

31-10-2025

# Deliverable 2.1 Self-standing modules: Market analysis and curriculum design

Contractual Date: 31-10-2025  
Actual Date: 31-10-2025  
Grant Agreement No.: 101123118  
Work Package: WP2  
Task Item: T2.1  
Lead Partner: TALTECH

**Author:** Peeter Ellervee (TALTECH)

## Abstract

This deliverable presents the market analysis and the development of self-standing learning modules for the RESCHIP4EU project, which aims to enhance embedded systems education across Europe. The market analysis complements the analysis for WP1 D1.1. and outlines additional needs by the partner countries.

11 self-standing modules have been developed and implemented targeting both already working professionals and fresh graduates who are seeking for job. In addition, these modules can be incorporated into full-scale courses at universities to fill gaps in their curricula.

© EIT Digital on behalf of the RESCHIP4EU project.

The activities leading to these results has received funding from the European Community's DIGITAL Programme under Grant Agreement No. 101158828 (RESCHIP4EU).

This document is licensed under a [Creative Commons Attribution 4.0 license](https://creativecommons.org/licenses/by/4.0/).



## Versioning and contribution history

Version	Date	Authors	Notes
0.1	29/09/2025	Romane Léauté (EITD), Peeter Ellervee (TALTECH)	Initial structure
0.2	9/10/2025	Romane Léauté (EITD), Peeter Ellervee (TALTECH)	Sub-sections clarified, structure updated
0.3	14/10/2025 16/10/2025	Majzik István (BME) Hannu-Matti Järvinen (TAU)	BME self-standing modules Market analysis (Finland)
0.4	17/10/2025	Dabóczy Tamás (BME), Alessandro Savino (POLITO)	BME self-standing modules; market analysis (Italy)
0.5	21/10/2025	Cristiane Haberl (EA), Franco Callegati (UNIBO)	Implementation platform; market analysis (Italy)
0.6	22/10/2025	Hashem Haghbayan (UTU)	UTU self-standing modules
0.7	25/10/2025	Christian Pilato (POLIMI)	POLIMI self-standing modules
0.8	26/10/2025	Peeter Ellervee (TALTECH)	Full version
1.0	29/10/2025	Romane Léauté (EITD), Alessandro Savino (POLITO), Hannu-Matti Järvinen (TAU)	Proofread and quality check
1.1	30/10/2025	Peeter Ellervee (TALTECH)	Final version

# Table of Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. Market Analysis</b>	<b>6</b>
<b>2.1 Background</b>	<b>6</b>
<b>2.1.1 Estonia</b>	<b>7</b>
<b>2.1.2 Finland</b>	<b>7</b>
<b>2.1.3 France</b>	<b>8</b>
<b>2.1.4 Hungary</b>	<b>8</b>
<b>2.1.5 Italy</b>	<b>8</b>
<b>2.2 Reasons for self-standing modules</b>	<b>10</b>
<b>3. Curricula Design</b>	<b>10</b>
<b>3.1 Programme objectives</b>	<b>10</b>
<b>3.2 List of self-standing modules</b>	<b>11</b>
<b>3.3 Topic groups of self-standing modules</b>	<b>13</b>
3.3.1 Background	13
3.3.2 Assessment and grading	14
3.3.3 Certification schemes	14
<b>3.4 Self-standing modules descriptions</b>	<b>15</b>
3.4.1 Model Based Design and Verification (BME)	15
3.4.2 Signal processing algorithms in embedded systems (BME)	16
3.4.3 The Art behind Reconfigurable and Adaptive Computing Systems (POLIMI)	17
3.4.4 Design Methods for HW Security and Trust (POLIMI)	19
3.4.5 Open-source ASIC design (POLIMI)	20
3.4.6 Computer Architectures (POLITO)	21
3.4.7 Artificial Intelligence Safety (POLITO)	22
3.4.8 GPU Programming (POLITO)	23
3.4.9 Testing and Fault Tolerance (POLITO)	25
3.4.10 Embedded Platforms and Embedded Control (UTU)	26
3.4.11 CPU–FPGA Heterogeneous Computing for Artificial Intelligence Applications (UTU)	27
<b>3.5 Implementation platform</b>	<b>28</b>
<b>Conclusions</b>	<b>28</b>
<b>References</b>	<b>30</b>

## Table of tables

Table 1: Self-standing modules by topic groups

14

## Table of figures

**No table of figures entries found.**

# 1. Introduction

This deliverable contains two main sections.

Section 2 presents the market analysis to develop self-standing modules for the embedded systems design. The market analysis complements the analysis for WP1 D1.1. and outlines additional needs by the partner countries.

Section 3 describes the self-standing learning modules, their grouping into topic areas, and the implementation platform.

## 2. Market Analysis

### 2.1 Background

The market analysis was essentially carried out in parallel with WP1 Task 1.1 while focusing on the market needs of different target audiences. This is because the self-standing learning modules will focus also on professionals in need of up-skilling and re-skilling, as well as on job seekers. It should be added that the target audience is also broader compared to that of Task 1.1. The goal was also to understand the needs of the job market in the shorter term - the target audience of the self-standing learning modules includes professionals that are already on the job market and looking to learn new skills in the short term.

This market analysis builds upon the findings presented in Deliverable D1.1 by identifying additional needs specific to the partner countries. Deliverables D1.1 and D1.2 are complementary and should be read together for a global understanding of the market context.

Summarizing from WP1 D1.1, it is important to overcome educational gaps, collaboration between academia and industry is essential. Academic learning frequently doesn't prepare students for practical work due to the complexity and cost of industry tools, causing gaps between theoretical knowledge and practical experience. Universities must foster a mindset of continuous learning, teaching fundamentals, and applying them to modern problems. Engineering education should focus on developing problem-solving skills and adapting to digital transformation and new technologies.

Moreover, the European Chips Act<sup>1</sup> strongly emphasizes the need for innovative, accessible, and industry-aligned upskilling initiatives to ensure Europe has the talent required to lead in microelectronics research, design, and manufacturing. The RESCHIP4EU online modules address this skills challenge.

- The European Chips Act specifically identifies a shortage of skilled professionals as a major obstacle to strengthening the semiconductor ecosystem in the EU. Addressing this gap is essential for Europe to achieve its goal of doubling its global market share in semiconductors to 20% by 2030.
- One of the Act's core objectives is to support education, upskilling, and reskilling initiatives across Europe, particularly through online modules, postgraduate programs, short-term courses, apprenticeships, and practical laboratory experiences. These initiatives are essential to produce a workforce capable of supporting advanced chip design, manufacturing, and research.
- The Act calls for the development of learning-outcome-based curricula and new degree programs in partnership with industry. This ensures training is relevant to current and future occupational profiles in the fast-changing microelectronics sector.
- The establishment of national competence centers and the European Chips Skills Academy, coordinated actions are being taken to harmonize training

---

<sup>1</sup> <https://digital-strategy.ec.europa.eu/en/policies/european-chips-act>

standards, recognize micro-credentials, and anticipate evolving skills needs across EU countries. The RESCHIP4EU programme has established synergies with the European Chips Skills Academy, ensuring that its training modules are referenced and promoted within the Academy's platform. This partnership supports the broader European strategy to upskill the semiconductor workforce and reinforces the alignment of its educational content with sector-wide priorities and best practices.

## 2.1.1 Estonia

In addition to the market analysis done for WP1 T1.1, additional analysis was performed for the Engineering Academy programme where TalTech is one of the partners. The aim of the programme is to improve wider higher education in engineering, including the areas of electronics, computer and control engineering. The analysis focused on the courses of the study programme to identify how well different relevant topics have been covered and whether there exist unnecessary duplications. Also, since professionals from industry need courses to improve their skills, additional sub-topics were identified.

Based on that, there are needs to improve skills in the areas of embedded hardware and software, chip and IP design, testing and verification, plus soft and professional skills. The same applies for the skills of distributed computing and control, artificial intelligence and machine learning, security and reliability. Course content updates are under way together with the development of micro-degrees targeting already working professionals.

## 2.1.2 Finland

The market research in Finland was conducted as a data triangulation and a methodological triangulation: (1) Systematic literature review of analysing industry reports and articles related to embedded systems was done to track needed skills, trends, challenges and educational gaps in the global industry, (2) conventional content analysis of 114 job advertisements in the field of embedded systems was carried out to identify in-demand skills, tools and technologies with emphasis on current needs in Finland, and (3) directed content analysis of 11 interviews among employees in embedded systems companies was done to gain a deep understanding and get a direct perspective on the embedded systems industry needs on both current and future needs in Finland.

Current needs include embedded hardware, operating systems, development boards, build and deployment, communication and networking technologies, software development, quality assurance and testing, architecture, development processes, engineering, and soft and professional skills.

Future needs include cloud and edge computing, artificial intelligence and machine learning, embedded hardware and software, data analytics, security, and system architecture skills. [Saukonoja 2025]

## 2.1.3 France

For France, both for industry and academy, a skilled workforce is essential. It is important attracting young people to STEM education, optimally training individuals through traditional pathways (bachelor's, master's, engineering schools, doctorate), and ensuring continuous professional development to facilitate skill enhancement and even the retraining of individuals from other industrial sectors, to develop a qualified workforce. Based on the analysis, the need focuses on engineers able to work in an international context, having a strong general knowledge of electronics systems architecture and basic components, and how to simulate or measure them. Specific attention must be paid to the training of engineers with specific skills in fundamentals of microelectronics devices (even back to physics) and analog design.

## 2.1.4 Hungary

One of the driving forces of the Hungarian economy is automotive engineering where, for instance, autonomous driving is a strong driving force as application development (and not component development). For this, the engineers need to understand both the functionality of embedded systems and high-level information processing algorithms, including artificial intelligence techniques. Additionally, proficiency in model-based systems design and analysis is often expected, as model-based system engineering is considered an effective approach for managing and mastering complexity of contemporary embedded systems that are typically distributed, intelligent, mobile, and cooperative, and thus require the interaction of dozens of software and hardware components. Related skills include architecture design, considering extra-functional requirements such as safety and dependability, capturing interfaces and interactions, and model-based testing and verification.

## 2.1.5 Italy

As part of the broader European embedded systems market, Italy is experiencing significant growth driven by the rapid adoption of IoT, industrial automation, and the increasing integration of embedded solutions in automotive, healthcare, and consumer electronics. The European market is particularly focused on automotive innovation, industrial automation, and innovative infrastructure, all of which are strong sectors in Italy due to its robust manufacturing base, automotive industry (with major players like Fiat and Ferrari), and a growing ecosystem of technology startups and SMEs.

According to the EdTech Observatory of the School of Management at the Politecnico University of Milan, in 2022, Italian companies planned and allocated 40% of their training budget to digital learning (around 480 thousand euros per company). At the same time, universities are considering allocating their budget to digital transformation (around 5.6%), encompassing more than 50% of the universities. Among the innovative tools, open badges are gaining 58% widespread adoption [EdTechIta2022].

In the Emilia-Romagna region, where the University of Bologna is located, the regional government has a strong and well-defined strategy (Data valley bene comune) for the adoption of digital technologies also in the public sector. Among others, the goal is to

establish an open badge digital certification system that can reach up to 200,000 citizens. Moreover, a recent report of the “Internet of Things Thematic Community” published in April 2025 shows that embedded and IoT technologies are penetrating also public services, with 65% of the local administration already deploying IoT systems for environmental monitoring, traffic control, water monitoring, and air quality monitoring.

The following analysis was conducted by triangulating publicly available industry reports, regional market data, and academic sources, combining top-down market sizing with bottom-up estimates of skills gaps. Sector-specific indicators from automotive, industrial automation, and education investment data were cross-referenced to provide a comprehensive view of Italy’s strategic and training needs.

Italy faces a convergence of factors shaping its competitiveness: strong industrial sectors increasingly dependent on embedded systems and chip-design capabilities, a persistent shortage of specialized technical skills, and a rapidly expanding e-learning market that offers a scalable channel to bridge this gap. These dynamics jointly underline why new modular online courses targeting embedded systems and chip design are both economically and strategically necessary.

Italy’s industrial backbone remains anchored in sectors where electronics, sensors, microcontrollers, and hardware/software integration are indispensable. The automotive industry, valued at about 92.7 billion euros in 2022, accounted for 5.2 percent of GDP and 9.3 percent of manufacturing turnover, with exports of around 18 billion euros in vehicles and 23.5 billion euros in components [ICE2023]. This sector’s transition toward electrification, connectivity, and autonomy has sharply increased the demand for embedded design and system-integration expertise. Similarly, the industrial automation and control market was valued at about 8.2 billion USD (7.6 billion euros) in 2023 and is projected to reach 14.2 billion USD (13.1 billion euros) by 2031, growing at a rate of about 7.5 percent annually [VerMarket2024]. This expansion reflects the digitalization of Italian manufacturing through robotics, industrial IoT, and edge-compute systems. Complementary industries such as packaging machinery, projected at about 9.6 billion euros by 2025, further reinforce the structural demand for embedded and hardware-design talent [Packmedia2024].

Despite this industrial momentum, Italy continues to show critical shortages in technical and digital skills. Only 45.8% of Italians aged 16-74 possess at least basic digital skills, well below the European Union average of 55.6%, which prompted the government’s National Digital Skills Strategy to target 70% by 2030 [EUDigSkills2024]. On the enterprise side, only about 8% of firms reported using artificial intelligence in 2024, reflecting broader limits in digital adoption [Reuters2025]. Industrial reports also cite an aging workforce and a shortage of qualified engineers as barriers to automation growth [VerMarket2024]. Many companies are also developing internal training programs to compensate for the lack of specialized candidates [Reuters, Italian Firms Bridge Skills Gap with Own Schooling, 2024, <https://www.reuters.com/markets/europe/italian-firms-bridge-skills-gap-with-own-schooling-2024-07-02>].

At the same time, Italy’s digital-learning market is expanding rapidly, offering a realistic and scalable mechanism to upskill professionals. The e-learning services

market reached about 6.4 billion USD (5.9 billion euros) in 2024 and is expected to grow to 18.7 billion USD (17.1 billion euros) by 2030, with a yearly growth rate close to 20 percent [GVR2024]. The online education market, which includes vocational and professional training, is forecast to expand from about 2.05 billion USD in 2024 to 14.8 billion USD by 2033, with a yearly growth rate above 24 percent [IMARC2024]. Corporate learning is a key driver, with companies dedicating roughly 40% of their training budgets to digital programs [EdTechIta2022].

This evidence demonstrates that the Italian market requires flexible and accessible professional training through online courses. Engineers and small and medium-sized enterprises in the automotive, industrial automation, IoT, and semiconductor sectors need specialized courses that traditional university programs cannot provide at the required scale or pace. Online modules can directly target these professionals by offering practical experience through remote laboratories or hardware kits and by providing micro-credentials such as open badges, which are increasingly used in corporate upskilling programs in Italy.

In conclusion, the Italian market shows both a structural shortage of embedded systems and chip design competencies and a strong economic foundation for digital course delivery. Developing modular, practice-oriented online courses focused on embedded design, SoC integration, and hardware-software co-development is therefore not only educationally justified but also economically strategic. These initiatives can directly support Italy's industrial transformation, enhance competitiveness in high-value sectors, and leverage the country's expanding digital-learning ecosystem.

## 2.2 Reasons for self-standing modules

The need for self/standing learning modules comes from two different directions, essentially:

- 1) Professionals with experience in one or another field related to embedded systems and chip design face now and then need to expand their base of knowledge because of the rapid development of technology. That is, there is a need for courses after graduating from university.
- 2) Universities often do not have the capabilities to cover all levels of technologies in needed detail and need additional sources of information to support specialized courses in curricula.

On-line smaller courses/modules would satisfy the need to provide additional education both for professionals and university courses. Based on that, eleven self-standing modules in three main areas - hardware design, embedded application design, and security and reliability.

# 3. Curricula Design

## 3.1 Programme objectives

The objective is to develop and deliver at least 10 self-standing modules specializing in embedded systems hardware and software design. The modules aim to address

these skills needs of Europe in these rapidly developing technologies and provide state-of-the-art content to uplift these skills in Europe. Some of the modules are design to be incorporated as parts of courses in the EIT Digital Master School's Embedded Systems Design Master's degree program.

The target audience of the modules is broad, including individuals with a work experience and job seekers. Their background can be either hardware or software developer, but they face needs to master both areas to manage modern complex designs.

## 3.2 List of self-standing modules

The list of self-standing modules is based on the preliminary list from the proposal. The list has been updated after analysis of the market needs.

The self-standing modules are independent and have its own content, assessment, and learning objectives defined. That is, there is no need to take them in on or another order. The modules can be combined based on their group topic (see below) to earn a groups certificate.

1. **Model Based Design and Verification (BME)** - This self-standing module aims to introduce modelling techniques for designing complex embedded systems consisting of both software and hardware components, interacting with the environment. The module focuses on system-level design, covering modelling techniques that can be used to represent requirements and design the system architecture that satisfies both functional and extra-functional (dependability and safety) requirements. At the level of system components, the modelling of the interfaces and state-based reactive behaviour is introduced. Regarding verification of the design, various model-based architecture analysis techniques, simulation, model checking, and model-based testing are presented. The module introduces SysML as system modelling language.
2. **Signal processing algorithms in embedded systems (BME)** - This self-standing module aims to introduce the most frequent signal processing algorithms implemented in embedded systems. The course thus focuses on processing time domain waveforms representing physical processes.
3. **The Art behind Reconfigurable and Adaptive Computing Systems (POLIMI)** - This self-standing module introduces the principles, technologies, and design philosophies behind reconfigurable and adaptive computing systems. As modern computing platforms grow increasingly complex and dynamic, traditional static hardware and software design approaches are no longer sufficient to ensure sustained performance and efficiency. This course explores how reconfigurable hardware, adaptive software, and self-aware architectures can dynamically adjust their configuration and behavior in response to changing workloads and environmental conditions. Students will discover how reconfigurable and adaptive systems enable runtime flexibility, performance optimization, and energy efficiency through concepts such as hardware/software co-design, runtime management, and dynamic partial reconfiguration. The course adopts an accessible approach, aiming to

democratize the understanding and use of FPGA-based technologies even among non-specialists.

4. **Design Methods for HW Security and Trust** (POLIMI) - This self-standing module provides a comprehensive introduction to hardware security and trust, focusing on threats that arise throughout the design, manufacturing, and deployment of integrated circuits and systems. The course explores the main categories of hardware attacks (including counterfeiting, hardware Trojans, transient execution attacks, fault attacks, and side-channel attacks). It presents state-of-the-art methods to prevent, detect, and tolerate them. Through theoretical and practical content, students will learn to embed trust and resilience into the hardware design process and understand how security-aware methodologies can be integrated across the entire IC lifecycle.
5. **Open-source ASIC design** (POLIMI) - This self-standing module introduces the principles and practices of open-source digital ASIC design flows, focusing on compiling high-level descriptions (HLS) into physical layouts (GDSII) using modern open-source EDA tools. The course begins with an introduction to the HLS-to-RTL flow, then walks through synthesis, verification, placement & routing, timing analysis, and benchmarking. Practical application is emphasized through hands-on exercises with open-source tools, including Yosys, Bambu, and OpenROAD.
6. **Computer Architectures** (POLITO) - This self-standing module aims at proving the students with the most relevant concepts about modern computer architectures for embedded systems. In particular, the module details the microprocessor architecture, highlighting the fundamentals of RISC processor cores, while detailing pros and cons of the processor pipeline organization. The positive impact of modern speculation mechanisms such as Branch Prediction Units and Cache Memories is also described in the course. Finally, an introduction to common peripherals is also provided. The RISC-V standard is used as an exemplification mechanism through the different topics presented in the course.
7. **Artificial Intelligence Safety** (POLITO) - This self-standing module provides a comprehensive introduction to the dependability and safety of artificial intelligence (AI) systems, with a special focus on their application in safety-critical domains. Students will explore the foundational principles of AI, including deep learning and the hardware architectures that support modern algorithms. The module covers the evolving landscape of AI standardization and industry regulations, such as the EU AI Act. Then, students will learn state-of-the-art solutions to assess, detect, and mitigate hardware-induced faults in AI systems. The course concludes with a discussion of future trends and challenges in the AI safety field.
8. **GPU Programming** (POLITO) - This self-standing module comprehensively introduces the Graphics Processing Unit (GPU), problem-solving, and High-Performance Computing (HPC) programming techniques. It begins with

fundamental concepts of parallel programming and C++ parallel features, transitions to understanding modern NVIDIA GPU architectures, and then delves deeply into the CUDA programming model. Key topics include writing basic CUDA kernels, managing the GPU memory hierarchy for optimal performance, handling thread synchronization, and applying these concepts to solve classical computational problems. The course emphasizes practical application through examples and hands-on exercises.

9. **Testing and Fault Tolerance** (POLITO) - Electronic systems used in safety- and mission-critical applications—such as in aerospace, automotive, railway, and biomedical sectors—must meet stringent reliability standards. This course introduces fundamental principles and practical techniques for ensuring system reliability through testing and fault tolerance. Topics include fault models, test generation, design for testability (DFT), and basic fault-tolerant design strategies. Emphasis is placed on digital systems and embedded platforms, focusing on both defect screening through testing and resilience through design.
10. **Embedded Platforms and Embedded Control** (UTU) - This standalone course focuses on the foundational skills needed for embedded software development using C. The course will cover C syntax and best practices relevant to low-level programming, hardware-software integration, and off-chip peripherals. The course will explore code optimizations for resource-constrained systems, debugging, and cross-compiling techniques.
11. **Chips and Smart and Real Time Embedded Systems Design** (UTU) - This self-standing online course explores the design and implementation of a soft-core processor on an FPGA. Students will learn the main components of a soft-core processor and will gain a solid understanding of the deployment and verification process of a fully functional embedded system.

### 3.3 Topic groups of self-standing modules

The self-standing modules have been divided into three topic groups based on their specialisation areas. This will help the learners to focus their studies in one or another direction important for their work/studies.

#### 3.3.1 Background

An embedded system consists of both hardware and software that may be built either from off-the-shelf or from full-custom components. Security and reliability are the third important issue when designing embedded systems. These three groups correspond to three main wider areas (see [Table 1](#)):

1. Hardware Design – design of embedded platforms and components. This includes single chips, multi-chip modules, PCB-s, and multi-PCB systems. Although embedded software is considered as a part of a platform, it is part of the next group.
2. Embedded Application Design – design of embedded software targeting different application fields. Abstraction levels vary from bare-bone

programming through signal processing algorithms to edge computing to cloud computing to use of AI.

3. Security & Reliability – problems and solutions at different abstraction levels and design phases for both hardware and software. Off-the-self and full-custom components require different approaches.

Table 1: Self-standing modules by topic groups

Course module	Hardware Design	Embedded Application Design	Security & Reliability
Signal processing algorithms in embedded systems		X	
Model Based Design and Verification			X
The Art behind Reconfigurable and Adaptive Computing Systems	X	X	
Design Methods for HW Security and Trust	X		X
Open-source ASIC design	X		
Computer Architectures	X		
Artificial Intelligence Safety			X
Testing and Fault Tolerance			X
GPU Programming		X	
Embedded Platforms and Embedded Control		X	
Chips and Smart and Real Time Embedded Systems Design	X	X	

### 3.3.2 Assessment and grading

In general, the assessment for a self-standing module consists of on-line quizzes after each part, plus on-line exercises. Evaluation focuses on comprehension, creativity, and the ability to connect theoretical principles with practical scenarios. In the case the assessment is different from the overall model, it has been stated in the corresponding self-standing module's description. To pass a module, the quizzes, test and/or exercises should be completed with 70% success rate (retake is possible).

### 3.3.3 Certification schemes

Certification is awarded upon the completion of either a single self-standing module or of three modules from a topic group. For single self-standing modules, the certification is issued by EITD and the partner university responsible for the module.

Certificate for a single self-standing module – the participant has finished all tasks and has been evaluated positively.

Certificate for a topic group of self-standing modules – the participant has finished three modules from a corresponding group.

## 3.4 Self-standing modules descriptions

In this section the developed and implemented self-standing modules are described in detail – motivation, aim, structure, learning outcomes, target audience, length and keywords.

### 3.4.1 Model Based Design and Verification (BME)

**Motivation:** Embedded systems are becoming increasingly complex: they are often distributed, intelligent, mobile, and cooperative, and they contain dozens of software and hardware components. Model-based design is considered as an approach to manage and master complexity in the design phase by providing well-defined methods and structured, unambiguous design artefacts (i.e., models with precise syntax and semantics, including graphical representation) that also allow automated verification and synthesis.

**Aim:** This self-standing module aims to introduce modelling techniques for designing complex embedded systems (cyber-physical systems). Today, most embedded systems are software-based, however, the design of the system should consider not only the software layer but also the physical components such as the computation platform, accelerators, sensors, and actuators, as well as the environment of the system. The module focuses on system-level design and introduces modelling techniques that can be used to represent requirements and design a system architecture that satisfies both functional and extra-functional (dependability and safety) requirements. At the level of system components, modeling of interfaces, interactions, and state-based reactive behavior is introduced. Regarding verification of the design, various model-based architecture analysis techniques, simulation, and model-based testing are presented. The module lays the foundations for the design competencies not only through the presentation of the modelling and verification methodology, but also through the introduction of a de-facto standard system modelling language, SysML. The skills and knowledge provided by this module can be used, among others, in the automotive and railway industry.

**Structure of the module:** The module includes 9 lectures:

1. Introduction to system modelling and the SysML language.
2. Requirements modelling, representing system context and uses cases.
3. Structure modeling using functional decomposition, interfacing, and encapsulation.
4. Platform modelling and allocation of functions to platform elements.
5. Architecture design solutions for safety and fault tolerance (types and roles of redundancy and fault handling in critical systems).
6. Model-based architecture analysis, using architecture trade-off analysis, and systematic analysis of fault effects.
7. Modeling reactive components using state machines.
8. Modelling inter-component communication and system-level behavior using message sequence diagrams.
9. Model based testing, simulation, and verification.

**Learning outcomes:**

Deliverable D.2.1

Project RESCHIP4EU  
(101158828)

- Explain the role of models in system design.
- Know requirements modelling techniques.
- Understand platform and architecture modelling, and apply architecture analysis techniques; know architecture design solutions for safety and fault tolerance.
- Use state machines to model reactive components and use sequence diagrams to model inter-component communication.
- Know the role and limitations of simulation and model-based testing.

**Target audience:** Engineers, MSc students.

**Length:** Approx. 18 hours of lecture videos, and additional 8 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** embedded systems, architecture modelling, safety and dependability, architecture analysis, modelling reactive components, modelling interactions, formal verification, model based testing.

### 3.4.2 Signal processing algorithms in embedded systems (BME)

**Motivation:** Embedded systems accomplish autonomous tasks by observing the physical environment, processing this information, and making decisions based on that. As the processing power increases continuously, more and more complex task can be accomplished within the embedded processors. This course focuses on the signal processing part of the system to get the audience familiar with the most common algorithms.

**Aim:** This self-standing module aims to introduce the most frequent signal processing algorithms implemented in embedded systems. The course thus focuses on processing time domain waveforms representing physical processes.

#### **Structure of the module:**

1. Introduction to embedded systems (architecture, components, role of signal processing)
2. Filtering: linear filtering (FIR and IIR filters, design methods, properties, realization considerations)
3. Spectral analysis: Discrete Fourier Transform (interpretation, properties, coherent/noncoherent sampling, different forms).
4. Sensor fusion: complementary filtering, Kalman filter for sensor fusion
5. Inverse filtering: numerical correction of static and dynamic distortions

#### **Learning outcomes:**

- The student is familiar with the concept of embedded system, its architecture, role
- The student can use spectral analysis tools, can interpret the results, and can systematically adjust parameters to minimize errors
- The student is familiar with filter design possibilities, and can design linear filter with required properties, can select appropriate filter structure and design strategy

- The student can design filters to combine the output of several sensors to improve accuracy by means of sensor fusion methods
- The student can numerically compensate the distortions of measurement system both in the case of linear, static nonlinear, and dynamic nonlinear cases.
- The student is familiar with ill-posed inverse problems and can implement regularization methods to suppress measurement noise.

**Target audience:** Engineers, MSc students.

**Length:** Approx. 12 hours of lecture videos, and additional 13 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** embedded systems, artificial intelligence, machine learning, recurrent networks, convolutional NN, sensor fusion, outlier detection, classification, clusterisation.

### 3.4.3 The Art behind Reconfigurable and Adaptive Computing Systems (POLIMI)

**Motivation:** Computing systems today face ever-changing demands driven by diverse applications in communications, AI, and embedded domains. These systems must adapt to new tasks, optimize for power or performance constraints, and even recover from unexpected conditions—all without human intervention. Reconfigurable computing provides the means for such adaptability: architectures can be partially reconfigured at runtime to change their functionality, optimize performance, or balance energy use. However, understanding these concepts often requires specialized hardware design knowledge. This course breaks that barrier, introducing reconfigurable and adaptive computing concepts in an intuitive way, enabling students and researchers from various backgrounds to explore how adaptability and self-awareness are shaping the future of computing.

**Aim:** The aim of this module is to provide students with a foundational understanding of reconfigurable and adaptive computing principles, emphasizing both conceptual and practical aspects. The course highlights the mechanisms enabling runtime adaptability—such as FPGA fabrics, system-level reconfiguration controllers, and design space exploration techniques—and discusses their impact on performance, flexibility, and sustainability. By the end of the module, students will be able to recognize when and how to apply reconfigurable architectures, understand their internal organization, and appreciate their role in the evolution of modern computing systems.

#### **Structure of the module:**

1. Introduction to Adaptive Computing Systems – Motivation for adaptability; dynamic behaviour in computing systems; examples of reconfiguration in everyday technologies.
2. FPGA Computing Systems Overview – Introduction to FPGAs; configurable logic blocks, interconnects, and bitstream configuration; FPGA-based reconfigurable computing principles.

3. Hardware/Software Co-design and Reconfigurability – Interaction between hardware and software layers; CAD improvements for reconfigurable computing; SoC and multi-chip reconfiguration models.
4. Runtime Management and Partial Dynamic Reconfiguration – Principles and classification of runtime reconfiguration; managing reconfiguration overhead; bitstream relocation and virtual homogeneity.
5. System Design Space Exploration – Exploring performance, area, and energy trade-offs in adaptive systems; methodologies for runtime adaptability; use cases and scenarios where reconfiguration improves efficiency.
6. Hands-on Examples and FPGA Demonstrations – Simplified FPGA configuration exercises; bitstream generation and manipulation; runtime adaptation examples.
7. Complex Adaptive Systems and Future Directions – Adaptive architectures as living systems; trends in self-reconfigurable hardware and autonomic computing; sustainability and long-term adaptability.
8. Closing Remarks and Reflections – The future of adaptive computing; societal and technological implications of self-aware systems.

### Learning outcomes:

- The student understands the fundamental principles of reconfigurable and adaptive computing systems.
- The student can describe how hardware and software modify their configuration dynamically in response to environmental or workload changes.
- The student identifies the architectural elements enabling reconfigurability, including reconfigurable fabrics and heterogeneous SoCs.
- The student analyzes the trade-offs among flexibility, performance, energy efficiency, and design complexity.
- The student applies basic methodologies and tools to conceptualize a simple adaptive or reconfigurable computing system.
- The student evaluates when and why adaptive architectures should be employed across various domains.
- The student reflects on the broader implications of adaptability and self-reconfiguration in computing innovation and sustainability.

**Target audience:** Students and researchers in computer and electronic engineering, as well as professionals and domain experts interested in understanding and applying reconfigurable computing concepts. The course is accessible to learners without prior hardware design experience.

**Length:** Approx. 16 hours of lecture videos, plus 8–12 hours of guided exercises and 60 hours of self-study, equivalent to 3 ECTS.

**Assessment & Grading:** Assessment consists of short quizzes after each module (40%) and a final mini-project (60%) where students conceptualize or simulate a reconfigurable computing system for a chosen application. Evaluation focuses on comprehension, creativity, and the ability to connect theoretical principles with practical scenarios.

**Keywords:** Reconfigurable Computing, Adaptive Systems, FPGA, Dynamic Reconfiguration, Self-aware Systems, Hardware/Software Co-design, Runtime Management, Design Space Exploration, Partial Reconfiguration.

### 3.4.4 Design Methods for HW Security and Trust (POLIMI)

**Motivation:** Hardware is the root of trust in modern computing systems. As integrated circuits become increasingly complex and globally distributed, ensuring their integrity and authenticity has become a major challenge. Globalized supply chains, untrusted manufacturing, and sophisticated adversaries have introduced new forms of vulnerabilities that cannot be addressed by software alone. Engineers and researchers must therefore adopt security-by-design principles at the hardware level. This course equips learners with the essential knowledge and techniques to understand, model, and mitigate hardware threats, preparing them to design resilient and trustworthy systems in both academic and industrial contexts.

**Aim:** The aim of this module is to provide students with the foundational concepts and practical design methodologies for securing hardware systems. It introduces key attack models and corresponding countermeasures, ranging from circuit-level defences to architectural and system-level design-for-trust techniques. Emphasis is placed on the integration of security verification and validation into the digital design flow, enabling students to understand trade-offs between performance, cost, and trustworthiness in hardware design.

#### Structure of the module:

1. Introduction to Hardware Security and Trust – Foundations, terminology, and threat models; vulnerabilities in the IC supply chain; reverse engineering and counterfeiting; taxonomy and detection of hardware Trojans.
2. Runtime and Microarchitectural Attacks – Transient execution vulnerabilities (Spectre, Meltdown); fault injection and scan-chain attacks; timing, power, and EM side-channel analysis.
3. Design Obfuscation, Monitoring, and Runtime Defence Techniques – Logic locking and hardware obfuscation; runtime anomaly detection; co-obfuscation; probabilistic and approximate hardware for resilience.
4. Side-Channel Countermeasures, Design-for-Trust, and Security Verification – Balanced logic styles, masking, noise injection; design-for-trust principles; secure supply chain integration; verification of security properties at RTL and gate levels.
5. Advanced Verification, Validation, and Metrics – Formal methods and dynamic validation; detection of hardware Trojans; design space exploration under security constraints; quantitative evaluation of security and trust metrics.

#### Learning outcomes:

- The student understands fundamental concepts and threat models in hardware security and trust.
- The student can identify and analyze hardware security risks across the IC supply chain.
- The student is able to evaluate the impact of various attacks, including hardware Trojans, transient execution, fault, and side-channel attacks.

- The student can select and apply appropriate countermeasures at the design and verification levels.
- The student is capable of integrating security-aware and design-for-trust methodologies into standard hardware design flows.
- The student can design, analyze, and evaluate secure and trustworthy digital architectures.

**Target audience:** MSc students in Electrical and Computer Engineering, PhD students in related fields, and engineers involved in digital hardware design, verification, and security evaluation.

**Length:** Approx. 16 hours of lecture videos, plus 8–12 hours of guided exercises and 60 hours of self-study, equivalent to 3 ECTS.

**Assessment & Grading:** Assessment consists of individual quizzes (40%) and a final design-oriented project (60%) in which students analyze a hardware system, identify potential vulnerabilities, and propose concrete countermeasures or verification strategies. The final evaluation emphasizes the student's ability to connect theoretical principles with practical design and verification methods.

**Keywords:** Hardware Security, Hardware Trust, Counterfeiting, Hardware Trojan, Side-Channel Attack, Fault Attack, Logic Locking, Design-for-Trust, Supply Chain Security.

### 3.4.5 Open-source ASIC design (POLIMI)

**Motivation:** The democratization of semiconductor design is a central challenge for innovation in the digital era. The rapid evolution of open-source EDA tools and the availability of open PDKs are making chip design accessible to a much broader community of engineers, researchers, and startups. However, the practical integration of these tools into a consistent and reproducible design flow remains a technical challenge. This course bridges the gap between theoretical ASIC design concepts and the practical use of open-source tools, enabling learners to understand and apply a complete open-source HLS-to-GDSII workflow suitable for real-world applications and research projects.

**Aim:** This module aims to provide students with a comprehensive understanding of open-source ASIC design flows, from high-level behavioral descriptions to physical layout generation. The course focuses on the key tools, data formats, and design steps required to move from algorithmic design to verified hardware implementation. Emphasis is placed on understanding timing closure, design constraints, and physical verification in open environments, preparing students to apply these methodologies both in academic research and industrial contexts.

#### Structure of the module:

1. Introduction to ASIC design flow and open-source EDA ecosystem
2. High-Level Synthesis (HLS) fundamentals and open-source toolchains
3. RTL synthesis with Yosys and technology mapping
4. Timing analysis and constraint management
5. Floorplanning and placement with OpenROAD
6. Routing and optimization techniques

7. Design Rule Checking (DRC) and Layout vs. Schematic (LVS) validation
8. From GDSII to tape-out: open PDKs and fabrication opportunities
9. Hands-on project: from C description to GDSII using OpenLane

### Learning outcomes:

- The student understands the complete digital design flow from HLS to GDSII using open-source tools.
- The student can synthesize and analyze RTL code using Yosys and perform place-and-route with OpenROAD/OpenLane.
- The student is able to identify timing and physical design issues, applying appropriate optimization strategies.
- The student gains practical experience with open PDKs and understands how open ASIC design enables education, research, and prototyping.

**Target audience:** Advanced MSc students, PhD students, researchers, and engineers seeking practical experience with open-source ASIC flows.

**Length:** Approx. 16 hours of lecture videos, plus 8–12 hours of guided exercises and 60 hours of self-study, corresponding to 3 ECTS.

**Assessment and grading:** Assessment is based on weekly quizzes (40%) and a final project (60%), where students demonstrate their ability to complete a small design from behavioral description to layout using the tools introduced in the course.

**Keywords:** Open-Source EDA, ASIC Design, HLS, RTL Synthesis, Static Timing Analysis, Place & Route, GDSII.

### 3.4.6 Computer Architectures (POLITO)

**Motivation:** Modern embedded systems are at the heart of countless technological innovations, powering everything from smart devices and industrial automation to automotive and medical applications. As these systems become more complex and performance-driven, a deep understanding of their underlying computer architectures is essential for engineers and computer scientists. However, the rapid evolution of processor designs, the shift towards open standards like RISC-V, and the increasing importance of advanced features such as pipelining, branch prediction, and cache memory present a steep learning curve for students and professionals alike.

**Aim:** This self-standing module aims to provide the students with the most relevant concepts about modern computer architectures for embedded systems. In particular, the module details the microprocessor architecture, highlighting the fundamentals of RISC processor cores, while detailing the pros and cons of the processor pipeline organization. The positive impact of modern speculation mechanisms, such as Branch Prediction Units and Cache Memories, is also described in the course. Finally, an introduction to common peripherals is also provided. The RISC-V standard is used as an exemplification mechanism through the different topics presented in the course.

### Structure of the module:

1. Introduction to embedded systems design and programming
2. Processor pipeline description

3. Hazards, stalls, exceptions, and other hard-to-implement processor characteristics
4. Programming models: Top-down Software Design
5. RISC-V assembly language
6. Interrupts and Real-Time events
7. Low power consumption design strategies
8. Memory hierarchy
9. Embedded system peripherals

### Learning outcomes:

- The student can identify, analyze, recognize, and determine the most relevant elements of a computer architecture when addressing a specific project for an embedded system.
- The student can implement a software project for embedded systems following a top-down strategy.
- The student knows how to develop efficient programs given a particular computer architecture, considering important aspects as power consumption and performance.
- The student is able to write low-level programs for RISC-V processor cores."

**Target audience:** Engineers, MSc students.

**Length:** Approx. 24 hours of lecture videos, and an additional 8 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** Computer Architecture; RISC; Branch Prediction Units; Cache Memories; exceptions; Low power and high performance implementation; RISC-V standard.

### 3.4.7 Artificial Intelligence Safety (POLITO)

**Motivation:** As artificial intelligence (AI) systems become increasingly integrated into safety-critical domains—such as healthcare, transportation, and industrial automation—their dependability and safety have emerged as urgent priorities. Unlike traditional deterministic software, AI-driven systems introduce new complexities and unpredictable behaviors, making it challenging to guarantee their reliability and trustworthiness in scenarios where failures can have catastrophic consequences. There is a growing need for engineers and practitioners who not only understand the foundational principles of AI and deep learning but are also equipped to address the unique safety and dependability challenges these technologies present. This includes navigating the evolving landscape of AI standardization and regulation, such as the EU AI Act, and mastering state-of-the-art techniques for detecting and mitigating hardware-induced faults in AI systems.

**Aim:** This self-standing module provides a comprehensive introduction to the dependability and safety of artificial intelligence (AI) systems, with a special focus on their application in safety-critical domains. Students will explore the foundational principles of AI, including deep learning and the hardware architectures that support modern algorithms. The module covers the evolving landscape of AI standardization and industry regulations, such as the EU AI Act. Then, students will learn state-of-the-art solutions to assess, detect, and mitigate hardware-induced faults in AI systems.

The course concludes with a discussion of future trends and challenges in the AI safety field.

### Structure of the module:

1. Introduction and Motivation
2. Hardware architectures for AI
3. European AI Act
4. AI Safety and Reliability
  - Vulnerability Setup
  - Error and Fault models in AI systems
  - Safety and Reliability Assessments
  - Uncertainty in DNNs
5. Safety and Reliability Improvement
6. Conclusions and Future Trends

### Learning outcomes:

- Describe the fundamental concepts of artificial intelligence, including its economic and social impacts, and recognize both its benefits and associated risks, especially in safety-critical systems.
- Explain the principles of dependability in complex systems and the role of deep learning in modern AI applications.
- Identify and compare the main hardware architectures used to run AI algorithms, such as GPUs, ASICs, TPUs, and hardware accelerators.
- Summarize the current AI standardization landscape and outline key international industry standards relevant to AI system development and deployment.
- Evaluate academic and industrial methodologies for assessing the safety of AI systems, with a focus on fault injection (FI)-based approaches at software, architectural, and physical levels.
- Distinguish between fault mitigation and detection techniques for enhancing the safety of AI systems, including both active and passive approaches and the use of specialized test libraries.
- Discuss emerging trends and future challenges in the field of AI safety, informed by the latest research and regulatory developments (e.g., the EU AI Act).

**Target audience:** Engineers, MSc students.

**Length:** Approx. 16 hours of lecture videos, and an additional 10 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** Artificial Intelligence Safety, Reliability, Standardization Landscape, European Regulation on AI, Fault Injections, Uncertainty, AI Architectures.

### 3.4.8 GPU Programming (POLITO)

**Motivation:** In today's data-driven world, the demand for computational power is rapidly outpacing what traditional CPUs can deliver. High-Performance Computing (HPC) and Graphics Processing Units (GPUs) have become essential tools for tackling

complex scientific, engineering, and artificial intelligence challenges that require massive parallelism and accelerated processing. As industries and research fields—from climate modelling to genomics, finance, and AI—continue to generate ever-larger datasets and more sophisticated models, the ability to harness the full potential of modern hardware is a critical skill for both students and professionals.

**Aim:** This self-standing module comprehensively introduces the Graphics Processing Unit (GPU), problem-solving, and High-Performance Computing (HPC) programming techniques. It begins with fundamental concepts of parallel programming and C++ parallel features, transitions to understanding modern NVIDIA GPU architectures, and then delves deeply into the CUDA programming model. Key topics include writing basic CUDA kernels, managing the GPU memory hierarchy for optimal performance, handling thread synchronization, and applying these concepts to solve classical computational problems. The course emphasizes practical application through examples and hands-on exercises.

### Structure of the module:

1. Foundations of Parallel Computing
2. Introduction to GPU Architecture
3. CUDA Programming Fundamentals
4. CUDA Memory Management
5. Synchronization and Execution Control
6. Case Studies and Further Topics

### Learning outcomes:

- Explain the fundamental concepts of parallel computing and the motivation for using GPUs.
- Describe modern NVIDIA GPU architectures' key components, characteristics, and memory hierarchy.
- Understand the CUDA programming model (Kernels, Threads, Blocks, Grids).
- Write, compile, and execute basic CUDA C++ programs.
- Effectively allocate and transfer data between the CPU (host) and the GPU (device).
- Analyze and optimize GPU memory access patterns (coalescing, shared memory usage, bank conflicts).
- Implement correct synchronization between CUDA block threads using and handling potential race conditions.
- Apply CUDA programming techniques to implement parallel solutions for classical computational problems (e.g., vector addition, matrix multiplication, prefix sum).
- Understand the concept of asynchronous execution using CUDA Streams.
- Utilize basic CUDA libraries and profiling tools.

**Target audience:** Advanced MSc students, PhD students, researchers, engineers, and software developers seeking high-performance computing skills.

**Length:** Approx. 16 hours of lecture videos, and an additional 10 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** GPU, CUDA, Parallel Programming, C++, High-Performance Computing (HPC), NVIDIA, GPGPU, Kernel Programming, Parallel Algorithms, Memory Hierarchy, Synchronization, Threading.

### 3.4.9 Testing and Fault Tolerance (POLITO)

**Motivation:** As our world becomes increasingly reliant on embedded electronic systems in sectors such as aerospace, automotive, railway, and biomedical engineering, the consequences of system failures grow ever more significant. In these safety- and mission-critical domains, even minor malfunctions can lead to catastrophic outcomes, including threats to human life, environmental harm, and substantial financial losses. Therefore, ensuring the reliability and resilience of these systems is not just a technical requirement but a societal imperative.

The importance of these topics is underscored by stringent international standards—such as ISO 26262 for automotive, IEC 61508 for industrial, DO-178C for aerospace, and IEC 62304 for medical devices—which mandate systematic approaches to reliability, testing, and fault tolerance throughout the product lifecycle. Mastery of these techniques is essential not only for regulatory compliance and market acceptance, but also for building public trust in technologies that increasingly shape our daily lives.

**Aim:** Electronic systems used in safety- and mission-critical applications—such as in aerospace, automotive, railway, and biomedical sectors—must meet stringent reliability standards. This course introduces fundamental principles and practical techniques for ensuring system reliability through testing and fault tolerance. Topics include fault models, test generation, design for testability (DFT), and basic fault-tolerant design strategies. Emphasis is placed on digital systems and embedded platforms, focusing on both defect screening through testing and resilience through design.

#### Structure of the module:

1. Introduction to Testing and Dependability
  - a. Definition and attributes of dependability
  - b. Fault models: stuck-at, bridge, open, delay, SEU, SET
2. Test Generation Techniques and Tools
  - a. Fault simulation
  - b. Automatic Test Pattern Generation (ATPG)
3. Design for Testability (DFT) Techniques
  - a. Scan-based design
  - b. Built-In Self-Test (BIST)
  - c. Boundary Scan (BS)
  - d. SoC testing standards: IEEE 1500 and IEEE 1687
4. Basics of Fault-Tolerant System Design
  - a. Redundancy techniques: hardware, information, time
  - b. Reliability evaluation methods: FMEA, radiation testing, fault injection

#### Learning outcomes:

- Understand the concepts of testing, dependability, and fault models
- Use software tools for test generation, fault simulation, and DFT insertion

- Understand and apply techniques for system hardening and reliability evaluation
- Design and implement BIST and utilize Boundary Scan techniques
- Understand standards for testing SoC devices (IEEE 1500 and 1687)
- Perform Failure Mode and Effects Analysis (FMEA)
- Evaluate the fault tolerance of a system through practical approaches

**Target audience:** Engineers, MSc students.

**Length:** Approx. 16 hours of lecture videos, and an additional 10 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** fault tolerance, testing, dependability, embedded systems, digital circuits, ATPG, scan, BIST, boundary scan, FMEA, fault injection, SoC testing, reliability.

### 3.4.10 Embedded Platforms and Embedded Control (UTU)

Embedded systems are the intelligent core of modern electronic devices, seamlessly integrating hardware and software to perform dedicated tasks efficiently, reliably, and often in real time. From household appliances, automotive control units, and medical instruments to industrial automation, drones, and AI-powered edge devices, embedded systems form the technological backbone of today's connected world. This course provides a comprehensive introduction to embedded systems design, programming, and control, combining theoretical foundations with hands-on experience on modern embedded platforms such as the ARM Cortex-A9, NVIDIA Jetson Nano, and Jetson TX2. Students will learn the architecture and components of embedded systems, low-level C/C++ programming, peripheral interfacing, and real-time operating system (RTOS) concepts. Through progressive labs and projects, participants will design, implement, and optimize embedded applications in both bare-metal, i.e., without OS, and with real-time OS environments, culminating in a final project demonstrating control and AI-capable embedded systems.

#### Structure of the module:

1. Introduction to Embedded Systems
2. Embedded C and C++
3. Embedded Hardware Interfaces
4. Embedded Control Fundamentals
5. Real-Time Operating Systems (RTOS)
6. ARM Cortex-A9 Platform
7. NVIDIA Jetson Nano and TX2
8. Project and Demonstration

By the end of the course, students will be able to:

1. Understand the architecture and components of modern embedded systems.
2. Develop embedded applications using C and C++ for real hardware.
3. Interface and control common peripherals (GPIO, UART, I<sup>2</sup>C, SPI, ADC, PWM, etc.).

4. Apply real-time operating system (RTOS) concepts for multitasking and timing-critical applications.
5. Understand the fundamentals of embedded control, including sensor feedback and actuator control loops.
6. Deploy and test embedded applications on ARM Cortex-A9, Jetson Nano, and Jetson TX2 platforms.
7. Optimize software for performance, power, and reliability.

**Target audience:** Engineers, MSc students.

**Length:** Approx. 16 hours of lecture videos, and an additional 10 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** Embedded systems, embedded C, code optimization.

### 3.4.11 CPU–FPGA Heterogeneous Computing for Artificial Intelligence Applications (UTU)

In today's era of rapidly growing AI workloads and data-intensive applications, heterogeneous computing has become essential for achieving high performance and energy efficiency. This course introduces the principles and practical techniques of heterogeneous computing, focusing on systems that combine CPUs and FPGAs to accelerate artificial intelligence (AI) and machine learning workloads. Students will explore how computation is distributed across heterogeneous architectures, how to design and implement hardware accelerators, and how to integrate them with CPUs using modern tools and frameworks. Through a combination of theory, design labs, and project-based learning, participants will gain hands-on experience developing FPGA-based AI accelerators, optimizing data movement and parallelism, and deploying end-to-end AI inference pipelines.

Structure of the module:

- Understand the architecture and design principles of heterogeneous computing systems.
- Develop FPGA hardware accelerators for compute-intensive tasks. Implement and integrate AI inference kernels between CPU and FPGA.
- Optimize performance, latency, and energy efficiency in heterogeneous platforms.
- Use industry tools (such as Xilinx Vitis, Intel OneAPI, or OpenCL) for hardware–software co-design.
- Evaluate trade-offs among CPU, GPU, and FPGA acceleration for AI workloads.

After completing the course, students will be able to:

Describe and analyze the design trade-offs in heterogeneous architectures.

- Implement and optimize FPGA-based accelerators for AI tasks.
- Partition and co-schedule workloads across CPU and FPGA.
- Use hardware–software co-design workflows to accelerate neural network inference.

- Evaluate acceleration performance using metrics like throughput, latency, and power.

**Target audience:** Engineers, MSc students.

**Length:** Approx. 16 hours of lecture videos, and an additional 10 hours for solving quizzes and exercises, equivalent to 1 ECTS.

**Keywords:** FPGA, soft-core processors, embedded systems, RISC, MicroBlaze, HLS.

## 3.5 Implementation platform

The RESCHIP4EU online courses are hosted on the EAC-ICARUS platform: <https://eit.icarus.education/reschip4eu/>

The EAC-ICARUS platform is significant for the project as it is the implementation platform for the self-standing modules designed in the curriculum. The RESCHIP4EU project aims to create these modules, which cover crucial areas like Hardware Design, Embedded Application Design, and Security & Reliability, for target audiences like engineers and MSc students. Utilizing EAC-ICARUS as the deployment tool enables the delivery of this specialized, high-demand education to the intended audience, fulfilling the project's goal of addressing educational gaps and industry needs identified in the market analysis for embedded systems. The platform, therefore, acts as the essential digital infrastructure for making the project's educational content accessible and functional.

The AI-powered, multilingual capabilities for videos, of the EAC-ICARUS platform is a crucial component for maximizing the reach and accessibility across the EU and beyond of the RESCHIP4EU project's specialized technical modules. By offering automatic transcriptions, translations of videos in multiple languages, EAC-ICARUS directly tackles the challenge of the "Digital Divide" and aligns with the goal of providing equitable access to knowledge. This capacity for multilingual delivery scales the impact of the curriculum far beyond a single language group, making the content usable by a truly global audience and promoting the worldwide development of in-demand skills identified in the market analysis.

Upon successful completion of all module assessments, the platform automatically issues certificates to learners, providing immediate recognition of their new competencies and supporting professional advancement. Instructors are granted real-time visibility into each learner's journey through an intuitive dashboard that tracks engagement, scores and time-on-task, allowing timely interventions and data-driven refinement of the learning experience. Its open architecture welcomes diverse pedagogical assets: textual readings, high-definition videos, synchronous interactive sessions, adaptive quizzes, so that every concept can be presented in the format that best accelerates comprehension and retention.

## Conclusions

The market analysis in D1.1 and D2.1 reveal that there is a strong need for smaller learning modules to support the education of embedded systems design in Europe.

Based on the analysis, 11 self-standing learning modules have been developed and implemented.

The developed self-standing learning modules are designed to address current and emerging skills needs in Europe’s semiconductor and microelectronics ecosystem, directly contributing to the objectives of the EU Chips Act and related capacity-building initiatives. These modules target two key audiences: professionals already active in the field who wish to upskill or reskill in line with rapid technological advances, and recent graduates seeking to enhance their employability and gain specialised expertise relevant to the labour market.

Beyond individual learners, the modules are also intended as flexible learning assets for universities and training providers. They can be integrated into existing degree programmes to complement or update curricula, ensuring that academic pathways remain aligned with industry demands. By bridging the gap between education, research, and industrial innovation, this approach supports the creation of a resilient talent pipeline and contributes to Europe’s broader strategy to strengthen its leadership in semiconductor design, manufacturing, and system integration.

## References

- [RESCHIP4EU] <https://www.eitdigital.eu/eu-collaborations/reschip4eu>
- [Saukonoja] Saukonoja, Olivia. *Needs Assessment in Embedded Systems Education*. 2025.  
<https://trepo.tuni.fi/bitstream/handle/10024/226662/SaukonojaOlivia.pdf>
- [EdTechIta2022] <https://www.osservatori.net/convegno/hr-innovation-practice/osservatorio-edtech-convegno-risultati/>
- [ICE2023] ICE, Automotive Sector in Italy, 2023,  
<https://www.ice.it/en/index.php/invest/sectors/automotive>
- [VerMarket2024] Verified Market Research, Italy Factory Automation and Industrial Control Market, 2024, <https://www.verifiedmarketresearch.com/product/italy-factory-automation-and-industrial-control-market>
- [PackMedia2024] Packmedia, Italian Packaging Machinery Sales Exceeds 9.5 billion Euro, 2024, <https://packmedia.net/facts-and-data/italian-packaging-machinery-sales-exceeds-eu95-billion>
- [EUDigSkills2024] EU Digital Skills and Jobs Platform, Italy Snapshot of Digital Skills, 2024, <https://digital-skills-jobs.europa.eu/en/latest/briefs/italy-snapshot-digital-skills>
- [Reuters2025] Reuters, Just 8 Percent of Italian Enterprises Using AI, 2025,  
<https://www.reuters.com/technology/just-8-italian-enterprises-using-ai-many-people-lack-digital-know-how-2025-05-21>
- [GVR2024] Grand View Research, Italy E-Learning Services Market, 2024,  
<https://www.grandviewresearch.com/horizon/outlook/e-learning-services-market/italy>
- [IMARC2024] IMARC Group, Italy Online Education Market, 2024,  
<https://www.imarcgroup.com/italy-online-education-market>