**Awesome Engine**: done

It’s already setup and sending data to our IoT Central app.

After dealing with auth stuff, parameters and some debugging, we’ve just used a simpler solution that works very well: a very small python script (around 70 lines of code, taken from here [https://github.com/Azure/iot-central-firmware/tree/master/RaspberryPi](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2FAzure%2Fiot-central-firmware%2Ftree%2Fmaster%2FRaspberryPi&data=02%7C01%7CHarald.Wentein%40microsoft.com%7Cd8639889e1f1415447ad08d686248d50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636843888882557371&sdata=moaee1rjSGvdnyq0mzdPEchrpABx2QWpzx1iHy7omKo%3D&reserved=0)) with basic modifications to use the Grove libs. I’m attaching the file just in case it’s useful (since it’s extremely simple, no need to generate connection strings, etc.) Plus: modifying 4 lines of code, it can be used from any PC to mock the device with no other dependencies.

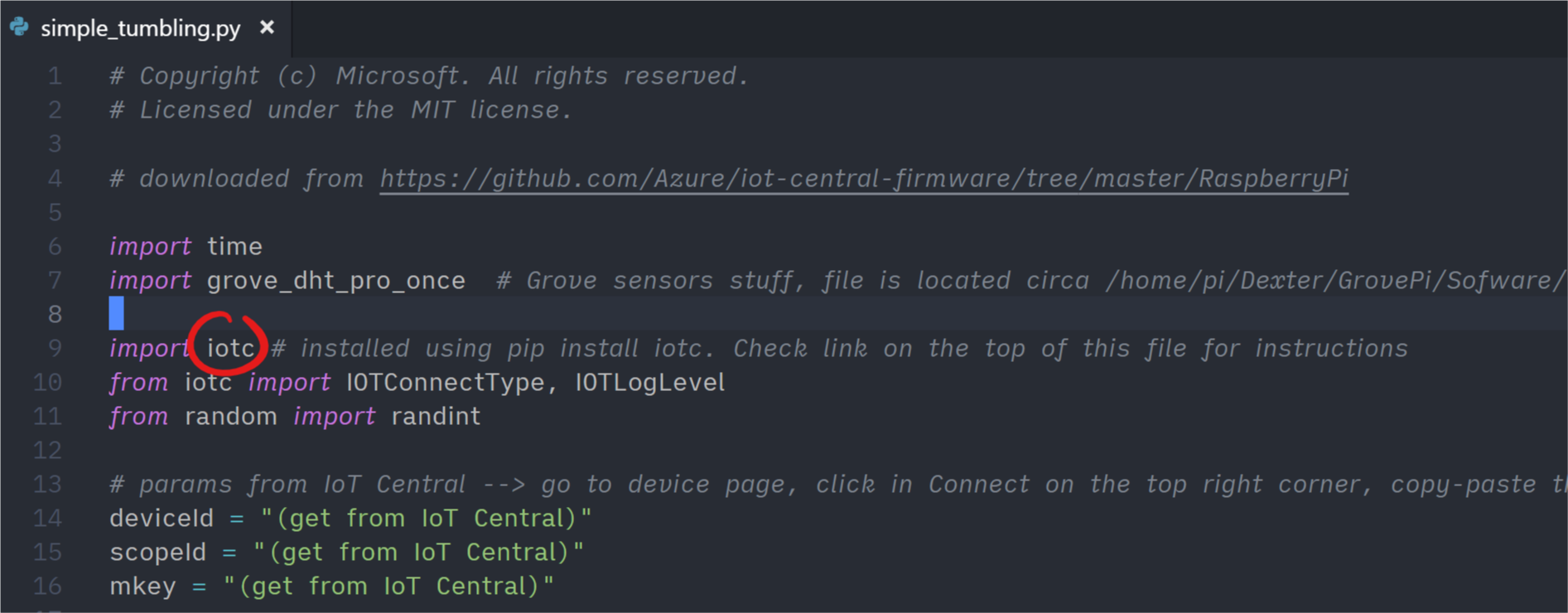




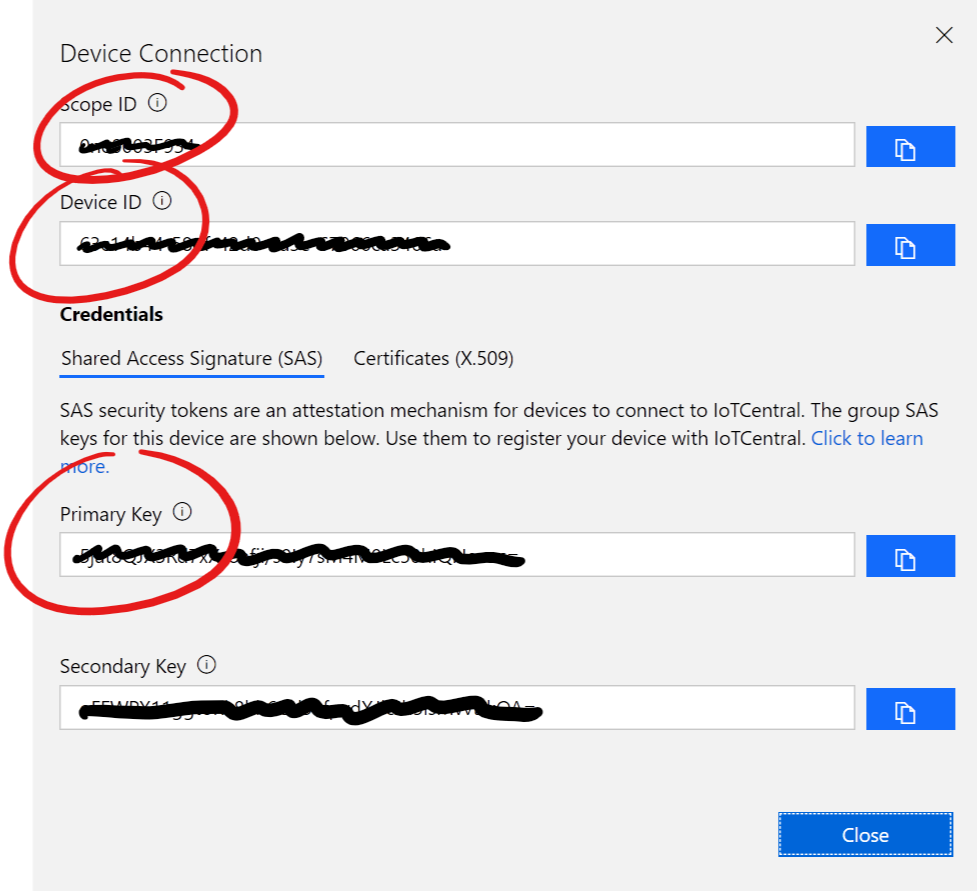
New **simple\_tumbling.py** file with more comments inline. It’s embedded in this document and also stored in the same location as this document.

Setup simplified basic instructions:

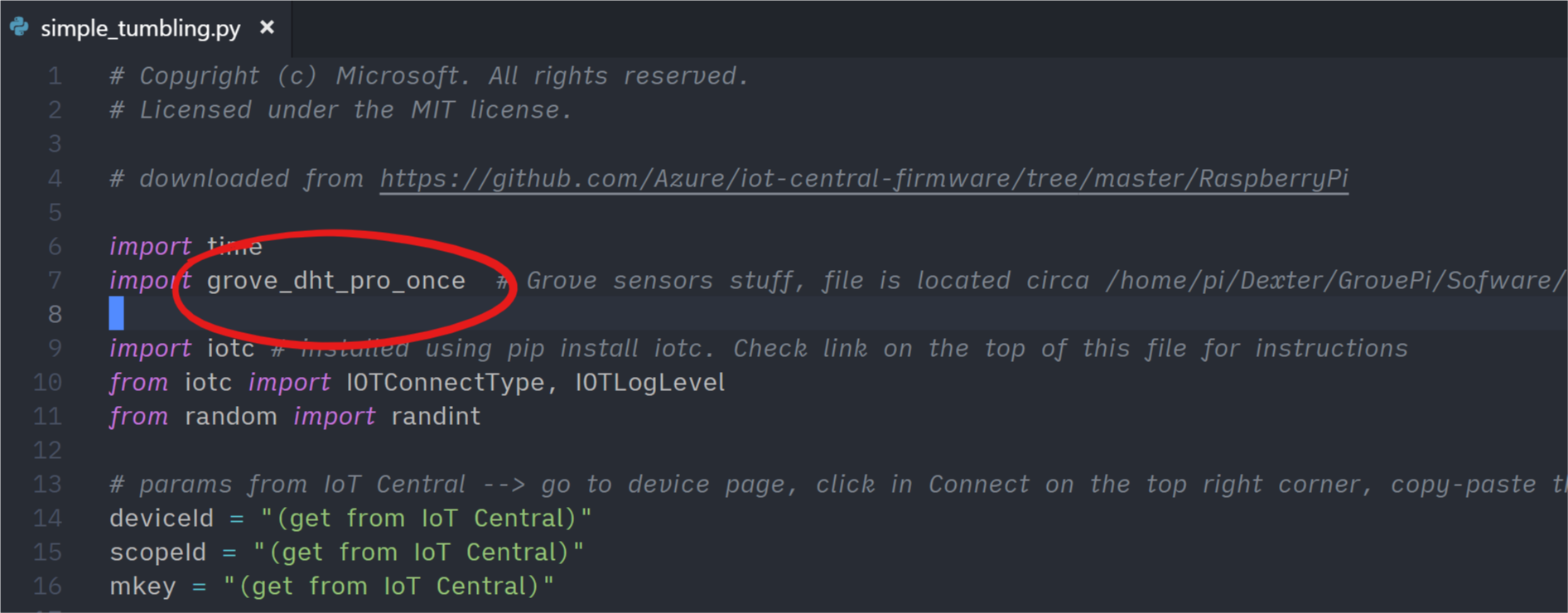
1. The script can be stored in any place, and it’s **not** automatically fired on startup. In our environment, we manually (for the moment) startup every time with a console and the command **python simple\_tumbling.py**. It gives some output to the console.
2. The script uses a python library used to send data form Raspberry Pi to IoT Central. The library is named **iotc** and needs to be installed using **pip install iotc**. It’s documented here: <https://github.com/Azure/iot-central-firmware/tree/master/RaspberryPi>



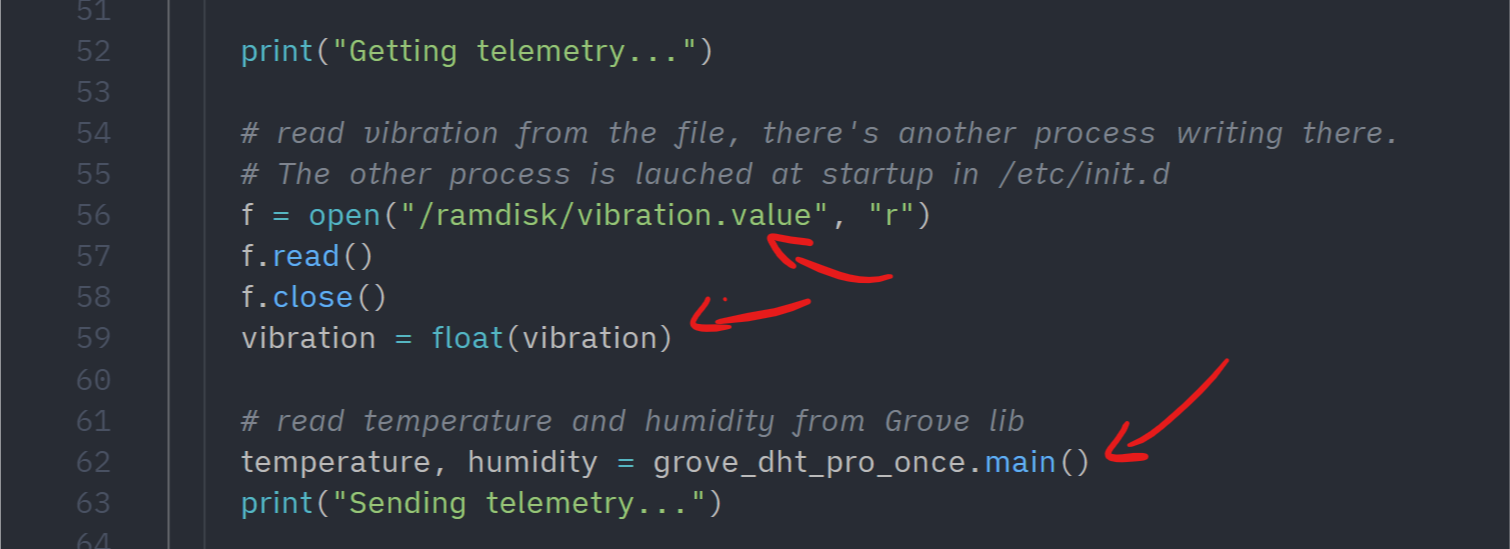
1. Also, once **iotc** is imported, you need to get 3 values from IoT Central device and paste in the script. In IoT Central, go to *device page*, click in *connect* on the top right corner, copy-paste these parameters.



1. The script also uses a library named **grove\_dht\_pro\_once**, used for reading temperature and humidity from sensors. Nothing is needed, just make sure that the path where grove sdks are installed is in the path (it should be already)



1. The vibration is taken from a file. For some reason, there’s another process launched automatically at startup that starts reading vibration from the sensor and writing into a file. The python script just reads the text file to get the value. The file path is “**/ramdisk/vibration.value”**.



1. The startup stuff (can be used to change the device small screen text, as in the first picture of this document), is in a file named init.d stored at **/etc/init.d**. Although this file executes stuff in startup, I’ve been told that this is not the right place to launch our python script. I don’t know exactly why (not an expert!), so we used this to change the small green screen, but nothing more.