An Embedded Systems Laboratory for Aerospace Students ⋆

Juan A. de la Puente, * **Alejandro Alonso,** * **Jorge Garrido,** * **Juan Zamorano** *

* *Grupo de Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos (STRAST)*
Information Processing and Telecommunications Center
Universidad Politécnica de Madrid (UPM), Spain

Abstract: The UPM Master in Space Systems (MUSE) is a two-year graduate program focused on space systems technology. The program is largely project-based, with the UPMSat-2 satellite mission as a general framework to which the various course subjects are applied. The course on Data Housekeeping deals with the hardware and software aspects of the on-board computer systems in charge of data acquisition, monitoring and control within a spacecraft, with special reference to UPMSat-2, as well as the command and supervision functions of the associated ground segment. Since the real hardware used on the satellite is expensive and thus not amenable to use in laboratory work, except in very limited situations, a simplified engineering model of it has been developed by the authors. The model is based on a cheap, reduced size microcomputer board kit that can be easily acquired and taken home by the students in order to carry out the laboratory assignments and supplementary work as needed. The hardware and software components of the laboratory kit, its use in the course, and its impact on the students performance, are described in the paper.

Keywords: Embedded computer control systems and applications; On-board computer systems; Real-time systems; Control education; Aerospace computer control.

1. INTRODUCTION

The UPM Master in Space Systems (MUSE) is a two-year graduate program focused on space systems engineering from an industry point of view (Pindado et al., 2016). The program is organized and directed by IDR/UPM [1], with the collaboration of other research groups at the University, including STRAST/UPM [2] for all matters related to computers and software engineering.

The master program has a Project-Based Learning (PBL) orientation. Most of the subjects require students to find solutions to real-life problems in a satellite mission, UPMSat-2 (Pindado et al., 2017, 2018; Roibas et al., 2019). The mission is based on a micro-satellite with a mass of 50 kg, to be launched on a sun-synchronous orbit at 600 km altitude. The satellite is powered by a Li-ion battery fed by solar panels, and has a passive thermal control. The attitude control is based on magnetic sensors and actuators, and uses a novel control algorithm developed at UPM (Cubas et al., 2015; Zamorano et al., 2017). Examples of course subjects related to the satellite subsystems are *Orbital dynamics and attitude control*, *Heat transfer and thermal control*, *Power subsystems*, *Space structures*, *Communications*, and *Data housekeeping*. The latter uses the UPMSat-2 On-board Computer (OBC) as a case study for embedded computer engineering and real-time software development, and is under the responsibility of the authors of this paper.

Most of the students of MUSE have a background in aeronautical engineering, with limited programming skills and no knowledge of embedded system programming. Therefore, laboratory work on an embedded platform is a strong requirement for the course. Following the general approach for the master program, an engineering model of the UPMSat-2 OBC hardware (figure 1) has been used in the past years as a platform for laboratory work, and selected portions of the on-board software (OBSW) have been used to illustrate the main concepts of software development.

In spite of its value for hands-on practical work, this approach has some drawbacks that have hindered student performance: First of all, the OBC engineering model is complex and hard to program. It has a comparatively high cost, thus impeding the acquisition of an adequate number of laboratory workplaces, and the required software development tools are not openly available to the students, which restricts the possibilities of doing personal work at home or elsewhere. For these reasons, laboratory work in the first years of the program was mostly limited to demonstrations by the teachers, with little active participation by the students.

Nevertheless, the current availability of low-cost microcomputer development board opens the way to other so-

⋆ This work has been partially funded by the Spanish National R&D&I plan (project PRECON-I4, TIN2017-86520-C3-2-R).

[1] Instituto Universitario de Microgravedad "Ignacio Da Riva".

[2] Real-time Systems and Architecture of Telematic Services.

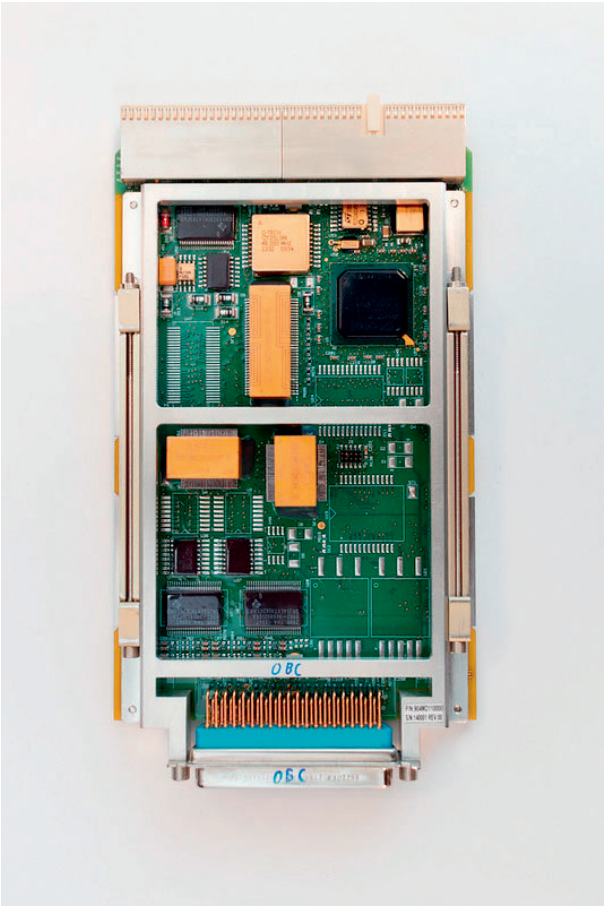

Fig. 1. Engineering model of the UPMSat-2 OBC.

lutions based on the concept of *portable* or *take home* laboratories (Rossiter et al., 2019; Durfee et al., 2004). Under this approach, the students are provided with a low-cost hardware platform that can be programmed with open source tools on their own computers. This makes the laboratory equipment accessible on a flexible schedule basis, and motivates students to develop skills on their own mockup of the satellite computer.

This article presents a take home laboratory designed and built in our group, and its use in the *Data housekeeping* course at UPM. Section 2 provides an overview of the course contents and its academic environment. The next section describes the design of the laboratory platform and the associated programming tools. Section 4 includes a description of the laboratory work that is proposed to the students. A summary assessment can be found in section 5. Finally, section 6 presents the conclusions of the work and some ideas for future developments.

2. ACADEMIC CONTEXT

The course on *Data housekeeping* is scheduled in the second semester of the first year of the MUSE program. The academic program is centred on space technology, and this is the only course focused on computers or software engineering. According to the industrial orientation of the master program, the course is aimed at getting the students acknowledged with the main concepts and problems underlying the design and operation of computer systems embedded in spacecraft, and understanding the complexity of such systems. Given the limited background of most students in computers and programming, the level is necessarily introductory, and no pretence that they will become skilled programmers or computer engineers in the scope of course can be made. Therefore, the objectives are rather to understand the components of on-board computer systems, the main techniques used in developing on-board software, and the importance of keeping quality control of all the elements in the development process. ECSS standards [3] are used to integrate the latter aspects with other courses in the program, and to provide an industrial framework for concepts and techniques. For the same reason, the Ada programming language (ISO 8652), which is extensively used in the European space industry and is the main language of the UPMSat-2 on-board software (Garrido et al., 2015), has been chosen for the software parts of the course.

According to the above considerations, the course syllabus consists of the following topics:

Introduction. Functions and basic structure of on-board computers. Examples of space mission failures caused by software faults.

Computer structure. Basic organisation of a digital computer. Binary representation of information. Processor, memory, input/output systems and common peripheral devices for on board computers.

Programming. Machine code and high-level languages. Compilation toolchain. Embedded systems and cross-compilation. Introduction to the Ada language.

Operating systems. Process and memory management and device handling. Concurrent programming and tasking kernels for embedded systems.

Real-time. Real-time requirements. Task scheduling and real-time kernels. Interfacing with physical devices.

System development and standards. Software life cycle. Verification and validation activities. Relevant space standards.

As above explained, these topics are presented at an introductory level in class lectures. In order to facilitate the understanding of the issues involved by the students, lectures are completed with demonstrations covering features, capabilities, and design decisions of the UPMSat-2 on-board computer. Discussions are used to encourage student participation and promote an active attitude in the classroom.

As a complement to lectures and demonstrations, laboratory work on a course project is required from students. The project is intended to cover most of the topics in the course through a number of assignments on programming a small subset of the satellite functionality on a low-cost embedded system platform. The hardware platform and the associated software development environment are described in the next section.

[3] European Cooperation for Space Standardization. See `https://ecss.nl/standards/` for more information.

3. LABORATORY KIT

3.1 Hardware architecture

The laboratory platform has been designed as a simplified mockup of the UPMSat-2 OBC. In order to make it accessible to the students, the hardware architecture only includes a few hardware subsystems that can be considered essential (figure 2).

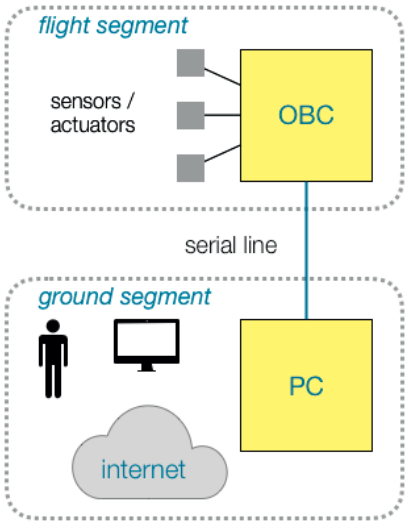

Fig. 2. Architecture of the laboratory platform.

Like the original UPMSat-2, the laboratory system includes both a flight and a ground segment. The work of the students concentrates on the flight segment, which is based on a computer board acting as a mockup of the satellite on-board computer (OBC). The ground segment is based on a PC running a mockup of the mission ground station. Both segments are connected by a radio link, which in our case is simulated by a serial line connection between the OBC board and the student PC (figure 3).

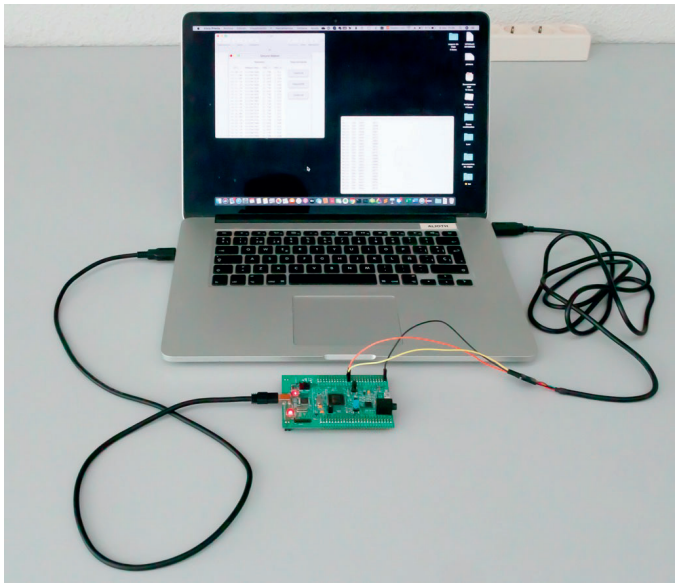

Fig. 3. Embedded laboratory platform.

The student assignments consist mostly in developing embedded software to run on the OBC board. The ground station software, which is a reduced version of the real one used for the mission, has been developed by the teaching team, and need not be modified by the students.

3.2 OBC board

The OBC board is a low-cost microcomputer development board, STM32F4Discovery, with a 32-bit ARM Cortex-M4 microcontroller unit (MCU) including an FPU, 1 MB flash storage, and a variety of input/output devices. This board has been selected for its low cost [4] and the availability of Ada software development tools. The open source GNAT cross-compilation tools for ARM32, which are available for Windows, MacOS, and Linux host platforms, [5] have been chosen for the laboratory. Additionally, the Ada Drivers Library for Cortex-M and Cortex-R, [6] which contains device handlers for all the physical devices included in the board, has been very helpful for hiding the complexities of low-level device interfaces.

The current version of the laboratory kit uses only two sensors, a temperature sensor and a voltage sensor, both located in the MCU chip. This is enough to practice the operation of the data handling functions, and simplifies cabling since no external sensors have to be added.

The MCU also includes four USART interfaces. One of them is used to provide the serial line connection to the ground station. The USART interface is accessible through three GPIO pins on the board. A USB-RS232 header cable is used to provide a virtual serial (COM) port on a USB connector at the PC end. [7]

3.3 Ground station

The ground station mockup is implemented on a laptop computer by means of a software application developed by the teaching staff. Its functionality is a subset of the original UPMSat-2 ground station, replacing the radio link by the serial line connection. The software is written in Ada and provides a simple graphical interface for showing telemetry data received from the OBC and generating telecommands to be sent to it (figure 4). The students can use the software on their own computers to test the embedded software they write for the OBC module.

4. LABORATORY USAGE

4.1 Learning objectives

The main objective of the course is to study the main components of the on-board data handling system in a space system, and to understand the complexity of the hardware and software development processes. In this context, the aim of the laboratory activities is to get first-hand knowledge of the basic aspects of the embedded

[4] The cost of the laboratory kit is below 40 €, about three orders of magnitude less than the real OBC engineering model.
[5] `https://www.adacore.com`
[6] `https://github.com/AdaCore/Ada_Drivers_Library`
[7] See e.g. `https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_USB_RS232_CABLES.pdf`.

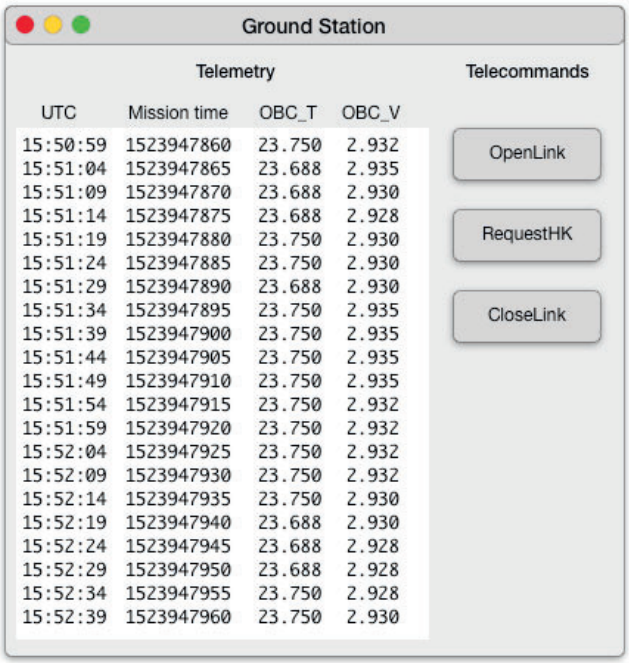

Fig. 4. Screen capture of the ground station application.

software development process in order to be able to assess the main issues involved. At the end of the course, the students are expected to be able to understand, compile and execute simple real-time programs on an embedded platform which is representative of the actual hardware that can be found in space missions.

4.2 General organization

The laboratory kit is made available to the students since the beginning of the course. It is used in the introduction and computer structure modules as a simple example of an embedded computer, with emphasis on the hardware architecture and the differences with a real on-board computer system. Then it is used in the rest of the course as a programming platform on which programming assignments are given to the students, starting with simple programming examples and ending with a fairly complete course project.

Typical assignments are:

(1) Install the cross-compilation tools on the student PC and run an elementary program.
(2) Simple housekeeping program, with a single sensor and no tasks.
(3) Tasking program, with two sensors.
(4) Real-time program, including temporal analysis.
(5) Full On-board data handling (OBDH) system, including communications with ground station and telecommand interpretation.

The last assignment is the final project of the course.

4.3 Programming assignments

The main problem is the limited programming skills of the students, and the complete lack of knowledge of Ada for most of them. Therefore, a gradual approach has to be adopted, in which the program assignments are simplified, using only a basic subset of the language, and the most complex components of the program are provided by the teachers as a basis for student work.

Assignment 1 only requires the OBC board and the USB programming interface to the student PC. The elementary program used in this assignment only uses the LEDs on on the OBC board, with no higher-level human interface. The students are given the source code of the program, and are only required to compile and run it, possibly with some slight changes.

Assignments 2, 3, and 4 include programming exercises of increasing complexity. These exercises use a simple terminal emulator (e.g. PuTTY [8]) on the computer to read the data received from the OBC board through the serial line. Data read from the temperature and voltage sensors on the STM32 MCU are sent to the computer though the serial line in plain ASCII format, so that the values can be directly seen on the terminal application, without the need to use the more complex ground station application. The software modules that are supplied to the students, including some ones that they are requested to further develop, are a subset of those making up the final project.

4.4 Course project

The final assignment makes use of the full configuration, including the ground station software, to develop a complete OBDH (On-Board Data Handling) system.

The software architecture of the OBDH system, shown in figure 5, reflects that of UPMSat-2 on a simpler scale. The names of the modules and other program elements have been kept the same as in the original on-board software as long as possible, in order to make it easier for the students to relate the laboratory work with the real software system.

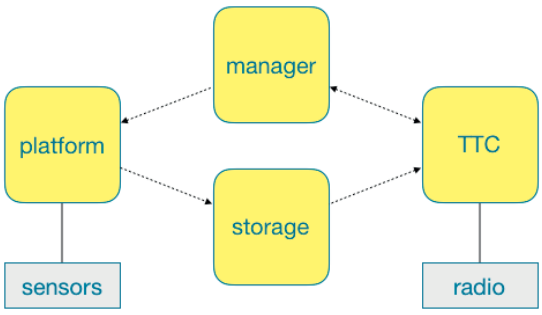

Fig. 5. Software architecture of the OBDH project.

The main components of the OBDH sofware architecture are:

Platform. This module includes the housekeeping task, which periodically reads the platform sensors and stores the values in the storage module.
Storage. This module is a buffer that keeps recently read sensor values to be sent to the ground station as telemetry (TM) messages.

[8] `https://www.putty.org`

TTC. This module implements two-way communications with the ground station computer through the serial line. Its functionality can be decomposed into two main elements:
  - Telemetry (TM). This submodule is in charge of sending telemetry messages of various types, including housekeeping data, from the OBC to the ground station.
  - Telecommand (TC). This submodule receives telecommands from ground, which are decoded and sent to the manager for execution.

  The TTC module uses a simple protocol to communicate with the ground station, which is a reduced version of that used in the UPMSat-2 satellite.

Manager. The function of this module is to execute telecommands and request some actions from other modules when necessary. It is a very much simplified version of the UPMSat-2 manager, whose main function is to supervise the general state of the satellite and carry out changes in the operation mode when needed.

The software provided to the students includes an operational version of all the modules, which can be tested on the ground station. This version includes only basic telemetry and a basic set of telecommands to start communication. The students are required to study the software and implement a new telecommand requesting a complete TM message.

4.5 Communication protocol

Telecommands and telemetry messages are exchanged between the OBC board and the ground station software using a semi-duplex protocol. To this purpose, two operation modes are identified for the TTC subsystem:

- *Idle*. This mode emulates the situation when the satellite is not reachable from the ground station, because it is not visible from the ground antenna. The TTC module sends periodic TM messages with basic housekeeping data, including only the last values read from the sensors. After each basic TM message, the TTC subsystem listens for an OpenLink telecommand sent from the ground station. Upon reception of such a TC, the TTC module signals the Manager to change the mode to *coverage*.
- *Coverage*. When in this mode, the OBC sends all pending TM messages, including errors if any, and listens for additional TCs from ground. The OBC is in this mode for a fixed time, which corresponds to the maximum antenna coverage time in the satellite, after which it switches again to *idle*.

The current version of the software does not provide for simulating communication delays, interferences, or other complexities. Visibility windows can be simulated manually using the *open link* and *close link* telecommands.

For a complete description of the original UPMSat-2 communication protocol see Garrido et al. (2018).

5. EXPERIENCE AND ASSESSMENT

The laboratory kit has been used with all the students in the academic year (2019–20), after some limited experiments in previous years. The feedback from the students has been very positive, and their understanding of the issues related to on-board computer programming has clearly improved with respect to those students who only attended demonstration sessions on the original OBC engineering model in previous years.

Some students have found difficulties in setting up the programming environment. The concepts of cross-compilation and debugging are new to them, and in some cases even the details of the compilation process, including linking and code generation, had to be acquired from the very beginning. Online documentation and class lectures have been expanded in order to cope with these difficulties.

As a last-time consideration, the situation in this academic year (2019–20) has been complicated by the COVID-19 pandemic. The University has been closed for all on-site activities for half the Spring semester, and all teaching has been switched to online activities from mid March. In this context, the fact that all the students had the laboratory kit at home has allowed them to complete the full laboratory program, which has also been supported by online lectures and group discussions. Overall, the experience has been found to be very positive, with favourable feedback from the students, and consequently it is planned to be continued in the following academic years.

6. CONCLUSIONS

An embedded laboratory kit designed specifically for a Master in Space Engineering, including open source programming tools that can be installed on student computers running Windows, MacOS, or Linux, has been described.

The main contribution of this work is the design of a comprehensive sequence of laboratory assignments that enable students to get themselves acquainted with design problems similar to those found in industrial on-board computer systems. The laboratory work ends with a course project involving reduced-scale versions of the components of the UPMSat-2 On-board Data Handling System, which is also used as a reference in previous assignments. Basic software for the work is provided by the teachers, with additional components and modifications to be added by the students.

One major subsystem that has been left out of this development is the UPMSat-2 attitude control system (Zamorano et al., 2017). The main reasons for it are the complexity of the system and the difficulty of performing realistic experiments. The possibility of building a reduced size mockup of the satellite attitude control system, working only on a 2D plane, is being studied at the moment, and may be an interesting extension in the near future.

The laboratory software and the associated documentation is publicly available at `https://github.com/STR-UPM/OBDH_LABS` under GPL and other open source licences.

REFERENCES

Javier Cubas, Assal Farrahi, and Santiago Pindado. Magnetic attitude control for satellites in polar or sun-synchronous orbits. *Journal of Guidance, Control, and Dynamics*, 38(10):1947–1958, aug 2015.

W. Durfee, P. Li, and D. Waletzko. Take-home lab kits for system dynamics and controls courses. In *Proceedings of the American Control Conference*, volume 2, pages 1319–1322, 2004.

Jorge Garrido, Juan Zamorano, Juan A. de la Puente, Alejandro Alonso, and Emilio Salazar. Ada, the programming language of choice for the UPMSat-2 satellite. In *Data Systems in Aerospace — DASIA 2015*. Eurospace, 2015.

Jorge Garrido, Juan Zamorano, Alejandro Alonso, and Juan A. de la Puente. Timing Analysis of the UPMSat-2 Communications Subsystem. *IFAC-PapersOnLine*, 51 (10):217–222, 2018.

ISO 8652. *Ada 2005 Annotated Reference Manual. ISO/IEC 8652:1995(E) with Technical Corrigendum 1 and Amendment 1*, 2007.

Santiago Pindado, Ángel Sanz, Sebastián Franchini, Isabel Pérez-Grande, Gustavo Alonso, Félix Sorribes-Palmer, Javier Cubas, Andrés García, Elena Roibás, and Antonio Fernández. MUSE (Master in Space Systems), an advanced master's degree in space engineering. In *1st Annual International Conference on Engineering Education & Teaching*, 2016. ATINER'S Conference Paper Series, No: ENGEDU2016-1953.

Santiago Pindado, Elena Roibás-Millán, Javier Cubas, Andrés García, Angel Sanz, Sebastián Franchini, María Isabel Pérez-Grande, Gustavo Alonso, Javier Pérez-Álvarez, Felix Sorribes, Antonio Fernandez-López, Mikel Ogueta-Gutierrez, Ignacio Torralbo, Juan Zamorano, Juan Antonio de la Puente, Alejandro Alonso, and Jorge Garrido. The UPMSat-2 satellite: an academic project within aerospace engineering education. In *2nd Annual International Conference on Engineering Education & Teaching*, 2017. URL `https://www.atiner.gr/engedu`.

Santiago Pindado, Javier Cubas, Elena Roibás-Millán, and Félix Sorribes-Palmer. Project-based learning applied to spacecraft power systems: a long-term engineering and educational program at UPM University. *CEAS Space Journal*, 10(3), 2018. doi: 10.1007/s12567-018-0200-1.

Elena Roibas, Jose Miguel Alvarez, Santiago Pindado, Javier Pérez-Álvarez, and Angel Sanz. UPMSat-2 communications system design , integration and testing , within MUSE (Master in Space System) academic plan. In *4th Annual International Conference on Engineering Education & Teaching*, June 2019.

J. Rossiter, S. Pope, B. Jones, and J. Hedengren. Evaluation and demonstration of take home laboratory kit. *IFAC-PapersOnLine*, 52(9):56–61, 2019. doi: 10.1016/j.ifacol.2019.08.124.

Juan Zamorano, Jorge Garrido, Javier Cubas, Alejandro Alonso, and Juan A. de la Puente. The design and implementation of the UPMSAT-2 Attitude Control System. *IFAC-PapersOnLine*, 50(1):11245 – 11250, 2017. 20th IFAC World Congress.