(client);
8   }
9
10  // is "analog" command?
11  if (command == "analog") {
12    analogCommand(client);
13  }
14
15  // is "mode" command?
16  if (command == "mode") {
17    modeCommand(client);
18  }
19
20  // --- Here starts the lcd code ---
21
22  // is "lcd" command?
23  if (command == "lcd") {
24    lcdCommand(client);
25  }
26 }
27
28 void lcdCommand(BridgeClient client) {
29   String text = client.readStringUntil('/');
30   String text2 = "";
31   lcd.clear();
```

```
32  if (text.length()-2 > 16)
33  {    text2 = text.substring(16, text.length()-2);
34      lcd.print(text.substring(0,16));
35      lcd.setCursor(0, 1);
36      lcd.print(text2);
37  }
38  else
39  {
40      lcd.print(text.substring(0,text.length()-2));
41  }
42
43  client.print(F("LCD: "));
44  client.print(text);
45 }
```

main.js

```
1 $('#send').click(function(){
2     var url = "http://192.168.XX.XX/arduino";
3     var lcdUrl = "/lcd/";
4     $.get(url+lcdUrl+$("#text").val()+"/", function(response){
5         console.log(response);
6         $("#console").text(response);
7     });
8 });
```

index.html

```
1 <div class="input-group container">
2     <input type="text" class="form-control" placeholder="" id="text">
3     <span class="input-group-btn">
4         <button class="btn btn-secondary" type="button"
5             id="send">Send</button>
6     </span>
7 </div>
8 <ol class="breadcrumb">
9     <li class="breadcrumb-item active">Console:</li>
```

```
10 <li class="breadcrumb-item active" id="console"></li>
11 </ol>
```

步骤 4: **Do it!**

所以, 为了使这个项目工作, 请按照下列步骤操作:

- 按照照片所示进行连线, 并按照说明进行操作
- 将我修改过的 **Bridge** 示例上传到 **Arduino** 中。
- 将 **index.html**、**js/** 和 **css/** 文件夹上传到 **Arduino** 的 **www/** 文件夹。
- 修改 **main.js** 并写下 **Arduino Yun** 的 **ip**。
- 打开 http://arduinoYunIp/path_to/www/index.html, 玩 ~

以下是本文的 **GitHub** 地址: <https://github.com/martinKindall/yunWebPanel>

原文链接: <http://www.tecnomacs.com/2016/12/03/lets-do-a-simple-webserver-ui-for-arduino-yun>

原文链接: <https://www.wandianshenme.com/play/arduino-yun-bootstrap-jquery-build-simple-web>