Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

The PLATO Instrument Control Unit: Design, Implementation, and Scientific Performance

Rosario Cosentino

rosario.cosentino@inaf.it

Osservatorio Astrofisico di Catania

Mauro Focardi

Arcetri Astrophysical Observatory

Anna Maria Di Giorgio

Institute for Space Astrophysics and Planetology

Andrea Russi

Institute for Space Astrophysics and Planetology

Giovanni Giusi

Institute for Space Astrophysics and Planetology

Simone Chiarucci

Arcetri Astrophysical Observatory

Carlo Del Vecchio Blanco

Kayser (Italy)

Luca Serafini

Kayser (Italy)

Gunter Laky

Space Research Institute

Dominik Loidolt

University of Vienna

Harald Ottacher

Space Research Institute

Harald Jeszenszky

Space Research Institute

Roland Ottensamer

University of Vienna

Devitt Dini

Kayser (Italy)

Giuseppe Giglio

Kayser (Italy)

Armin Luntzer

University of Vienna
Alessio Pannocchia
Kayser (Italy)
Marina Vela Nuñez
Arcetri Astrophysical Observatory
Marco Verna
Kayser (Italy)
Maria Farina
Institute for Space Astrophysics and Planetology
Scige John Liu
Institute for Space Astrophysics and Planetology
Luca Pilato
Kayser (Italy)
Marco Passerai
Kayser (Italy)
Natalia Auricchio
Osservatorio di Astrofisica e Scienza dello Spazio
Fulvio De Persio
Agenzia Spaziale Italiana
Elisabetta Tommasi
Agenzia Spaziale Italiana
Daniele Brienza
Agenzia Spaziale Italiana

Research Article

Keywords:

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11316502/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

The PLATO Instrument Control Unit: Design, Implementation, and Scientific Performance

Rosario Cosentino*[1,8], Mauro Focardi[2], Anna Maria Di Giorgio[3], Andrea Russi[3], Giovanni Giusi[3], Simone Chiarucci[2], Carlo Del Vecchio Blanco[4], Luca Serafini[4], Gunter Laky[5], Dominik Loidolt[6], Harald Ottacher[5], Harald Jeszenszky[5], Roland Ottensamer[6], Devitt Dini[4], Giuseppe Giglio[4], Armin Luntzer[6], Alessio Pannocchia[4], Marina Vela Nunez[2], Marco Verna[4], Maria Farina[3], Scige John Liu[3], Luca Pilato[4], Marco Passerai[4], Natalia Auricchio[4], Fulvio De Persio[9], Elisabetta Tommasi[9], Daniele Brienza[9]

[1]Fundación Galileo Galilei – INAF, La Palma, Spain, [2]INAF, Osservatorio Astrofisico di Arcetri, Firenze, Italy, [3]INAF, Istituto di Astrofisica e Planetologia Spaziali, Roma, Italy, [4]Kayser Space, Livorno, Italy, [5]Institute of Space Research (IWF), Austrian Academy of Sciences, Graz, Austria, [6]Department of Astrophysics, University of Vienna, Vienna, Austria, [7]INAF, Osservatorio di Astrofisica e Scienza dello Spazio di Bologna, Bologna, Italy, [8]INAF, Osservatorio Astrofisico di Catania, Catania, Italy, [9] Agenzia Spaziale Italiana (ASI), Roma, Italy

Abstract

The PLAnetary Transits and Oscillations of stars (PLATO) mission, selected as ESA's third medium-class mission of the Cosmic Vision programme 2015-2025, is designed to detect and characterize exoplanets and exoplanetary systems through long-duration, high-precision photometric observations. At the core of the PLATO payload data processing system lies the Instrument Control Unit (ICU), which provides the interface between the spacecraft and a highly distributed network of processing units managing cameras.

This paper presents a comprehensive overview of the PLATO ICU, focusing on its architecture, design evolution, software implementation, qualification and validation through extensive testing campaigns. The ICU performs a central role in payload operations, including commanding and control, telemetry handling, time distribution, and science data aggregation and compression. Its design is based on a fully redundant architecture combining a LEON 3 SPARC V8 processor, a SpaceWire-based communication network, and a software and hardware-assisted data compression system.

The evolution of the ICU design across ESA review milestones is presented, together with the consolidation of both software and hardware components. Special attention is given to the Unit's Application Software (ASW), which implements ICU and payload control together with data-handling functionalities. The results of functional and performance tests carried out on Engineering and Qualification Models until the delivery of the Flight Model are presented, demonstrating the robustness and compliance of the system with mission requirements. These results confirm the readiness of the ICU for its physical integration and implementation within the PLATO payload.

* rosario.cosentino@inaf.it

Introduction

PLATO, as a survey mission, represents a major step forward in the detection of exoplanets and exploration of exoplanetary systems, with a primary focus on the detection and characterization of terrestrial planets orbiting bright, nearby stars. By means of long-duration, high-precision photometry enabling both transit detection and asteroseismic analysis, PLATO will allow the determination of planetary radii and ages and, through complementary observations from ground, planetary masses with unprecedented precision, thereby addressing fundamental questions on the formation and evolution of planetary systems in a statistical context.

Achieving these objectives requires not only an optimised photometric accuracy and wide field of view, but also a highly efficient and reliable data processing system capable of managing a large amount of data, preserving the scientific integrity of the measurements. In particular, the ability to detect shallow transit signals and perform high-precision photometric measurements imposes stringent constraints on data handling, timing accuracy, and system stability.

In this context, the Instrument Control Unit plays a key role within the payload data processing chain, extending its capabilities beyond those related to traditional payload control systems, in order to include data aggregation, synchronization, and combined SW and HW compression. It must ensure reliable and deterministic control of the payload while sustaining the high data throughput generated by the multi-camera system without compromising performance or data integrity. This design minimizes the risk of introducing additional noise or systematic effects during on-board processing, thereby preserving the photometric precision required at camera-system level to detect Earth-sized planets orbiting a wide range of host star types in the habitable zone, down to Solar-like targets.

1 The PLATO Payload and Data Processing Concept

The PLATO payload is based on a distributed architecture in which multiple subsystems operate in parallel to acquire and process scientific data. The instrument includes twenty-four normal cameras dedicated to the observation of fainter targets, complemented by two fast cameras optimized for brighter stars. This combination allows PLATO to cover a wide dynamic range in stellar brightness while maintaining high photometric precision required to detect rocky planets and Earth twins.

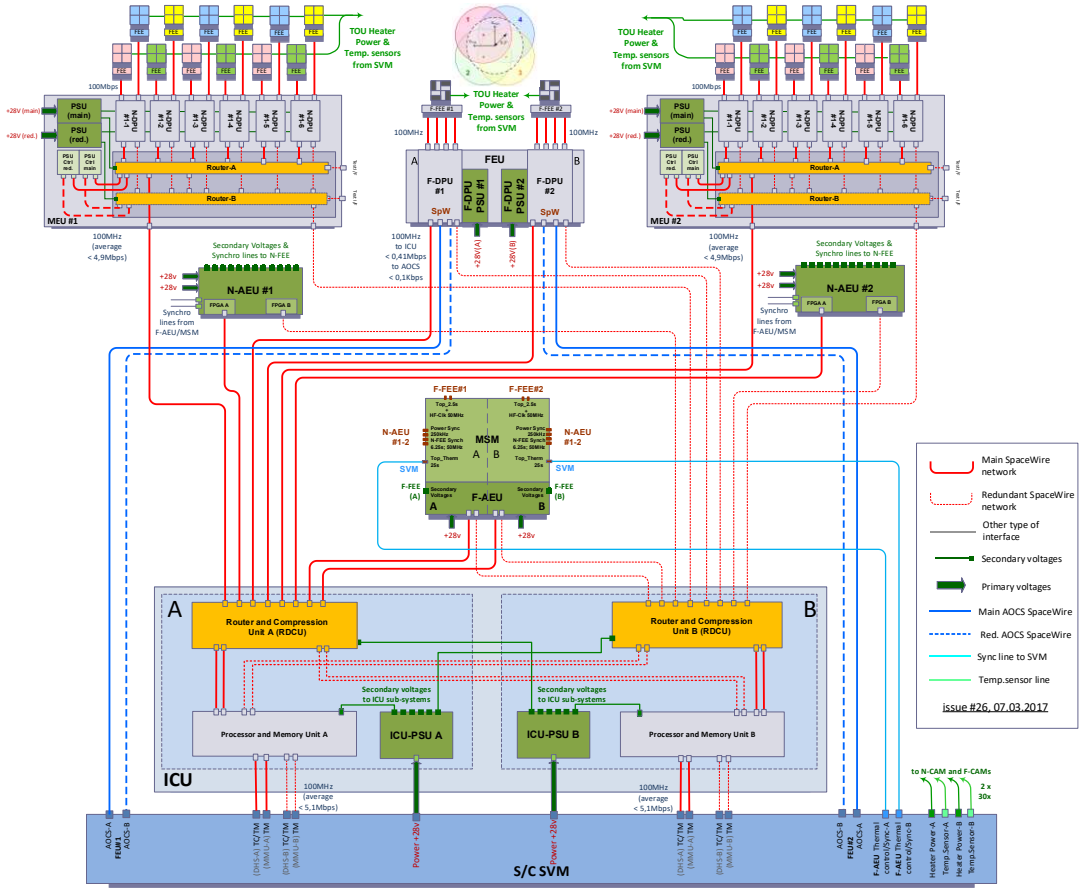


Figure 1 - Overview of the PLATO payload architecture. The instrument is based on a multi-camera concept including 24 normal cameras and 2 fast cameras, managed by a distributed Data Processing System (DPS). Scientific data are pre-processed at Data Processing Unit (DPU) level and then transmitted to the Instrument Control Unit (ICU), which performs centralized data handling and manages interfaces to the spacecraft (**Image courtesy of DLR**).

Each camera is associated with its own focal plane array and front-end electronics, forming an autonomous acquisition unit. The raw data produced at the detector level are first processed locally by dedicated Data Processing Units (DPUs) organized in groups of six and structured into two higher-level units, referred to as Main Electronics Units (MEUs). The DPU units perform initial operations such as image calibration, targets selection and cropping, background subtraction, and extraction of photometric data, including light curves, “imagettes” (from image cropping around selected targets) and centroids.

Once pre-processing has been completed, the resulting science products are transmitted to the ICU through a SpaceWire-based network. The ICU is then responsible for collecting and merging these data streams, applying further processing steps where necessary, and ultimately compressing and packetizing the data for transmission to the spacecraft.

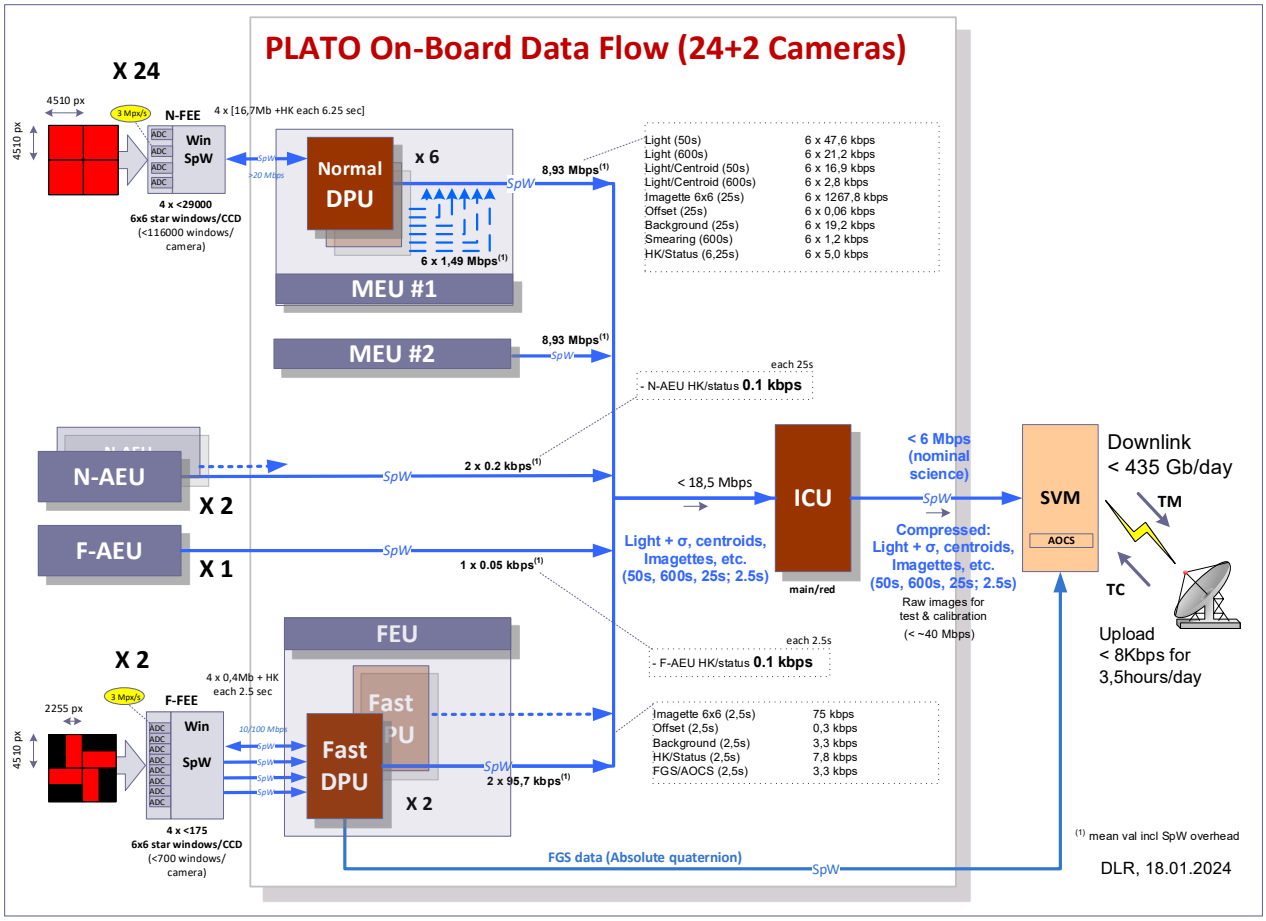


Figure 2 - Simplified representation of the PLATO DPS data flows. Data acquired by the cameras are processed by Normal and Fast DPUs and transmitted to the ICU through a SpaceWire network. The ICU collects them and performs compression and packetization before transmission to the spacecraft (**Image courtesy of DLR**).

The data acquisition rates are defined by the different operational modes of the cameras. Fast cameras operate at a cadence of a few seconds (2.5 s), while normal cameras provide data at longer intervals (25 s). This results in a heterogeneous and continuous data stream that must be handled efficiently by the ICU.

The overall architecture, shown in Figure 1, reflects a structured data flow and processing tasks distribution, where local processing is performed at the DPU level, and global coordination and data consolidation are centralized in the ICU. This approach allows the system to scale efficiently while maintaining a high degree of flexibility.

2 The Role of the Instrument Control Unit

The ICU is the central element of the PLATO payload data processing system and is responsible for coordinating the activities of all subsystems. Its functions extend across multiple domains, encompassing command and control, data handling, synchronization, and system monitoring.

From an operational perspective, the ICU acts as the entry point for all telecommands originating from the spacecraft, as a PUS (Packet Utilisation Standard) terminal. These commands are decoded, validated, and

either executed locally or forwarded to the appropriate subsystem. This requires the ICU to maintain a consistent and up-to-date representation of the payload configuration and operational state.

In parallel, the ICU continuously collects housekeeping data from all units within the payload. These data are used to monitor the health of the system and to detect potential anomalies. In case of deviations from the expected behaviour, the ICU is in charge of producing events and triggering actions, thereby contributing to the overall robustness of the instrument as a whole.

Another key responsibility of the ICU is the management of scientific data. After receiving pre-processed data from the DPUs, the ICU performs additional operations, including formatting, data compression, and packetization. Data compression represents a critical function of the ICU, enabling the reduction of the high data volume generated by the multi-camera system to fit within the spacecraft's downlinking telemetry constraints. This is achieved through a combination of hardware-assisted and software-based compression techniques, designed to optimize compression efficiency while maximising the compression ratio and preserving the scientific integrity of the measurements.

Time synchronization is also a critical aspect of the ICU functionality. The unit receives timing information from the spacecraft and distributes it to all subsystems, ensuring coherent operations across the entire payload. This is particularly important given the distributed nature of the instrument and the need for precise science data timestamping.

3 Data Products and End-to-End Scientific Data Handling

The scientific data products generated by the PLATO payload include imagettes, light curves, centroids, and auxiliary parameters related to background and instrumental effects. These products are derived from the initial processing performed at DPU level and are subsequently handled by the ICU for transmission to Ground.

Within the ICU, the data handling chain is designed to preserve the structure and content of these products while adapting them to telemetry constraints. Data received from the DPUs are first buffered in memory and organized into internal structures that reflect the logical grouping of the scientific information. This step is essential to ensure that data belonging to the same observation window remain coherent.

Following buffering, the data are processed by the compression system, which operates either in hardware or software depending on the data type and operational mode. The compression is applied at the level of defined data entities, allowing efficient reduction of data volume while preserving the internal organization of the scientific products.

Once compression is completed, the data are packetized according to the PUS telemetry format and transmitted to the spacecraft. The packetization scheme is designed to allow complete reconstruction of the original data on ground by exploiting a set of metadata, ensuring that no information is lost or ambiguously encoded.

Particular attention has been devoted to the handling of edge cases, such as partial data availability, missing packets, or timing inconsistencies. The ICU software includes mechanisms to detect and report such occurrences, enabling appropriate mitigation strategies during ground processing.

The overall data handling strategy reflects a strong coupling between system design and scientific requirements. By maintaining the integrity and traceability of the data throughout the processing chain, the ICU ensures that the final science products are suitable for high-precision photometry.

The end-to-end data handling chain implemented in the ICU is illustrated in Figure 3

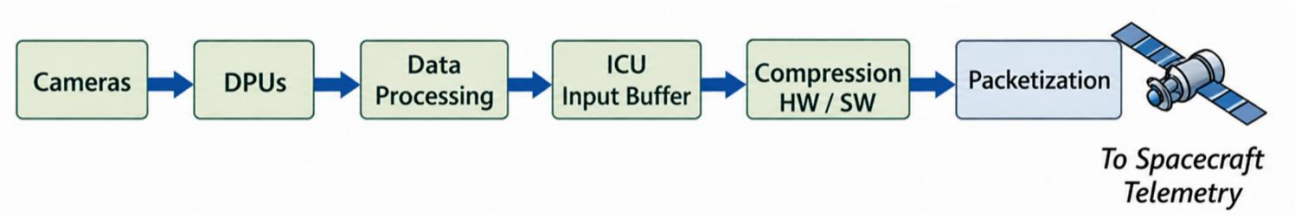


Figure 3 - End-to-end data handling chain of PLATO. Scientific data received from the DPUs are buffered, organized, processed, and compressed before being packetized for transmission to the spacecraft. The compression is lossless, ensuring preservation of the scientific information.

4 Hardware Architecture

The hardware architecture of the ICU has been designed to meet stringent requirements in terms of reliability, performance, and fault tolerance. At its core, the system is based on a LEON 3 (UT700) SPARC V8 processor running at 133MHz, which provides the computational capabilities required to handle both control and data processing tasks.

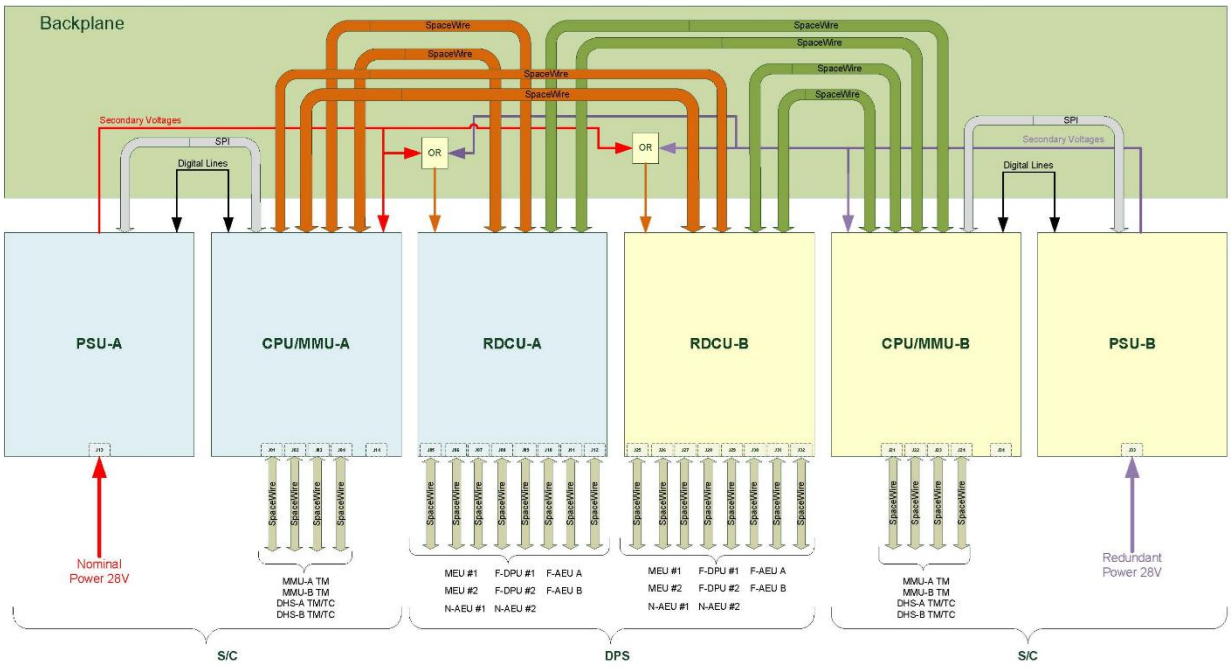


Figure 4 - Functional block diagram of the PLATO Instrument Control Unit. The ICU integrates a processor module, memory subsystem, SpaceWire interfaces, and the Router and Data Compression Unit (RDCU). The architecture is fully redundant, with two cold-redundant processing chains.

The memory subsystem is organized to support different operational needs. A non-volatile memory is used to store software images and configuration data, while a large SDRAM provides the working space for program execution and data buffering. Error detection and correction mechanisms are implemented to ensure data integrity in the presence of radiation-induced faults. In addition, a PROM memory is included in order to ensure a reliable bootstrap of the unit through the boot software. The presence of all these different memory types, managed by the processor, makes the ICU memory bus demanding in terms of timing design and optimization.

Communication within the payload is based on a SpaceWire network configured to operate at up to 100 Mbps, which provides high-speed, low-latency data transfer capabilities between the ICU and the distributed payload units. The ICU interfaces with multiple subsystems through dedicated links supporting both command and data transmission, within a network topology that includes nominal and redundant communication paths, as illustrated in Figure 1. In order to cope with the number of SpW interfaces, an FPGA connected with the main processor has been added in order to expand the processor interface capability with additional SpW links and memory buffer capability for efficient data management.

A distinctive feature of the ICU architecture is the inclusion of the Router and Data Compression Unit board (RDCU), along with a CPU and MMU (Mass-Memory Unit) board and a PSU (Power Supply Unit). The RDCU integrates a SpaceWire router along with a hardware-based data compression engine implemented in an FPGA. By offloading compression tasks to dedicated hardware, the system is able to achieve high throughput while reducing the computational load on the main processor.

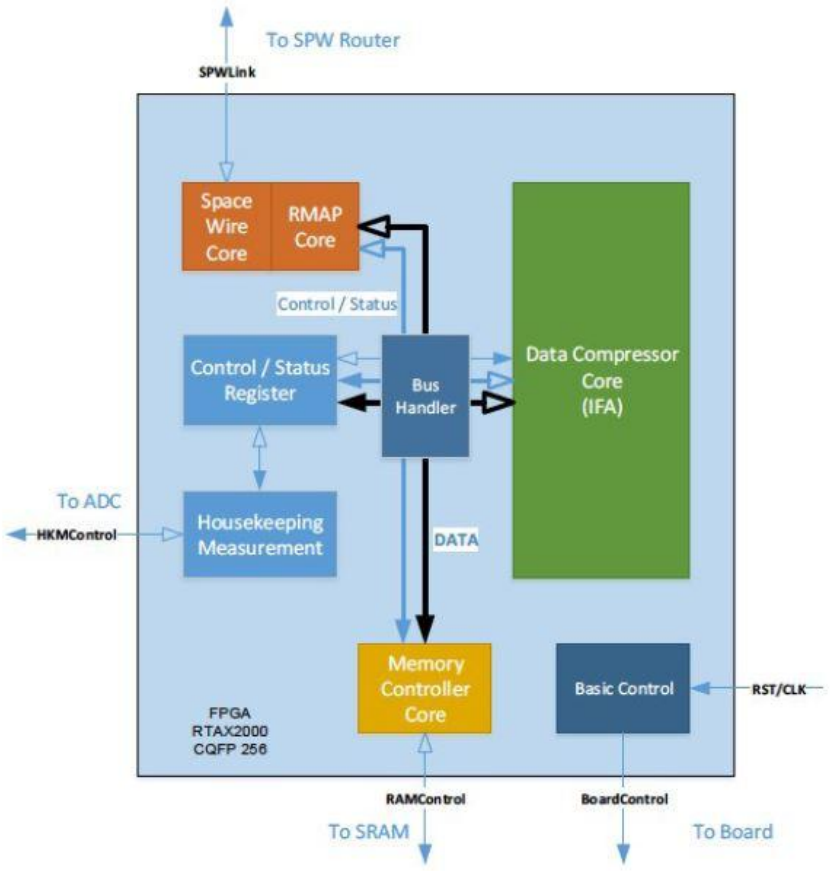


Figure 5 - Architecture of the Router and Data Compression Unit (RDCU). The RDCU combines SpaceWire routing capabilities with a hardware-based data compression engine implemented in FPGA, enabling efficient handling of high data rates.

Redundancy is implemented at multiple levels within the ICU. The unit comprises two identical processing chains (CPU+RDCU+PSU) operating in cold redundancy. The backplane provides cross-strapping capability, allowing both nominal and redundant RDCUs to be managed by either the nominal or redundant CPU+PSU assembly, ensuring that a failure in one chain does not compromise the overall functionality of the system. This design choice significantly enhances the reliability of the instrument and is essential for long-duration space missions.

5 SpaceWire Network Architecture

The SpaceWire payload network is designed to support the high data throughput generated by the multi-camera system while ensuring deterministic behaviour and fault tolerance. The network follows a partially redundant architecture in which the DPUs are connected to the ICU through the RDCUs, which provide SpaceWire routing and data-handling functionalities, enabling flexible data distribution and efficient aggregation of multiple data streams. The ICU single-chain network comprises 8 links to the payload DPUs, 4 internal links between the CPU and RDCUs, and 4 links to the spacecraft, providing sufficient bandwidth and redundancy.

Redundancy at PL network level is implemented through the presence of nominal and redundant communication paths. In case of link or node failure, alternative routing paths can be established to maintain connectivity between the DPUs and the ICU. This approach enhances system robustness and ensures continuity of scientific data acquisition.

In addition to scientific data transfer, the SpaceWire network is also used for telecommand and telemetry exchange, including the distribution of time information. Time synchronization is achieved through PUS Service 9 and SpaceWire Time Codes with low latency, allowing coherent operation across all payload subsystems and with the spacecraft avionics.

The overall network architecture, including nominal and redundant links, provides a comprehensive view of the interconnections between payload units and highlights the role of the RDCU as central routing element.

6 Software Architecture

The ICU software architecture is organized in a layered structure comprising Boot Software (Boot SW), Basic Software (BSP, mainly hardware Drivers), and Application Software (ASW), each providing distinct and complementary functionalities.

The Boot Software is executed immediately at Unit start-up or after processor reset and is responsible for the initial system booting. Its main functions include hardware initialization, basic diagnostics, validation of memory integrity, and management of the Application Software images stored in a one-time programmable PROM memory. In addition, the Boot Software controls the boot sequence by selecting and loading a valid ASW image into executable memory and ensuring safe system start-up conditions. The Boot Software, with its limited capability to accept telecommands, can be also used for ASW patching and for managing safe mode in case of failures. Its limited functionality contributes to its low complexity and high reliability.

The Basic Software provides the core low-level services required for the operation of the ICU, including hardware abstraction, memory management, communication interfaces (e.g. SpaceWire), timing services, and fault detection, isolation and recovery (FDIR) mechanisms. It acts as an intermediate layer between hardware resources and higher-level functionalities, ensuring portability, determinism, and robustness of the system.

On top of these layers, the Application Software along with the chosen Real-Time Operating System (RTEMS) kernel implements the mission-specific functionalities of the ICU. These include payload control, telecommand execution, telemetry generation, science data handling, and real time management of the subsystem interfaces. The ASW operates on top of the services provided by the Basic Software and relies on the Boot Software for system initialization and its activation.

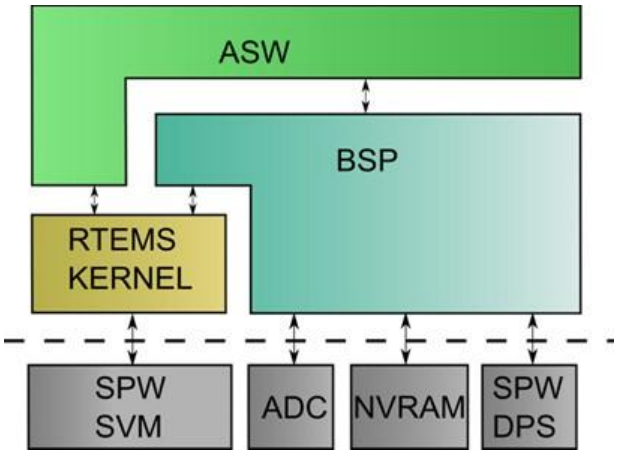


Figure 6 - Layered software architecture of the ICU. The system includes the Operating System, Basic Software (BSP), and Application Software. The Application Software, developed under INAF responsibility, implements payload control, data handling, SW compression and monitoring functionalities.

This layered approach ensures a clear functions separation, enhances system reliability, and supports maintainability and flexibility throughout the mission lifecycle.

7 Robustness and Fault Management

The long-duration observation strategy of the PLATO mission imposes stringent requirements on system reliability, availability and robustness. The ICU has therefore been designed to operate autonomously and to handle a wide range of potential failure conditions without compromising the overall mission objectives.

A key element of this approach is the implementation of extensive monitoring capabilities. The ICU continuously acquires housekeeping data from all subsystems and performs HW and SW limit checking to detect system anomalies. In case of deviations from nominal conditions, the system can trigger predefined actions, such as reconfiguration of operational modes or activation of redundant units.

The SpaceWire network, which forms the backbone of the payload communication system, is also subject to periodic monitoring. Error counters and link status indicators are used to detect communication issues, allowing timely intervention and minimizing the risk of data loss.

Memory management represents another critical aspect. The ICU implements mechanisms for memory protection, scrubbing, and error detection, ensuring the integrity of both software and data structures. In addition, the capability to upload and patch software during the mission provides flexibility in addressing unforeseen issues or to implement enhanced functionalities.

Failure scenarios have been carefully considered during the design phase. These include loss of communication with subsystems, corruption of data in memory, and anomalies in timing signals. The combination of hardware redundancy and software-based recovery procedures (managed by FDIR – Fault Detection Isolation and Recovery tasks) allows the ICU to maintain functionality in the presence of such events.

This robust design is essential to guarantee the continuity of scientific observations, particularly given the long uninterrupted observation periods required by the mission.

8 Design Evolution and Development Status

8.1 Development and Validation

The development of the ICU followed a Proto Flight approach driven by a structured process aligned with ESA standards, progressing through successive design and review phases for de-risking. Early activities focused on defining the system architecture and identifying the key technological solutions. These concepts were progressively consolidated through the Preliminary and Critical Design Reviews (PDR, CDR), leading to the implementation and integration of high TRL (Technology Readiness Level) hardware and software components.

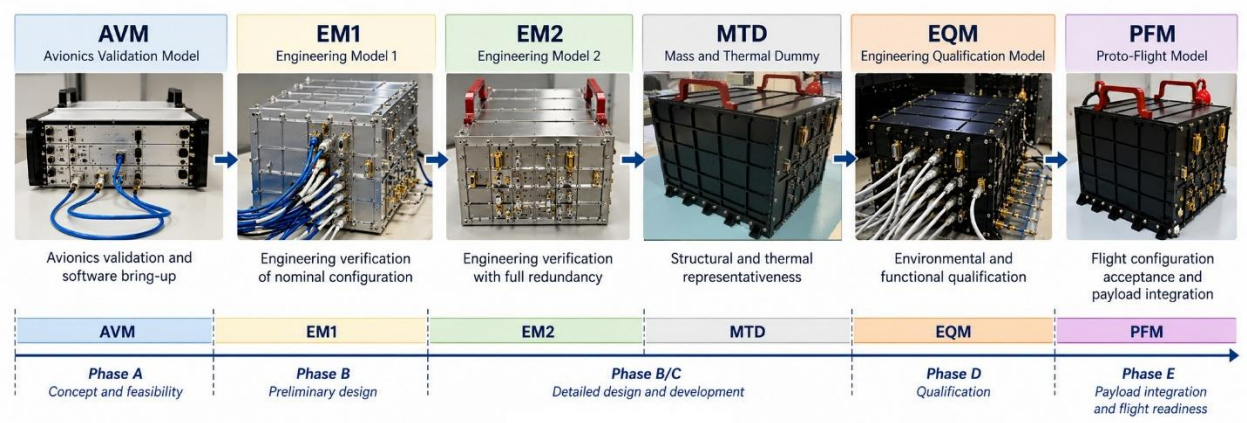


Figure 7 - Evolution of the PLATO Instrument Control Unit (ICU) development models across the ESA project lifecycle, from the Avionics Validation Model (AVM) to the Proto-Flight Model (PFM), highlighting the progressive increase in functional and performance maturity, redundancy verification, environmental qualification, and flight representativeness.

Table 1 - Summary of the PLATO ICU development models, including their primary validation objectives and the main verification activities performed during the incremental development and qualification process.

Model	Objective	Validation scope
AVM (Avionic Validation Model)	Early architectural and software validation	Preliminary validation of ICU architecture, Boot Software, BSP, spacecraft interfaces, and EGSE integration
EM#1 (Engineering Model #1)	Functional validation of nominal architecture	Verification of nominal SpaceWire interfaces, Verification of hardware architecture and interface, DPS integration, preliminary performance assessment, and ASW preliminary validation
EM#2 (Engineering Model #2)	Validation of redundant architecture and distributed integration	Verification of redundant ICU architecture, distributed DPS integration, and software performance characterization
MTD (Mass and Thermal Dummy)	Structural and thermal representativeness	Preliminary validation of ICU mechanical and thermal design Support to spacecraft structural and thermal qualification activities
EQM (Engineering Qualification Model)	Environmental and functional qualification	Qualification under vibration, thermal-vacuum, thermal cycling, and EMC conditions, including robustness and redundancy verification
PFM (Proto-Flight Model)	Flight configuration acceptance and payload integration	Final payload integration, end-to-end telemetry validation, operational readiness verification, and proto-flight acceptance

Figure 7 illustrates the evolution of the ICU development models from the initial Avionics Model (AVM) and Mechanical and thermal Dummy (MTD) up to the Proto-Flight Model (PFM), following a progressive

approach based on functional verification, redundancy consolidation, environmental qualification, and increasing flight representativeness.

Particular attention was devoted to the interaction between hardware and software elements, as well as to the management of overall system complexity. Iterative testing campaigns played a key role in identifying potential issues, supporting de-risking activities, and ensuring compliance with mission requirements.

The validation activities were performed using a representative test configuration including the ESA/Prime provided Spacecraft Interface Simulator (SIS), payload data flow simulators (SIMUCAM) and additional Electrical Ground Support Equipment (EGSE). This setup enabled end-to-end verification of the data flow from the DPUs to ICU telemetry to the Spacecraft, including the performance assessment of SpaceWire communication interfaces and redundant operational chains. The use of synchronized EGSE on both S\C and DPS sides allowed to verify also the requirements on time code distribution performance and science data timestamping.

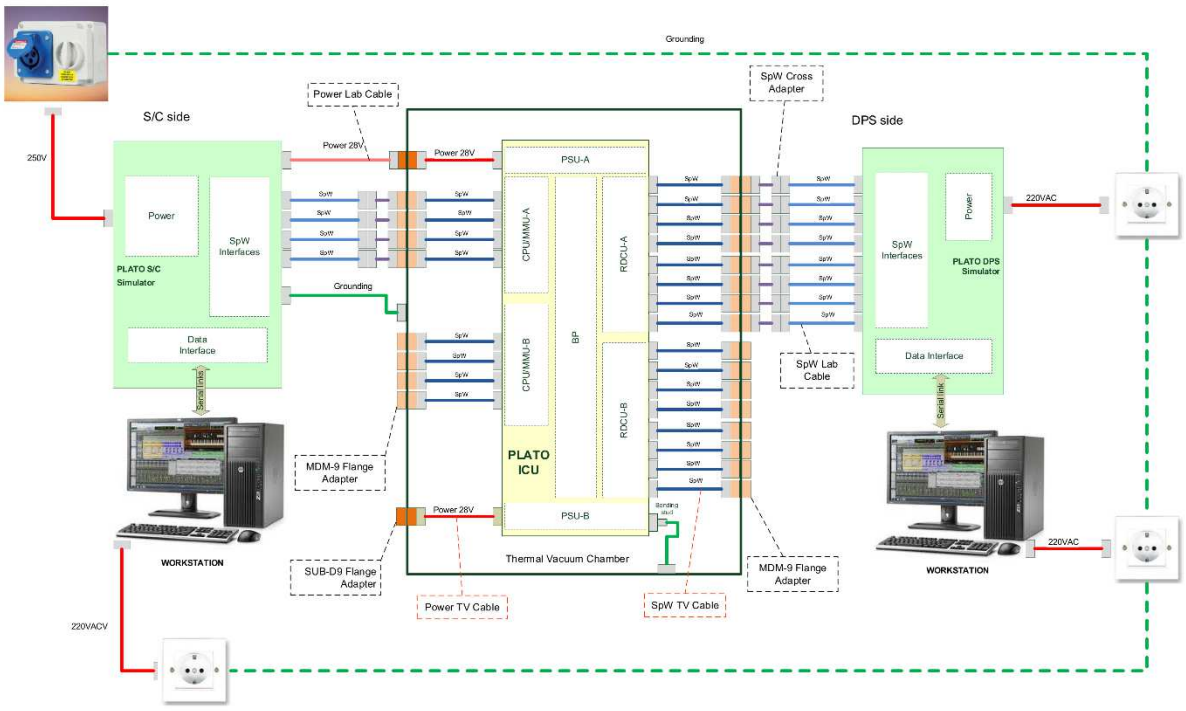


Figure 8 - Test configuration used for ICU validation, including spacecraft interface simulator, payload data flow simulators, and electrical ground support equipment (EGSE).

The development has since progressed to the Proto-Flight Model (PFM) acceptance, through the Qualification Model (E)QM, which has been completed and delivered following its qualification with functional and performance test campaigns. The ICU PFM model is currently undergoing payload-level integration and system verification activities at S/C Prime contractor level.

8.2 PERFORMANCE ASSESSMENT

The main results obtained during the ICU functional and performance test campaigns are summarized in the following section.

The stability of housekeeping parameters observed during the test campaigns is particularly relevant for monitoring the scientific performance of the mission. Variations in temperature, voltage, or current may introduce systematic effects capable of degrading photometric precision. All monitored parameters have been observed to remain stable under steady-state conditions within the expected operational ranges in each phase of the qualification and acceptance phases, confirming the results obtained by analysis during the design phase and allowing cross-correlation. The observed fluctuations remain consistent with nominal electronic noise, confirming that the ICU does not introduce additional noise effects into the scientific measurement chain.

Representative housekeeping parameters measured during ICU performance tests are shown in Figure 9.

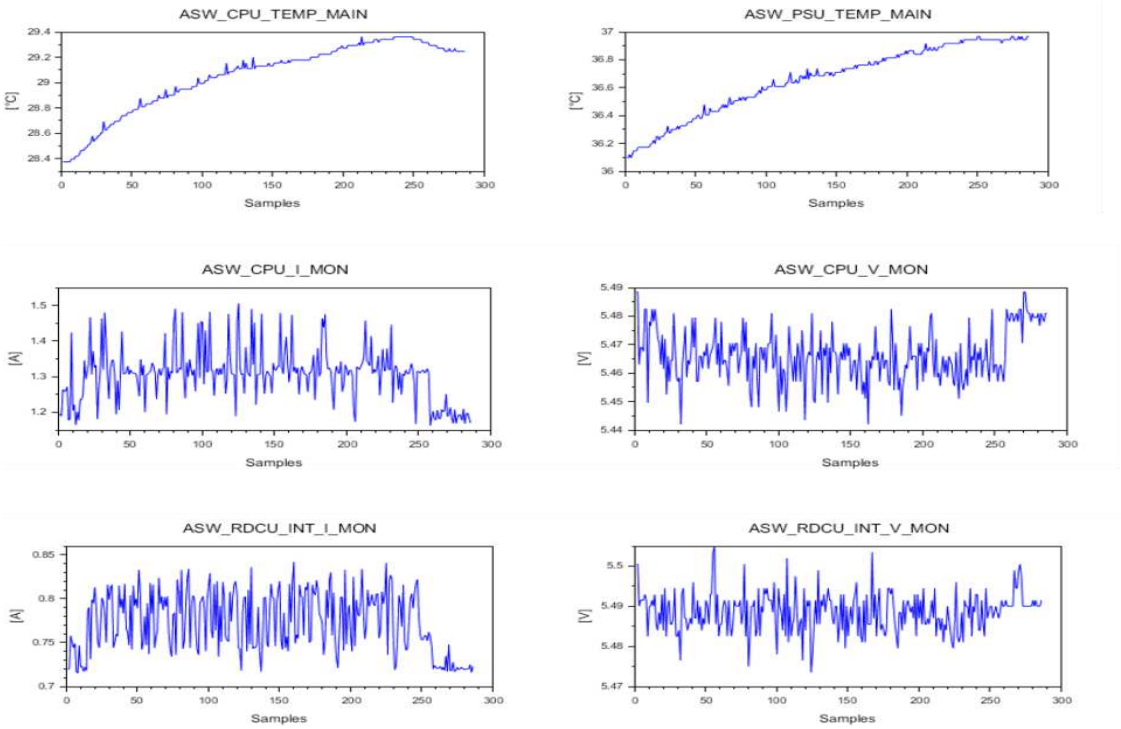


Figure 9 - Example of housekeeping parameters monitored during ICU performance tests, including voltage and temperature. The stability of these parameters confirms nominal system behaviour.

The measured data rate clearly shows the decoupling between the burst-like data generation at payload level and the stabilized output stream at ICU level, confirming the correct operation of the buffering and data handling chain. An example of the measured data rate behaviour during ICU performance tests is shown in Figure 10.

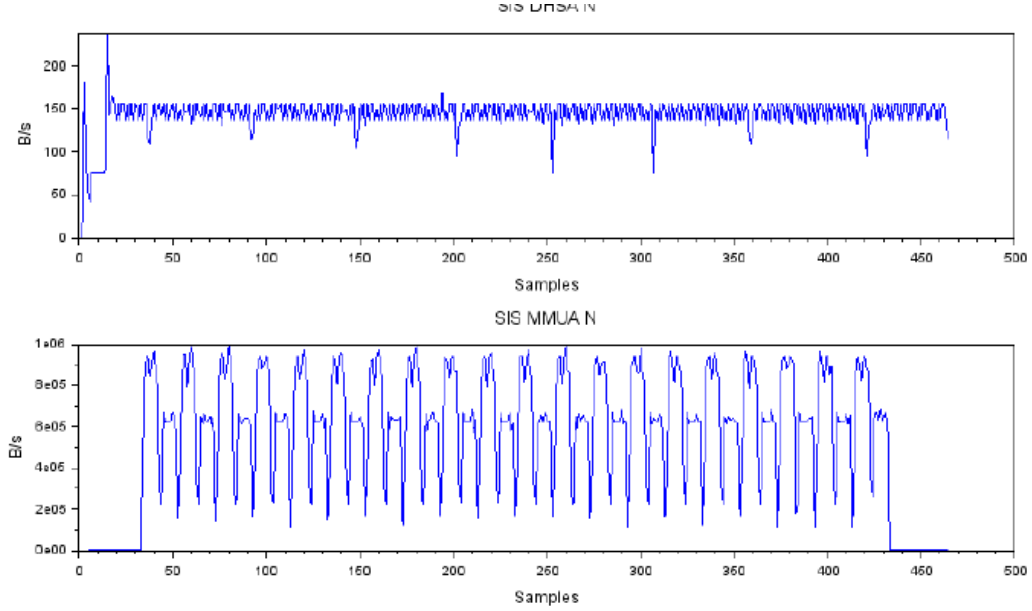


Figure 10 - Example of data rate behaviour measured during ICU performance tests. The payload data generation (MMU) shows a burst-like pattern, while the ICU output (DHSA) provides a stable and continuous data stream. This demonstrates the effectiveness of the buffering and data handling mechanisms implemented in the ICU.

The ICU functional and processing capabilities were assessed through a comprehensive set of functional and performance tests, confirming the system capability to sustain the expected data throughput while preserving data integrity and timing constraints.

The reference period of the tests aimed at assessing the compression performance has been defined to be compatible with the cadence of all used data types (i.e. 4800 s containing 192 HW-compressed exposures, 96 short cadence exposures, and 8 long cadence exposures).

The test campaign successfully validated the parallel execution of both hardware and software compression algorithms, confirming that all simulated science data could be processed within the allocated timeframes without any overlap between consecutive exposures. A systematic comparison of the decompressed outputs against the original input data demonstrated a 100% success rate, confirming the functional correctness of the implemented logic across all acquired products.

Performance evaluation and time margins were assessed individually for each compression pipeline.

For hardware-compressed data, the test campaign demonstrated an average time margin of approximately 16 seconds out of 100, which fully satisfies system requirements and confirms operational viability for real-flight scenarios.

For software-compressed data, the time margins across all currently tested products maintained a value well above 50%.

Analogously, the compression results provided significant margins in the compression ratio with respect to the requirements. Hardware compression achieves ratios of approximately 2.9, while software compression ranges between 3.0 and 4.2 depending on the data cadence and entropy, with higher efficiency observed for long-cadence data (600s). The combined compression performance remains stable around 2.9, ensuring

effective data volume reduction without loss of scientific information. The resulting average output data rate is 5.74 Mbps in the worst case.

CPU load analysis indicates that peak values approach saturation during combined compression scenarios, while average utilization remains well within acceptable operational limits, confirming the robustness of the processing architecture under nominal conditions.

A quantitative summary of the main performance metrics derived from these activities is reported in Table 2.

Table 2 - Summary of ICU performance metrics derived from functional and performance test campaigns.

Parameter	Measured Value	Requirement	Status
HW Compression Ratio	2.86	≥ 2.5	Compliant
SW Compression Ratio	3.0 – 4.2	≥ 2.5	Compliant
Combined Compression Ratio	2.88	≥ 2.5	Compliant
HW Compression Time	< 100 s (4 exposures)	< 100 s	Compliant
SW Compression Time	2.5 s (short cadence), 25 s (long cadence)	< cadence window	Compliant
Peak CPU Load	~97%	< 100%	Compliant
Average CPU Load	21–27%	—	Nominal
SpaceWire Architecture	8 (DPU) + 4 (CPU) + 4 (S/C) links	—	Verified
Housekeeping Sampling Rate	1 Hz	≥ 1 Hz	Compliant

The reported results demonstrate overall compliance with the mission requirements in terms of compression efficiency and timing performance, while a slight exceedance in the data rate is observed under non-representative test conditions.

*This condition represents a conservative scenario and is not expected under nominal operational conditions with a complete and representative science data set.

9 Design Impact on Scientific Performance

The Instrument Control Unit design guarantees the achievement of the scientific performance requirements of the PLATO mission. While the ICU is primarily a data collecting and management subsystem, its role in the data processing chain, including compression, makes it a critical element in providing the needed amount of data for on-ground post-processing and preserving the quality and integrity of the scientific observations.

One of the most important aspects is the preservation of photometric precision. The detection of terrestrial exoplanets relies on the identification of very shallow transit signals, often at the level of a few tens of parts per million. Any degradation introduced during on-board data handling, such as quantization effects, packet loss or compression artifacts, would directly affect the detectability of these tiny signals. For this reason, the ICU implements a fully lossless compression scheme, ensuring that no scientific information is lost during data reduction. The compression ratio (CR), typically of the order of at least 2.5, allows the system to comply with telemetry constraints while keeping data integrity. Timing accuracy represents another key factor. The scientific exploitation of PLATO data, particularly in the context of asteroseismology, requires precise and stable time tagging of observations. The ICU is responsible for distributing the spacecraft time to all subsystems and preserving synchronization across the overall payload. Any timing jitter or drift would

propagate into the light curves potentially introducing systematic errors in frequency analysis. The adopted architecture ensures coherent time distribution and stable synchronization over long observation periods.

The handling of data completeness is equally critical. The ICU must manage continuous data streams from multiple sources with different cadences, buffering and processing them without introducing losses. The combination of hardware buffering and software-controlled data flow allows the system to sustain the expected data rates while minimizing the risk of data gaps. This is essential to avoid interruptions in the light curves, which would degrade both transit detection sensitivity and asteroseismic analysis.

Another important contribution of the ICU design lies in the robustness of the system. The implementation of cold redundancy, combined with autonomous monitoring and fault management functionalities, ensures that scientific operations can continue even in the presence of failures. This is particularly relevant for a mission like PLATO, which relies on long, uninterrupted observation campaigns.

Overall, the ICU design has been driven by the need to manage the payload subsystems and preserve the scientific content of the retrieved data throughout the on-board processing chain as a whole. The combination of lossless compression, precise timing, robust data handling, and system redundancy ensures that the quality of the scientific products is not limited by the on-board infrastructure.

The achieved timing stability and continuous data handling ensure that no additional noise is introduced at levels relevant for transit detection, typically at the level of a few tens of ppm.

This is particularly relevant for the detection of Earth-sized planets around solar-like stars, which represents one of the primary science objectives of PLATO.

10 CONCLUSIONS

This paper presented the architecture, design evolution, and validation of the Instrument Control Unit (ICU) developed for the PLATO mission. The ICU plays a key role within the payload data processing system, ensuring reliable control of the instrument and efficient handling of the large data volume generated by the multi-camera system.

The adopted architecture, based on a distributed processing concept combined with a centralized DPS management, has proven effective in addressing the stringent requirements of the mission in terms of data throughput, timing accuracy, system reliability and availability. In this framework, the interaction between Data Processing Units and the ICU enables scalable data handling while preserving overall system flexibility.

A critical aspect of the design is the implementation of hardware-assisted data compression, which allows the reduction of scientific data volume to comply with spacecraft telemetry constraints without overloading CPU processing resources. This solution, combined with a modular and layered software architecture, contributes to the robustness and maintainability of the system, supporting long-term mission operations.

The ICU development successfully progressed through all major design and verification phases, culminating in the completion and delivery of the Proto-Flight Model (PFM) following comprehensive qualification and acceptance test campaigns at Unit and DPS level. The unit is currently undergoing integration and verification stages at payload level, supporting system-level testing activities at the Prime contractor premises.

Overall, the results demonstrate that the adopted design approach effectively meets the mission requirements and provides a reliable and flexible solution for the management of complex, high-throughput spaceborne instrumentation.

From a scientific standpoint, the ICU can be regarded as an integral part of the measurement chain rather than a purely supporting subsystem. Its design directly influences the fidelity of the data produced on board and delivered to ground, and therefore contributes to the ability of the mission to achieve its scientific objectives.

11 ACKNOWLEDGEMENTS

This work has been funded thanks to the Italian Space Agency (ASI) support to the Project, as defined by the ASI-INAF agreement n. 2022-28-HH.0 “Scientific activity for the PLATO Mission - D Phase”.

The authors wish to thank all the PLATO Mission Consortium teams for their invaluable work across time, from the Mission Selection and Adoption to final implementation and verification.

12 References

1 Rauer, H., Catala, C., Aerts, C., et al. (2014). The PLATO 2.0 mission. Experimental Astronomy, 38, 249–330.

2 European Space Agency (ESA). (2017). PLATO Definition Study Report (Red Book). ESA/SRE(2017)1.

3 Di Giorgio, A. M., Cosentino, R., Focardi, M., et al. (2018). The PLATO payload and ICU architecture. Proceedings of SPIE, 10698, Space Telescopes and Instrumentation 2018.

4 Focardi, M., Del Vecchio, C., Cosentino, R., et al. (2019). PLATO on-board software and data processing architecture. Proceedings of SPIE, 11180.

5 Focardi, M., Cosentino, R., Di Giorgio, A. M., et al. (2020). PLATO ICU data processing architecture and on-board management. Proceedings of SPIE, 11443, Space Telescopes and Instrumentation 2020.

6 Loidolt, D., Ottensamer, R., Luntzer, A., et al., “A combined software and hardware data compression approach in PLATO,” Proc. SPIE 11443 (2020).

7 Cosentino, R., Focardi, M., Di Giorgio, A. M., et al. (2022).The PLATO Instrument Control Unit: design and performance. Proceedings of SPIE, 12180, Space Telescopes and Instrumentation 2022.

8 Focardi, M., Cosentino, R., Di Giorgio, A. M., et al. (2023). PLATO ICU Design Description Document. PLATO-INAF-PL-DD-002.

9 Russi, A., Giusi, G., Di Giorgio, A. M., Cosentino, R. (2026). PLATO ICU Application Software User Manual. PLATO-INAF-PL-MAN-0012, Issue 1.10.