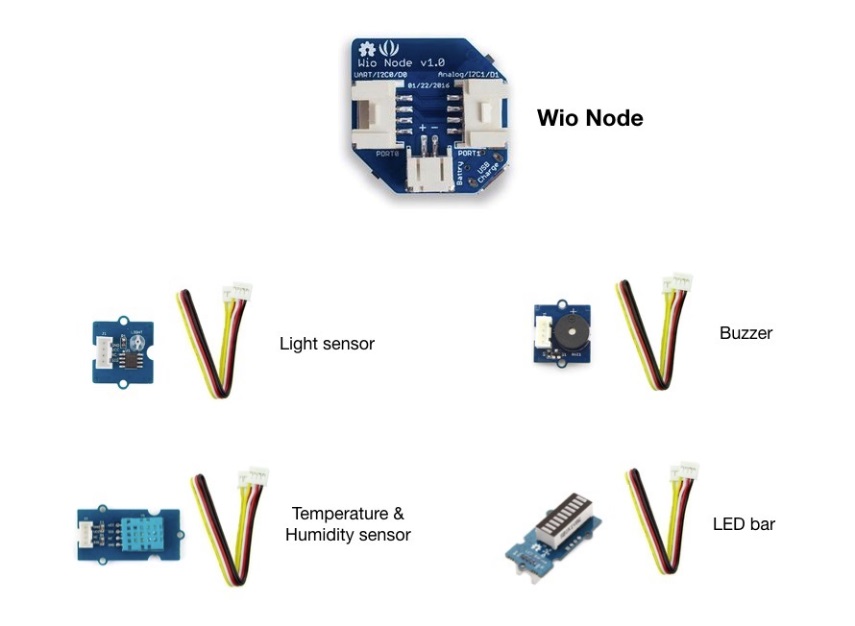
**实验练习**

**1. 准备**

如果需要帮助可以观看视屏[Exercise #1 Video](https://youtu.be/3lqmSyd8QcY) 。

**1.1 领取开发板和传感器Get board and sensors**

指导老师会发给您一套开发设备和工具的套装，零件配置如下图，

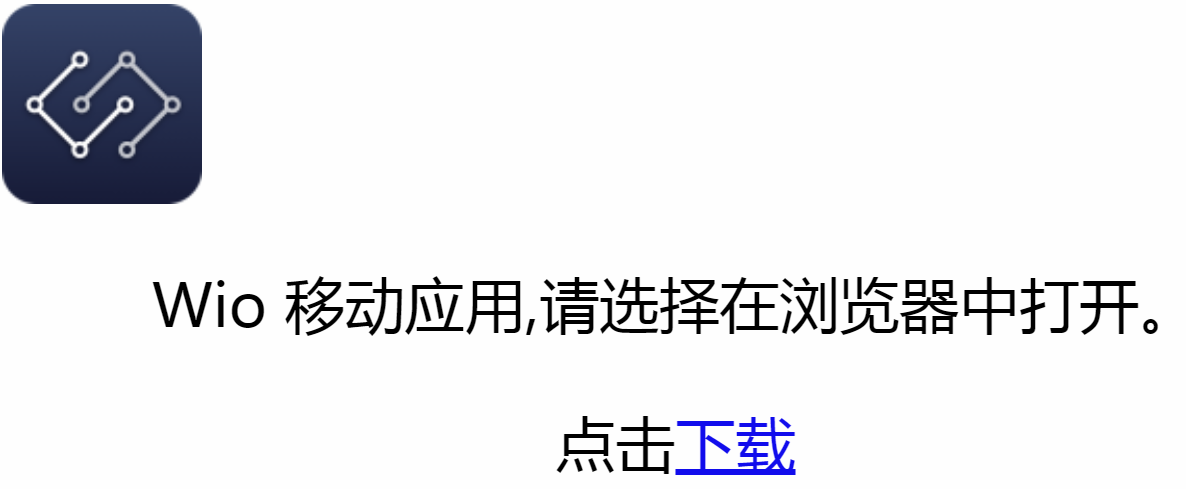


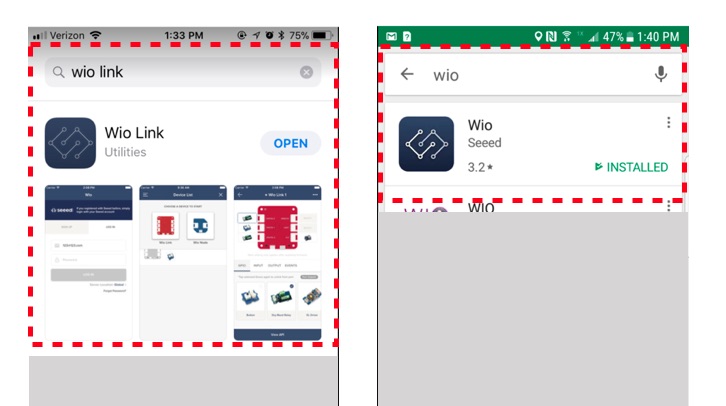
Wio Node开发板的盒子上标有“设备名称”，如“**dev-803**” (注意：视频中我们使用序列号Serial number *D2D117*，我们现在改为使用**dev-[Seq#]** 格式，方便在IoT云服务中找到设备)。

**1.2 下载Wio移动应用**

我们使用Wio移动应用配置开发板上的传感器。如果您使用iPhone手机，您可以在应用商店（Apple Store）中查找"wio link"，并安装。 如果您使用Android手机，您可以在应用市场中查找"wio"，版本为2.3.6，华为应用市场，腾讯应用宝等应用市场均可以下载。也可以扫描下边二维码下载安装。



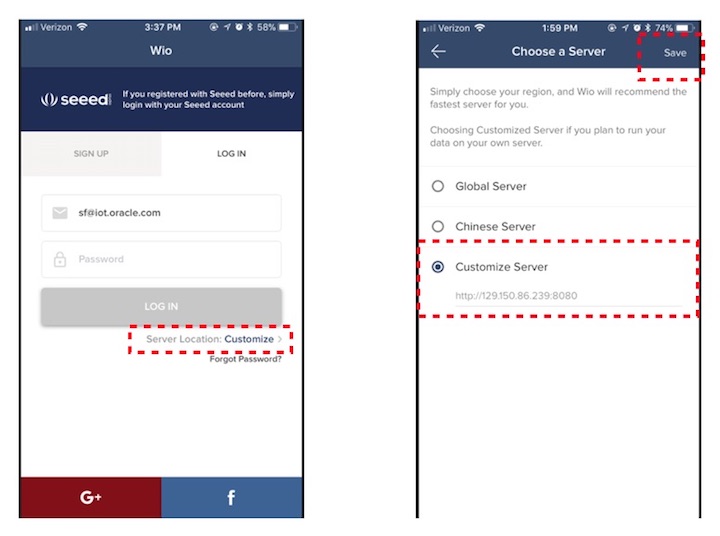




**1.3 配置Wio移动应用**

Wio虚拟服务器（virtualization server）部署在Oracle云，Wio 应用需要配置指向如下定制的服务器：

customize server: http://129.150.86.239:8080



**1.4 登录**

我们使用共用的账号，所有的Wio Node开发板都注册到这个账号。指导老师会给你登录帐号和密码（帐号：sz@iot.oracle.com 密码：123456）。



在您拿到的Wio Node开发板的盒子上有设备名称，如**dev-803**。 在Wio移动应用中，从列表里找到自己的设备。

**2. 连接光传感（light sensor ）和蜂鸣器（buzzer）**

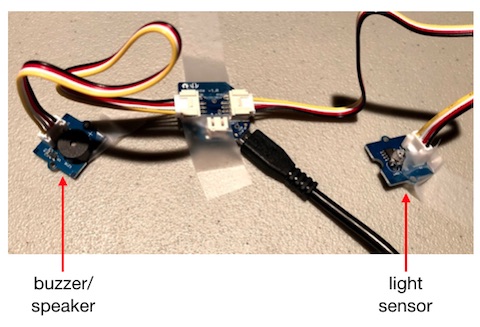
如果需要帮助，请查看视频[Exercise #2 Video](https://youtu.be/mVPDxE4H2XQ) 。

**2.0 使用micro-USB线为Wio Node 供电**

通电一会儿后，在Wio移动应用中刷新设备列表，您应该可以看到您的设备状态变为在线（online）状态。

**2.1 将传感器连接到开发板**

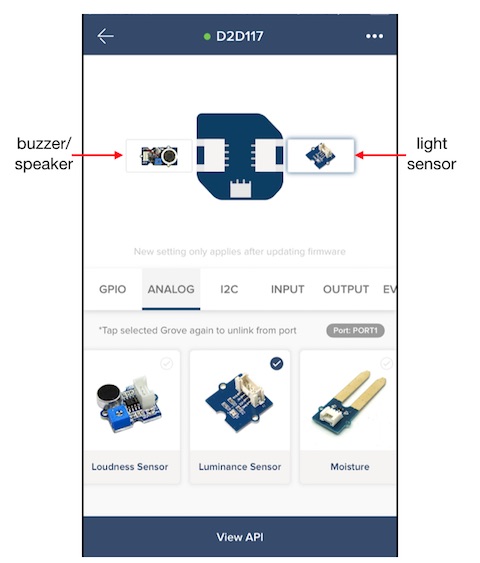
请将光传感器Light sensor和蜂鸣器buzzer连接到Wio开发板，如下图所示：



注意：我们连接光传感器*Light Sensor* (也就是***Luminance Sensor***)到右侧的模拟端口，同时，连接蜂鸣器*Buzzer* (也就是***Speaker***)到左侧的数字端口。

**2.2 在Wio App中连接传感器**

在Wio App中， 我们需要连接光传感器*Light Sensor* (也就是***Luminance Sensor***)和蜂鸣器*Buzzer* (也就是***Speaker***)，与步骤2.1中完全一致，如下图所示



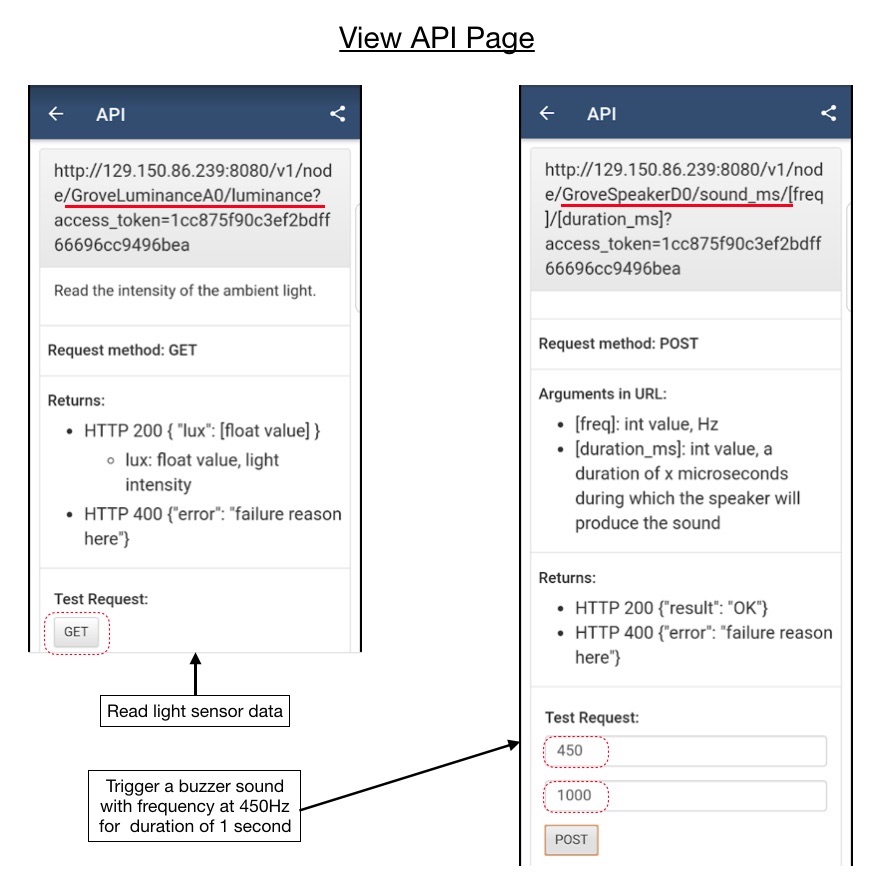
Wio服务器从此获知开发板和传感器的连接方式，从而服务器生成相应的固件更新来更新开发板。

**2.3 更新固件firmware**

每当按照步骤2.2在Wio App更新传感器时，红色的**Update Firmware**按钮都会显示。需要点击按钮触发Wio服务器通过OTA（Over The Air）更新开发板固件。

**2.4 查看API页面**

可以使用**View API**按钮或者菜单导航到API页面。查看API页面的详细信息。您将会找到各种与传感器和蜂鸣器交互的APIs接口。您能够通过API调用与开发板上的传感器和蜂鸣器交互。



**2.5 读取光传感器light sensor的数据**

参考步骤2.4的图，查找 **luminance** API ，通过点击GET按钮读取光传感器读数。

**2.6 通过蜂鸣器buzzer播放声音**

参考步骤2.4的图，查找**sound\_ms** API, 填写声音频率frequency和持续时间duration，通过点击POST按钮触发蜂鸣器发声。

**2.7 小结**

刚刚您只是使用 Wio App和它的API页面与开发板上的传感器和蜂鸣器交互。到目前为止，不需要编写程序。在练习#3中，我们将编写Javascript程序通过js库调用这些API。

**3. 编写javascript程序读取光传感器light sensor读数，触发蜂鸣器buzzer发声**

如果需要帮助，请查看视屏[Exercise #3 Video](https://youtu.be/B8qzL055iMo) 。

**3.1 下载并安装Node.js**

（也可以从指导老师获取出获得安装包。）

Windows安装Node.js方法：[Install Node.js on Windows](https://treehouse.github.io/installation-guides/windows/node-windows.html)

MacBook安装Node.js方法：[Install Node.js on MacBook](https://treehouse.github.io/installation-guides/mac/node-mac.html)

Linux 安装Node.js方法： [Install Node.js on Linux](https://treehouse.github.io/installation-guides/linux/node-linux.html)

在命令行窗口输入"node -v"验证安装的Node.js版本。

**3.2 下载项目源代码**

访问<https://github.com/raymondxie/iotws18>, 阅读主页内容，了解各个源文件的使用方法。然后下载源代码，解压到项目文件夹： **iotws18-master**

在命令行窗口，切换当前目录到**iotws18-master**, 并执行下面命令

npm install

这将取回在package.json声明的这个项目必需的npm库 。

**3.3 检查配置**

你需要得到一个访问令牌**access\_token** 来执行访问开发板和传感器的程序。切换目录到项目文件夹 **iotws18-master**, 打开**sensor-config.js**文件 ，查看"**wio\_iot**"设置：

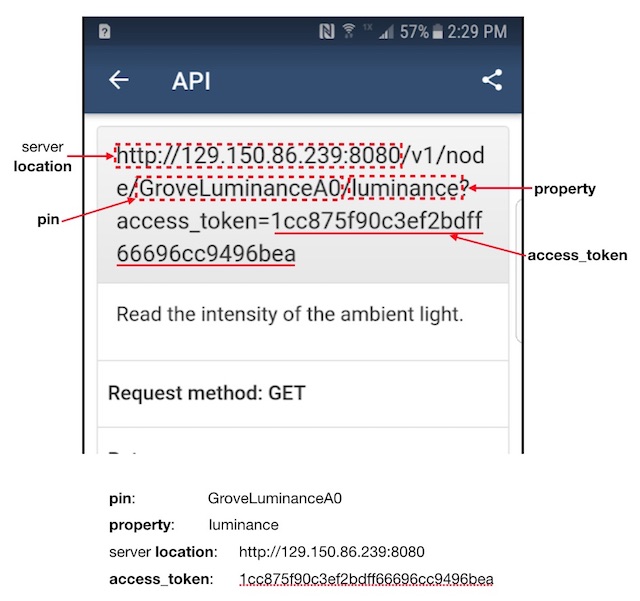
exports.wio\_iot = {

"location": "http://129.150.86.239:8080",

"access\_token": "94ef3060fb964724e42c1f3d1c6cd116"

};

请确认**location**设置为正确的Wio server，并且修改**access\_token**为Wio App中显示的你的开发板的访问令牌。参考Wio App的API页面，如下图所示:



**3.4 编写Javascript触发蜂鸣器发声**

我们提供了一个入门代码**testWio.js**, 查看文件中的API调用。您可以打开命令行窗口，导航到项目目录**iotws18-master**, 执行下面命令进行测试：

node testWio.js

应该听到蜂鸣器发声，同时看到光传感器的读数。

现在轮到你来写Javascript程序了，实现当光传感器读数低于500时蜂鸣器发声报警。

使用**testWio.js** 样例作为调用API的参考，试写Javascript代码读取光传感器读数，根据读数判断是否触发蜂鸣器发声。如果您不熟悉Javascript，您可以执行solution code **testWio-solution.js**跳过这一步，进入步骤3.5

**3.5 执行solution code**

node testWio-solution.js

现在，将光传感器放在手掌中挡住光线，观察日志输出，看光传感器读数是否低于500，并且蜂鸣器发声报警。

**3.6 小结**

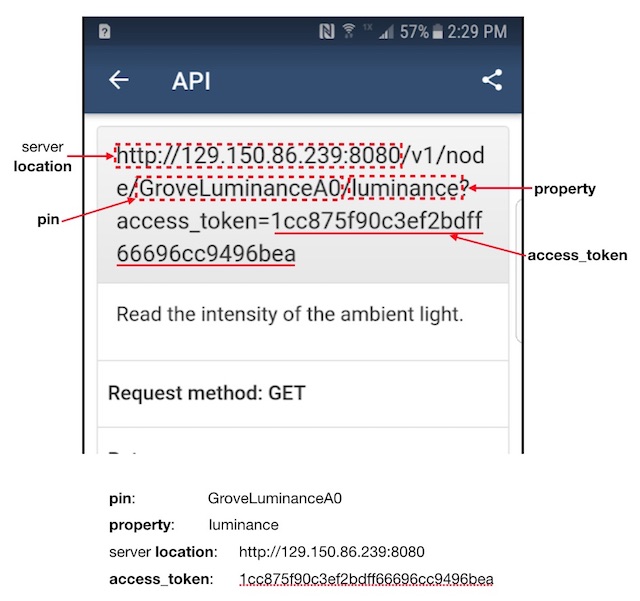
现在您已经编写了一段Javascript程序，用编程的方式与开发板和传感器以及蜂鸣器交互。

**4. 通过Oracle IoTCS与设备通信**

**4.1 连接和配置文件**

你需要配置**sensor-config.js**文件设定传感器与开发板上的哪个接口（针连接器）相连，传感器读数映射到IoTCS中定义的设备模型的那些属性，同时你也需要配置Wio虚拟服务器地址和访问令牌access token 。

参考步骤3.3的图片，该图描述了如何获取 **pin、property**、**token**和 **location** 配置。



在**sensor-config.js**设置相应的值。

{

"type": "INPUT",

"pin": "GroveLuminanceA0",

"property": "luminance",

"attr": "light",

"val": 0

},

exports.wio\_iot = {

"location": "http://129.150.86.239:8080",

"access\_token": "**~~1cc875f90c3ef2bdff66696cc9496bea~~**"

};

**4.2 安全开通Security provision**

向指导老师申请安全开通文件Security provision file和密码。文件名应该类似"**Code-Shenzhen-dev-[sequence#].csv**", 如, **Code-Shenzhen-dev-801.csv**. 将它放到项目文件夹**iotws18-master**下。

**4.3 运行IoTCS客户端程序**

我们提供了**wio-iotcs-client.js** 代码和**sensor-config.js**配置文件来完成所有的程序逻辑**。** 确认"sensor-config.js" 配置符合您开发板的设置。在命令行窗口执行下面命令：

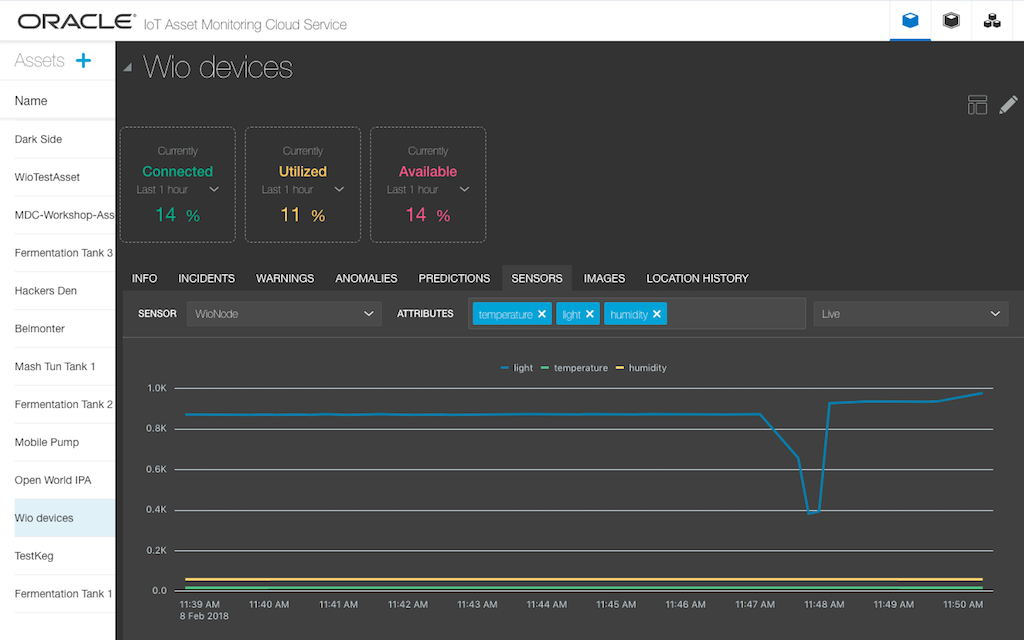
node wio-iotcs-client.js [provision-file] [provision-password]

[provision-file], [provision-password]是指导老师给你的文件和密码。如，**Code-Shenzhen-dev-801.csv 和Shenzhen1**

node wio-iotcs-client.js Code-Shenzhen-dev-101.csv Shenzhen1

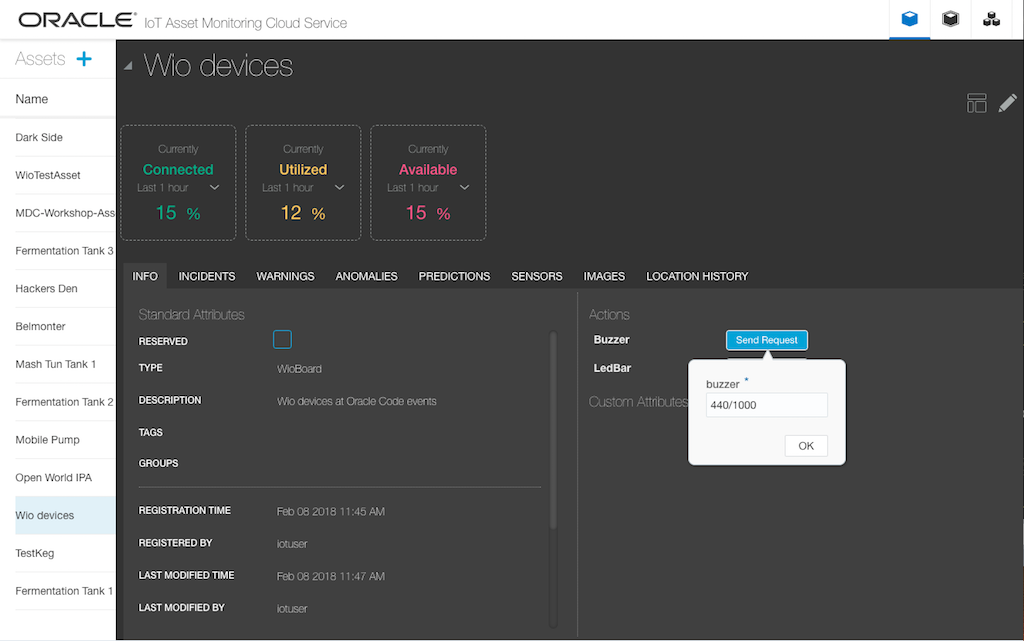
你应该看到日志输出和光传感器读数。当光传感器读数变化时，数据被发送到IoTCS。

**4.4 在IoT Asset Monitor中查看传感器读数**

与指导老师一起通过IoT Asset Monitor查看传感器读数和图标，界面如下图所示， 

**4.5 从IoT Asset Monitor触发蜂鸣器发声**

与指导老师一起通过IoT Asset Monitor触发您开发板上的蜂鸣器发声。界面如下图，

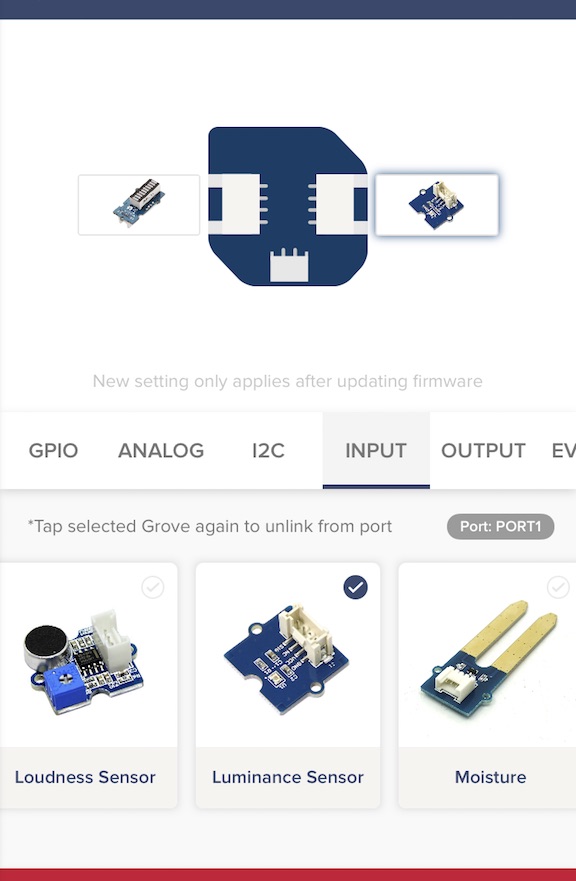


**5. 试试其它的传感器**

到目前为止，我们完成了所有的练习。如果你愿意，你可以继续实验温湿度传感器和LED bar。只要按照练习#2进行, 将光传感器替换为温湿度传感器，将蜂鸣器替换为LED bar。

这里有一些将各种传感器连接到开发板的例子，供您参考：

[Wio-node with light sensor and LED bar](https://github.com/raymondxie/iotws18/blob/master/images/wionode-light-bar.jpg)

  
[Wio-node with temperature sensor and a buzzer](https://github.com/raymondxie/iotws18/blob/master/images/wionode-temp-buzzer.jpg)



**6. 回家可以继续玩Go home and try more**

当您回家后，开发板不能继续连接到研讨会使用的Wifi网络, 因此您需要按照链接中的指南重新设置开发板- [Getting Started](http://wiki.seeed.cc/Wio_Node/#get-started), 为开发板设置您家的Wifi SSID和密码，以便它可以访问Internet。

同时，你也可以使用Wio **Global server**替换研讨会中使用的**Customize server**。您需要注册一个**Global server**的账号，将开发板加入到您的账号中。请按照**Getting Started**入门指南的从步骤1到步骤5进行设置。您可以用我们提供的LED bar替换指南中的Grove LED 。