

Documentação - UAV-Toolkit

Autores:

Jesimar da Silva Arantes

Veronica Vannini

O presente documento tem como objetivo documentar todo o trabalho técnico que a Equipe UAV Team vem desenvolvendo, a fim de facilitar a reprodutibilidade dos nossos trabalhos posteriormente.

Tópicos:

- 1 - Montando o Drone F02192-X
- 2 - Configuração e Utilização da Intel Edison
- 3 - Configurando o Ambiente para HITL
- 4 - Executando Experimentos SITL (Sem Dronekit)
- 5 - Executando Aplicações do DroneKit
- 6 - Orçamento para Comprar Equipamentos
- 7 - Instalação de Softwares
- 8 - Dicas do FlighGear
- 9 - Utilizando o AutoFG
- 10 - Orientações para o Voo
- 11 - Outras Orientações
- 12 - Criando Rede Hotspot

1 - Montando o Drone F02192-X

A seguir encontra-se a documentação relativa a montagem do Drone F02192-X.

1 - Conjunto de Peças

A seguir estão descritas as peças usadas na montagem do drone F02192-X, as mesmas estão mostradas na Figura 1.1.

- 1x HJ 450 450F Nylon Fiber Flamewheel Frame Airframe kit HJ450
- 2x D2212 920KV Clockwise Counter-clockwise CW Brushless Motor
- 2x D2212 920KV Clockwise Counter-clockwise CCW Brushless Motor
- 4x 30A Brushless ESC
- 2x CW CCW propeller 9443
- 1x APM2.8 APM 2.8 Multicopter Flight Controller 2.5 2.6 Upgraded Built-in Compass
- 1x 6M GPS with Compass L5883 25cm Cable
- 1x 140MM High Landing Gear Skid Wheels Tripod Black
- 5x 10cm Servo Extension Lead Wire Cable MALE TO MALE
- 1x DJI GPS Folding Antenna Mount Holder Metal
- 1x Carbon Fiber Camera Gimbal Mount FPV Damping PTZ
- 12x 3.5mm Bullet Connector (banana plug)
- 1x Hook & Loop Fastening Tape
- 1x T Plug Male Connector Silicone Wire With 11.5CM 14awg
- 1x IMAX RC B3 Pro Compact Balance Charger
- 1x 11.1V 2200MAH 30C 3S1P battery
- 1x Flysky FS-i6 6CH 2.4G AFHDS 2A LCD Transmitter iA6 Receiver Mode 2



Figura 1.1: Kit Drone F02192-X completo.

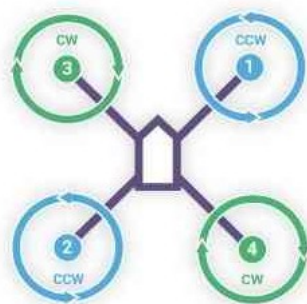
2 - Passo para Montagem do Drone

2.1 - Montagem do frame do quadricóptero F02192-X. Nota os meus ESCs já vieram soldados ao Frame do Drone. Nota: para parafusar o drone no chassi deve-se usar cola trava rosca.



Figura 1.2: Frame do quadricóptero montado.

2.2 - Acoplar os motores no frame e os parafusar conforme Figura 1.3 (no meu drone o motor branco gira no sentido anti-horário e o motor preto gira no sentido horário). Fazer as ligações dos ESCs nos motores. Após essa etapa o drone deve ficar de forma semelhante a Figura 1.4. Nota: meu drone nessa etapa já estava com os ESCs soldados. Pode-se ligar os conectores dos ESCs no Motor de qualquer forma, não existe o perigo de queimar o equipamento (O que poderá ocorrer é que o motor pode se ligar de forma invertida).



QUAD X

Figura 1.3: Regra para determinação dos números e sentidos de rotação dos motores.



Figura 1.4: Drone com motores conectados no Frame.

2.3 - Acoplar mecanicamente o piloto automático (APM) no topo do chassi. A parte frontal do drone é a parte do frame vermelha. A parte frontal da APM possui uma seta. Colocar uma

esponja entre a APM e o topo do chassi de forma de diminuir a vibração. Usar a fita 3M dupla face para colar essa estrutura.

2.4 - Conectar os cabos dos ESCs ao Piloto Automático (PA). Na Figura 1.5 temos os quatros ESCs. Os conectores azuis são conectados nos motores. As pontas brancas e pretas estão soldadas no frame do drone. O conectores com três pinos vermelho (vcc), marrom (gnd) e laranja (sinal) devem ser conectados no piloto automático. Será APENAS um dos ESCs que irá fornecer energia para o piloto automático. Dessa forma, os conectores de vcc e gnd dos outros três ESCs devem ser cortados, ou seja, apenas o pino de sinal deve ser ligado ao piloto automático. Em meu caso o motor 1 está sendo usado para alimentar o PA. Deve-se conectar os pinos no piloto automático na parte de OUTPUTS de forma semelhante a Figura 1.6. Motor 1 no pino 1, motor 2 no pino 2, motor 3 no pino 3, motor 4 no pino 4. Olhar bem os símbolos de gnd (-), vcc (+) e sinal (s).



Figura 1.5: Conjunto de ESC.

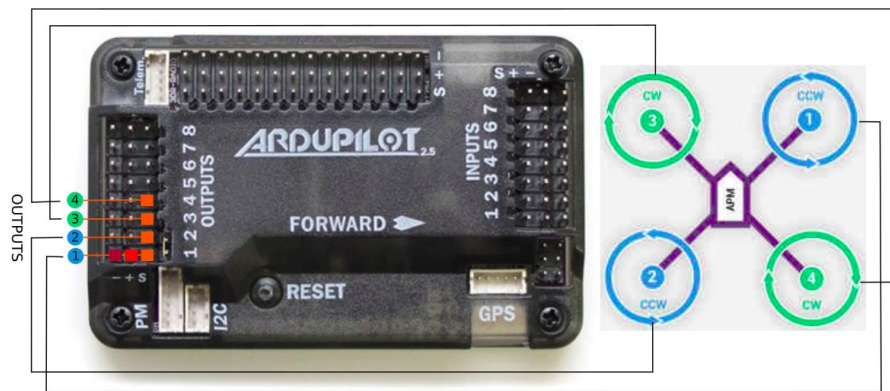


Figura 1.6: Conexão dos motores no piloto automático.

2.5 - Acoplar o GPS no drone, dica colocar ele na parte da frente do drone. A conexão do GPS no PA é bem simples deve-se apenas conectar o GPS na entrada do GPS e o conector do compasso na entrada abaixo do GPS.

2.6 - Acoplar o receptor de rádio na parte superior do drone. Conectar no PA na parte de INPUT um fio no pino 1 que vai até o CH1 do receptor de rádio. Um fio no pino 2 que vai até o CH2, repetir esse processo até o pino 6 se ligar no CH6. Ligar um pino gnd (-), vcc (+) e sinal (s) no INPUT pino 8 que vai até o receptor de rádio no pino B/VCC. As ligações devem ser semelhantes a Figura 1.7 abaixo.

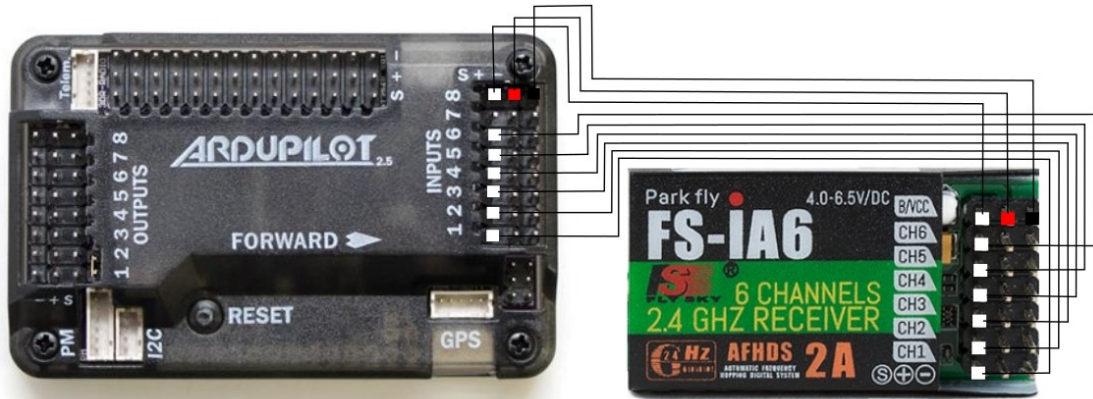
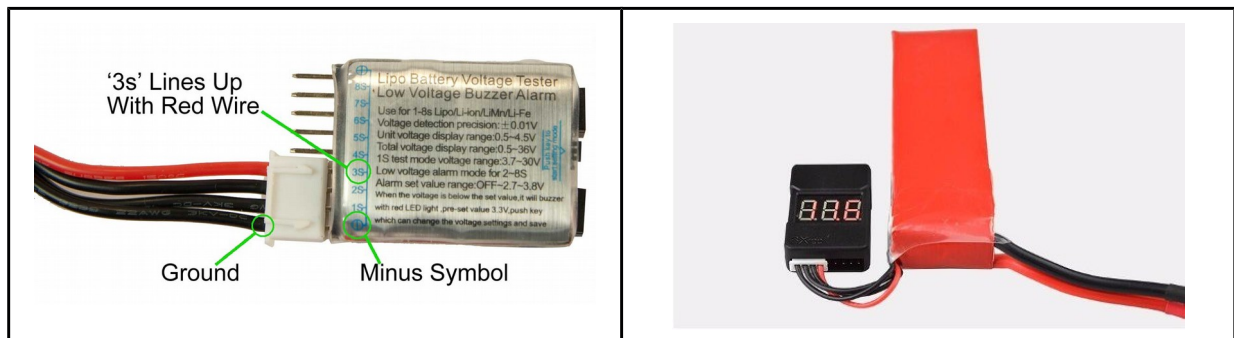


Figura 1.7: Conexão do receptor de rádio no PA.

OBS - Colocar esponja dentro do piloto automático para não alterar pressão barométrica.

2.7 - Ao fazer o voo deve-se ligar a bateria no módulo de alarme de baixa voltagem de bateria, também conhecido como Testador de Voltagem da Bateria. A forma correta de se fazer essa ligação é conforme as figuras abaixo. Caso conecte errado não tem risco de queima, apenas não irá funcionar. Ligar os quatros conectores da bateria no testador de bateria e ligar o negativo no fio preto.



2 - Configuração e Utilização da Intel Edison

A seguir encontra-se a documentação para configuração e utilização da Plataforma Intel Edison. A Figura 2.1 mostra uma foto da Intel Edison.

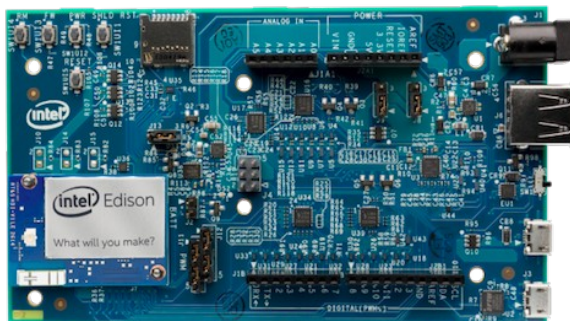


Figura 2.1 - Foto da Intel Edison.

O Kit da Intel Edison Acompanha:

- 1x Módulo Intel Edison
- 1x Arduino Breakout Kit

Características Técnicas da Intel Edison:

- Processador Intel® Atom™ SoC de 500 MHz de 22nm com CPU dual-core
- Wi-Fi dual-band (2.4 e 5GHz) integrado
- Bluetooth LE 4.0 integrado
- Memória: 1GB LPDDR3
- Armazenamento: 4GB eMMC
- Sistema Operacional: Linux - Yocto
- Suporte a Python, Java, C/C++, Arduino e CPLEX

Especificações Arduino Breakout:

- Dimensões: 127 x 72 x 12 mm
- Compatibilidade com Arduino Uno (com 4 PWM, ao invés de 6 do Arduino)
- 20 entradas/saídas digitais e 4 saídas PWM
- 6 Entradas analógicas
- 2 UART
- Interface I2C
- Interface SPI
- Conector ICSP de 6 pinos
- Conexão micro USB
- Slot cartão microSD
- Configuração de nível de sinal (3.3V ou 5V) por jumper
- Adaptador para fonte: 7 V~15 VDC / 500 mA

Usuários Criados na Intel Edison:

- Usuário: uav
- Usuário: root

IP da Intel Edison (IP_EDISON):

- Rede LCR: 192.168.205.220
- Rede eduroam: 172.28.107.227
- RedeDrone: 192.168.0.102

- RedeDroneC: 192.168.43.133
- RedeDroneCjes: 192.168.43.9
- RedeDronePC: 10.42.0.199
- Alice 192.168.0.100

Softwares Instalados Na Intel Edison:

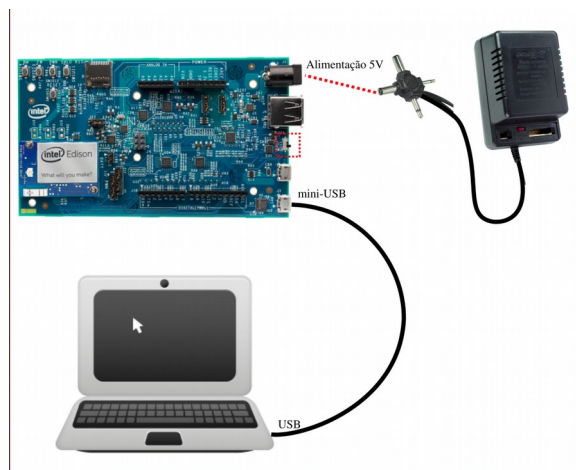
- mavproxy
- dronekit
- dronekit-sitl
- screen
- cplex
- python (acho que é nativo)
- java (acho que é nativo)

Notas:

- Sempre desligue a Edison usando o comando: `shutdown -h now`

Logando na Edison sem Wifi com cabo USB

A montagem de hardware deve estar disposta da seguinte forma:



- 1 - Instale o Putty no Windows ou no Linux no computador
- 2 - Abra o Putty digitando no terminal o seguinte comando: `$ putty`
- 3 - Entre com as seguintes informações após selecionar Session em seguida Serial.
 Serial line: `/dev/ttyUSB0`
 Speed: `115200`
 Em seguida clique em Open.

OBS: não deixar a conexão com o ESC conectada durante este teste se a edison estiver na tomada.

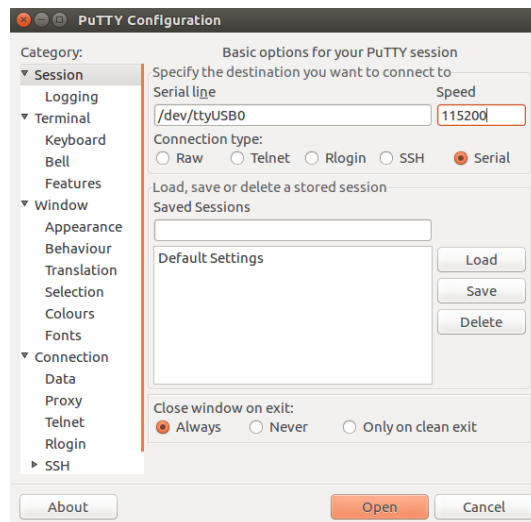


Figura 2.2 - Tela do sistema Putty.

4 - Uma vez feito isso a seguinte tela deverá aparecer.

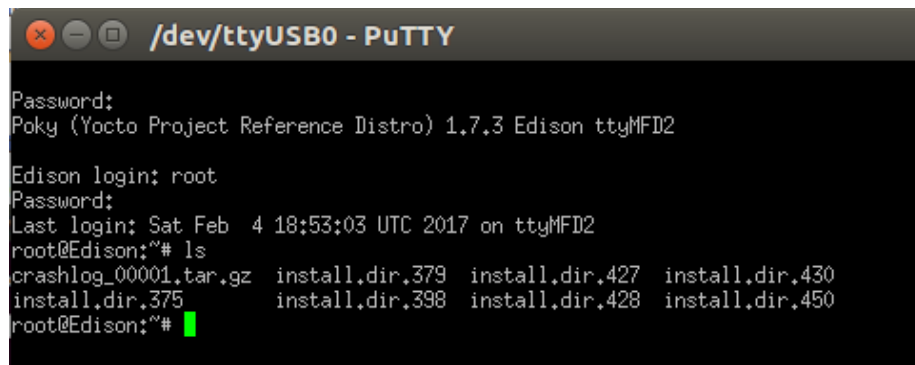


Figura 2.3 - Tela do terminal do putty após logado na Intel Edison.

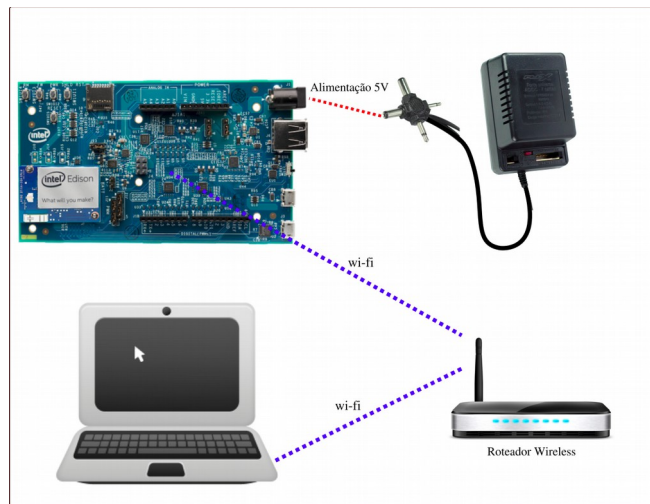
- 5 - O login é: root
- 6 - Digite a senha
- 7 - Pronto agora você já tem acesso total a plataforma Intel Edison.
- 8 - Para sair (deslogar) da Intel Edison basta digitar: # exit
- 9 - Uma forma alternativa para logar na Intel Edison a partir do Linux pode ser visto em: <https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-linux>
- 10- Uma forma alternativa para logar na Intel Edison a partir do Windows pode ser visto em: <https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows>

Configurando Wifi na Intel Edison

- 1 - Digite no terminal dentro da Intel Edison: \$ configure_edison --wifi
- 2 - Aguarde a varredura das redes Wifi.
- 3 - Escolha a rede wifi que deseja conectar a Intel Edison.
- 4 - Digite a senha que deseja se conectar com a Intel Edison.
- 5 - Para descobrir o IP da Intel Edison basta digitar: \$ ifconfig
- 6 - Então basta fazer SSH de outro computador na Intel Edison.

Logando com Wifi sem cabo USB

A montagem de hardware deve estar disposta da seguinte forma:



- 1 - Conectar a Edison e o Computador na rede wifi local.
- 2 - Usar o SSH para logar na Intel Edison usando o comando: `$ ssh root@IP_EDISON`
- 3 - Digite a senha
- 4 - Pronto você já tem acesso completo a plataforma Intel Edison.

Colocar arquivos de execução do ProOF na Intel Edison

- 1 - Primeiro passo é copiar a pasta do work_space para dentro do cartão de memória
 - Pasta: `/media/sdcard/`
 - Comando: `scp work_space.zip root@IP_EDISON:/media/sdcard/Testes_Jesimar/`
- 2 - Copie o arquivo cplex.jar para a pasta work_space/code/Java/lib/
 - Comando: `cp ./installs/cplex/lib/cplex.jar ./Testes/work_space/code/Java/lib/`.

Rodar o ProOF na Intel Edison

- 1 - Copie o arquivo subset_jobs.sh para a pasta do work_space
- 2 - Entre na pasta do work_space que deseja rodar.
- 3 - Execute o comando `./subset_jobs.sh 1 50 &`
Obs: os argumentos do comando acima indicam o primeiro e último job que desejo rodar.
- 4 - Pegar os resultados rodados na Intel Edison:
- 5 - Compacte a pasta usando o TAR:
- 6 - Comando: `tar -czvf result_work_space.tar.gz work_space/`

Pegar os resultados do ProOF na Intel Edison

- 1 - Instalar o programa Filezilla
- 2 - Abrir o programa Filezilla
- 3 - Em host coloque o IP da Intel Edison: `IP_EDISON`
- 4 - Coloque o usuário da Intel Edison: `root`
- 5 - Digite a senha
- 6 - Coloque a porta: `22`

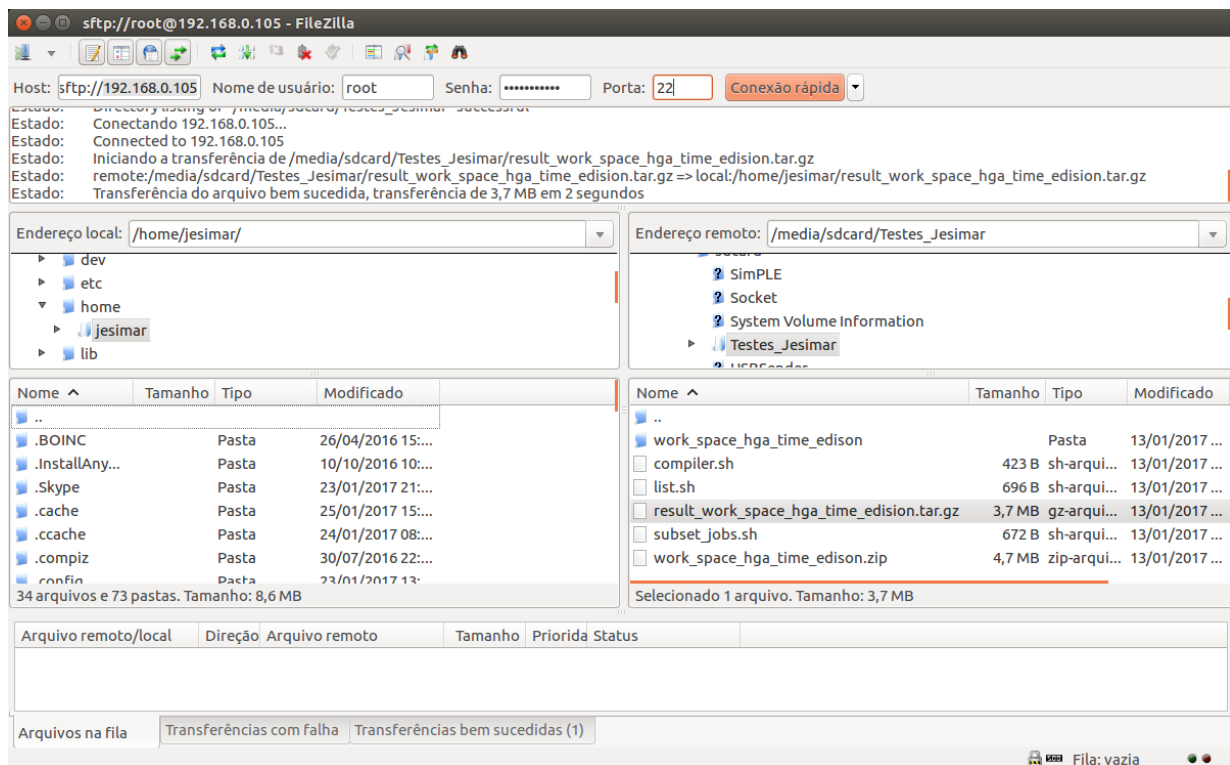
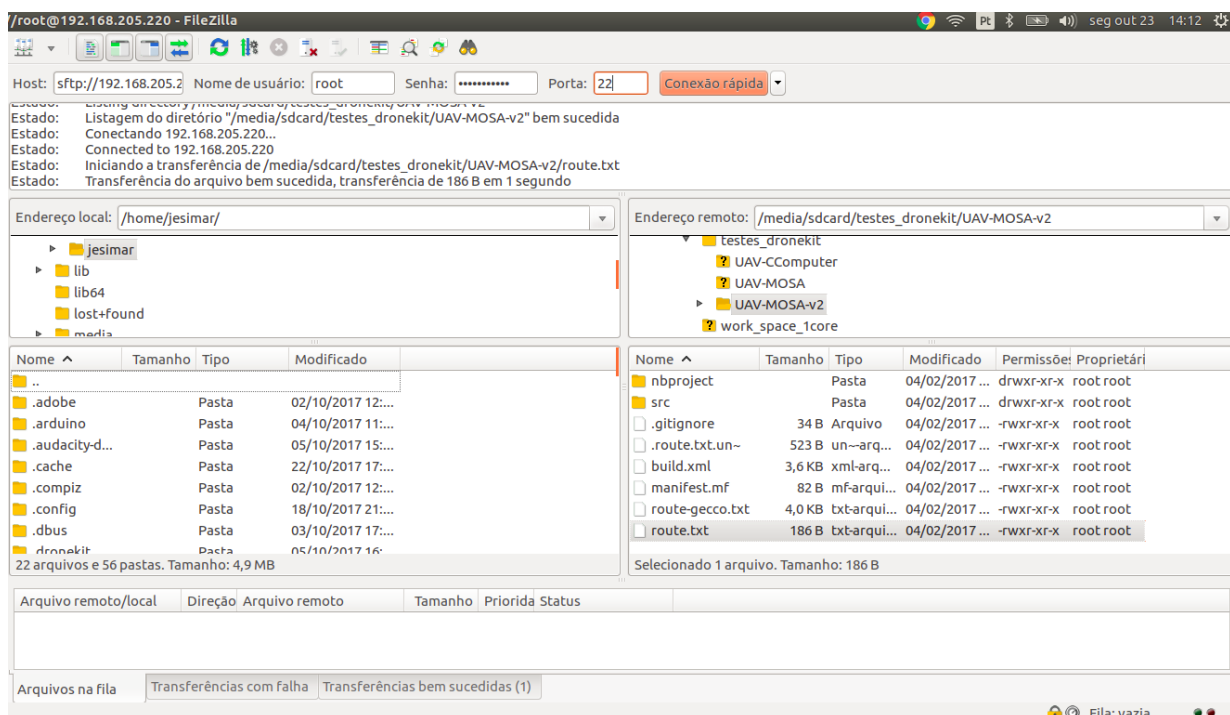


Figura 2.4 - Tela do Filezilla para enviar e pegar os resultados na Intel Edison.

Acessar a Intel Edison através do Filezilla

- 1 - Abrir o programa Filezilla
- 2 - Em host coloque o IP da Intel Edison: IP_EDISON
- 3 - Coloque o usuário da Intel Edison: root
- 4 - Digite a senha
- 5 - Coloque a porta: 22



Usando esse programa você poderá tanto enviar arquivo para a Edison quanto pegá-los.

Monitorar USB Intel Edison

Comando para listar dispositivos da Edison:

```
$ ls /dev/tty*
```

Comando para executar mavproxy na Edison :

```
$ mavproxy.py --master /dev/ttyUSB0,57600 --out 172.28.253.253:14550 --out 127.0.0.1:14551
```

OBS: O IP 172.28.253.253 é onde será executado a estação de base Mission Planner. Observação não funciona com a estação de base QgroundControl, pois estamos usando APM e o QGC só suporta PixHawk. Após fazer isso pode-se dar comandos através da estação de base Mission Planner. Mas aparentemente não é suportado comandos através do MAVProxy.

Interrompendo IPTABLES no Linux (Ubuntu)

No terminal digite:

```
$ iptables -L
```

```
$ systemctl stop firewalld
```

```
$ iptables -F
```

```
$ iptables -L
```

//Desabilita iptables para sempre

```
$ sudo systemctl disable firewalld
```

Rodando Blink Intel Edison

Para testar os GPIOs da Intel Edison pode-se executar o exemplo básico do Blink 13. Esse exemplo faz piscar o led 13 da placa do arduino. Abaixo encontra-se o código que roda essa aplicação. Antes de rodar esse exemplo é necessário instalar as bibliotecas mraa e time (**conferir comando**).

| |
|--|
| Arquivo: blink-io13.py Executar: python blink-io13.py |
| <pre>import mraa import time gpio = mraa.Gpio(13) gpio.dir(mraa.DIR_OUT) while True: gpio.write(1) time.sleep(0.2) gpio.write(0) time.sleep(0.2)</pre> |

Ligando a Intel Edison no Ardupilot (APM)

Para ligar a Intel Edison no Ardupilot faça os seguintes passos:

1 - Conecte o conversor USB Serial na porta USB da Intel Edison.

2 - Para poder usar essa porta USB na Edison antes certifique-se que o microswitich (destacado em vermelho na Figura 2.5) esteja no modo Host (em direção a porta USB normal).

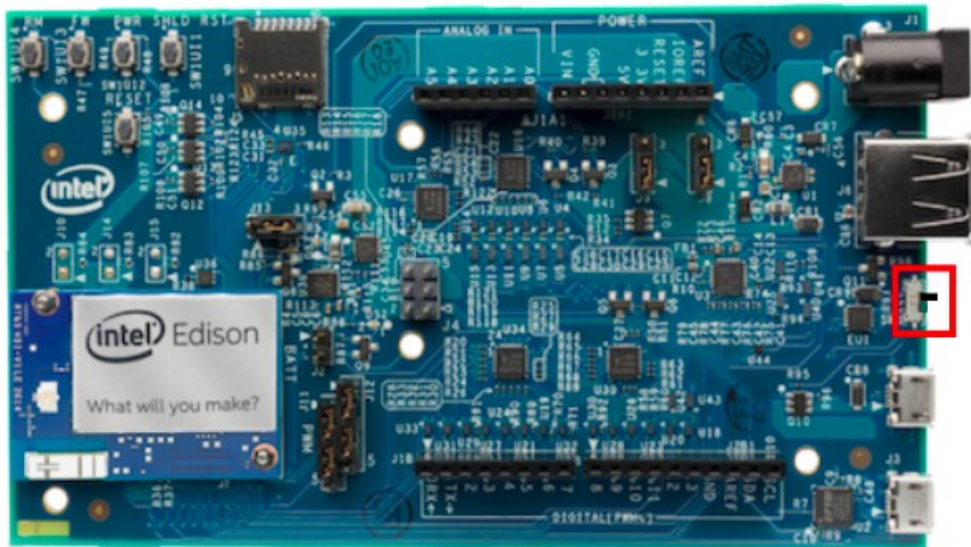
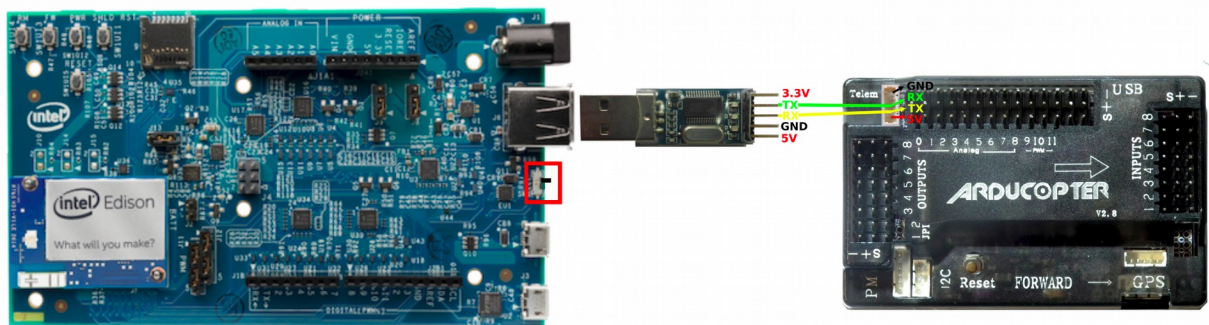


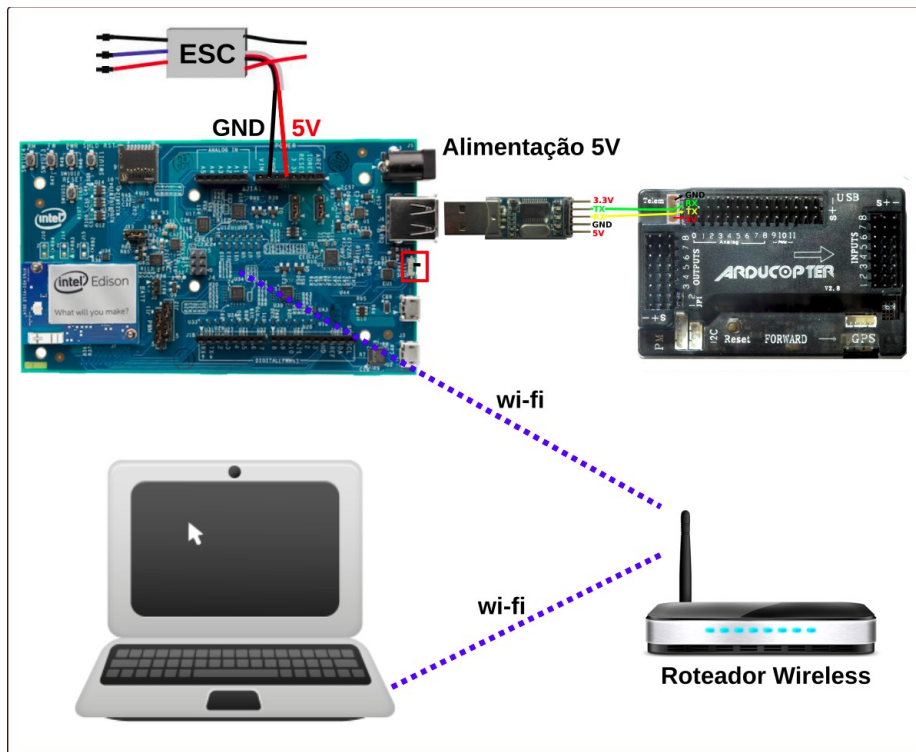
Figura 2.5 - Trocando entre as portas USB.

3 - A Ligação entre a Intel Edison e a APM via USB Serial é feita da seguinte maneira conforme visto na Figura 2.6. Esse ligação é feita através da telemetria da APM. Os pinos de RX e TX são ligados de forma invertida conforme mostra a figura. Os pinos de GND e 5V **NÃO DEVEM SER LIGADOS**. Cuidado o conversor USB Serial não deve ser alimentado ele alimenta outras placas.



2.6 - Ligação da Intel Edison com conversor USB Serial e Aducopter (APM 2.8).

4 - A montagem final deve ficar conforme a figura abaixo.



Alimentando a Intel Edison pelo ESC do Drone.

Os ESCs dos drones F02192-X são alimentados com 11.1V, e possuem como saída 3 pinos (GND, 5V e Sinal). Atualmente já estamos usando um dos ESCs (pino de 5V e GND) para alimentar o Arducopter. Usaremos um outro ESC (pino de 5V e GND) para alimentar a Intel Edison. Dessa forma, precisamos conectar o pino de 5V do ESC no pino de 5V da Edison e o GND do ESC no GND da Edison.

Comandos Gerais da Intel Edison:

Comando Linux para saber a data e hora do sistema.

```
root@Edison:/# date
```

Comando Linux para atualizar a data e hora do sistema. A hora não fica exatamente igual a nossa hora (pega-se o horário de New York)

```
root@Edison:/# rdate www.nist.gov
```

Comando linux para atualizar a hora de acordo com a hora de São Paulo.

```
root@Edison:/# timedatectl set-timezone America/Sao_Paulo
```

Comando linux para desligar a Intel Edison.

```
root@Edison:/# shutdown -h now
```


3 - Configurando o Ambiente para HITL

Descrição:

Este tutorial visa descrever os passos necessários para se instalar e configurar um ambiente de Hardware-In-The-Loop (HITL).

Ambientes Testados:

- Sistema Operacional: Linux - Ubuntu 16.04

Softwares Utilizados:

- QGroundControl 3.1.0
 - Sistemas Operacionais: Windows, Linux, Mac OS X
 - Site: <http://qgroundcontrol.com/>
 - Site para Instalação: <http://qgroundcontrol.com/downloads/>
- X-Plane 10
 - Sistemas Operacionais: Windows, Linux, Mac OS X
 - Site: <http://www.x-plane.com/>
 - Site para Instalação: <http://www.x-plane.com/desktop/try-it/>

Hardwares Utilizados:

- Piloto Automático: Pixhawk
- Rádio Telemetria - Módulo Air:
- Rádio Telemetria - Módulo Ground:
- Transmissor do Rádio Controle: Taranis X9D Plus
- Receptor do Rádio Controle:

Esquema da Arquitetura:

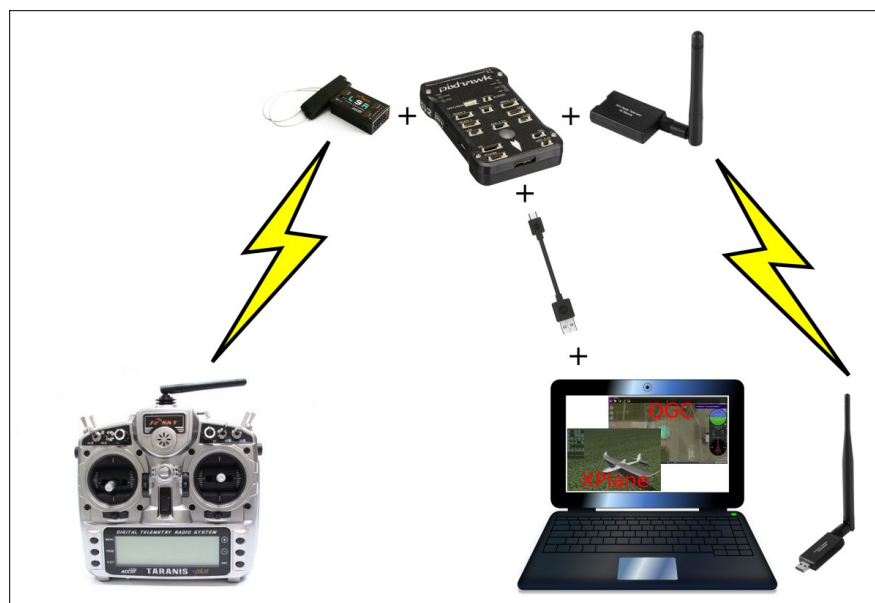


Figura 3.1: Esquema da arquitetura de HITL montada.

A Figura 3.1 mostra a Estação de Controle de Solo (GCS) rodando os softwares QGroundControl e XPlane. O transmissor de Rádio Controle (RC) se comunica com o

receptor de rádio de acoplado no piloto automático Pixhawk. Por sua vez, o piloto automático através do módulo de telemetria (air) envia suas informações recebidas para o módulo de telemetria (ground).

Vantagens:

- QGroundControl: Multiplataforma ao contrário do Mission Planner, dá suporte a HITL ao contrário do Mission Planner.
- XPlane: Realismo, qualidade do simulador.

Desvantagens:

- QGroundControl: Funciona somente com pilotos automáticos (PA) da Pixhawk da (3DR), ao contrário do Mission Planner que dá suporte a outros PA como o APM.
- XPlane: Pago, mais pesado.

Como Configurar:

- Configurar o X-Plane e o QGroundControl segundo o link:
- Site: <https://pixhawk.org/users/hil>
- Os seguintes passos devem ser feitos para configurar o ambiente HITL:
 - Abra o QGroundControl
 - Vá na segunda aba (Vehicle Setup)
 - Selecionar o menu Firmware
 - Conecte o sistema de hardware da Pixhawk no computador
 - Assim que reconhecer o PA irá aparecer uma opção. Clique em PX4 stack
 - Selecionar o menu Airframe -> Selecione Simulation -> HILStar (XPlane)
 - Selecionar Apply e Restart
 - Desconectar e reconectar o sistema de Hardware (Pixhawk).
 - Selecionar e Calibrar o Radio (rádio é obrigatório)
 - Em Flight Modes
 - Escolha um canal do rádio e configure os flight modes.
 - Deixe o FlightMode no modo Manual
 - Abra o XPlane
 - Vá em Widgets -> HIL Config
 - Os dois checkbox devem estar selecionados (check box estranho)
 - Selecione connect
 - O PA vai estar desarmado então coloque o throttle na posição inferior direita e segure até armar.
 - A partir disso é só voar no modo manual
 - Se tiver alguma missão basta colocar no modo Auto (Mission) [Não testado]
 - No XPlane aperte Shift+6 (Shift+4, Shift+5 também funcionam) para mudar a tela.

4 - Executando Experimentos SITL (sem DroneKit)

A seguir serão descritos um os passos a passos para executar experimentos Software-In-The-Loop (SITL) após ter todo o ambiente do Ardupilot configurado no seu computador.

O que é Necessário:

- [Software FlightGear](#)
- [Ardupilot](#)
- [MAVProxy](#)
- [MAVExplorer.py](#)

Ambientes Testados:

- Sistema Operacional: Linux - Ubuntu 16.04
- Sistema Operacional: Linux - Manjaro

Parâmetros Usados:

- LOCAL_INST: Significa o local da instalação, por exemplo: /home/jesimar/

Comandos para Iniciar o FlightGear pelo Ardupilot

Entre na pasta onde encontra-se o arquivo fg_quad_view.sh ou fg_plane_view.sh.

Em geral, esse arquivo fica na pasta: LOCAL_INST/ardupilot/Tools/autotest/

#abre no FlightGear com aeronave Copter

```
$ ./fg_quad_view.sh
```

#abre no FlightGear com aeronave Plane

```
$ ./fg_plane_view.sh
```

Iniciar o Sim_Vehicle

O Sim_Vehicle é um programa que irá fazer as simulações SITL.

1 - Entre na pasta ArduCopter ou ArduPlane: `$ cd ArduCopter`

```
$ sim_vehicle.py -j4 -L KSFO --console --map
```

```
$ sim_vehicle.py -j4 -L GECCO2017 --console --map
```

```
$ sim_vehicle.py -j4 -L SCBRASIL --console --map
```

OBS: O arquivo de localização chamado "locations.txt" pode ser usado para colocar as coordenadas, altitude e orientação da aeronave, onde deseja-se iniciar a missão. Geralmente fica localizado em: LOCAL_INST/ardupilot/Tools/autotest/locations.txt

OBS: Nunca colocar altitudes muito altas para o copter. Altitude entre 0 a 100 metros.

2 - Comando para carregar uma missão através de um arquivo de missão.

```
$ wp load LOCAL_INST/ardupilot/Tools/autotest/ArduPlane-Missions/mission.txt
```

```
$ wp load ../Tools/autotest/ArduPlane-Missions/CMAC-toff-loop.txt
```

OBS: Esse comando não consegue carregar missões que encontram-se dentro de diretórios que contenham o caracter espaço. Por exemplo diretório: "Área de Trabalho".

OBS: Essa regra do espaço também é válida para carregar missões via interface gráfica.

3 - A seguir tem-se um conjunto de comandos que podem ser enviados para o MAVProxy que definirão a simulação SITL.

3.1 - Muda o modo de voo para guided:

\$ mode guided

3.2 - Arma os motores preparando para a decolagem da aeronave:

\$ arm throttle

OBS: O comando "arm takeoff" deve começar em no máximo 15 segundos depois de armado senão os motores serão desarmados

3.3 - Decola a aeronave até 40 de altitude:

\$ takeoff 40

3.4 - Muda o modo de voo para auto:

\$ mode auto

3.5 - Pouso a aeronave:

\$ mode land

3.6 - Retorna para o início da rota:

\$ mode rtl

3.7 - Iniciar uma aplicação para visualizar o mapa da missão e rota da aeronave:

\$ module load map

3.7 - Iniciar uma aplicação para visualizar o console que contém dados da aeronave:

\$ module load console

3.7 - Iniciar uma aplicação para visualizar o gráfico da rota:

\$ module load graph

\$ graph gtakeoff

\$ graph gaccel

Iniciar o MAVExplorer

O MAVExplorer.py é um programa para plotar gráficos e fazer análises das simulações SITL.

1 - Comando para abrir o MAVExplorer.

\$ MAVExplorer.py

Após digitar tal comando uma interface gráfica será aberta.

2 - Usando a interface dessa tela abra o arquivo .BIN com os dados da rota gerado. Esse arquivo é automaticamente gravado em LOCAL_INST/ardupilot/ArduCopter/logs/ ou LOCAL_INST/ardupilot/ArduPlane/logs/

2.1 - Um outro local onde costuma-se encontrar esses logs é em:

\$HOME/Documentos/QGroundControl/Telemetry/

3 - Comandos para plotar os gráficos

\$ MAV> graph GPS.Alt

5 - Executando Aplicações do DroneKit

Abaixo encontra-se o passo a passo para executar aplicações sobre o DroneKit.

O que é Necessário:

- [DroneKit Python](#)
- DroneKit SITL
- MAVProxy
- QGroundControl
- APM Planner2

Ambientes Testados:

- Sistema Operacional: Linux - Ubuntu 16.04
- Sistema Operacional: Linux - Manjaro

O primeiro passo após configurar o ambiente do dronekit é rodar a aplicação básica:

\$ python hello.py

Nota: é necessário que seja o python 2.7.13. NÃO funciona com python3

```
print ("Start simulator (SITL)")
import dronekit_sitl
sitl = dronekit_sitl.start_default()
connection_string = sitl.connection_string()

# Import DroneKit-Python
from dronekit import connect, VehicleMode

# Connect to the Vehicle.
print("Connecting to vehicle on: %s" % (connection_string,))
vehicle = connect(connection_string, wait_ready=True)

# Get some vehicle attributes (state)
print ("Get some vehicle attribute values:")
print (" GPS: %s" % vehicle.gps_0)
print (" Battery: %s" % vehicle.battery)
print (" Last Heartbeat: %s" % vehicle.last_heartbeat)
print (" Is Armable?: %s" % vehicle.is_armable)
print (" System status: %s" % vehicle.system_status.state)
print (" Mode: %s" % vehicle.mode.name)  # settable

# Close vehicle object before exiting script
vehicle.close()

# Shut down simulator
```

```
sitl.stop()
print("Completed")
```

Um próximo passo é tentar rodar a aplicação basic_sitl.py:

Abra um terminal e execute:

```
$ dronekit-sitl copter
```

ou ainda

```
$ dronekit-sitl plane-3.3.0 --home=-35.363261,149.165230,584,353
```

Abra outro terminal e execute:

```
$ python basic_sitl.py
```

```
# Import DroneKit-Python
from dronekit import connect, VehicleMode

# Connect to the Vehicle.
connection_string = 'tcp:127.0.0.1:5760'
print("Connecting to vehicle on: %s" % (connection_string,))
vehicle = connect(connection_string, wait_ready=True)

# Get some vehicle attributes (state)
print "Get some vehicle attribute values:"
print " GPS: %s" % vehicle.gps_0
print " Battery: %s" % vehicle.battery
print " Last Heartbeat: %s" % vehicle.last_heartbeat
print " Is Armable?: %s" % vehicle.is_armable
print " System status: %s" % vehicle.system_status.state
print " Mode: %s" % vehicle.mode.name # settable

# Close vehicle object before exiting script
vehicle.close()

print("Completed")
```

Um próximo passo é tentar rodar a missão da usp no campus 2 faça o seguinte:

Abra um terminal e execute:

```
$ dronekit-sitl copter (fica muito longe do primeiro WP)
```

ou

```
$ dronekit-sitl copter --home=-22.005640,-47.932474,870,70
```

ou

```
$ dronekit-sitl plane-3.3.0 --home=-22.005640,-47.932474,870,70
```

Abra outro terminal e execute:

```
ImportError: No module named serial -> $ sudo pip install pyserial
```

```
ImportError: No module named MAVProxy.modules.lib -> MAVProxy mal instalado ou
instalado no Python3 (precisa rodar tudo no python2)
```

```
$ mavproxy.py --master tcp:127.0.0.1:5760 --sitl 127.0.0.1:5501 --out 127.0.0.1:14550
--out 127.0.0.1:14551
```

No terminal do mavproxy abra o map (opção 1):

```
MAV> module load map
```

Abra uma estação de controle de solo (opção 2):

Abra o QGroundControl (a conexão dele com o drone automática)

ou

Abra o APM Planner2 (conexão automática com udp 127.0.0.1:14550)

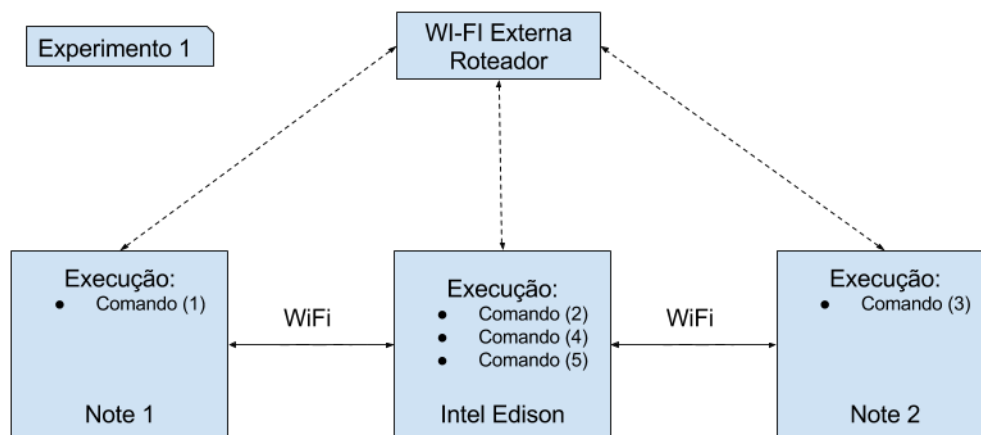
Abra outro terminal e execute:

```
$ python mission_usp_campus2.py --connect 127.0.0.1:14551 (precisa do arquivo SITL-USP-Campus2.txt)
```

ou

```
$ python simple_goto.py --connect 127.0.0.1:14551
```

Para rodar a missão da usp no campus 2 (na edison integrado com notebook):



Opção 1: usando comunicação wireless

Abra um terminal e execute (no notebook 1):

(1) `$ dronekit-sitl copter --home=lat,lng,alt,angulo`

- **Exemplo Camp2:** `$ dronekit-sitl copter --home=-22.005640,-47.932474,870,70`
- **Exemplo Camp1:** `$ dronekit-sitl copter --home=-22.006728,-47.898173,870,70`
- **Exemplo Camp1:** `$ dronekit-sitl copter --home=-22.005874,-47.898744,870,70`
- **Exemplo Camp1:** `$ dronekit-sitl copter-3.2.1 --home=-22.005925,-47.898643,870,70`

Abra um terminal e execute ([na intel edison](#)):

(2) `$ mavproxy.py --master tcp:IP_NOTE1:5760 --sitl IP_NOTE1:5501 --out udp:IP_NOTE2:14550 --out 127.0.0.1:14551`

- **Exemplo:** `$ mavproxy.py --master tcp:192.168.205.181:5760 --sitl 192.168.205.181:5501 --out udp:192.168.205.159:14550 --out 127.0.0.1:14551`
- `mavproxy.py --master tcp:192.168.205.231:5760 --sitl 192.168.205.231:5501 --out udp:192.168.205.231:14550 --out 127.0.0.1:14551`
- `mavproxy.py --master tcp:192.168.205.231:5760 --sitl 192.168.205.231:5501 --out udp:192.168.205.231:14550 --out 127.0.0.1:14551`
- `mavproxy.py --master tcp:10.42.0.93:5760 --sitl 10.42.0.93:5501 --out udp:10.42.0.93:14550 --out 127.0.0.1:14551`

- `mavproxy.py --master tcp:192.168.43.8:5760 --sitl 192.168.43.8:5501 --out udp:192.168.43.8:14550 --out 127.0.0.1:14551`

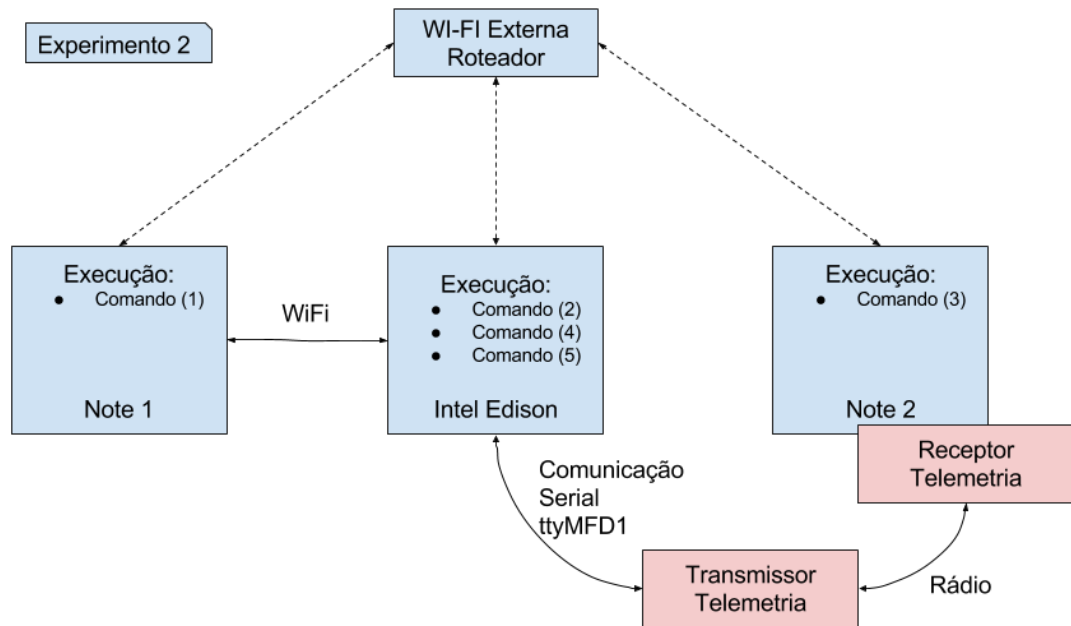
(3) Abrir o QGroundControl (*Estação de Base*)

(4) `$ python main.py /media/sdcard/testes_dronekit/UAV-CComputer`

(5) `$ java -jar dist/UAV-MOSA.jar /media/sdcard/testes_dronekit/UAV-MOSA`

OBS: Esse experimento pode ser realizado com apenas um notebook, neste caso os comando do note1 e note2 são realizados no mesmo computador em terminais diferentes.

OBS2: Não é necessário conectar a Intel Edison com o piloto automático neste experimento.



Opção 2: usando comunicação via Rádio

Abra um terminal e execute (no notebook 1):

(1) `$ dronekit-sitl copter --home=lat,lng,alt,angulo`

- **Exemplo:** `$ dronekit-sitl copter --home=-22.005640,-47.932474,870,70`

Abra um terminal e execute ([na intel edison](#)):

(2) `$ mavproxy.py --master tcp:IP_NOTE1:5760 --sitl IP_NOTE1:5501 --out=/dev/ttyMFD1,57600 --out 127.0.0.1:14551`

- **Exemplo:** `$ mavproxy.py --master tcp:192.168.205.181:5760 --sitl 192.168.205.181:5501 --out=/dev/ttyMFD1,57600 --out 127.0.0.1:14551`

(3) Abrir o QGroundControl (*Estação de Base*)

(4) `$ python main.py /media/sdcard/testes_dronekit/UAV-CComputer`

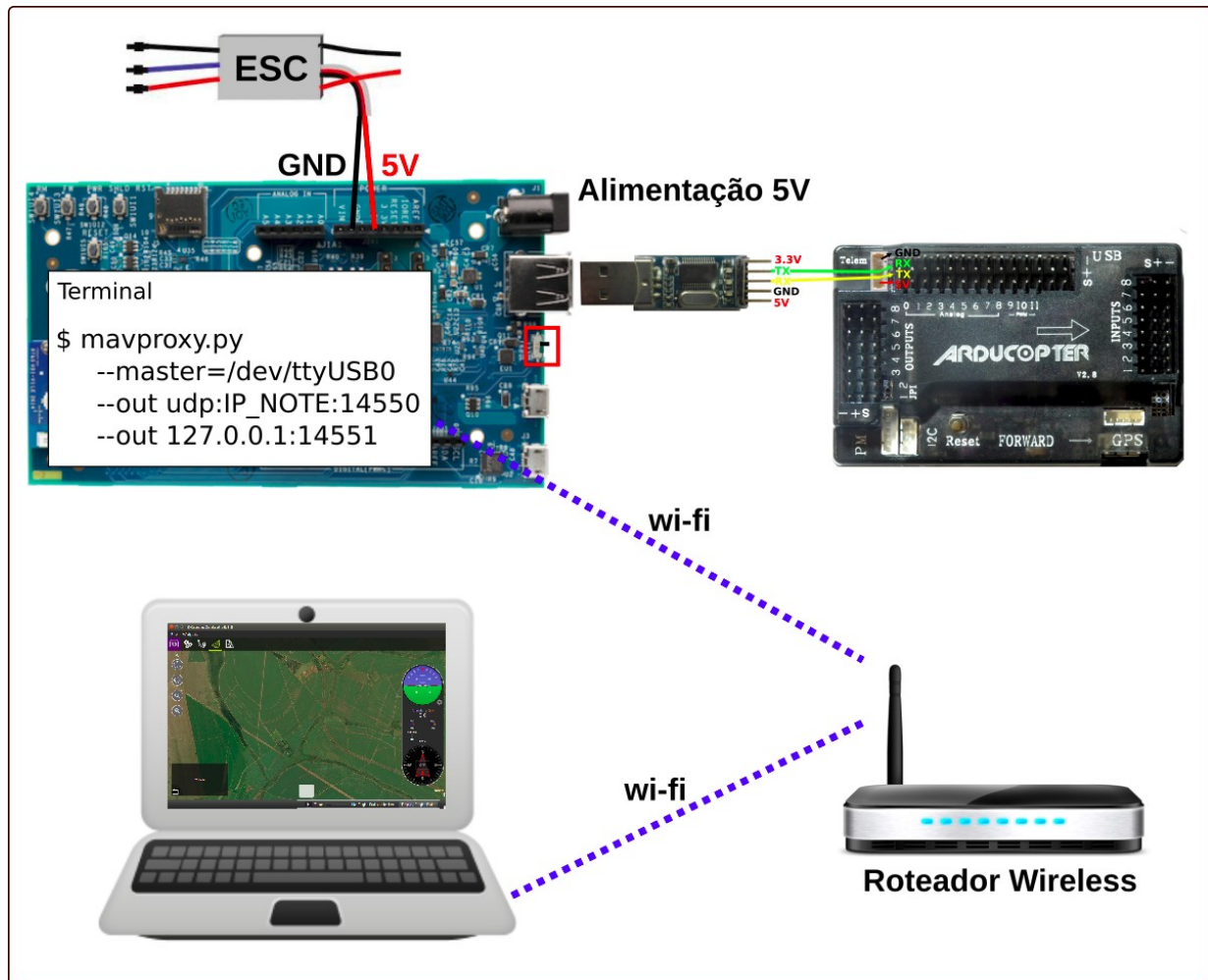
(5) `$ java -jar dist/UAV-MOSA.jar /media/sdcard/testes_dronekit/UAV-MOSA`

Em outro computador execute (no notebook 2):

OBS: Esse experimento pode ser realizado com apenas um notebook, neste caso os comando do note1 e note2 são realizados no mesmo computador em terminais diferentes.

OBS2: Não é necessário conectar a Intel Edison com o piloto automático neste experimento.

Para rodar exemplo básico onde a Intel Edison controla a APM faça o seguinte:



Notas:

A Intel Edison e o APM devem estar conectadas [via USB-Serial](#) (cabo USB-to-TTL)

Não é necessário telemetria

Para armar pode-se estar em qualquer modo de voo

Para a comunicação do notebook com a Intel Edison é necessário ambos estarem na mesma rede wifi.

Na Intel Edison faça:

```
1 - mavproxy.py --master=/dev/ttyUSB0 --out udp:IP_NOTE:14550 --out 127.0.0.1:14551
```

Exemplo: mavproxy.py --master=/dev/ttyUSB0 --out udp:192.168.0.100:14550 --out 127.0.0.1:14551

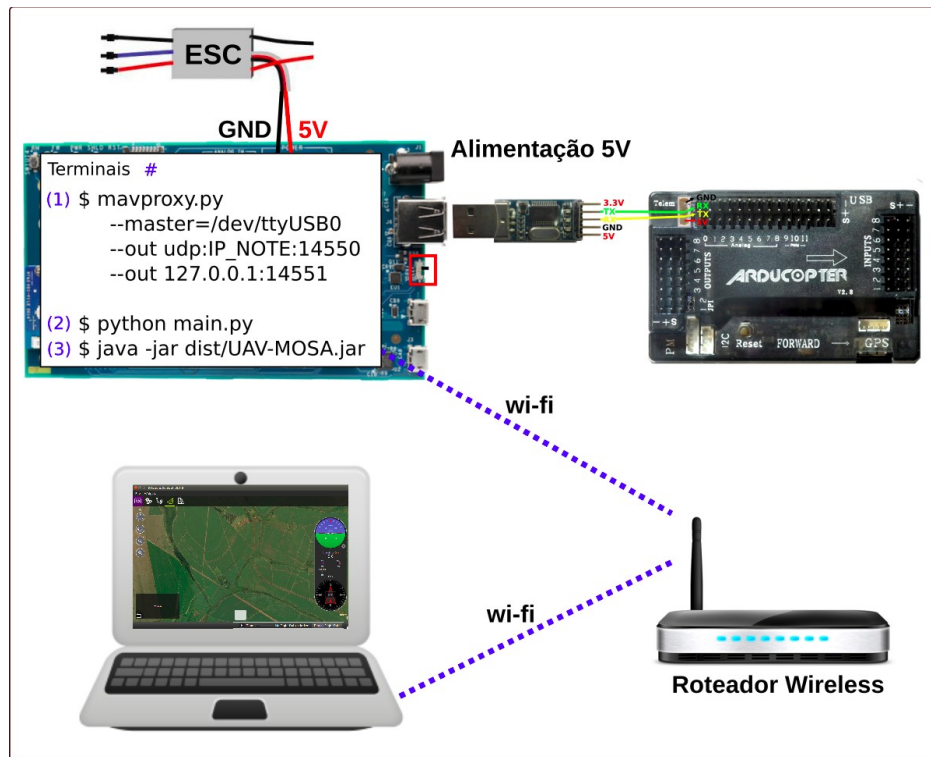
Em um notebook faça:

2 - Abrir no notebook ip (192.168.205.157) o QGroundControl

3 - No QGC mande arm os motores (mova slider para lado)

4 - No QGC mande desarmar os motores (mova slider para lado)

Para rodar exemplo completo com dronekit, MOSA-v0 na Intel Edison controlando a APM faça o seguinte:



Abra um terminal e execute ([na intel edison](#)):

(1) \$ mavproxy.py --master=/dev/ttyUSB0 --out udp:IP_NOTE:14550 --out 127.0.0.1:14551

- **Exemplo:** mavproxy.py --master=/dev/ttyUSB0 --out udp:172.20.17.193:14550 --out 127.0.0.1:14551
- mavproxy.py --master=/dev/ttyUSB0 --out udp:192.168.205.231:14550 --out 127.0.0.1:14551
- mavproxy.py --master=/dev/ttyUSB0 --out udp:192.168.43.124:14550 --out 127.0.0.1:14551

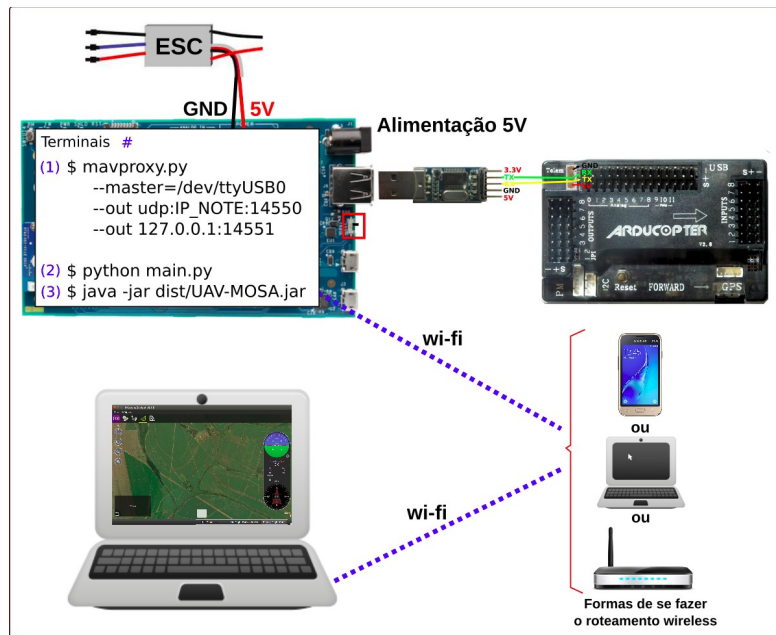
(2) Abrir o QGroundControl (*Estação de Base*)

(3) \$ python main.py /media/sdcard/testes_dronekit/UAV-CComputer

(4) \$ java -jar dist/UAV-MOSA.jar /media/sdcard/testes_dronekit/UAV-MOSA

Em outro computador execute (no notebook 2):

A Figura abaixo mostra diferentes formas avaliadas de se fazer o roteamento Wireless nos voos autônomos. Nesses experimentos foram avaliados Roteamento através de Celular, Notebook e usando um Roteador Wireless. O problema do roteador Wireless é o acesso a energia elétrica, pois dependendo de onde é realizado o experimento não se tem energia elétrica em campo. Até o momento o roteamento via celular e notebook foram equivalentes.



Ambiente para testes locais do MOSA:

Abra um terminal e execute:

```
$ dronekit-sitl copter --home=-22.005640,-47.932474,870,70
```

```
$ dronekit-sitl copter --home=-22.00592573,-47.89864304,870,70
```

Abra outro terminal e execute:

```
$ mavproxy.py --master tcp:127.0.0.1:5760 --sitl 127.0.0.1:5501 --out 127.0.0.1:14550
--out 127.0.0.1:14551
```

Abra uma estação de controle de solo:

Abra o QGroundControl

Abra outro terminal e execute:

```
$ python main.py
```

Abra outro terminal e execute:

```
$ java -jar dist/UAV-MOSA.jar
```

Para visualizar o resultado da simulação (após a simulação) pode-se utilizar o MAVExplorer.py para abrir o arquivo mav.tlog

Os logs de telemetria do QGC ficam armazenados, após efetuar uma simulação :

- \$HOME/home/jesimar/Documentos/QGroundControl/Telemetry/

Comando para Abrir o MAVExplorer:

- \$ MAVExplorer.py mav.tlog
 - MAV> map

Comando para Abrir o MAVExplorer:

- \$ MAVExplorer.py arquivo.bin
 - MAV> graph GPS.Alt

6 - Orçamento para Comprar Equipamentos

Softwares:

- Licença do X-Plane 11
 - Preço Unitário: U\$59,99 [R\$185,00]
 - Site: <http://www.x-plane.com/desktop/buy-it/>
 - Desvantagem: Versão beta e não funciona HITL
- Licença do X-Plane 10
 - Preço Unitário: U\$49,99 [R\$155,00]
 - Site: <http://www.x-plane.com/desktop/buy-it/>
 - Vantagem: Tem mais suporte, mais barata e suporta HITL

Hardwares:

- 2x Placa Mini Pixhawk (Opção 1)
 - Preço Unitário: U\$199,99 [R\$618,00]
 - Site: <https://store.3dr.com/products/3dr-pixhawk>
 - Incluso:
 - GPS
 - Placa de potência
 - Acessórios
- 2x Placa Pixhawk (Opção 2)
 - Preço Unitário: U\$70,00 [R\$216,00]
 - Site: AliExpress [New Arrival Pixhawk 2.4.8 PX4 Autopilot PIX 32 Bit]
 - Incluso:
 - GPS
 - Placa de potência
 - Acessórios
- 2x Conjunto de Rádio Telemetria
 - Já temos um, só precisa trocar o conector de 5 para 6 pinos
 - Preço Unitário: U\$16,00 [R\$50,00]
 - Site: AliExpress [TTL 3DR Radio Telemetry Kit Air Ground 915Mhz]
 - Incluso:
 - Módulo Telemetria Air
 - Módulo Telemetria Ground
 - Frequência 915MHz
 - Cabo serial 5P-6P para APM
 - Cabo serial 6P-6P para Pixhawk
- 2x Receptor Spektrum DX5e para Pixhawk (ou 1x receptor + 1x conversor serial)
- 1x Rádio Controle - Taranis X9D Plus com 1 receptor
 - Preço Unitário: U\$225,00 [R\$720,00]
 - Site: <https://www.horusrc.com/en/frsky-2-4g-accst-taranis-x9d-plus-transmitter-with-x8r-receiver-and-aluminum-box-carton-package.html>
 - Preço Unitário: U\$227,99 [R\$724,90] AliExpress [FrSky Taranis X9D Plus Transmitter]
- 1x Receptor para Rádio Controle do Taranis X9D
 - Preço Unitário: U\$33,90 [R\$105,00]
 - Site: AliExpress [FrSky 2.4G Telemetry Receiver X8R Taranis X9D]

- 1x Intel Edison só chip principal
 - Preço Unitário: R\$389,90
 - Site: <http://www.filipeflop.com/pd-2c0db8-modulo-intel-edison.html?ct=14f148&p=1&s=1>
- 1x Intel Edison Breakout Board
 - Preço Unitário: R\$459,90
 - Site: <http://www.filipeflop.com/pd-349a4f-intel-edison-breakout-board.html?ct=14f148&p=1&s=1>
- 1x Intel Edison para Arduino
 - Preço Unitário: R\$649,90
 - Site: <http://www.filipeflop.com/pd-207dbd-kit-intel-edison-para-arduino.html>
- 1x Odroid XU4
 - Preço Unitário: US\$59,00 [R\$182,00]
 - Site: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825
- 1x Raspberry Pi 3 Model B
 - Preço Unitário: US\$35,00 [R\$110,00]
 - Site: <https://www.element14.com/community/community/raspberry-pi>
- 1x BeagleBone Black
 - Preço Unitário: R\$399,90
 - Site: <http://www.filipeflop.com/pd-1436b0-beaglebone-black-rev-c.html?ct=93349&p=1&s=1>

Preços em Reais:

- Não incluídos possíveis valores de frete, etc.
 Valor Opção 1 = $155,00 + 2 \times 216,00 + 2 \times 50,00 + 105,00 + 720,00$
 Valor Opção 1 = $155,00 + 432,00 + 100 + 105,00 + 720,00$
 Valor Opção 1 = R\$1512,00

 Valor Opção 2 = $155,00 + 2 \times 618,00 + 2 \times 50,00 + 105,00 + 720,00$
 Valor Opção 2 = $155,00 + 1236,00 + 100 + 105,00 + 720,00$
 Valor Opção 2 = R\$2316,00

7 - Instalação de Softwares

Estações de Controle de Solo:

QGroundControl:

- Programa: QGroundControl
- Versão: 3.1.3
- Tamanho Arquivo: 52 MB
- Espaço Ocupado em Disco: 139 MB
- Dependências: `sudo apt-get install espeak libespeak-dev libudev-dev libsdl2-dev`
- Instalação: https://docs.qgroundcontrol.com/en/getting_started/download_and_install.html
- Executar Terminal:
- Site: <http://qgroundcontrol.com/>

APM Planner2:

- Programa: APM Planner2
- Versão: 2.0.24
- Tamanho Arquivo: 37 MB
- Espaço Ocupado em Disco: 300 MB
- Instalação: <http://firmware.us.ardupilot.org/Tools/APMPlanner/>
- Comando Instalação: `sudo dpkg -i apm_planner_2.0.24_xenial64.deb`
- Executar Terminal: `apmplanner2`
- Site: <http://ardupilot.org/planner2/index.html>

MAVProxy:

- Programa: MAVProxy
- Versão: 1.6.1
- Instalação: http://ardupilot.github.io/MAVProxy/html/getting_started/download_and_installation.html
- Dependências: `sudo apt install python-dev python-opencv python-wxgtk3.0 python-pip python-matplotlib python-pygame python-lxml`
- Instalação: `pip install MAVProxy`
- Configuração: Coloque-o nas variáveis de ambiente:
 - Comando: `$ echo "export PATH=$PATH:$HOME/.local/bin" >> ~/.bashrc`
- Site: <http://ardupilot.github.io/MAVProxy/html/index.html>
- Executar Terminal: `mavproxy.py`

Mission Planner:

- Programa: Mission Planner
- Versão: 1.3.48
- Tamanho Arquivo: 57 MB
- Instalação: <http://firmware.us.ardupilot.org/Tools/MissionPlanner/>
- Site: <http://ardupilot.org/planner/index.html>

LibrePilot (Não Usado):

- Programa: LibrePilot
- Versão: 16.09

- Tamanho Arquivo: 42 MB
- Instalação: <https://librepilot.atlassian.net/wiki/display/LPDOC/Downloads>
- Comando Instalação: `sudo dpkg -i librepilot_16.09-0trusty1_amd64.deb`
- Executar Terminal: `librepilot-gcs`
- Site: <https://www.librepilot.org/site/index.html>

Simuladores de Voo:

FlightGear:

- Programa: FlightGear
- Versão FlightGear: 2016.4.4
- Versão SimGear: 2016.4.4 (Vem junto no pacote)
- Tamanho Arquivo: 1587 MB
- Tamanho Ocupado em Disco: 2573 MB
- Instalação: `sudo apt install flightgear`
- Executar Terminal: `fgfs`
- Site: <http://www.flightgear.org/>

FGRUN:

- Programa: FGRUN
- Tamanho Arquivo: 1,5 MB
- Tamanho Ocupado em Disco: 5,5 MB
- Instalação: `sudo apt install fgrun`
- Executar Terminal: `fgrun`
- Site: http://wiki.flightgear.org/FlightGear_Launch_Control

MORSE:

- Programa: MORSE
- Versão: 1.4
- Instalação: <https://www.openrobots.org/morse/doc/stable/user/installation.html>
- Instalação: `sudo apt install morse-simulator`
- Site: <https://www.openrobots.org/wiki/morse/>

X-Plane10

Gazebo

Outras Ferramentas:

Google Earth Pro:

- Programa: Google Earth Pro
- Versão: Google Earth Pro 7.3.0.3832 (64-bit)
- Tamanho Arquivo: 57 MB
- Instalação: <https://www.google.com.br/earth/download/gep/agree.html>
- Executar Terminal: `google-earth`
- Executar Terminal: `google-earth-pro`
- Site: <https://www.google.com.br/earth/>

DroneKit:

- Programa: DroneKit

- Versão Dronekit: 2.9.0
- Versão Dronekit-SITL: 3.2.0
- Instalação: http://python.dronekit.io/guide/quick_start.html
- Dependências: `sudo apt-get install python-pip python-dev`
- Instalação Dronekit: `pip install dronekit`
- Instalação Dronekit SITL: `pip install dronekit-sitl`
- Executar Terminal: `dronekit-sitl`
- Site: <http://dronekit.io>
- Site: <http://python.dronekit.io/>

VirtualEnv:

Programa: VirtualEnv

Versão: 15.1.0

Instalação: `sudo pip install virtualenv`

Executar Terminal: `virtualenv env`

Ardupilot:

Instalação: <http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>

Site: <http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>

`git clone git://github.com/ArduPilot/ardupilot.git`

`cd ardupilot`

`git submodule update --init --recursive`

Screen

- Programa: Screen
- Versão: 4.05.00
- Descrição: O Screen é uma aplicação usada para rodar os programas na Intel Edison abaixo encontram-se alguns comandos suportados pelo screen:
- Mais informações acesse:
 - www.gnu.org/software/screen/
 - www.digitalocean.com/community/tutorials/how-to-install-and-use-screen-on-an-ubuntu-cloud-server
 - www.gnu.org/software/screen/manual/screen.html
- Comandos para instalar o screen: `sudo apt-get install screen`
- Comando para listar as aplicações que estão em execução:
 - `$screen -ls`
 - `$screen -list`
- Comando para matar uma aplicação que estava em execução no screen:
 - `$screen -X -S [session # que vc quer matar] quit`
- Criação:
 - `$ screen -S some_name proc`
- Kill detached session:
 - `$ screen -S some_name -X quit`
- Parâmetros:
 - `-S nome`: define um nome para a sessão.

8 - Dicas do FlighGear

Iniciar o flightgear com as opções libera um recurso interessante de visualização de mapa.
\$ fgfs --httpd=8080

Abra o navegador:

<http://localhost:8080/gui/map/>

Atalhos:

- F10: Mostra o menu de opções do FGFS
- F11: Abri o painel do AutoPilot da aeronave

Diretórios do FGFS:

A seguir estão listados uma série de diretórios importantes do FGFS

- Documentação: /usr/share/games/flightgear/Docs/
- Aeronaves: /usr/share/games/flightgear/Aircraft/
- Aeroportos: /usr/share/games/flightgear/Airports/
- Protocolos: /usr/share/games/flightgear/Protocol/
- Trajetórias de Voo: /usr/share/games/flightgear/AI/FlightPlans/

Comandos:

- Insert: Move o leme (rudder) para a esquerda
- h: habilita/desabilita o painel de controles
- v: muda o modo de visualização
- s: comando para dar partida na aeronave Cessna 172p

9 - Utilizando o AutoFG

OBS: Não se pode pausar a simulação, pois, afeta o PID

OBS: Não se pode iniciar o controlador muito antes de se iniciar o adaptador, pois, afeta o PID

Módulos

Módulo: aps-plotter-aircraft

Sempre configurar o local de início da aeronave: latitude, longitude, altitude

Módulo: aps-control-target

Sempre configurar o local de início da aeronave: latitude, longitude, altitude

Módulo: aps-model-flightgear

Sempre configurar o local de início da aeronave: latitude, longitude, altitude

Caso deseje adicionar entradas ou saídas do flightgear mudar:

```
model.flightgear.protocol.fields.input  
model.flightgear.protocol.fields.output
```

Caso adicionar falha alterar os campos:

```
model.flightgear.fail.time  
model.flightgear.fail.type
```

Caso queira especificar uma missão definir o mesmo em um arquivo:

exemplo: sdsc-mission.json

Deve-se sempre adicionar esse arquivo de missão no /config/aps-model-flightgear.properties

Módulo: aps-adapter-route

Quando mudar a minha rota para uma rota tridimensional do MPGA mudar:

```
Mpga.java  
método: getRoute()
```

```
MissionProcessor.java  
método: mpgaFinished()
```

Quando mudar o meu método de pouso mudar:

```
Mpga.java  
método: run(...)
```

Experimentos

4 mapas -> 4 falhas -> velocidade vento [10 knots]

Configuração do FlightGear

*Deve-se copiar o arquivo agcontrol.xml para a pasta /usr/share/games/flightgear/Protocol/

*Deve-se dar a permissão de execução para a pasta /usr/share/games/flightgear/Protocol/

*Ex. sudo chmod +777 Protocol

*Deve-se copiar a pasta Waypoint com todos os arquivos dentro para a pasta /usr/share/games/flightgear/Models/Waypoint/

*Deve-se dar permissão de execução para todos os arquivos na pasta Waypoint.

*Deve-se dar permissão de execução para todos os arquivos na pasta /usr/share/games/flightgear/AI/

Ordem para execução dos projetos

1 - Projeto: aps-model-flightgear
Rodar: ApsFlightGearModel.java

2 - Projeto: aps-control-target
Rodar: ApsTargetControl.java

3 - Projeto: aps-adapter-route
Rodar: ApsRouteAdapter.java

4 - Projeto: aps-plotter-aircraft
Rodar: ApsAircraftPlotter.java

5 - Projeto: aps-model-flighgear
Rodar: FlightGearRunner.java

GNU PLOT

Comando:

\$gnuplot arquivo.gplot

10 - Orientações para o Voo

Equipamentos para Check-up:

Antes de efetuar o voo do drone faça o check-up dos seguintes equipamentos:

- Primeiro verifique que não tenha nenhum fio solto.
- Verifique que a antena do GPS esteja bem presa.
- Verifique se a bateria está bem presa.
- Certifique-se que a bateria do Drone esteja carregada.
- Certifique-se que as pilhas do controle do drone estejam carregadas.
- Coloque a frente do drone (pés vermelho) para a mesma direção que você se encontra. Ou seja, você e o drone devem estar orientados na mesma direção.
- Verifique no Mission Planner ou QGroundControl a quantidade de GPS que estão sendo capturados.
- OBS: Para voar no modo Loiter é necessário pegar pelo menos 4 GPSs.
- Dica: Aprenda a primeiro a voar no modo Loiter. O drone estabiliza na posição que estiver (através do GPS).
- Dica: Depois, aprenda a voar no modo AltHold. O drone estabiliza apenas a altitude (através do barômetro).
- Dica: Por, fiz aprenda a voar no modo Stabilize. O drone não estabiliza nada, tudo é feito de forma manual.
- Antes de ligar o controle certifique-se que o Throttle esteja no mínimo.
- Antes de ligar o controle certifique-se que o modo esteja no Stabilize.
- Então faça os seguinte passos:
 - 1 - Ligue o controle.
 - 2 - Ligue a bateria.
 - 3 - Para armar coloque o controle do throttle para a direita (5 segundos)
 - 4 - Coloque no modo Loiter
 - 5 - Dê potência no motor para voar, após passar do centro do controle a aeronave começará a voar.
 - 6 - Quando quiser pousar, desça a aeronave até o solo.
 - 7 - Para desarmar coloque controle do throttle para a esquerda (5 segundos)
 - 8 - Desconecte a bateria.
 - 9 - Desligue o controle de rádio.
- Nunca aperte o botão de trim do controle.
- Não deixe ninguém mexer no controle do VANT.

11 - Outras Orientações

Para se conectar o computador com telemetria usando o software Mission Planner no Drone com piloto automático e com Telemetria faça o seguinte:

Vá no Mission Planner e dê um connect na porta COM5 com frequência 57600.

12 - Criando Rede Hotspot

Para se criar uma rede hotspot no celular smartphone Samsung Galaxy Y faça o seguinte:

- 1 - Abra o menu de “Configurações”.
- 2 - Abra o menu de “Conexões sem fio e rede”.
- 3 - Abra o menu de “Ancoragem e Roteador Wi-Fi”.
- 4 - Ative o “Ponto de acesso Wi-Fi”.
- 5 - Se necessário vá em “Defis. de ponto de acesso Wi-Fi”.
- 5.1 - Nesse menu vá novamente em “Defis. de ponto de acesso Wi-Fi”.
- 5.2 - Coloque o nome da rede criada.

Sugestões

Rede: RedeDroneC

Segurança: WPA2 PSK

Senha:

- 6 - Pronto a rede Wi-Fi já está criada.

13 - Acessando Intel Edison Sem Digitar Senha

1 FORMA:

1 - Instale o programa **sshpass**:

Ubuntu/Debian: `sudo apt-get install sshpass`

Fedora/CentOS: `yum install sshpass`

Arch: `pacman -S sshpass`

2 - Modo de usar:

`sshpass -p "sua_senha" ssh -o StrictHostKeyChecking=no SEU_USUARIO@SOME_SITE.COM`

Customizando a porta:

`sshpass -p "sua_senha" ssh -o StrictHostKeyChecking=no SEU_USUARIO@SOME_SITE.COM:2400`

3 - Exemplos de Uso:

`sshpass -p 'yourpassword' ssh user@ip`

`sshpass -p 'yourpassword' ssh root@192.168.43.9`

2 FORMA:

1 - Instale o programa **expect**:

Ubuntu/Debian: `sudo apt-get install expect`

2 - Tutoriais:

<http://www.admin-magazine.com/Articles/Automating-with-Expect-Scripts>

https://docs.oracle.com/cd/E35328_01/E35336/html/vmcli-script.html

<https://en.wikipedia.org/wiki/Expect>