**Problem**: Quadcopter gets armed, gets GPS location but I am not able to spin motors or lift it even one meter, I have tested udacidrone, dronekit-python, mavproxy.py, ardupilot in all cases it is not able to takeoff.

Definitely I am doing something wrong but after days of work I am still not able to figure out what should be changed and what I have to modify. I need help from experts in this area. On web I have not seen implementation of Jetson nano with quadcopter however NVIDIA’s redtail github (<https://github.com/NVIDIA-AI-IOT/redtail>) supports jetsonTX1 and TX2. There is some initiative for jetson nano (<https://github.com/mtbsteve/redtail>) however its not ready to implement and test.

I have tested this F450 assembly using FS-i6 transmitter and it fly’s , so there is no issue with quadcopter , it is something related to connection between jetson-nano and pixhawk .

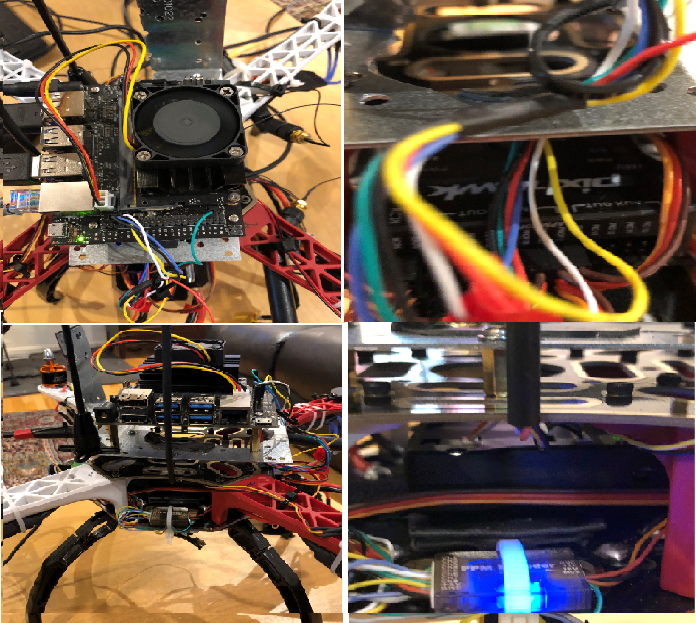
***I NEED HELP AND GUIDANCE FOR*** *jetson-nano and pixhawk implementation on Quadcopter (please do not point me towards available solution for TX1, TX2, XAVIER or Pi4, I am exclusively trying to implement with nano). I have used the same jetson-nano for jetbots and door camera etc. so i am sure there is no problem or issue with jetson-nano*

I think

Companion computer: Jetson-nano

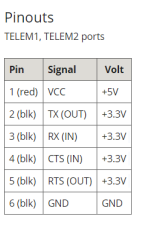
Quadcopter: F450 Frame + PXI PX4 Flight Control + 920KV Motor + GPS + FS-i6 Transmitter (<https://www.amazon.com/gp/product/B01HEQQDNK/ref=ppx_yo_dt_b_asin_title_o06_s00?ie=UTF8&psc=1>) .

Jetson nano and F450 assembly photo’s/pictures

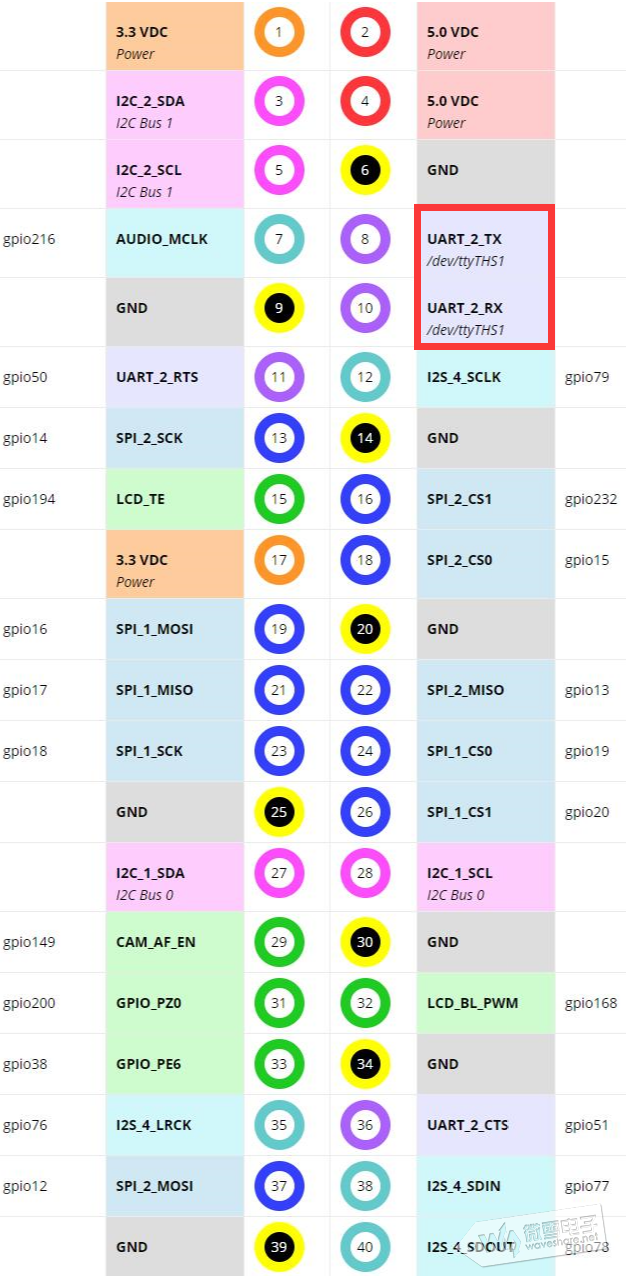


Jetson Nano has been connected to Pixhawk using Telem2.

PIXHAWK Telem2: Below is the connection point for Telem2 and Telem1



Below is the connection for jetson-nano J41

µ §

**Jetson-nano-j41 and Pixhawk Telem2 connection details:**

Jetson -nano j-41 power pin (#2 or 4) is not connected to Pixhawk (pin 1 ) .

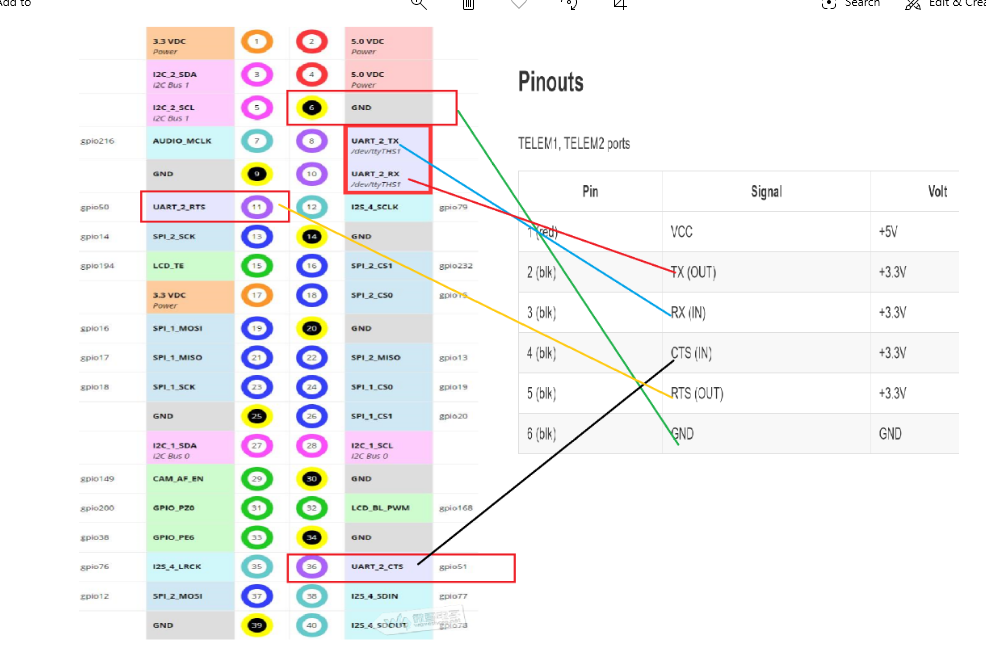
Following are my connection details :

1. GND jetson-nano (pin #4) is connected to GND of Pixhawk (6th pin)
2. TX of jetson nano (8th pin) is connected to RX of Pixhawk(3rd pin)
3. RX of jestson nano is connected to TX of Pixhawk (2nd pin)
4. RTS Pin 11 of jetson nano is connected to RTS of Pixhawk (pin #5)
5. CTS pin 36 of jetson nano is connected to CTS of Pixhawk (4th pin)

Pixhawk gets its power sully from quadcopter battery and Jetson nano has a separate (5v 4 amps) power supply.

I have tested all the below mentioned code/program using 3 pin (connecting GND, RX and TX of both systems ) and 5 pin ((connecting GND, RX ,TX ,RTS,CTS of both systems) connection with no success of takeoff.

Note:- By connecting Jetson-nano TX to Pixhawk TX and RX of Jetson to RX of Pixhawk , I was not able to communicate with drone as a result I have used TX to RX and RX to TX.



**DRONEKIT-PYTHON**

For connectivity testing between companion computer (jetson nano) and flight controller I have used dronekit and results have been satisfactory

**check\_drone\_kit.py**

*from dronekit import connect*

*# Connect to the Vehicle*

*vehicle = connect('/dev/ttyTHS1', wait\_ready=True, baud=57600)*

*# vehicle is an instance of the Vehicle class*

*print('Autopilot Firmware version: %s' % vehicle.version)*

*print('Autopilot capabilities (supports ftp): %s' % vehicle.capabilities.ftp)*

*print('Global Location: %s' % vehicle.location.global\_frame)*

*print('Global Location (relative altitude): %s' % vehicle.location.global\_relative\_frame)*

*print('Local Location: %s' % vehicle.location.local\_frame) #NED*

*print('Attitude: %s' % vehicle.attitude)*

*print('Velocity: %s' % vehicle.velocity)*

*print('GPS: %s' % vehicle.gps\_0)*

*print('Groundspeed: %s' % vehicle.groundspeed)*

*print('Airspeed: %s' % vehicle.airspeed)*

*print('Gimbal status: %s' % vehicle.gimbal)*

*print('Battery: %s' % vehicle.battery)*

*print('EKF OK?: %s' % vehicle.ekf\_ok)*

*print('Last Heartbeat: %s' % vehicle.last\_heartbeat)*

*print('Rangefinder: %s' % vehicle.rangefinder)*

*print('Rangefinder distance: %s' % vehicle.rangefinder.distance)*

*print('Rangefinder voltage: %s' % vehicle.rangefinder.voltage)*

*print('Heading: %s' % vehicle.heading)*

*print('Is Armable?: %s' % vehicle.is\_armable)*

*print('System status: %s' % vehicle.system\_status.state)*

*print('Mode: %s' % vehicle.mode.name) # settable*

*print('Armed: %s' % vehicle.armed) # settable*

dlinano@jetson-nano:~$ sudo python3 check\_drone\_kit.py

[sudo] password for dlinano:

Autopilot Firmware version: APM:Copter-3.4.6

Autopilot capabilities (supports ftp): False

Global Location: LocationGlobal:lat=39.0455337,lon=-77.5109696,alt=102.69

Global Location (relative altitude): LocationGlobalRelative:lat=39.0455337,lon=-77.5109696,alt=-0.96

Local Location: LocationLocal:north=4.214805603027344,east=-6.181936740875244,down=-2.8174514770507812

Attitude: Attitude:pitch=-0.008600138127803802,yaw=2.4240646362304688,roll=0.011644798330962658

Velocity: [0.19, -0.38, 0.01]

GPS: GPSInfo:fix=3,num\_sat=9

Groundspeed: 0.3061045706272125

Airspeed: 0.3061045706272125

Gimbal status: Gimbal: pitch=None, roll=None, yaw=None

Battery: Battery:voltage=0.0,current=None,level=None

EKF OK?: True

Last Heartbeat: 0.05023435199998971

Rangefinder: Rangefinder: distance=None, voltage=None

Rangefinder distance: None

Rangefinder voltage: None

Heading: 138

Is Armable?: True

System status: STANDBY

Mode: POSHOLD

Armed: False

dlinano@jetson-nano:~$

**mavproxy.py to arm quadcopter and this test also has been successful**

dlinano@jetson-nano:~$ sudo mavproxy.py --master=/dev/ttyTHS1 --baudrate 57600 --aircraft Copter-3.3

Connect /dev/ttyTHS1 source\_system=255

no script Copter-3.3/mavinit.scr

Log Directory: Copter-3.3/logs/2020-02-01/flight10

Telemetry log: Copter-3.3/logs/2020-02-01/flight10/flight.tlog

Waiting for heartbeat from /dev/ttyTHS1

MAV> GPS lock at -4 meters

online system 1

POSHOLD> Mode POSHOLD

fence breach

APM: APM:Copter V3.4.6 (e707341b)

APM: PX4: de6b667d NuttX: 8c965992

APM: Frame: QUAD

APM: PX4v2 002A001F 34385113 30323534

Received 593 parameters

Saved 593 parameters to Copter-3.3/logs/2020-02-01/flight10/mav.parm

arm throttle

POSHOLD> Got MAVLink msg: COMMAND\_ACK {command : 400, result : 0}

ARMED

**Arming vehicle using Dronekit packages and this has been successful**

*dlinano@jetson-nano:~$ sudo python3*

*[sudo] password for dlinano:*

*Python 3.6.9 (default, Nov 7 2019, 10:44:02)*

*[GCC 8.3.0] on linux*

*Type "help", "copyright", "credits" or "license" for more information.*

*>>> from dronekit import connect, VehicleMode, LocationGlobalRelative*

*>>> from dronekit import connect, Command, LocationGlobal*

*>>> from pymavlink import mavutil*

*>>> import time, sys, argparse, math*

*>>> vehicle = connect('/dev/ttyTHS1', wait\_ready=True, baud=57600)*

*>>> vehicle.mode = VehicleMode("GUIDED")*

*>>> home = vehicle.location.global\_relative\_frame*

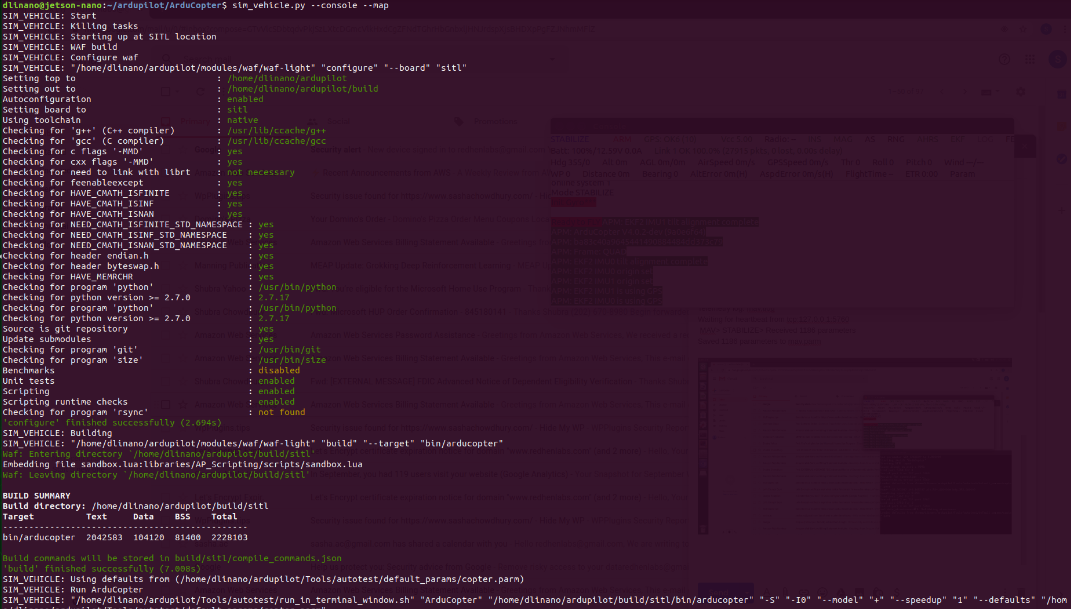
*>>> print(home)*

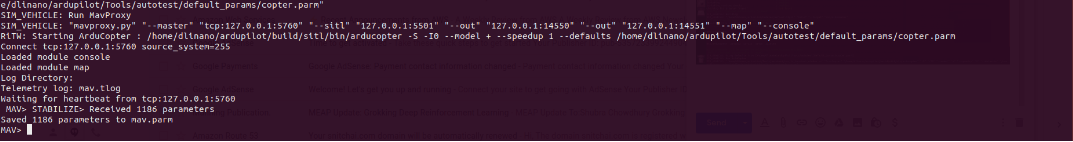
*LocationGlobalRelative:lat=39.0455834,lon=-77.5109233,alt=3.06*

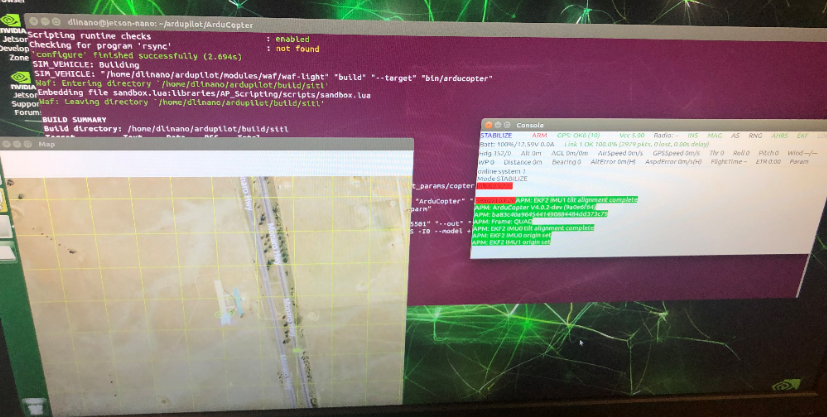
*>>> vehicle.armed=True*

*>>>*

**Testing using ARDUPILOT :** Both sim\_vehicle.py -w and sim\_vehicle.py --console –map executed successfully.







dlinano@jetson-nano:~/ardupilot/ArduCopter$ sim\_vehicle.py --console --map  
SIM\_VEHICLE: Start  
SIM\_VEHICLE: Killing tasks  
SIM\_VEHICLE: Starting up at SITL location  
SIM\_VEHICLE: WAF build  
SIM\_VEHICLE: Configure waf  
SIM\_VEHICLE: "/home/dlinano/ardupilot/modules/waf/waf-light" "configure" "--board" "sitl"  
Setting top to                           : /home/dlinano/ardupilot  
Setting out to                           : /home/dlinano/ardupilot/build  
Autoconfiguration                        : enabled  
Setting board to                         : sitl  
Using toolchain                          : native  
Checking for 'g++' (C++ compiler)        : /usr/lib/ccache/g++  
Checking for 'gcc' (C compiler)          : /usr/lib/ccache/gcc  
Checking for c flags '-MMD'              : yes  
Checking for cxx flags '-MMD'            : yes  
Checking for need to link with librt     : not necessary  
Checking for feenableexcept              : yes  
Checking for HAVE\_CMATH\_ISFINITE         : yes  
Checking for HAVE\_CMATH\_ISINF            : yes  
Checking for HAVE\_CMATH\_ISNAN            : yes  
Checking for NEED\_CMATH\_ISFINITE\_STD\_NAMESPACE : yes  
Checking for NEED\_CMATH\_ISINF\_STD\_NAMESPACE    : yes  
Checking for NEED\_CMATH\_ISNAN\_STD\_NAMESPACE    : yes  
Checking for header endian.h                   : yes  
Checking for header byteswap.h                 : yes  
Checking for HAVE\_MEMRCHR                      : yes  
Checking for program 'python'                  : /usr/bin/python  
Checking for python version >= 2.7.0           : 2.7.17  
Checking for program 'python'                  : /usr/bin/python  
Checking for python version >= 2.7.0           : 2.7.17  
Source is git repository                       : yes  
Update submodules                              : yes  
Checking for program 'git'                     : /usr/bin/git  
Checking for program 'size'                    : /usr/bin/size  
Benchmarks                                     : disabled  
Unit tests                                     : enabled  
Scripting                                      : enabled  
Scripting runtime checks                       : enabled  
Checking for program 'rsync'                   : not found  
'configure' finished successfully (2.694s)  
SIM\_VEHICLE: Building  
SIM\_VEHICLE: "/home/dlinano/ardupilot/modules/waf/waf-light" "build" "--target" "bin/arducopter"  
Waf: Entering directory `/home/dlinano/ardupilot/build/sitl'  
Embedding file sandbox.lua:libraries/AP\_Scripting/scripts/sandbox.lua  
Waf: Leaving directory `/home/dlinano/ardupilot/build/sitl'  
  
BUILD SUMMARY  
Build directory: /home/dlinano/ardupilot/build/sitl  
Target          Text     Data    BSS    Total    
-----------------------------------------------  
bin/arducopter  2042583  104120  81400  2228103  
  
Build commands will be stored in build/sitl/compile\_commands.json  
'build' finished successfully (7.008s)  
SIM\_VEHICLE: Using defaults from (/home/dlinano/ardupilot/Tools/autotest/default\_params/copter.parm)  
SIM\_VEHICLE: Run ArduCopter  
SIM\_VEHICLE: "/home/dlinano/ardupilot/Tools/autotest/run\_in\_terminal\_window.sh" "ArduCopter" "/home/dlinano/ardupilot/build/sitl/bin/arducopter" "-S" "-I0" "--model" "+" "--speedup" "1" "--defaults" "/home/dlinano/ardupilot/Tools/autotest/default\_params/copter.parm"  
SIM\_VEHICLE: Run MavProxy  
SIM\_VEHICLE: "mavproxy.py" "--master" "tcp:[127.0.0.1:5760](http://127.0.0.1:5760/)" "--sitl" "[127.0.0.1:5501](http://127.0.0.1:5501/)" "--out" "[127.0.0.1:14550](http://127.0.0.1:14550/)" "--out" "[127.0.0.1:14551](http://127.0.0.1:14551/)" "--map" "--console"  
RiTW: Starting ArduCopter : /home/dlinano/ardupilot/build/sitl/bin/arducopter -S -I0 --model + --speedup 1 --defaults /home/dlinano/ardupilot/Tools/autotest/default\_params/copter.parm  
Connect tcp:[127.0.0.1:5760](http://127.0.0.1:5760/) source\_system=255  
Loaded module console  
Loaded module map  
Log Directory:  
Telemetry log: mav.tlog  
Waiting for heartbeat from tcp:[127.0.0.1:5760](http://127.0.0.1:5760/)  
 MAV> STABILIZE> Received 1186 parameters  
Saved 1186 parameters to mav.parm

**Test using UDACITY DRONE Packages:**  Connection to drone has been successful and I was able to arm drone. But again no luck with takeoff

*dlinano@jetson-nano:~$*

*dlinano@jetson-nano:~$ sudo python3*

*Python 3.6.9 (default, Nov 7 2019, 10:44:02)*

*[GCC 8.3.0] on linux*

*Type "help", "copyright", "credits" or "license" for more information.*

*>>> from udacidrone import Drone*

*>>> from udacidrone.connection import MavlinkConnection*

*>>> conn = MavlinkConnection('/dev/ttyTHS1,57600',PX4=True, threaded=False)*

*>>> drone = Drone(conn)*

*Logs/TLog.txt*

*>>> quit()*

*dlinano@jetson-nano:~$*

**UDACIDRONE packages used to arm take off landing etc.** It gets armed but no action on rotor/motor, what am I missing: Program stalls at landing transition and never ends and log files are not generated.

I have used the simple flight up and down program (up\_and\_down.py) Project1 link/slide 11 A Simple Flight Plan

dlinano@jetson-nano:~$ cat udaci\_drone.py

import time

from enum import Enum

import numpy as np

from udacidrone import Drone

from udacidrone.connection import MavlinkConnection

from udacidrone.messaging import MsgID

class Phases(Enum):

MANUAL = 0

ARMING = 1

TAKEOFF = 2

LANDING = 3

DISARMING = 4

class UpAndDownFlyer(Drone):

def \_\_init\_\_(self, connection):

super().\_\_init\_\_(connection)

self.target\_position = np.array([0.0, 0.0, 0.0])

self.in\_mission = True

# initial state

self.flight\_phase = Phases.MANUAL

# register all your callbacks here

self.register\_callback(MsgID.LOCAL\_POSITION,

self.local\_position\_callback)

self.register\_callback(MsgID.LOCAL\_VELOCITY,

self.velocity\_callback)

self.register\_callback(MsgID.STATE,

self.state\_callback)

def local\_position\_callback(self):

if self.flight\_phase == Phases.TAKEOFF:

# coordinate conversion

altitude = -1.0 \* self.local\_position[2]

# check if altitude is within 95% of target

if altitude > 0.95 \* self.target\_position[2]:

self.landing\_transition()

def velocity\_callback(self):

if self.flight\_phase == Phases.LANDING:

if ((self.global\_position[2] - self.global\_home[2] < 0.1) and

abs(self.local\_position[2]) < 0.01):

self.disarming\_transition()

def state\_callback(self):

if not self.in\_mission:

return

if self.flight\_phase == Phases.MANUAL:

self.arming\_transition()

elif self.flight\_phase == Phases.ARMING:

self.takeoff\_transition()

elif self.flight\_phase == Phases.DISARMING:

self.manual\_transition()

def arming\_transition(self):

print("arming transition")

self.take\_control()

self.arm()

# set the current location to be the home position

self.set\_home\_position(self.global\_position[0],

self.global\_position[1],

self.global\_position[2])

self.flight\_phase = Phases.ARMING

def takeoff\_transition(self):

print("takeoff transition")

target\_altitude = 3.0

self.target\_position[2] = target\_altitude

self.takeoff(target\_altitude)

self.flight\_phase = Phases.TAKEOFF

def landing\_transition(self):

print("landing transition")

self.land()

self.flight\_phase = Phases.LANDING

def disarming\_transition(self):

print("disarm transition")

self.disarm()

self.flight\_phase = Phases.DISARMING

def manual\_transition(self):

print("manual transition")

self.release\_control()

self.stop()

self.in\_mission = False

self.flight\_phase = Phases.MANUAL

def start(self):

self.start\_log("Logs", "NavLog.txt")

print("starting connection")

super().start()

self.stop\_log()

if \_\_name\_\_ == "\_\_main\_\_":

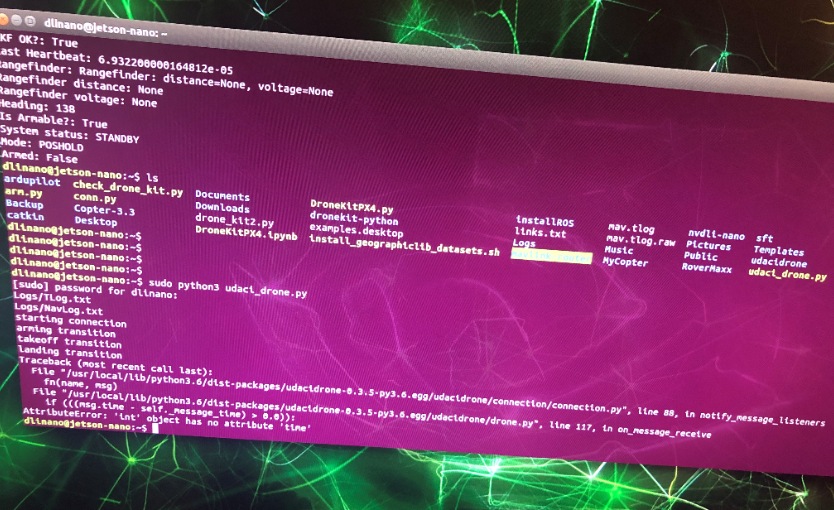
conn = MavlinkConnection('/dev/ttyTHS1,57600',PX4=True, threaded=False)

drone = UpAndDownFlyer(conn)

time.sleep(2)

drone.start()

First time after an hour I stopped the program and second time I let it run till batteries depleted to non-functional status.



**Software Installation:**

udacidrone, pymavlink, pygame, dronekit-python, ardupilot, ros melodic for jetson nano,

**PYTHON 3 Installed packages**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| absl-py (0.7.1) | actionlib (1.12.0) | Adafruit-GPIO (1.0.4) | Adafruit-PureIO (0.2.3) | Adafruit-SSD1306 (1.6.2) | angles (1.9.11) |
| apt-clone (0.2.1) | apturl (0.5.2) | asn1crypto (0.24.0) | astor (0.8.0) | attrs (19.3.0) | backcall (0.1.0) |
| beautifulsoup4 (4.6.0) | bleach (3.1.0) | blinker (1.4) | bondpy (1.8.3) | Brlapi (0.6.6) | camera-calibration-parsers (1.11.13) |
| catkin (0.7.19) | certifi (2019.11.28) | cflib (0.1.8) | chardet (3.0.4) | conda (4.3.16) | controller-manager (0.15.1) |
| controller-manager-msgs (0.15.1) | cryptography (2.1.4) | cupshelpers (1.0) | cv-bridge (1.13.0) | cycler (0.10.0) | decorator (4.4.0) |
| defer (1.0.6) | defusedxml (0.6.0) | diagnostic-analysis (1.9.3) | diagnostic-common-diagnostics (1.9.3) | diagnostic-updater (1.9.3) | distro-info (0.18ubuntu0.18.04.1) |
| dronekit (2.9.2) | dronekit-sitl (3.3.0) | dynamic-reconfigure (1.6.0) | entrypoints (0.3) | feedparser (5.2.1) | future (0.18.2) |
| gast (0.2.2) | gazebo-plugins (2.8.4) | gazebo-ros (2.8.4) | gencpp (0.6.2) | geneus (2.2.6) | genlisp (0.4.16) |
| genmsg (0.5.12) | gennodejs (2.0.1) | genpy (0.6.9) | graphsurgeon (0.3.2) | grpcio (1.21.1) | h5py (2.9.0) |
| html5lib (0.999999999) | httplib2 (0.9.2) | idna (2.8) | image-geometry (1.13.0) | importlib-metadata (1.5.0) | intelhex (2.2.1) |
| interactive-markers (1.11.4) | ipykernel (5.1.1) | ipython (7.5.0) | ipython-genutils (0.2.0) | ipywidgets (7.4.2) | jedi (0.13.3) |
| jetcam (0.0.0) | jetcard (0.0.0) | Jinja2 (2.10.1) | jsonschema (3.0.1) | jupyter (1.0.0) | jupyter-clickable-image-widget (0.1.0.dev0) |
| jupyter-client (5.2.4) | jupyter-console (6.0.0) | jupyter-core (4.4.0) | jupyterlab (0.35.6) | jupyterlab-server (0.2.0) | kdl-parser-py (1.13.1) |
| keyring (10.6.0) | keyrings.alt (3.0) | language-selector (0.1) | laser-geometry (1.6.4) | launchpadlib (1.10.6) | lazr.restfulclient (0.13.5) |
| lazr.uri (1.0.3) | louis (3.5.0) | lxml (4.5.0) | macaroonbakery (1.1.3) | Mako (1.0.7) | MarkupSafe (1.1.1) |
| matplotlib (2.1.1) | MAVProxy (1.8.18) | mavros (0.33.3) | message-filters (1.14.3) | mistune (0.8.4) | mock (3.0.5) |
| monotonic (1.5) | more-itertools (8.2.0) | nbconvert (5.5.0) | nbformat (4.4.0) | notebook (5.7.8) | numpy (1.16.4) |
| oauth (1.0.1) | oauthlib (2.0.6) | olefile (0.45.1) | packaging (20.1) | PAM (0.4.2) | pandocfilters (1.4.2) |
| parso (0.4.0) | pexpect (4.7.0) | pickleshare (0.7.5) | Pillow (6.0.0) | pip (9.0.1) | pluggy (0.13.1) |
| portpicker (1.3.1) | prometheus-client (0.7.0) | prompt-toolkit (2.0.9) | protobuf (3.0.0) | psutil (5.6.7) | ptyprocess (0.6.0) |
| py (1.8.1) | py-cpuinfo (5.0.0) | pycairo (1.16.2) | pycosat (0.6.3) | pycrypto (2.6.1) | pycups (1.9.73) |
| pygame (1.9.6) | Pygments (2.4.2) | pygobject (3.26.1) | PyICU (1.9.8) | PyJWT (1.5.3) | pymacaroons (0.13.0) |
| pymavlink (2.4.6) | PyNaCl (1.1.2) | pyparsing (2.4.6) | pyRFC3339 (1.0) | pyrsistent (0.15.2) | pyserial (3.4) |
| pytest (5.3.5) | python-apt (1.6.5+ubuntu0.2) | python-dateutil (2.8.0) | python-debian (0.1.32) | python-qt-binding (0.3.6) | pytz (2018.3) |
| pyusb (1.0.2) | pyxdg (0.25) | PyYAML (3.12) | pyzmq (18.0.1) | qt-dotgraph (0.3.16) | qt-gui (0.3.16) |
| qt-gui-cpp (0.3.16) | qt-gui-py-common (0.3.16) | qtconsole (4.5.1) | requests (2.22.0) | requests-unixsocket (0.1.5) | resource-retriever (1.12.5) |
| rosbag (1.14.3) | rosboost-cfg (1.14.7) | rosclean (1.14.7) | roscreate (1.14.7) | rosgraph (1.14.3) | roslaunch (1.14.3) |
| roslib (1.14.7) | roslint (0.11.2) | roslz4 (1.14.3) | rosmake (1.14.7) | rosmaster (1.14.3) | rosmsg (1.14.3) |
| rosnode (1.14.3) | rosparam (1.14.3) | rospy (1.14.3) | rosservice (1.14.3) | rostest (1.14.3) | rostopic (1.14.3) |
| rosunit (1.14.7) | roswtf (1.14.3) | rqt-action (0.4.9) | rqt-bag (0.4.12) | rqt-bag-plugins (0.4.12) | rqt-console (0.4.8) |
| rqt-dep (0.4.9) | rqt-graph (0.4.10) | rqt-gui (0.5.0) | rqt-gui-py (0.5.0) | rqt-image-view (0.4.13) | rqt-launch (0.4.8) |
| rqt-logger-level (0.4.8) | rqt-moveit (0.5.7) | rqt-msg (0.4.8) | rqt-nav-view (0.5.7) | rqt-plot (0.4.9) | rqt-pose-view (0.5.8) |
| rqt-publisher (0.4.8) | rqt-py-common (0.5.0) | rqt-py-console (0.4.8) | rqt-reconfigure (0.5.1) | rqt-robot-dashboard (0.5.7) | rqt-robot-monitor (0.5.9) |
| rqt-robot-steering (0.5.10) | rqt-runtime-monitor (0.5.7) | rqt-rviz (0.6.0) | rqt-service-caller (0.4.8) | rqt-shell (0.4.9) | rqt-srv (0.4.8) |
| rqt-tf-tree (0.6.0) | rqt-top (0.4.8) | rqt-topic (0.4.10) | rqt-web (0.4.8) | ruamel.yaml (0.16.6) | ruamel.yaml.clib (0.2.0) |
| rviz (1.13.6) | SecretStorage (2.3.1) | Send2Trash (1.5.0) | sensor-msgs (1.12.7) | setuptools (41.0.1) | simplejson (3.13.2) |
| six (1.14.0) | smach (2.0.1) | smach-ros (2.0.1) | smclib (1.8.3) | spidev (3.4) | ssh-import-id (5.7) |
| system-service (0.3) | systemd-python (234) | tensorrt (5.0.6.3) | termcolor (1.1.0) | terminado (0.8.2) | testpath (0.4.2) |
| tf (1.12.0) | tf-conversions (1.12.0) | tf2-geometry-msgs (0.6.5) | tf2-kdl (0.6.5) | tf2-py (0.6.5) | tf2-ros (0.6.5) |
| topic-tools (1.14.3) | torch (1.1.0a0+b457266) | torchvision (0.2.2.post3) | tornado (6.0.2) | traitlets (5.0.0.dev0) | ubuntu-drivers-common (0.0.0) |
| udacidrone (0.3.5) | uff (0.5.5) | ufw (0.36) | unattended-upgrades (0.1) | unity-scope-calculator (0.1) | unity-scope-chromiumbookmarks (0.1) |
| unity-scope-colourlovers (0.1) | unity-scope-devhelp (0.1) | unity-scope-firefoxbookmarks (0.1) | unity-scope-manpages (0.1) | unity-scope-openclipart (0.1) | unity-scope-texdoc (0.1) |
| unity-scope-tomboy (0.1) | unity-scope-virtualbox (0.1) | unity-scope-yelp (0.1) | unity-scope-zotero (0.1) | urdfdom-py (0.4.1) | urllib3 (1.25.8) |
| utm (0.4.0) | uvloop (0.9.1) | wadllib (1.3.2) | wcwidth (0.1.8) | webencodings (0.5.1) | websockets (4.0.1) |
| wheel (0.34.2) | widgetsnbextension (3.4.2) | xacro (1.13.4) | xkit (0.0.0) | zipp (2.1.0) | zope.interface (4.3.2) |

**PYTHON 2 Packages Installed**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| actionlib (1.12.0) | angles (1.9.11) | asn1crypto (0.24.0) | attrs (17.4.0) | autobahn (17.10.1) | Automat (0.6.0) |
| backports.functools-lru-cache (1.4) | beautifulsoup4 (4.6.0) | bondpy (1.8.3) | bzr (2.8.0.dev1) | camera-calibration-parsers (1.11.13) | catkin (0.7.19) |
| catkin-pkg (0.4.15) | catkin-pkg-modules (0.4.15) | cbor (1.0.0) | chardet (3.0.4) | click (6.7) | colorama (0.3.7) |
| configobj (5.0.6) | constantly (15.1.0) | controller-manager (0.15.1) | controller-manager-msgs (0.15.1) | cryptography (2.1.4) | cv-bridge (1.13.0) |
| cycler (0.10.0) | Cython (0.26.1) | decorator (4.1.2) | defusedxml (0.5.0) | diagnostic-analysis (1.9.3) | diagnostic-common-diagnostics (1.9.3) |
| diagnostic-updater (1.9.3) | docutils (0.14) | dronekit (2.9.2) | dronekit-sitl (3.3.0) | dynamic-reconfigure (1.6.0) | empy (3.3.2) |
| enum34 (1.1.6) | funcsigs (1.0.2) | future (0.18.2) | futures (3.2.0) | gazebo-plugins (2.8.4) | gazebo-ros (2.8.4) |
| gcovr (3.4) | gencpp (0.6.2) | geneus (2.2.6) | genlisp (0.4.16) | genmsg (0.5.12) | gennodejs (2.0.1) |
| genpy (0.6.9) | gps (3.17) | graphsurgeon (0.3.2) | gyp (0.1) | html5lib (0.999999999) | httplib2 (0.9.2) |
| hyperlink (17.3.1) | idna (2.6) | image-geometry (1.13.0) | incremental (16.10.1) | intelhex (2.2.1) | interactive-markers (1.11.4) |
| ipaddress (1.0.17) | kdl-parser-py (1.13.1) | keyring (10.6.0) | keyrings.alt (3.0) | laser-geometry (1.6.4) | launchpadlib (1.10.6) |
| lazr.restfulclient (0.13.5) | lazr.uri (1.0.3) | lxml (4.5.0) | lz4 (0.10.1) | matplotlib (2.1.1) | MAVProxy (1.8.18) |
| mavros (0.33.3) | mercurial (4.5.3) | message-filters (1.14.3) | monotonic (1.5) | mpi4py (2.0.0) | netifaces (0.10.4) |
| nose (1.3.7) | numpy (1.13.3) | oauth (1.0.1) | olefile (0.45.1) | PAM (0.4.2) | paramiko (2.0.0) |
| pexpect (4.8.0) | Pillow (5.1.0) | pip (9.0.1) | pluggy (0.6.0) | psutil (5.6.7) | ptyprocess (0.6.0) |
| py (1.5.2) | py-ubjson (0.8.5) | pyasn1 (0.4.2) | pyasn1-modules (0.2.1) | pycairo (1.16.2) | pycrypto (2.6.1) |
| pydot (1.2.3) | pygame (1.9.6) | Pygments (2.2.0) | pygobject (3.26.1) | pymavlink (2.4.6) | PyNaCl (1.1.2) |
| PyOpenGL (3.1.0) | pyOpenSSL (17.5.0) | pyparsing (2.2.0) | pyserial (3.4) | pytest (3.3.2) | python-dateutil (2.6.1) |
| python-gnupg (0.4.1) | python-qt-binding (0.3.6) | python-snappy (0.5) | PyTrie (0.2) | pytz (2018.3) | pyxdg (0.25) |
| PyYAML (3.12) | qrcode (5.3) | qt-dotgraph (0.3.16) | qt-gui (0.3.16) | qt-gui-cpp (0.3.16) | qt-gui-py-common (0.3.16) |
| resource-retriever (1.12.5) | roman (2.0.0) | rosbag (1.14.3) | rosboost-cfg (1.14.7) | rosclean (1.14.7) | roscreate (1.14.7) |
| rosdep (0.18.0) | rosdep-modules (0.18.0) | rosdistro (0.8.0) | rosdistro-modules (0.8.0) | rosgraph (1.14.3) | rosinstall (0.7.8) |
| rosinstall-generator (0.1.18) | roslaunch (1.14.3) | roslib (1.14.7) | roslint (0.11.2) | roslz4 (1.14.3) | rosmake (1.14.7) |
| rosmaster (1.14.3) | rosmsg (1.14.3) | rosnode (1.14.3) | rosparam (1.14.3) | rospkg (1.2.2) | rospkg-modules (1.2.2) |
| rospy (1.14.3) | rosservice (1.14.3) | rostest (1.14.3) | rostopic (1.14.3) | rosunit (1.14.7) | roswtf (1.14.3) |
| rqt-action (0.4.9) | rqt-bag (0.4.12) | rqt-bag-plugins (0.4.12) | rqt-console (0.4.8) | rqt-dep (0.4.9) | rqt-graph (0.4.10) |
| rqt-gui (0.5.0) | rqt-gui-py (0.5.0) | rqt-image-view (0.4.13) | rqt-launch (0.4.8) | rqt-logger-level (0.4.8) | rqt-moveit (0.5.7) |
| rqt-msg (0.4.8) | rqt-nav-view (0.5.7) | rqt-plot (0.4.9) | rqt-pose-view (0.5.8) | rqt-publisher (0.4.8) | rqt-py-common (0.5.0) |
| rqt-py-console (0.4.8) | rqt-reconfigure (0.5.1) | rqt-robot-dashboard (0.5.7) | rqt-robot-monitor (0.5.9) | rqt-robot-steering (0.5.10) | rqt-runtime-monitor (0.5.7) |
| rqt-rviz (0.6.0) | rqt-service-caller (0.4.8) | rqt-shell (0.4.9) | rqt-srv (0.4.8) | rqt-tf-tree (0.6.0) | rqt-top (0.4.8) |
| rqt-topic (0.4.10) | rqt-web (0.4.8) | rviz (1.13.6) | scipy (0.19.1) | SecretStorage (2.3.1) | sensor-msgs (1.12.7) |
| service-identity (16.0.0) | setuptools (39.0.1) | simplejson (3.13.2) | six (1.14.0) | smach (2.0.1) | smach-ros (2.0.1) |
| smclib (1.8.3) | subprocess32 (3.2.7) | tensorrt (5.0.6.3) | tf (1.12.0) | tf-conversions (1.12.0) | tf2-geometry-msgs (0.6.5) |
| tf2-kdl (0.6.5) | tf2-py (0.6.5) | tf2-ros (0.6.5) | topic-tools (1.14.3) | trollius (2.0.1) | Twisted (17.9.0) |
| txaio (2.8.1) | u-msgpack-python (2.1) | udacidrone (0.3.5) | uff (0.5.5) | unity-lens-photos (1.0) | urdfdom-py (0.4.1) |
| vcstools (0.1.42) | wadllib (1.3.2) | webencodings (0.5) | wheel (0.34.2) | wsaccel (0.6.2) | wstool (0.1.17) |
| wxPython (3.0.2.0) | wxPython-common (3.0.2.0) | xacro (1.13.4) | zope.interface (4.3.2) |  |  |