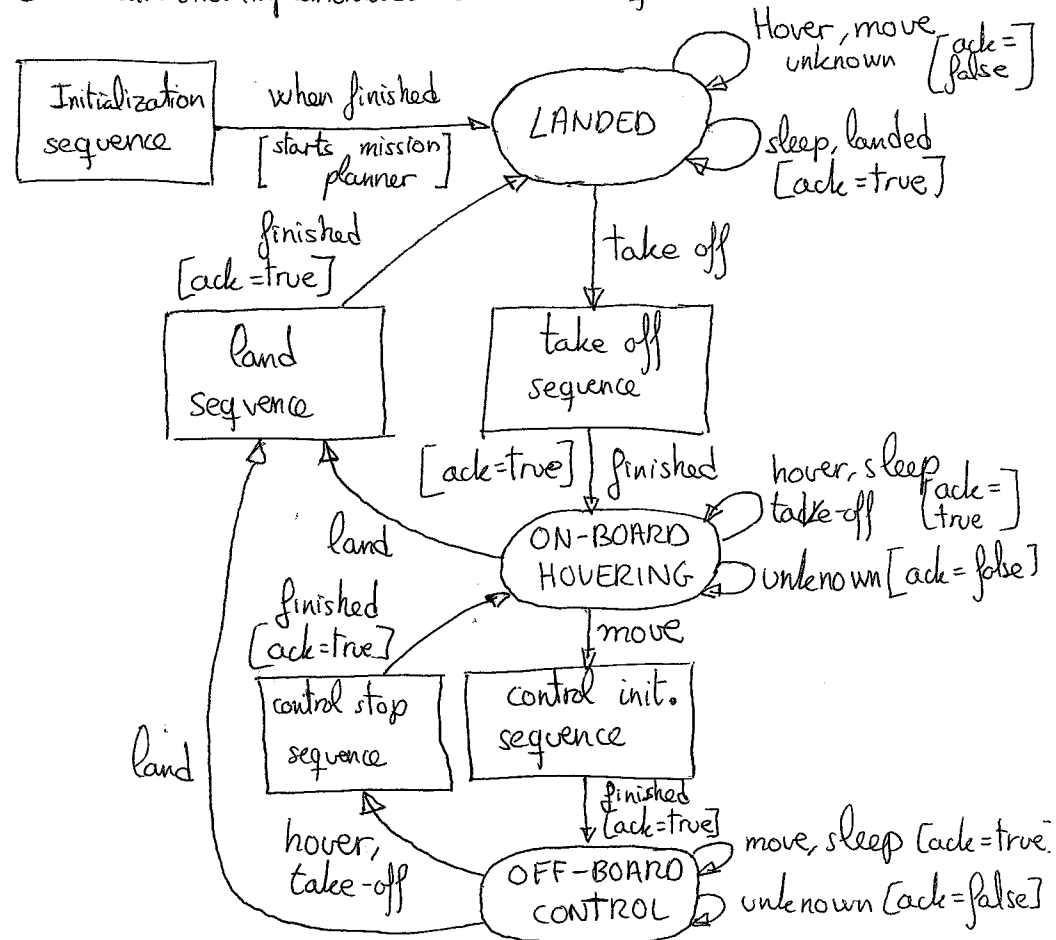


Drone Architecture Brain:

Inicio: el drone esta conectado por Wifi y todos los modulos incluidos el brain están "stopped". Al darle a la 's', ~~se~~ el brain se activa y éste activa el mission-planner. Brain queda a espera de ordenes

Monitoriza: estado de la conexión wifi; el modo de funcionamiento del su drone y su telemetria; posición estimada de su drone; posición estimada y brain.isStarted() de otros drones; si los otro modulos están en funcionamiento y si están started.

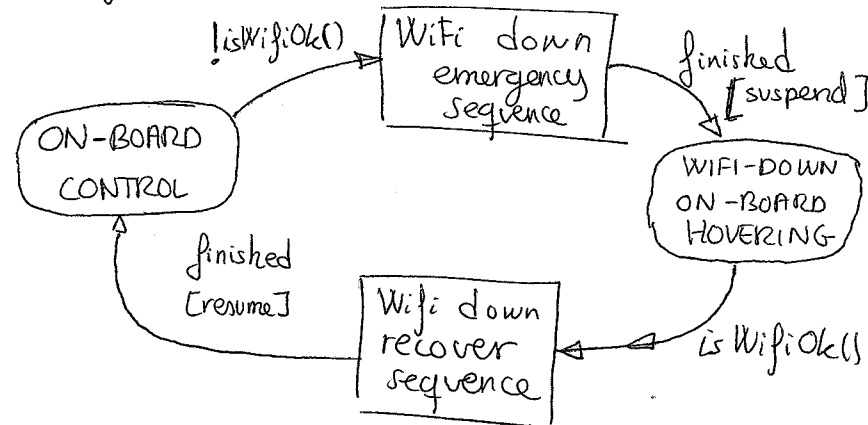
El Brain esta implementado como una máquina de estados:



Supers

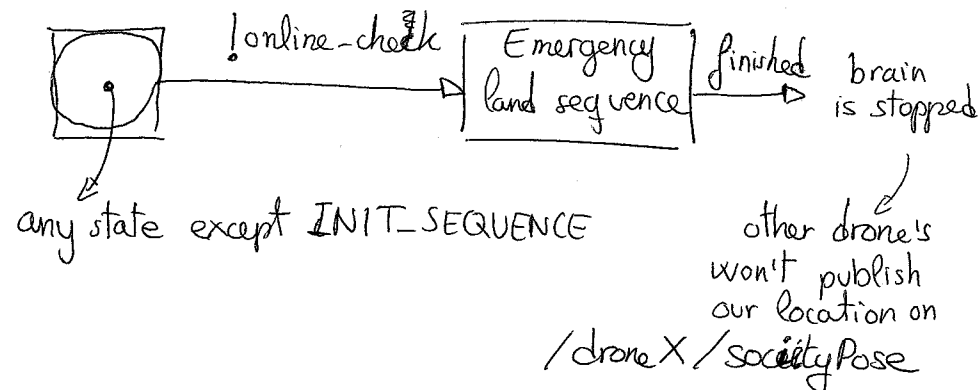
Failure recovery modes:

⊗ isWifiOk()



⊗ modules online check (online-check bool)

↳ not checked while in INIT SEQUENCE
↳ ~~an~~ currently there is a started-check bool which performs differently depending on the current state of the brain. started-check is currently unused.



Añadir piloto estado de vuelo. ✓ pasado código a Ramón ✓ ② Control posición a Ramón

Secuencia de inicio:

(salvo interferar) ↑

① Añadir piloto

Init sequence { Brain empieza iniciado (es el único)
0. esperar a que todos los módulos estén online
1. Start Mission Planner (cuando isWifiOk())
2. Quedarse a espera de ordenes

Wait for first take-off order

3. Te piden Take-off o sleep (solo digo ok)

Drone HL Command msg y Drone HL Command Ack. msg

4. Take-off

* Activación (comprobando si ya están encendidos o no)

ArucoEye

EKF

localizer

obstacle process

Trajectory planner

* Entonces take-off
y esperar a que haga hover

⊗ mandar ok a mission planner

Descripción de instrucciones de
mission planner >> brain

⊗ hover

1a hover ARDrone

⊗

Apagar controlador

1b

hover con controlador /* */

2 mandar ack

⊗ land

1 land y desactivar controlador

2 mandar ack (comprobar landed)

⊗ move in trajectory

"comprobar todo en orden"

encender controlador (position hover)

mandar ack

⊗ sleep

mandar ack

1 Si el estado interno muere
go task = ~~land~~ land
y brain >> stop()

2 ~~Wifi~~ Wifi not ok