Small Unmanned Aerial Vehicle Avionic-System for Research Purposes

Christopher Ruwisch, Fabian Peddinghaus, Flávio J. Silvestre
Chair of Flight Mechanics, Flight Control and Aeroelasticity
TU Berlin
christopher.ruwisch@tu-berlin.de
peddinghaus@tu-berlin.de
flavio.silvestre@tu-berlin.de

Technische Universität Berlin

Background

The amount of unmanned aerial vehicles (UAVs) on the market has drastically increased over the past decade. Avionic systems for small UAVs are often tailored to specific needs and applications. For research purposes, the hard- and software of the avionic system are commonly in-house developed and can solely be used for a single UAV. Because of dimensional, computational, and precision factors, a new system must be devised every time significant components change. A well-established platform, the PixHawk autopilot, is widely used for research on UAVs due to its versatile application. However, a problem arises if the researcher needs to switch avionic components or has demands for more computational power, higher sensor accuracy, or difficulties regarding interfaces. At the Chair of Flight Mechanics, Flight Control, and Aeroelasticity at TU Berin, a small-scale avionic system for research and educational purposes has been developed. It aims to increase the flexibility in both hard- and software, allowing for more effective development of novel flight control systems.

Hardware Architecture

The system architecture is based on different hardware modules which are connected via CAN to the main computational unit (Raspberry Pi 4B).These modules are:

1. An Attitude Heading Reference System (AHRS) module including a connector for GPS
2. A Radio Actuator Interface (RAI) module
3. A Power Supply Unit (PSU) module
4. An Air Data System (ADS) module (based on an STM32F0)

Each module employs an ESP32 microcontroller with two cores running at 240MHz. It is free-programmable to the desired needs through an open-source toolchain. The ESP32 is used due to an integrated WiFi and Bluetooth chip for wireless communication and a build-in CAN controller for wired communication. This allows experiments like flight tests with wireless communication between the modules. The footprint of the boards is based on the size of the Raspberry Pi 4B, such that all boards can be stacked on top of each other.

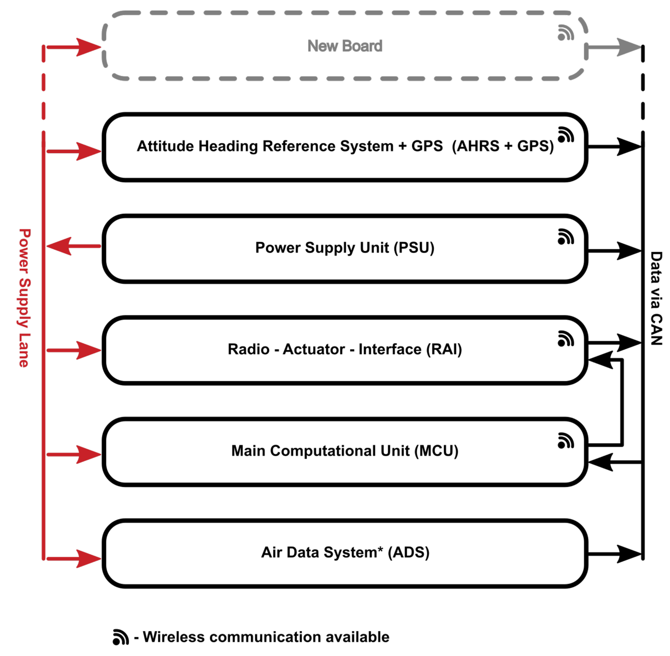
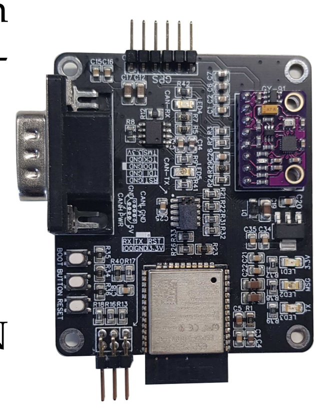


Figure 1: Avionic System (* air data module installed on wing)

AHRS + GPS module

This module is used to attain attitude, velocity and positioning information:

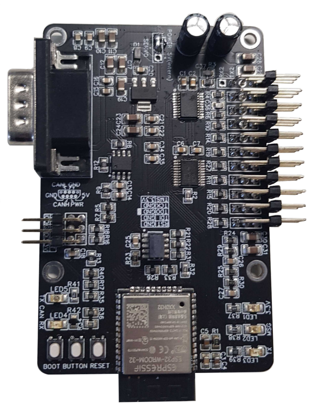
- Sensor: GY-91 (I^2C)
  - MPU 9250
  - BMP 280
- GPS: U-Blox NEO-M8N (UART, NMEA protocol)



RAI module

This module is used as an interface to the aircrafts actuators and the RC-receiver:

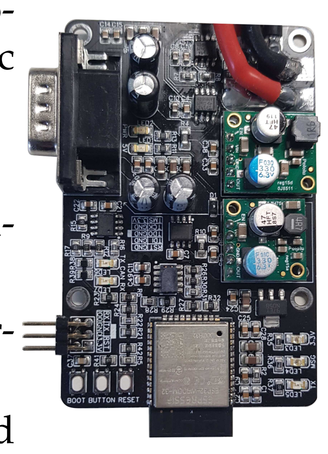
- Reading SUMD-Channel via UART from Graupner receiver
- Up to 10 PWM-Channels (for Servo/Motor control)
- Separated power circuits for ESP32 and actuators
- Software switch between human or autopilot control



PSU module

This module is used to provide power to the avionic system:

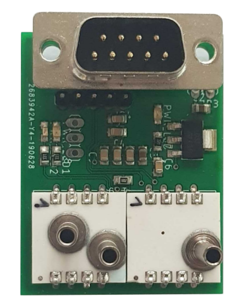
- Up to 10A @ 5V
- Voltage and current measurement
  - Overall voltage and current
  - Actuator voltage and current
  - Sensor/computational voltage and current



ADS module

This modules is used for airspeed and barometric height information:

- AMS 5915 - 10 mbar differential: up to 32 m/s
- AMS 5915 - 700-1200 mbar barometric
- Sensors connected via I^2C
- Small footprint / integrated in the wing



Main computational unit / Software architecture

The Main Computational Unit (MCU) consists of a Raspberry Pi 4B running a modified Raspian kernel with an RT-Preempt patch that enables real-time capabilities. The software architecture of the MCU is based on eProsima's fast Real-Time Publish-Subscribe (FastRTPS) system, which is also used as the primary data distribution service (DDS) in ROS2. FastRTPS is a publisher-subscriber scheme where each process can publish or subscribe to a particular topic in order to send or receive data. This architecture gives high flexibility since each process can access data by subscribing to a specific topic. Furthermore, all processes can be modified and tailored to the researcher's demands.

The basic software architecture consists of 10x concurrently running processes:

- 5x processes as publisher (RAI IN, GPS, AHRS, AIR, PSU)
- 4x processes as publisher and subscriber (SENSORFUSION, CONTROL, DOWNLINK, RAI OUT)
- 1x process as subscriber (LOG)

The red processes are connected directly to the CAN interface, using SocketCAN for easy access to sensor and actuator data. The incoming data is selected via specific message identifiers and preprocessed in a designated process. Afterwards, the data is published under a specific topic, making it available to all subscribing processes. The data flow can be seen in the figure to the right (Figure 2). In RAI OUT, the processed data is sent to the actuator module. The DOWNLINK process is used for system information. A telemetry module is connected via USB to the Raspberry Pi. MAVLink is used as the communication protocol for the telemetry connection, which allows different dialects tailored to the system's specific needs. For data logging, the LOG process subscribes to every active topic and logs the data in binary format to the SD-card of the Raspberry Pi. The log process depends on the topic data, which is published by the system and can be adapted through *IDL*-files that are used for creating a topic. The logger and the decoder of the binary formatted log files are automatically generated based on all used topics, described by the *IDL*-files.

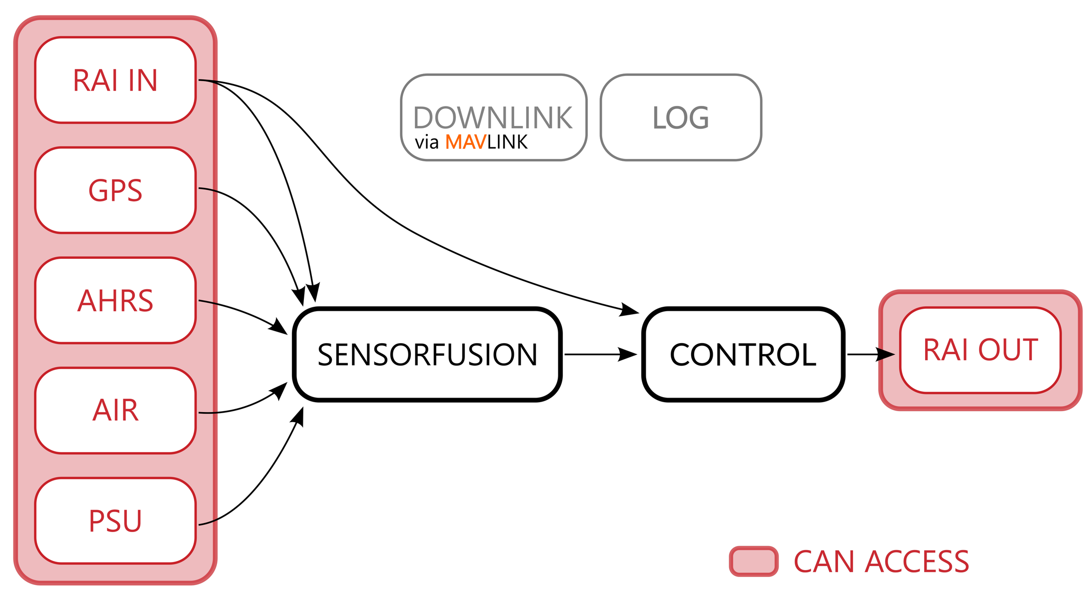


Figure 2: MCU - Process information view

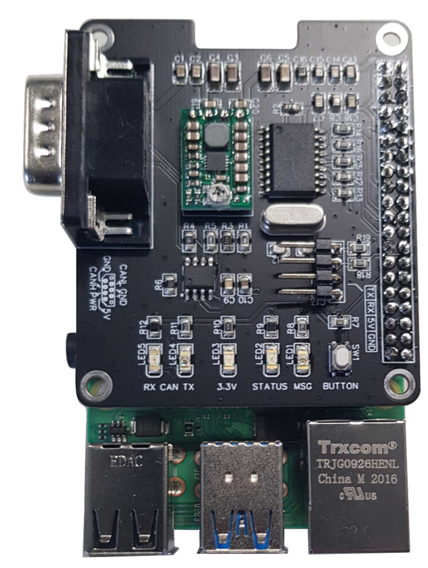


Figure 3: MCU - Raspberry Pi with CAN interface

Conducted flights and outlook

The system has been integrated into a small-scale Cessna model aircraft. Several successful flight tests have been conducted. The data from the flights was both stored on-board and transmitted in flight to a ground station. Further improvements and the development of new modules are part of the planned next steps. With this in-house built avionic-system, a flexible platform for flight tests is available for research purposes and adaptable to task specific needs.

Figure 4: Experimental test-platform during flight