

# André Louisa Marques

SOFTWARE ENGINEER

Portugal

🏠 [andrelousamarques.com/contact](https://andrelousamarques.com/contact)

## Experience

---

### Thales Edisoft Portugal

SOFTWARE COMPONENT ENGINEER

*Oeiras - Lisbon, Portugal*

*September 2015 - Today*

#### Project "MUSHER: Manned Unmanned System for Helicopter"

*June 2024 - Present*

The "Manned Unmanned System for Helicopter" (MUSHER) project aims at improving the European capacity of operating both manned and unmanned aerial platforms in a multienvironment (civil, military or mixed). The main goal of the project is to design a scalable European Manned Unmanned Teaming (e-MUMT) system, allowing manned platforms (Helicopter) and Unmanned Aerial Vehicle (UAV) from European forces to interoperate operational scenarios selected from NATO studies and deliver the maximum capacity for the operational mission with high level interoperability and UAV autonomy, crew load reduction, and improved safety. Developed as a consortium, our team was responsible for the mission debriefing station, which ingests mission data and allows it to be replayed for analysis. It also provides KPI reporting.

#### MY CONTRIBUTIONS

- Responsible for the development of the frontend (mainly a map where the mission actors and data is displayed over a mission timeline, and KPI information display) and backend of the debriefing station

#### TECHNOLOGIES

- Typescript, StencilJs, Python, OpenLayers, Grafana, MySQL

#### Project "Triton-X"

*July 2021 - Present*

Project for a Luxembourg client (Luxspace) to develop several workpackages for the Luxspace Triton-X microsatellite platform.

#### MY CONTRIBUTIONS

- Technical lead of 2+ developers for 3 work packages (RTE (Remote Terminal) Analog (responsible for the Thermistors and Sun Sensors), RTE Driver (responsible for the Magnetorquers, Heaters and the Hold-Down and Release Mechanisms and PCU (Power Conversion Unit)
- Requirements gathering and refinement, architecture design, code review, software development
- Performed the first RTE SW commissioning (preliminary SW version that exercised the HW such that the client could validate the HW) at the client premises. SW was developed prior to the commissioning without ever being tested in the HW before hand. The team only had access to a HW board that had a similar processor
- Performed the commissioning of the EGSE Battery Conditioning Unit (BCU) software
- Responsible for the maintenance and support of the EGSE workpackage

#### TECHNOLOGIES

- C, FreeRTOS, IBM Doors, MS Visio, MagicDraw, RUCM, Pclint, Java

Project for the European Space Astronomy Center (ESAC) to create a web application ([datalabs.esa.int](http://datalabs.esa.int)) where scientific users can have access to data from space missions and to develop and run their applications close to satellite mission data, instead of having to downloading TBs of data to their local machines and setup their local environment to work on the data.

**MY CONTRIBUTIONS**

- Technical lead of 1+ developer at EDISOFT, and responsible for following up the work of the consortium members and parallel projects
- Requirements gathering, architecture design and development of system components
- Technical documentation, such as design, interface, deployment, etc
- Creation of PCR (Product Change Request) and analysis of problem reports raised by users
- Responsible for code reviews and merges with the master branch
- Responsible for maintaining the development environment (in coordination with the IT department)
- Responsible for the approval of the system at the development environment and for deploying it to production (in coordination with client's IT)
- Responsible for coordinating the integration of two parallel projects developed by two independent teams (from France/Luxembourg and Estonia)
- Presented hands-on-session for users at ESAC on how to use the system
- Configured Jenkins for automated testing (including code coverage) using pytest and selenium. Integration with ESAC's Jira to run tests and capture reports

**TECHNOLOGIES**

- Python, Kubernetes, Rancher, Docker, Nginx, Typescript, Selenium, Jira, Drone

**Project "RTEMS AGGA-4"**

November 2021 - November 2022

Project for a Romanian client (Deimos Romania) to develop a Space pre-qualified RTEMS BSP for the AGGA-4 (ATMEL AT7991) platform.

**MY CONTRIBUTIONS**

- Technical co-lead of two developers
- Responsible for the coordination and support of the testing efforts (testing infrastructure setup, test creation)

**TECHNOLOGIES**

- C, Perl, RTEMS, bash, Linux

**Project "RTEMS GCC Patch"**

November 2019 - September 2020

Upgrade of the RTEMS by EDISOFT toolchain to apply the Gaisler TN-0009 technical note workaround for a detected problem in the Gaisler GR712 hardware (LEON3FT Stale Cache Entry After Store with Data Tag Parity Error)

**MY CONTRIBUTIONS**

- Context: Previous versions of RTEMS used a physical Debian 5 machine for building and testing. For this project we didn't have this machine or a physical machine where we could install Debian 5 and the debugger interface hardware required by the test boards hardware was updated (from GRMON to GRMON2). This new version of the debugger interface was not compatible with Debian 5
- Designed and implemented a test environment based on a physical host running Debian 10 to which the test board's hardware is connected, and two Virtual Machines:
  - Debian 5 – where RTEMS is built and the test suite is executed
  - Debian 10 – where the tests are deployed into the hardware
- The test operator machine is the Debian 5 VM; On this machine the RTEMS test suite (based on perl scripts) was kept as original as possible while on the system itself all calls to GRMON and the test boards hardware are scripted to seem like they are available on the Debian 5 VM, but in practice there is bash scripting routing the data around the machines and the test hardware via SSH. The GRMON2 is unstable at times so a monitor was designed to restart the test if a problem was detected, so the testsuite could keep running at all times.
- Updated unit/integration/validation test automation based on Perl scripts to account for test infrastructure changes.
- Produced technical documentation (test plans, verification reports, user manuals, etc)

**TECHNOLOGIES**

- Perl, SSH, GRMON, Bash, Linux, VirtualBox

## Project "RTEMS ARM R5"

November 2019 - September 2020

Port of the RTEMS by EDISOFT Operating System to the ARM architecture, namely the ARM R5 radiation tolerant processor on a Zynq Ultrascale+ board and development of an AMP version capable of using the board's two R5 cores

### MY CONTRIBUTIONS

- Developed TCL scripts to interact with the Xilinx SDK to deploy the test into the Zynq Ultrascale+ board hardware
- Updated test automation Perl scripts to deploy tests to the Zynq Ultrascale+ board hardware and capture the reports via Uart
- Contributed to the RTEMS port effort from SPARC to ARM

### TECHNOLOGIES

- Perl, Embedded, TCL, Xilinx SDK, Zynq, SPARC, ARM

## Project "PASSARO Clean Skies"

August 2018 - May 2019

Airbus needed a prototype for a system that would aid the decision process for engineers building the planes in the assembly line with debugging and tracing of integration problems, by predicting the best troubleshooting among a set of possibilities when an error or incidence is given and to store the correctness of the decisions to learn for the future.

### MY CONTRIBUTIONS

- Development of the backend for the initial prototype, which loads the aircraft model into graphs, captures sensor and test data and provides a REST API to interact with the aircraft models allowing the troubleshooting process

### TECHNOLOGIES

- Java, Spring, OrientDB, ArangoDB, Neo4J

## Project "NEOSAT FOP"

November 2017 - March 2019

EDISOFT was requested to develop several evolutions for the Thales Alenia Space (TAS) OPENSCC product, in the frame of the NEOSAT programme.

### MY CONTRIBUTIONS

- Responsible for the following PUS (Packet Utilization Standard) service implementations: 1 (Telecommand Verification Service - with UI), 5 (Event Reporting Service), 11 (On-board Operations Scheduling Service - with UI), 12 (On-board Monitoring Service - with UI). I also supported the development and testing of PUS 6 (Memory Management Service) and 13 (Large Data Transfer Service).
- The UIs above were developed in typescript and angular.JS using components from the TAS Mainframe library (similar to the Twitter Bootstrap framework)
- Adapted legacy router component in charge of telemetry reception and packetization to use the new PUS services
- Developed UI end-to-end tests with protractor and unit and validation tests in Junit
- Automated application deployment and test execution with Maven, Ant, Batch and Shell scripting
- Configured Jenkins to gather quality reports (SonarQube)
- Configured Clover to gather test coverage data

### TECHNOLOGIES

- Java, Spring, Typescript, Jenkins, Maven, Ant, Shell

## Project "RTEMS OHB"

August 2017 - January 2018

In the frame of creating a common SCSW platform as the basis for future OHB satellite projects, OHB requested EDISOFT to unify all different RTEMS versions in a dedicated version specific for OHB satellites. The RTEMS OHB project aimed to produce the dedicated Space-Qualified version including modifications to the semaphore priority inheritance algorithm, addition of CPU usage statistics, integration of the RUAG COLE LEON2 BSP, update of the development toolchain (GCC and BINUTILS) and update of the coverage analysis tool (GCOV).

### MY CONTRIBUTIONS

- Update of the development toolchain (GCC and BINUTILS), including the update of the RTEMS testsuite coverage infrastructure to gather coverage data from GCOV at the test hardware and compile coverage reports on a host machine (GCC 4.2 to GCC 7.2). This was tricky because the GCOV API within the GCC is not a stable API as it is not meant to be used by external tools, however this is required for the embedded systems under test as they do not have a filesystem

### TECHNOLOGIES

- C, GCC, GCOV

## Project "E-Sail EGSE"

March 2016 - February 2019

The aim of the E-SAIL satellite mission is to detect AIS messages in order to track vessels beyond the coverage already provided by the existing terrestrial AIS. This global maritime surveillance capability is beneficial to many European entities to counteract illegal activities and to improve maritime safety. EDISOFT was assigned to develop an EGSE, including all the software and hardware components, to test the E-SAIL Satellite at subsystem and system level. The activities included the sub-systems integration (e.g.: CORTEX CRT QUANTUM, DVB-S2 Modem, UART, USART, DIO, Up/Down Converters, CAN units, SpaceWire Units), sub-systems development (e.g.: Cortex Driver, UART, USART, DIO, SPI, DVB, TCP-IP, CAN, Power Meter, Power Supply, RF Switch, Attenuator unit) and operation of ESA SCOS-2000. This project followed the ESA ECSS development standards E40 and Q80.

### MY CONTRIBUTIONS

- Responsible for the development of drivers for several EGSE front ends (Special Check Out Equipment (SCOE)), such as UART, USART, SPI, DIO, Digital Multimeter and S-Band Up and Down converters.
- Produced a library to interface the EGSE equipment with the ESA SCOS-2000 software used to test the E-SAIL satellite.
- Developed validation tests in TCL for the developed SCOE drivers and the required TC/TM data on the SCOS MIB database.
- Contributed to the project requirements and documentation (validation test plan, user manual, maintenance manual, verification report, interface control document, detailed design document, software requirements specification).
- Used Jenkins to gather quality reports (SonarQube).
- Developed scripts based on Ant and bash to automate application deployment, testing and coverage gathering (JaCoCo).
- Developed scripts to parse hardware user manuals to generate Java code that would handle hardware configuration.
- Developed scripts to operate the EGSE (ant and bash).

### TECHNOLOGIES

- Java, C, JNI, TCL, ant, bash

## Project "RTEMS ARM"

September 2016 - May 2017

The RTEMS ARM project was requested by Thales and intended to qualify, according to GSWS DAL-B level, RTEMS by EDISOFT on an ARM Cortex-M7 target platform. The project required a software port of RTEMS by EDISOFT to the ARM architecture, namely the Atmel SAM V71 Xplained Ultra Evaluation Kit board. EDISOFT was tasked to update the toolchain (GNU ld, as, gcc) from SPARC to ARM, update the kernel to run on the ARM CPU, add support for peripherals (two UARTs) and update the testsuite.

### MY CONTRIBUTIONS

- Supported the port activities
- Developed an application to run the tests via the ATMEL ATPROGRAM application and gather the reports via an UART port

### TECHNOLOGIES

- Embedded, ARM, C

The RTEMS Qualification Extensions project aimed at the development of a space pre-qualification (DAL-B) of SpaceWire (SpW) and MIL-STD-1553 (1553) drivers for the RTEMS by EDISOFT version. The produced RTEMS version was targeting the Gaisler GR712RC board. Additionally, a Monitoring tool was developed. The purpose of the RTEMS Monitoring tool was to support the application developer, giving information collected from the runtime execution of the application (RTEMS API calls, hardware interrupts, thread stack monitoring and configuration monitoring). The full qualification activities were developed, including software planning, software specification, software design, software implementation, software integration, software validation, software acceptance, operations and maintenance and all the verification activities. "Qualified" means that:

1. An interface specification exists for each qualified API element.
2. Each qualified interface has a functional specification.
3. The function is validated by a test.
4. You get 100% branch and statement coverage from the validation tests. For some justified exceptions you can use unit tests.

#### MY CONTRIBUTIONS

- Contributed to the development of the monitoring tool by adding support to SpaceWire bus to transfer RTEMS system events from the target running RTEMS to a host computer.
- Contributed to the RTEMS by EDISOFT SpaceWire driver by adding support for Time Codes and developing the required tests.
- Contributed to the project documentation (user manual, validation test plan, verification report) and verification activities (galileo and Misra-C standards).
- Supported the acceptance activities for the SpaceWire driver and Monitoring Tool.

#### TECHNOLOGIES

- Embedded, SpaceWire, C, Java

## Education

---

### University of Beira Interior

*Covilhã, Portugal*

#### MASTER'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

*September 2014 - December 2016*

- Developed Monitoring Tool for the RTEMS by Edisoft operating system concerning multicore systems and system execution monitoring while working at Edisoft, as my Master's degree dissertation
- Implemented security protocols and algorithms using Java and Python, such as AES (CBC and CTR modes based on the NIST standard) and the Schnorr signature algorithm
- Attended cryptography class similar to Dan Bonet's Cryptography I coursera course, where I also presented a report on the algorithm submissions for the CAESAR competition for authenticated encryption
- Implemented a secure keyring application. Documented attack vectors and scenarios, identified security properties
- Developed a software engineering project with large Portuguese Telecommunications company (Portugal Telecom), where we were required to perform penetration testing on a virtual machine to find vulnerabilities, document (classify with CVE number, OWASP top 10 security risks, develop attack models) and then exploit them
- Developed application based on Android and AWS to capture seismic data from users (Mercalli score) and show the aggregated data on a map available online
- Researched and presented a state of the art on Elastic Resource Scaling with Virtual Machines in the context of Cloud Computing
- AVERAGE: 17.7 out of 20

### University of Beira Interior

*Covilhã, Portugal*

#### BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

*September 2010 - July 2014*

- RTEMS - Developed device driver for the SD card interface on the Raspberry Pi under the RTEMS operating system, as my final course project
- Summary of the final project output: <https://lists.rtems.org/pipermail/devel/2014-August/007842.html>
- AVERAGE: 15.2 out of 20

## Skills

---

<b>Programming (professional use)</b>	C, Java (JNI, Swing, Spring, JMS), Python, Perl, Typescript, Javascript
<b>Programming (other)</b>	GO, C++, C#, Android, PHP, Ocaml, Assembly (x86)
<b>Scripting &amp; other</b>	Bash, Sed, Awk, Windows Batch, TCL, SQL, markup (XML, XSD, JSON, YAML)
<b>DevOps</b>	Docker, Kubernetes, Jenkins, Jira, Bugzilla, AWS, Gitlab CI
<b>Software Tools</b>	GNU tools (binutils, gcc), Ant, Maven, Makefile, VirtualBox, VmWare, LaTeX, OpenLDAP
<b>Testing tools and libraries</b>	JaCoCo, Clover, GCOV, pytest, SonarQube, Junit, PowerMockito, Protractor, Selenium
<b>Version Control</b>	Git, Subversion
<b>Software Standards</b>	Misra-C, Galileo Software Standards, ECSS-E-ST-40C and ECSS-Q-ST-80C
<b>Embedded Platforms (SPARC)</b>	LEON3 (Gaisler GR712RC)
<b>Embedded Platforms (ARM)</b>	Atmel ATSAMV71Q21 (ARM Cortex M7), Raspberry Pi (ARM1176JZF-S), Zynq Ultrascale+ (ARM R5)
<b>Embedded Peripherals</b>	UART, USART, SPI, I2C, GPIO, SPW, CAN
<b>Database</b>	Postgres, MySQL, SQL Server, SQLite, Redis, H2, MongoDB, Neo4J, OrientDB, ArangoDB
<b>Operating Systems</b>	Linux (Gentoo/Debian/Fedora/CentOS/Arch/OpenSuse), RTEMS, Windows
<b>Development Software</b>	Eclipse, Netbeans, Visual Studio, VIM
<b>Back-end</b>	Node.js, REST, GraphQL
<b>Front-end</b>	Javascript, Typescript, StencilJs, AngularJs, HTML5
<b>Electronics</b>	Through-hole soldering, Oscilloscope
<b>Languages</b>	Portuguese, English

## Open Source

---

### RTEMS

GOOGLE SUMMER OF CODE

*As Mentor*

*May - August 2016*

- Co-mentored two students on Raspberry Pi projects and produced the mid term evaluation for one of them
- Student Project 1: <https://summerofcode.withgoogle.com/archive/2016/projects/6397834018422784>
- Student Project 2: <https://summerofcode.withgoogle.com/archive/2016/projects/5992398500921344>

### RTEMS

GOOGLE SUMMER OF CODE

*As Student*

*May - August 2015*

- Expanded the GPIO work done in my previous GSOC participation for the Raspberry Pi into a generic/system wide API usable by all RTEMS supported hardware platforms. Also ported the previous Raspberry Pi I2C work to a newer Linux-based i2c API, and worked with the Raspberry Pi SD card support using the FreeBSD SD/MMC stack
- Blog: <https://asuolgsoc2014.wordpress.com/>
- Final report: [https://web.archive.org/web/20170714002917/https://devel.rtems.org/wiki/GSoC/2015/Final\\_Report#AndreMarques:RaspberryPiLowLevelPeripheralsandSDCard](https://web.archive.org/web/20170714002917/https://devel.rtems.org/wiki/GSoC/2015/Final_Report#AndreMarques:RaspberryPiLowLevelPeripheralsandSDCard)
- GPIO API: <https://gitlab.rtems.org/rtems/rtos/rtems/-/blob/main/bsps/shared/dev/gpio/gpio-support.c>
- RPI GPIO: <https://gitlab.rtems.org/rtems/rtos/rtems/-/tree/main/bsps/arm/raspberrypi/gpio>
- I2C driver: <https://gitlab.rtems.org/rtems/rtos/rtems/-/tree/main/bsps/arm/raspberrypi/i2c>
- SPI driver: <https://gitlab.rtems.org/rtems/rtos/rtems/-/tree/main/bsps/arm/raspberrypi/spi>

### RTEMS

GOOGLE SUMMER OF CODE

*As Student*

*May - August 2014*

- Added support for the GPIO interface, as well as I2C and SPI data buses on the Raspberry Pi hardware for the RTEMS operating system
- Blog: <https://asuolgsoc2014.wordpress.com/>
- Final report: [https://www.rtems.org/gsoc2014\\_final\\_report](https://www.rtems.org/gsoc2014_final_report)
- Final report (wiki): [https://web.archive.org/web/20221201105705/https://devel.rtems.org/wiki/GSoC/2013/Raspberry\\_Pi\\_BSP\\_Peripherals](https://web.archive.org/web/20221201105705/https://devel.rtems.org/wiki/GSoC/2013/Raspberry_Pi_BSP_Peripherals)

### RTEMS

CONTRIBUTOR

*February 2014*

- Contributed with unit tests for the RTEMS operating system
- Unit tests: <https://gitlab.rtems.org/rtems/rtos/rtems/-/blob/main/testsuites/fstests/fsrename/test.c>