**Section 1**

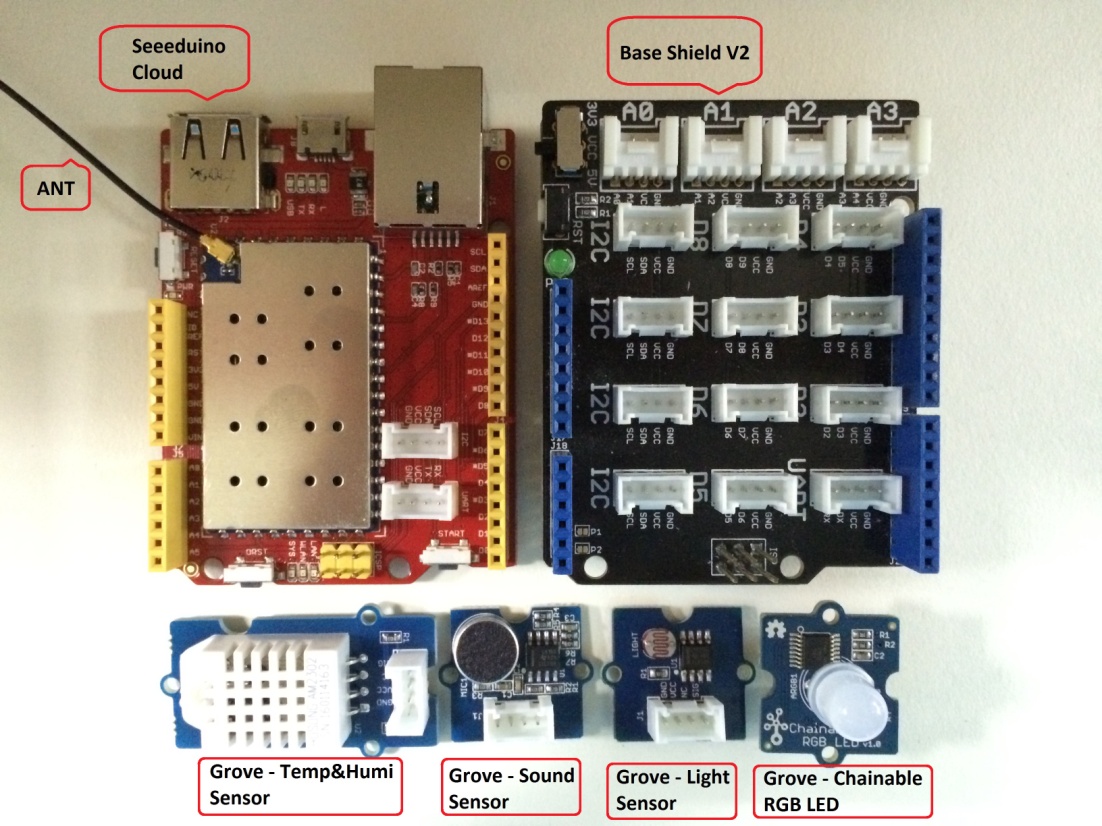
For the Seeeduino Cloud development board device that have done files copy and configuration.

1. Hardware prepare

(1) List

① [Seeeduino Cloud](https://www.seeedstudio.com/item_detail.html?p_id=2123); ② [Base Shield V2](https://www.seeedstudio.com/item_detail.html?p_id=1378);  ③ [Grove - Temp&Humi Sensor](https://www.seeedstudio.com/item_detail.html?p_id=838);

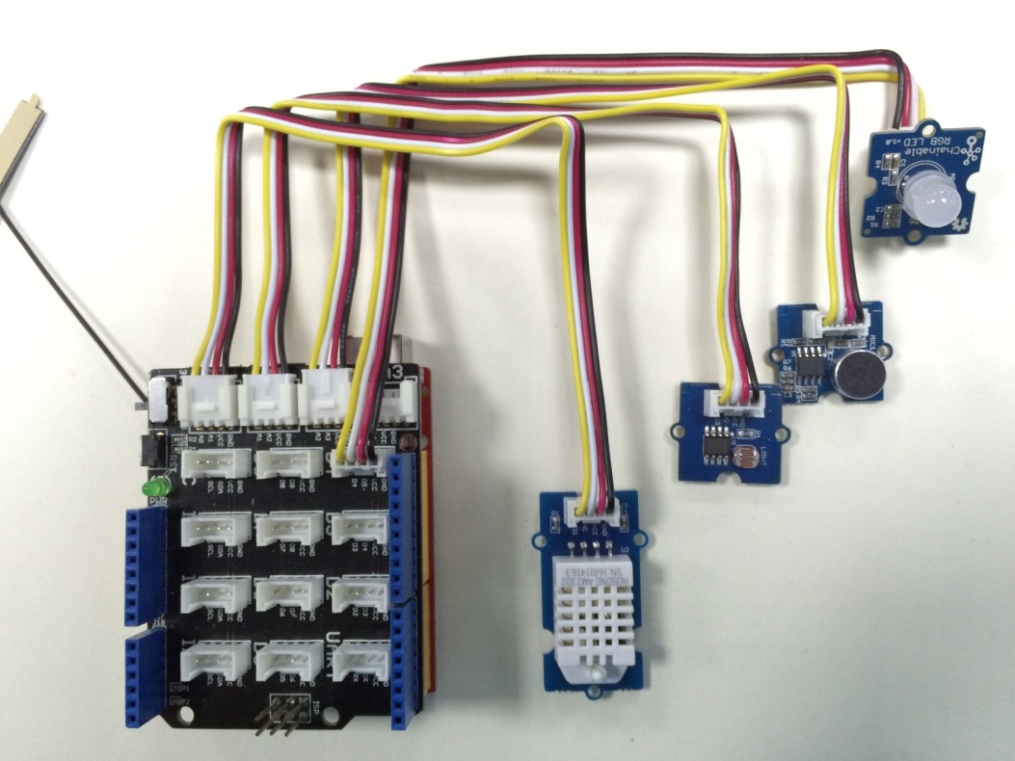
④ [Grove - Light Sensor](https://www.seeedstudio.com/item_detail.html?p_id=746); ⑤ [Grove - Sound Sensor](https://www.seeedstudio.com/item_detail.html?p_id=752); ⑥ [Grove - Chainable RGB LED](https://www.seeedstudio.com/item_detail.html?p_id=850).



(2) Connection

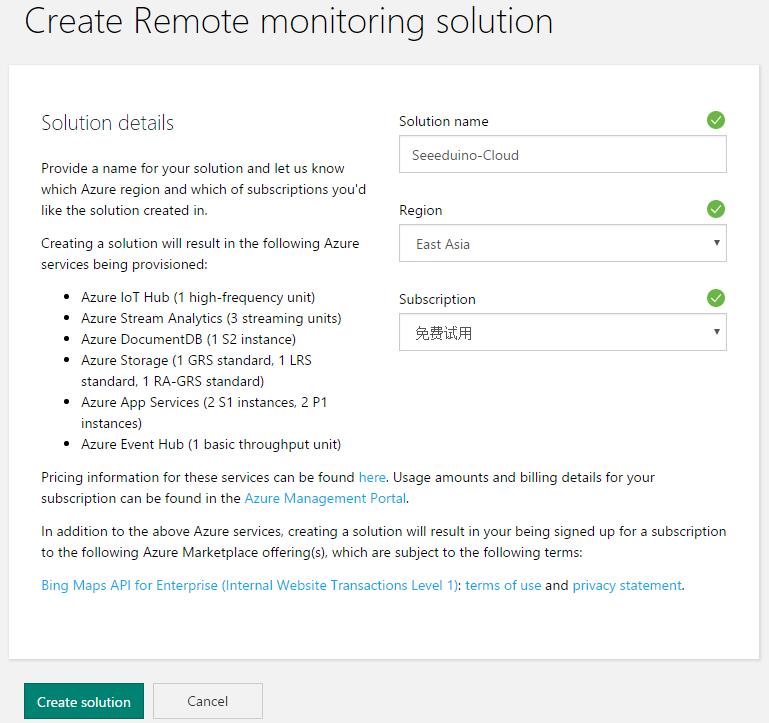
① Grove - Temp&Humi Sensor -> A0; ② Grove - Light Sensor -> A1

③ Grove - Sound Sensor -> A2; ④ Grove - Chainable RGB LED -> D4 & D5

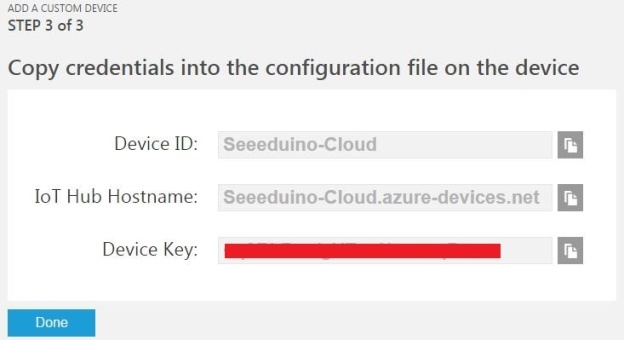
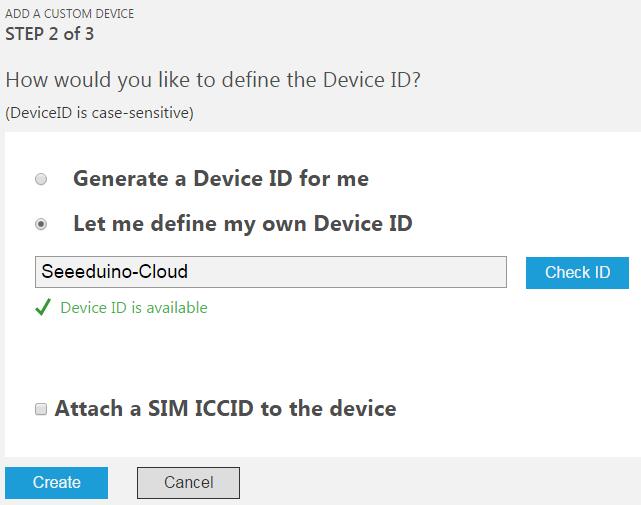


2. Create a new IoThub server

(1) Register and log in [www.azureiotsuite.com](http://www.azureiotsuite.com), and create a new remote monitoring solution



(2) Add a new custom hardware device, then note the host name, device id and device key.



3. Network configuration

Use PUTTY tool connect Seeeduino Cloud to a usable wifi, can make reference to this [wiki](http://www.seeedstudio.com/wiki/Seeeduino_Cloud).

NOTE : The WiFi default mode is AP ([Access](javascript:void(0);) [Point](javascript:void(0);)), when finish the configuration it will change to Station mode. Long press the Cloud Reset button and released after 5 seconds, it will reset the WiFi settings.

4. Input Azure connection string

(1) Use vi to key the host name, device id and device key to the file " AzureConnectionString ". The path is " /root/AzureConnectionString ".

(2) Or use echo command to input the string, follow as:

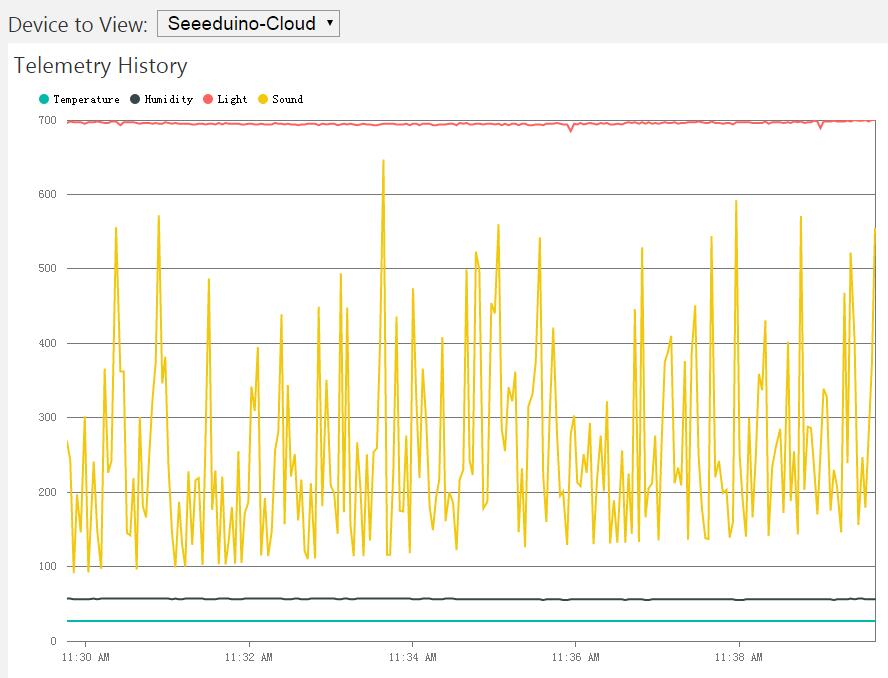
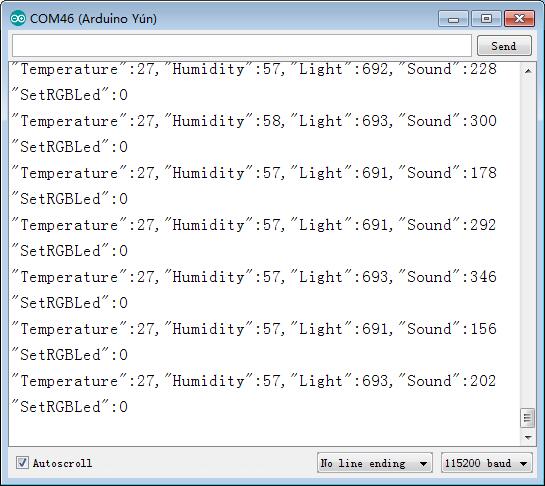
echo -n 'HOST\_NAME:???.azure-devices.???

DEVICE\_ID:???

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

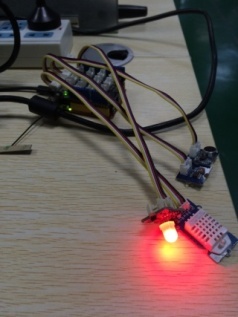
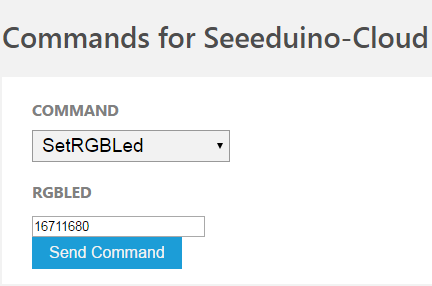
5. Open Dashboard Monitor and Serial Monitor wait for test.

(1) Sensor data monitor:

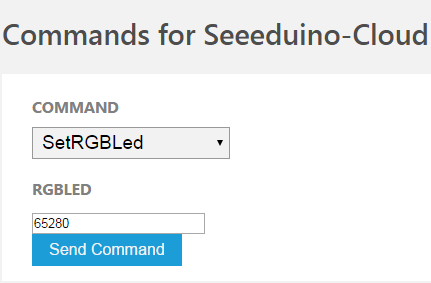


(2) RGB led control

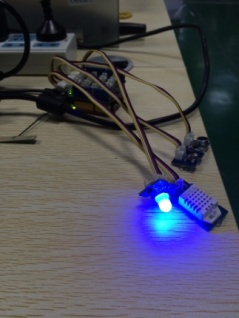
Set red color: 16711680 -> 0xff0000



Set green color: 65280-> 0x00ff00



Set blue color: 255-> 0x0000ff



**Section 2**

For the Seeeduino Cloud device that haven't done files copy and configuration.

1. Hardware prepare

See also Section 1 part 1.

2. Create a new IoThub server

See also Section 1 part 2.

3. Copy files and configuration

(1) Connect PC to Seeeduino Cloud AP, and use PUTTY connect to Seeeduino Cloud

(AP : SeeeduinoCloud-xxxxx; IP : 192.168.240.1; Log in name : root; Password : seeeduino)

(2) Copy all the file in path " ./Config " to a U disk

(3) Plug it in and get the mount name of U disk



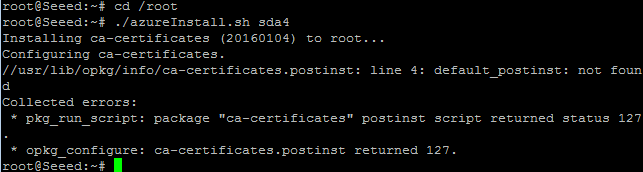
(4) Manual copy the file " azureInstall.sh " to the "/root"



(5) Run the " azureInstall.sh " with a parameter (mount name of U disk)

(NOTE : Ignore the error message 127 when install ca-certificates)

(NOTE : The Azure IoT Hub Client have been setup auto start when system reboot)



(6) Check the /root, can see follow as



4. Input Azure connection string

See also Section 1 part 4.

5. Network configuration

See also Section 1 part 3.

7. Copy the " ./Source/Arduino/xxx.ino " and " ./Source/Arduino/libraries " to Arduino IDE, build the xxx.ino file and upload to MCU core (ATMega32u4).

6. Open Dashboard Monitor and Serial Monitor for test.

See also Section 1 part 5.

**Section 3**

For the Azure IoT Hub Client source code Build part.

1. Development environment

Ubuntu

2. Download azure-iot-sdks

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

3. Install cmake and 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

NOTE : Make sure the cmake version is 3.+, and the gcc version is 4.9.+

4. Copy the source code " ./ Source/OpenWrt/remote\_monitoring.c " to the file

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

5. Build sdk and sample

cd c/build\_all/arduino

sudo ./setup.sh

sudo ./build.sh

6. Copy the file

" .../openwrt/sdk/build\_dir/target-mips\_r2\_uClibc-0.9.33.2/azure-iot-sdks-1/serializer/samples/ remote\_monitoring /remote\_monitoring ", and change the file name to " AzureIoTHubClient "

(AzureIoTHubClient is a executable file)

7. Follow the section 2 and 1 to copy files, setup and run the sample.