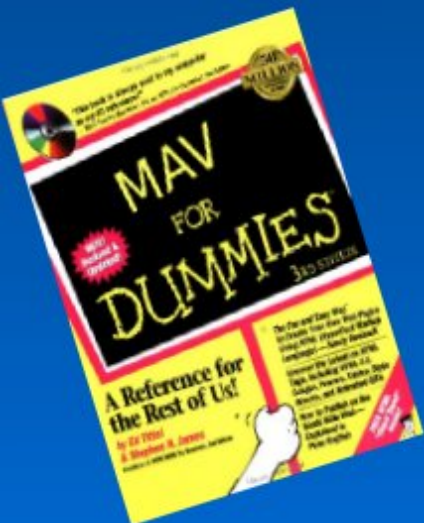


# PaparaDzIY

Pascal Brisset and Antoine Drouin

ENAC-CENA  
Toulouse, France

September 15, 2004



# Project Paparazzi



## Quick and Dirty UAV

- Hobby
- Low budget
- Two persons
- Two years
- JMD'03, EMAV'04 flight competitions

Learned lessons ?

Do It Yourself: Guidelines and tools for amateurs.

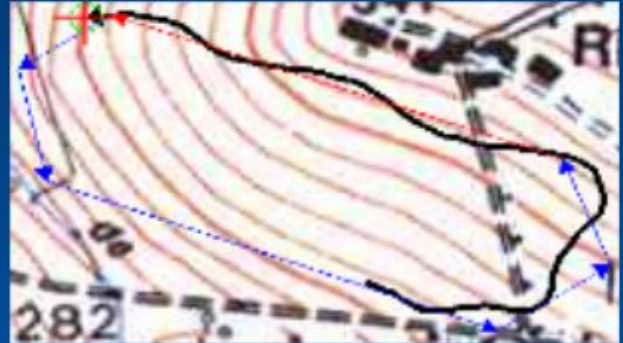


# Outline

- **Objectives and Motivation**
- Components
  - Aircraft Model
  - Autonomous Flight
  - Simulation and Flight Tests
- Methodology
  - Safety
  - COTS
- Conclusion

# Goals

- Autonomous flight
- Safety
- Low cost
- Low weight
- Open project



[www.nongnu.org/paparazzi](http://www.nongnu.org/paparazzi)



# Motivations

- To understand and learn
  - Electronics (sensors, EMI, RF,...)
  - Automatics (data fusion, control loops)
  - Software (airborne systems, datalink, HMI,...)
  - Flight mechanics and aerodynamics
- To have fun
- To take part in flight competitions



# Outline

- Objectives and Motivation
- **Components**
  - Aircraft Model
  - Autonomous Flight
  - Simulation and Flight Tests
- Methodology
  - Safety
  - COTS
- Conclusion

# Model Aeronautics



Not well described in books. Based on experience

- Assembly : nothing to reinvent - capitalize on model aircraft experience.
- Pilot : good skills for tricky situations.



# Airframe



First goal: To fly

Ex: Twinstar Multiplex

- “Ready to fly”
- Easy to operate
  - Electric
  - Hand launch
  - Heavy duty



Fly well, fly often





# Safe Manual Control

- PPM/PCM decoding
- Actuators command
- No radio-command programming
- Robust minimal code

Paparazzi solution:

- Automatic code generation
- 4ko



# Downlink

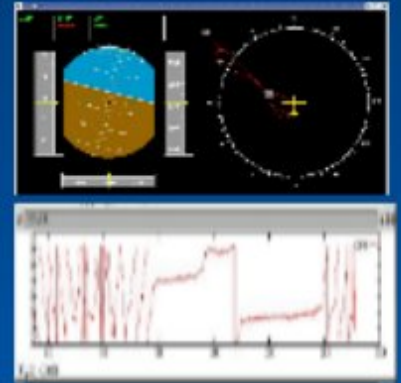


Monitoring of the flight parameters is required:

- Real time
- Post flight analysis

Paparazzi solution:

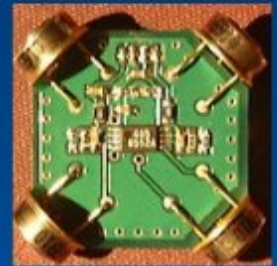
- Modems and audio channel of video camera transmitter
- Generic protocol



# Sensors



- Attitude : differential infrared thermometer
  - absolute data
  - no drift - computationally light



- GPS
  - Position
  - Course
  - Altitude
  - Speed



# 1st step : Automatic Flight

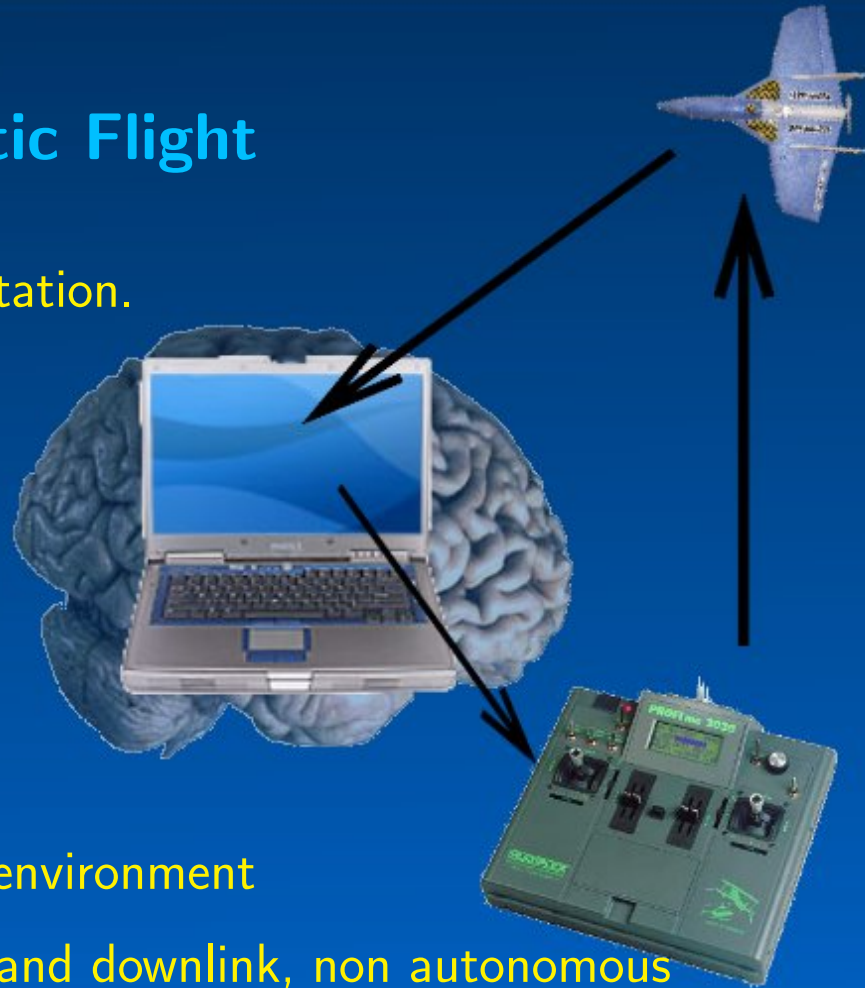
Control from the ground station.

Requires only:

- Stabilizer
- GPS receiver
- Computer-RC link

Pros: Simplicity, software environment

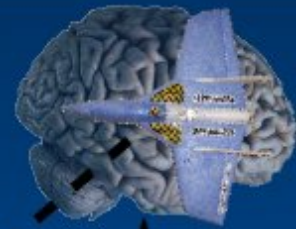
Cons: Latency, critical up and downlink, non autonomous



## 2nd step : Autonomous Flight

Airborne controllers: stabilization and navigation

- Attitude
- Heading
- Waypoint
- Track
- Mission





# Simulation

## Basic flight model

- Debug and non-regression test
- Help to adjust navigation algorithms
- Difficult to be realistic for low level controllers tuning

Paparazzi solution: “Hardware in the loop” simulator

# Flight Tests

- Prepare a program
- Follow the program as much as possible
- Monitor the flight
- Analyze recorded data
- Archive them









# Outline

- Objectives and Motivation
- Components
  - Aircraft Model
  - Autonomous Flight
  - Simulation and Flight Tests
- **Methodology**
  - Safety
  - COTS
- Conclusion



# Safety

- Design
  - Segregate critical code
  - Implement fail safe modes
    - \* Automatic motor cut upon ground proximity
    - \* Basic “back home” navigation mode
- Operation
  - Be paranoid (if it may fail, it will fail)
  - Use check lists
  - Respect model aircraft rules



# Extreme Programming

[www.extremeprogramming.org](http://www.extremeprogramming.org) : a software methodology for “risky projects with dynamic requirements”

Some rules:

- Refactor whenever and wherever possible
- Unit tests
- Integrate often
- Make frequent small releases
- All production code is pair programmed

# Use “open” COTS



Need for a full control of the components.

- RC receiver:
  - Difficult to tune and build (HF part)
  - Piggy-backed a commercial unit
- FMA stabilizer:
  - Keep the infrared sensor
  - Dump the controller



# Open Software



Natural answer: the Free Software solution (FSF, GNU)

- Understanding and fixing
- Active support
- Expanding, adapting

Examples : Linux, Gcc, Autopilot, Rtty, FlightGear, Ivy, Zinc,...





[www.openatc.org](http://www.openatc.org)

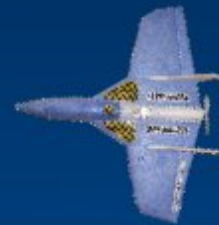


[www.flightgear.org](http://www.flightgear.org)



UAV control station

# Paparazzi Control Station



The screenshot displays the Paparazzi Control Station interface, which is used for controlling a multi-rotor aircraft. The interface is divided into several panels:

- Top Left:** A topographic map showing the flight path of the aircraft. The path is marked with a blue line, and several waypoints are labeled, including "la Calle", "Ricou", "la Bourdette", and "Pourciel".
- Top Right:** A status panel showing the aircraft's current position and altitude. It includes a heading indicator (a circular dial) and a vertical scale for altitude, with a current reading of 489. The heading is 120 degrees.
- Bottom Left:** A video panel showing a live feed of the camera mounted on the aircraft. The video shows a view of the ground, including a road and some vegetation.
- Bottom Right:** A code editor and a graph panel. The code editor shows the flight plan in XML format, including waypoints and mission parameters. The graph panel displays the aircraft's altitude over time, with a red line representing the altitude and a vertical scale on the right.

```
<block name="init">
  <mission name="mission">
    <heading course="0" mode="g" speed="0" roll="0" yaw="0" pitch="0" yaw_rate="0" />
    <heading course="0" pitch="0" mode="llh" climb="0" roll="0" yaw="0" yaw_rate="0" />
  </block>
</block name="init">
<block name="land">
  <condition cond="(Roll>10)" duration="land" />
  <wait />
  <go spin="1" pitch="rc_pitch" />
  <go spin="2" pitch="rc_pitch" />
  <go spin="3" pitch="rc_pitch" />
  <go spin="4" pitch="rc_pitch" />
</block>
</block name="land">
<block name="land">
  <condition cond="(Roll>10)" duration="init" />
  <go spin="1" />
  <go spin="2" />
  <go spin="3" heading="route" pitch="0,1" mode="glide" />
  <go spin="4" />
  <goto block="land" />
</block>
</block name="init">
```

Graph 1: Altitude vs Time (0 to 150 seconds). The altitude starts at approximately 480m and fluctuates between 470m and 510m.

Graph 2: Altitude vs Time (0 to 150 seconds). The altitude starts at approximately 480m and remains relatively stable around 480m.



# Real UAV is too expensive to play!

Do it yourself with Paparazzi

- Request for support, documentation, features
- Find bugs!

First users

- Fondtech: Minimal cost landmine survey
- Adelaide University: Computer vision
- ENAC: Teaching





<http://www.nongnu.org/paparazzi>



# By the way : nothing new for terrorists



Recurrent question on open project UAV forums

- Commercially available elsewhere
- Not a turnkey system
- High complexity, low payload, small range

Why would they bother building their own?