**第一章**

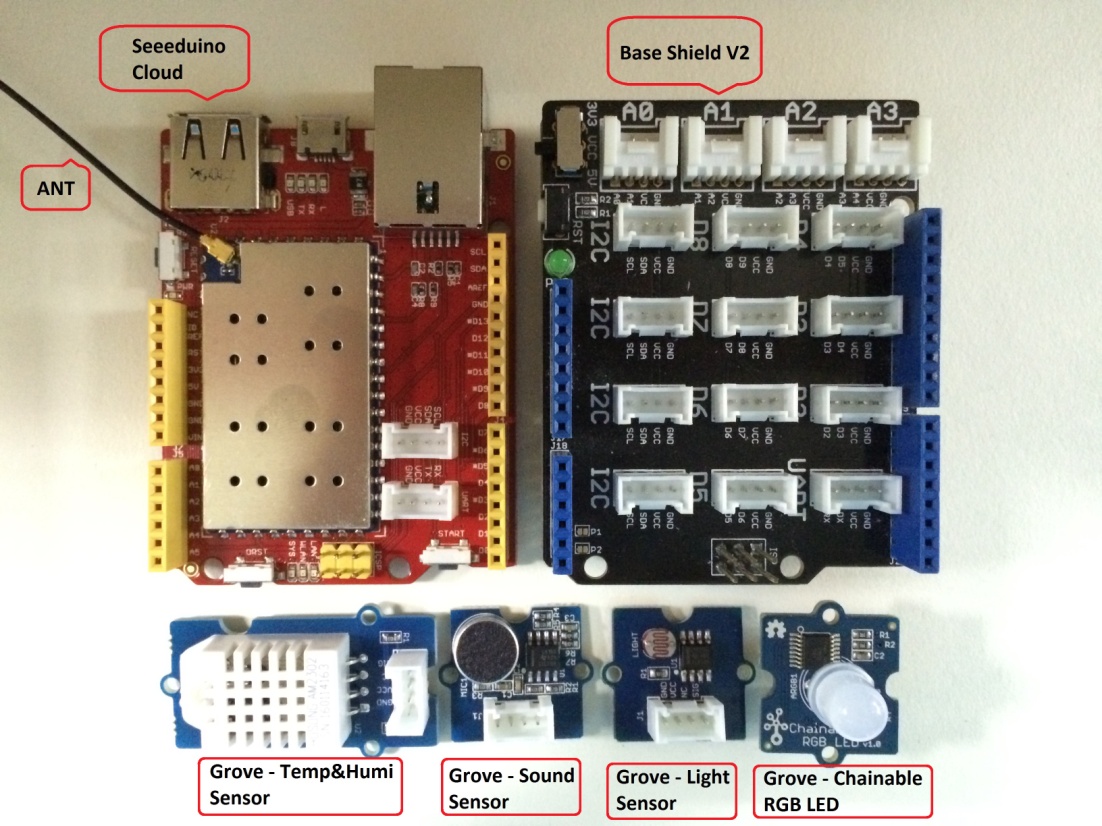
适用于 Seeeduino Cloud 设备已经完成文件拷贝以及配置。

1. 硬件准备

（1）列表

① [Seeeduino Cloud](https://www.seeedstudio.com/item_detail.html?p_id=2123)； [Base Shield V2](https://www.seeedstudio.com/item_detail.html?p_id=1378)； [Grove - Temp&Humi Sensor](https://www.seeedstudio.com/item_detail.html?p_id=838)；

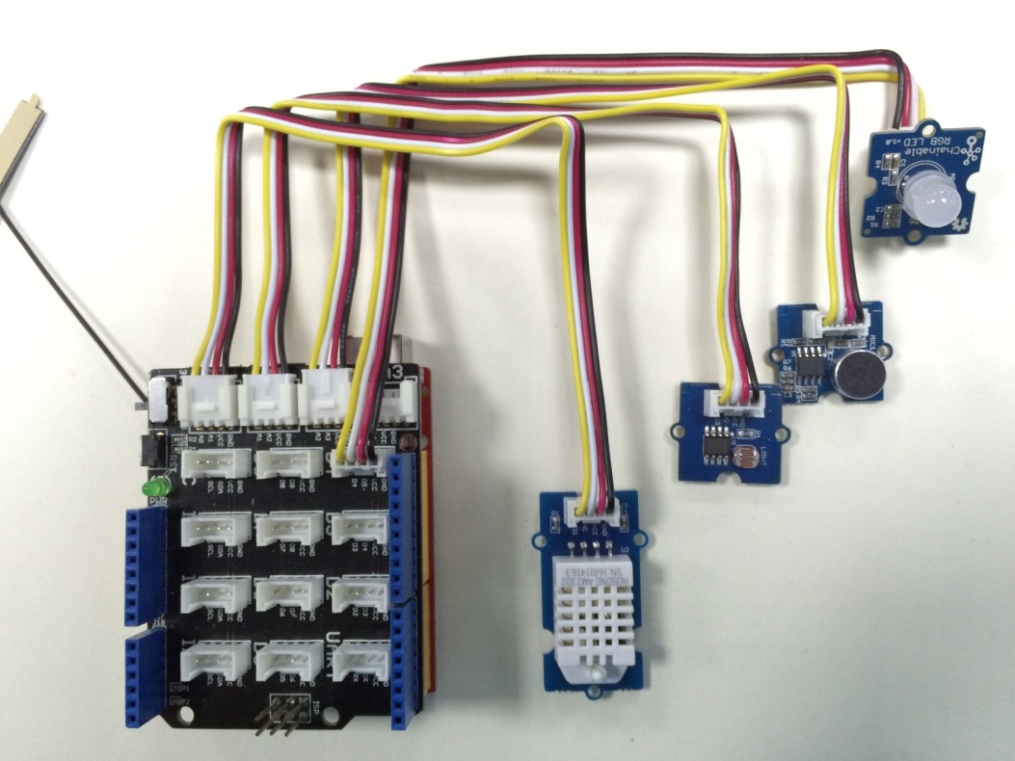
[Grove - Light Sensor](https://www.seeedstudio.com/item_detail.html?p_id=746)； [Grove - Sound Sensor](https://www.seeedstudio.com/item_detail.html?p_id=752)；⑥ [Grove - Chainable RGB LED](https://www.seeedstudio.com/item_detail.html?p_id=850)。



（2）连接

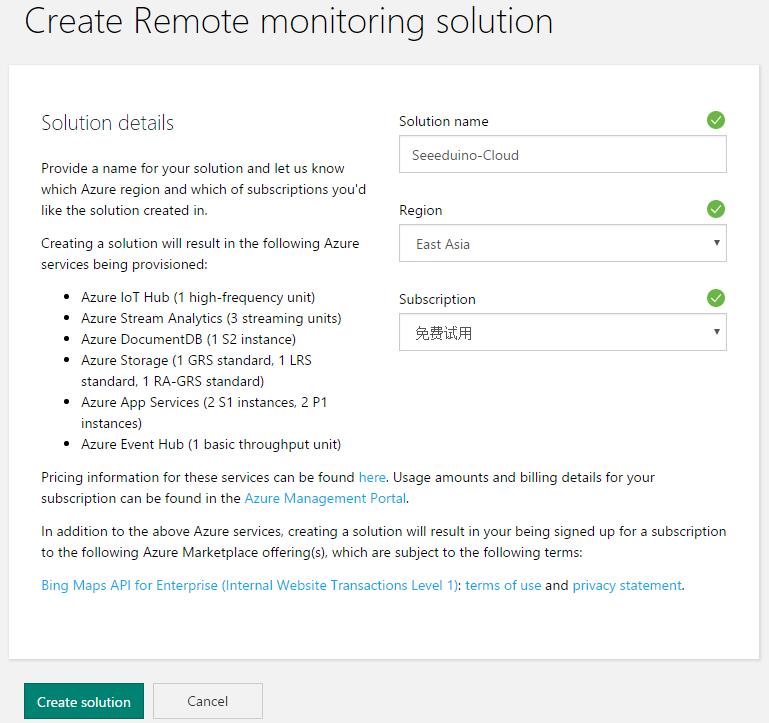
Grove - Temp&Humi Sensor -> A0； Grove - Light Sensor -> A1；

Grove - Sound Sensor -> A2； Grove - Chainable RGB LED -> D4 & D5。

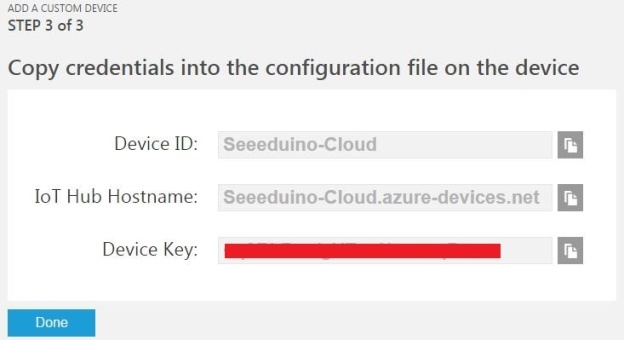
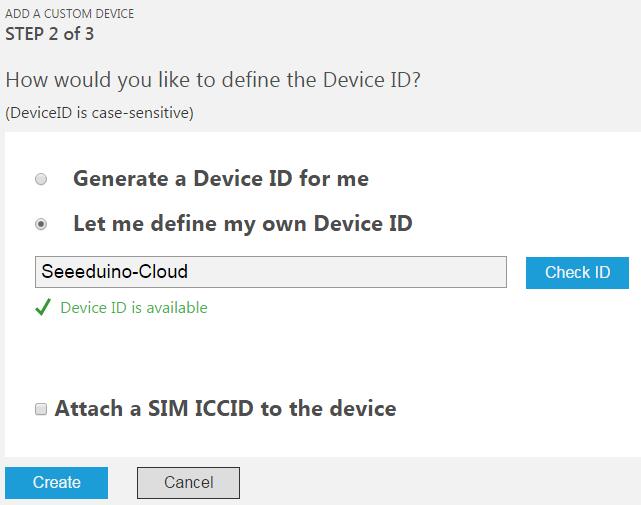


2. 创建新 IoTHub 服务器

（1）注册与登录 [www.azureiotsuite.com](http://www.azureiotsuite.com)，并创建新的远程监测方案



（2）增加新的客户硬件设备，记录 host name, device id 和 device key



3. 网络配置

使用 PUTTY 工具将 Seeeduino Cloud 连接到可用的 WiFi, 可参考 [wiki](http://www.seeedstudio.com/wiki/Seeeduino_Cloud) 描述。

备注：WiFi 的默认模式为 AP（[Access](javascript:void(0);) [Point](javascript:void(0);)）。当完成配置后它将变为 Station 模式。

长按 Cloud 复位按钮，并于 5 秒后释放，WiFi 将恢复默认设置。

4. 输入 Azure 连接字符串

（1）使用 vi 键入 host name、device id 与 device key 到文件 " AzureConnectionString "。文件路径为 " /root/AzureConnectionString "。

（2）或者使用 echo 命令输入字符串，如下：

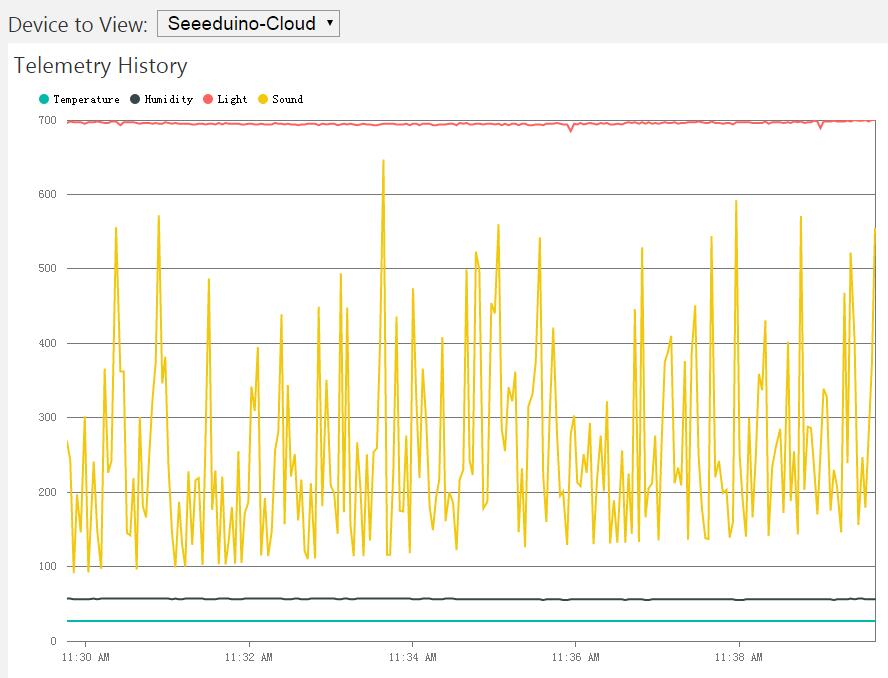
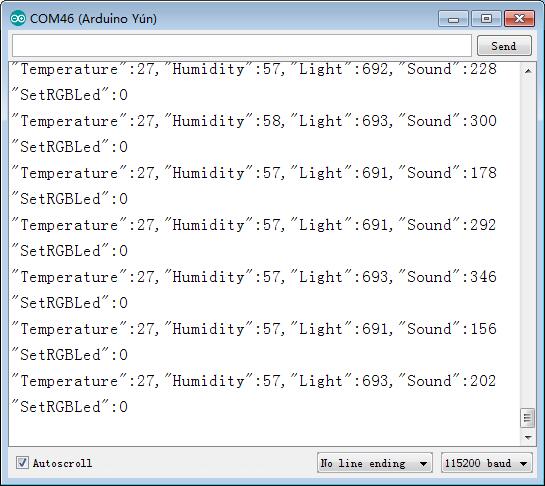
echo -n 'HOST\_NAME:???.azure-devices.???

DEVICE\_ID:???

DEVICE\_KEY:???' > /root/AzureConnectionString

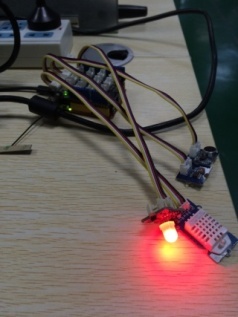
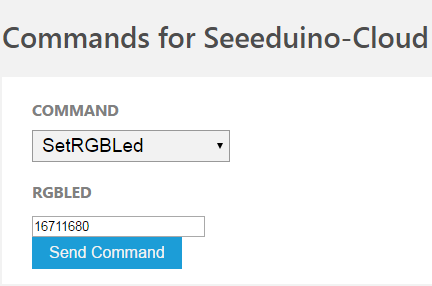
5. 打开 Dashboard 与 Serial 监测面板，等待测试

（1）传感器数据监测：

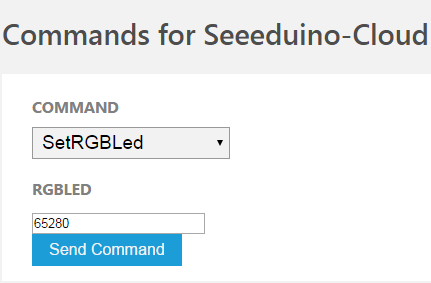


（2）RGB 灯控制

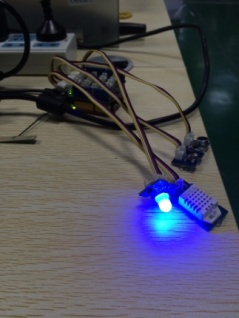
设置红色：16711680 -> 0xff0000



设置绿色：65280-> 0x00ff00



设置蓝色：255-> 0x0000ff



**第二章**.

适用于 Seeeduino Cloud 设备未做文件拷贝以及配置。

1. 硬件准备

参见第一章第 1 部分。

2. 创建新 IoTHub 服务器

参见第一章第 2 部分。

3. 文件拷贝与配置

（1）将 PC 连接到 Seeeduino Cloud 热点，并使用 PUTTY 登录到 Seeeduino Cloud

（AP：SeeeduinoCloud-xxxxx；IP：192.168.240.1；账号：root；密码：seeeduino）

（2）将路径 " .../Backup Flies " 下的所有文件拷贝到 U 盘

（3）插入 U 盘，并获取设备的挂载名称



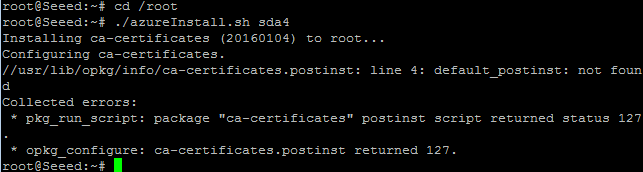
（4）手动复制文件 " azureInstall.sh " 到路径 "/root"



（5）带参数运行 " azureInstall.sh " （参数为 U 盘挂载名称）

（备注 : 安装证书时忽略第 127 号错误信息提示）

（备注 : Azure IoT Hub Client 进程将随开机自启动）



（6）检查路径 /root，得到文件列表如下



4. 输入 Azure 连接字符串

参见第一章第 4 部分。

5. 网络配置

参见第一章第 3 部分。

7. 拷贝路径 " .../Arduino Code/xxx.ino " 与 " .../Arduino Code/libraries " 下的文件到 Arduino IDE，编译 xxx.ino 文件，并下载到 MCU 内核（ATMega32u4）

6. 打开 Dashboard 与 Serial 监测面板，等待测试

参见第一章第 5 部分。

**第三章**

适用于 Azure IoT Hub Client 源码交叉编译。

1. 开发环境

Ubuntu

2. 下载 azure-iot-sdks

git clone --recursive <https://github.com/Azure/azure-iot-sdks.git>

3. 安装 cmake 与 gcc

sudo add-apt-repository ppa:george-edison55/cmake-3.x # cmake ppa

sudo add-apt-repository ppa:ubuntu-toolchain-r/test # gcc ppa

sudo apt-get update

sudo apt-get install cmake

sudo apt-get install gcc-4.9 g++-4.9

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 60 --slave /usr/bin/g++ g++ /usr/bin/g++-4.9

备注 : 确认 cmake 版本 3.+，以及 gcc 版本 4.9.+

4. 拷贝源码 " .../Source Code/remote\_monitoring.c " 到文件

" ... /azure-iot-sdks/c/serializer/samples/remote\_monitoring/remote\_monitoring.c "

5. 编译 SDK 与例程

cd c/build\_all/arduino

sudo ./setup.sh

sudo ./build.sh

6. 拷贝文件

" .../openwrt/sdk/build\_dir/target-mips\_r2\_uClibc-0.9.33.2/azure-iot-sdks-1/serializer/samples/ remote\_monitoring /remote\_monitoring "，并将文件名称改为 " AzureIoTHubClient "

（AzureIoTHubClient 是一个可执行文件）

7. 按照第二章与第一章的步骤拷贝文件、配置与运行例程。