CURRICULUM VITAE

Alberto Petrucci

Embedded Linux Engineer & Dottorando

Progetto software embedded mission-critical e studio metodi di AI verificabile per sistemi spaziali, unendo Linux low-level, avionica, co-design hardware-software e rigore accademico.

Embedded Linux | Software spaziale | AI affidabile | RISC-V / ARM / SPARC | Yocto & BSP | ECSS

CONTATTI

petrucci.alberto@gmail.com
capriele.bitbucket.io
linkedin.com/in/alberto-petrucci
L'Aquila, Italia · Italiana

01 PROFILO

Sono un **Ingegnere dei Sistemi Embedded e Dottorando**, specializzato in **Intelligenza Artificiale Affidabile per il settore aerospaziale**. Il mio percorso nasce dal desiderio di colmare il divario tra ricerca accademica rigorosa e applicazioni industriali mission-critical. Attualmente ricopro un duplice ruolo: progetto software avionico presso **Thales Alenia Space** e conduco la mia ricerca di dottorato al **Gran Sasso Science Institute (GSSI)**.

La mia formazione presso l'Università degli Studi dell'Aquila mi ha fornito solide basi in Ingegneria Automatica, che ho progressivamente orientato verso il **co-design hardware-software per sistemi spaziali**. Diversamente dallo sviluppo software tradizionale, il mio lavoro affronta i severi vincoli di dimensioni, peso e potenza (SWaP) delle piattaforme orbitali. Mi dedico a colmare l'"Edge-Computing Gap" nel settore aerospaziale: rendere possibile l'esecuzione sicura a bordo dei satelliti di modelli di AI complessi e non deterministici, senza dipendere dal controllo di terra.

I miei contributi di ricerca includono il **framework Aerosafe**, che adatta la safety assurance dei sistemi di machine learning agli standard ECSS, e l'**architettura SafeKAN**, un approccio accelerato in hardware basato sulle Kolmogorov-Arnold Networks per l'anomaly detection spiegabile. Il mio obiettivo è contribuire alla prossima generazione di **sistemi multi-satellite** attraverso autonomia distribuita e intelligenza di bordo verificabile.

Desidero portare questa esperienza industriale e questo rigore di ricerca nel mondo accademico, formando una nuova generazione di ingegneri consapevoli che un algoritmo di controllo vale quanto l'hardware su cui vola.

Prevedo di completare il dottorato entro la fine di agosto 2026.

02 ESPERIENZA PROFESSIONALE

02/2022 - Presente | L'Aquila, Italy

System Software Responsible / Embedded Software Engineer

Thales Alenia Space

Strumenti utilizzati

- IBM Doors
- Eclipse
- Visual Studio Code
- GNAT Studio / GRMON / Codeo

Linguaggi, protocolli e tecniche utilizzati

- Agile
- CI/CD
- C/C++
- Python
- Ada

Nel mio ruolo di Responsabile Software di Sistema e Ingegnere Software Embedded, ho contribuito a numerosi progetti di alto profilo, ognuno dei quali ha richiesto competenze specializzate e una comprensione approfondita dei sistemi embedded e dello sviluppo software per missioni spaziali. Di seguito una panoramica delle mie principali responsabilità e contributi:

PLATINO 1 / PLATINO 2

Come Ingegnere Software Embedded per le missioni PLATINO, mi sono occupato dell'implementazione di nuove funzionalità basate su requisiti dettagliati definiti nel baseline. Questo ha comportato una stretta collaborazione con il team di sistema per valutare e progettare soluzioni che sfruttassero efficacemente l'heritage esistente ottimizzando l'efficienza dell'implementazione. Mi sono inoltre occupato della gestione della configurazione del satellite, assicurando che gli aggiornamenti venissero accuratamente registrati nel DataBank del progetto per mantenere la tracciabilità e l'allineamento con i requisiti di sistema.
Inoltre, ho sviluppato procedure di test dettagliate per verificare e convalidare le nuove funzionalità, garantendo che fossero conformi ai requisiti della missione. Un altro aspetto critico del mio ruolo è stato affrontare i Software Problem Report (SPR), che includevano la replica dei problemi, l'identificazione delle cause principali, l'implementazione delle soluzioni e la convalida della loro risoluzione. Per supportare l'esecuzione del progetto, ho anche gestito l'ambiente di sviluppo, garantendo una piattaforma stabile e robusta per il team.

GALILEO 2nd Generation

Nel progetto GALILEO 2nd Generation, ho ricoperto il ruolo di Responsabile Software di Sistema. Il mio obiettivo principale era definire i requisiti software essenziali per il raggiungimento degli obiettivi generali del sistema. Questo ha richiesto una profonda comprensione sia delle esigenze tecniche che operative della missione. Ho lavorato per garantire che tutti i requisiti fossero correttamente implementati, supervisionando il processo di sviluppo per mantenere l'allineamento con gli obiettivi del progetto.
La supervisione dell'esecuzione complessiva del progetto è stata una parte fondamentale delle mie responsabilità, che includeva il monitoraggio dei progressi, la gestione dei rischi e la garanzia del rispetto degli standard qualitativi in ogni fase. Inoltre, ho coordinato le attività delegate a società esterne, assicurandomi che fossero conformi ai regolamenti dell'ESA (European Space Agency). Questo ha incluso la supervisione dei processi di Independent Software Verification and Validation (ISVV), un passo cruciale per garantire l'affidabilità e la sicurezza del software per le operazioni di missione.

Costellazione Iride NIMBUS

Come Responsabile della Gestione dei Dati per la costellazione Iride NIMBUS, ho gestito la documentazione di sistema, concentrandomi in particolare sul Baseline dei Requisiti Software Avionici e sulla Specifica PUS (Packet Utilization Standard). Questi documenti erano fondamentali per garantire che il software soddisfacesse gli standard tecnici e operativi richiesti per la missione.
Oltre alla documentazione, ho svolto un ruolo chiave nella gestione del budget dati del sistema, che ha comportato il controllo dei protocolli e delle interfacce di comunicazione. In particolare, ho gestito il flusso di dati per i bus CAN e SpaceWire, garantendo uno scambio di dati efficiente e affidabile in tutto il sistema. Inoltre, ho gestito il budget di comunicazione satellite-terra sia per le bande X che S, assicurandomi un utilizzo ottimale dei canali di comunicazione e mantenendo l'allineamento con i vincoli della missione.
La piattaforma NIMBUS è strettamente legata alla ricerca, essendo stata sviluppata in collaborazione con il GSSI (Gran Sasso Science Institute). Servirà come base per il satellite OASIS e sarà tra i primi micro-satelliti utilizzati per la ricerca scientifica sulle particelle spaziali.

Nel corso della mia carriera in questi progetti, ho sviluppato una forte competenza nei sistemi embedded, nella gestione dei requisiti software e nelle operazioni delle missioni spaziali, dimostrando costantemente la capacità di fornire risultati di alta qualità in ambienti sfidanti e innovativi.

01/2024 - 05/2024 | L'Aquila, Italy

Senior Django & Flutter developer

Strumenti utilizzati

- Visual Studio Code

Linguaggi, protocolli e tecniche utilizzati

- Python & Django
- Dart & Flutter
- Kotlin
- Swift
- Javascript
- MySQL
- REST Api
- Git

Ho sviluppato un gestionale multipiattaforma in grado di offrire tutte le esigenze richieste dal cliente. Le funzionalità implementate sono:

- Gestione utenti: Creazione, modifica, eliminazione e gestione dei permessi degli utenti.
- Gestione dei corsi: Monitoraggio della partecipazione, organizzazione dei corsi e implementazione differenti modalità di visualizzazione.
- Gestione dei materiali: Controllo dell'inventario, tracciamento dell'utilizzo e gestione della distribuzione dei materiali.
- Gestione dell'hardware e dei software informatici: Catalogazione e monitoraggio delle risorse IT, comprese manutenzioni e aggiornamenti.

Per ognuno di questi punti lo sviluppo è stato suddiviso in due parti:

- **Lato server (Django)**:
  - Progettazione e implementazione dei modelli dati.
  - Sviluppo delle logiche di business.
  - Creazione delle API REST per l'interfacciamento con il client.
- **Lato client (Flutter)**:
  - Sviluppo delle interfacce grafiche.
  - Implementazione delle funzionalità per l'inserimento, aggiornamento e cancellazione dei dati.
  - Gestione delle comunicazioni con il server tramite le API REST.

L'intero sistema è versionato automaticamente con Git. Grazie all'uso di Flutter e alle best practice della programmazione multipiattaforma, è stato possibile generare rilasci per piattaforme desktop, mobile e web, garantendo un'esperienza utente uniforme su tutti i dispositivi.
Per migliorare e velocizzare lo sviluppo delle parti comuni, ho utilizzato un programma (da me sviluppato) in grado di autogenerare tutto il codice di interfacciamento tra i due frameworks (Django e Flutter).

04/2020 - 02/2022 | L'Aquila, Italy

Senior Embedded Linux Developer (Technical Leader)

Strumenti utilizzati

- Visual Studio Code
- Eclipse
- GIT + Agile
- Jenkins

Linguaggi, protocolli e tecniche utilizzati

- C/C++
- I2C
- Linux kernel i2c_driver
- Linux kernel i2c_client
- DMA
- Linux kernel V4L2 (Video for Linux)
- Linux kernel led class (struct led_classdev)
- Yocto

Ho coordinato l'assegnazione di tasks tecnici ad un teams di sviluppatori per la realizzazione di uno scanner industriale basato su linux.
Inoltre ho contribuito:

- nella gestione del sistema di build Yocto, per la:
  - scrittura delle ricette per diverse applicazioni e script di sistema (i sorgenti delle apps erano localizzati su server GIT)
  - scrittura delle ricette per consentire build di target differenti (debug/release e per diverse boards)
- nello sviluppo dei seguenti drivers kernel basati su I2C per gestire:
  - un sensore camera OV9282
  - un sensore camera SC031GS
  - un led driver LM3644
  - differenti devices mediante GPIO

I primi due drivers sono stati integrati con V4L2 (Video For Linux) per interagire direttamente con lo stream video proveniente dagli stessi. Ho creato inoltre una system-class ("camera") per rappresentare un'API comune a tutte le applicazioni di alto livello. Ho gestito la sequenza di avvio del kernel linux in quanto alcuni sensori, come SC031GS e LM3644, avevano uno stretto legame nell'inizializzazione dei relativi drivers: ovvero il primo era dipendente dal secondo e, di conseguenza, si doveva garantire la corretta sequenza di avvio.
Tutti i frames catturati con V4L erano inviati ad un'applicazione di alto livello in grado di elaborarli per l'estrazione di informazioni come:

- lettura codici a barre 1D (EAN 13, EAN 8, ...)
- lettura di codici 2D (QrCode, Data Matrix, Aztec, ...)

Per garantire la separazione degli environment sul build server ho optato per una soluzione basata su una docker che si occupada dell'esecuzione dei relativi scripts di build.
Ho effettuato diversi porting di codici sorgenti da architettura ARM ad architettura Linux (Ingenic x1500/x2100).
Sono stato coinvolto anche nella realizzazione di due pipeline Jenkins per l'automatizzazione del processo di build relativo:

- alle builds notturne
- alle builds di rilascio (con relativo versionamento su Serena Dimension)

Per garantire la separazione degli environment sul build server ho optato per una soluzione basata su una docker che si occupava dell'esecuzione dei relativi scripts di build.
Attualmente sto contribuendo anche al porting di diverse funzionalità (relative ai prodotti wireless) da architettura ARM ad architettura Linux (Ingenic x1500/x2100)

10/2018 - 04/2020 | Mirandola, Italy

Embedded Software Developer

Medtronic Bellco

Strumenti utilizzati

- MPLAB X
- Octave
- SVN
- GIT

Linguaggi, protocolli e tecniche utilizzati

- C/C++, Python
- UART, SPI
- PIC, dsPIC, NIOS
- Kalman Filtering
- Non linear Control (Backstepping)

Il progetto è iniziato con la realizzazione di un test-bench per verificare la fattibilità di un prodotto futuro il cuo scopo principale era quello di generare il P.D. fluid (un liquido utilizzato nella dialisi peritoneale) correttamente. In particolare ho sviluppato:

- il driver per controllare tre differenti pompe a pistone
- il driver per controllare la gear pump
- il driver per leggere i dati dal flow-meter e rispettivo algoritmo di filtraggio (basato su Kalman) per ottenere una stima corretta del volume di acqua utilizzato e del flusso.

Ho aggiornato l'algoritmo di controllo per consentire l'inseguimento corretto della traiettoria di stato composta da: temperatura, flusso, RPM delle 4 pompe e stato di apertura/chiusura di deverse elettro-valvole.
Ho sviluppato un framework, in Python, in grado di codificare e decodificare messaggi JSON per consentire la comunicazione tra il layer applicativo (GUI) e il layer hardware (pompe, elettro-valvole, sensori e così via). Questo framework era in grado di controllare una GUI dinamica che rispondeva in maniera interattiva ai messaggi di basso livello. In più è stato utilizzato per la realizzazione di un water purification bench in grado di generare acqua ultra-pura mediante un sistema controllato basato su osmosi inversa e l'utilizzo di filtri particolari.
La seguente attività ha portato anche alla stesura di diversi brevetti e quattro di questi sono stati accettati e verranno pubblicati nel corso del 2022.

09/2017 - 10/2018 | Venaria, Italy

Software Integration Engineer

Magneti Marelli

Ho lavorato per un anno come System Integrato con lo scopo principale di semplificare ed automatizzare gli steps di build di un InfoCluster (Infotainment + Cluster) basato su android. Durante questo periodo ho automatizzato tutti gli steps necessari per avere una build Android (opportunatamente modificata) funzionante. Ho scritto in seguito, differenti JNI (Java Native Interface) di sistema per la realizzazione di un'interfaccia comune per la comunicazione tra applicativo Android (mondo Java) e driver kernel (mondo HAL, C/C++). Grazie al team audio ho integrato all'interno del kernel il driver audio per il controllo di una radio FM basata sul Dirana3.

03 ATTIVITÀ PROFESSIONALE AUTONOMA

06/2018 - 12/2018 | L'Aquila, Italy

Embedded Software Developer

SPITECNO

Strumenti utilizzati

- Mbed Studio
- MbedOS
- ARMC 6 Compiler
- GIT

Linguaggi, protocolli e tecniche utilizzati

- C/C++, Python
- UART, SPI
- STM32F407VG
- Modbus

Ho collaborato per la realizzazione di un prototipo/prodotto di una scheda elettronica in grado di gestire diverse sorgenti di INPUT/OUTPUT sfruttando come protocollo di comunicazione il ModBUS. Il progetto era diviso in tre parti principali:

- Implementazione dello stack TCP/IP stack e del protocollo ModBUS
- Implementazione del firmware di base per supportare tre diverse boards con tre differenti pinouts
- Implementazione di un bootloader in grado di supportare gli aggiornamenti del firmware da remoto (OTA)

Per la prima parte ho implementato tutte le funzioni ModBUS (0x01, 0x03, 0x05, 0x02, 0x04, 0x06, 0x0F, 0x10) per supportare la gestione di alcuni registri "virtuali" da remoto. I registri sono chiamati "virtuali" poichè ho emulato una EEPROM in una FLASH seguendo gli standards STM32. Per la seconda parte invece, ho implementato una FSM (Finite State Machine) in grado di supportare e riconoscere autonomamente le seguenti boards:

- una board avente solo DIGITAL INPUT
- una board avente DIGITAL INPUT + DIGITAL OUTPUT
- una board avente ANALOG INPUT + ANALOG OUTPUT

Ho assegnato i registri ModBUS con quelli relativi alle porte Digitali/Analogiche.
Infine, per la terza parte ho implementato l'OTA per consentire l'aggiornamento del firmware da remoto. Questo è stato fatto partizionando la flash in differenti regioni tra le quali ce n'era:

- una riservata al bootloader
- una riservata al firmware in esecuzione
- una riservata al salvataggio del firmware-update

L'algoritmo di aggiornamento effettuan anche un check per verificare l'integrità e l'origine dello stesso (usando SHA1 + la coppia chiave pubblica/privata).
Lato utente invece, ho sviluppato una GUI scritta in Python che:

- gestisce tutti gli steps per creare un'immagine valida (firmata ed impacchettata in modo tale che venga correttamente riconosciuta del bootloader della board)
- consente all'utente di effettuare l'aggiornamento via OTA

Il prototipo è passato in produzione, è attualmente in vendita ed è usato da diversi clienti.

03/2017 - 07/2017 | L'Aquila, Italy

Embedded Software Developer

R13

Progetto: Smart Road Signs

Linguaggi, protocolli e tecniche utilizzati

- C/C++
- ARM mbed
- Kalman Filtering
- UART, SPI, bootloader

Con uno spin-off (R13) dell'Università dell'Aquila ho sviluppato il firmware per una board basata sull'architettura Cortex-M4 in grado di gestire numerosi sensori:

- un modulo wireless basato sul ESP8266 (WT8266-S1)
- un modulo radio basato sul RTX-MID-3V
- un modulo GSM+GPRS basato sul SIM900
- un modulo GPS basato sul SIRFStar V
- un modulo SPI per gestire l'SD-Card
- un modulo Bluetooth basato sul RN41

Lo scopo principale di questo progetto era quello di controllare un pannello di legs per segnalare pericoli sulle autostrade. I moduli wireless e ragio erano utilizzati per consentire ad un operatore esterno di cambiare la configurazione del pannello (anche mediante smartphone). Invece l'SD-Card era utilizzate per salvare tutte le diverse configurazioni ed i logs del GPS. Questo era un progetto di ricerca proposto da SISAS: una compagnia italiana che si occupa della produzione della maggior parte dei cartelli stradali italiani. Infine questo progetto è passato alla fase di produzione ed è attualmente in uso su diverse delle nostre autostrade.

03/2017 - 06/2017 | L'Aquila, Italy

Software Developer

R13

Progetto: Road Signs Identifications

Linguaggi, protocolli e tecniche utilizzati

- C/C++
- OpenCV
- Kalman Filtering
- Neural Networks

Con uno spin-off (R13) dell'Università dell'Aquila ho sviluppato un sistema, basato su OpenCV, in grado di tracciare ed identificare un segnale stradale partendo da una sequenza di immagini. In particolare si riuscivano a tracciare i cartelli stradali blu e rossi ed identificare un sottoinsieme di essi. Queste informazioni venivano inoltre sfruttate per la mappatura (mediante GPS) dei cartelli stessi.

07/2015 - 08/2016 | L'Aquila, Italy

Web Developer

Progetto: IECON2016

Linguaggi, protocolli e tecniche utilizzati

- PHP
- CakePHP
- AJAX
- PHPRedis
- ISPConfig

Area: CMS (Conference Management System) Mi sono occupato dello sviluppo e della gestione della conferenza dal punto di vista tecnico ed organizzativo. In particolare ho sviluppato tutte le tecnologie necessarie per la gestione automatica della conferenza:

- Un server basato su linux Ubuntu in grado di fornire i seguenti servizi:
  - Server Mail (Dovecot + Client RoundCube)
  - Server PHP
  - ISPCONFIG
- Una piattaforma web
- Un'applicazione mobile compatibile con (Android, iOS e Windows Phone)

La piattaforma web era in grado di gestire:

- la prima sottomissione del paper da parte dell'autore
- la prima e la seconda revisione da parte dei revisori nominati
- la sottomissione finale con relativo controllo del plagio
- registrazione alla conferenza (mediante pagamento)

L'applicazione mobile era in grado di consentire a tutti i partecipanti la consultazione del calendario degli eventi giornaliero nonché la consultazione dei relativi articoli presentati. Inoltre sfruttando l'XMPP (E**x**tensible **M**essaging and **P**resence **P**rotocol), è stata aggiunta una chat privata (simile al ben noto WhatsApp) relativa alla sola conferenza per gestire eventuali emergenze e necessità di tutti i partecipanti. Tale operazione veniva garantita con la presenza, nei giorni della conferenza, di un personale dedicato alla consultazione della chat 24/24.

01/2015 - 08/2015 | L'Aquila, Italy

Embedded Software Developer

Progetto: Chemical process control

Linguaggi, protocolli e tecniche utilizzati

- Cortex-M4
- Uart, SPI, I2C
- PHP + CakePHP
- AJAX
- WebSocket

Ho collaborato con il laboratorio chimico dell'Università degli studi de L'Aquila per lo sviluppo di tutto il necessario per l'automatizzazione di un processo chimico. In particolare, ho contribuito allo sviluppo delle seguenti parti:

- il firmware di una board realizzata con l'FRDM-K64F: un micro-controllore basato sull'architettura Cortex-M4
- interfaccia Web

Per il primo punto ho sviluppato:

- i drivers principali per interfacciare la board con alcune periferiche come elettro-valvole proporzionali, DAC e porta ETH
- gli algoritmi per leggere diversi dati e controllare un forno mediante una PWM e un relè a stato solido

Per il secondo punto, invece, ho implementato un'interfaccia web in grado di:

- configurare il micro-controllore
- leggere alcune informazioni da 2 dispositivi dell'ABB
- leggere la temperatura da una termocoppia
- leggere la pressione da un manometro

Con questa strumentazione, attualmente in uso, è possibile:

- configurare la scheda di controllo per gestire diversi cicli termici e le relative condizione di ingresso ed uscita
- avviare l'esecuzione di quest'ultimi
- programmare l'avvio di una siringa contenente acqua distillata se vengono raggiunte temperature troppo elevate
- raccogliere i seguenti dati (temperatura del forno, composizione del gas all'interno del forno, stato elettro-valvole, pressione)
- elaborare i dati raccolti analizzando le diverse curve ottenute

04 ISTRUZIONE E FORMAZIONE

COMPETENZE INGEGNERISTICHE

- Matematica e Fisica di base
- Analisi e controllo dei sistemi lineari
- Analisi e controllo dei sistemi non lineari
- Teoria della stima e del filtro di Kalman
- Elettronica digitale e sistemi embedded
- Sistemi ibridi
- Controllo robusto
- Sistemi di ritardo
- Controllo e ottimizzazione ottimali
- Robotica
- Teoria dei segnali, teoria della modulazione applicata alle telecomunicazioni
- Elettronica analogica
- Elaborazione del segnale digitale, elaborazione dell'audio e delle immagini
- Fondamenti di scienze elettriche, azionamenti elettrici ed elettronica di potenza

PROGETTI INGEGNERISTICI

- Trajectory tracking applicato della traiettoria per evitare gli ostacoli in auto
- Pendolo inverso
- Implementazione Java di un semplice gioco: Tank War
- Implementazione C++ di un firmware per la scheda TIVA per creare effetti musicali
- Implementazione PLL digitale in Matlab/C++
- Controllo robusto di quadricotteri basato sull'Hybrid-BackStepping
- Diversi progetti relativi alla simulazione e al controllo di un sistema reale

COMPETENZE DI CONTROLLO E AUTOMAZIONE

Algoritmi di Controllo Non-Lineare

- Backstepping
- Sliding Mode
- SuperTwisting
- SAB-SMC (Static Adaptive Backstepping - Sliding Mode Control)
- DAB-SMC (Dynamic Adaptive Backstepping - Sliding Mode Control)

Algoritmi di Controllo Lineare

- PID (Proportional Integral Derivative Controller)
- LQR (Linear Quadratic Regulator Controller)
- LQG (Linear Quadratic Gaussian Controller: LQR + Kalman)

Algoritmi di Filtraggio

- Linear Kalman Filter
- EKF (Extended Kalman Filter)
- QKF (Quadratic Kalman Filter)
- UKF (Unscented Kalman Filter)

Osservatori

- Linear Observer (Luenberger)
- Non Linear Observer (Luenberger-Like, Germani)
- Sliding Mode Observer
- Kalman based Observer

05 LINGUE

Madre Lingua: Italiano

Inglese

Ascolto	Lettura	Interazione Orale	Produzione Orale	Scrittura
B2	B2	B2	B2	B2

06 ONORIFICENZE E RICONOSCIMENTI

- Uno dei pochi a risolvere il problema del "Project Euler 254" (risolto statisticamente solo dallo 0,20% dei partecipanti) ↗ ↗
- Vincitore del primo round di Codeflows 2019 (tra i primi 200 su più di >5000 partecipanti)
- Vincitore di una delle quattro borse di studio per studenti di Laurea Magistrale a cura dell'Istituto Nazionale di Fisica Nucleare (INFN) - Gran Sasso Science Institute (GSSI), con merito. Da Gennaio 2016 a Gennaio 2018.

07 PUBBLICAZIONI E BREVETTI

7 October 2026
Article: **Don’t Retest Everything: Auditable Regression Scoping for Safety-Critical Systems**
Published at: **ASE 2026**

16 September 2026
Article:
Learning state-space models with Kolmogorov-Arnold networks: an autonomous underwater engineering perspective
Published at: **ISA Transactions**

12-19 April 2026
Article:
Deploying SafeKAN for Anomaly Detection in Safety-Critical Satellite Operations: An Industry-Guided Study
Published at: **International Conference on Software Engineering (ICSE 2026)**

12/08/2025
Article: **Intelligent Coordination System for Autonomous Domestic Heating: an AI-Driven Test-Bench** ↗
Published at: **MetroLivEnv 2025**

26/06/2024
Article: **AI/ML for safety-critical software: the case of the space domain** ↗
Published at: **IEEE Software**

25/09/2023
Article:
A model based approach to sw quality models: from Matlab/Simulink model to quality metrics, an AOCS SW experience
↗
Published at: **ESA SW PA Workshop 2023**

29/01/2021
Patent: **Dialysis system and method** ↗

28/10/2020
Patent: **Apparatus for pumping and dialysis** ↗

28/10/2020
Patent: **[US] Peritoneal dialysis system and cassettes therefor** ↗

28/10/2020
Patent: **Peritoneal dialysis system and cassettes therefor** ↗

29/04/2020
Patent: **Peritoneal dialysis fluid from concentrate containers** ↗

08 TESI

Laurea Magistrale:"Indoor localization with Ultra Wide Band sensors and quadcopter's advanced control: an application with the CrazyFlie"

- **Laurea Triennale:**"Tracking and Visual Servoing of a quadcopter"

09 CERTIFICAZIONI

- **Iscrizione all'Ordine Degli Ingegneri della Provincia dell'Aquila** - Dicembre 2020
- **Esame di abilitazione alla professione** in "**Ingegneria dell'Informazione**" - Luglio 2020
- **Laurea magistrale cum laude e menzione accademica** in "**Ingegneria Informatica e Automatica**" - 22 Luglio 2017
- **Laurea triennale cum laude** in "**Ingegneria dell'Informazione**" - 24 Luglio 2015
- **CCNA Routing and Switching: Introduction to Networks** - 16 Giugno 2015
- Diploma di scuola media superiore presso il Liceo Scientifico "A. Bafile" PNI ottenuto a Luglio 2012
- **Trinity College London B2** - Aprile 2011
- **British Institutes ESOL Certificate B1** - 14 Ottobre 2010

10 CONFERENZE E SEMINARI

- **12-18/04/2026**
  Partecipazione come autore ad **ICSE (International Conference on Software Engineering) 2026** tenutosi a Rio de Janeiro dal 12 al 18 Aprile 2026.
- **22-25/09/2025**
  Partecipazione come autore all'**ESA SW PA Workshop** tenutosi a Noordwijk dal 22 al 25 Settembre 2025.
- **25-28/09/2023**
  Partecipazione come autore all'**ESA SW PA Workshop** tenutosi a Madrid dal 25 al 28 Settembre 2023.
- Partecipazione + gestione ed organizzazione della conferenza IECON2016 tenutasi a Firenze dal 24 al 27 Ottobre 2016
- Partecipazione alla conferenza RomaDrone tenutasi a Roma il 23 Marzo 2016
- Partecipazione, come studente visitatore, in diverse attività presso l'"Harbin Institute of Technology" in Cina

11 ALTRE COMPETENZE

- Setup di un server Linux con: ISPConfig, Dovecot, Postfix e RoundCube

12 STACK TECNOLOGICO

Programmazione

C · C++ · Python · Ada · Rust · Dart · JavaScript · Java

Embedded e piattaforme

Embedded Linux · Yocto / BitBake · U-Boot · Driver Linux · ARM mbed · PIC / dsPIC · HLS / FPGA

Protocolli

I²C · SPI · UART · CAN · SpaceWire · Ethernet · TCP/IP · Modbus · RS-232 / RS-485

Framework e AI

Flutter · Django · React · Node.js · Kolmogorov-Arnold Networks

Sistemi e strumenti

Linux · Git · Docker · Jenkins · GNAT Studio · GRMON · IBM DOORS · MPLAB X

Dati

MySQL · MongoDB · Oracle · JSON · XML

13 PROFILI PROFESSIONALI

- GitHub: **https://github.com/capriele**
- Bitbucket: **https://bitbucket.org/capriele**
- GitLab: **https://gitlab.com/capriele**
- LinkedIn: **https://www.linkedin.com/in/alberto-petrucci/**
- Orcid: **https://orcid.org/0009-0000-7363-8388**