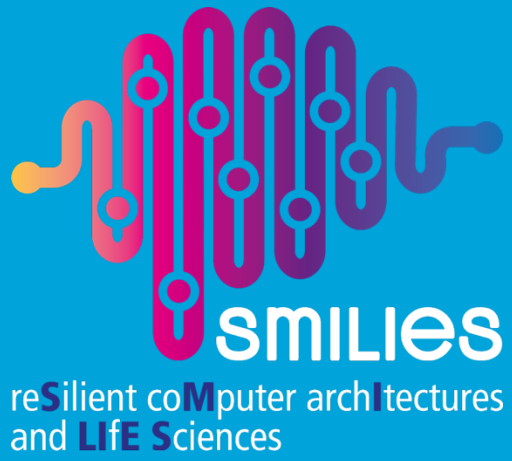



Politecnico di Torino

Department of Control and Computer Engineering


# COMPUTER ARCHITECTURE AND OPERATING SYSTEMS

# EMBEDDED OPERATING SYSTEMS

STEFANO DI CARLO

# A JOINT COURSE FOR TWO CURRICULA

- [01GYKUV] — Computer architectures and operating systems (Master of science-level in Cybersecurity)
  - Official description and rules
- [02NPSYG, 02NPSOV] — Operating systems for embedded systems
  - Official description and rules

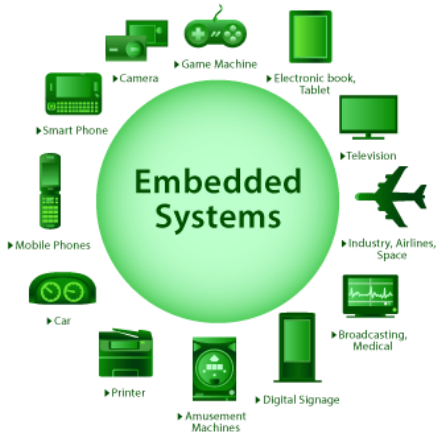
# TEACHERS

**Stefano Di Carlo**
stefano.dicarlo@polito.it

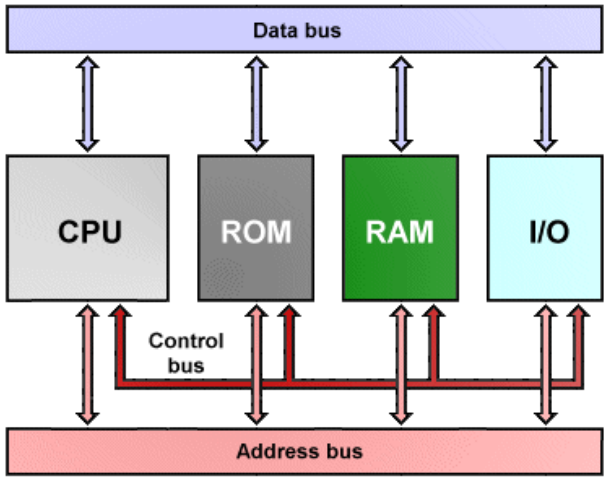
**Tzamn Melendez**
tzamn.melendez@polito.it

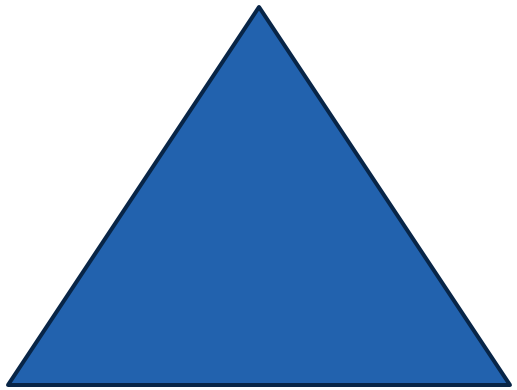
**Enrico Magliano**
enrico.magliano@polito.it

# GOAL



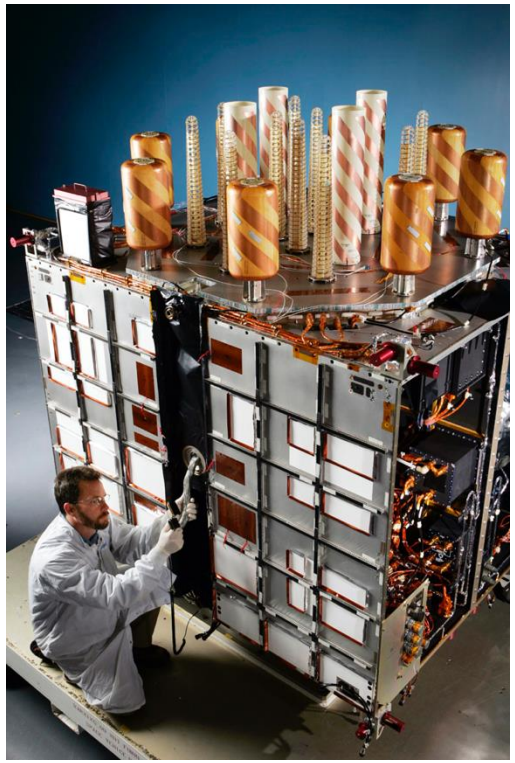

Computer Architecutres

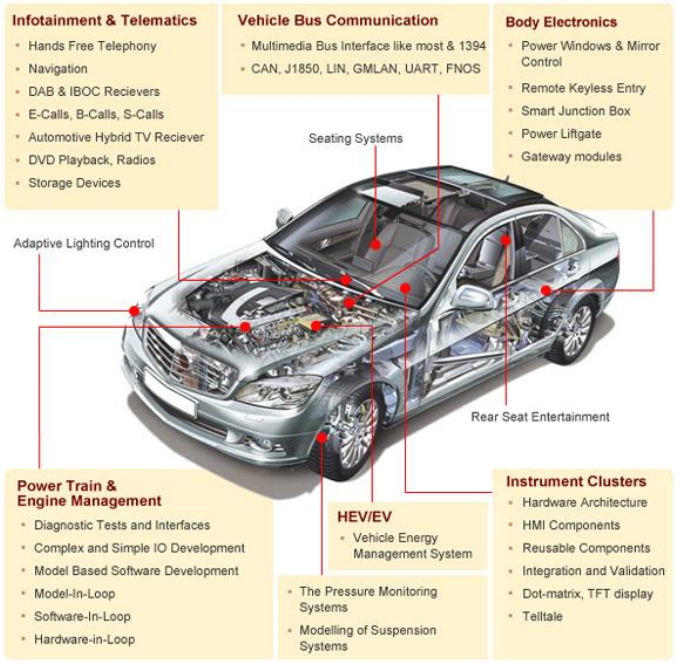


# WHY SHOULD I CARE ABOUT THIS TOPIC?

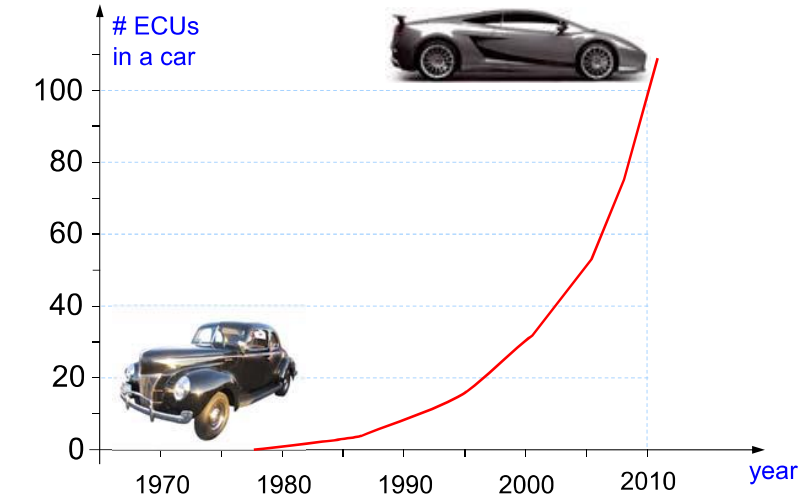
- Embedded systems are everywhere

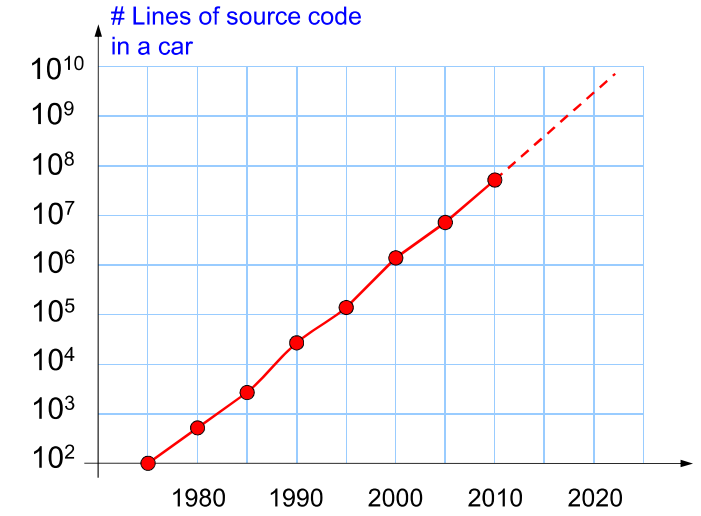





# WHY SHOULD I CARE ABOUT THIS TOPIC?

- Software + hardware is everywhere
- Complexity is growing dramatically
- The old-fashion approach (1 main loop) is no longer feasible to cope with this complexity
- Operating systems are needed to raise the abstraction level and to make programmer life easier

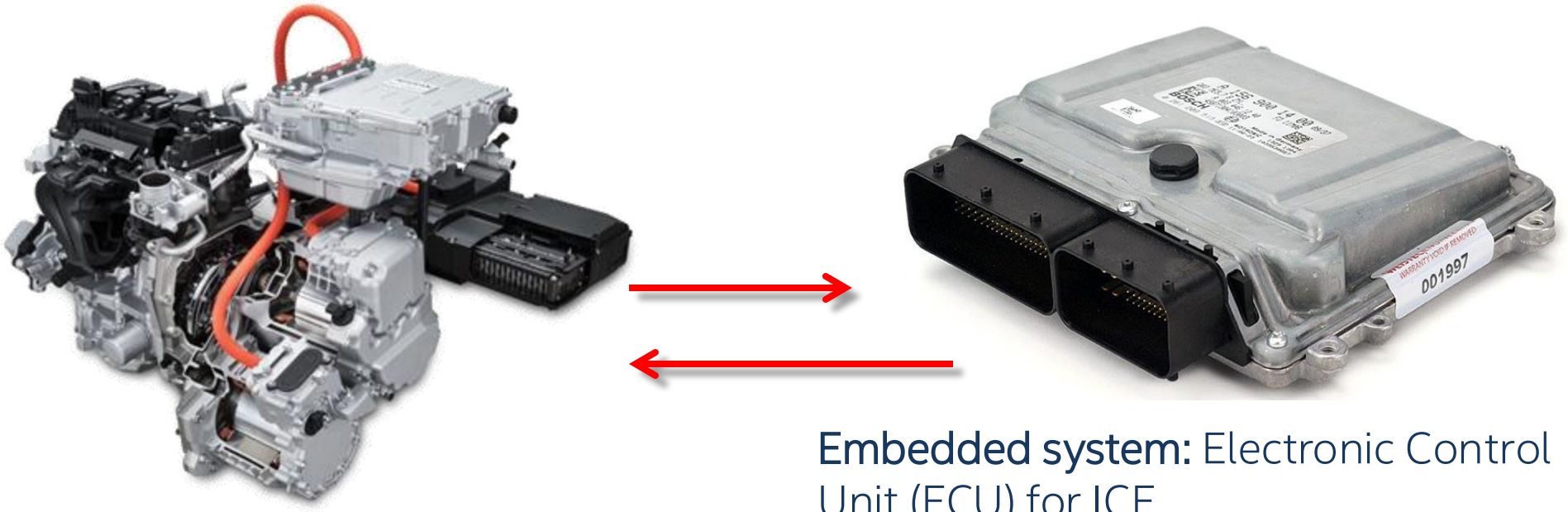


# INFORMAL DEFINITION OF EMBEDDED SYSTEMS

- Embedded Systems are special-purpose computers designed for specific applications

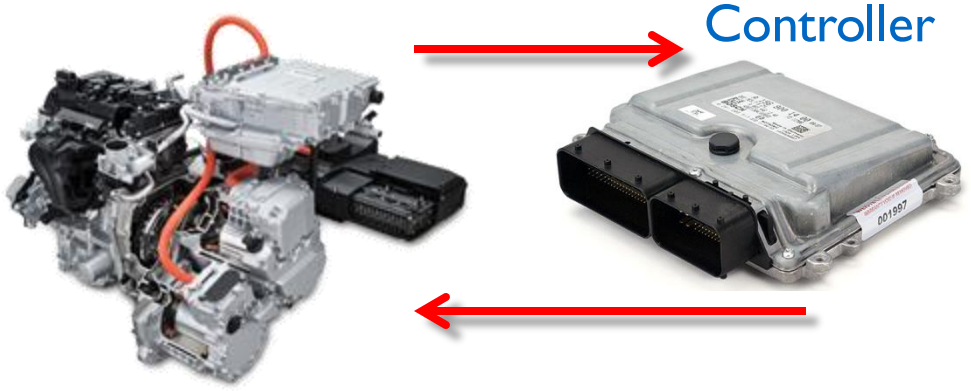
**Specific application:** Hybrid powertrain (HyPWT) control



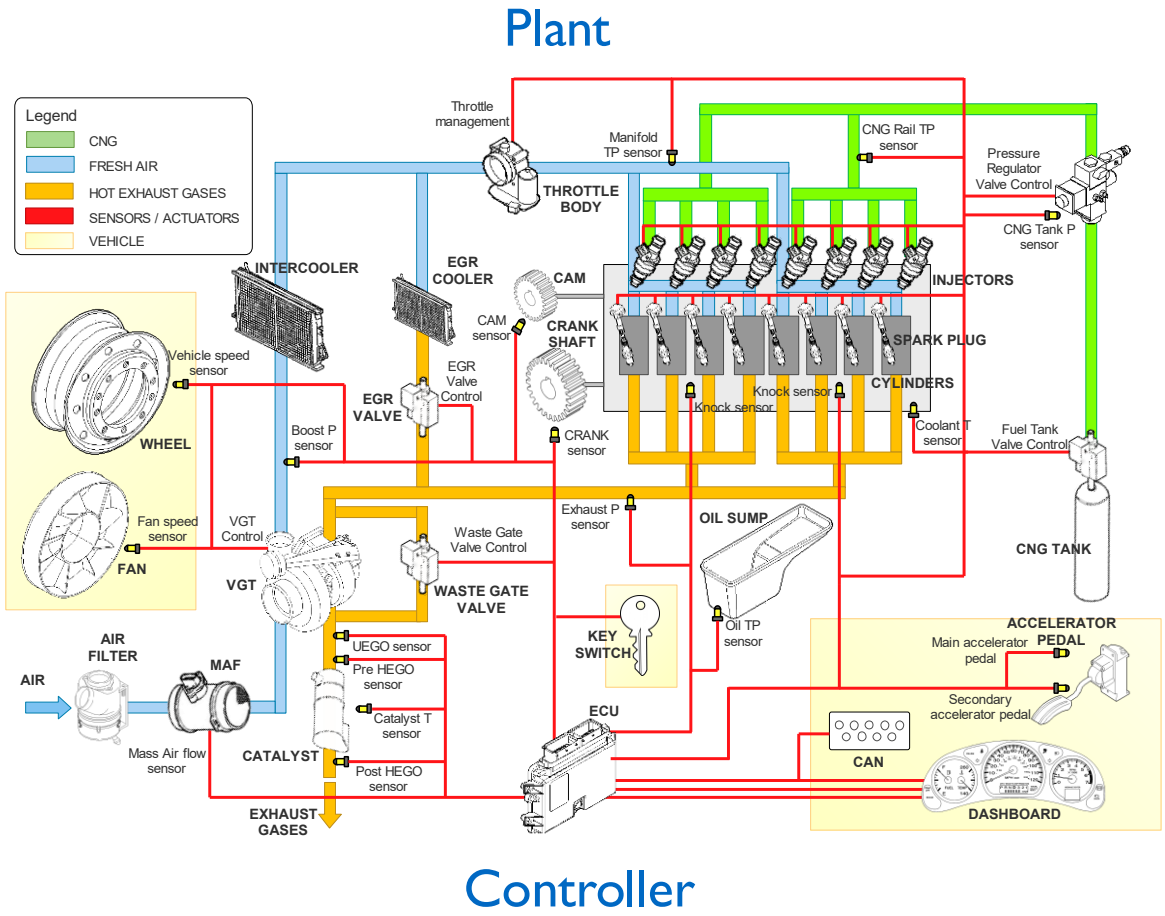
**Embedded system:** Electronic Control Unit (ECU) for ICE

# INFORMAL DEFINITION OF EMBEDDED SYSTEMS

- An embedded system (i.e., the controller) interacts with a physical system (i.e., plant) through actuators/sensors

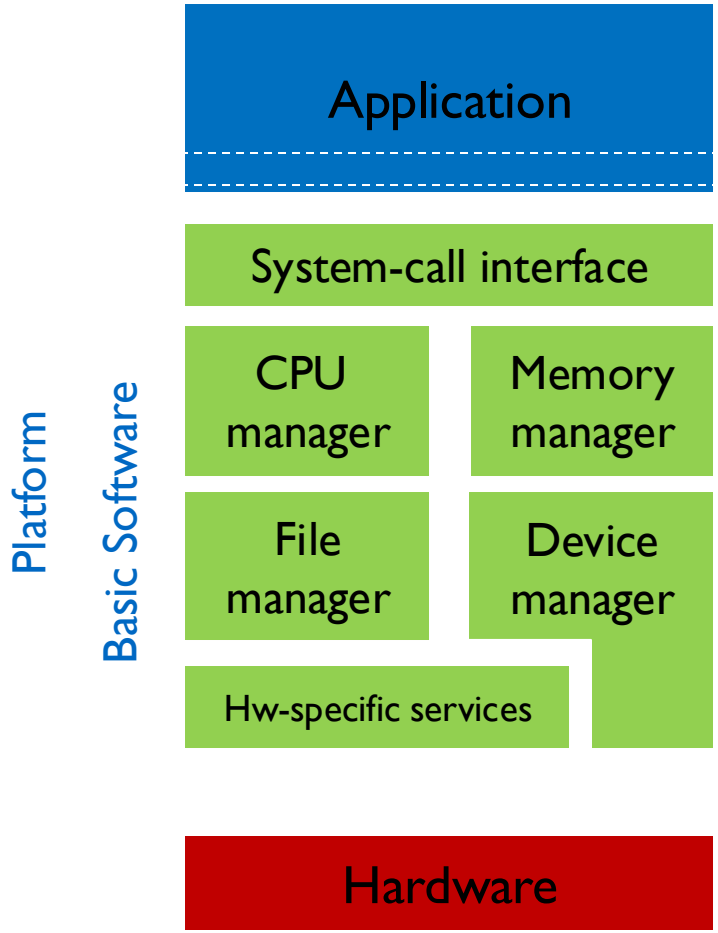



- Embedded system runs software: application sw + basic sw



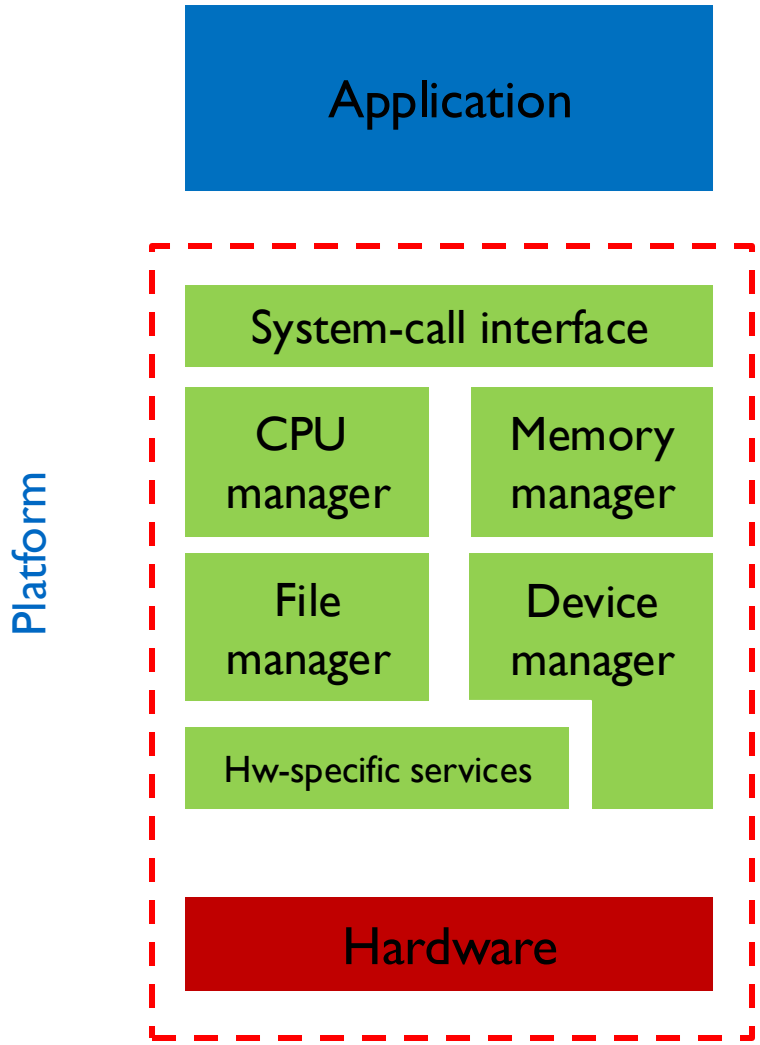
# CONCEPTUAL MODEL FOR EMBEDDED SYSTEMS

- Two main components
  - Application
  - Platform
- Application
  - It is the software that implements the functionalities the embedded system is intended
- Platform
  - It is the combination of SW components (a.k.a., basic software) and HW components that provide the services needed by the application to run



# FOCUS OF THIS COURSE

- The architecture of a computing system (3 CFU):
  - Advanced description of the basic microprocessor architecture
  - Introduction to modern microprocessor architectures
  - CISC, RISC, and superscalar processor architectures, behavior and performance
  - Microprocessor-based systems architecture
- Development flow of bare metal embedded system applications (1 CFU):
  - Cross-compiling flow
  - Emulators and simulators
  - Debugging and profiling
  - Virtual platforms
- Operating Systems for Embedded Systems (3 CFU):
  - Real-time and process scheduling
  - Anatomy of an operating system for embedded systems
  - Analysis of the architecture and services offered by operating systems for embedded: (e.g., FreeRTOS, Embedded Linux)
- Programming of an embedded operating system (1 CFU):
  - Management and device drivers.
  - Use of an evaluation board.s
  - How to use development tools to build an operating system for embedded systems.
  - Design and development of device drivers for custom devices



# EXAM – PART 1.1 (MANDATORY)

- **Written test** that covers all topics discussed in class.
- The test may include various types of questions, such as:
  - multiple-choice questions
  - exercises focusing on specific topics
- The test is given using Moodle and Lockdown browser
- Each question is assigned a score, which is announced during the exam.
- The maximum score achievable in this part is 18
- The duration of the test is 60 minutes.

# EXAM – PART 1.2 (ONLY FOR STUDENTS WITH PROJECT)

- Written test with additional questions for students working on the project that are not requested to give the programming part of the written exam
- Questions more oriented to the programming topics covered during the course
- The test may include various types of questions, such as:
  - multiple-choice questions
  - exercises focusing on specific topics
- The test is given using Moodle and Lockdown browser
- Each question is assigned a score, which is announced during the exam.
- The maximum score achievable in this part is 7
- The duration of the test is 15~30 minutes.

# EXAM – PART 1.3 (ONLY FOR STUDENTS WITHOUT PROJECT)

- 1 or 2 programming exercises that can cover FreeRTOS, ARM assembly programming, Linux Device drivers
- The test is developed on computer without using lockdown browser
- You will have  access to the toolchain and documentation:
  - The allowed documentation is provided during the exam
  - The use of Copilot or other generative AI is forbidden. Any violation will be reported to the dedicated authority and a suspension of the career for 1 year will be requested
- The maximum score achievable in this part is 7
- The duration of the test is 30~40 minutes.

# EXAM – PROJECT

- A **project** that emphasizes the application of the concepts learned in class and enhances the student's ability to independently find documentation and acquire new skills beyond the scope of the course.
- The project aims to assess:
  - The student's proficiency in implementing and developing embedded applications.
  - The efficiency of the implementation.
  - The student's ability to work effectively in a team.
  - The student's presentation skills.
  - The individual contributions of each student to the project.
- The maximum score is 10

# EXAM – PROJECT

- Projects are a TEAMWORK
  - If you want to work on a project you must enroll by  using the project on-line spreadsheet:
  - Every team must include a minimum of 3 students and a maximum of 6 (at least 4 students is recommended)
- Common issues:
  - All members must be present when delivering the project. Talk with your colleagues and agree in advance on the timeline to develop the project. Commitment is important. If somebody underperforms everybody is affected.
  - If a member of the team wants to withdraw, it is possible by informing in due time all other members and the professor with a written email. Its project evaluation will be 0.
  - If a member of the team disappears, it will be penalized with a negative evaluation of the project (- 2 points on the final grade)
  - If you do not deliver the project, you will be required to attend an exam and pass Part 1.3.

# EXAM – PROJECT

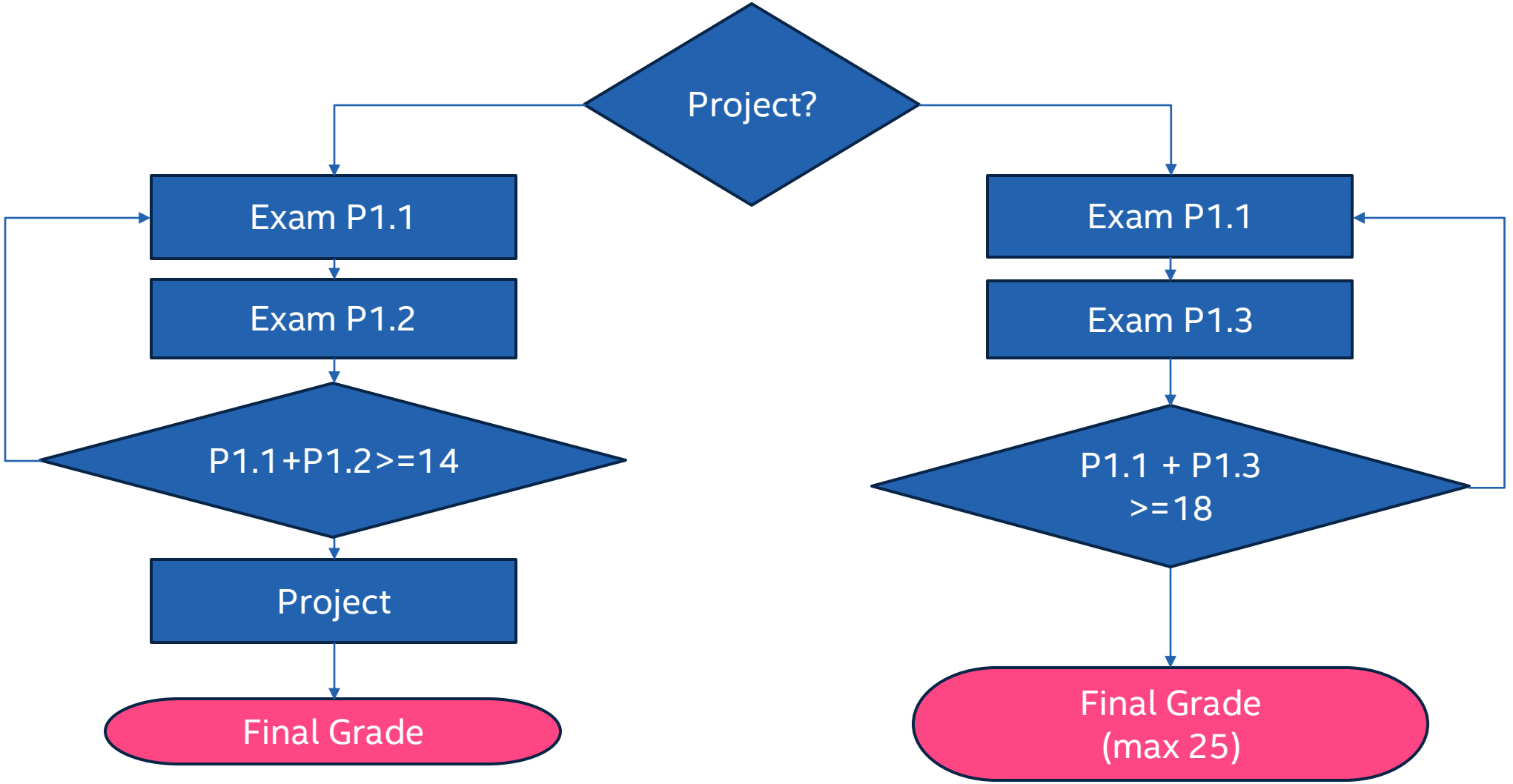
- Projects must be completed by Sept. 30th, 2026.
- The evaluation must involve the entire team.
  - Make sure you agree with your team mates on when you want to deliver your project
  - The evaluation can be done online
- The project is a one-shot opportunity
  - You have the opportunity to be assigned to a project during the first academic year in which you attend the course.
  - If you do not opt for the course this year you will not be able to do it next year
  - If you join a team and withdraw you will not be able to join another team next year
- The project is an opportunity to increase and to evaluate your capability to build a team and to work with other people. If personal issues among team members arise try to work them out. If this is not possible, organize a meeting with the entire team and the professor

# EXAM – PROJECT TIMELINE

- October 15th — Deadline to communicate team composition
- November 1st — Topic assignment
- Project development (free but strongly advised to develop the project during the course to take advantage of the tutoring hours)
- Project presentations: one day dedicated to project presentations at the end of each session of exams
- Projects must be completed before the beginning of the next edition of the course

- THE PROJECT REQUIRES TO STUDY NEW TOPICS!!!!!!

# EXAM WORKFLOW



# EXAM EVALUATION

- Part 1.1 and 1.2 will be automatically corrected in Moodle:
  - + You will know the result of the exam immediately
  - + No errors during the evaluation
  - - Small errors (e.g., calculations) may propagate and led to loss of points

We decided for this approach based on the feedback of students from the last year that complained about the time to evaluate the exam.

If after checking your exam you realize that you were penalized due to a simple error (e.g., miscalculation) you can appeal, and we will look into the case. Use this possibility carefully.

# EXAM – FINAL GRADE

- The final grade is determined by summing up the different parts capping it to 32.
- The highest distinction, "Laude," is awarded if the total grade equals 32.
- Written exams and the project can be completed within different timeframes.
- You will have your grade recorded only when all parts of the exam are complete

# MIDTERM ASSESSMENTS

- Midterm Assessments (Part 1.1 & 1.2)
  - 3 assessments during the official schedule
  - Each covers only the topics of its period
  - Each worth 1/3 of the total points for Part 1.1 & 1.2
- Evaluation Rules
  - **With project**: score ≥14 in Part 1.1 + 1.2 → skip exam, only project
  - **Without project**: if satisfied with Part 1.1 score → exam covers only Part 1.3
  - If you attend the written exam the results of the assessment are invalidated

# TEACHING MATERIAL

- All teaching material will be available at:
  - https://baltig.polito.it/teaching-material/02npsov-operating-systems-for-embedded-systems
- Labs
  - Your own PC
  - Free tools and/or virtual machines provided by the teacher
- Videos and other teaching material:
  - Every class will be broadcasted on Zoom and recordings will be available

# IN PERSON VS REMOTE

- Every class will be broadcasted live on Zoom and recorded. You can choose your best way of attending the course
- A few remarks:
  - Interaction with remote students during lectures will be limited (I will not consistently check the chat)
  - LABS will not be broadcasted and recorded
  - Statistics tell that ~90% of students that attend in person and actively interact during class pass the exam without problems while remote students tend to struggle (qualitative data)

# USE OF GENERATIVE AI

- Generative AI is a type of artificial intelligence that can learn from and mimic large amounts of data to create content such as text, images, music, videos, code, and more, based on inputs or prompts.

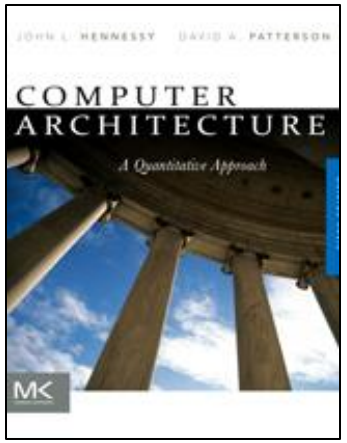

- I support responsible experimentation with generative AI tools, but there are important considerations to keep in mind when using these tools, including information security and data privacy, compliance, copyright, and academic integrity.

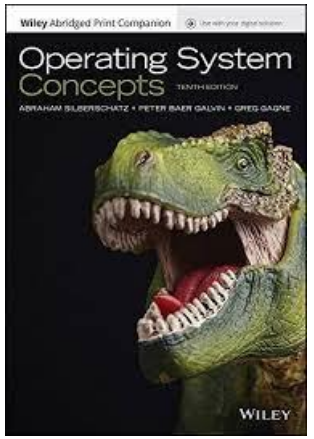
# USE OF GENERATIVE AI

- GEN-AI is very effective when prompted to generate code for our course or solve exercises and can be used efficiently to:
  - Generate new exercises and check the solutions
  - Explain code that you do not understand
  - Search for additional information
- HOWEVER!!!!
  - AI-generated content can be inaccurate, misleading, or entirely fabricated (sometimes called “hallucinations”). You must be able to identify these hallucinations.
  - You must be able to write code or solving exercises by yourself to be able to identify the hallucinations.
  - During the exam you will not have access to these tools.
  - Using GEN-AI to work on your project is ok but you will be discussing it with the professor without access to the tool.

# BOOKS

- J.L. Hennessy, D.A. Patterson
  Computer Architecture: a Quantitative Approach
  Morgan Kaufmann Publishers, Inc., V Edition, 2012
- Silberschatz, Galvin, Gagne, "Operating System Concepts", John Wiley & Son



# SCHEDULE

- Official timetable
  - Check the official schedule since it is sometimes updated
  - A detailed schedule with topics will be available on GIT
- LAB groups are available on GIT.
  - Only new students enrolled in A.Y. 2025/26 are assigned a group.
  - Other students can attend if space is available
  - You are free to switch day as far as space is available
- Remarks
  - Some lectures will be delivered on-line

# COMMUNICATIONS

Discord

https://discord.gg/awAZhezr

PoliTO

- Emails
- Push notifications