Design, construction, and test of a reliable, redundant on-board computer (OBC) for the OUFTI-1 CubeSat of the University of Liège

Kerem Can*, Damien Teney*, Amandine Denis†, Jonathan Pisane*, Jacques G. Verly*

**Dept. of EECS, University of Liège, Grande Traverse 10, 4000, Liège, Belgium*
Tel: +32 4 366 2642, Fax: +32 4 366 2989, Email: jpisane@ulg.ac.be
†LTAS, University of Liège, Chemin des Chevreuils 3, 4000, Liège, Belgium

We describe the architecture of the on-board computer (OBC) of the educational nanosatellite OUFTI-1 of the University of Liège. The OBC is the brain of the satellite. For redundancy, we quickly converged towards the use of a pair of computer boards. The issues are: What should these computers be? How should they be connected and communicate? Which one should start and continue operation for as long as possible (the "default" OBC)? How should failures be detected? How should control be passed to the "backup" OBC? Should control ever go back to the "default" OBC?

By virtue of the principle of reuse, we decided that at least one of the computer boards should already have flown in space. We selected the FM430 of Pumpkin, the company which provided the frame of our CubeSat. One more reason for considering it was that it uses the low-power MSP430 processor that has already been tested (e.g. against radiations) by several CubeSat teams. Although it would have seemed reasonable to use a second FM430, we quickly realized that many of the functionalities and electronic components were useless for us. We were also beginning to run out of space within the 1 liter of space available. We thus decided that our second computer board should be homemade, starting with the FM430 and removing the unnecessary parts. A key issue was to decide which of these two computer boards to use as the default OBC. Reasoning that our future CubeSats would most probably be built from the homemade version, we decided to use this version as the default OBC.

A critical part of the project is the management of the communication between the two OBCs. We decided to have bidirectional reporting. We also needed to ensure that both OBCs would not be able to take control of the satellite at the same time. Our solution uses the I2C bus and the exchange of an "Alive" signal. This allows us to ensure that the default OBC has priority in the handling of this signal, and that the backup OBC can take control only if the default OBC is out of service.

The default OBC was built and successfully tested. The taking over by the backup OBC in case of failure of the default OBC was successfully tested by using a serial communication line to tell the default OBC to turn off or to stop sending the "Alive" signal.