Project Name : SCORPYOS

Document Type : RP

Title : D5.1 – Final Public Report

Issue : 1A

		Name & Function	Signature
Approval	:	Rodolfo Martins *(Project Manager)*	

Document Category

A: Document for Approval		**R**: Document for Review		**I**: Document for Information	X

Distribution List

Name	Org.	Function	Copies
Kostas Marinis	ESA	Technical Officer	Electronic Copy

Contact List

Name	Org.	Function	Contact
Rodolfo Martins	EVOLEO	Project Manager	Phone: +351 229 424 327 Email: info@evoleo.tech

Document Change Record

Iss./Rev.	Date	Affected Sections	Reason for Change
1	2025-03-12	All	First Issue
1A	2025-03-31	All	Proprietary, Copyright information and references removed

The rest of the page has been left intentionally blank

Table of Contents

Figure List

Table List

The rest of the page has been left intentionally blank

1 GENERAL INFORMATION

1.1 Introduction

The SCORPYOS project, developed by EVOLEO, has created and presented a breadboard of a reference design for a "Minimalistic Supervisor." This equipment encompasses both hardware (HW) and a suite of software/firmware (SW/FW) tools aimed at facilitating the seamless configuration and parameterization of the operation of inputs, outputs, and internal functions without necessitating specific coding skills.

This document is the final public report of the work that has been carried out under ESTEC contract 4000135813/21/NL/GLC/ov “SCORPYOS – Supervisor COncept foR sPace SystemS” – Minimalistic Supervisor based on ATMegaS128.

1.2 Acronym List

Acronym	Description
COTS	Commercial off the Shelf
DIG	Digital Design
EGSE	Electrical Ground Support Equipment
ELE	Electrical Engineer
EM	Engineering Model
FDIR	Failure Detection Isolation and Recovery
FMEA	Failure Mode & Effect Analysis
FPGA	Field Programmable Gate Array
FW	Firmware
IP	Intellectual Property
LEO	Low Earth Orbit
MIMPS	Mega Instructions per Second
MPSoC	Multi Processor System on Chip
OBC	On-Board Computer
OS	Operative System
P/L	Payload
PF	Platform
PLIU	Payload Interface Unit
RTD	Resistor Temperature Dependant
RM	Reconfiguration Module
RR	Requirement Review
RTOS	Real Time Operating System
S/C	Spacecraft
SEE	Single Event Effect
SEFI	Single Event Functional Interrupt
SEL	Single Event Latchup
SYS	Systems Engineering
TBD	To Be Defined
TC	Telecommand
TID	Total Ionizing Dose
TM	Telemetry

Table 1-1 – Acronym List.

2 Objectives and constraints

The widespread use of COTS in LEO is constrained by their susceptibility to faults and the difficulty in implementing custom fault detection and recovery actions (FDIR) closely to the system in question.

Thus, faults may pass undetected until they propagate outside the COTS system and cause a significant impact on the spacecraft. This delay in recovery actions may contribute to lower in-orbit reliability and availability. The challenge is in supporting designers who wish to protect their systems via recurrent methods and tools which are also cost-effective.

EVOLEO proposes to develop and demonstrate an elegant breadboard of a reference design for a "Minimalistic Supervisor", including HW and a set of SW/FW tools to easily configure/parametrize the operation of the inputs/outputs and internal functions without the need for specific coding.

The SCORPYOS project targets the development of a Supervisor Concept for Space Systems, operating alone or as a supporting unit, focusing on small subsystems, high simplicity of usage and LEO missions.

The ultimate goal of the implementation of the SCORPYOS concept is to significantly increase the availability of the COTS system under supervision via a reference design HW and a set of SW/FW tools that allow a user to easily configure/parametrize the reference design to its use-case.

Therefore, there are three basic key goals to be accomplished within this activity that drive the entire thinking of the project:

1. To design and implement an elegant breadboard (EBB) reference design, HW and SW/FW tools, for a minimalistic supervisor to support COTSbased subsystems capable of fulfilling a large spectrum of the market/use cases needs in terms:
    a. Technical performance and capabilities (FDIR functions, dependability, flexibility of use)
    b. Commercial soundness (cost/time to market/recurrence)

2. To validate via testing the applicability of the reference design to, at least, two actual use case scenarios via streamline (re)configuration and parametrization of the reference design to the use-case scenario.

3. To layout in collaboration with industry partners future steps for adoption of the developed technology.

These objectives are to consider the application of the technology development to a:

- Closer to production, flexible and recurrent oriented reference designs for minimalist supervisor for COTS-based subsystems.
- Small to minisatellite classes (+20kg) in LEO for New Space using COTS.
- Small subsystems which require improvements in availability and overall reliability.

3 Feedback from Industry

EVOLEO investigated use cases, including similarities and distinguishing factors, and consulted with partners such as Airbus, ASP and Realtra, to develop a clear scenario of use cases of interest to the industry.

Below is the feedback obtained by the different companies.

3.1 Airbus

Can we use this to supervise Xilinx Versal? And as boot method for KU060?

Consider PMbus for power monitoring

To protect/snoop AI System-on-module (ex: NVIDIA Jetson board)

3.2 ASP

May be relevant to consider SCORPYOS has basis for NewSpace PSDU – more granular

Should there be redundant supervisors?

Consider memory required to store and process all power telemetries

What telemetries and how can we access them? – integration with OBC FDIR concept

3.3 Realtra

Can we consider SCORPYOS for hard/non - LEO missions? Lunar for example, where we can't fly VPU/GPU?

What faults can you detect without changing current VIKI (Ariane VideoKit) design and HK datapool?

Will provide test telemetry (nominal and fault) for post-processing using SCORPYOS algorithms.

3.4 Summary

The SCORPYOS system is designed to be versatile and applicable to various equipment, extending from Commercial Off-The-Shelf (COTS) devices to those operating beyond Low Earth Orbit (LEO).

Its primary objective is to safeguard specific System on Chip (SoC) and AI engines, including Versal, Jetson, and Myriad.

The value of intelligent monitoring will first be demonstrated on standard equipment, establishing its efficacy and reliability.

This foundational success will then pave the way for moving towards more tailored designs, incorporating the advanced capabilities of the SCORPYOS system.

The rest of the page has been left intentionally blank

4 Design Summary

4.1 Microcontroller selection

Options have been identified and trade-off metrics defined. The following bullet points were considered to select feasible devices:

- The device should be an MCU with embedded interfaces and known radiation tolerance compatible with LEO
- The devices should be available at the time of RR - minimum available as EM but preferably with a qualified option (Military grade or QML equivalent is acceptable)
- It should have a representative development kit available
- The cost of the device should be lower than 10k€ and preferably lower than 5k€ - applies to flight model which could be military grade, QML-Q or equivalent grade
- Preferably, the device should be capable of AI/ML inference for future proofing. This relates to CPU architecture and instruction set.

Devices pre-emptively excluded from analysis:

- **FPGAs:** due to design complexity and limitation in scalability of design (example: RT Kintex US+, RT PolarFire, RTAX2000
- **Complex SoC:** Due to design complexity and no radiation tolerant parts within the budget (example: Zynq7000, Zynq US+, Versal, PolarFireSoC)

The following feasible microcontrollers were identified. Some were excluded from further analysis.

- **SAM3X8ERT:** Radiation tolerant Cortex-M3 with internal ADC/DAC
- **SAMV71Q21RT:** Radiation tolerant Cortex-M7 with floating point unit, memory management unit, tightly coupled memory, and ECC protected volatile and non-volatile memories.
- **SAMRH71:** Radiation hard version of the SAMV71 with some changes to embedded memory and interface options.
- **ATMEGAS128**: Radiation tolerant version of the 8-bit ATMega128. Excluded due to low pin-count and low CPU performance.
- **ATMEGAS64:** Similar to the ATMEGAS128 with the inclusion of CAN controller. Excluded due to low pin-count and low CPU performance.
- **GR716(B):** GR716B not yet available.
- **MSP430FR5969-SP:** Low power 16-bit RISC MCU in FRAM technology. Excluded due to low pincount and low CPU performance.
- **VA41630:** Rad-hard Cortex-M4 with TMR'd register and 1x SpaceWire, integrated ADC/DAC and ECC protected volatile and non-volatile memories.
- **VA41627:** Similar to VA41630 but without ADC/DAC, CAN, Ethernet and SpaceWire interfaces

Device	Radiation	CPU	Cost	Functions	IO	Memory	Interfaces
SAM3X8ERT	TID 30krad SEL 62MeV	Cortex-M3 - 84MHz	HiREL RT Plastic - 349€ (240MOQ) EM (967€) QML Q EQ - 3384€(5 MOQ)	Memory Protection unit (MPU) Temp Sensor Internal RC oscillator	103 IO + 4x32b PIO	512 KB Flash 100KB SRAM 16KB ROM Memory controller with ECC	USB 4 USART 6SPI 8x12b PWM 16x12b ADC 2x12b DAC Ethernet 2x CAN
SAMV71Q21RT	TID 30krad SEL 62MeV	Cortex-M7 100MHz	TBD	Floating point unit Tighly coupled memory (TCM) MPU DSP instructions Embedded voltage regulator RC oscillator Temp sensor AES256 key algorithm Hash (SHA1,256)	194 IO + 7xPIO	2MB Flash with ECC (DEC) 384 KB SRAM TCM (ECC) 768 KB SRAM multiport (ECC) memory controller	1x Ethernet QSPI 2x CAN 3x USART 5x UART 2x SPI 2x I2C 2x PWM 1x 12b DAC 2x AFEC (ADC)
SAMRH71	TID 100krad (20krad for NVM) SEL 62.5 MeV	Cortex-M7 100MHz	TBD	FPU MPU TCM DSP instructions Oscillator PLL Clock and power failure detectors	194 IO + 7xPIO	128kB Flash with ECC (DEC) 384 KB SRAM TCM (ECC) 768 KB SRAM multiport (ECC) memory controller with ECC	10x SPI/UART/i2C 1x Ethernet 1x QSPI 1x CAN FD 2x SpaceWire ports + Router 1x 1553
VA41630 (same has 41620 but with internal 256Kb NVM FRAM)	TID 300krad SEL 110MeV	Cortex M4 100MHz	EM 6000€ Mil-PFR-38534-K 11327€ Plastic QFP (Vorago HiRel) 4927€ (MOQ 2)	DSP instructions TMR registers FPU Internal oscillator On-chip LDO	140 GPIO	64KB SRAM DATA EDAC 256KB SRAM Instruction EDAC 256KB NVM Memory Scrubber	3x UART 3x I2C 3x SPI 2x CAN 2.0 Ethernet SpaceWire 8ch 12b ADC 2c 12b DAC
VA41627	Same as VA41630 but without ADC,DAC, CAN, Ethernet and SpW, with internal 256Kb NVM FRAM						

Table 4-1 – Summary specifications for proposed MCU

Device	Cost	Performance	Interfaces	Embedded memory	Radiation tolerance	Development flow	Final Score
SAM3X8ERT	TBD	5	7	5	7	8	102
SAMV71Q21RT	TBD	9	8	10	7	8	133
SAMRH71	TBD	9	8	8	8	8	131
VA41630	TBD	7	9	7	10	6	130

Table 4-2 –MCU trade-off table

Radiation tolerance of all devices is generally good but the Vorago devices due to their HARDSIL technology are radiation hardened and thus have virtually unlimited access to all orbits, environments and missions without any radiation related constraints.

Cortex-M4 and M7 (both Vorago and Microchip devices) have DSP instructions set which is useful for certain machine learning algorithms which can benefit from ARM's CMSIS-DSP library for DSP functions. Nevertheless, note that DSP performance of M7 can be double of M4.

The Cortex M4 and M7 from Microchip and Vorago contain a single precision floating point unit which may simplify the deployment of the inference engine on the MCU, since training usually uses floating-point data on GPU graphic cards. This MCU would not require or limit the need for quantitation to integers and potentially have achieve higher accuracy than inference engines running on integer operations.

There is an advantage to Microchip devices when it comes to development environment, tools, and technical support.

In terms of interfaces, the SAMRH71 and VA41630 do provide substantial options including SpaceWire but the SAMRH71 does not include internal ADC/DAC. The SAMV71RT does not include SpaceWire but does have an analog front end controller (AFEC) for programmable gain, single end and differential inputs. This would greatly simplify the design of analog acquisition chain and minimize board area.

The information collected suggested that the best candidates were SAMV71, SAMRH71 and VA41630 and, based on costs and delivery times, the SAMV71 was the choice to start the development process

4.2 Budgets and Specs

The supervisor unit electronics card provides processing and interface features to supervisor and control an external system. The main features of the unit are:

Processing	ARM-Cortex M7 @ 300MHz with FPU
Memory	2048Kb Flash 384Kb SRAM 256Mb SRAM (option) 256Mb NOR Flash (option) ECC protected
Interfaces	2x High Density flight connectors 4x RS-422 (one of them used for configuration) 2x CAN 7x GPIO 4x Digital inputs (5V Logic Level) 4x Analog inputs (0V~5V) 1x SPI (3x Slave Select)
Power	Vin: 8V~12V (LCL protected with UVP/OVP/OTP) <1.2W
Dimensions	145x139x29.5mm
Mass	390g
Operating temperature	-40ºC – 100ºC
Reliability	Automotive, QML-Q equivalent and JAN Latch up and high current protections Overtemperature protection Under-voltage & Over-voltage LCL protections Granular current monitoring and power switching
Radiation tolerance	SEL > 40MeV*cm2/mg TID > 20krad

Table 4-3 – Summary of supervisor unit specs

The rest of the page has been left intentionally blank

4.3 Mechanical Design

The mechanical design follows the proposed concept with the goal of achieving a small and lightweight unit.

Flight and ground connectors are placed in adjacent faces of the unit for easier AIT.

Estimated mass:	390g
Dimensions:	145x139x29.5mm

Table 4-4 –Physical dimensions

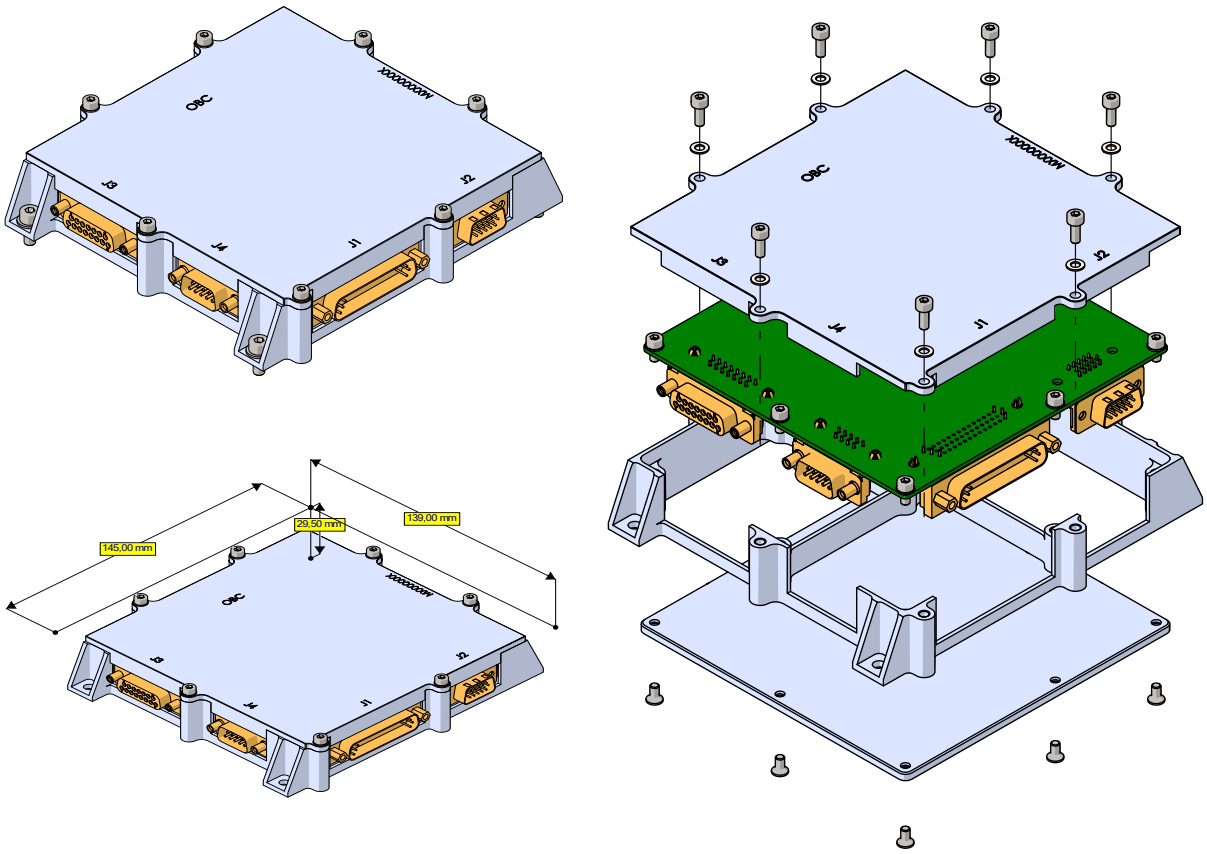

Figure 4-1 – Renders and drawings for mechanical assembly

After conceptualizing the mechanical design, the project advanced to the production phase. Initially, the design was generated using 3D printing to validate the proposed system in terms of dimensions and structural integrity.

Following successful validation, the part was then machined to meet the final specifications.

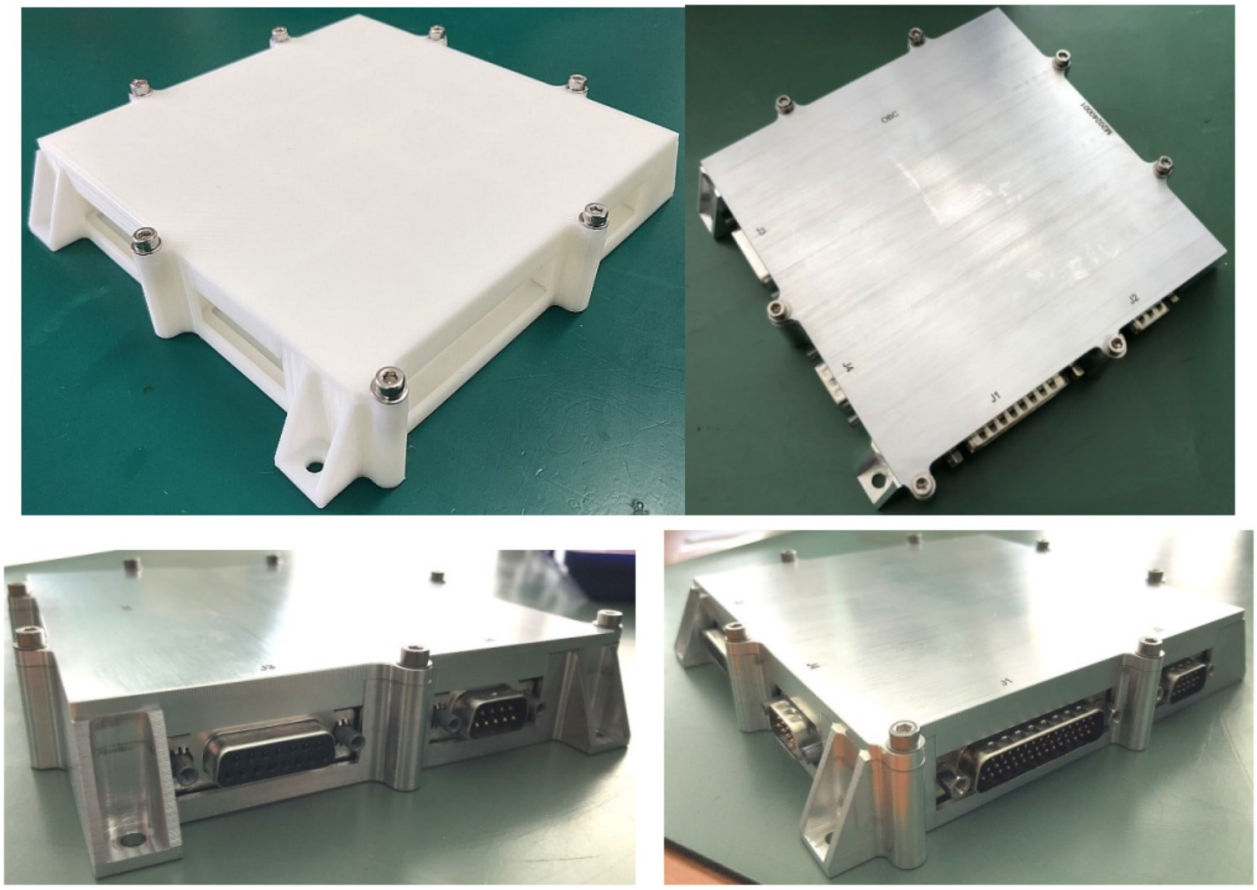

Figure 4-2 – Implemented mechanical assembly

The rest of the page has been left intentionally blank

4.4 Electrical Design

The unit provides two sets of interfaces – flight and ground.

Flight interfaces (J1, J2) are meant for interfacing with other S/C units such as power supplies, system under supervisor, main OBC.

Ground interfaces (J3, J4) are dedicated to programming and live monitoring during AIT.

All connectors are rectangular DSUB High Density (HD), except for J4 which is a regular 9 pin DSUB.

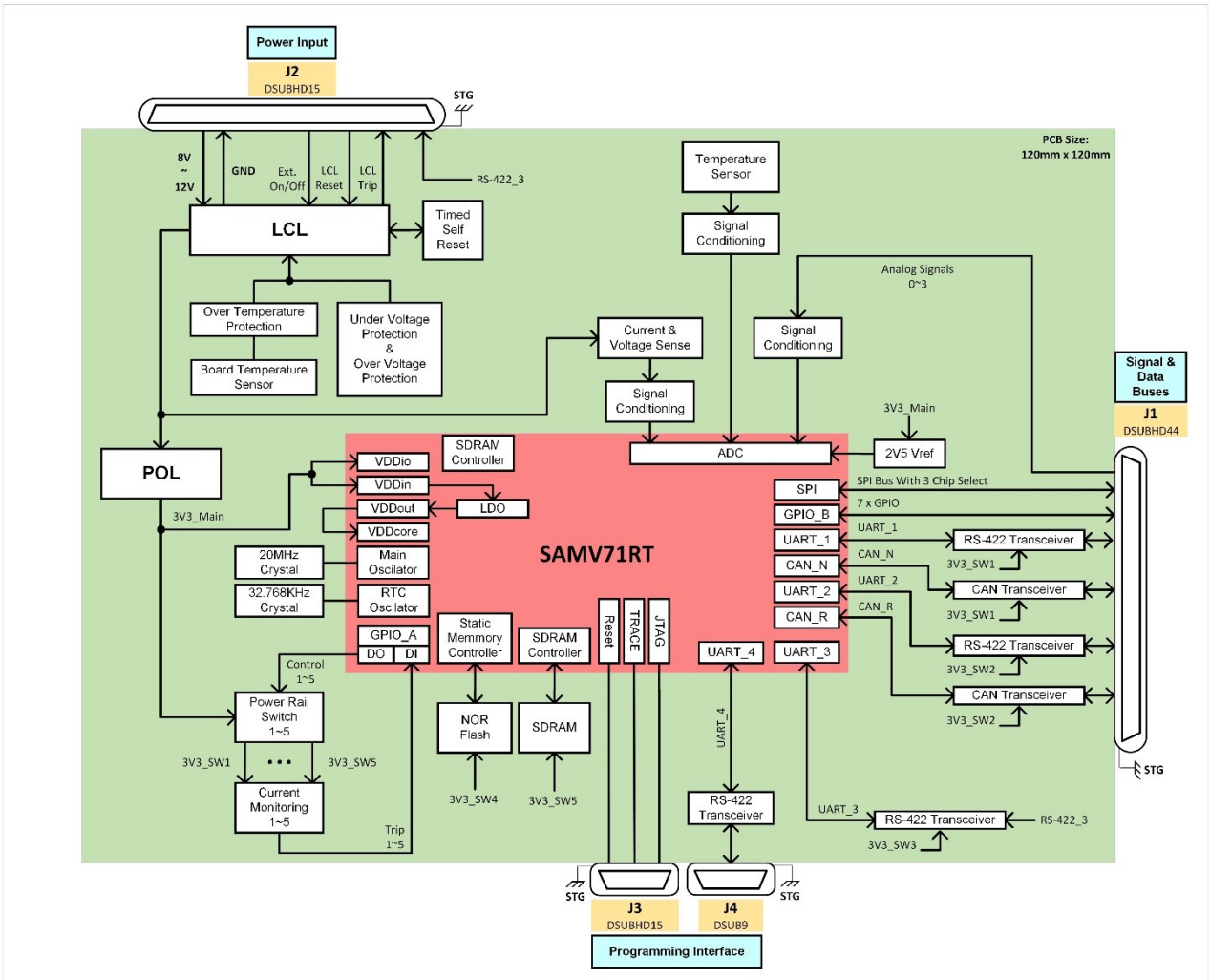

Figure 4-3 – SCORPYOS block diagram

J1 – HD44 – Data and control		
Type		**Nº pins**
RS-422_1	Standard	4
RS-422_2	Standard	4
CAN_N & CAN_R	Reliable	4
SPI (3x CS)	Standard	6
GPIO (I/O and interrupts)	Standard	7
Digital Inputs	Standard	4
Analog Inputs (comparison and samples)	Standard	4
GND	Standard	8

J2 – HD15 – Power		
Type		**Nº pins**
PWR_main_in	Reliable	1
PWR_main_rtn	Reliable	1
GND	Reliable	5
LCL reset	Reliable	1
LCL Trip	Reliable	1
RS-422_3	Reliable	4
LCL Control	Reliable	2
----------	----------	------

Table 4-5 – Flight connectors summary pinout.

J3 – HD15 – Debug	
Type	**Nº pins**
TRACE _ debug	5
GND	2
JTAG	5
Reset test	2

J4 – DSUB9 - Programming	
Signal	**Pin**
RS-422_4	4
GND	5
----------	---
----------	---

Table 4-6 – Ground connectors summary pinout.

The LCL circuit limits the current on the input in case of an overcurrent situation. The circuit limits the current to a defined threshold and only allows a maximum current. The LCL circuit contains an undervoltage, overvoltage and overtemperature protection.

The analog signals, intraboard and from interfaces, are conditioned and then sampled by the ADC embedded in the microcontroller. This 12-bit sample and hold ADC already includes a programmable gain amplifier and integrated multiplexers for up to 12 independent analog channels.

Index	Input Signal
0	Analog_IF_0
1	Analog_IF_1
2	Analog_IF_2
3	Analog_IF_3
4	I_Main
5	V_Main
6	Board Temp
7	MCU Temp

Table 4-7 – ADC channels

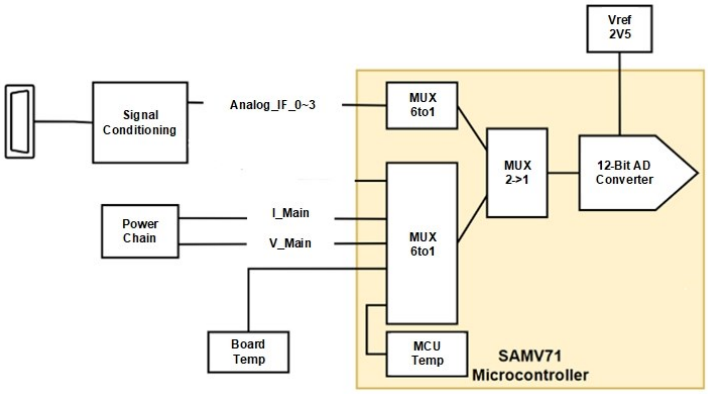

Figure 4-4 – Analog sampling architecture

Specific board elements are powered by monitored and switchable 3V3 rails derived from the main 3V3 supply.

These rails provide protection against high current events in specific IC components such as memories and transceivers. The current in these rails is monitored, converted to a voltage and then compared to a reference voltage.

Power Rails	Load
1	RS422_1 Transceiver
	CAN_N Transceiver
2	RS_422_2 Transceiver
	CAN_R Transceiver
3	RS422_3 Transceiver
4	NOR Flash
5	SDRAM

Table 4-8 – Individually monitored and switchable 3V3 power rails

The SCORPYOS unit contains redundant CAN bus interfaces. The power rail for each transceiver is monitored and switchable according to the proposed power rail architecture.

The microcontroller performs the supervision and control functions defined using the visual design tool. It handles all analog and digital interfaces provided by the system.

Figure 4-5 – Board After production

4.5 Firmware Design

The firmware implements a component-based architecture focused on the decomposition of the design into individual functional or logical components that represent well-defined communication interfaces containing methods, events, and properties. It provides a higher level of abstraction and divides the problem into sub-problems, each associated with component partitions.

The firmware is structured in several layers, providing a clear separation between the different components. These will be organized into application, handler and driver layers, ensuring modularity and facilitating system maintenance and evolution. The following diagram shows the components that will be developed to meet the use cases mentioned above.

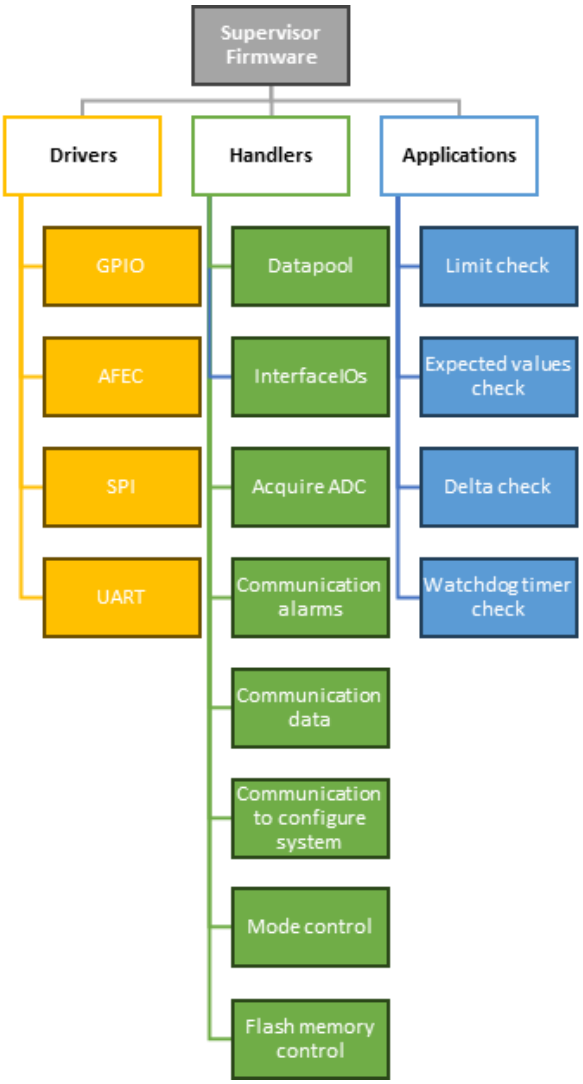

Figure 4-6 – Firmware components

- The GPIO allows control and monitoring of digital pins, enabling the reading of states and the activation of external devices. Each GPIO could be configured with or without interrupts enabled.
- The AFEC (Analog Front-End Controller) is responsible for the interface with analog devices, converting analog signals into usable digital data.
- The SPI (Serial Peripheral Interface) device driver is a synchronous serial communication interface used to interconnect peripherals and sensors. In this use case, the SPI will be used to access the external flash to save the configurations performed on system.
- The UART (Universal Asynchronous Receiver-Transmitter) driver is a asynchronous serial communication interface for exchanging data between the control unit and the SCORPYOS and for configure the system.
- The datapool handler will be responsible for storing and managing data collected from different sources, centralizing system information. Each application should get the information from datapool to process that data.
- The InterfaceIOs Handler have the responsibility to do the interface between GPIOs driver and different handlers/ Applications. With this interface it will be possible to read all the status, write/read a status for a specific pin and inform the system that a interrupt happens on system.
- The handler used to acquire the ADC values manages the acquisition of analog data, converting signals from sensors and devices into digital data for use by applications.
- The communication alarms handle the sending of alarms generated by the system, ensuring that failures or critical events are communicated correctly.
- The communication data handler manages the exchange of data between the different modules and the system, ensuring the integrity and continuous flow of information. In these use cases, the control unit that receive the alarms will also generate the data but could be different systems.
- The handler communication to configure the system is responsible for the communication required to configure the system, allowing adjustments and updates as needed through UART driver.
- The handler layer contains the mode control to manages the system's different operating modes, ensuring that the firmware adapts according to the context or state of the device.

The applications are created to implement the different versions methods, expect the data field check, because this is implemented directly on the communication data handler.

This modular firmware structure facilitates maintenance, improves code readability, and allows the system to be scalable as new requirements arise.

4.5.1 Verification Methods

Limit Check is a fundamental verification method used in supervisory systems to ensure that the values of specific parameters remain within predefined ranges, considered safe or operational. This method is particularly effective in monitoring analog readings such as temperature, current, voltage, reaction wheel RPM, although it can also be applied to certain types of digital data.

Effective implementation of Limit Check in the SCORPYOS system will provide a fundamental layer of protection, enabling rapid detection of abnormal conditions and facilitating proactive responses to maintain the integrity and optimal performance of the supervised system.

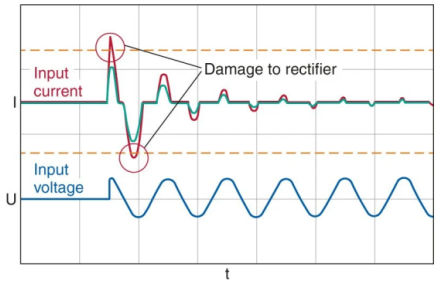

Figure 4-7 – Example of input current thresholds

Expected Value Check is a sophisticated verification method used in digital supervisory systems, particularly effective in monitoring parameters that contain information encoded in specific bits. This method involves applying a bit mask to a parameter and then comparing the result with a predefined value. Could be used to check component status and operation modes or detect errors in Communication Protocols.

Integrating Expected Value Check into the SCORPYOS system will provide a robust capability to monitor and validate complex states of digital systems. This is particularly valuable in environments where multiple states or configurations are encoded in bit registers, allowing for accurate and efficient supervision of critical subsystems.

Delta Check is a verification method used to monitor the variation between consecutive readings of a given parameter, ensuring that this variation remains within predefined limits. This method is particularly useful in detecting abrupt or abnormal changes in dynamic systems.

Effective implementation of Delta Check in the SCORPYOS system will provide an additional layer of monitoring, increasing the robustness and reliability of system supervision, especially in the early detection of gradual or sudden deviations in critical parameters.

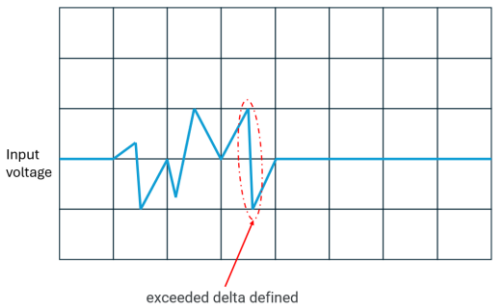

Figure 4-8 – Example of input voltage signal with delta overshoot detection

The **Watchdog Timer Check** is a critical supervision method used to monitor the continuous and correct operation of external units or processes. This method is based on the principle that a functional unit must periodically signal its normal operation by resetting a dedicated timer. This should be used to monitoring critical subsystems.

Integrating Watchdog Timer Check into the SCORPYOS system will provide a critical layer of protection against silent failures or lockups in external drives. This functionality is especially valuable in systems that require high reliability and continuous operation, allowing rapid detection and response to potentially critical anomalies.

Data Field Check is a versatile verification method used to monitor and analyze specific data fields received through various communication protocols, such as UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface), or CAN (Controller Area Network). This method allows the application of verification techniques traditionally used for analog or digital values to data transmitted by serial communication systems.

Integrating Data Field Check into the SCORPYOS system will provide a robust capability to monitor and analyse data from multiple sources and communication protocols. This functionality is particularly valuable in complex systems where multiple subsystems interact and transmit critical data. By applying familiar checking methods to these data fields, SCORPYOS can provide comprehensive and adaptive monitoring capable of detecting anomalies and ensuring the operational integrity of the system.

The rest of the page has been left intentionally blank

4.6 Configuration Tool

The configuration software is divided into two subsystems: the configuration and the system monitoring tools.

The configuration tool allows for the setup and customization of system parameters, ensuring that the system operates according to the specific requirements of the application.

The system monitoring tool provides real-time data and diagnostics, enabling users to observe and analyse the system's status.

Access to this software is restricted to authorized users only, requiring a valid username and password. This ensures that only individuals with the necessary credentials can configure and monitor the system, maintaining the integrity and security of the operations.

Figure 4-9 – Login page

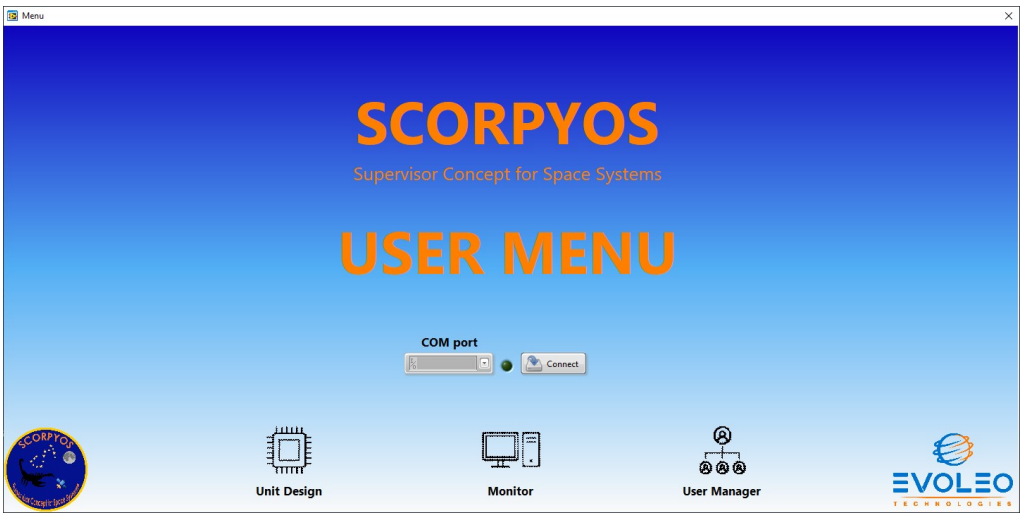

Figure 4-10 – Main Menu

In the system configuration menu, users will have the capability to save various types of configurations referred to as use cases. For each configuration, it will be necessary to first define the hardware settings, followed by the operational mode of the observability system.

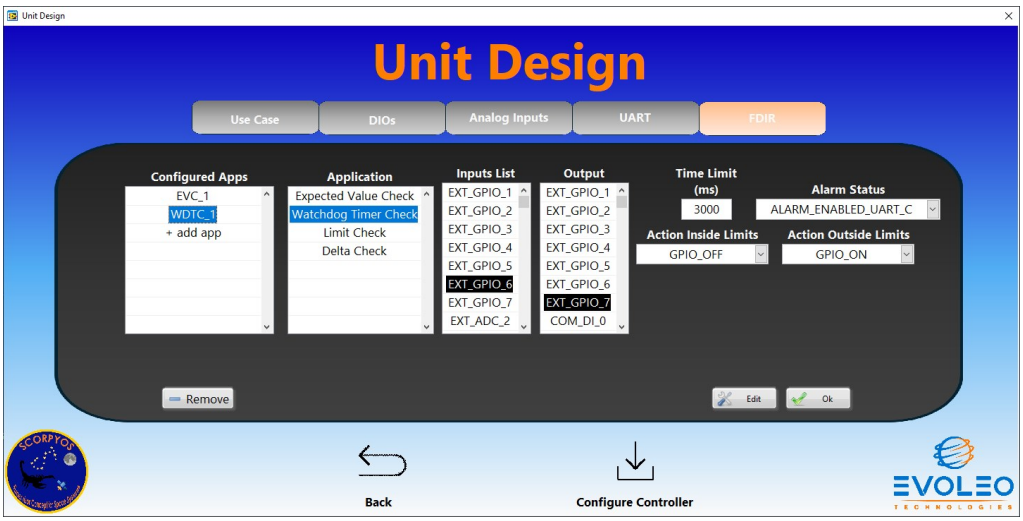

Figure 4-11 – Unit Design Menu

To configure the hardware, the following menus are available:

- DIOs (Digital Input/Outputs)
  - Here, you can configure the digital input and output pins. For each digital output pin, you can set the default state of the pin, determining whether it is in a high or low state.
- Analog Inputs
  - This menu allows you to select which analog inputs will be acquired by the system.
- UART (Universal Asynchronous Receiver-Transmitter)
  - In this menu, you can choose which UART communication channels will be active and configure their individual settings, such as baud rate and parity.

These comprehensive configuration options ensure that the system can be tailored to meet the specific needs of various use cases, providing flexibility and precision in its operation.

To configure the operational mode of the observability system, the FDIR (Fault Detection, Isolation, and Recovery) menu is available.

This menu allows for the configuration of up to 10 observability systems to operate simultaneously. Each of these systems can be configured as follows:

- Expected Value Check:
  - Define the list of inputs.
  - Configure the output pin.
  - Set the bit mask.
  - Configure alarm generation and the respective communication channel.
  - Define the actions to be taken when the system is within and outside the specified limits.
  - Set the expected value.
  - Link the bits to the inputs.

- Watchdog Timer Check:
  - Define the list of inputs.
  - Configure the output pin.
  - Set the time limit for the watchdog activation.
  - Configure alarm generation and the respective communication channel.
  - Define the actions to be taken when the system is within and outside the specified limits.
- Delta Check:
  - Specify the analog input.
  - Configure the output pin.
  - Set the acquisition period.
  - Configure alarm generation and the respective communication channel.
  - Define the actions to be taken when the system is within and outside the specified limits.
  - Set the minimum and maximum limit values.
- Limit Check:
  - Specify the analog input.
  - Configure the output pin.
  - Set the acquisition period.
  - Configure alarm generation and the respective communication channel.
  - Define the actions to be taken when the system is within and outside the specified limits.
  - Set the minimum and maximum limit values.

These comprehensive configuration options ensure that the system can be tailored to meet the specific needs of various use cases, providing flexibility and precision in its operation.

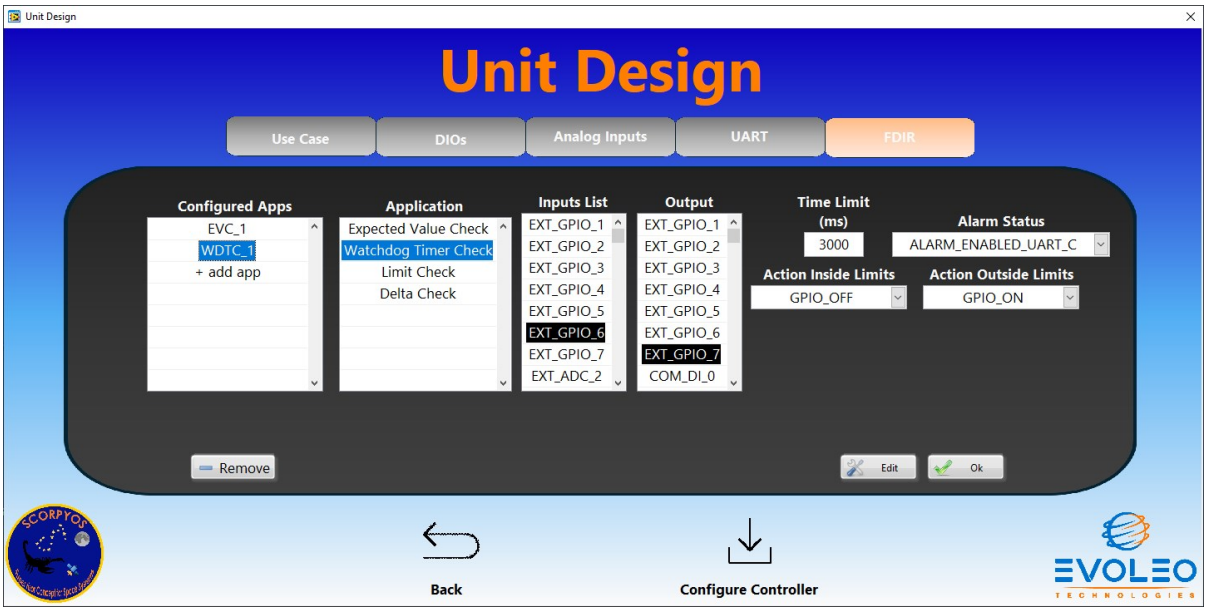

Figure 4-12 – FDIR options

The monitoring menu allows users to send either manual or automatic commands to the system, with the frequency of automatic commands defined by the user.

The implemented commands provide the following information:

- System Status Information, getting details such as the reset source, the watchdog reset counter, and the general reset counter.
- System Identification Information, getting Details such as the hardware and firmware version, serial number, and the project name.
- Datapool Status:
  - The status of all GPIOs (General-Purpose Input/Outputs).
  - The values of the analog inputs.
  - The status of each observability system (limit check, delta check, watchdog check, and expected value check).
  - The acquired analog and digital values through communication.

These comprehensive monitoring capabilities ensure users can efficiently oversee and manage system performance, ensuring optimal operation and quick identification of any issues.

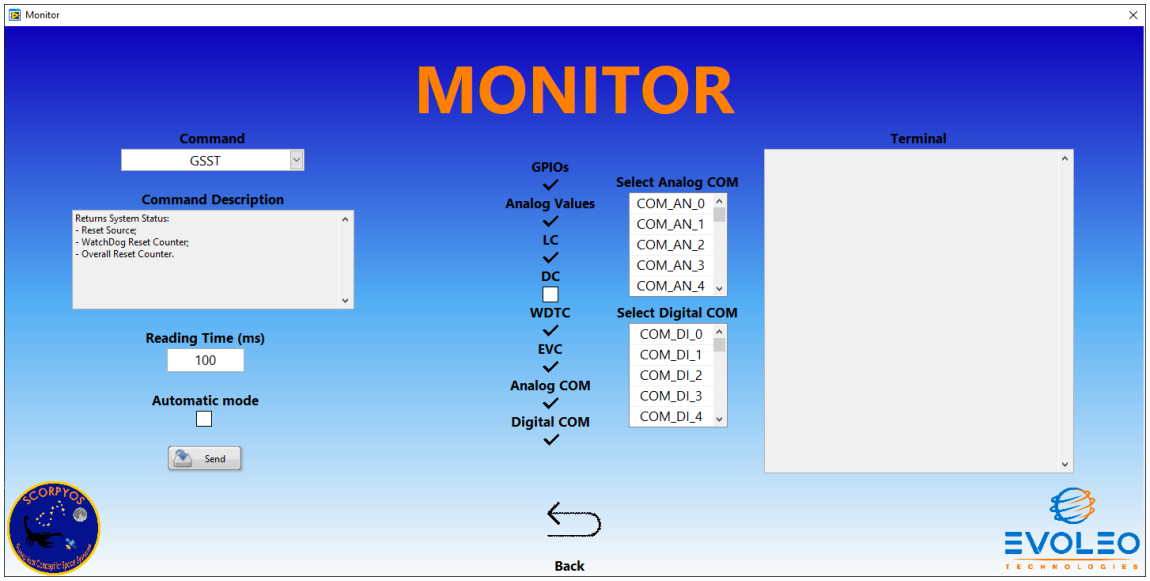

Figure 4-13 – Monitor menu

4.7 Send Data & receive Alerts Tool

To validate that the system can obtain data through communication and analyze it as if it were acquired through hardware, an application was also developed to inject simulated data.

This application enables the system to obtain and decode all alerts generated by SCORPYOS. With this software, it is possible to inject up to 160 digital states and 20 analog states into the SCORPYOS system.

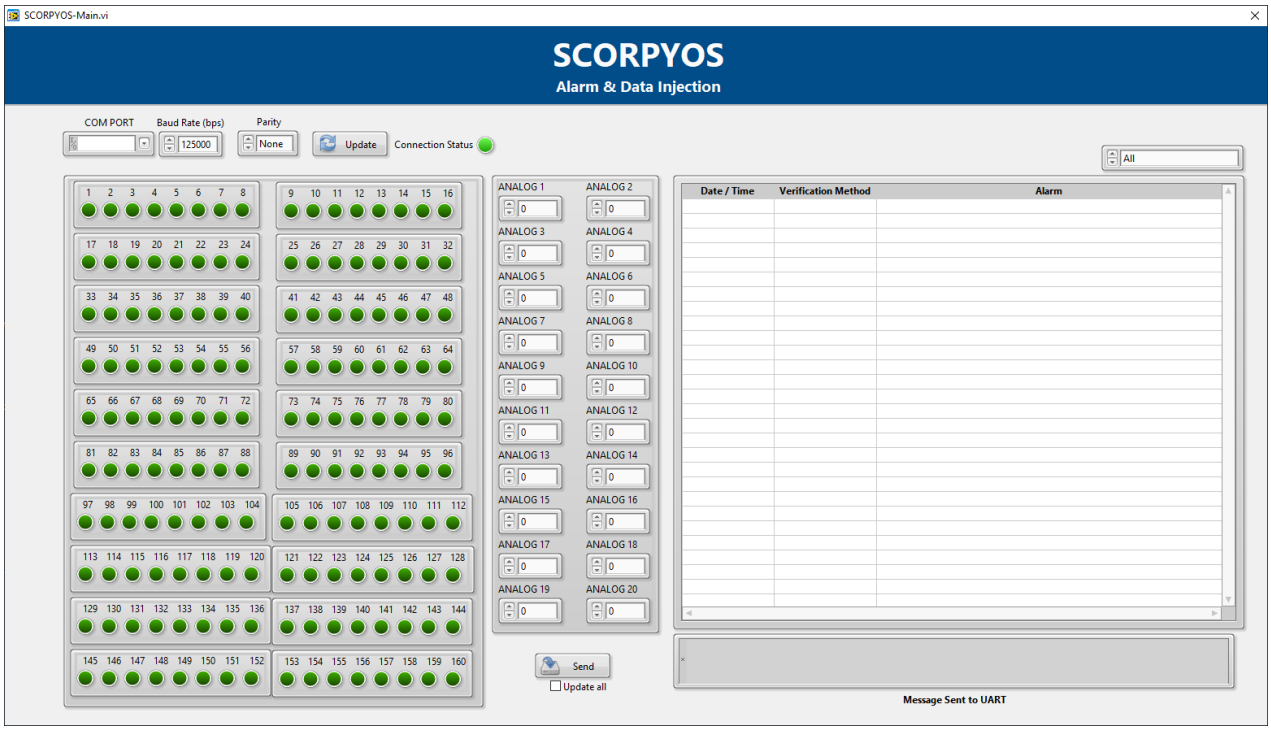

Figure 4-14 – Send Data & Receive Alerts Tool

The rest of the page has been left intentionally blank

5 Tests Performed

5.1 Simple use cases

Three fundamental use cases were defined and demonstrate the adaptability and effectiveness of SCORPYOS in different operational scenarios.

These use cases cover the supervision of digital, analog and communication systems, illustrating the ability of SCORPYOS to integrate with diverse architectures and respond to a wide range of monitoring requirements.

- The **first use case** demonstrates the application of SCORPYOS in the monitoring of digital inputs, using verification methods such as Expected Value Check and Watchdog Timer Check. It illustrates the system's ability to interpret logical states and respond to specific patterns or absence of "heartbeat" signals.
- In the **second scenario**, SCORPYOS is configured to monitor the analog signals, using methods such as Limit Check and Delta Check. This case highlights the system's ability to detect deviations from predefined values and significant variations between consecutive readings.
- The **third use case** focuses on SCORPYOS' ability to monitor and analyse data transmitted via communication interfaces, such as UART. It demonstrates the integration of message validation with content analysis, applying methods such as Data Field Check and Limit Check to received data.

In each of these cases, SCORPYOS demonstrates its flexibility through:

- Adaptive Configuration: Using the UART D interface, the system allows precise adjustments of operational parameters, adapting to the specific requirements of each application.
- Varied Verification Methods: Implementation of different verification techniques, suitable for each type of signal or data monitored.
- Automated Responses: Ability to generate digital outputs and alerts in response to detected anomalous conditions.
- Data Integration: Storage and processing of data in an internal data pool, allowing complex analyses and correlations between different inputs.

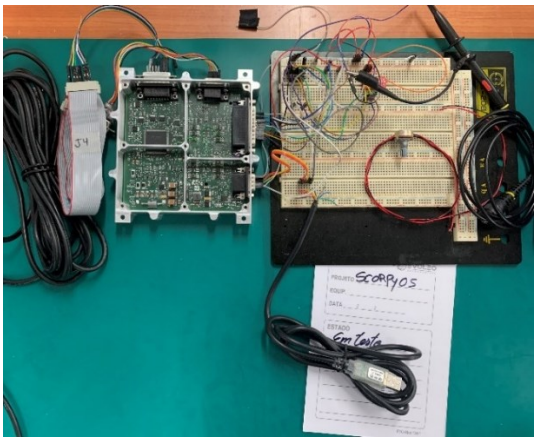

Figure 5-1 – Test Setup

These use cases not only demonstrate the technical versatility of SCORPYOS, but also its applicability in critical scenarios where accurate monitoring and rapid response are essential.

The SCORPYOS integrated test intends to test the system's main features, regarding:

- System's protection (Over Voltage and Under Voltage Protection)
- System GPIOs (Analog Inputs, Digital Inputs and Outputs)
- System communication ports
- System's Alarms

The tests were carried out in a dedicated laboratory at EVOLEO. The test was carried out on benches with ESD protection. There were no relevant test condition requirements.

5.1.1 Supervision of the digital system

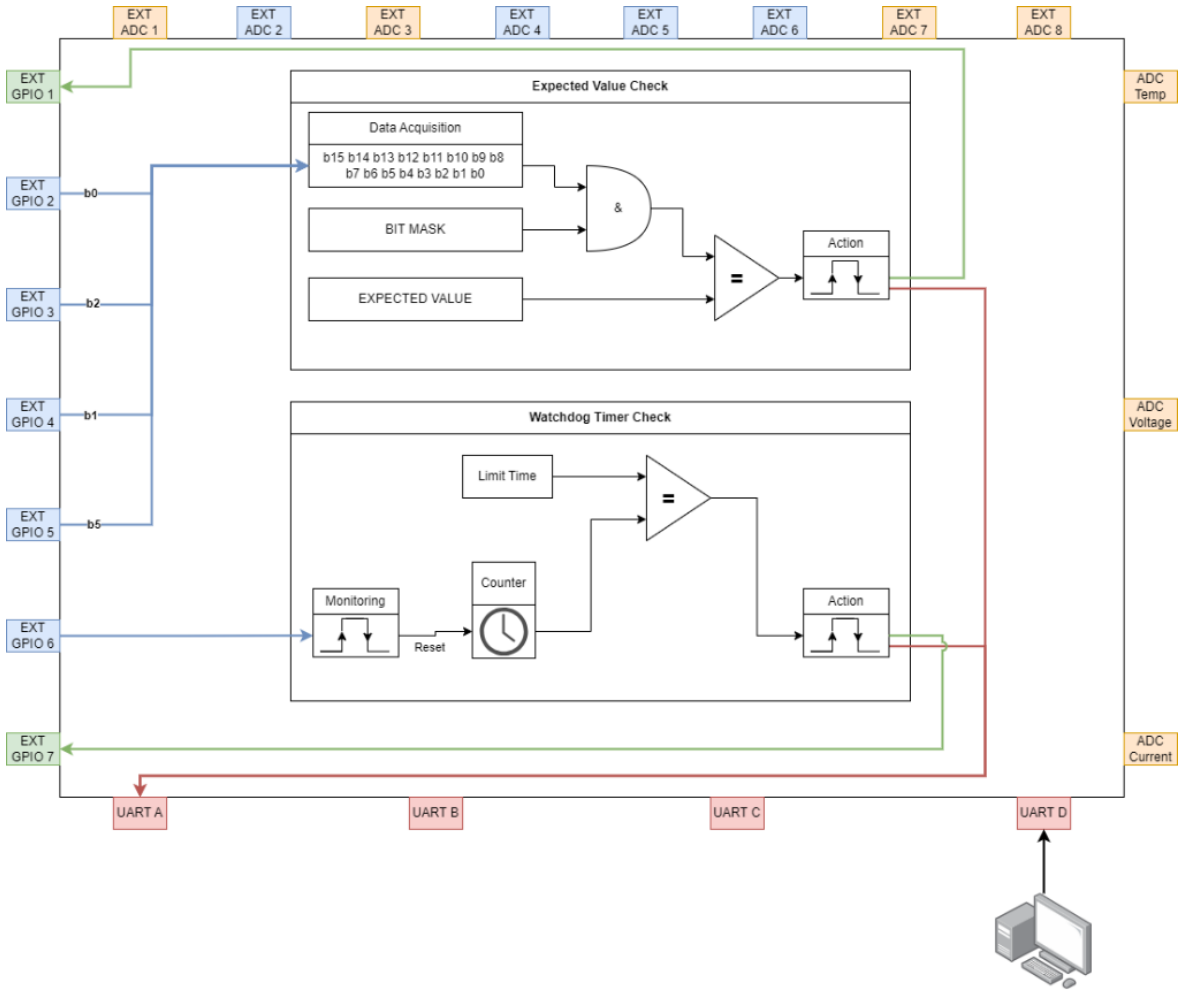

Figure 5-2 –Supervision of the digital system

This use case outlines the implementation of the SCORPYOS system as a supervisor of a digital system, employing verification methods based on digital inputs. Two primary verification methods will be implemented: Expected Value Check and Watchdog Timer Check.

5.1.2 Supervision of the analog system

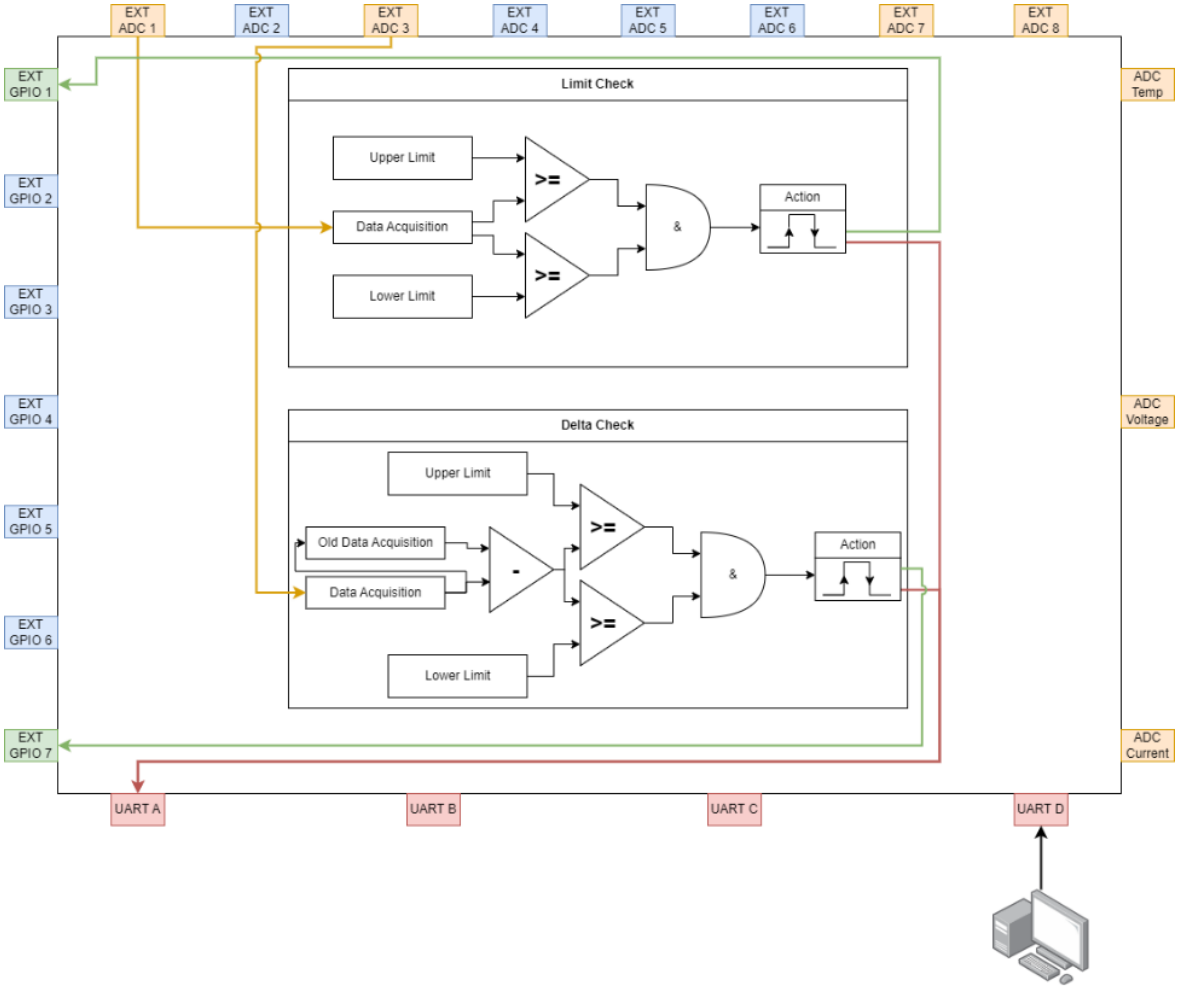

Figure 5-3 –Supervision of the analog system

This use case outlines the implementation of the SCORPYOS system as a supervisor of an analog system, employing verification methods based on analog inputs.

Two primary verification methods will be implemented: Limit Check and Delta Check.

5.1.3 Supervision of the communication system

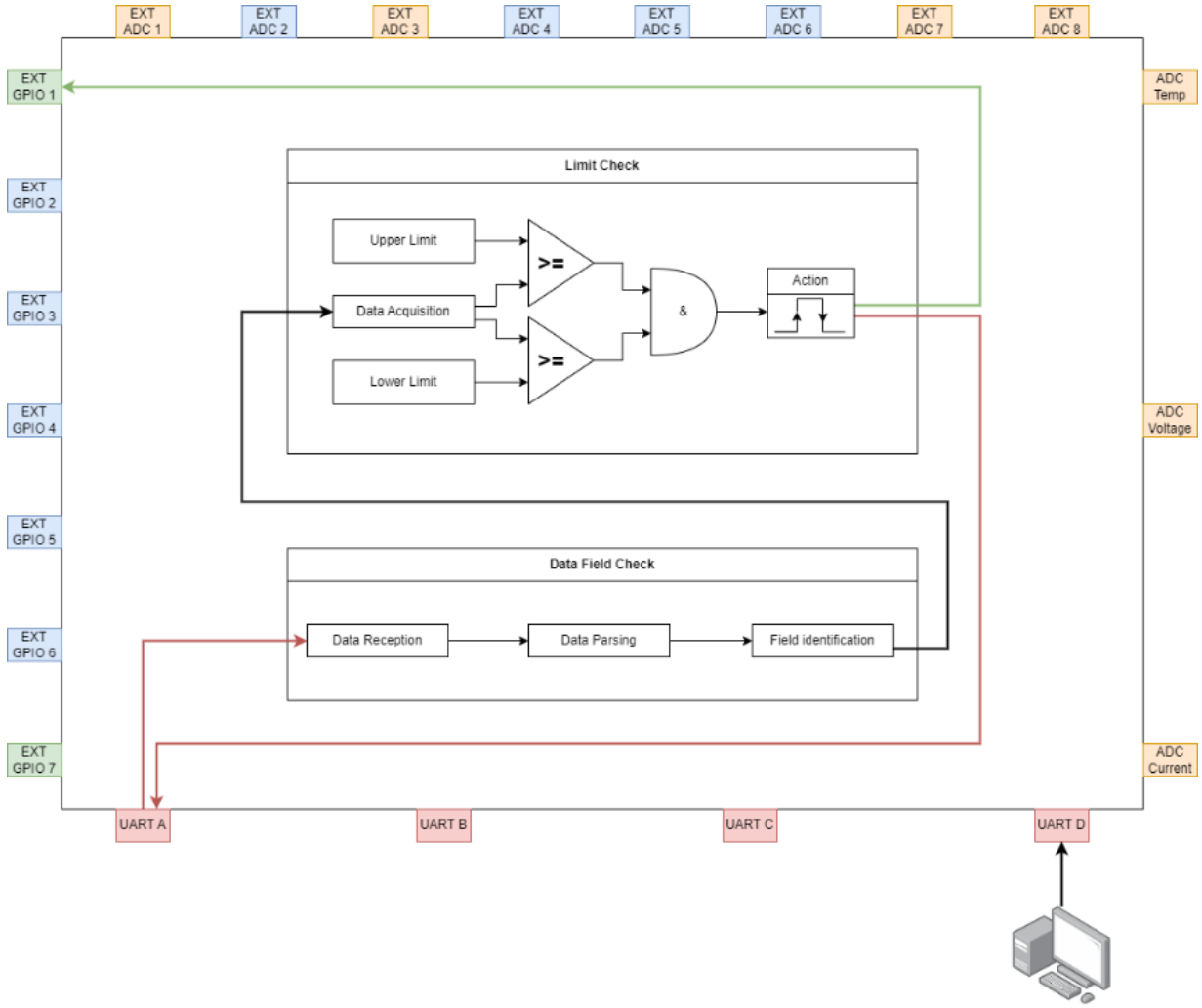

Figure 5-4 –Supervision of the communication system

This use case outlines the implementation of the SCORPYOS system as a supervisor of a communication system, using the UART interface as the primary data source.

The main focus will be the application of the Limit Check verification method on transmitted analog data, preceded by a message integrity validation.

5.2 Supervise the CHICS board and the DC/DC

In this test, a Processing Board (based on the Xilinx US+ MPSoC, also developed by EVOLEO) was used with SCORPYOS for being supervised.

Additionally, there was another independent system, a power supply, used to simulate a DC/DC converter.

SCORPYOS was responsible for detecting any issues within this system, actuating a GPIO to inform the system of any problems encountered.

This use case demonstrates the application of SCORPYOS monitoring the status of another system, and to do that, it will monitor the digital inputs, analog inputs and communication protocols, using verification methods such as Expected Value Check and Watchdog Timer Check, Limit Check and the Data Field Check.

It illustrates the system's ability to interpret logical states and respond to specific patterns or absence of "heartbeat" signals, the system's ability to detect deviations from predefined values and the SCORPYOS' ability to monitor and analyse data transmitted via communication interfaces, such as UART.

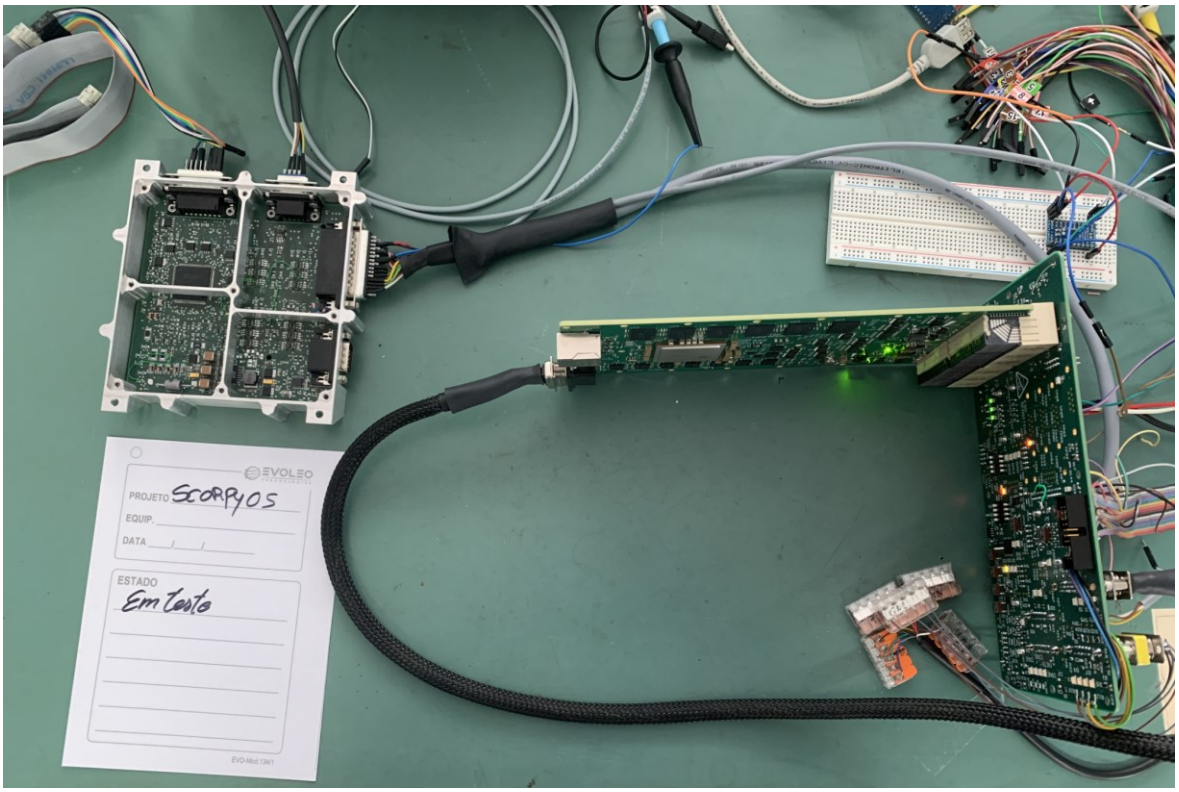

Figure 5-5 – Test Setup – Interface with CHICS

The SCORPYOS integrated test intends to test the system's main features, regarding:

- System GPIOs (Analog Inputs, Digital Inputs and Outputs)
- System communication ports
- System's Alarms

5.2.1 Supervision of the digital system

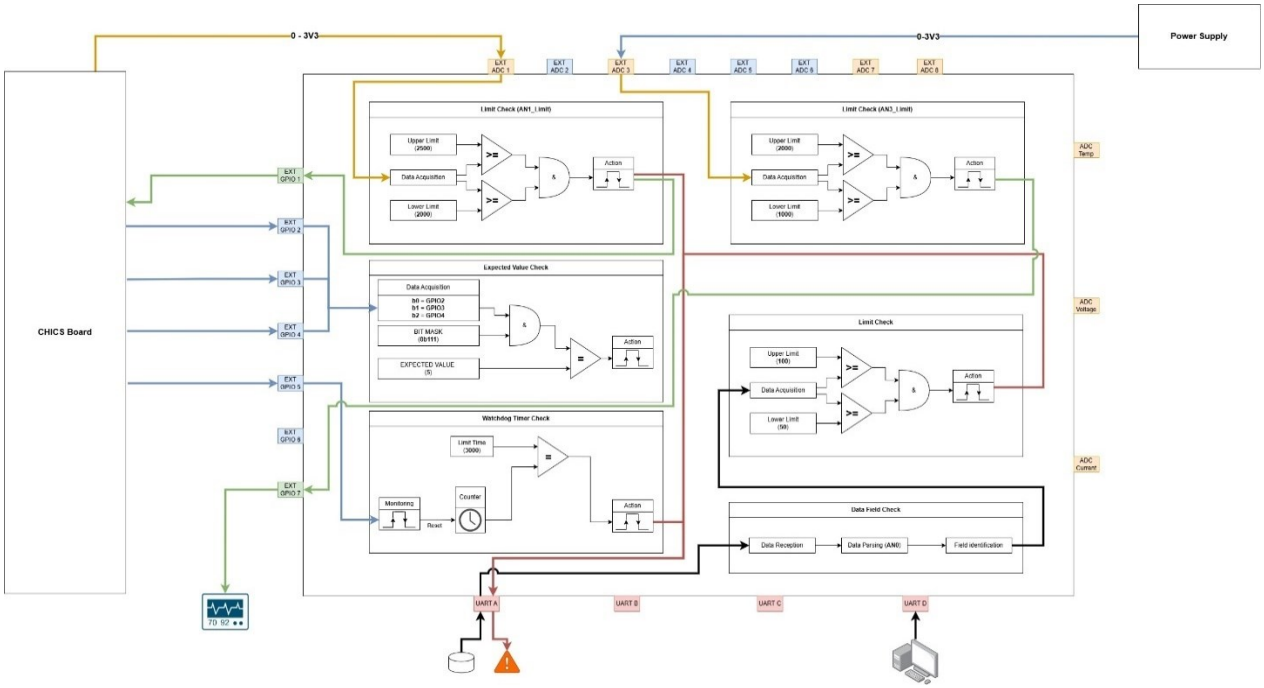

Figure 5-6 –Supervision of the digital system

The rest of the page has been left intentionally blank

6 Conclusion

Following the completion of the tests, it was confirmed that the system is fully functional and capable of analysing external issues, effectively communicating any detected problems to the system. The system accurately describes the nature of the issues through communication channels.

Looking ahead, there is potential for further enhancement of the system's configurability.

Future work could include the ability to define whether inputs are pull-up or pull-down, implement new observability features, enable logical operations, and interconnect multiple observability systems.

As this is a demonstrator system, future versions can be evolved according to emerging needs. Each module can be seamlessly integrated into the existing system, allowing for easy expansion and continuous improvement.

END OF DOCUMENT