**Solution for connecting Jetson-Nano to Pixhawk**: (*thanks to all who pitched in and a special thanks to ppoirier*)

**Result** : Propeller rotated as programmed in [simple\_goto.py](https://github.com/dronekit/dronekit-python/blob/master/examples/simple_goto/simple_goto.py)

**Main issue**: Firmware on Pixhawk required upgrade, this was suggested by “**ppoirier**” on ardupilot discussion forum (<https://discuss.ardupilot.org/t/nvidia-jetson-nano-for-pixhawk/46052/30>)

**Suggestion:** Upgraded from **Autopilot Firmware version: APM:Copter-3.4.6** **to Autopilot Firmware version: APM:Copter-4.0.1.**

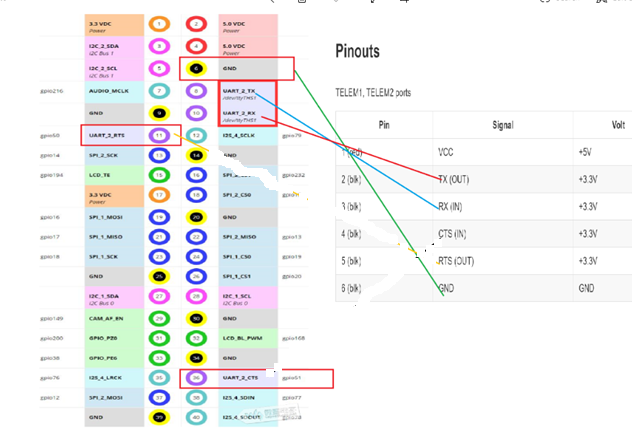
[**mtbsteve**](https://discuss.ardupilot.org/u/mtbsteve)mentioned  <https://github.com/mtbsteve/redtail> is up and running along with a full implementation of the original Redtail project on Arducopter plus a bunch of enhancements.  
<https://discuss.ardupilot.org/t/nvidia-jetson-nano-for-pixhawk/46052/32>

**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 (10th pin) is connected to TX of Pixhawk (2nd pin)



How to upgrade **Autopilot Firmware :**

1. **First use MissionPlanner to update firmware (**<https://ardupilot.org/planner/docs/common-loading-firmware-onto-pixhawk.html>**)**
2. **Next recalibrate Accel Calibration and Compass, for this I used QGroundControl**

**Following are the steps to configure QGroundControl**

https://docs.qgroundcontrol.com/en/getting\_started/download\_and\_install.html

*sudo usermod -a -G dialout $USER*

*sudo apt-get remove modemmanager -y*

*sudo apt install gstreamer1.0-plugins-bad gstreamer1.0-libav -y*

Download https://s3-us-west-2.amazonaws.com/qgroundcontrol/latest/QGroundControl.AppImage

*chmod +x ./QGroundControl.AppImage*

*./QGroundControl.AppImage (or double click)*

perform calibration mainly the

1. Accel Calibiration

2. Compass calibration (preferablly sitting outside )

**Two tests performed and both achieved satisfactory results:**

**Test1**: [vehicle\_state.py](https://github.com/dronekit/dronekit-python/blob/master/examples/vehicle_state/vehicle_state.py) (<https://github.com/dronekit/dronekit-python/tree/master/examples/vehicle_state>)

Customization on [vehicle\_state.py](https://github.com/dronekit/dronekit-python/blob/master/examples/vehicle_state/vehicle_state.py) , I have hardcoded the parameters for connection\_string

*#vehicle = connect(connection\_string, wait\_ready=True)*

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

**output of Test1:**

*Connecting to vehicle on: <dronekit.Vehicle object at 0x7f835e9240>*

*CRITICAL:autopilot:PreArm: GPS glitching*

*Get all vehicle attribute values:*

*Autopilot Firmware version: APM:Copter-4.0.1*

*Major version number: 4*

*Minor version number: 0*

*Patch version number: 1*

*Release type: rc*

*Release version: 0*

*Stable release?: True*

*Autopilot capabilities*

*Supports MISSION\_FLOAT message type: True*

*Supports PARAM\_FLOAT message type: True*

*Supports MISSION\_INT message type: True*

*Supports COMMAND\_INT message type: True*

*Supports PARAM\_UNION message type: False*

*Supports ftp for file transfers: True*

*Supports commanding attitude offboard: True*

*Supports commanding position and velocity targets in local NED frame: True*

*Supports set position + velocity targets in global scaled integers: True*

*Supports terrain protocol / data handling: True*

*Supports direct actuator control: False*

*Supports the flight termination command: True*

*Supports mission\_float message type: True*

*Supports onboard compass calibration: True*

*Global Location: LocationGlobal:lat=39.0454788,lon=-77.5109107,alt=89.06*

*Global Location (relative altitude): LocationGlobalRelative:lat=39.0454788,lon=-77.5109107,alt=-0.927*

*Local Location: LocationLocal:north=-2.4225430488586426,east=0.04539155960083008,down=0.9788298010826111*

*Attitude: Attitude:pitch=-0.02892705425620079,yaw=-2.761756181716919,roll=0.00626177154481411*

*Velocity: [-0.44, 0.23, 0.01]*

*GPS: GPSInfo:fix=3,num\_sat=11*

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

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

*EKF OK?: True*

*Last Heartbeat: 0.7351576400005797*

*Rangefinder: Rangefinder: distance=None, voltage=None*

*Rangefinder distance: None*

*Rangefinder voltage: None*

*Heading: 201*

*Is Armable?: True*

*System status: STANDBY*

*Groundspeed: 0.48128166794776917*

*Airspeed: 0.4520000219345093*

*Mode: GUIDED*

*Armed: False*

*Home location: LocationGlobal:lat=39.04550552368164,lon=-77.51091003417969,alt=89.98999786376953*

*Set new home location*

*New Home Location (from attribute - altitude should be 222): LocationGlobal:lat=39.0454776,lon=-77.5109096,alt=222.0*

*New Home Location (from vehicle - altitude should be 222): LocationGlobal:lat=39.04547882080078,lon=-77.51091003417969,alt=222.0*

*Set Vehicle.mode = GUIDED (currently: GUIDED)*

*Add `attitude` attribute callback/observer on `vehicle`*

*Wait 2s so callback invoked before observer removed*

*CALLBACK: Attitude changed to Attitude:pitch=-0.028342360630631447,yaw=-2.761568546295166,roll=0.006377635523676872*

*CALLBACK: Attitude changed to Attitude:pitch=-0.028491877019405365,yaw=-2.761573076248169,roll=0.006517069414258003*

*CALLBACK: Attitude changed to Attitude:pitch=-0.02878626063466072,yaw=-2.761592149734497,roll=0.006305241957306862*

*CALLBACK: Attitude changed to Attitude:pitch=-0.029111385345458984,yaw=-2.7616162300109863,roll=0.005692942067980766*

*CALLBACK: Attitude changed to Attitude:pitch=-0.029359156265854836,yaw=-2.761615753173828,roll=0.005402371287345886*

*CALLBACK: Attitude changed to Attitude:pitch=-0.029208604246377945,yaw=-2.7615883350372314,roll=0.0054700858891010284*

*CALLBACK: Attitude changed to Attitude:pitch=-0.02889634668827057,yaw=-2.7616171836853027,roll=0.006072578951716423*

*CALLBACK: Attitude changed to Attitude:pitch=-0.028583664447069168,yaw=-2.7616231441497803,roll=0.0069753676652908325*

*Remove Vehicle.attitude observer*

*Add `mode` attribute callback/observer using decorator*

*Set mode=STABILIZE (currently: GUIDED) and wait for callback*

*Wait 2s so callback invoked before moving to next example*

*CALLBACK: Mode changed to VehicleMode:STABILIZE*

*Attempt to remove observer added with `on\_attribute` decorator (should fail)*

*Exception: Cannot remove observer added using decorator*

*Add attribute callback detecting ANY attribute change*

*Wait 1s so callback invoked before observer removed*

*CALLBACK: (heading): 201*

*CALLBACK: (airspeed): 0.11700000613927841*

*CALLBACK: (groundspeed): 0.25296738743782043*

*CALLBACK: (last\_heartbeat): 0.8704058809998969*

*CALLBACK: (last\_heartbeat): 0.8761682439999277*

*CALLBACK: (last\_heartbeat): 0.8848013730003004*

*CALLBACK: (last\_heartbeat): 0.8966339159997005*

*CALLBACK: (last\_heartbeat): 0.8991714470002989*

*CALLBACK: (gps\_0): GPSInfo:fix=3,num\_sat=11*

*CALLBACK: (last\_heartbeat): 0.9155551410003682*

*CALLBACK: (last\_heartbeat): 0.9365757670002495*

*CALLBACK: (last\_heartbeat): 0.9408005590003086*

*CALLBACK: (last\_heartbeat): 0.9495019690002664*

*CALLBACK: (last\_heartbeat): 0.9564203890004137*

*CALLBACK: (last\_heartbeat): 0.959036828000535*

*CALLBACK: (location.local\_frame): LocationLocal:north=-0.8009014129638672,east=0.7146644592285156,down=132.6950225830078*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (last\_heartbeat): 0.9698005549998925*

*CALLBACK: (last\_heartbeat): 0.9982773680003447*

*CALLBACK: (last\_heartbeat): 9.432400020159548e-05*

*CALLBACK: (last\_heartbeat): 0.05038922099993215*

*CALLBACK: (attitude): Attitude:pitch=-0.024946313351392746,yaw=-2.7615976333618164,roll=0.010313747450709343*

*CALLBACK: (location.global\_relative\_frame): LocationGlobalRelative:lat=39.0454716,lon=-77.5109039,alt=-132.675*

*CALLBACK: (location.global\_frame): LocationGlobal:lat=39.0454716,lon=-77.5109039,alt=89.32*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (last\_heartbeat): 0.9698005549998925*

*CALLBACK: (last\_heartbeat): 0.9982773680003447*

*CALLBACK: (last\_heartbeat): 9.432400020159548e-05*

*CALLBACK: (last\_heartbeat): 0.05038922099993215*

*CALLBACK: (attitude): Attitude:pitch=-0.024946313351392746,yaw=-2.7615976333618164,roll=0.010313747450709343*

*CALLBACK: (location.global\_relative\_frame): LocationGlobalRelative:lat=39.0454716,lon=-77.5109039,alt=-132.675*

*CALLBACK: (location.global\_frame): LocationGlobal:lat=39.0454716,lon=-77.5109039,alt=89.32*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (velocity): [-0.16, 0.15, -0.01]*

*CALLBACK: (last\_heartbeat): 0.08280577000004996*

*CALLBACK: (battery): Battery:voltage=0.0,current=None,level=None*

*CALLBACK: (last\_heartbeat): 0.09489966999990429*

*CALLBACK: (last\_heartbeat): 0.10634799000035855*

*CALLBACK: (heading): 201*

*CALLBACK: (airspeed): 0.0990000069141388*

*CALLBACK: (groundspeed): 0.22426146268844604*

*CALLBACK: (last\_heartbeat): 0.11478033399998822*

*CALLBACK: (last\_heartbeat): 0.12629631100026018*

*CALLBACK: (last\_heartbeat): 0.1342360449998523*

*CALLBACK: (last\_heartbeat): 0.1411041520004801*

*CALLBACK: (last\_heartbeat): 0.14937795200057735*

*CALLBACK: (gps\_0): GPSInfo:fix=3,num\_sat=10*

*CALLBACK: (last\_heartbeat): 0.1591080210000655*

*CALLBACK: (last\_heartbeat): 0.1746934760003569*

*CALLBACK: (last\_heartbeat): 0.19531545100016956*

*CALLBACK: (location.local\_frame): LocationLocal:north=-0.7577090263366699,east=0.6934318542480469,down=132.67344665527344*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (last\_heartbeat): 0.21268582300035632*

*CALLBACK: (last\_heartbeat): 0.26297993900061556*

*CALLBACK: (last\_heartbeat): 0.3134828590000325*

*CALLBACK: (last\_heartbeat): 0.3182143760004692*

*CALLBACK: (attitude): Attitude:pitch=-0.024778300896286964,yaw=-2.7616121768951416,roll=0.010530060157179832*

*CALLBACK: (last\_heartbeat): 0.3248859690002064*

*CALLBACK: (location.global\_relative\_frame): LocationGlobalRelative:lat=39.045472,lon=-77.5109041,alt=-132.656*

*CALLBACK: (location.global\_frame): LocationGlobal:lat=39.045472,lon=-77.5109041,alt=89.34*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (velocity): [-0.13, 0.14, -0.01]*

*CALLBACK: (last\_heartbeat): 0.33667913699991914*

*CALLBACK: (battery): Battery:voltage=0.0,current=None,level=None*

*CALLBACK: (last\_heartbeat): 0.3423529570000028*

*CALLBACK: (last\_heartbeat): 0.34972252799980197*

*CALLBACK: (last\_heartbeat): 0.3600759900000412*

*CALLBACK: (heading): 201*

*CALLBACK: (airspeed): 0.29600000381469727*

*CALLBACK: (groundspeed): 0.1988341063261032*

*CALLBACK: (last\_heartbeat): 0.37838470800033974*

*CALLBACK: (last\_heartbeat): 0.39673233200028335*

*CALLBACK: (gps\_0): GPSInfo:fix=3,num\_sat=10*

*CALLBACK: (last\_heartbeat): 0.4174636309999187*

*CALLBACK: (last\_heartbeat): 0.43582427599994844*

*CALLBACK: (last\_heartbeat): 0.44652894099999685*

*CALLBACK: (last\_heartbeat): 0.45290464499976224*

*CALLBACK: (location.local\_frame): LocationLocal:north=-0.709986686706543,east=0.6740903854370117,down=132.65574645996094*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (last\_heartbeat): 0.4597364490000473*

*CALLBACK: (last\_heartbeat): 0.465943922000406*

*CALLBACK: (last\_heartbeat): 0.516472727999826*

*CALLBACK: (last\_heartbeat): 0.5668334580004739*

*CALLBACK: (attitude): Attitude:pitch=-0.024639783427119255,yaw=-2.7615737915039062,roll=0.010666463524103165*

*CALLBACK: (last\_heartbeat): 0.5793902289997277*

*CALLBACK: (location.global\_relative\_frame): LocationGlobalRelative:lat=39.0454725,lon=-77.5109042,alt=-132.64*

*CALLBACK: (location.global\_frame): LocationGlobal:lat=39.0454725,lon=-77.5109042,alt=89.35*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (velocity): [-0.11, 0.13, -0.02]*

*CALLBACK: (battery): Battery:voltage=0.0,current=None,level=None*

*CALLBACK: (last\_heartbeat): 0.5929619079997792*

*CALLBACK: (last\_heartbeat): 0.5959536119999029*

*CALLBACK: (last\_heartbeat): 0.6048378890000095*

*CALLBACK: (heading): 201*

*CALLBACK: (airspeed): 0.27000001072883606*

*CALLBACK: (groundspeed): 0.17714251577854156*

*CALLBACK: (last\_heartbeat): 0.6261010710004484*

*CALLBACK: (last\_heartbeat): 0.6284703709998212*

*CALLBACK: (last\_heartbeat): 0.634057784000106*

*CALLBACK: (last\_heartbeat): 0.6499026689998573*

*CALLBACK: (last\_heartbeat): 0.6586420479998196*

*CALLBACK: (gps\_0): GPSInfo:fix=3,num\_sat=11*

*CALLBACK: (last\_heartbeat): 0.6801525239998227*

*CALLBACK: (last\_heartbeat): 0.6924212710000575*

*CALLBACK: (last\_heartbeat): 0.7065871240001798*

*CALLBACK: (location.local\_frame): LocationLocal:north=-0.6582937240600586,east=0.6640357971191406,down=132.6395263671875*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (last\_heartbeat): 0.7168523039999855*

*CALLBACK: (last\_heartbeat): 0.7675500180002928*

*CALLBACK: (last\_heartbeat): 0.8181183550004789*

*CALLBACK: (attitude): Attitude:pitch=-0.02446504682302475,yaw=-2.7615339756011963,roll=0.010788487270474434*

*CALLBACK: (location.global\_relative\_frame): LocationGlobalRelative:lat=39.045473,lon=-77.5109047,alt=-132.628*

*CALLBACK: (location.global\_frame): LocationGlobal:lat=39.045473,lon=-77.5109047,alt=89.37*

*CALLBACK: (location): <dronekit.Locations object at 0x7f822a2b38>*

*CALLBACK: (velocity): [-0.08, 0.11, -0.02]*

*CALLBACK: (battery): Battery:voltage=0.0,current=None,level=None*

*CALLBACK: (last\_heartbeat): 0.8406518120000328*

*CALLBACK: (last\_heartbeat): 0.8492957739999838*

*CALLBACK: (last\_heartbeat): 0.8597079870005473*

**Test2** [simple\_goto.py](https://github.com/dronekit/dronekit-python/blob/master/examples/simple_goto/simple_goto.py) (<https://github.com/dronekit/dronekit-python/tree/master/examples/simple_goto>)

, I have hardcoded the parameters for connection\_string

*#vehicle = connect(connection\_string, wait\_ready=True)*

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

*In this test props rotated and the everything worked as planned*

**Output of Test2:**

*starting copter simulator (SITL)*

*SITL already Downloaded and Extracted.*

*Ready to boot.*

*warning: TCG doesn't support requested feature: CPUID.01H:ECX.vmx [bit 5]*

*/lib64/ld-linux-x86-64.so.2: No such file or directory*

*Connecting to vehicle on: tcp:127.0.0.1:5760*

*Basic pre-arm checks*

*Arming motors*

*Waiting for arming...*

*Taking off!*

*Altitude: -130.824*

*Altitude: -130.853*

*Altitude: -130.867*

*Altitude: -130.873*

*Altitude: -130.835*

*Altitude: -130.811*

*Altitude: -130.789*

*Altitude: -130.72*

*Altitude: -130.68*