Using the Paparazzi UAV System for Scientific Research

Gautier Hattenberger*, Murat Bronz†, and Michel Gorraz
ENAC, MAIAA, University of Toulouse, F-31400 Toulouse, France

ABSTRACT

This paper presents an overview of the *Paparazzi* UAV system and its recent use in scientific research. *Paparazzi* is an open-source project that aims at providing a complete solution to fly fixedwing aircraft and rotorcrafts. Several hardware boards and sensors are also developed within the project. Since several years, it has been used by various institutes for scientific research. The recent use on scientific research for meteorological studies is presented as an example.

1 INTRODUCTION

For the last ten years many open-source autopilot systems have been developed by various communities [1]. Some of them are designed for hobby flights while others aim at more professional applications. The *Paparazzi*[1] project [2], started in 2003 by teachers of the French Civil Aviation University (ENAC), is one of the oldest with the goal to provide an affordable and reliable platform for education and academic research. It is now support by other institutes such as MAVLAB of the TU-Delft, individual developers and some private UAV companies from several countries.

This paper propose an overview of the *Paparazzi* system with its ground and airborne software architecture, and some of the recent hardware solutions developed for the project. As an example application, a meteorological study led by ENAC and Meteo-France is presented.

2 OVERVIEW OF THE PAPARAZZI SYSTEM

The global view of the system (see figure 1) is very typical of UAV systems and as not much changed since the beginning of the project:

Ground segment: all the ground software and hardware infrastructures that are used to prepare, monitor and analyze the flight.

Airborne segment: the aircraft, its hardware parts including payload and all the embedded software to control the flight (from stabilization to decision making).

Communication segment: defines all the communication links and protocols between the ground and airborne segments.

Safety link: safety remote control that can be separated from the ground station for short range direct control.

Figure 1: Global view of the *Paparazzi* system

3 FLYING PLATFORMS

The *Paparazzi* system initially designed for robust small fixed-wing aircrafts in 2003, however during the following years several other configurations and concepts have been integrated to the system such as high-aspect ratio gliders, multirotors, and transitioning vehicles[3, 4]. Today, Paparazzi flies on 16cm *Quark*[5] up to 4.3 meter spanned Adler UAV from University of Stuttgart [6] as shown in figure 2. The popular UAV platform *AR.Drone 2* from Parrot[2] (figure 3) can also be used to run the *Paparazzi* software [7].

Figure 2: *Quark* and Adler UAV as two examples of *Paparazzi* flying fixed-wing aircrafts

4 SOFTWARE ARCHITECTURE

The software architecture (see figure 4 is organized on the ground side around several agents connected through the *Ivy*

*gautier.hattenberger@enac.fr
†murat.bronz@enac.fr
[1]http://paparazziuav.org
[2]http://ardrone2.parrot.com/

Figure 3: *AR.Drone 2* with additional GPS and *Asctec Hummingbird* frame as two examples of *Paparazzi* flying rotorcrafts

software bus [8].

One or several UAVs (in blue) are connected to the ground (in brown) with RF-modems or WIFI connections. The *link* agent manages the ground-based radio modem and distributes telemetry data across the network (a single computer, a local network or the internet) where it can be used locally or remotely by the *server*, an agent that logs, distributes, and preprocesses these messages. A set of synthetic messages are then dispatched to other agents, such as the ground control station, or *gcs*, that provides a graphical user interface. Some other tools can be used with the *Ivy* bus such as the *messages* agent that display all the messages, or simulator agents (aircraft, environment) that allow to test the flight plans in various conditions for instance.

Usually, the ground softwares are running on a standard laptop. The project mainly support the Debian/Ubuntu GNU/Linux operating system, but other Linux based distributions and MacOSX systems are possible. In addition, the ground station can run on low cost, low power boards such as the popular RaspeberryPi[3].

4.1 Ground station

The standard ground station interface is one of the oldest part of the system and as not much change during the last years, although some improvements have been done like a vertical view. For historical reasons, the *gcs*, like most of the ground agents, is coded in OCaml and is using GTK as a graphical toolkit.

The *gcs* propose the following features:

- simultaneous flying multi UAV support
- 2D Map capable of displaying Google Satellite, OpenStreetMaps Images and Microsoft Satellite Maps
- real-time movable waypoints
- system status rapid overview with the strip (see figure 6)
- supports rotary and fixed-wing e.g. Airplanes, helicopters, coaxial and quadrotors

[3]http://www.raspberrypi.org/

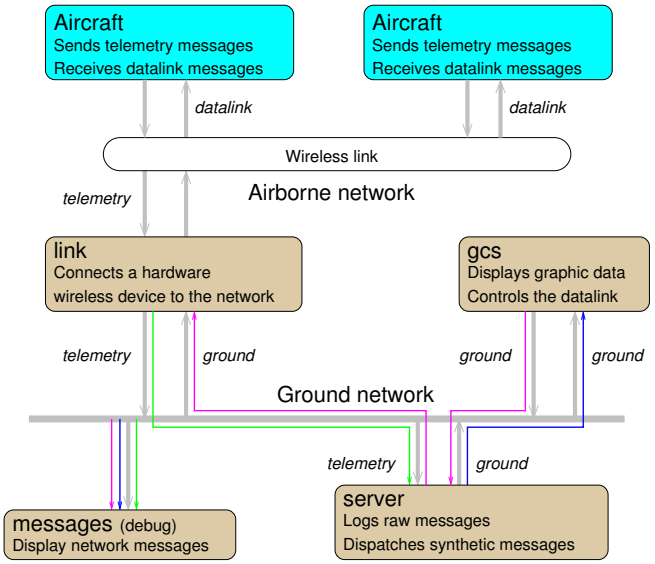

Figure 4: Global view of the *Paparazzi* system

- definable Hotkeys for quick simple in the field control
- voice status output
- full freely configurable GUI layout with various plugins (vertical view, video feedback, text and graphical widgets integrated to the 2D map)

Recently, an Android application (see figure 7) has been developed to be used as a remote *gcs*, with the Android device connected to the main ground station and displaying aircraft position, status and flight plan. Basic high level controls are also possible, and more features will be added in the future to be a viable alternative to the standard interface.

4.2 Airborne code

Based on the old airborne code design, a great effort has been made in order to have a clear separation of the different layers (see figure 8) that would ease the transition to a real-time OS based solution (the current one still use a static scheduling). Especially the data between the sensors and the estimation filters are managed by a publisher/subscriber mechanism inspired by the ground station *Ivy* bus. The resulting estimated state is then published in a blackboard type interface for the control and navigation loops.

Previously, the attitude estimation was based on infrared thermopiles, but is now replaced by IMU-based filters such as complementary filters or kalman filters. Some advance filtering solutions have been tested using the *Paparazzi* system [9].

4.3 Flight plans and mission control

The *Paparazzi* system is using a unique flight plan language, based on *XML*, that allows to:

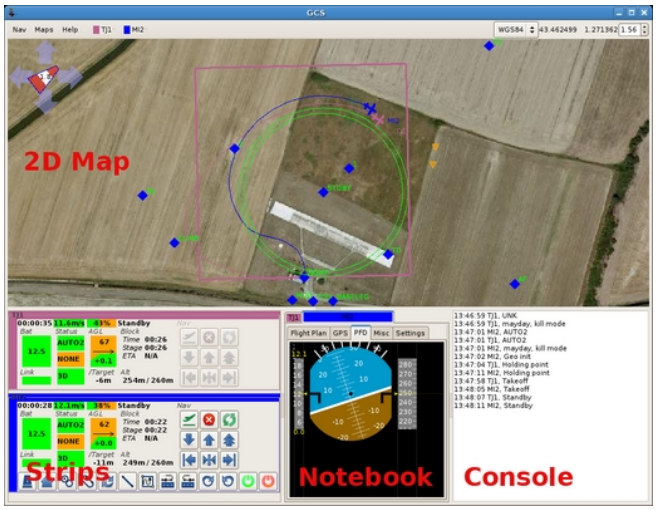

Figure 5: The ground station interface

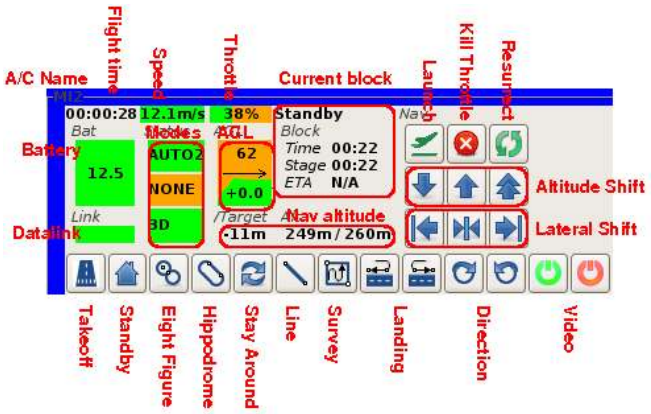

Figure 6: The GCS strip allows a rapid overview of the UAV status

- define a list of waypoints
- define sequences of small tasks (called *blocks*)
- perform *for* and *while* loops
- perform actions based on global or local exceptions
- call a list of predefined navigation patterns (circle, line, survey,...)
- call any custom navigation function written in C

The flight plan description is used to generate a C code compiled with the rest of the autopilot. This static behavior allows to use custom and complex navigation patterns (for instance in photogrammetry applications). The main drawbacks are that since the list of waypoints and blocks is defined before the mission, it is not possible to add new ones at runtime (but the operator can move the waypoints and switch to another block).

Figure 7: The *PPRZonDroid* application running on an Android tablet

In order to overcome these limitations in the case of dynamically generated missions (especially when using a task planning system), a dynamic mission controller has been added. The principle is that only basic tasks (goto, circle, segment) can be used, but they can be loaded in queue list where they will be executed in sequence. A new task can be added at any time (at the end, before existing tasks, or replacing all current tasks) and be received from the datalink system, from an internal module, or from an extra embedded computer.

4.4 Communications

The communication's protocol and the standard set of message used by *Paparazzi* are now reaching their limits. Especially the number of available slots for downlink messages is rather small, while many of this messages could be reorganized in a better way. Some work have been started in this direction. One of the future improvements should also come from an evolution of the current encapsulation protocol shown figure 9 by introducing classes of message or component ID.

An other plan for the future is to support the *Mavlink* protocol[4] or at least a part of it.

4.5 Real-Time OS

Since the beginning of *Paparazzi*, the scheduling of the autopilot tasks is done statically. On the old generation of boards, the computation power exclude the use of too complex real-time operating system (RTOS). The new generation of micro-controllers allows to use such systems. Several Open-Source options are available such as *NuttX* or *FreeRTOS*.

In order to develop embedded logging capabilities, a

[4] http://qgroundcontrol.org/mavlink/start

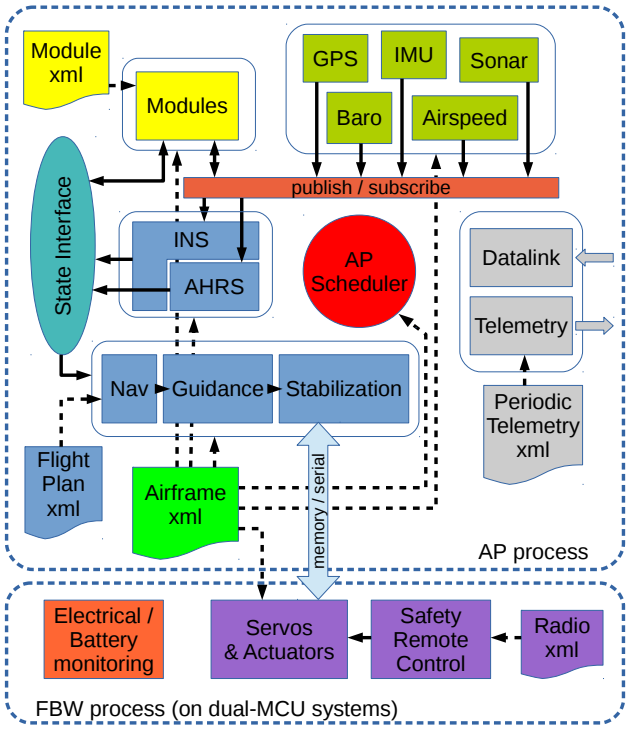

Figure 8: The fixedwing airborne software architecture

RTOS was needed in order to guarantee that the autopilot operations are not delayed when writing on the SD-card. The choice as been made to use *ChibiOS*[5], since it is pretty powerful OS that offers all the necessary components needed for logging.

Currently, the RTOS handles only two tasks, one for logging and one for the rest of the autopilot. For the future developments of *Paparazzi*, it is planned to split the different tasks of the autopilot, allowing accurate timing of the critical tasks (state estimation, control) while others have low priorities (navigation, payload control, logging,...).

On the *AR.Drone 2* platform, the autopilot runs on a linux based OS. It is then possible to easily use multi-threaded processes, and run vision algorithms using the integrated cameras for instance.

5 Hardware solutions

Many controller and sensor boards have been developed for the *Paparazzi* project. The most recent autopilot boards are based on the STM32 micro-controller line from ST-Microelectronics.

The latest autopilot board developed by ENAC is called *Apogee* (see figure 10). It has the same size as the previous generation, with a micro-SD card slot and more powerful micro-controller unit capable of handling the autonomous

[5]http://www.chibios.org

Figure 9: The current standard encapsulation protocol of *Paparazzi* messages used for RF binary communications

navigation of the MAV, a wider range of payload with more inputs/outputs, and high speed data logging (see characteristics in table 1). The design has been done especially thinking of scientific applications requiring data logging for not loosing important data over the telemetry link and without the need of an external board.

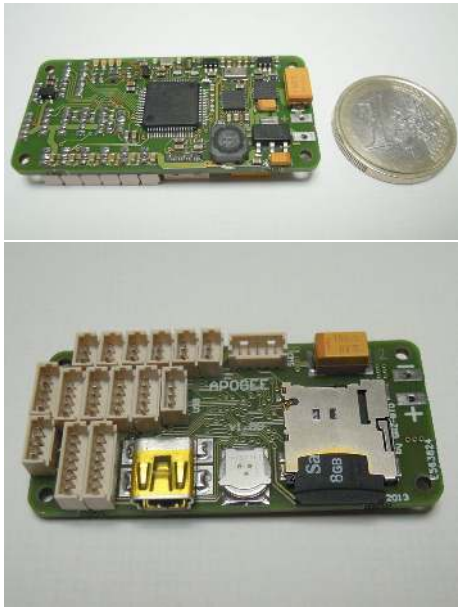

Figure 10: *Apogee* autopilot controller board

Other autopilots have been developed for the *Paparazzi* project with different objectives, such as easier integration on rotorcrafts (*KroozSD* board) or the smallest autopilot *Lisa/S*[6] (see figure 11).

All of these boards are integrating IMU sensors (usually from the MPU line from Invensense), barometer and even GPS U-blox module for the *Lisa-S*.

[6]http://1bitsquared.com/

Processor	STM32F405
IMU	MPU-6050
Barometer	MPL3115A2
Logging	MicroSD slot over SDIO interface
Connectivity	6 PWM, 3 UART, 2 I2C 1 SPI, 1 SWD, 1 USB 4 AUX (PWM, ADC, GPIO)
Remote control	1 PPM input, 1 serial input
Power	1.5A switching power supply 1 5V/500mA power switch
Size x weight	53x25 mm, 10.4 grams, 4 layers PCB

Table 1: *Apogee* autopilot characteristics

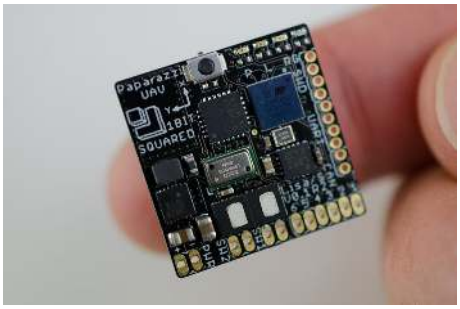

Figure 11: *Lisa/S*, the smallest complete autopilot

6 A Scientific application: meteorological studies

The *Paparazzi* system serves as a complete package for the scientific research missions. Its highly flexible architecture and open-source software helps the researchers and the scientists for their studies. As an example, the *VOLTIGE* (Vecteur d'Observation de la Troposphère pour l'Investigation et la Gestion de l'Environnement) project is presented here.

6.1 Objectives

The main objective of the *VOLTIGE* project is to fly with a fleet of UAVs simultaneously in and out of fog and get temperature, pressure, humidity, solar radiation, cloud and fog detection, and turbulence measurements. The UAVs that are used on the mission are lighter than $2\,kg$ in order to facilitate flight permission and regulation issues. This is a limiting factor for the payload capacity of each vehicle. Different sensors are integrated into different UAVs according to the required flight type. *VOLTIGE* project has mainly three type of flight profiles, turbulence measurements are taken from steady horizontal flight inside the fog at different levels, radiation measurements are taken above the fog which implies the detection of the fog boundary and vertical profile measurements are taken by flying from ground level to the upper boundary of the fog.

Paparazzi System has an automatic collision avoidance system which serves an important role on the low visibility missions such as flying inside the fog with multi-UAVs.

6.2 Platforms Used

Initially, two different platforms are used for the *VOLTIGE* flights which are shown in figure 12. *Funjet* is more suitable for vertical profile measurements as it can climb more efficiently to higher altitudes. *EasyStar* is mostly used for the horizontal flights at constant altitude as it is more stable and has more payload capacity.

Figure 12: The two type of platforms used in *VOLTIGE* project

Several flight campaigns have been performed with these two airframes, and concluded that the *Funjet* airframe is reaching its maximum take-off weight limits with the additional equipments used, which makes it harder to hand-launch and land in small areas. On the other hand *EasyStar* airframe serves well for carrying the required payload with an easy hand-launch and landing characteristics but lack of climbing to the required altitude. As the goal of the project is to fly several UAVs at the same time, both airframes suffer from a short endurance performance (which is limited to 35 minutes) for the complete mission.

Figure 13: *FENIX* UAV designed for *VOLTIGE* project mission requirement

In order to overcome these problems a new airframe is designed particularly for the *VOLTIGE* project mission requirements. A flying-wing configuration, called *FENIX*, with four control-surface is designed to separate elevator and aileron controls. The fuselage is mainly optimized in order to facilitate the integration and operation of the selected payload sensors and electronic cards. The endurance performance has

been increased up to two hours for the level flight and kept over one hour for the climbing flight for the vertical profiles of the mission. This improvement will mainly help for the safety of the multi-UAV operation.

7 Conclusion and future work

The current state of the *Paparazzi* system has been described in the paper. Latest hardware and software capabilities are explained on a recent scientific application which requires robust operation in harsh weather conditions with multiple UAVs flying simultaneously while recording synchronized data for meteorological measurements.

Many developments are still undergoing to make the *Paparazzi* autopilot safer for real world operations or academic research, easier to use for new users while keeping its versatility. The efforts should be focused on the proper support of a real-time operating system, an improved ground control station and a need definition of the communication layer and its messages.

Acknowledgments

The authors would like to thanks all the Paparazzi community for the development, testing and support to this project.

References

[1] Daewon Lee Hyon Lim, Jaemann Park and H.J. Kim. Build your own quadrotor. *IEEE Robotics & Automation Magazine*, September.

[2] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The paparazzi solution. *MAV2006, Sandestin, Florida*, 2006.

[3] Pranay Sinha, Piotr Esden-Tempski, Christopher A Forrette, Jeffrey K Gibboney, and Gregory M Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *Aerospace Conference, 2012 IEEE*, pages 1–17. IEEE, 2012.

[4] C De Wagter, D Dokter, G de Croon, and B Remes. Multi-lifting-device uav autonomous flight at any transition percentage, 2013.

[5] Murat Bronz, Jean-Philippe Condomines, Gautier Hattenberger, et al. Development of an 18cm micro air vehicle: Quark. In *Proceedings of the International Micro Air Vehicle Conference and Flight Competition IMAV 2013*, 2013.

[6] Marc Schwarzbach, Uwe Putze, Ursula Kirchgaessner, and Maria v Schoenermark. Acquisition of high quality remote sensing data using a uav controlled by an open source autopilot. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 595–601. American Society of Mechanical Engineers, 2009.

[7] Bart Remes, Dino Hensen, Freek van Tienen, Christophe De Wagter, Erik van der Horst, and Guido de Croon. Paparazzi: how to make a swarm of parrot ar drones fly autonomously based on gps. In *International Micro Air Vehicle Conference and Competition IMAV2013*, September 2013.

[8] Marcellin Buisson, Alexandre Bustico, Stéphane Chatty, Francois-Régis Colin, Yannick Jestin, Sébastien Maury, Christophe Mertz, and Philippe Truillet. Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs. In *Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine)*, pages 223–226. ACM, 2002.

[9] Jean-Philippe Condomines, Cédric Seren, Gautier Hattenberger, et al. Nonlinear state estimation using an invariant unscented kalman filter. *AIAA Guidance Navigation and Control Conference*, pages 1–15, 2013.