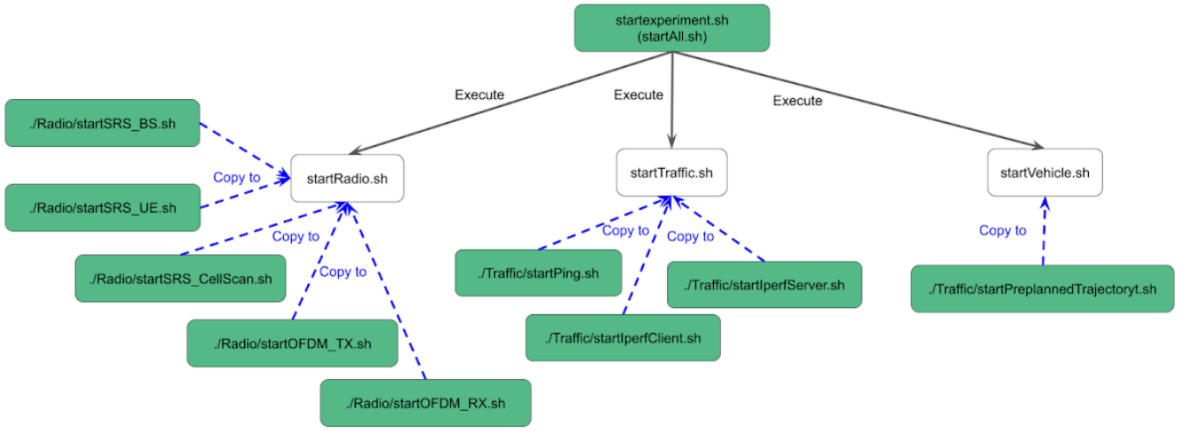
AERPAW hands-on tutorial

This tutorial assumes you have LW1 as node 1 and LPN1 as node 2.

Designing an experiment in AERPAW requires you to modify startexperiment.sh script.

A quick way to design an experiment is to choose from existing sample experiments.



You first copy a sample script to create startRadio.sh then uncomment the corresponding line in startexperiment.sh. Similarly for startVehicle.sh (or startTraffic.sh)

```
root@E-VM-M1:~$ cat startexperiment.sh
#!/bin/bash
/root/stopexperiment.sh

source /root/.ap-set-experiment-env.sh
source /root/.bashrc

export AERPAW_REPO=${AERPAW_REPO:-/root/AERPAW-Dev}
export AERPAW_PYTHON=${AERPAW_PYTHON:-python3}
export PYTHONPATH=/usr/local/lib/python3/dist-packages/
export EXP_NUMBER=${EXP_NUMBER:-1}

if [ "$AP_EXPENV_THIS_CONTAINER_NODE_VEHICLE" == "vehicle_uav" ]; then
    export VEHICLE_TYPE=drone
elif [ "$AP_EXPENV_THIS_CONTAINER_NODE_VEHICLE" == "vehicle_ugv" ]; then
    export VEHICLE_TYPE=rover
else
    export VEHICLE_TYPE=none
fi

if [ "$AP_EXPENV_SESSION_ENV" == "Virtual" ]; then
    export LAUNCH_MODE=EMULATION
elif [ "$AP_EXPENV_SESSION_ENV" == "Testbed" ]; then
    export LAUNCH_MODE=TESTBED
else
    export LAUNCH_MODE=none
fi

export RESULTS_DIR="${RESULTS_DIR:-/root/Results}"
export TS_FORMAT="${TS_FORMAT:-'[%Y-%m-%d %H:%M:%.S]'}"
export LOG_PREFIX="$(date +%Y-%m-%d_%H_%M_%S)"

export TX_FREQ=3.32e9
export RX_FREQ=3.32e9


export PROFILE_DIR=$AERPAW_REPO"/AHN/E-VM/Profile_software"
cd $PROFILE_DIR"/ProfileScripts"

#./Radio/startRadio.sh
#./Traffic/startTraffic.sh
#./Vehicle/startVehicle.sh

schedule_stop.sh 30
```

Experiment 1:

In this experiment, you will set up LW1 as channel sounder transmitter and LPN1 as channel sounder receiver

User manual page: [GE2: Channel Sounder]

ssh into nodes

LW1

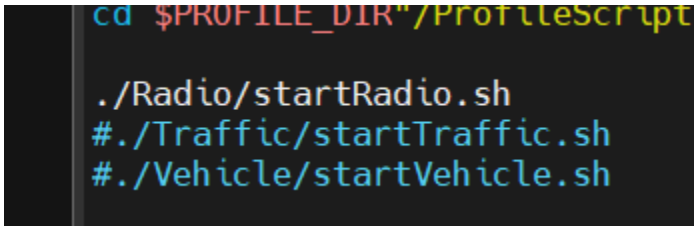
```
ssh -i /drives/c/Users/oozdemi/.ssh/id_rsa root@192.168.133.1
```

LPN1

```
ssh -i /drives/c/Users/oozdemi/.ssh/id_rsa root@192.168.133.2
```

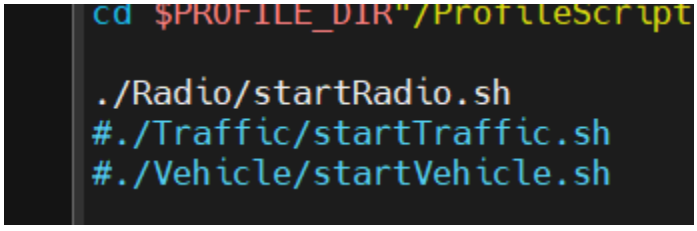
Node 1 (LW1)

```
cd /root/Profiles/ProfileScripts/Radio
cp Samples/startGNURadio-ChannelSounder-TX.sh startRadio.sh
nano /root/startexperiment.sh -> (uncomment ./Radio/startRadio.sh)
```



Node 2 (LPN1)

```
cd /root/Profiles/ProfileScripts/Radio
cp Samples/startGNURadio-ChannelSounder-RX.sh startRadio.sh
nano /root/startexperiment.sh -> (uncomment ./Radio/startRadio.sh)
```



You have now designed your experiment, now let us run it

Log in to oeo-console

```
ssh -i /drives/c/Users/oozdemi/.ssh/id_rsa -L 5760:127.0.0.1:5760
root@192.168.133.62
```

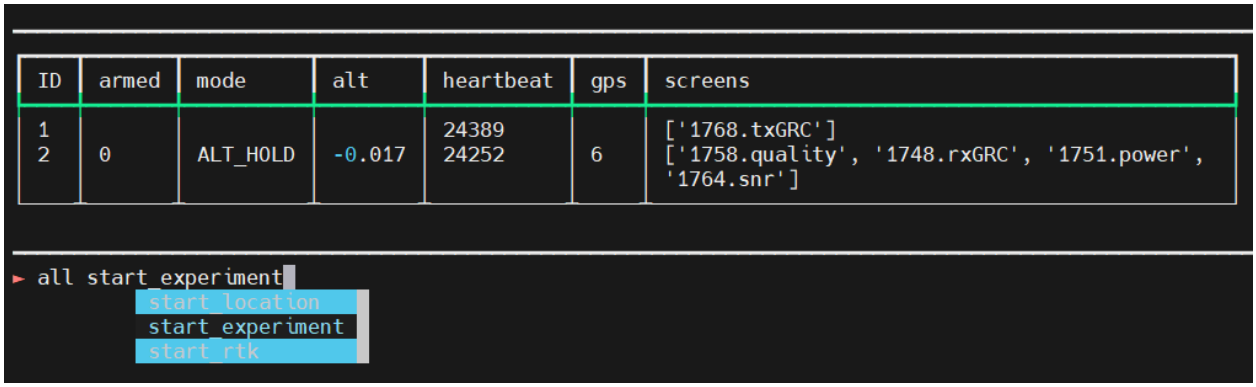
Run

```
./startOEOConsole.sh
```

Then run

```
all start_experiment
```

You should see



Go to node 2 and type the following to check if you can see power measurements

```
screen -r power
```

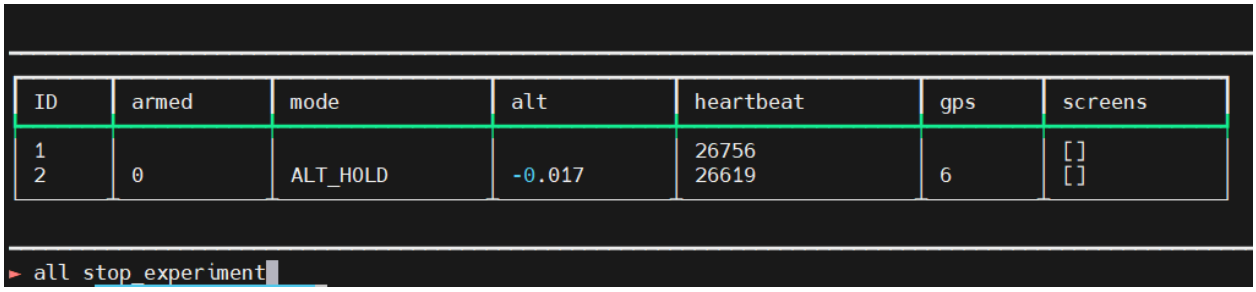
To go to the main terminal type

```
Ctrl-A then Ctrl-D
```

To stop an experiment type

```
all stop_experiment
```

The screens should disappear



Finally, it is a good idea to reset the experiment by running

```
/root/reset.sh
```

either from node 1 or node 2, which resets the vehicle and channel emulator into the initial state. This may avoid some problems that you may encounter when you run an experiment later.

```
root@E-VM-M1:~$ /root/reset.sh
Resetting vehicle and channel emulator, please wait...
100%|
root@E-VM-M1:~$
```

When an experiment is run, the logs created are saved under /root/Results in each node. Go to /root/Results folder in each node and type

`ls -alth`

to check if the logs from this experiment exist.

```
root@E-VM-M1:~/Results$ ls -alth
total 20K
-rw-r--r-- 1 root root 1.7K May 13 23:56 2026-05-13_23_49_56_radio_channelsoundertxgrc_log.txt
drwxr-xr-x 1 root root 4.0K May 13 23:49 .
-rw-r--r-- 1 root root   69 May 13 23:49 experiment_schedule.log
drwx------ 1 root root 4.0K May 13 23:49 ..
```

```
root@E-VM-M2:~$ cd Results/
root@E-VM-M2:~/Results$ ls -alth
total 1.9M
drwx------ 1 root root 4.0K May 13 23:56 ..
-rw-r--r-- 1 root root 2.1K May 13 23:56 2026-05-13_23_49_56_radio_channelsounderrxgrc_log.txt
-rw-r--r-- 1 root root 620K May 13 23:56 2026-05-13_23_49_56_power_log.txt
-rw-r--r-- 1 root root 620K May 13 23:56 2026-05-13_23_49_56_snr_log.txt
-rw-r--r-- 1 root root 620K May 13 23:56 2026-05-13_23_49_56_quality_log.txt
drwxr-xr-x 1 root root 4.0K May 13 23:49 .
-rw-r--r-- 1 root root   69 May 13 23:49 experiment_schedule.log
```

Experiment 2:

Now, let us add a preplanned trajectory sample vehicle experiment to Experiment 1 above

User manual page: [Running Sample Applications]

In node 2 (LPN1)

`cd /root/Profiles/ProfileScripts/Vehicle`

`cp Samples/startPreplannedTrajectory.sh startVehicle.sh`

`nano /root/startexperiment.sh` (uncomment `./Vehicle/startVehicle.sh`)

```
./Radio/startRadio.sh
#./Traffic/startTraffic.sh
./Vehicle/startVehicle.sh
```

When we ssh into oo-console for experiment 1 we used

`ssh -i /drives/c/Users/oozdemi/.ssh/id_rsa -L 5760:127.0.0.1:5760 root@192.168.133.62`

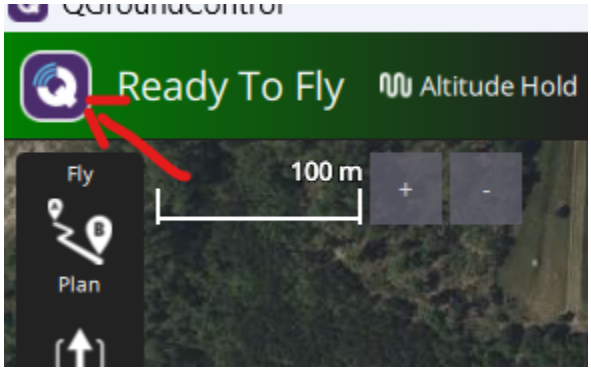
Here `-L 5760:127.0.0.1:5760` is added for port forwarding so that we can view the vehicle in locally running QGroundcontrol

User manual page: Port Forwarding

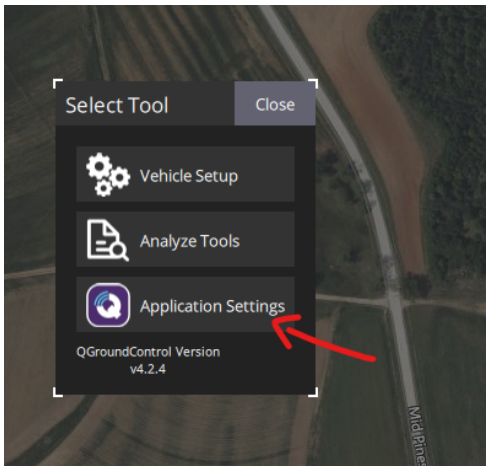
See also: QGroundControl Local Installation

QGC is configured as follows

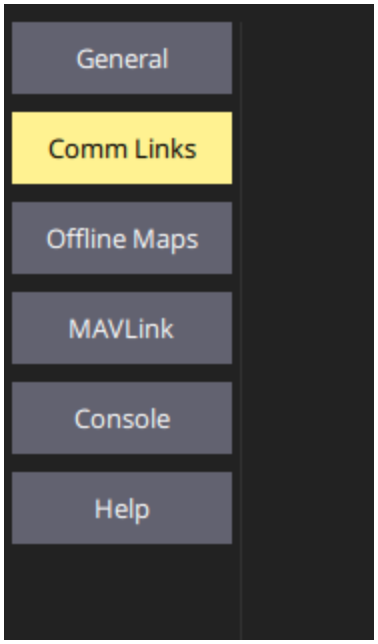
Click



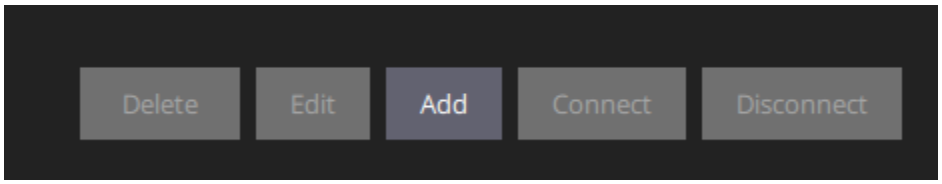
Application Settings

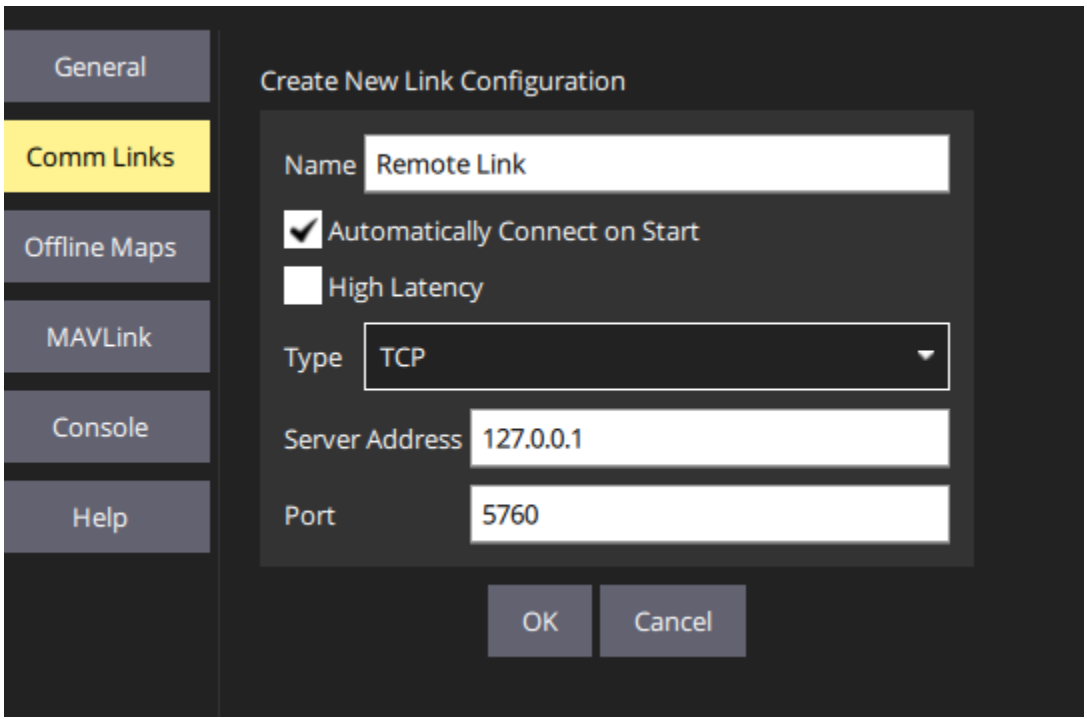


Comm Links



Add

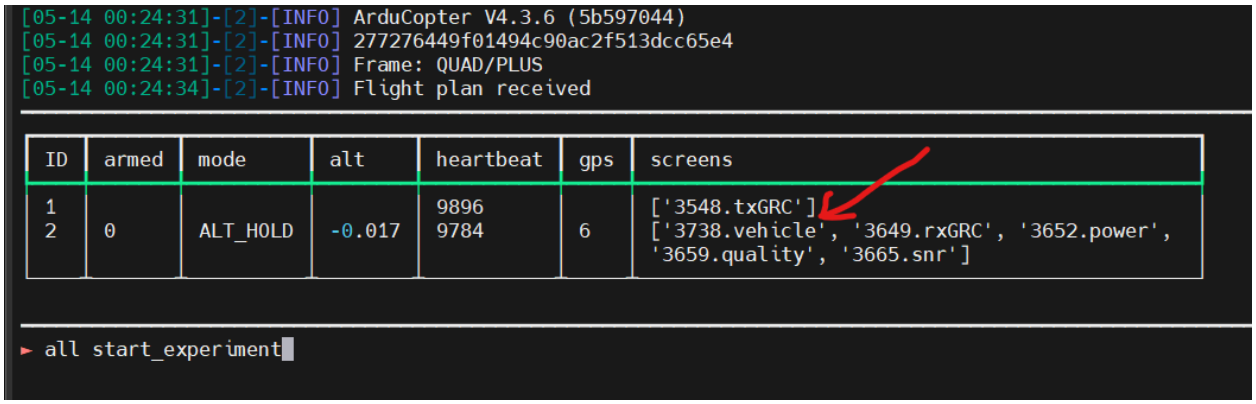




Click OK then you should see

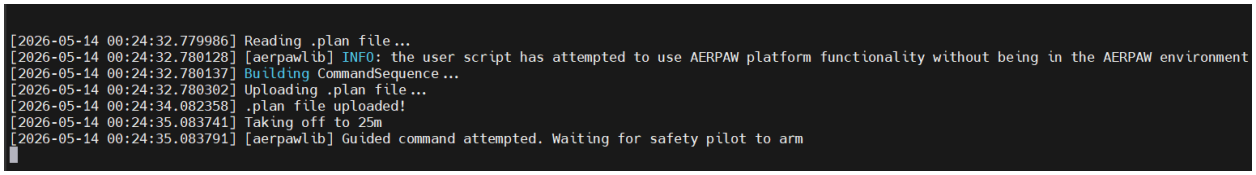


Start experiment from oeo-console



Go to node 2 and type

`screen -r vehicle`



`ctrl+A - ctrl+D` to go back to main terminal

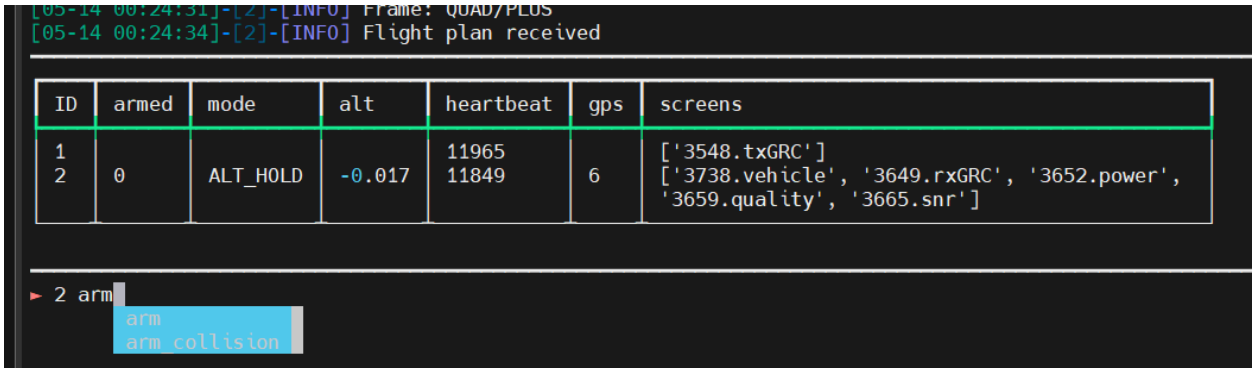
Type

`screen -r power`

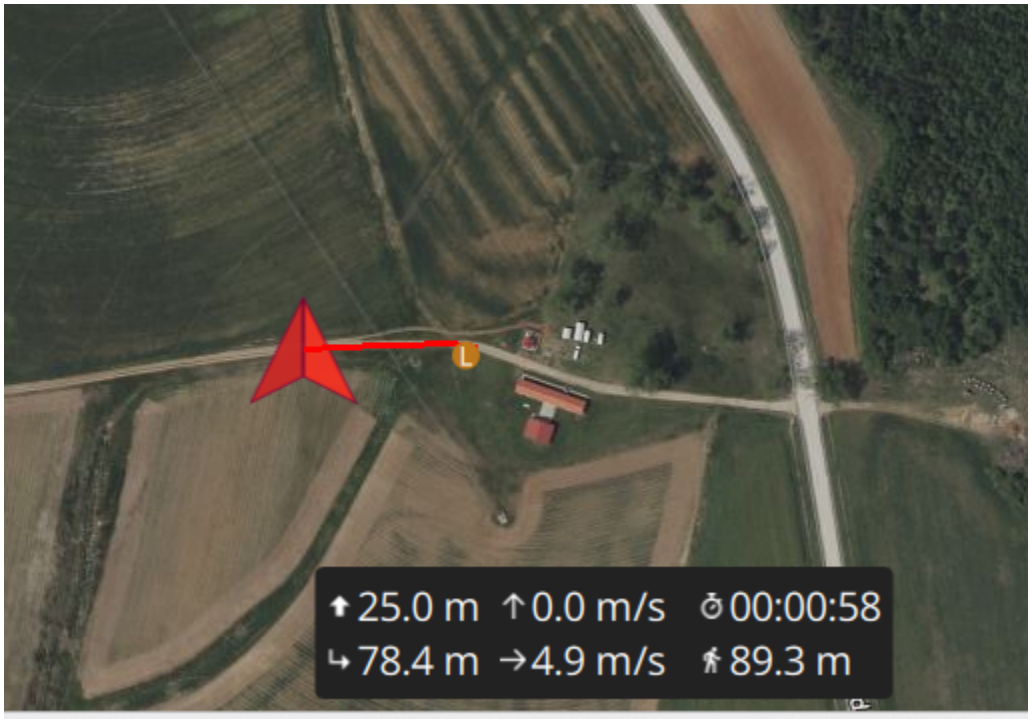
The vehicle needs to be armed for the UAV to take off. You do it by typing

```
2 arm
```

In oeo-console



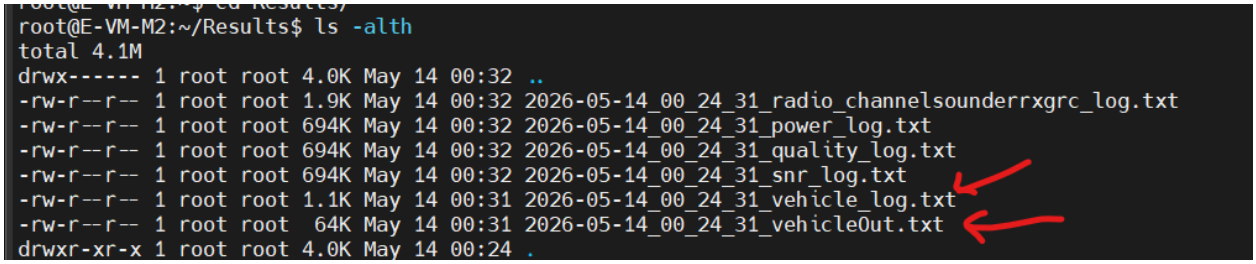
Watch the UAV to take off and follow the default preplanned trajectory in QGC



Also watch the power screen to see if power value decreases when the distance between the transmitter and the receiver gets larger and vice versa.

After the vehicle lands stop and reset the experiment.

In node 2 Results folder you should see two log files corresponding to the vehicle, check what they contain



AERPAW provides post-processing scripts to plot measurements as a function of geographic location

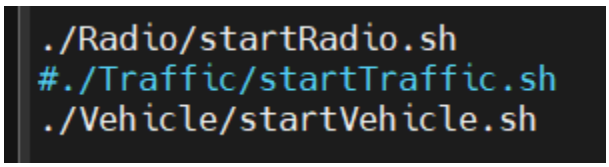
Experiment 3:

In this experiment you will set up LPN1 as LTE base station with a preplanned trajectory and LPN1 as LTE user equipment.
User manual pages: SE1: Multi-Node LTE SISO and E2E1: Single-UAV LTE eNB Serving Ground UEs
ssh into node 2 (LPN1)
Radio

```
cd /root/Profiles/ProfileScripts/Radio
cp Samples/startSRSRAN-SISO-EPCandENB.sh startRadio.sh
nano /root/startexperiment.sh -> (uncomment ./Radio/startRadio.sh)
```

Vehicle

```
cd /root/Profiles/ProfileScripts/Vehicle
cp Samples/startPreplannedTrajectory.sh startVehicle.sh
nano /root/startexperiment.sh (uncomment ./Vehicle/startVehicle.sh)
```



ssh into node 1 (LW1)
Radio

```
cd /root/Profiles/ProfileScripts/Radio
cp Samples/startSRSRAN-SISO-UE.sh startRadio.sh
nano /root/startexperiment.sh -> (uncomment ./Radio/startRadio.sh)
```

Traffic

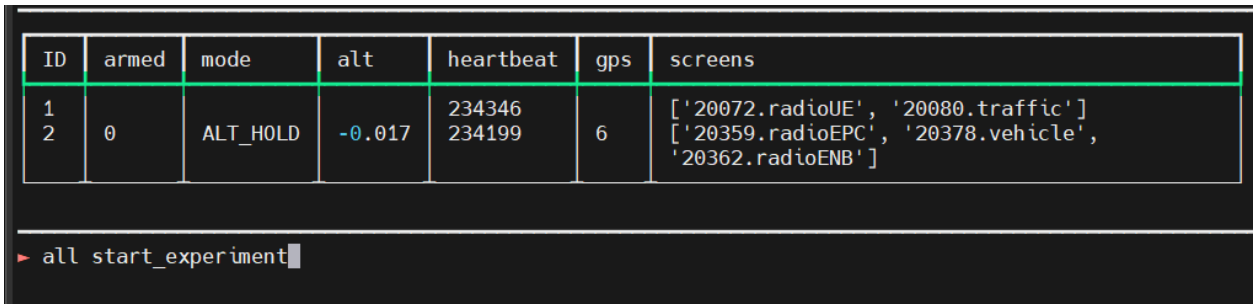
```
cd /root/Profiles/ProfileScripts/Traffic
```

`cp Samples/startPing.sh startTraffic.sh`

`nano /root/startexperiment.sh -> (uncomment ./Radio/startRadio.sh)`

```
./Radio/startRadio.sh
./Traffic/startTraffic.sh
#./Vehicle/startVehicle.sh
```

Start experiment from oeo-console

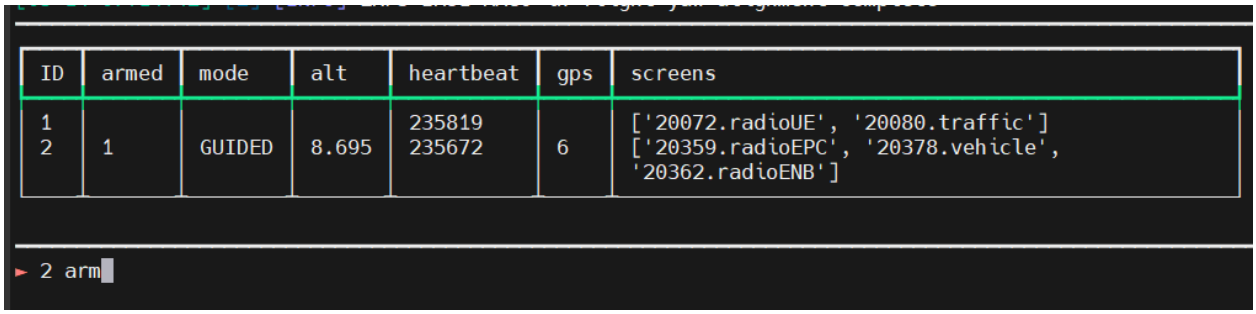


Go to node 1 and check the traffic screen

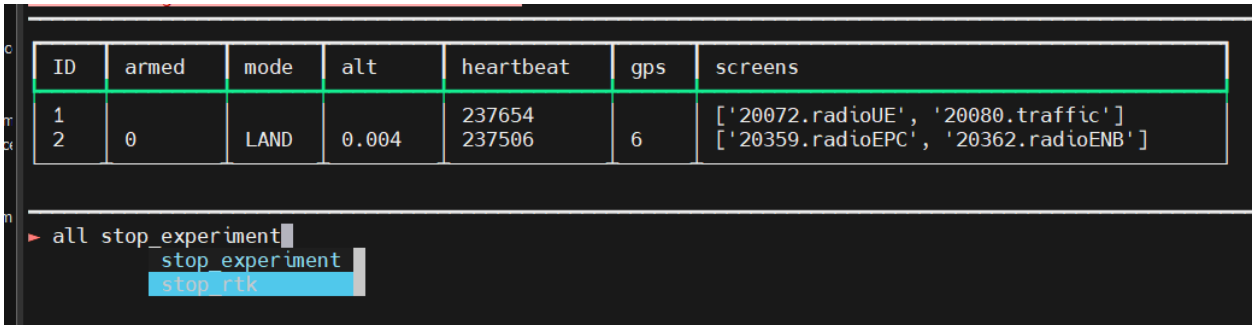
`screen -r traffic`

Arm the vehicle

`2 arm`



Follow the vehicle moving in QGC. Check other screens in each of the nodes.
Once the vehicle lands the vehicle screen will terminate at that point stop experiment wait for the screens to disappear and reset experiment.



Check the logs generated.
Now try the same experiment but with a different plan file

ssh into node 2 (LPN1)
`nano /root/Profiles/ProfileScripts/Vehicle/startVehicle.sh`
Change the following line
from:

```
export
MISSION=$PROFILE_DIR"/vehicle_control/PreplannedTrajectory/Missions/default.pl
an"
```

to:

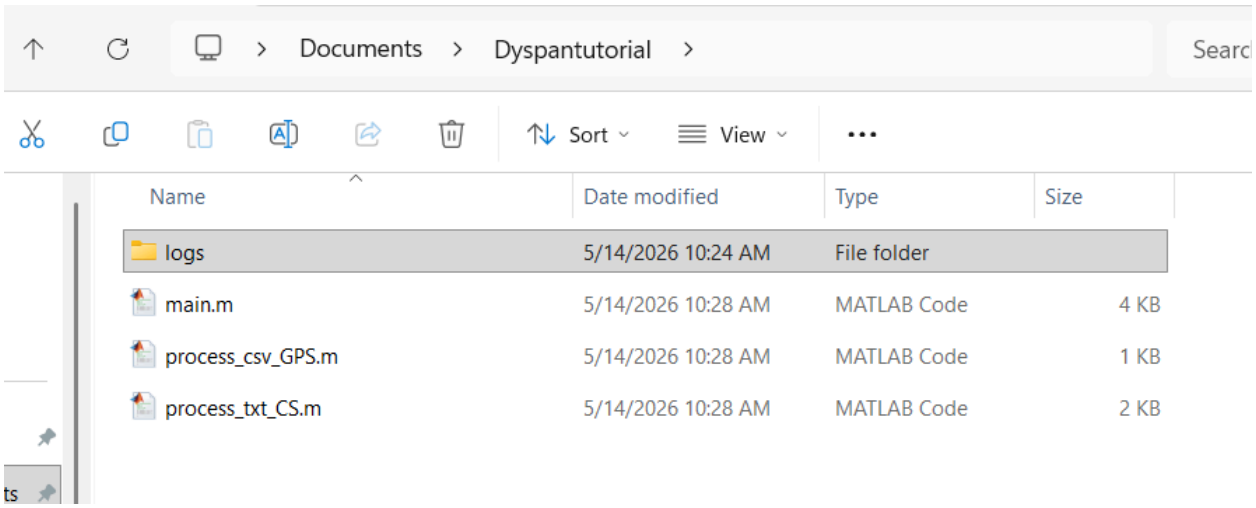
```
export
MISSION=$PROFILE_DIR"/vehicle_control/PreplannedTrajectory/Missions/lw1-to-lw2
.plan"
```

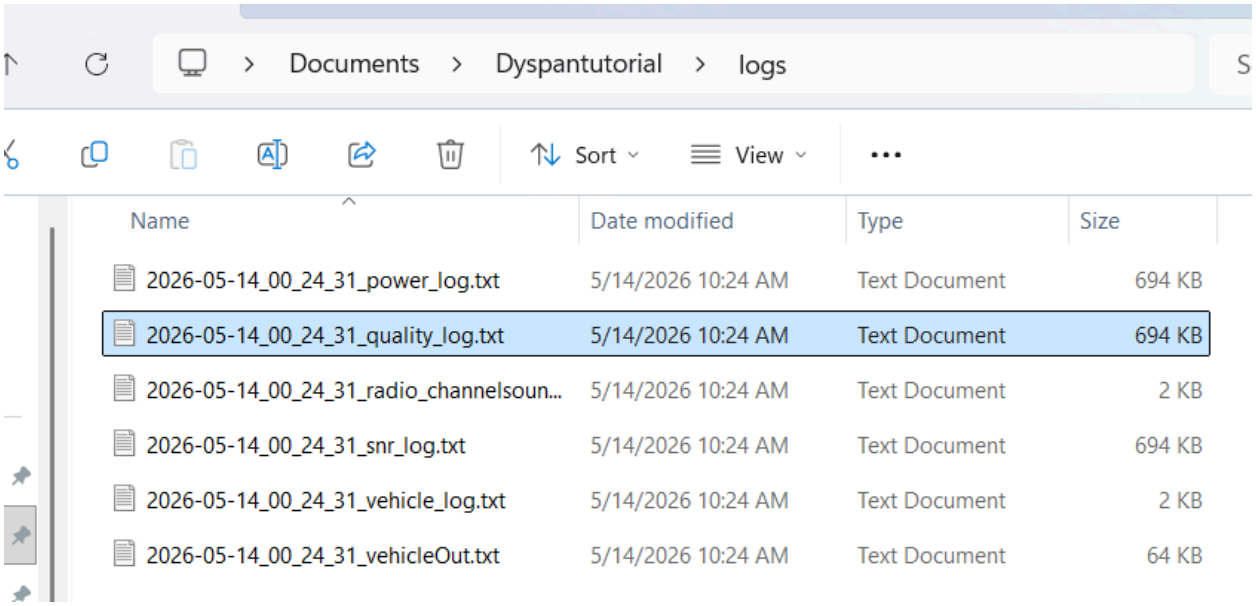
Run the experiment again. Note that this time the vehicle moves from LW1 to LW2 and comes back instead of following the default plan.

Optional: Postprocessing
User manual Page: [E2E2: Channel Sounder and Preplanned Trajectory with Post-processing]
Download logs from your experiment
You can drag and drop in Mobaxterm or you can use scp

```
scp -i /drives/c/Users/oozdemi/.ssh/id_rsa
root@192.168.133.2:/root/Results/2026-05-14_00_24_31_vehicleOut.txt
/drives/c/Users/oozdemi/Documents/
```

Download Matlab code here
Create a new folder and place the files as shown below





Edit main.m to process the logs you generated

```
%Replace log files with the new logs you generated
fname_gps='logs/2026-05-14_00_24_31_vehicleOut.txt';
fname_power="logs/2026-05-14_00_24_31_power_log.txt";
fname_quality="logs/2026-05-14_00_24_31_quality_log.txt";
```

Run main.m. This should generate plots similar to plots in page [E2E2: Channel Sounder and Preplanned Trajectory with Post-processing]().