Paparazzi - The Free Autopilot: Scientific Usage

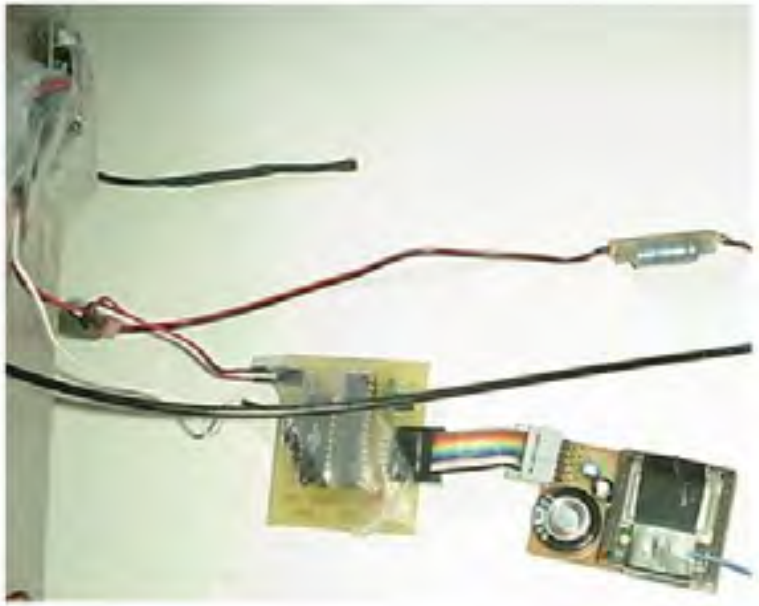
Martin Müller (MME) / Antoine Drouin (ENAC)

COST Workshop March 2011

History of Paparazzi

History of the Project

2003



History of the Project

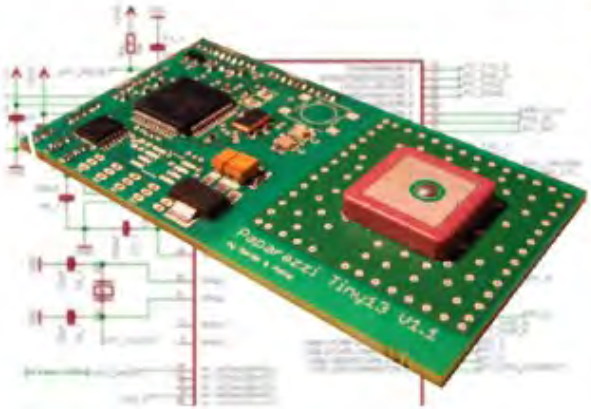
History of the Project

2004

History of the Project

2005

History of the Project



History of the Project

2006

History of the Project






History of the Project

2007

History of the Project

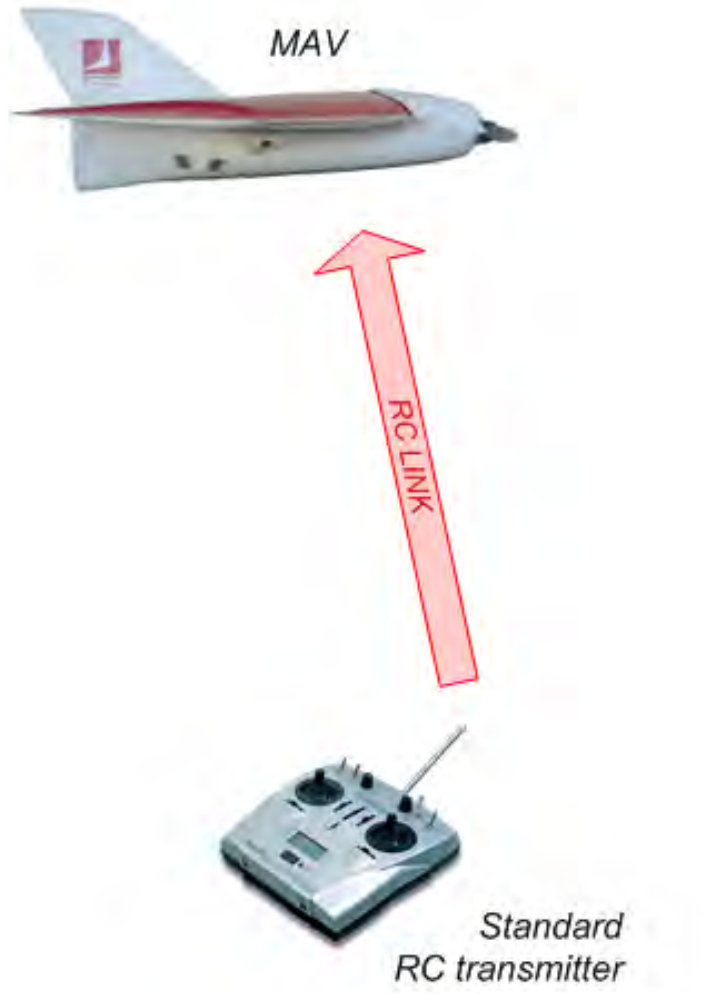
2008

History of the Project

2009

What is Paparazzi

What is Paparazzi

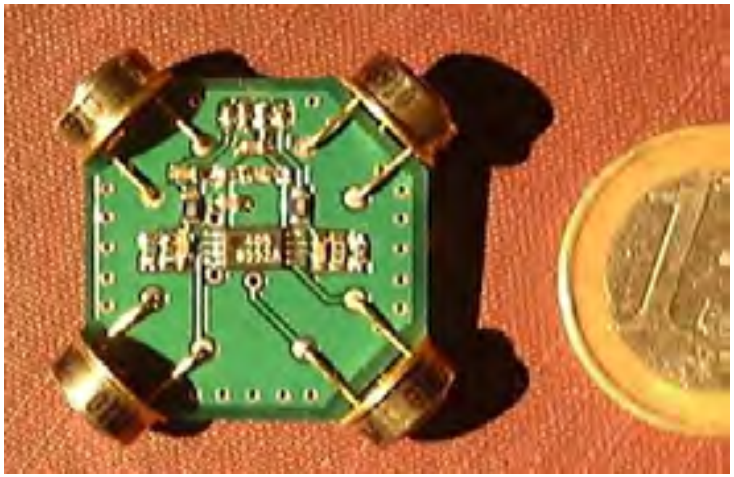


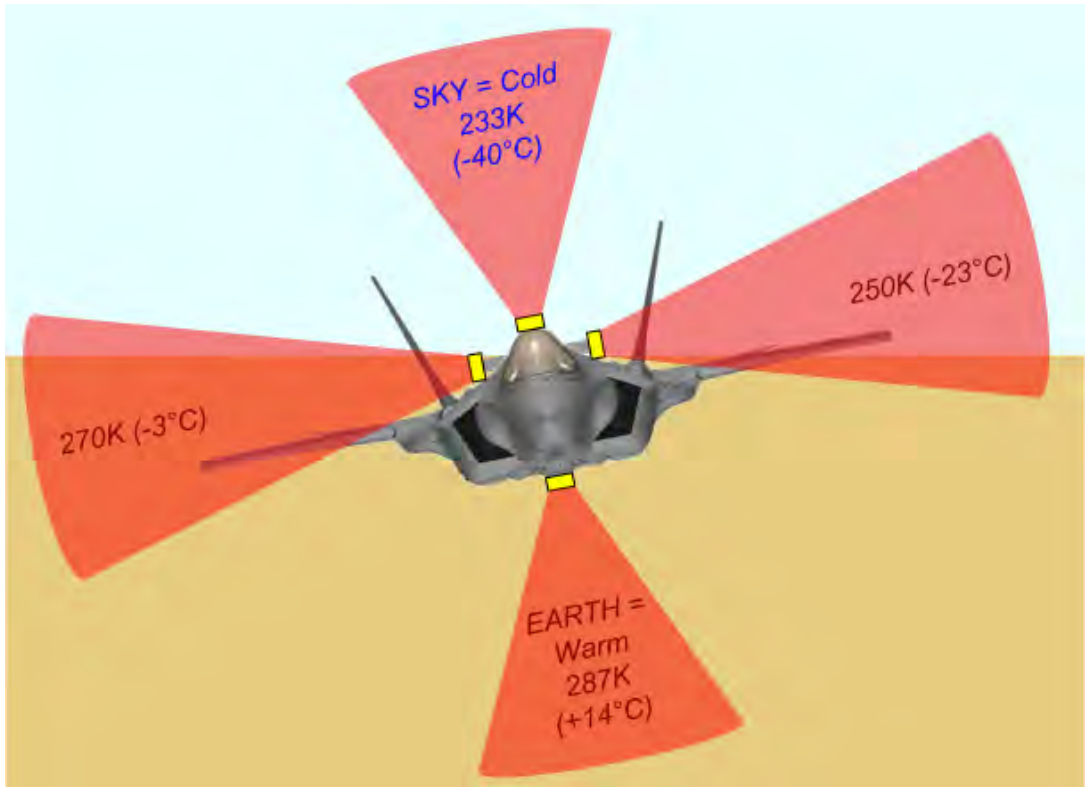
What is Paparazzi

What is Paparazzi

Paparazzi attitude sensors

Infrared Sensors (thermopiles)



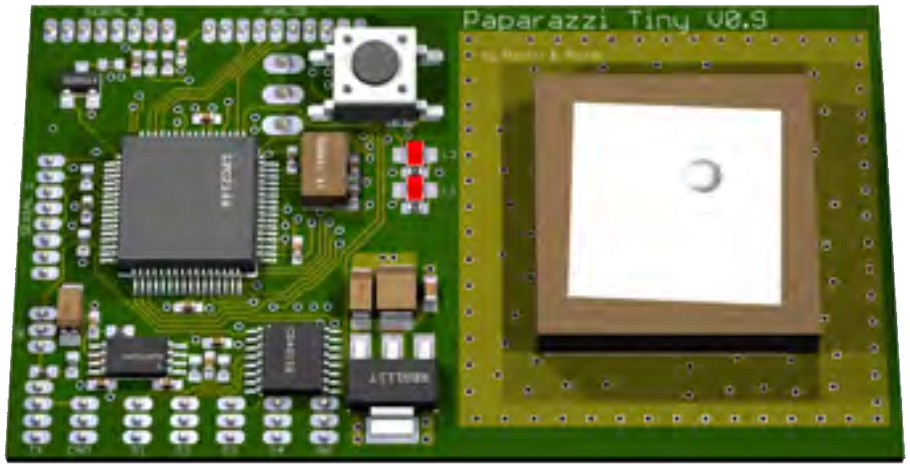


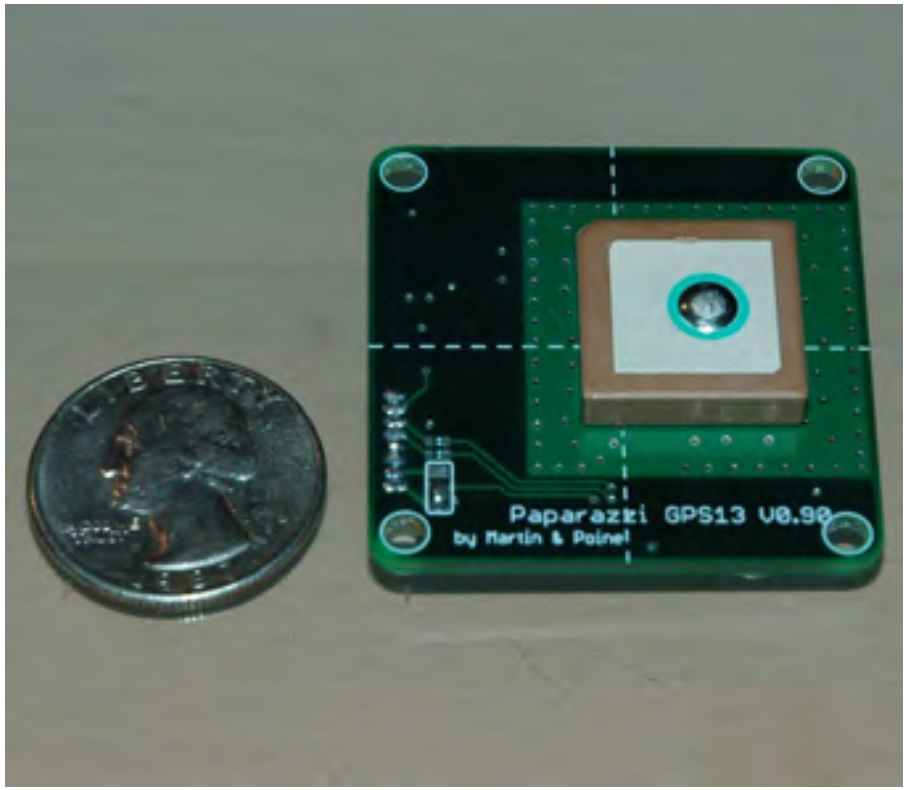
Paparazzi attitude sensors

Inertial Measurement Units (IMU)

Position sensors

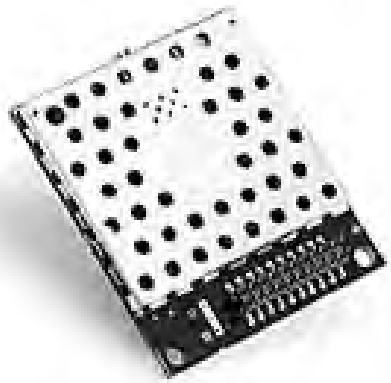
GPS



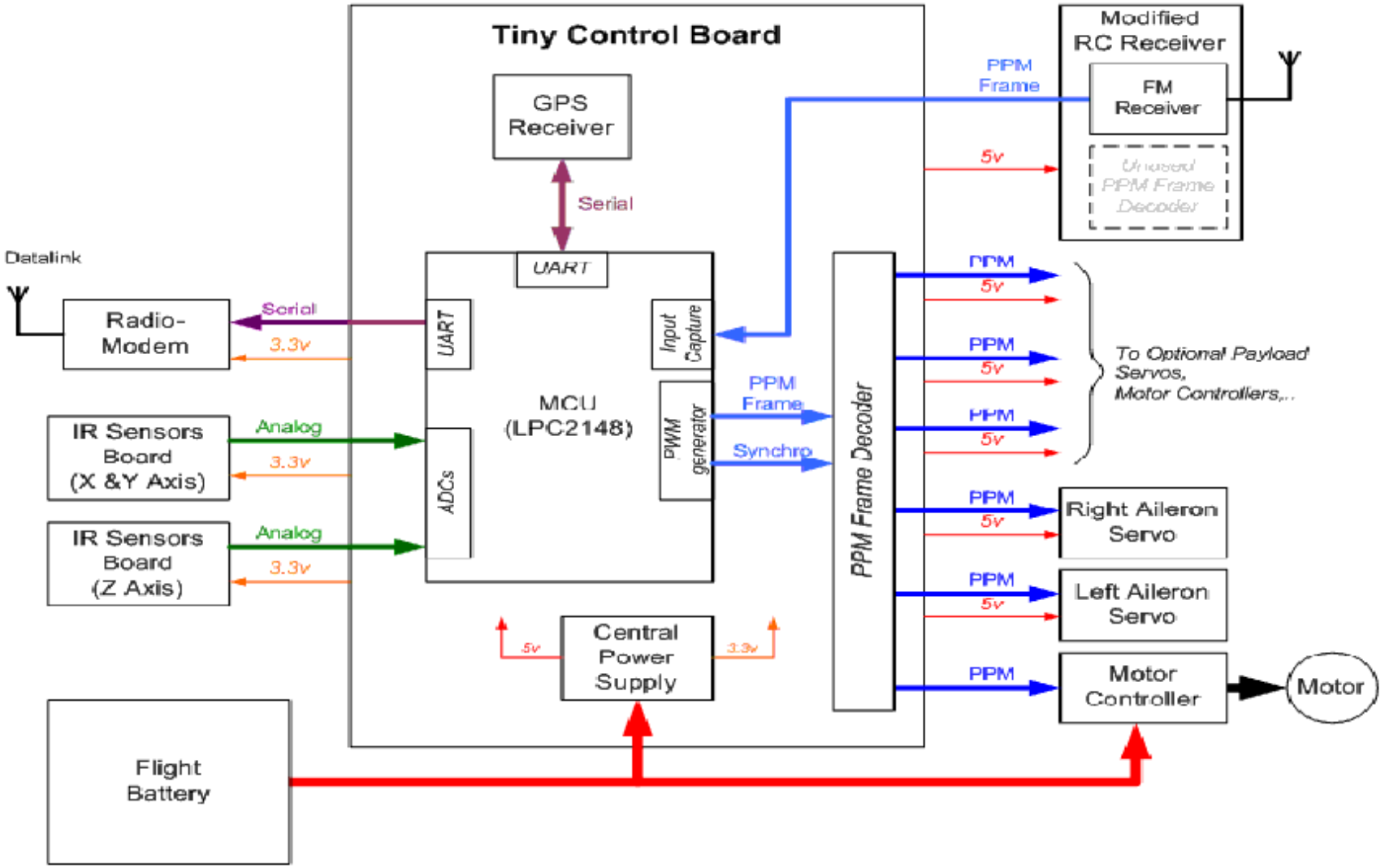


Communication

Modems



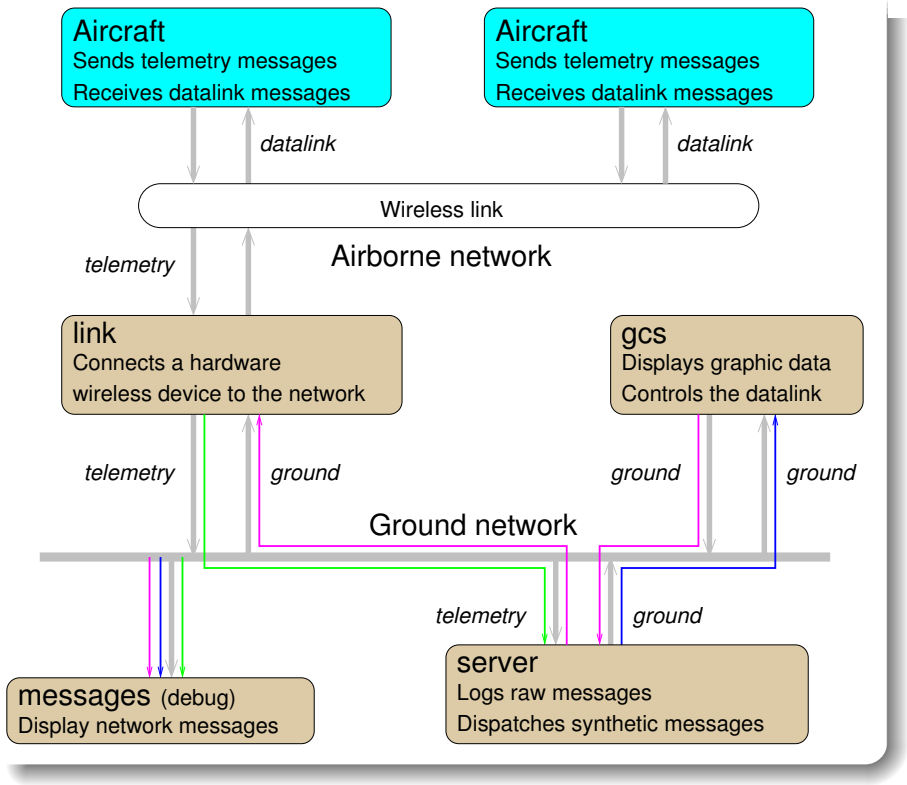
System Architecture Hardware



Distributed architecture

Multi Agents

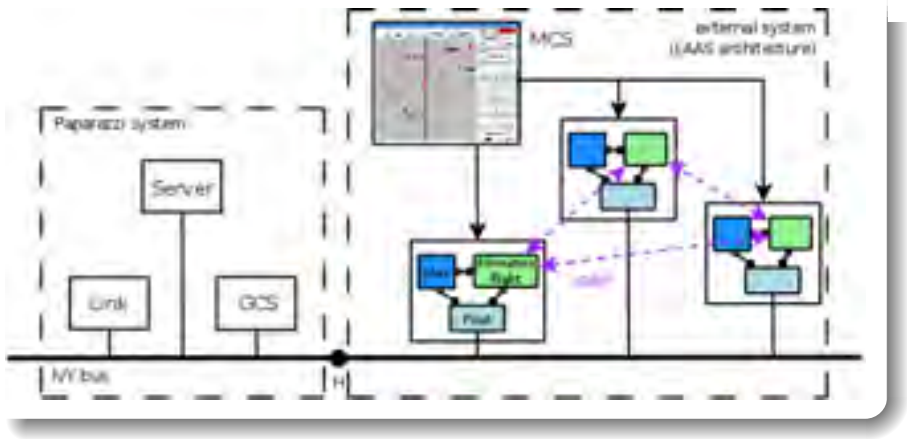
- Small programs performing small well defined tasks
- Flexibility for development
- Ease of connection with other systems
- Flexibility of operation



Distributed architecture

Multi Agents

- Small programs performing small well defined tasks
- Flexibility for development
- Ease of connection with other systems
- Flexibility of operation



Distributed architecture

Multi Agents

- Small programs performing small well defined tasks
- Flexibility for development
- Ease of connection with other systems
- Flexibility of operation

On Board Software Modules

1. **gp2y** Sharp GP2Y1010AU0F particle sensor
2. **dpicco** IST DigiPicco humidity sensor
3. **hih** Honeywell HIH-4030 humidity sensor
4. **sht** Sensirion SHT75 humidity sensor
5. **mlx** Melexis 90614 infrared radiation sensor
6. **temt** Vishay TEMT6000 ambient light sensor
7. **lm75** National LM75 temperature sensor
8. **temod** Hygrosens TEMOD-I2C-Rx temperature sensor for PT1000
9. **tmp102** Texas Instruments TMP102 temperature sensor

On Board Software Modules contd.

1. **enose** TGS822 chemical sensor
2. **adc** driver for airspeed sensor based on analog voltage
3. **ets** Eagle Tree Systems airspeed sensor
4. **srf08** Devantech Ultrasonic Range Finder SRF08
5. **bmp** driver for Bosch BMP085 pressure sensor
6. **ets** Eagle Tree Systems pressure sensor
7. **MS5534A** Intersema MS5534A pressure sensor
8. **scp** VTI SCP1000 pressure sensor
9. **micromag** PNI Micromag magnetic sensor

On Board Software Modules contd.

1. **digital cam** shoot/zoom/on/off/take pictures at regular intervals
2. **cam** pointing of a simple camera (servo for tilt, heading for pan)
3. **cam roll** roll camera
4. **drop** drop mechanism
5. **light** control blinking speed of the LEDs for night flights
6. **formation** formation flight control for fixed-wing aircraft

Data aquisition setup

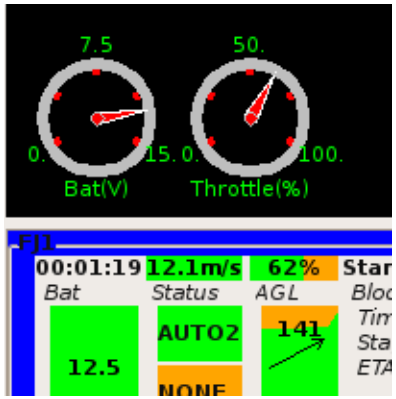
1. payload data collection
2. timestamping
3. logging

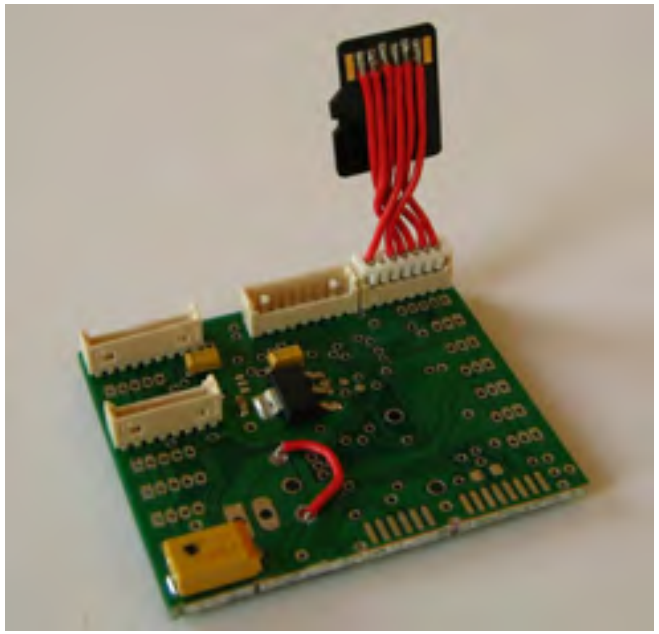
example message definition

```
<message name="HIH_STATUS" id="96">
  <field name="humid" type="uint16"/>
  <field name="fhumid" type="float" unit="rel_hum"/>
  <field name="ftemp" type="float" unit="deg_celsius"/>
</message>
```

Data aquisition

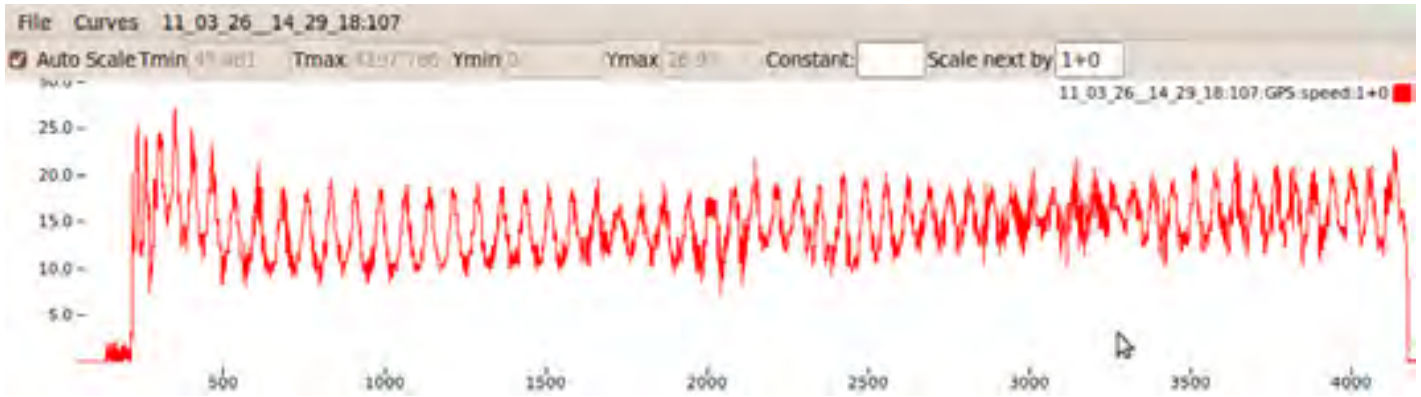
Live data through autopilot or extra measurement computer

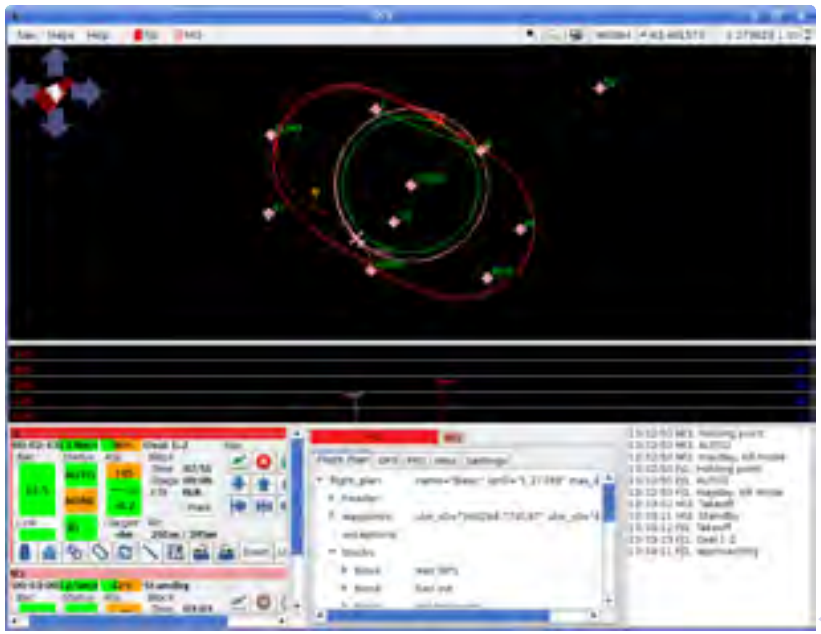




Post Flight Analysis

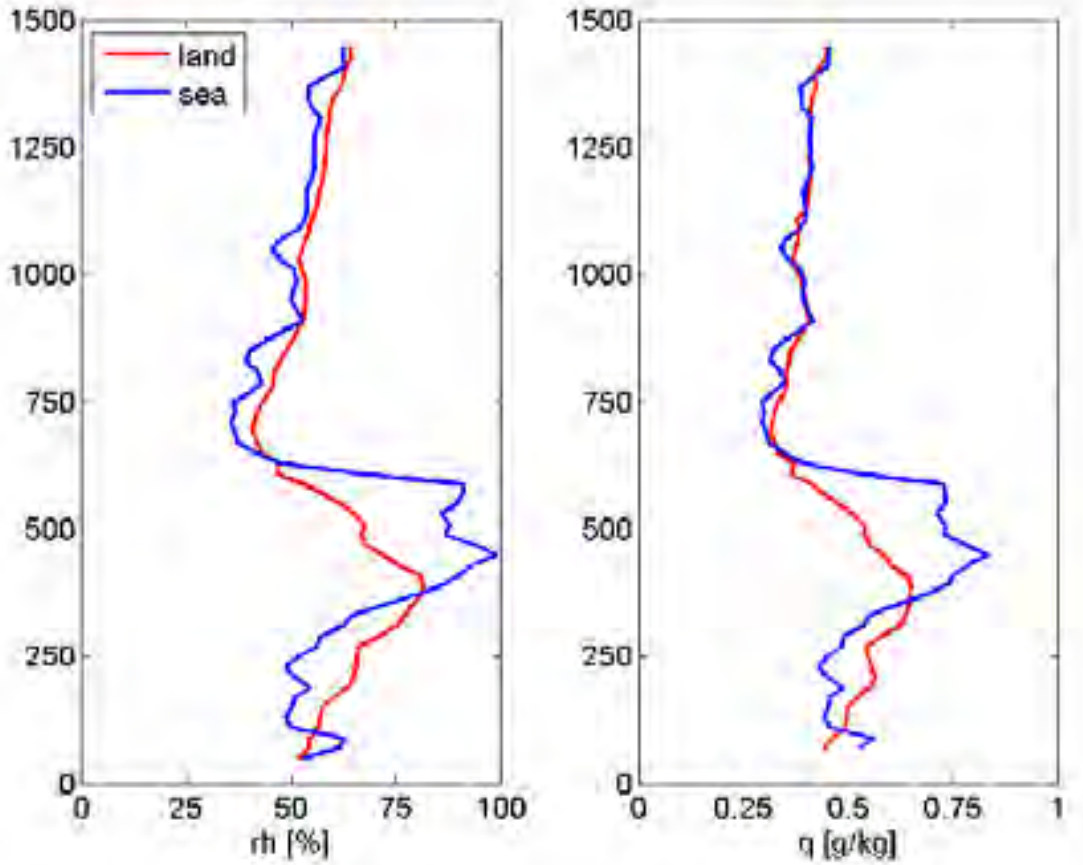
Live or recorded data analysis, replay



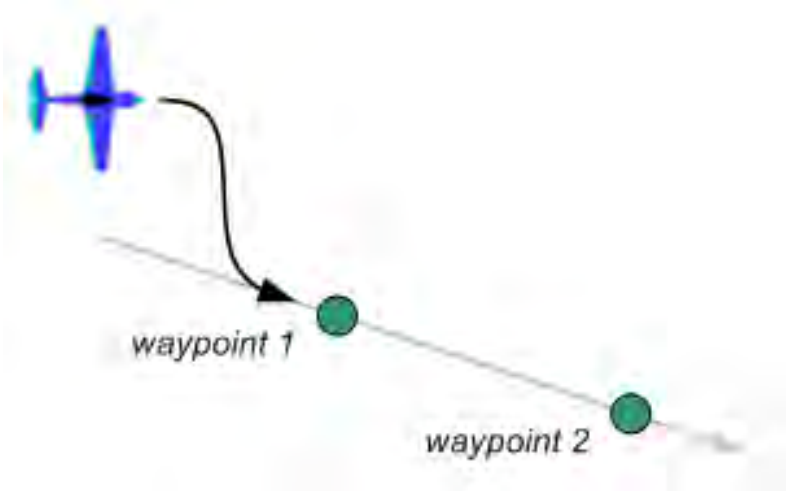


Post Flight Analysis

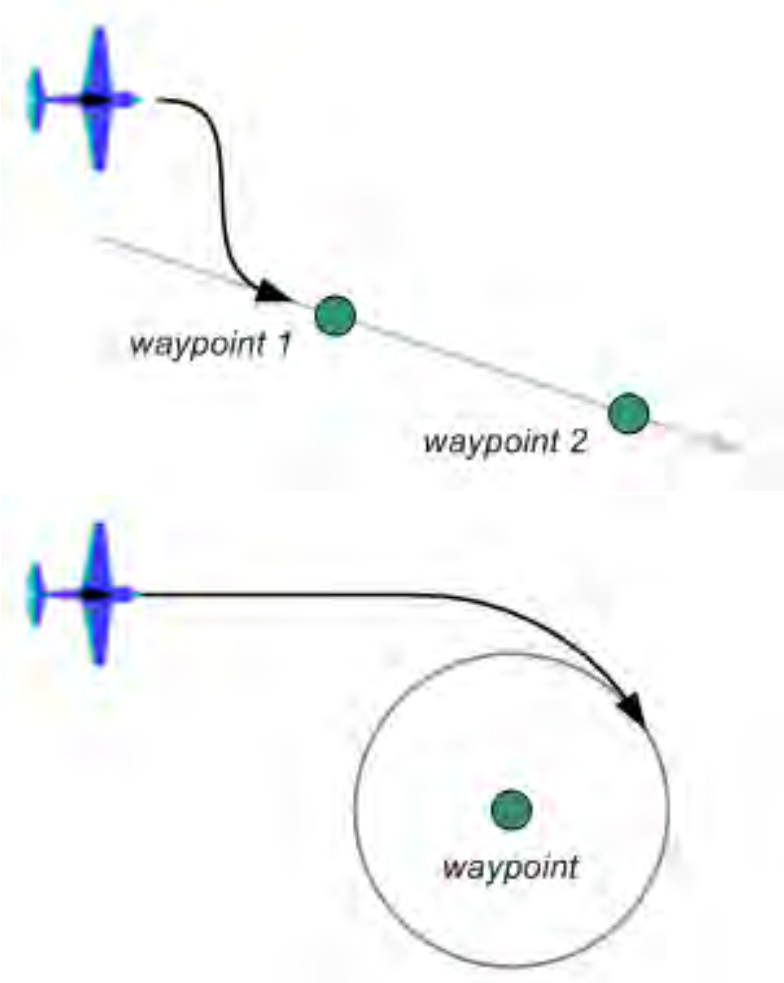
Import data to data analysis tools



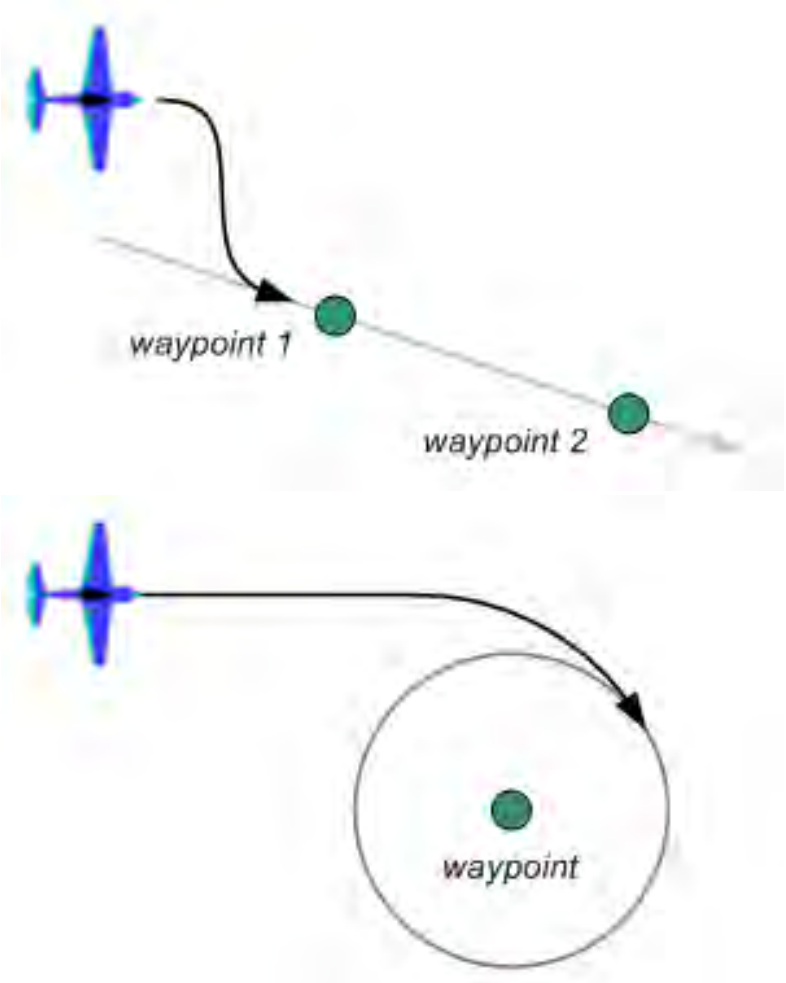
Flight Plan Language

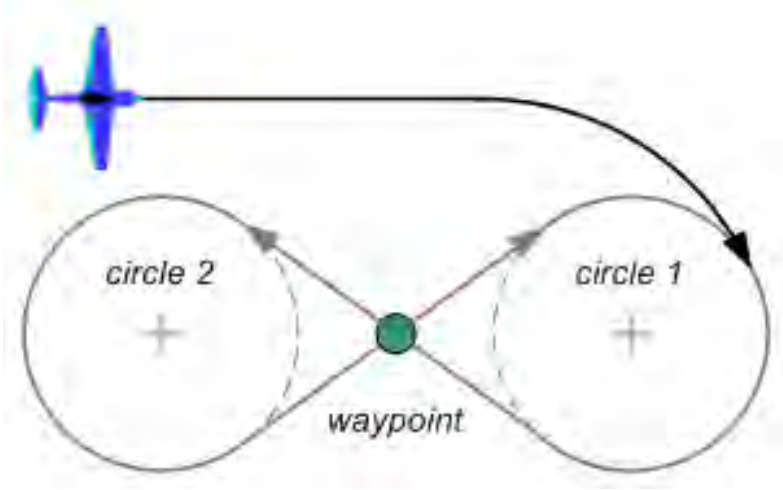


Flight Plan Language

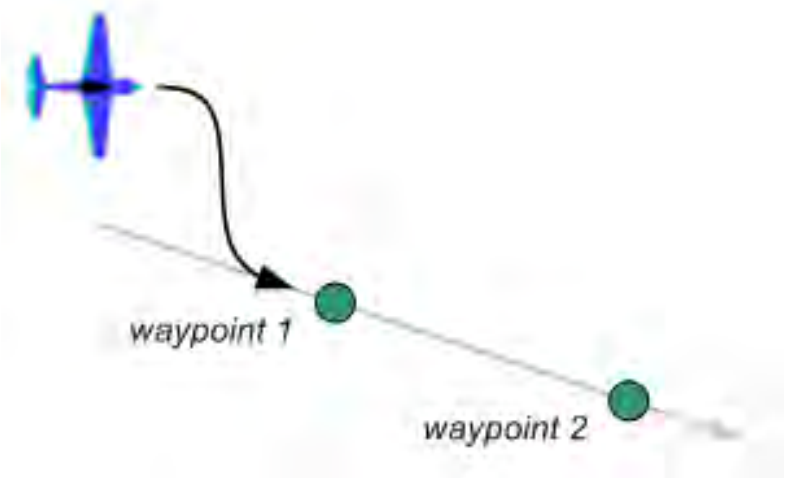


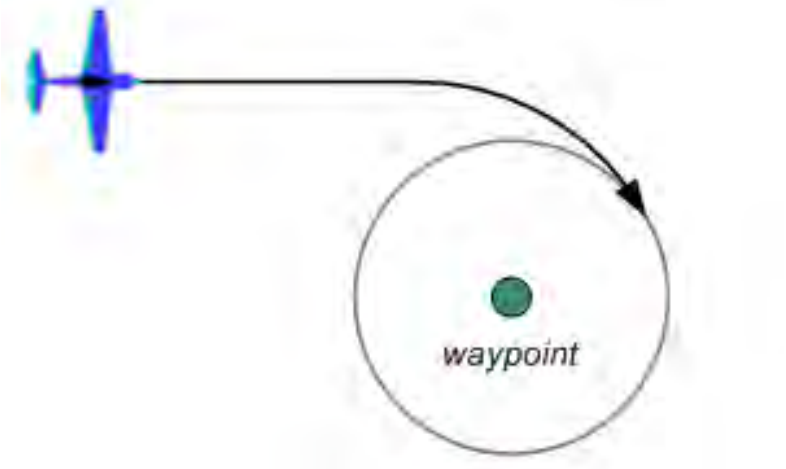
Flight Plan Language

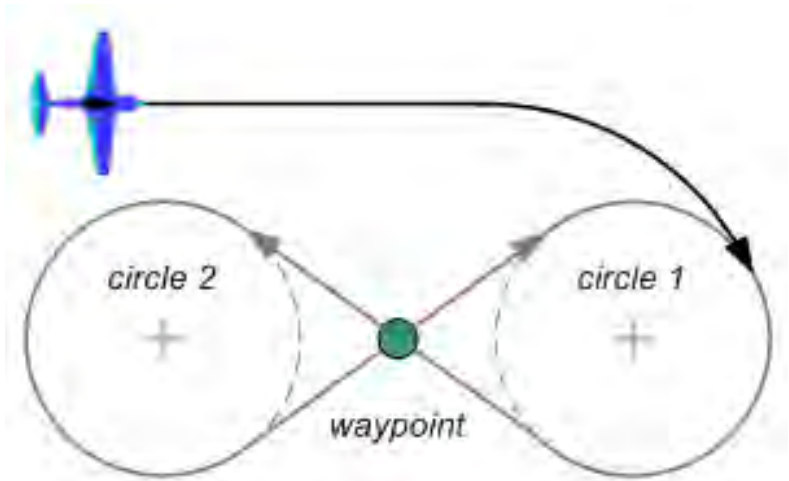


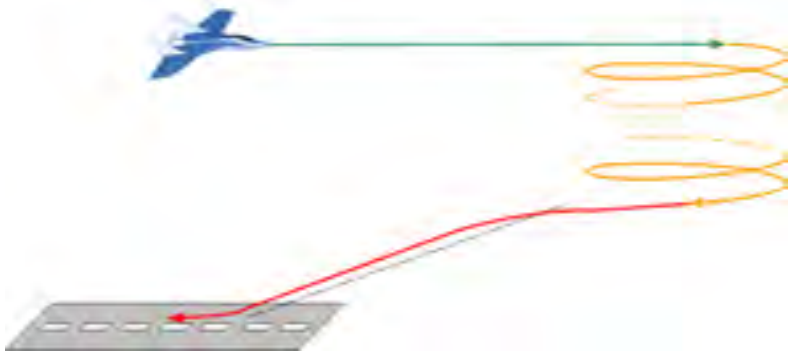


Flight Plan Language

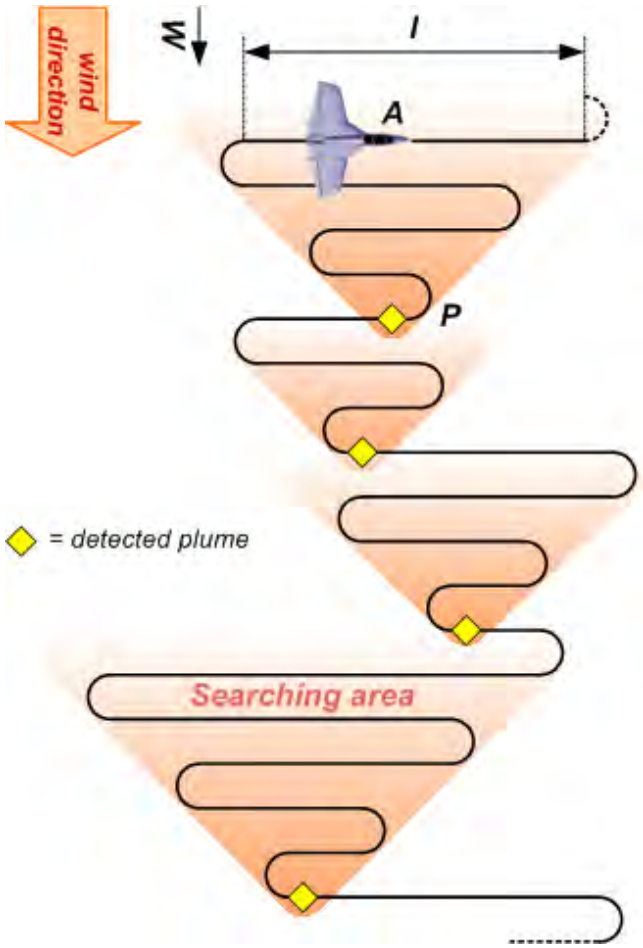








Flight Plan Language



Navigation Language

The more you can program, the less you have to control

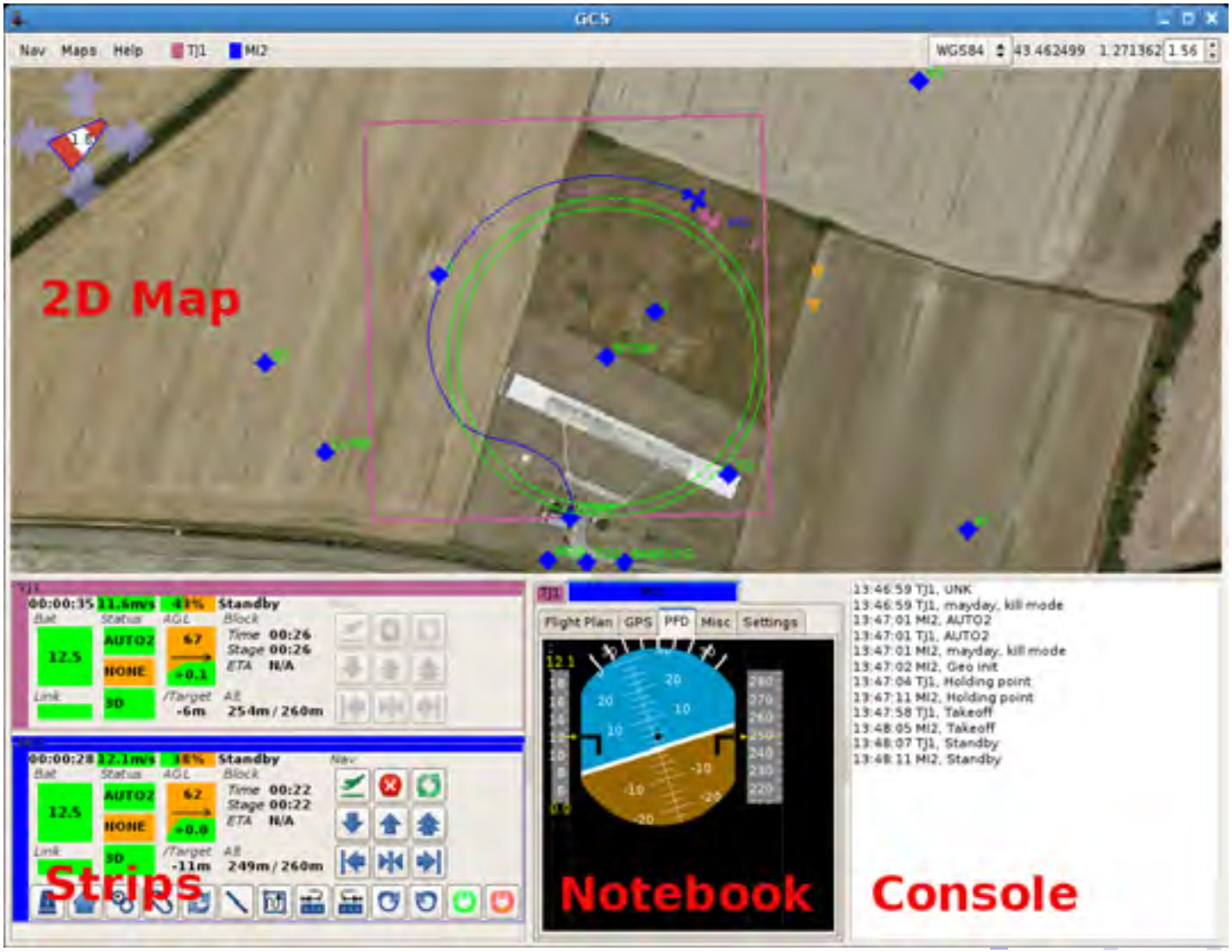
The system provides a way to automate a mission

Control, Statements, Expressions, exceptions...

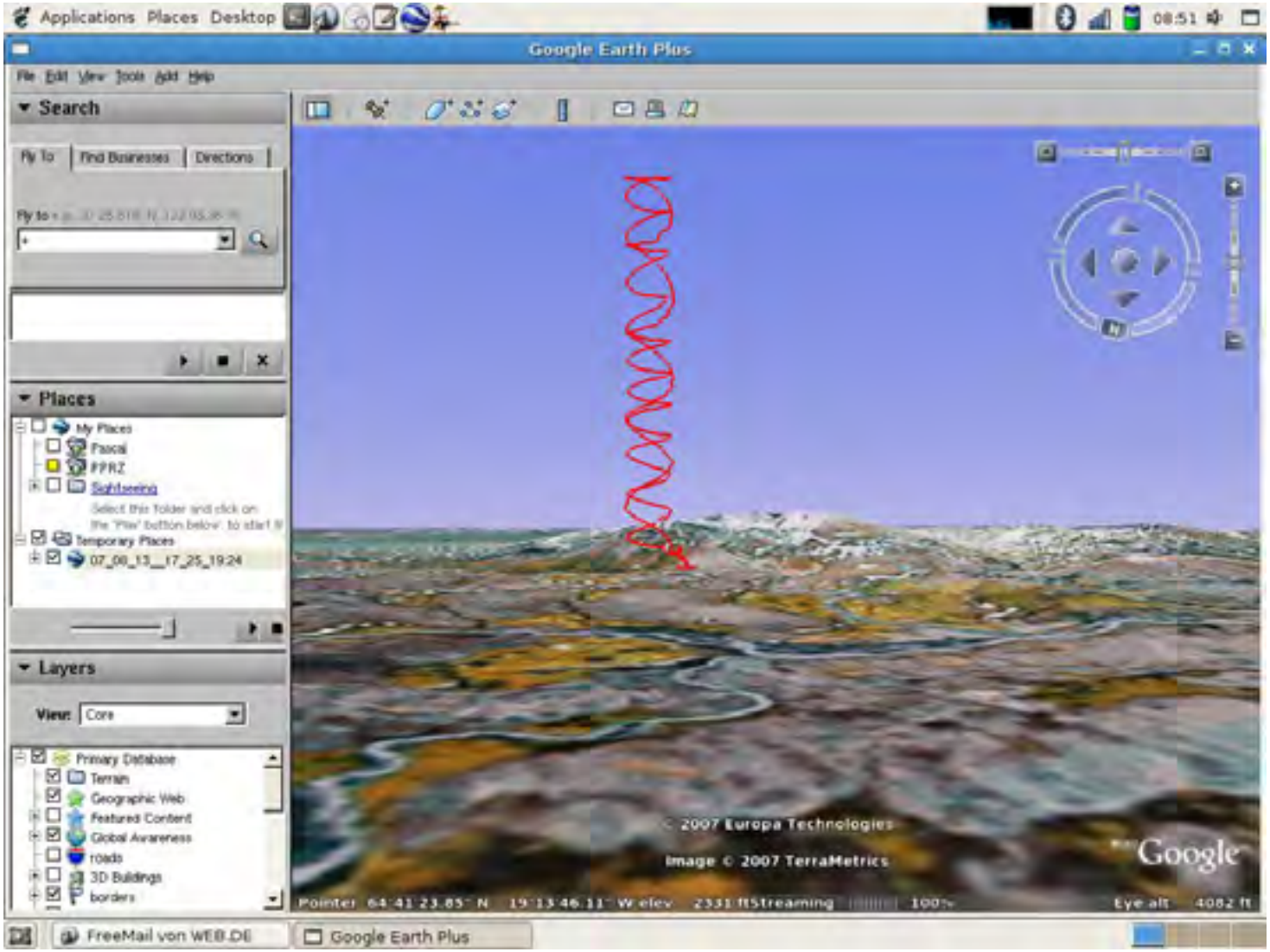
```
<for var = "i" from = "1" to = "5">
  <circle wp = "HOME" radius="75"
          alt = "ground_alt+50*$i"
          until = "stage_time>60" />
</for>

<exception cond="time_since_last_datalink > 22"
           deroute="Standby"/>
```

A Versatile Graphical User Interface

GCS using Google, Microsoft, OpenStreetMap or own maps



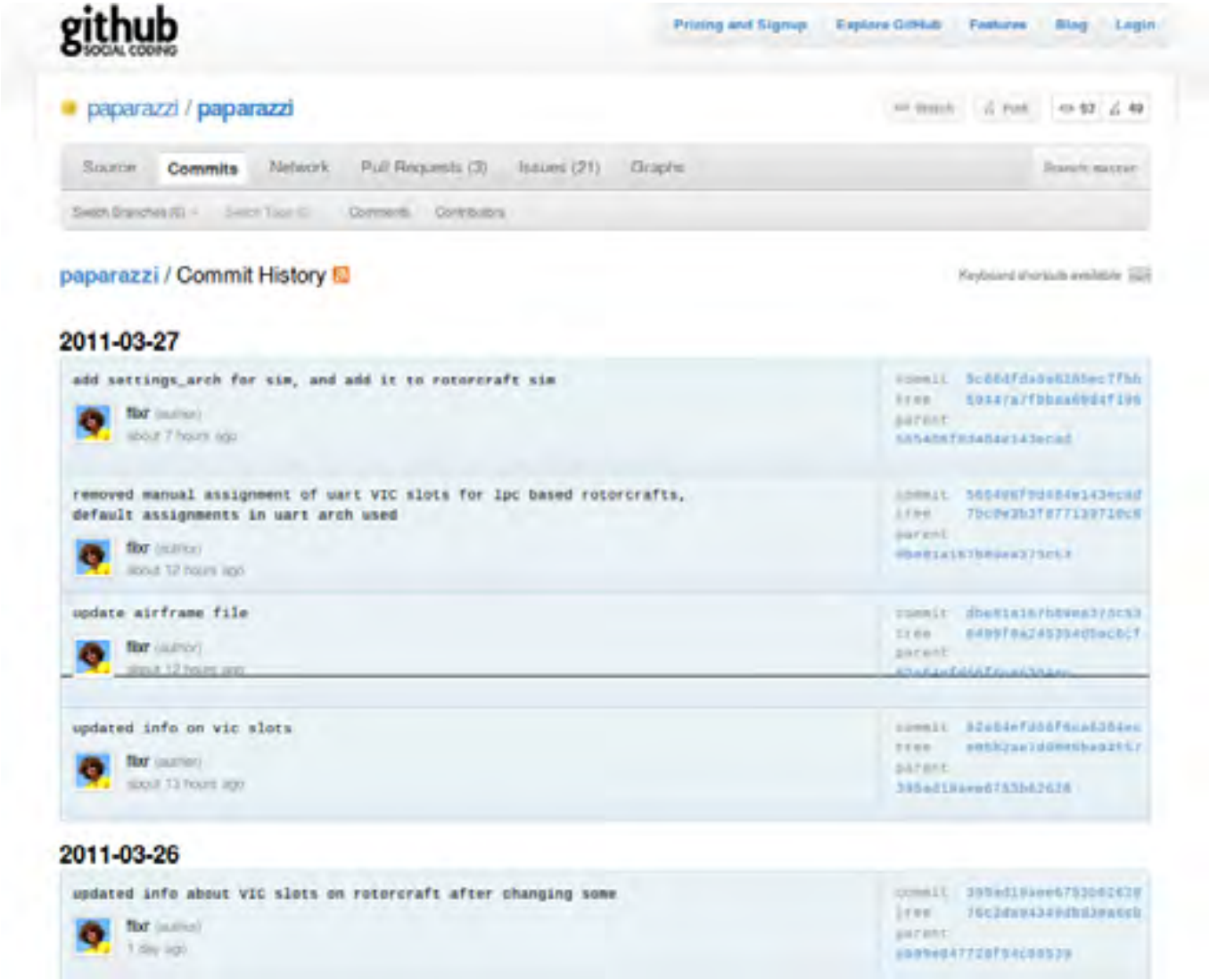
Integrating external Applications



It is Open Source, it's open and free

1. look into the source
2. you can change whatever is needed to suit your requirements
3. add sensors easily
4. other open source autopilots or parts can be included

Software repository



Commit your software early

Why use it ?

1. large user base
2. created/tested/used over a long period of time
3. affordable
4. absolute versability
5. proven to work in harsh environments
6. commercially available

If you use it

1. tell others you do so
2. share code and hardware, let the "open source" community find bugs and improve
3. publish early, adapt to coding styles/architectural layout
4. discuss with others : wiki, mailing list, irc, skype
5. better implement it once good than multiple times bad, don't reinvent the wheel
6. it is not rocket science

Online

http://paparazzi.enac.fr