

## Running SILT.

In order to run dancing swarm the following requirements must be met:

Operation System: Ubuntu 16.04 or later

### Preparing environment:

If your operation system is 64 bit, use sitl64.zip

1. Unpack provided archive into your working directory:
  - a. **unzip sitl(64).zip -d SITL**
2. SITL – is a directory name there you want sitl files to be placed.
3. Enter directory and add execution permissions:
  - a. **cd SITL**
  - b. **chmod +x swarm.bash Tools/autotest/run\_in\_terminal\_window.sh build/sitl/bin/arducopter**
4. Install required packages or make sure there are already installed:
  - a. **sudo apt-get install python-pip**
  - b. **sudo pip install future pymavlink MAVProxy**
  - c. **sudo apt-get install gnome-terminal**

### Configuring swarm size:

Open **swarm.bash** with any text editor. Find the lines :

```
# Range of system ids to use.  
SYSID_FIRST=1  
SYSID_LAST=20
```

This two lines defines the range of drones identities and there amount. Here 20 drone instances will be initialized in a dance swarm.

### Configuring drones positions:

Open **locations.py** with any text editor, and find the following part:

```
Y_count = 7  
X_count = 5  
Distance_between_drones = 3 # meters  
  
CNT_LAT = 56.973501 # Latitude of Drone formation center  
CNT_LON = 24.071459 # Longitude of Drone formation center  
Heading = 0  
Altitude = 1
```

This file generates initial location of each drone in swarm, to avoid all vehicles to be placed in a single point. If you run this script **python location.py** The following out will be displayed.

```
S1=56.9734471011,24.0713106617,1,202  
S2=56.9734471011,24.0713601078,1,78
```

S3=56.9734471011,24.0714095539,1,0

...

You need to insert this output into **Tools/autotest/locations.txt**

Run the swarm

Execute

**./swarm.bash**

Wait until all instances got initialized and "Press any key..." in "swarm.bash" terminal to terminate all processes. First run is required for setting up sitl environments for each drone and is not valid for swarm operations yet.

As a result new directory will be created **ddc-cache**. It contains subdirs for each drone identity. (1, 2, 3, 4....) Each drone keeps its logs, parameters and mission paths in its corresponding folder.

Run swarm again

Script will start instances. Each instance became available on a tcp ports following the formula:

**port= 5760 + I\*10 + 2**

For example:

5772 for drone with id= 1

5782 for drone with id= 2

5792 for drone with id= 3

5802 for drone with id= 4

In order to connect DDC-Client to instances, VSM should be configured respectively:

You can use **vsm.py** script to optimize configuration generation. If sitl instances running on a host **192.168.1.39**, and their number is 5, execute the following line:

**python vsm.py --host 192.168.1.39 --num 5**

Copy an output and place it into your **vsm-ardupilot.config** file.

**# N 1**

**connection.tcp\_out.1.port = 5772**

**connection.tcp\_out.1.address = 192.168.1.39**

**# N 2**

**connection.tcp\_out.2.port = 5782**

**connection.tcp\_out.2.address = 192.168.1.39**

**# N 3**

**connection.tcp\_out.3.port = 5792**

**connection.tcp\_out.3.address = 192.168.1.39**

**# N 4**

**connection.tcp\_out.4.port = 5802**

**connection.tcp\_out.4.address = 192.168.1.39**