

Autonomous Drone Engineer

C2 – Network and ssh

Paul.Guermonprez@intel.com

Autonomous Drone Solutions Architect



Local wireless connectivity

By default, Intel Aero is acting as **Wifi access point**.
It is convenient for the initial calibration and QGC user interface.

- Access point name is "**Aero-***"
where * is the mac address of your drone
- Password is "**1234567890**"
- You can connect your development station to this network
- Then launch ssh on IP **192.168.8.1**, login is "**root**", **no password**

**You can access Intel Aero over the access point included.
But the Drone is NOT connected to Internet.**

Local USB connectivity

If you plug a direct **micro USB Cable** between your PC and Intel Aero, it will create a

USB-ethernet adapter and allocate an IP address for your PC.

You can then access Intel Aero over this USB cable with IP **192.168.7.2**

Note: each OS and installation has different ways to handle networking. This method **may interfere with your setup** (ex: network-manager on Ubuntu). It is an **advanced method**. That's why we recommend you use the wireless method (see next slide).

You can access Intel Aero with USB-ethernet.

But it's an advanced method, use the wireless method instead.

```
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPL
options=4<VLAN_MTU>
ether b6:f9:2a:f4:77:ec
inet6 fe80::14ce:db43:c5fc:3150%en7 pref
inet 192.168.7.21 netmask 0xfffff000 bro
nd6 options=201<PERFORMNUD,DAD>
media: autoselect (100baseTX <full-duple
status: active
[grisbi:~ paulguermontprez$ ping 192.168.7.2
PING 192.168.7.2 (192.168.7.2): 56 data bytes
64 bytes from 192.168.7.2: icmp_seq=0 ttl=64 tim
```

Internet wireless connectivity

By default, Intel Aero is acting as **Wifi access point** but for development, you'll want to **connect Intel Aero to your Wifi** network, to access internet, install packages and access Aero remotely from your development computer.

This action is done using the keyboard, mouse and screen connected to Intel Aero (or with the USB-ethernet method). After this step, you'll access Aero remotely and won't need them anymore.

For details, consult:

<https://github.com/intel-aero/meta-intel-aero/wiki/08-Aero-Network-and-System-Administration#networking-internet-access>

What Intel Aero's IP?

If you are still in **Wifi access point** mode, it's 192.168.8.1

If you joined your **Wifi network as client**, it's been allocated by your DHCP server (see the DHCP section of your router/access point).

- Intel Aero is proposing a hostname on the network. If your computer supports this feature, you'll be able to use the name instead of the IP.

Ex: "ssh root@intel-aero.local"

- While you are still connected to Aero with the HDMI screen and keyboard, run the command "ip addr" to get your IP.
Ex: On my network it's 192.168.1.105
- You can ask your system administrator to make a DHCP reservation to have a constant IP.

```
root@intel-aero:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: usb0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether de:34:ec:8d:79:c2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.7.2/24 brd 192.168.7.255 scope global usb0
        valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1
    link/sit 0.0.0.0 brd 0.0.0.0
4: wlp1s0: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:c5:89:09:67:da brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.105/24 brd 192.168.1.255 scope global wlp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::a2c5:89ff:fe09:67da/64 scope link
        valid_lft forever preferred_lft forever
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:64:c5:d5:c5 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
        valid_lft forever preferred_lft forever
root@intel-aero:~#
```

ssh connectivity

Once Intel Aero is connected to the same network as our development station, we can connect remotely with **ssh**, copy files with **scp**.

- On Windows, you can use putty for ssh and filezilla for scp.
- On Mac and Linux, it's included.

Set a new password!

You're now connected over ssh

```
grisbi:~ paulguermonprez$ ssh root@192.168.1.105
The authenticity of host '192.168.1.105 (192.168.1.105)' can't be established.
ECDSA key fingerprint is SHA256:mCrpPFCLrEBpxNYofThd6mKsI273nl53SgiEWtwuA94.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.105' (ECDSA) to the list of known hosts.
[root@intel-aero:~# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@intel-aero:~# aero-get-version.py

BIOS_VERSION = Aero-01.00.12_Prod
OS_VERSION = Poky Aero (Intel Aero Linux Distro) 1.4.0-dev (pyro)"
AIRMAP_VERSION = 1.8
FPGA_VERSION = 0xc0

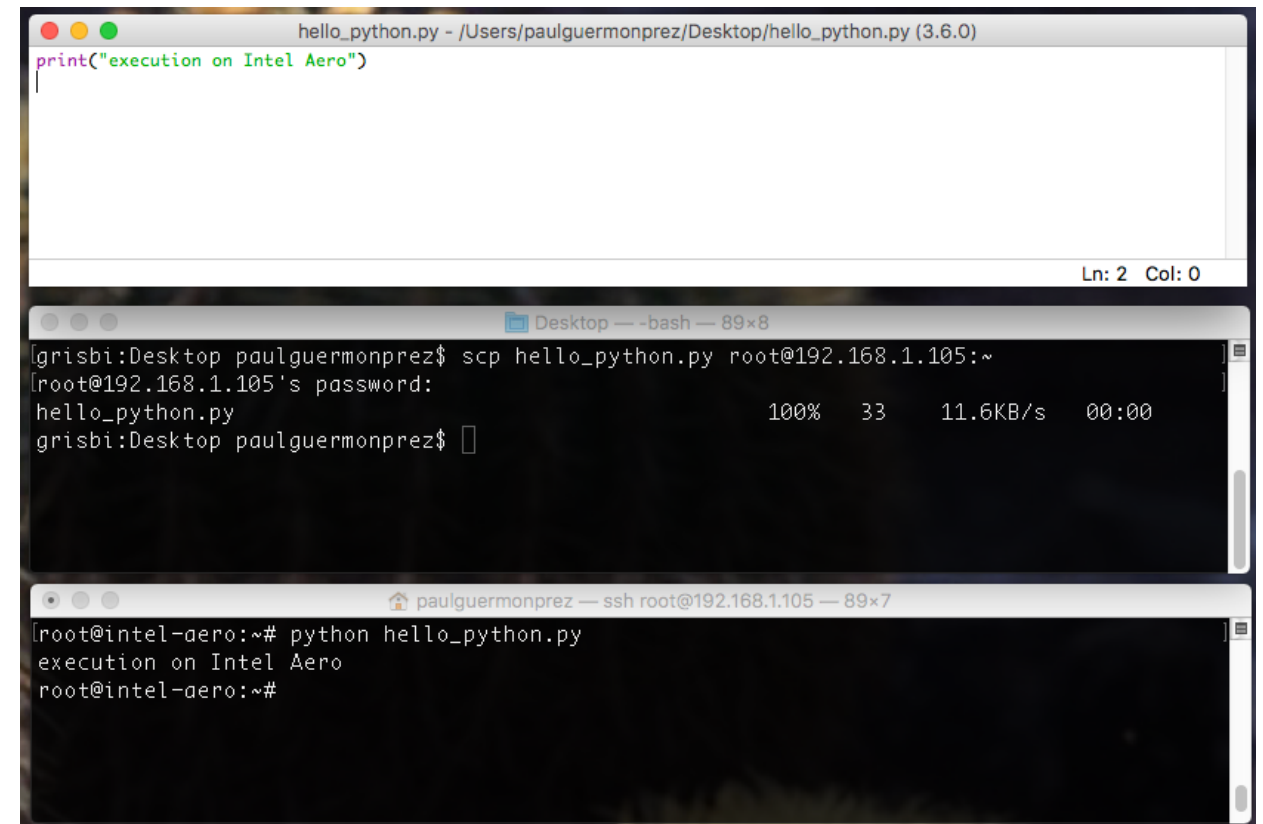
root@intel-aero:~#
```

Development methods

You can edit your source files, compile and execute **remotely on Intel Aero** over ssh. As an example: vi, python, make, gcc are included.

It is also possible to **edit sources on your station** and transfer them to Intel Aero for execution.

In this capture, I used IDLE editor on Mac, transferred the file with scp and executed on Aero.



```
hello_python.py - /Users/paulguermonprez/Desktop/hello_python.py (3.6.0)
print("execution on Intel Aero")
Ln: 2 Col: 0

Desktop — -bash — 89x8
[grisbi:Desktop paulguermonprez$ scp hello_python.py root@192.168.1.105:~
root@192.168.1.105's password:
hello_python.py                                100% 33 11.6KB/s 00:00
grisbi:Desktop paulguermonprez$ ]

paulguermonprez — ssh root@192.168.1.105 — 89x7
[root@intel-aero:~# python hello_python.py
execution on Intel Aero
root@intel-aero:~#]
```

Video recording - Yocto networking basics

Development methods - Ubuntu

If you installed Ubuntu, you have a full desktop running on Intel Aero.

You can use it to:

- Connect to WiFi networks
- Develop code with graphical IDEs

Video recording - Ubuntu networking basics

Conclusion

We:

- Connected Intel Aero to our **Wifi** network
- Discovered **nmcli**, a powerful networking tool
- Have **ssh** access from our workstation
- Can edit, compile, execute

We're ready to code!

Thanks

Paul.Guermonprez@intel.com

<https://intel-aero.github.io>

Released under Creative Commons-BY
creativecommons.org/licenses/by/2.0/