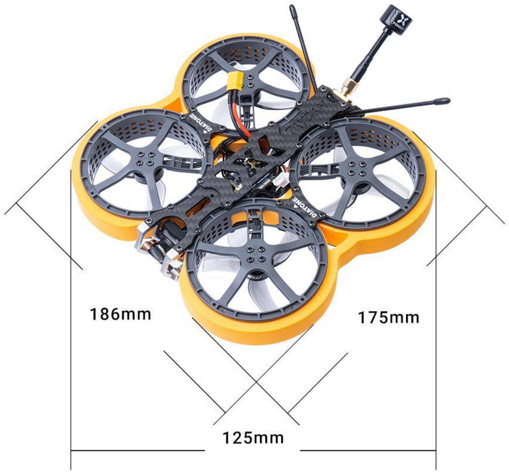


DSI

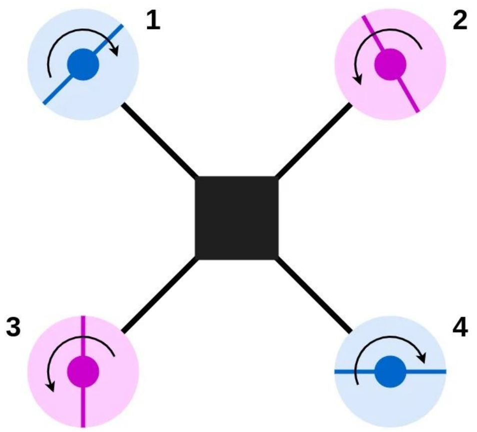
Group C Project

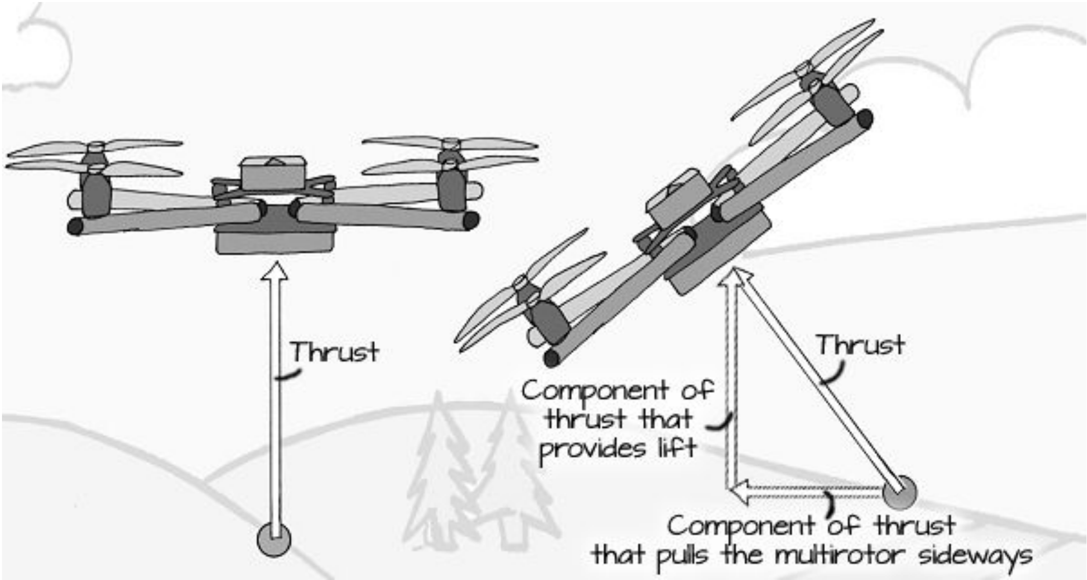
Autopilot for quadcopter

Felipe Gil Castiñeira
felipe@uvigo.gal

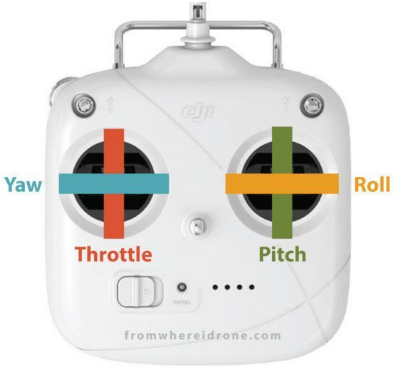
AtlantTIC Research Center for Information & Communication Technologies
UniversidadeVigo

Typical “hobby” drone: quadcopter





Controlling a drone

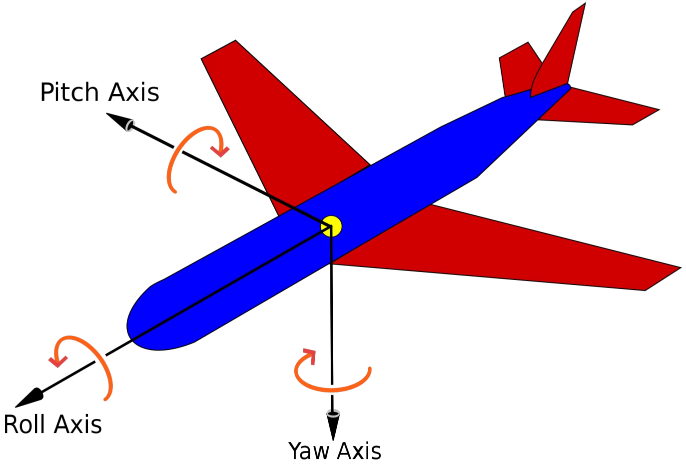


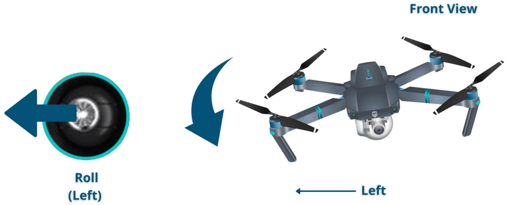
Throttle
controls how much lift your drone is creating which allows it to ascend and descend

Yaw
rotates the drone around its center either clockwise or counterclockwise

Pitch
controls the forward and backward movement of your drone

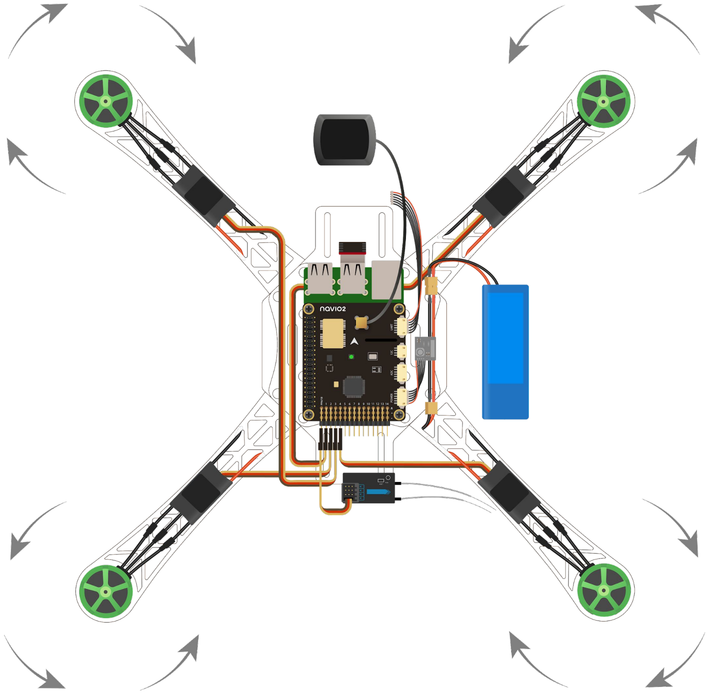
Roll
controls the right and left movement of your drone

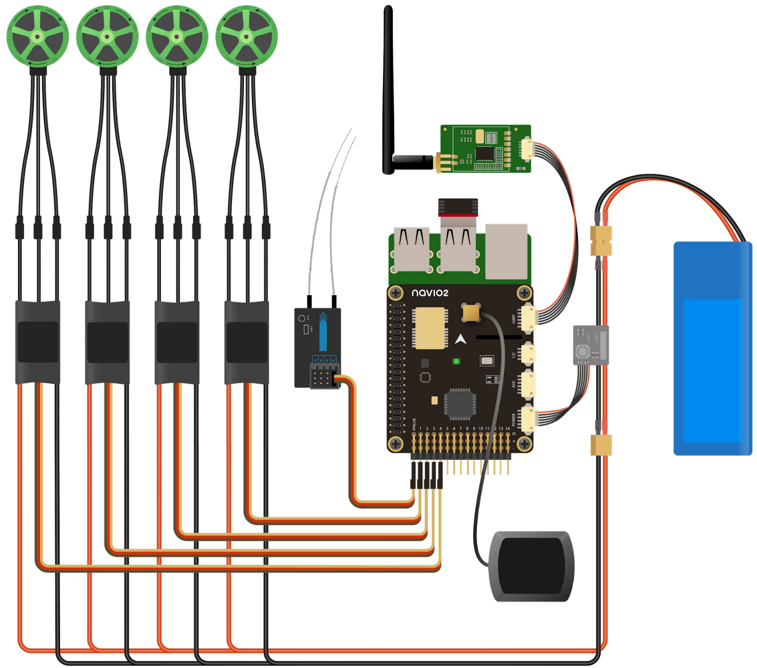




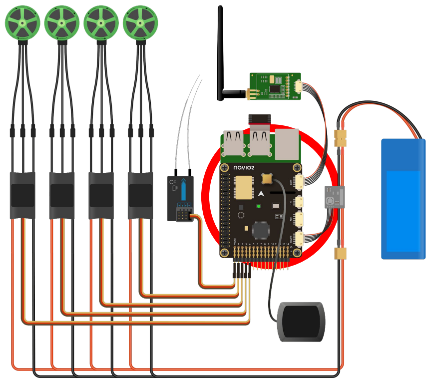
Quadcopter

System components (I)





System components (II)



Turn your Raspberry Pi®
into a drone controller

By using the might of sensors and controllers onboard

GNSS receiver

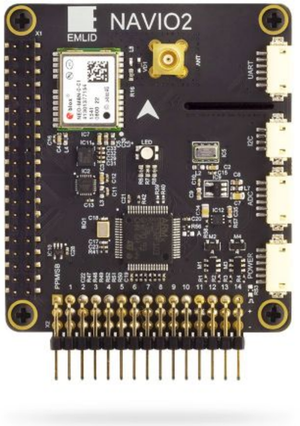
Tracks GPS, GLONASS, Beidou, Galileo and SBAS satellites. External antenna with MCX connector

Dual IMU

Accelerometers, gyroscopes and magnetometers for orientation and motion sensing

RC I/O co-processor

Accepts PPM/SBUS input and provides 14 PWM output channels for motors and servos



High resolution barometer

Senses altitude with 10 cm resolution

Extension ports

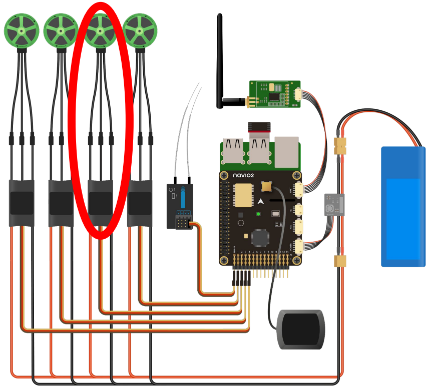
Exposed ADC, I2C and UART interfaces for sensors and radios

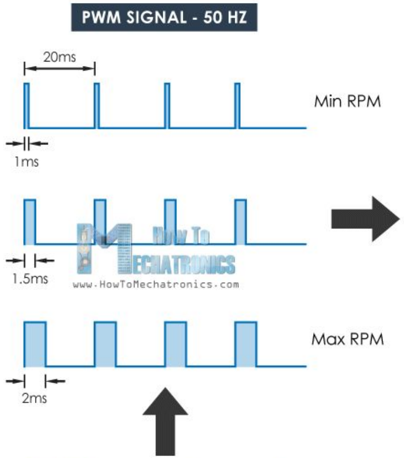
Triple redundant power supply

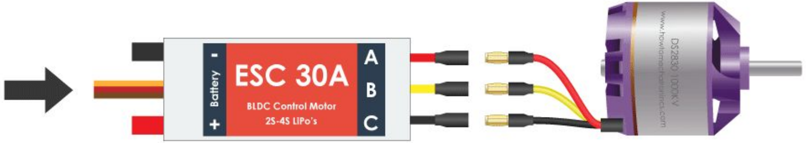
With overvoltage protection and powe module port for voltage and current sensing

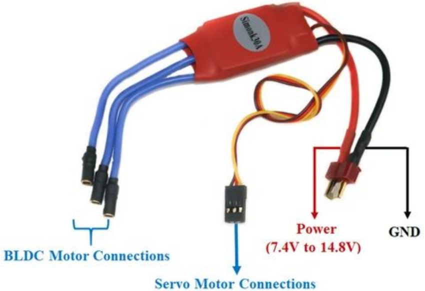
https://docs.emlid.com/navio2/

System components (III): ESC (Electronic Speed Controller)

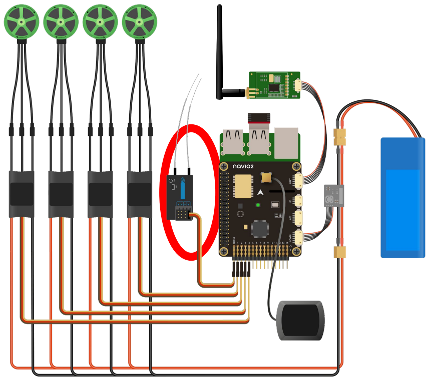


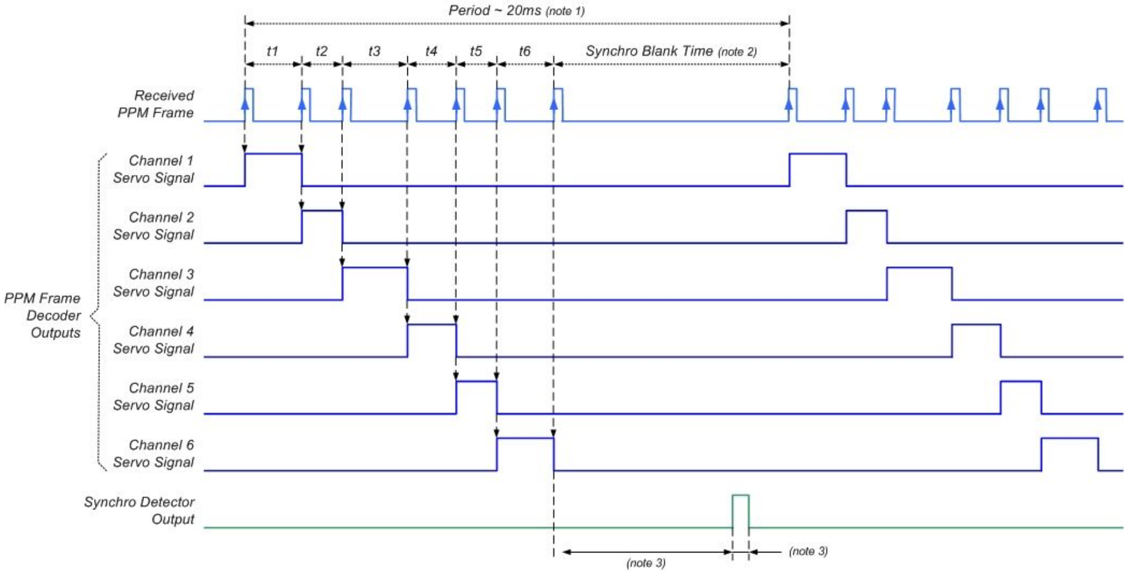






System components (IV): Radio receiver



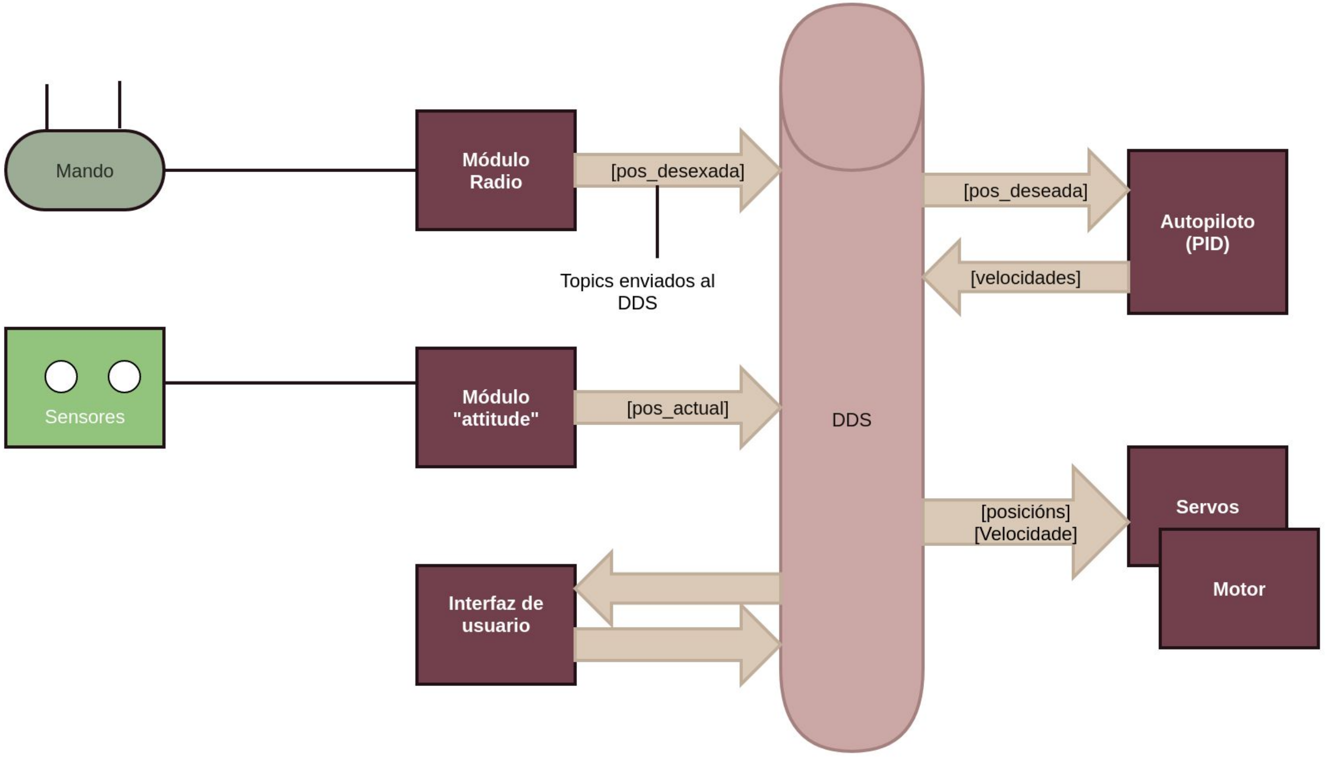


Software implementation (I)

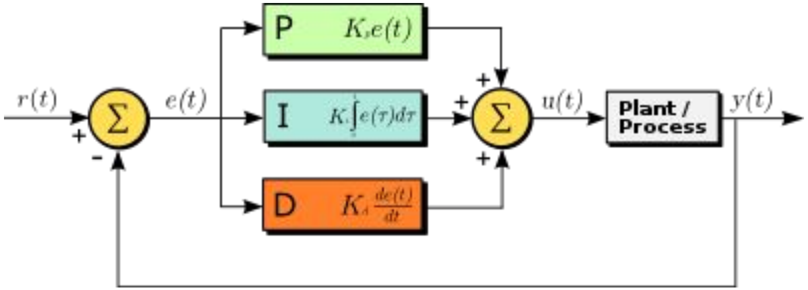
Linux based using a Raspberry

- Not the best option for an autopilot
  - Real time limitations
  - A microcontroller is a better option (cheaper, lighter, real-time, lower consumption…)
- But… easier to use in a time-constrained environment (this course)
  - Short learning curve (you all can make Linux programs).
  - Versatile (you can program using C, C++, Rust, Python… implement node.js servers, etc.).
  - Navio2 simplifies using the sensors with libraries:
    - https://docs.emlid.com/navio2/dev/navio-repository-cloning

Software implementation (II)



PID



$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt},$$

```
previous_error := 0
integral := 0

loop:
   error := setpoint − measured_value
   integral := integral + error × dt
   derivative := (error − previous_error) / dt
   output := Kp × error + Ki × integral + Kd × derivative
   previous_error := error
   wait(dt)
   goto loop
```

More things you will have to do/learn

- Setup the Emlid Linux distribution in your Raspberry.
  - Learn how to connect (recommended: use your phone as an AP and connect through Wi-Fi)

- “Arm” the motor (the ESC doesn’t work if you don’t send an “initialization” sequence --security mechanism for avoiding accidents--).
- Use CMake or other build environment (simplifies development).
- Use DDS/Zenoh.
- Know the attitude of your drone (current “roll”, “pitch” and “yaw” angles).
- Implement a PID.
- Transform the output of the PID into actions (new angles for the servos and speed for the motor).
- Debug a complex system.
- Deal with frustration.

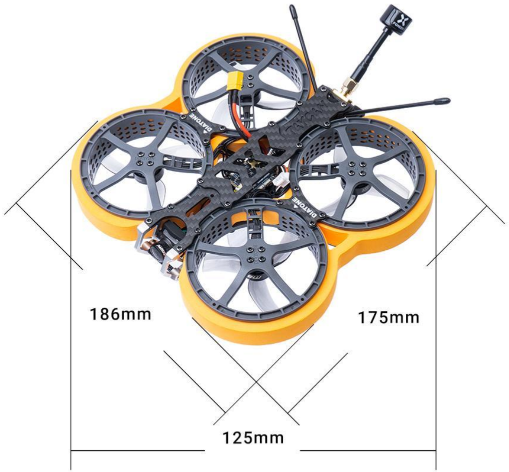
First tasks (sprint one)

- Have a meeting to coordinate the team
  - Check if everyone understood the project
  - Assign roles in the team (scrum master, product owner, team members)
  - Assign initial tasks
- Install the base system (e.g. Emlid image) in the Raspberry.
- Study and discuss how the system works.
  - Understand how the Emlid drivers for the Navio2 are implemented.
- Understand how a PID works and discuss about how the PID is going to be implemented.
- Learn about Zenoh.
  - Install Zenoh in your Linux computer and/or the Raspberry Pi.
- Discuss about the architecture of your system
  - Your system is going to be implemented as a distributed system (interconnected with Zenoh). Discuss about the blocks and functionality for each one.
- Discuss about the programming language (recommended C, C++, but other can be used).

DSI

Group C Project

Autopilot for quadcopter



AtlantTIC Research Center for Information & Communication Technologies

Universida de Vigo