European Rotorcraft Forum

HORUS - HIGH OPERATIONAL RELIABILITY FOR UNMANNED SYSTEMS

Benoit Figuet[1], Prosper Leibundgut[2], Andrea Pedrioli[1], Arno Frauenfelder, Hans Doran[2], Pierluigi Capone[1]

{benoit.figuet, prosper.leibundgut, andrea.pedrioli, hans.doran, pierluigi.capone}@zhaw.ch
arno.frauenfelder@bluewin.ch

ZHAW - Zurich University of Applied Sciences (Switzerland)
[1]ZAV – Centre for Aviation
[2]InES – Institute of Embedded Systems

Abstract

In an effort to increase Unmanned Aerial Vehicles (UAV) mission safety and reliability, this paper introduces the HORUS prototype. HORUS is a small hardware device that allows UAVs to be flown with two Commercial Off The Shelf (COTS) Flight Control Units (FCU) instead of a single one traditionally. The HORUS device offers the possibility to manually switch from one FCU to the other but also to switch autonomously if the UAV gets out of the allowed geographic boundaries or out of its defined flight envelope. Although envelope protection or geofencing features are commonly available on COTS FCUs, they are generally not robust to sensor or FCU failures. But, as a result of its design, which has been developed following manned aviation safety process guidelines, the HORUS device offers high reliability and integrity. In this context, the main objective of this paper is to introduce the HORUS concept, to describe the software and hardware architectures and to demonstrate its functionality by showing some flight test results.

1. INTRODUCTION

The number of Unmanned Aerial Vehicles (UAVs) flying in the sky has increased greatly in the last years and will most probably keep growing due to the new applications / businesses it can support. Even though UAVs appear to be rather harmless and reliable, they could be used to perform operations that involve significant risk that could lead to catastrophic events (collision with human on ground, airplane collision…) resulting in several casualties.

One problem is that today, most of the UAVs flying in the sky have been developed without following a safety process equivalent to manned aviation. While Safety Standards for UAV (open category [3]) are still being defined, the manned aviation ones are too restrictive and costly for most of the UAVs manufacturers. Therefore, most people are flying drones for which reliability number cannot really be guaranteed.

In this context, the objective of the Federal Office of Civil Aviation (FOCA) has funded the design of a system that enhances UAV safety. HORUS (High Operational Reliability for Unmanned Systems) is a device intended for use on unmanned aerial vehicles to increase safety during operations that pose increased risk. Such complex operations are addressed in the Specific Operations Risk Assessment (SORA) regulatory framework [4], which currently is being developed by the Joint Authorities for Rulemaking on Unmanned Systems (JARUS). The SORA assigns various Safety Assurance and Integrity Levels (SAIL) to specific operational scenarios and defines activities that need to be performed. For some operations it is required that the UAS is developed to authority recognized design standards or designed “with system safety and reliability” in mind.

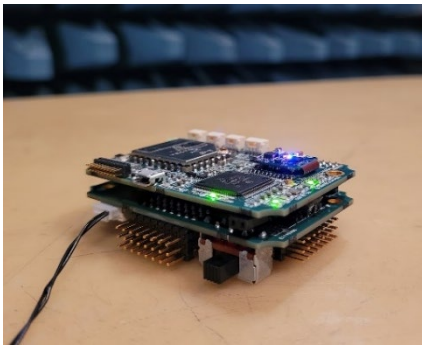

Figure 1 - HORUS prototype

The HORUS device, see Figure 1, intends to increase the operational safety of existing drone flight control systems by adding another layer of safety through supervision of the mission, creating flight control unit (FCU) redundancy and safe automatic flight termination, should the active FCU fail, and the operation get out of control.

While accident data of current widely deployed FCUs such as Pixhawk [2] may show a good statistical record in terms of reliability, they have not been developed according to the above-mentioned standards, which is why it might be

difficult to obtain certification for operations falling within the specific risk category and a higher SAIL.

HORUS equipped UAVs can be flown with any two completely independent autopilots, one of which will act as a hot standby. The HORUS device is loaded with a mission profile and flight envelope limits and conducts checks if the flight is taking place in a defined geographical corridor and within prescribed flight attitude / acceleration limits. A violation of the geographical limits (geofencing) or attitude parameters will initiate the switching to the hot standby FCU, which is given the chance to correct the erroneous flight path or attitude. In case the violations are not resolved, the system will initiate an automatic flight termination sequence, i.e., triggering the emergency rescue system (ERS).

2. HORUS CONCEPT

HORUS adds a safety layer decreasing the risk of serious undesirable events by means of geofencing, envelope protection and FCU redundancy. The high-level concept is described in the flowchart depicted in Figure 2 and is discussed below.

2.1. A Safety Focused Design

While geofencing and envelope protection are widely available features on standard UAV FCU stacks, HORUS novelty lies in the fact that HORUS has been developed following closely a safety process similar to the one applied for manned aviation. Indeed, the HORUS device has been designed with operations in mind that require a high SAIL, which mandates the formerly mentioned requirements to be fulfilled with different levels of robustness. To do so, a requirements-based design has been followed, while satisfying industry standards (RTCA/DO-178C [7], RTCA/DO-254 [5], RTCA/DO-160 [6], SAE-ARP 4754A [9] and SAE-ARP 4761 [8]) as far as it is feasible in the scope of the project. Therefore, it would be possible to operate an UAS in some additional operations, where standard UAS, equipped with only low-cost FCUs, cannot be engaged.

2.2. Flight Control Unit Redundancy

The HORUS device can be connected with up to two, potentially different, off the shelf FCU such as Pixhawk and APM [1] ones. For each FCU, eight control signals can be routed through the HORUS device. It is therefore possible to use HORUS with UAVs having up to eight control surfaces or propellers such as hexacopters for instance.

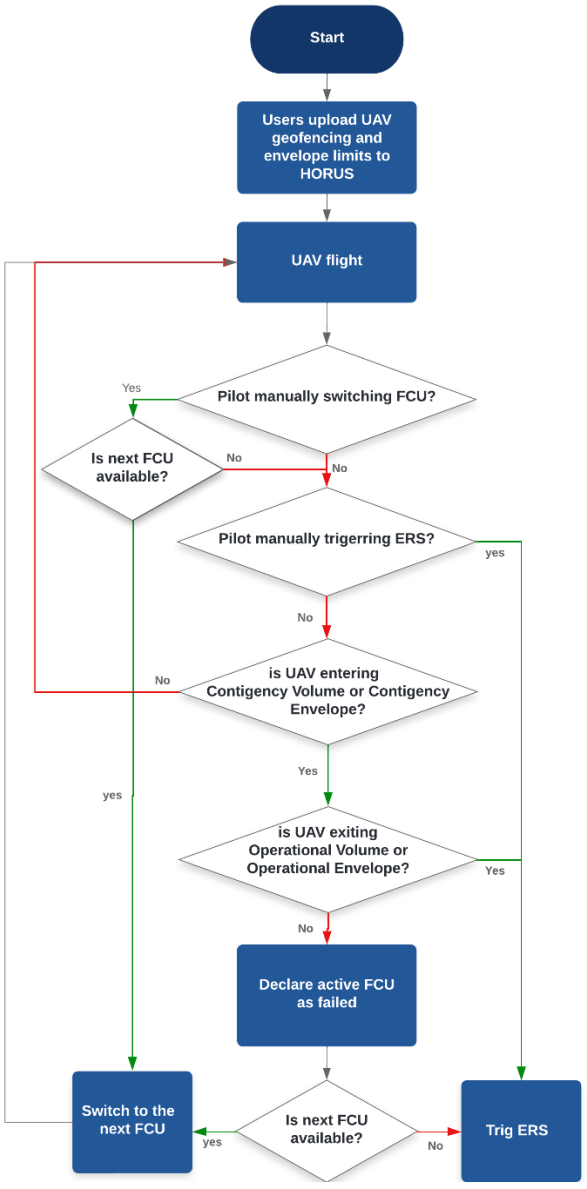

Figure 2 - HORUS high-level flow chart.

2.3. Emergency Rescue System (ERS)

The ERS is the last UAV safety barrier. If switching FCUs does not bring the UAV back into a control state, then the ERS is engaged. Firstly, when the ERS is engaged, the control signals passing through HORUS are brought to electrical ground. This has the effect of stopping the rotation of the propellers. Secondly, two outputs, a logical one and a Pulse-Width Modulation (PWM) one, are set to high, respectively the duty cycle ratio of the PWM signal is changed accordingly. This allows the user to connect a backup system to the HORUS device such as a drag parachute which will be deployed when the ERS is activated.

2.4. Automatic Monitoring

As shown in Figure 2, it is possible to "manually" switch from one FCU to another one or to trigger the ERS using a remote control. But, in an effort to increase autonomy, an automatic switching mechanism has been developed and is introduced below.

2.4.1. Geofencing

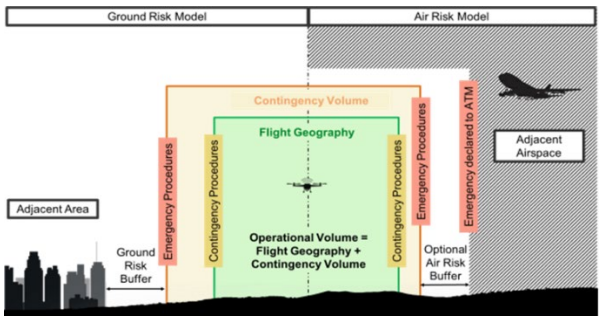

Figure 3 - SORA Semantic Model [4]

HORUS operators can define an "Operational Volume" (OV) representing the "Flight Geography" (FG) plus a "Contingency Volume" (CV) as defined in Figure 3. The OV is represented by a two-dimension polygon of latitude and longitude vertices as well as a maximum altitude. The FG is then obtained by subtracting a certain amount of volume to the OV that represents the CV.

When the different mentioned volumes are defined, HORUS monitors its position by means of triple redundant and dissimilar Global Navigation Satellite System (GNSS) and Inertial Navigation System (INS) sensors: the HORUS tracks at a constant rate whether the UAV is within the FG, the CV or outside of the OV.

Should the UAV go from the FG to the CG and the HORUS device will automatically declare the current FCU as invalid and make the second FCU the active one as shown in Figure 2. In case the second FCU does not manage to keep the UAV within the OV, the flight termination sequence is initiated and the ERS is triggered as described in section 2.3.

2.4.2. Attitude Envelope and Free-Fall Protection

Identically to the geofencing FG, the CV and the OV volumes, defined in section 2.4.1, the Flight Envelope (FE), the Contingency Envelope (CE) and the Operational Envelope (OE) correspond to the different roll and pitch angles allowed ranges that the operator can define for a specific UAV or mission.

The HORUS device monitors the UAV attitude and acceleration using a triple redundant INS sensor suite and verifies at a constant rate whether the UAV is within the FE, the CE or even outside of the OE.

When one of the UAV roll or pitch angle values exceeds the FE range and gets into the CE ones, the HORUS will detect it and then automatically switch to the second attached FCU as explained in section 2.2. The second FCU takes over and becomes the active FCU. In case the failure is not resolved by the second FCU and HORUS gets out of the OE, the flight termination sequence is initiated and the ERS is triggered as described in section 2.3.

When a UAV is hovering, only the gravitational acceleration is read by the accelerometers. Having the UAV free-falling would result in reading an acceleration close to $0m/s^2$. The free-fall protection mechanism relies on two values: a minimum acceleration value α and a time τ as described in [10]. If the UAV acceleration norm falls below the value α for more than τ seconds, then the current FCU is declared as invalid, and the HORUS device switches automatically to the second one. If the acceleration remains below α for another τ seconds, the flight termination sequence is initiated and the ERS is triggered as described in section 2.3.

2.4.3. Sensors Voting

As mentioned in sections 2.4.1 and 2.4.2, the HORUS device is equipped with a triple redundant and dissimilar INS + GNSS navigation suite. Each of the three sensors provides an individual position, attitude, and acceleration estimation.

The validity of each sensors value is then estimated by performing a cross-comparison between each combination of pair of sensors. A sensor is set to invalid when its value exceeds by a certain margin the two other sensor values for a certain amount of time. This architecture makes the HORUS device robust to a single sensor failure.

When the three sensors are detected as valid, the middle value is used for the geofencing / envelope protection monitoring. When one sensor is faulty then the mean value of the two valid ones is used. If more than one sensor is detected as faulty, the HORUS device is not capable of identifying the valid / faulted ones and therefore the ERS is triggered.

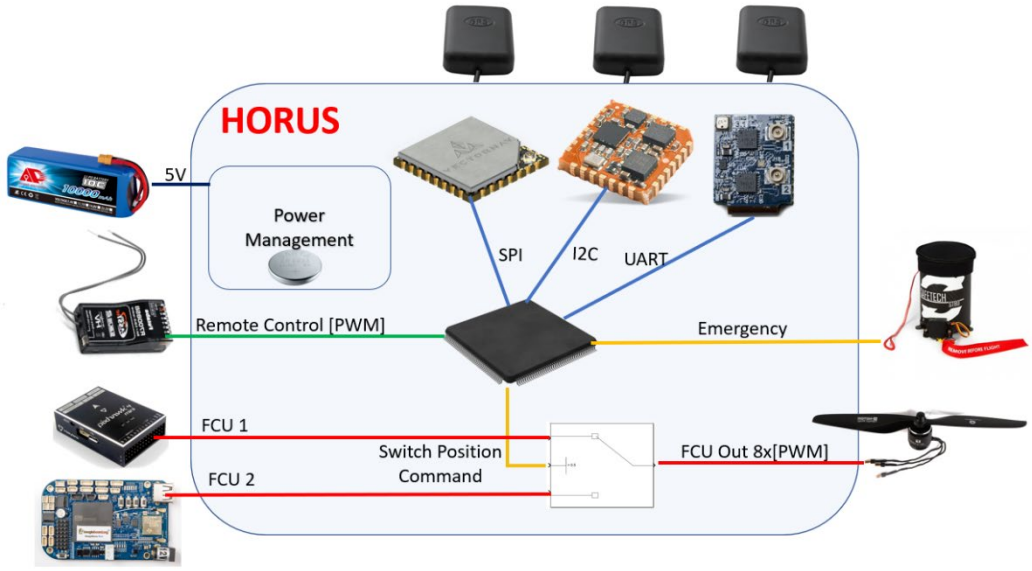

Figure 4 - HORUS high-level hardware architecture

3. HARDWARE ARCHITECTURE

The HORUS device consists of COTS components. The high-level hardware architecture is presented in Figure 4 and discussed below.

3.1. Micro Controller Unit (MCU)

HORUS is equipped with a Texas Instruments TMS570LS1224 MCU. It is an Arm-based MCU designed for safety applications. It is IEC61508 SIL 3 certified. IEC61508 is an international standard issued by the International Electrotechnical Commission and applied in industry dealing with the functional safety of safety-related electrical/electronic/programmable electronic systems. The Safety Integrity Level (SIL) corresponds to a level of reliability expected by the system, or the sub-function concerned, determined by the safety analysis. Corresponding SIL and associated probability levels are presented in Table 1. The MCU has 2 Central Processing Units (CPUs) in lockstep mode which are running the same operations at the same time in parallel. This redundancy allows MCU error detection.

Table 1- IEC61508 SIL and associated failure probabilities.

SIL	PROBABILITY OF DANGEROUS FAILURE PER HOUR
1	$\geq 10^{-6}$ to $< 10^{-5}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
4	$\geq 10^{-9}$ to $< 10^{-8}$

3.2. Sensors

To perform its geofencing and envelope monitoring, HORUS needs to estimate UAV attitude (roll, pitch, and yaw angles) as well as its position and acceleration. Since no reliability numbers were available for COTS Inertial Measurement Units IMU/GNSS, HORUS is equipped with three redundant and dissimilar IMU/GNSS units. They all feature three-axis gyroscopes, accelerometers as well as GNSS receivers. Each of the sensor unit uses a different communication line to exchange data with the MCU (SPI, I^2C and UART). This avoids a single point of failure and therefore increases reliability. HORUS position and attitude are then computed using a voting mechanism which is introduced in section 2.4.3.

3.3. Switches

The HORUS device can drive up to height command signals per FCU. The switching is realized by four metal-oxide semiconductors (CMOS) TMUX1574 switches provided by Texas Instruments. No redundancy was needed for these components since their estimated failure in time (FIT) is in the order of 0.1 failure per billion hours.

4. SOFTWARE ARCHITECTURE

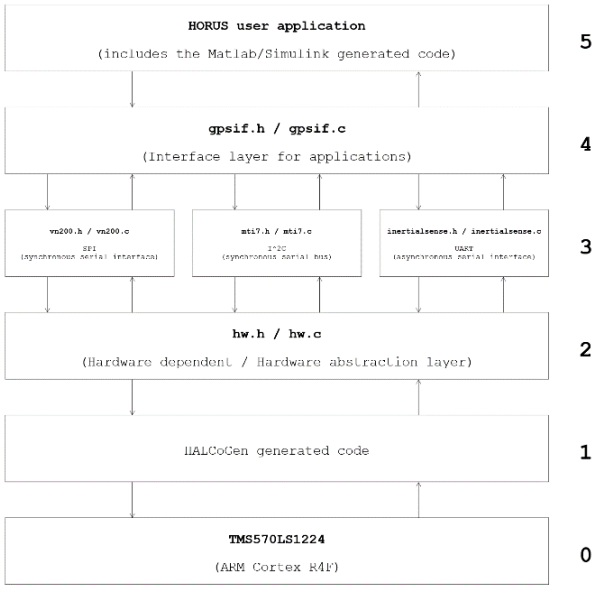

Figure 5 - HORUS layered software architeture.

The HORUS software architecture follows a layered approach as shown in Figure 5. The main asset of this architectural approach is a clear separation of logical software blocks. A layered software architecture approach implies the definition of interfaces between the software layers. Having clearly defined interfaces significantly facilitates the exchange of separate software layers. For example: If one considers switching the microcontroller hardware, layer 0 and layer 1, only the hardware abstraction layer, layer 2, has to be partially rewritten with regard to the interface definitions. The remaining software layers do not have to be altered in any sense.

4.1.1. HORUS user application – layer 5

The system core software which corresponds to the layer 5 in Figure 5, is developed using the Matlab-Simulink DO Qualification Kit. It provides a framework to manage high-level requirements, model and develop low-level ones (Simulink models), test and generate C code.

In Figure 6, the development cycle of the software is shown. Based on a safety assessment, the high-level requirements have been defined. Then, the low-level requirements have been modeled using the Matlab-Simulink software. Once a model is completed, it is thoroughly tested with regards to its corresponding requirement. In a next step, the C code of the Simulink model is generated. The C code is verified with respect to the design and checked for conformance to coding standards. And finally, the code is deployed to the hardware.

It is important to note that there is a bi-directional traceability between each requirement and each Simulink block as well as for each line of c-generated code.

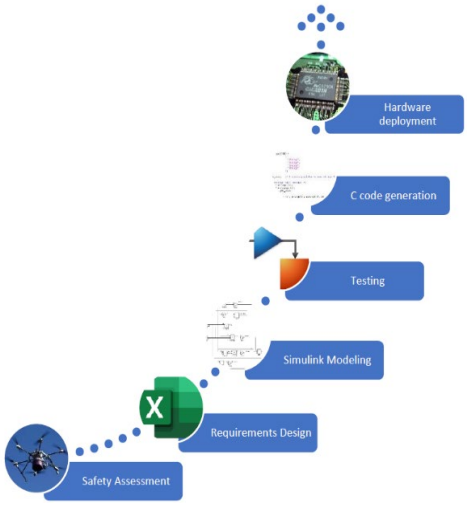

Figure 6 - HORUS user application software development cycle.

4.1.2. Base System – layers 0 to 4

The base system of HORUS, which runs on the microcontroller, consists of the following elements:

4.1.2.1. Scheduler

The currently used microcontroller (TMS570LS1224) is, from a logical point of view, a single core machine (ARM Cortex-R4F). The minimalist scheduler is realized with a hardware timer module, which sets the clock pulse for the rate at which the main program routine is invoked.

4.1.2.2. Main Program Routine

The main program routine is called every time one scheduler tick occurs. The procedure is as follows:

1. Retrieve data from GPS receivers.
2. Write retrieved data to the HORUS Simulink generated program, its input variable memory locations respectively.
3. Invoke the step function of the HORUS Simulink generated program.
4. Gather output values of the HORUS Simulink generated program and invoke measures accordingly.

4.1.2.3. Driver Modules

HORUS relies on three different GPS receivers from three different manufacturers. The hardware on-platform communication is realized intentionally through three different data carrier technologies. The microcontroller is equipped with dedicated

hardware units to cope with these different technologies. The microcontroller communicates through SPI (Serial Peripheral Interface), I^2C (Inter-Integrated Circuit) and UART (Universal Asynchronous Receiver / Transmitter) to exchange data between the CPU of the microcontroller and the GPS receivers.

The application protocols which are required to retrieve data from the GPS receivers are defined by the manufacturers of the GPS receivers. Each application protocol for the according GPS receiver was implemented from scratch according to its manufacturer specification. Although most of the GPS manufacturers provide source code with their own protocol implementation, in the scope of a safe and dependable system they do not conform to the principles of safe embedded systems programming. A practical and frequent example is the use of dynamic memory allocation, which was observed in every manufacturer-provided sources. It should preferably be avoided in safe and dependable program code, so that every memory address (range) is explicitly defined and traceable.

4.1.2.4. Hardware Modules

The hardware modules are a collection of source code which is generated by the manufacturer of the microcontroller that eases the register read/write sequences for the hardware module configuration registers of the microcontroller.

4.1.3. Bootloader

The HORUS bootloader is a program suite consisting of multiple components and program parts. Its main purpose is to provide an interface for the HORUS user to download data from a host (e.g. personal computer) to the microcontroller of the HORUS system. Data that are:

1. Program binaries of the HORUS user application.
2. HORUS mission data geofencing bounds.
3. HORUS UAV flight envelope data.

The HORUS bootloader consists of two main components: The hblc (HORUS bootloader client) and the hbl (HORUS bootloader). The hblc is a script bundle which can be interpreted by a Python3 interpreter on the host PC. The hbl is a program which is persistently stored in a flash memory section of the HORUS microcontroller and starts running as soon as the HORUS system is power cycled.

Through a menu, represented as enumerated list in the console of the hblc, the user is able to select which (download) operation that should be performed. Artefacts (see enumeration above) to be downloaded to the microcontroller are translated to an Intel HEX file structure in order to be transmitted to and stored on the HORUS microcontroller.

5. TESTS AND RESULTS

The HORUS device has been largely tested. Before testing it under real-world conditions, i.e., in a flight test, all HORUS features have been tested and validated standalone. In this section, the most important flight tests are presented.

5.1. Manual FCU Switch

The first flight tests were conducted by switching manually from one FCU to the other with the Remote Controller (RC). In Figure 7, the result is depicted. The dark blue line shows the manual switching duty cycle from the first to the second FCU. During the test, the pilot switched two times from FCU 1 to 2. The middle graph shows that during the test no flags have been raised. Those flags indicate when the UAV goes out of a region: out of flight geography (OFG) and out of operational volume (OOV) highlight a geography protection infraction. While out of flight envelope (OFE) and out of operational envelope (OOE) indicate an envelope protection violation. Then, on the bottom graph, the red line shows the switch position generated by the HORUS, which exactly overlaps the one of the manual switch. Lastly, the

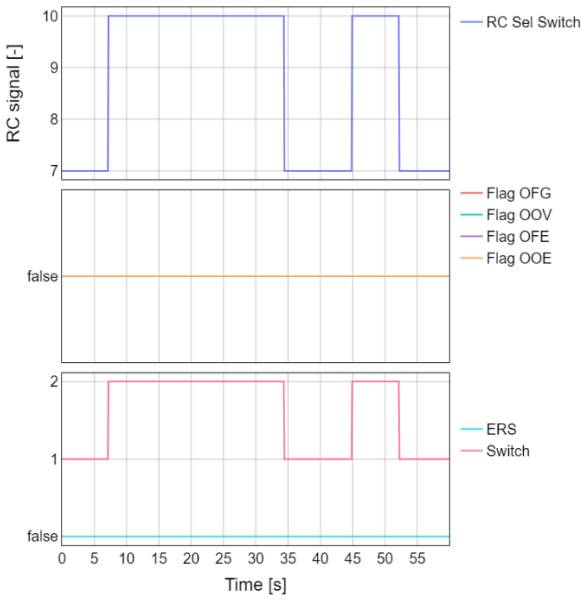

Figure 7 - HORUS test of RC manual switch of SP.

light blue line is the ERS engagement.

5.2. Geofencing Protection

Since ERS engagement would make the propeller stop and let the UAV fall on the ground, it was decided to flight test only the switch position from one FCU to the other. Figure 8 and Figure 9 describe the flight. The dots represent the UAV voted position and their colors describe the HORUS switch position. A green dot corresponds to a switch position 1 which means that the FCU1 is the FCU in command as opposed to a blue dot indicating that the FCU 2 is in command. The orange and red lines are corresponding to the FG and OV volumes respectively.

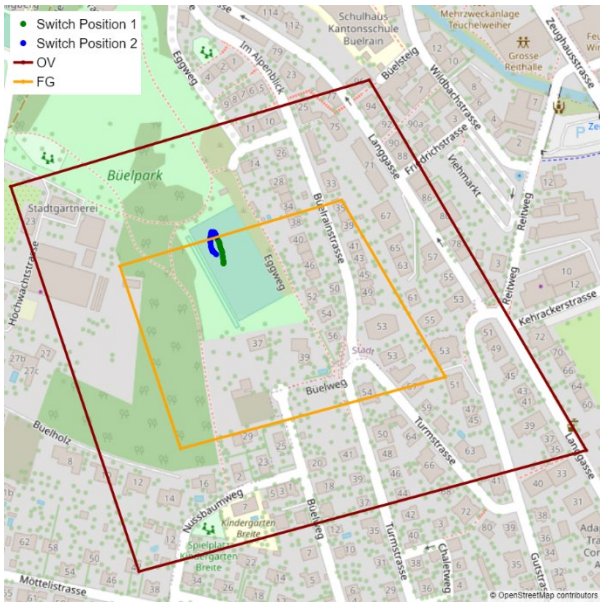

Figure 8 – UAV position during geofencing flight test.

Figure 9 - Zoom on UAV position during geofencing flight test.

The UAV started its flight within the FG volume with FCU 1 being the FCU in command. It was then intentionally flown to exit the FG. As soon as it got out of the FG, the OFG flag was raised, and the SP changed from 1 to 2 meaning that the FCU 2 took over as depicted in Figure 10.

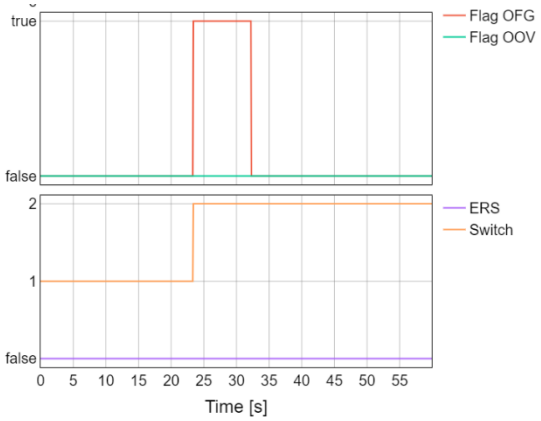

Figure 10 - HORUS signals during geofencing flight test.

5.2.1. Attitude Envelope and Free-Fall Protection

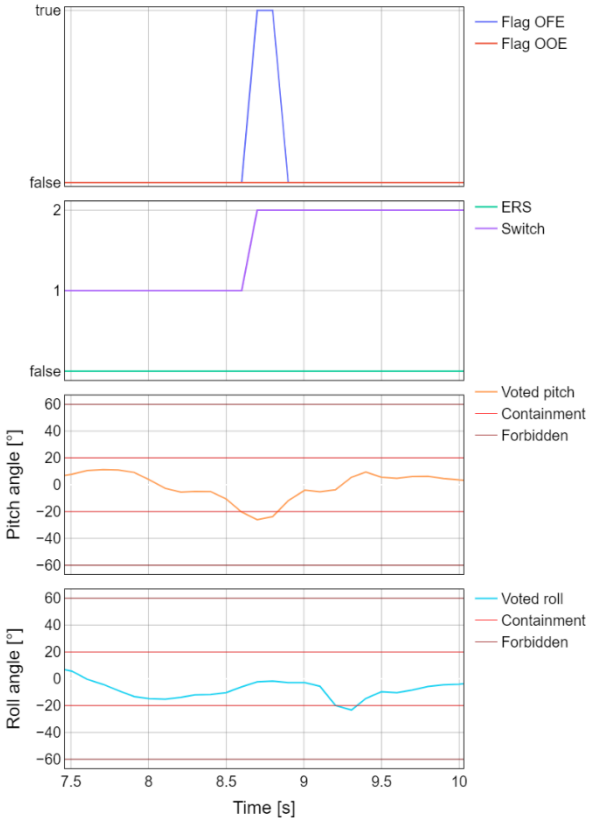

Figure 11- UAV angles and HORUS signals during flight envelope protection test.

The attitude envelope protection test was conducted by setting a FE and OE angles (both for pitch and roll attitude) of $\pm 20°$ and $\pm 60°$, respectively. In fact, the UAV took off and after some small maneuvers, the pilot tried to oscillate back and forward to exceed the FE limits. As it is shown in Figure 11, as soon as the UAV pitch angle exceeds -20°, the HORUS raised the OFE flag and switched to the second FCU. The first plot shows the flag raised during the flight test. Only the OFE flag (blue line) is set to true for the time that the UAV is inside the CE. The second graph shows the ERS (green) and the switch position (violet). The switch position remains at 2 because the CE region has been violated. Lastly, the third and fourth plots describe the pitch and roll angles, respectively as well as the upper/lower FE and OE regions.

The acceleration protection was tested without flying, but rather in a more controlled environment. Two kinds of tests were performed. The first test consisted of letting the HORUS free fall for a couple of meters, long enough to switch from the first to the second FCU and to trigger the ERS. Here the minimum norm and the delta time were set to $\alpha = 2m/s^2$ and $\tau = 0.3s$. The results of the first test can be seen in Figure 12.

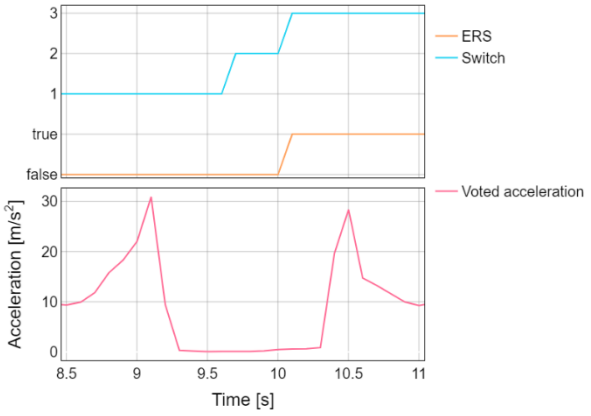

Figure 12 – First free-fall protection test, $\alpha = 0.2m/s^2$ and $\tau = 0.3s$.

The bottom graph shows the HORUS sensed acceleration. At around $9s$, the HORUS device is lifted in the air, for this reason, there is an acceleration peak, followed by a free fall of about $0.9s$. $0.3s$ after the beginning of the free falls, HORUS switches to the second FCU, and after another $0.3s$, the ERS is triggered.

Figure 13, shows the results of a second free-fall test. The difference with the first one is that the time τ is set to $1s$. We can see that the first free fall is long enough to switch from FCU 1 to FCU 2 but not enough to trigger the ERS. But during the second free-fall, at around 13s, the ERS is triggered since the FCU 1 was already declared as invalid.

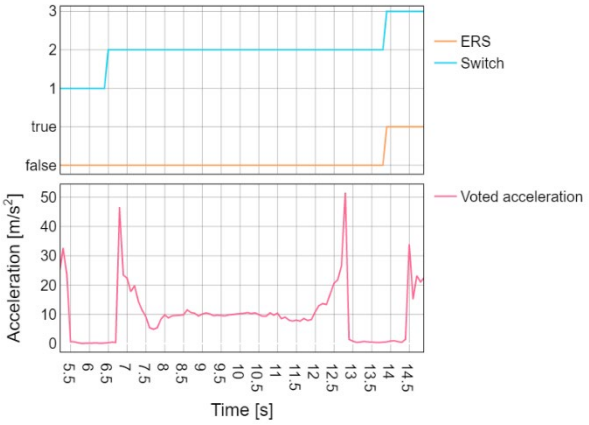

Figure 13 - Second free-fall protection test, $\alpha = 0.2m/s^2$ and $\tau = 1s$.

6. CONCLUSION

Following manned aviation safety process guidelines, the HORUS concept and prototype have been designed. HORUS allows UAV to be flown with two COTS FCUs and can switch from one to the other manually and autonomously if the UAV happens to fly out of its defined geographic boundaries or outside its flight envelope. HORUS also offers the possibility to trigger an ERS such as a parachute independently of the autopilot. Finally, the functionality of the prototype has been demonstrated through a campaign of flight tests. Future work will focus on communicating HORUS state to its attached FCU in order to provide more flexibility to the flight control laws.

7. BIBLIOGRAPHY

[1] ArduPilot. (2022). Von ArduPilot Mega: https://www.ardupilot.co.uk/ abgerufen

[2] Dronecode Foundation. (2022). *Pixhawk Reference Standards*. Von https://pixhawk.org/standards/#autopilot abgerufen

[3] EASA - European Union Aviation Safety Agency. (kein Datum). *Open Category - Civil Drones*. Von https://www.easa.europa.eu/domains/civil-drones/drones-regulatory-framework-background/open-category-civil-drones abgerufen

[4] Murzilli, L. (2020). *JARUS guidelines on specific operations risk assessment (SORA)*. Joint Authorities for Rulemaking

on Unmanned Systems. JARUS. Von http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_v2.0.pdf abgerufen

[5] RTCA. (2000). *DO-254, Design Assurance Guidance For Airborne Electronic Hardware.*

[6] RTCA. (2010). *DO160 - Environmental conditions and test procedures for airborne equipment.* RTCA.

[7] RTCA. (2012). *DO178C - Software considerations in airborne systems and equipment certification.* RTCA.

[8] SAE. (1996). *SAE-ARP 4761 Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment.* SAE International.

[9] SAE. (2010). *ARP4754A - Guidelines for development of civil aircraft and systems.* SAE International.

[10] Vladimir JANOUSEK, Z. J. (2018). *Setting up free-fall recognition with ST's MEMS accelerometers.* STMicroelectronics.