seame.space

sea:me

Software Engineering in Automotive and Mobility Ecosystems

Course Book

How to study SEA:ME - or offer it for free at your institution...

Overview

- 6 to 12 Month Advanced Studies to code the future of mobility
- Cost-free curriculum, learning environment, and equipment
- Internship / Work after core study - to practice the new skills
- Direct connections to experts, mentors, employers
- Experts from industry and academia co-developed and help update the curriculum. Including ones from Volkswagen, Microsoft, CARIAD, Bosch / ETAS, Red Hat, Micronova, IAV, NEC Laboratories, and more

What is SEA:ME?

SEA:ME (Software Engineering in Automotive and Mobility Ecosystems) is an innovative self-learning curriculum at master-level - in software engineering for mobility. It was first offered to students in 2022-23 in the European capital of mobility: Wolfsburg (Germany), home of VW.

This curriculum was built in Wolfsburg, with experts in mobility IT, coming from industry and academia. It is constantly extended, reviewed, updated, and further improved based on expert feedback and evolving trends affecting employers.

SEA:ME provides students with domain-specific skills and an understanding of foundational concepts of developing software solutions for mobility systems of the future. It includes a focus on in-vehicle technology (embedded systems), emerging autonomous driving systems, mobility ecosystems (smart / shared mobility, traffic management and smart city schemes, intermodal linkages of different mobility types, etc.). And in its current deployment season, 2024-25, it has added two new learning areas: Cybersecurity for Mobility, and Safety Critical Systems.

However, the defining features of SEA:ME go well beyond its content, of which parts can also be found in advanced university programs. What really sets SEA:ME apart are the following characteristics:

1. **Curriculum content built with industry and science experts** who volunteer their expertise toward designing, reviewing, updating and incrementally improving the curriculum in real time. In an open source setup that enables constant evolution of the program - matching current topics and emerging trends in the industry. This enables students to become more attractive to employers, to easily integrate into first internships / jobs, and to do realistic career planning while studying.
2. **Practice-focused team work instead of lecture attendance**, writing papers, or engaging in theory-laden discussions. SEA:ME learn-

ing is defined by working on projects, which present students with challenges to be solved in teams. This requires students to research, identify, select, and deploy their own tools (programming languages, software solutions). While being provided a dedicated lab environment with staff support and hardware to code and test on(to). Additionally, different students working on the same project / challenge will by design use different tools and come to different solution paths. And then learn from each other when sharing their approaches and paths with each other. All of this both intensifies and speeds up student learning outcomes. And their direct application to first jobs.

3. **"Peer Learning" methodology.** This approach to learning was pioneered by the likes of Montessori and later applied to IT education. We have brought it to the first master-level self-learning curriculum in mobility and IT. And further innovated by fusing it with team work on hardware in a dedicated lab environment.

4. **Team-based learning.** In SEA:ME, peer learning is taken up a notch, to working in dedicated student teams that make higher-level learning / challenges possible. By students combining their different (study) backgrounds, skill sets, and creativity types. And all of their individual experience before joining SEA:ME. Students team up for each project, and can request staff to greenlight reformed teams after completing each project. Thus, students remain flexible in choosing the learning projects that they individually want to master - and to find and join forces with other students of similar interest.

5. **Learning in an agile work format.** The teams of students are put into an agile team work framework, just as it is used in software developer teams. They are initially coached in the basics of such collaborative work, accompanied by academic staff that participates in regular stand-ups and reviews. And they are held to account regarding their continuous workflow and documentation.

6. **Graduates who are twice more qualified and satisfy employers.** For one, SEA:ME students stand out from regular IT graduates by learning foundational domain-specific coding concepts in mobility. In addition, they become self-reliant in problem solving and ideating, by learning how to complete projects on their own and researching their own solutions. Finally, they are easily integrated into existing developer teams via their learned and tested agile skill sets. All of this combined makes for an employment profile that a) leads to hiring over regular university graduates, and b) provides high employer satisfaction.

7. **Non-profit nature, open source development, and focus on replication.** SEA:ME has been developed in open source fashion and continues to be iterated and extended with support of experts, and increasingly of other educational institutions. To such institutions, SEA:ME is offered for replication, free of charge, as long as it is deployed in non-profit shape. The idea behind SEA:ME is to share it globally in open source fashion, to provide career and life opportunities for as many students as possible. (see "Replicate SEA:ME at your Institution").

SEA:ME has already successfully educated students in its first local installation in Wolfsburg - run, tested, reviewed, and improved by the creators of SEA:ME, in a specially set-up lab ("Wolfsburg Mobility Lab"). First graduates have earned internships and even direct job offers, from Bosch to CARIAD to Hyundai to regional start-ups. A cohort of 35 students is currently studying in Wolfsburg from July 2024 to June 2025.

Meanwhile, two first replications of SEA:ME are in launch mode in the Fall of 2024. In Korea, where the government funded COSS innovation consortium is integrating SEA:ME as a practice-related extracurricular activity to boost their students' job qualifications. And in Portugal, where the innovative educators from Shaken not stirred, are setting up a dedicated lab for advanced students in Porto. A third replication in Latin America is in sponsor acquisition phase. And we look forward to hearing from more institutions worldwide (see "Replication: Offer SEA:ME at your Institution").

Why Study SEA:ME?

This new curriculum is a one-of-a-kind opportunity to learn about software development for all types of mobility. We help you to educate yourself - to be a part of coding & shaping the future of how we all move. Also, it is completely cost-free.

There is nothing like this program yet - with its broad content, with your ability to choose which parts you focus on, and with our approach to learning that is community-based without teaching.

In just 6-12 months, you are trained job-ready - to start as a software developer in companies and institutions that provide software solutions for mobility. And you will already be connected with the partners that support our program - because they want to hire our graduates. This includes anywhere from traditional automakers, to new mobility / sharing providers, and to renowned global software companies that also provide backbone mobility software.

Our content is shaped by experts who work in the field - and who are a part of our community. This whole program is built around our learning community - built by us with students and with the experts that volunteer to support. We all boost each other in learning and innovating together.

While you are in the program, you learn from each other. Groups working on projects in their own focus areas help other groups understand what they are doing, and show each other their results and solutions. In this way, each student gets a broad overview of all mobility software areas even without working on all areas personally.

While you are acquiring technological expertise, we also introduce you to mobility concepts, how mobility affects our lives, what it currently looks like and how it can be changed for the better. Experts and a broad selection of learning materials will enable you to understand how different people in different places and circumstances experience mobility differently. And how you and all of us can leverage innovation and the potential of software to improve other people's lives and to make our own everyday lives more enjoyable through new types of mobility.

Each SEA:ME installation has its own lab space, where you can work on hardware and test your code until it works. You work in student teams, follow an agile workflow system, solve the challenges of each learning project as a team, and then present your solutions and get feedback from other students and staff. In this way, you speed up your learning process - and maximize your learning results. Because you create your own solutions, get advice when you get stuck - and then learn what alternative solutions the other student teams have found for the same challenge, and how this compares to your results.

The SEA:ME learning environment is fun, productive and fast. And qualifies you in the shortest amount of time to successfully apply for internships and jobs. Employers get to see you as more qualified than applicants from other learning backgrounds. Because you have gained the practice-related skills, the team work abilities, the experience in the agile workflow that companies use in their software developer teams. All of which university graduates in related programs usually lack. And because you have gained the coding skills and understanding of concepts that are "domain-specific" for mobility software development.

Which other software development graduates lack. This makes you easily integrable into existing developer teams - and therefore especially attractive to employers.

Apply to SEA:ME anytime, or log your interest, via: **https://seame.space/study-seame/**

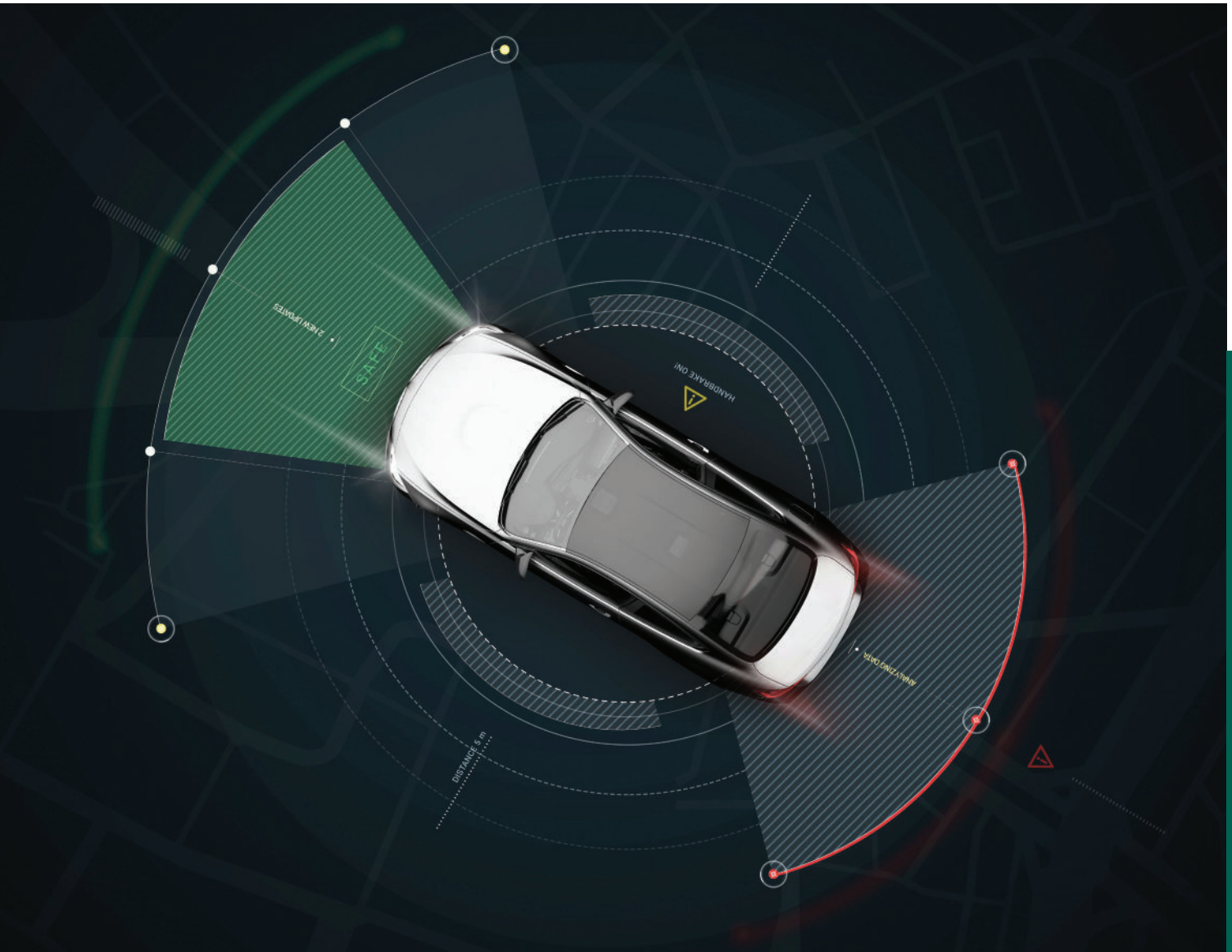
Why Study Mobility?

Mobility is one of the major factors shaping our lives and our surroundings. The future of our societies and how we live together. And not least how we can retain this planet as a living environment for all of us despite climate change.

At the same time, many improvements in mobility - for more access, choice, and freedom of movement - are becoming possible today by leveraging the innovative and disruptive potential of software.

Finally, the field of mobility software development offers some of the best career / personal development opportunities among any software-related fields of work. Today, companies and institutions in Germany and worldwide are competing with each other to hire, inspire, and retain specialized software developers. Not least because they constantly lack developers.

In the "Soft Curriculum" part of this Course Book, we lay out examples of the inspiration and opportunities that await you. And how you can both change your own life and make an impact on shaping our shared living environments locally and globally for the better.



How to Study SEA:ME

Below, we are providing in-depth information on the approach and on the modules / projects that form the new SEA:ME Curriculum.

SEA:ME students can choose which modules to focus on. In teams and individually, you may cover as much of the content as you can and want between 6 and 12 months. And you can adjust your choice while you are with us - based on new impressions and interests that you build. And after the learning process, you can apply the new skill sets in internship / work.

Overall, the contents presented in this course book - and provided in our new curriculum - can provide you with a comprehensive understanding of software engineering in automotive and mobility ecosystems. All of the content is created for students with a focus on peer-learning - i.e. learning that is driven by community, mutual inspiration, learning from experts and each other - but not from sitting in class listening to teachers.

By incorporating hands-on projects, case studies, and direct industry insights, you develop the skills and knowledge that you need to succeed in this dynamic and rapidly evolving field. Including the crucial job-related skills of a) learning how to learn, and b) learning how to work with others in order to innovate, ideate, and create.

Program Contents

Mobility Software Development

This section focuses specifically on the development of software for the mobility industry, covering topics such as embedded systems, software architecture for vehicles, and safety-critical systems.

The development of software for the mobility industry requires a specific set of principles and considerations due to the unique nature of the technical environment. Some of these core principles are:

1. **Embedded Systems:** Software for use in vehicles is often developed for embedded systems, which are small computer systems integrated into other products. The software must be optimized for low power consumption, small memory footprints, and real-time performance.
2. **Software Architecture for Vehicles:** The software architecture for vehicles must be designed to accommodate the specific needs of the automotive environment. This includes considerations for networked systems, the integration of sensors and actuators, and the need for high reliability and safety.
3. **Safety-Critical Systems:** Safety is a top priority in the mobility industry, and software for vehicles must be designed to meet the highest safety standards. This includes implementing fail-safe mechanisms, redundant systems, and rigorous testing and validation processes.
4. **Functionality and Usability:** Mobility software must provide a high level of functionality and usability for drivers and passengers. This includes considerations for human-machine interaction, navigation and infotainment systems, and advanced driver assistance systems.
5. **Interoperability:** The software must be designed to work seamlessly with other systems and technologies within the vehicle. This can be achieved by integration of different communication protocols and use of open standards where possible.
6. **Security:** The software must be designed to protect against hacking and other security threats, as well as ensuring the privacy and confidentiality of sensitive data.
7. **Scalability and Flexibility:** The software must be designed to accommodate future updates and changes to the vehicle, as well as the ability to easily integrate new technologies and features.
8. **Robustness and Resilience:** All software has to operate reliably and consistently in any conditions, including extreme temperatures, vibration, and electromagnetic interference.
9. **Certification and Compliance:** The software must meet industry standards and regulations for safety, security, and environmental sustainability. This includes certification from organizations such as ISO, SAE, and the European Union.

By following these core principles, software developers in the mobility industry can ensure that their systems are safe, reliable, and meet the highest standards of performance and quality.

Peer Learning Techniques

This section introduces you to the various methods and techniques for peer learning - including collaboration, teamwork, and mentorship - that you will use during your SEA:ME studies.

Peer learning is a form of education where individuals learn from each other in a collaborative and supportive environment. The goal is to create a community of learners who can share their knowledge, skills, and experiences to help each other grow and succeed. In the case of the SEA:ME Curriculum, this is further improved by including experts / practitioners who become part of the learning community and share their knowledge with students.

There are several methods and techniques for successful peer-to-peer learning, including:

1. **Collaboration:** This involves working together with others on a common goal or project. Collaboration can take many forms, including group projects, study groups, and peer-reviews. This allows individuals to learn from each other's perspectives, skills, and experiences and to develop their teamwork and communication skills.

2. **Teamwork:** Teamwork involves working together as a team to achieve a common goal. This can include sharing responsibilities, delegating tasks, and supporting each other. Teamwork helps individuals to learn how to work effectively with others, to develop leadership skills, and to build trust and cooperation.

3. **Mentorship:** Mentorship is a relationship between an experienced individual and a less experienced individual. The mentor provides guidance, support, and advice to the mentee to help them achieve their goals. Mentorship can take many forms, including one-on-one relationships, group mentorship, and online communities. This helps individuals to learn from the experiences and insights of others and to develop their own skills and expertise. In the SEA:ME curriculum, we make use of dedicated fellows - practitioners from industry, science and mobility consulting - who are volunteering to support our students on their learning journey. This provides direct access not just to expertise and feedback, but also to learning about what is demanded in real-life work environments where software for mobility is produced.

4. **Learning Communities:** These are spaces - online, offline, or hybrid - where individuals can connect and collaborate with each other. They allow individuals to learn from a wider range of perspectives and to connect with others who share their interests and goals [. They allow for self-directed shared learning, by participants teaming up around shared interests and ideas, driving their own projects forward and sharing with each other.

5. **Self-directed Learning:** Self-directed learning involves taking responsibility for one's own learning and development. This can include setting goals, creating a learning plan, and seeking out resources and support as needed. Self-directed learning allows individuals to develop their autonomy, creativity, and critical thinking skills.

6. **Gamification:** Gamification is the use of game design elements in non-game contexts to engage and motivate individuals. This can include using points, badges, and leaderboards to create a competitive and fun learning experience. Gamification can help individuals to develop their motivation and engagement with learning.

7. Action Learning: Action learning involves applying the knowledge and skills acquired in a learning experience to a real-world problem or challenge. This helps individuals to develop their practical skills and to gain experience in solving real-world problems.

By using these methods and techniques, peer-to-peer learning can help students to develop their knowledge, skills, and confidence in a supportive and collaborative environment. It also allows individuals to take an active role in their own learning and development and to connect with others who share their interests and goals. And it also trains students in how real-life work environments of software developers function - and in how they can succeed in collaborating and knowledge-sharing with their colleagues and become successful in what they do.

Reflective Practice and Self-assessment

This section provides students with opportunities for self-reflection and self-assessment, helping them to evaluate their own learning and progress and set goals for future growth.

Self-reflection and self-assessment are core aspects of peer-to-peer learning that allow students to evaluate their own learning and progress. These activities enable students to gain insight into their own strengths and weaknesses, identify areas for improvement, and set goals for future growth. The aim is to help students understand their own learning style, identify their individual learning needs, and develop effective strategies for meeting those needs.

Self-reflection involves taking time to think critically about one's own learning experiences and the outcomes of those experiences. It may involve asking questions like: What did I learn from this project? How did I approach this task? What could I have done differently? What worked well and what didn't? What can I learn by comparing my approach to that of other students, learning from each other?

Self-assessment, on the other hand, involves evaluating one's own performance against a set of criteria or standards. This may involve self-grading assignments, evaluating one's own work against a rubric, or using self-assessment tools like checklists or questionnaires.

The goal of self-reflection and self-assessment is to help students take ownership of their own learning and development. By critically examining their own performance, they can identify areas for improvement, set goals, and make changes that will help them achieve their goals. Additionally, these activities help students develop self-awareness, self-confidence, and self-efficacy, all of which are important skills that can be applied in various settings, both in and outside of the learning environment.

SEA:ME Warm Up

A 2-week intensive workshop to introduce peer learning and collaboration among students using GitHub is a great way to help students develop valuable skills in teamwork, software development, and open-source collaboration.

Goals:

1. Introduce students to GitHub and how it can be used for collaboration and version control.
2. Introduce students to the Qt framework and provide an overview of how it can be used to develop GUI applications using C++/Qt/Qml.
3. Encourage students to collaborate with each other and to help each other learn.
4. Provide students with the opportunity to practice open-source collaboration.
5. Foster a sense of community and teamwork among the students.

Objectives:

1. By the end of Day 1, students should be able to create their own GitHub accounts, create repositories, and make commits.
2. By the end of Day 2, students should have a basic understanding of the Qt framework and be able to create a simple C++ application.
3. By the end of Week 1, students should be able to work in pairs or small groups to develop a Qt application, and should be comfortable using GitHub to collaborate on their project.
4. By the end of Week 2, students should have joined an open-source project on GitHub that uses Qt, and should have made contributions to the project.

Throughout the workshop, students should be encouraged to communicate with each other and to help each other learn. By the end of the workshop, students should feel more comfortable collaborating with others and should have a greater appreciation for the value of peer learning.

more info on the warm-up

Tech Curriculum

This section gives an overview of how you can apply your knowledge and skills in real-world scenarios through group projects focused on software engineering in mobility ecosystems. Students choose which projects in which of the 5 below focus areas they want to work on. Toward this, students form teams, switch teams, and collaborate in and between teams from project to project. Academic staff reviews team performance and advises / greenlights on team change requests. In this way, students tailor their own learning path and what / how much they learn in up to 12 months. With support from staff that ensures maximum learning output.

Module 1: Distributed Embedded Systems

Distributed embedded systems in the mobility industry are complex systems that combine hardware, software, and networking components to create advanced and highly integrated systems for vehicles. These systems typically include a wide range of components, including microcontrollers, sensors, actuators, and communication networks.

One of the key features of distributed embedded systems is their use of multiple microcontrollers, each of which is responsible for a specific function or task. For example, one microcontroller may be responsible for controlling the engine, while another may be responsible for controlling the infotainment system. These microcontrollers are connected through a communication network, such as Controller Area Network (CAN) or Local Interconnect Network (LIN), which enables them to exchange data and collaborate to achieve the desired behavior of the vehicle.

Another important aspect of distributed embedded systems is the use of software. This includes both low-level software, such as drivers and firmware, and high-level software, such as user interfaces and applications. The software used in these systems must be highly reliable and performant, as it is responsible for critical functions such as safety and performance.

The use of networking technologies, such as Ethernet and VSOMEIP/CommonAPI, is also critical for distributed embedded systems in the mobility industry. These technologies enable vehicles to connect to other vehicles, to infrastructure, and to the internet, creating a highly connected and intelligent mobility ecosystem.

The development of distributed embedded systems in the mobility industry requires a deep understanding of a wide range of technologies, including hardware, software, and networking. Additionally, it requires close collaboration between hardware and software engineers, as well as designers and testers, to ensure that the systems are reliable, performant, and meet the complex requirements of the industry and users.

Overall, distributed embedded systems in the mobility industry are critical components of the modern vehicle, and play a key role in enabling the development of advanced and highly integrated systems for vehicles. Whether it is for safety, performance, connectivity, or entertainment, these systems are essential for delivering a high-quality and innovative driving experience.

Learning distributed embedded systems in the mobility industry through peer-to-peer educational projects is a great way to gain hands-on

experience and to develop practical skills that are highly relevant to current and future industry requirements. The following steps can be taken to get started:

1. **Familiarize yourself with the fundamentals:** Start by learning the basics of distributed systems, embedded systems, and the automotive industry. This may include topics such as automotive communication protocols (e.g. CAN, LIN, Ethernet), embedded software development (e.g. RTOS, programming languages, debug techniques), and automotive industry trends and requirements (e.g. autonomous vehicles, connectivity, electrification).
2. **Form a group:** Find other students with similar interests and form a group to work on educational projects together. Having a group can provide a supportive environment, facilitate collaboration, and provide opportunities to learn from one another.
3. **Identify project ideas:** Choose a project idea that aligns with your interests and skills, and that is relevant to the current or future requirements of the automotive industry. Some possible project ideas include developing a digital twin of a vehicle, building an autonomous vehicle control system, or designing a networked system for electric vehicle charging.
4. **Divide tasks:** Divide the project into smaller tasks that can be worked on by individual team members. This will make it easier to manage the project, keep track of progress, and ensure that everyone has a clear role to play.
5. **Start coding:** Start coding and implementing the project using the tools, technologies, and platforms that are relevant to the automotive industry (e.g. Qt, Wayland, VSOMEIP/CommonAPI, etc.).
6. **Test and refine:** Regularly test the project and refine it based on the results. This will help to identify and resolve issues, improve performance, and ensure that the project meets industry requirements.
7. **Present and share:** Once the project is complete, present it to others and share your experience and learning. This will provide an opportunity to showcase your work, receive feedback, and connect with others in the industry.

By following these steps, students can develop a deeper understanding of distributed embedded systems in the mobility industry and to gain practical skills that are in high demand in the job market. Additionally, working on peer learning educational projects can help to foster a sense of community and provide opportunities for collaboration and networking.

Project Descriptions

Hosted as Open Educational Resource (OER) on GitHub

DES Project 1 - DES Project - PiRacer Assembly

The purpose of this project is to provide students with hands-on experience in assembling and testing a PiRacer, a small, single-board computer-based racing car. The project will cover the basics of electronics, programming, and robotics, and will provide students with a foundation in these important areas of technology.

DES Project 2 - DES Project - Instrument Cluster

The PiRacer instrument cluster Qt application project is aimed at creating a real-time speedometer for a racing car. The application will run on a Raspberry Pi and receive speed data from a speed sensor via the Controller Area Network (CAN) protocol. This project will provide an opportunity for students to gain practical experience in software engineering, specifically in the areas of embedded systems, software architecture, and communication protocols. The project will also allow students to apply their knowledge of the Qt framework, which is widely used in the automotive industry for developing embedded applications. The successful completion of this project will demonstrate the students' ability to design and implement a real-world software solution, and their ability to effectively communicate their results..

DES Project 3 - DES Project - Head Unit

This project aims to develop a Head Unit Qt application running on a Raspberry Pi (RPi) alongside the previously developed PiRacer Instrument Cluster Qt application. The Head Unit application will provide additional features like ambient lighting, gear selection, and a media app. The project will use Yocto as the build system for the RPi..

DES Project 4 - DES Project - PDC System

The goal of this project is to enhance the Piracer by integrating a Park Distance Control (PDC) system, running on a Raspberry Pi (RPi) inside the head unit. The PDC system will utilize ultrasonic sensors to detect the distance between the vehicle and any obstacles, and provide an audible response to the driver. In addition, the full stack feature development will be integrated into the existing Yocto image, creating a comprehensive and fully functional system. The project aims to provide hands-on experience in developing and integrating advanced driver-assistance features, as well as provide a deeper understanding of the software and hardware involved in modern vehicles. The following sections will provide further details on the technical requirements, system architecture, software design, implementation, results, and references for this project.

DES Project 5 - DES-ME Project - SDV

In this project, you will learn how to design an SDV using Kubernetes/clusters by creating a network of nodes that run the hardware, software, and network components of the vehicle, and managing and orchestrating these components using Kubernetes. You will explore the different hardware and software components of an SDV, and learn how to integrate them using Kubernetes. You will also learn how to manage and orchestrate the network components of an SDV, ensuring that the different software components can communicate with each other and with other systems. Overall by the end of this project, you will have a solid understanding of how to design an SDV using Kubernetes/clusters.

Module 2: Autonomous Driving Systems

The term autonomous driving systems refers to advanced technology that allows vehicles to drive themselves without human intervention. These systems are designed to provide a high degree of safety, comfort, and efficiency, and to make driving more convenient and accessible for people.

Autonomous driving systems typically consist of a combination of hardware and software components. The hardware components include sensors, such as cameras, radar, and lidar, that are used to gather information about the vehicle's surroundings. This information is processed by the system's control unit, which includes a central computer and various microcontrollers. The software components of the system include algorithms that use the information gathered by the sensors to make decisions about the vehicle's movements and behavior.

Autonomous driving systems use various technologies and techniques to enable vehicles to drive themselves, including machine learning, computer vision, and control systems. Machine learning algorithms are used to analyze the vast amounts of data generated by the sensors and to make predictions about the behavior of other road users and the environment. Computer vision algorithms are used to interpret the images captured by cameras and to detect objects and features in the environment. Control systems algorithms are used to control the vehicle's movements, including steering, acceleration, and braking, based on the information gathered by the sensors and the predictions made by the machine learning algorithms.

The development of autonomous driving systems requires a multidisciplinary approach, bringing together expertise from fields such as computer science, electrical engineering, mechanical engineering, and more. The development process involves a combination of simulation, testing, and real-world deployment, as well as close collaboration between software engineers, hardware engineers, and designers.

Overall, autonomous driving systems have the potential to transform the mobility industry, providing safer, more convenient, and more accessible driving experiences for people. However, the development of these systems is complex and challenging, requiring close collaboration between various disciplines and a deep understanding of the technologies and techniques involved.

Learning autonomous driving systems through peer learning educational projects can provide students with hands-on experience and a deeper understanding of the current and future industry requirements. In this type of educational approach, students work together in small teams to design and develop projects related to autonomous driving systems. These projects are designed to be closely aligned with current and future industry requirements, ensuring that students gain the skills and knowledge that are most relevant and in-demand.

Here are a few steps that could be involved in learning autonomous driving systems through peer learning educational projects:

1. **Familiarization with key technologies:** Students start by gaining an understanding of the key technologies and concepts involved in autonomous driving systems, such as computer vision, machine learning, and control systems. Students also familiarize themselves with opportunities, limits, and ethical / governance questions around the use of autonomous vehicles.
2. **Team formation and project selection:** Students then form teams and select a project that they want to work on. The project should be relevant to autonomous driving systems and aligned with current and future industry requirements.
3. **Project planning and development:** Teams then plan and develop their projects, working together to integrate the various components of the autonomous driving system, such as sensors, control units, and software algorithms.
4. **Testing and validation:** Teams then test and validate their projects, both in simulation

and in real-world environments, to ensure that the system meets the desired requirements.

5. **Presentation and reflection:** Teams then present their projects to their peers and instructors, reflecting on what they learned and what could be improved.

Peer-to-peer educational projects provide students with the opportunity to work on real-world projects, develop their technical and team-working skills, and gain hands-on experience in developing autonomous driving systems. This approach also provides students with an understanding of the current and future industry requirements and helps to prepare them for careers in the mobility industry.

Project Descriptions

Hosted as Open Educational Resource (OER) on GitHub

ADS Project 1 - ADS Project - Autonomous Lane Detection with PiRacer

"Autonomous Lane Detection with PiRacer: A Hands-On Approach to Self-Driving Vehicles" is a peer-to-peer educational project aimed at introducing students and hobbyists to the exciting world of autonomous vehicles. The project focuses on using the Raspberry Pi and various sensors and actuators to build a low-cost yet powerful autonomous vehicle. The goal of the project is to demonstrate how a simple computer like the Raspberry Pi can be used to develop an autonomous vehicle that can detect lanes and maintain its position within them. With a hands-on approach, participants will learn about the fundamentals of autonomous vehicles and gain practical experience in building one themselves.

ADS Project 2 - ADS Project - Object Detection and Avoidance with PiRacer

In this peer learning project, you will learn how to build a state-of-the-art autonomous vehicle that can detect and avoid obstacles using the Raspberry Pi and various sensors. The PiRacer is an open-source platform designed to introduce you to the principles of robotics and computer vision, as well as the latest advancements in artificial intelligence and machine learning. Whether you are a beginner or an experienced engineer, you will find this project to be both challenging and rewarding, as you gain hands-on experience in the design, construction, and programming of autonomous vehicles..

ADS Project 3 - ADS Project - Road Surface Segmentation using PiRacer

Road Surface Segmentation is a crucial task in the development of autonomous vehicles. Accurate road surface segmentation is essential for maintaining the stability and safety of self-driving vehicles by providing a clear understanding of the available driving space in real-time. In this peer learning educational project, participants will develop an algorithm for road surface segmentation using the Piracer platform.

The objective of this project is to create an algorithm that can accurately segment the road surface in real-time video streams captured from a moving vehicle. The algorithm will be evaluated based on its ability to segment the road surface in various lighting and weather conditions, its computational efficiency, and its accuracy in the presence of complex objects, such as trees, buildings, and other vehicles.

Participants will use a combination of computer vision techniques and deep learning to develop their solution. They can use open-source tools and libraries, such as OpenCV, TensorFlow, and PyTorch, to implement their algorithm. The project will provide participants with hands-on experience in developing real-world computer vision applications, and it will help them build their skills in image processing, deep learning, and software development.

ADS Project 4 - Truck Platooning

In this project, you will develop a truck platooning system based on the techniques from the previous ADS projects (i.e., lane detection and object detection). The CARLA autonomous driving simulator is used, where we can deploy semi-trailer trucks equipped with sensors and actuators. You can deploy a number of such trucks in the virtual world. Each truck is equipped with a front camera and a front radar. The camera can be used to detect lanes and drivable areas. The radar can detect preceding vehicles including the preceding truck in the platoon. Each truck can be controlled by accelerating and braking as well as steering.

The objective of this project is to develop a truck platooning system with at least three trucks that can maintain a constant gap between them while maintaining the platoon within a lane based on the CARLA simulator.

Module 3: Mobility Ecosystems

Mobility ecosystems refer to the various components, technologies, and services that come together to enable safe, convenient, and efficient transportation. This includes everything from the vehicles themselves, to the infrastructure, services, and regulations that support them.

It also includes how environments in which mobility takes place are shaped and can be reshaped to enable different and new mobility offers and opportunities. Including by use of new opportunities provided by the power of software.

The following are some key components of a mobility ecosystem:

1. **Vehicles:** Automated, connected, and electrified vehicles are increasingly becoming the norm, and are a critical component of the mobility ecosystem. These vehicles are equipped with advanced technologies that support autonomous driving, vehicle-to-vehicle communication, and more.
2. **Infrastructure:** Infrastructure plays a critical role in enabling safe and efficient mobility. This includes everything from roadways and highways, to charging and fueling stations, to parking facilities.
3. **Services:** A wide range of services are essential for supporting the mobility ecosystem, including ride-sharing services, real-time traffic and navigation systems, and predictive maintenance services.
4. **Regulations:** Regulations play a crucial role in shaping the mobility ecosystem, including everything from vehicle safety and emissions standards, to licensing and liability laws.
5. **Data:** Data is increasingly becoming a key component of the mobility ecosystem, as it enables real-time monitoring, analysis, and prediction of various aspects of the transportation system, such as traffic patterns, vehicle performance, and more.
6. **Stakeholders:** A wide range of stakeholders are involved in the mobility ecosystem, including mobility service providers, automakers, technology companies, government agencies, smart city builders, and more.

The mobility ecosystem is constantly evolving and changing, driven by technological innovation, changing consumer preferences and evolving needs of both individual users and transportation / freight companies. As such, it is important for students and professionals in the mobility industry to have a deep understanding of the mobility ecosystem, as well as the various components and technologies that make it up. This knowledge can help them to develop new solutions and services that support the evolving needs of mobility providers and users.

Learning about mobility ecosystems, including Vehicle-to-Vehicle (V2V) communication, Vehicle-to-Infrastructure (V2I) communication, the Internet of Things (IoT), and other related technologies, can be done through a variety of peer learning educational projects that are close to current and future industry requirements. These projects can help students and professionals to develop a deep understanding of these complex systems and the challenges they face, and to gain the skills and knowledge necessary to innovate and contribute to the development of new and improved mobility solutions.

Here are a few suggestions for peer learning educational projects that can help students and professionals to learn about mobility ecosystems in the mobility industry:

7. **Developing a V2V or V2I communication system:** Students can work in teams to design, build, and test a V2V or V2I communication system that uses real-world data and simulates the interactions between vehicles and infrastructure in a complex transportation network.
8. **Creating an IoT-enabled vehicle monitoring system:** Teams can work on a project that involves building a monitoring system that

uses IoT technologies to collect data from vehicles, such as speed, location, fuel efficiency, and more.

9. **Building a connected and automated vehicle platform:** This project can involve designing, building, and testing an autonomous vehicle platform that integrates with various components of the mobility ecosystem, including V2V and V2I communication systems, IoT technologies, and more.

10. **Developing a predictive maintenance system for vehicles:** Teams can work on a project to build a predictive maintenance system that uses real-world data to identify and predict potential problems with vehicles, and to recommend actions to prevent them. This encompasses both vehicles transporting individuals, vehicles of ride-sharing / public transport, and vehicles of freight transport on roads and rail.

11. **Designing and building a traffic management system:** This project can involve developing a real-time traffic management system that uses data from various sources, such as sensors, cameras, and GPS, to monitor traffic patterns and make predictions about traffic flow, congestion, and more. The traffic management system needs to include both individual mobility, shared / public mobility, and transportation / freight management - linking and syncing all of the participants and modes of mobility into an efficient network that considers all users / stakeholders.

By working on these types of peer learning educational projects, students and professionals can gain hands-on experience with the technologies and systems that make up the mobility ecosystem, and develop the skills and knowledge necessary to contribute to the development of innovative solutions in the mobility industry.

Project Descriptions

Hosted as Open Educational Resource (OER) on GitHub

ME Project 1 - ME Project - Digital Trip Book

The use of technology in the automotive industry has revolutionized the way we drive, providing drivers with new tools and insights to make their journeys safer, more efficient, and more enjoyable. The DriveScore TripBook project takes this to the next level, by combining two innovative approaches to automotive software engineering: digital trip books and driving score assessments.

The digital trip book component of this project allows drivers to keep track of their driving history, including information about the routes they have taken, the distances they have traveled, and the driving scores they have received. This information can be used to identify patterns and trends in their driving behavior, and to make data-driven decisions about how to improve their driving style and reduce their carbon footprint.

The driving score component provides drivers with real-time feedback and recommendations on how to improve their driving style, based on the data collected by a Controller Area Network (CAN) Bus data recorder and analyzed using Artificial Intelligence (AI) / Machine Learning (ML) techniques. This innovative approach helps drivers to optimize their driving efficiency, increase road safety, and reduce their carbon footprint.

By combining these two approaches, the DriveScore TripBook project provides drivers with a comprehensive and data-driven view of their driving behavior, allowing them to make informed decisions about how to improve their driving style and enhance their overall driving experience.

ME Project 2 - ME Project - IoT-Enabled Fleet Management

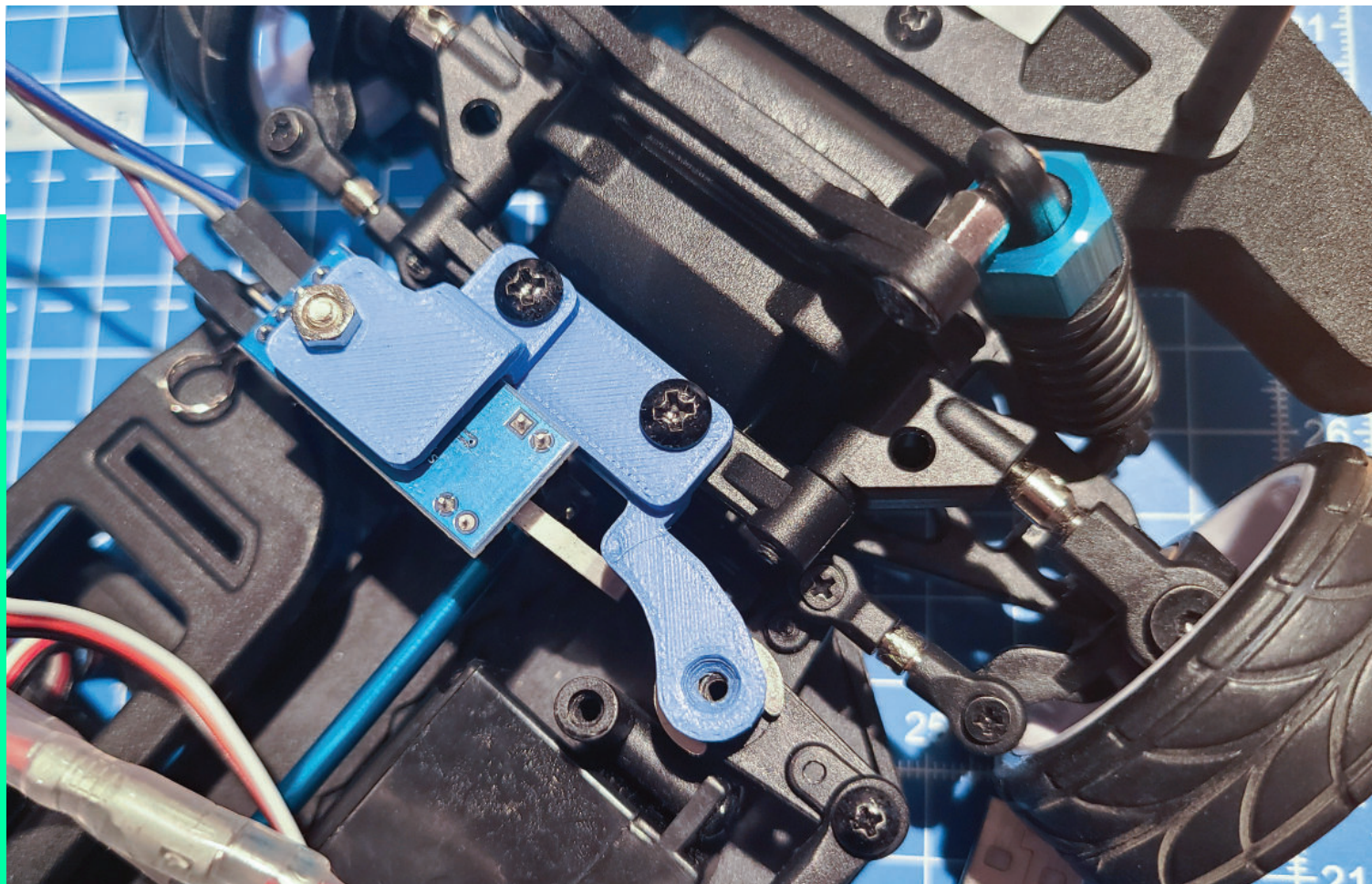
In the rapidly evolving world of logistics and supply chain management, new technologies are emerging that have the potential to revolutionize the way goods are transported and delivered. One such technology is IoT-enabled fleet management, which involves using connected devices and real-time data to optimize vehicle routes, reduce fuel consumption, and improve delivery times.

CARLA is a powerful simulation platform that provides a realistic environment for testing and developing autonomous vehicles. In this peer-to-peer educational project, participants will use telemetry data generated from CARLA simulations to explore the potential of IoT-enabled fleet management for improving logistics and efficiency.

The project will involve designing and implementing algorithms for IoT-enabled fleet management and evaluating their performance using the telemetry data. Participants will have the opportunity to work on real-world problems and gain hands-on experience with data analysis, simulation, and fleet management.

Through this project, participants will gain a deeper understanding of the benefits of IoT-enabled fleet management and the challenges and opportunities of implementing these technologies in the logistics and supply chain industries. They will also develop valuable skills in data analysis and simulation, making them well-prepared for careers in these exciting and rapidly evolving fields.

ME Project 3 - ME Project - Airport Experience



Module 4: Cybersecurity for Mobility

Automotive cybersecurity focuses on protecting vehicles from cyber threats. Modern vehicles are increasingly connected, integrating advanced technologies such as infotainment systems, GPS, and even autonomous driving capabilities.

This connectivity, while enhancing convenience and functionality, also exposes vehicles to potential cyber attacks. Here's a simple introduction to what automotive cyber security involves and why it's crucial:

Key Areas of Automotive Cybersecurity

1. **Vehicle Networks:** CAN Bus Security: The Controller Area Network (CAN) bus allows various components of the vehicle to communicate. Securing this network is essential to prevent unauthorized access or malicious commands.
2. **Telematics and Infotainment Systems:** Secure Communication: Ensuring that data exchanged between the vehicle and external systems (like GPS or music streaming services) is protected from interception or tampering.
3. **Firmware and Software Updates:** Over-the-Air (OTA) Updates: Securely delivering updates to a vehicle's software to fix vulnerabilities and improve performance without requiring a visit to the dealership.
4. **Data Privacy:** Driver Information Protection: Safeguarding personal data collected by the vehicle, such as location history, contact information, and usage patterns, from unauthorized access.

Why Automotive Cybersecurity is Important

- **Safety:** Cyber attacks on vehicles can lead to dangerous situations, such as unauthorized control of steering, brakes, or acceleration.
- **Privacy:** Vehicles collect vast amounts of personal data, which needs to be protected to prevent identity theft or tracking.
- **Reliability:** Ensuring the vehicle's systems are secure helps maintain their reliability and functionality, preventing disruptions caused by cyber attacks.
- **Trust:** As vehicles become more connected and autonomous, consumers need to trust that these systems are secure and will protect them and their data.

Best Practices

- **Regular Software Updates:** Keeping vehicle software up to date with the latest security patches.
- **Robust Authentication:** Using strong authentication methods to control access to vehicle systems.
- **Data Encryption:** Encrypting data both in transit and at rest to prevent unauthorized access.
- **Security Testing:** Regularly testing vehicle systems for vulnerabilities and addressing any found issues.

Project Descriptions

Hosted as Open Educational Resource (OER) on GitHub

MCS Project 1 - Over the Air (OTA) Update

The modern vehicle is rapidly transforming into a "driving computer," necessitating frequent updates to its software and firmware to manage and process data efficiently. Over-the-air (OTA) updates emerge as a pivotal technology in this context, enabling the wireless transmission and installation of new software or firmware directly from a server to the vehicle's system.

This project focuses on implementing OTA by updating new firmware with MQTT. Understand how subscribers, or clients, are authenticated by broadcasting through MQTT in real life through this deployment process. It will also be an approach that will lay the foundation for the development of OTAs applicable to various devices and high utility by increasing the understanding of OTAs in the MQTT method.

"In the contemporary automotive industry, software acts as the vehicle's heartbeat. The challenge of OTA updates is to maintain software currency while safeguarding vehicles against nascent cyber threats." This initiative is crucial for enhancing vehicle and user protection by ensuring the secure deployment of software updates.

By undertaking this ambitious project, we aim to investigate the OTA update process's capacity to safely upgrade vehicle software and firmware, thereby effectively countering all pertinent security threats. This endeavor will offer valuable insights for vehicle manufacturers and software developers ultimately equipping the automotive industry to navigate future challenges more adeptly.

Module 5:
Safety Critical Systems

In Mobility, safety-critical systems are specialized computer systems within vehicles that ensure key functions operate safely and reliably. These systems are crucial for controlling aspects such as braking, steering, and airbag deployment, where failure or malfunction could lead to serious accidents or fatalities.

Key Components of Safety-Critical Embedded Systems

1. **Vehicle Networks:** 1. **Electronic Control Units (ECUs):** ECUs are mini-computers embedded in vehicles to manage and control various functions, such as engine performance, transmission, and braking.

   a. Safety Role: Critical ECUs handle tasks like anti-lock braking systems (ABS) and electronic stability control (ESC), ensuring these systems perform correctly under all conditions.

2. **Sensors:** Devices that collect data from the vehicle's environment (e.g., speed, temperature, pressure).

3. **Actuators:** Components that execute commands based on sensor data (e.g., adjusting brakes, deploying airbags).

4. **Real-Time Operating Systems (RTOS):**

   a. Function: RTOS manage the timing and execution of tasks in safety-critical systems, ensuring they meet strict timing requirements.

   b. Importance: They guarantee that critical functions respond within milliseconds, which is essential for safety.

Why Safety-Critical Embedded Systems are Important

- **Safety:** Preventing Accidents:

  Quick Response: These systems ensure that functions like braking and airbag deployment happen almost instantaneously, preventing accidents or reducing their severity.

- **Reliability:**

  High Standards: Safety-critical systems are designed and tested to the highest standards to ensure they work reliably under all conditions.

- **Compliance:**

  Regulations: There are stringent regulatory standards (such as ISO 26262) that these systems must comply with, ensuring their safety and reliability.

Challenges in Developing Safety-Critical Systems

- **Complexity:**

  Modern vehicles can have dozens of ECUs and millions of lines of code, making development and testing complex.

- **Real-Time Requirements:**

  Ensuring that all tasks are performed within strict timing constraints is challenging but essential.

- **Interference and Fault Tolerance:**

  Systems must be designed to handle faults and interference without compromising safety, often through redundancy and fail-safe mechanisms.

Best Practices

- **Robust Design and Testing:**

  Using rigorous design methodologies and extensive testing to identify and mitigate potential failures.

- **Redundancy:**

  Implementing redundant systems so that if one component fails, another can take over to maintain safety.

- **Continuous Monitoring:**

  Systems continuously monitor their own performance and the performance of other systems to detect and respond to any issues in real-time

Project Descriptions

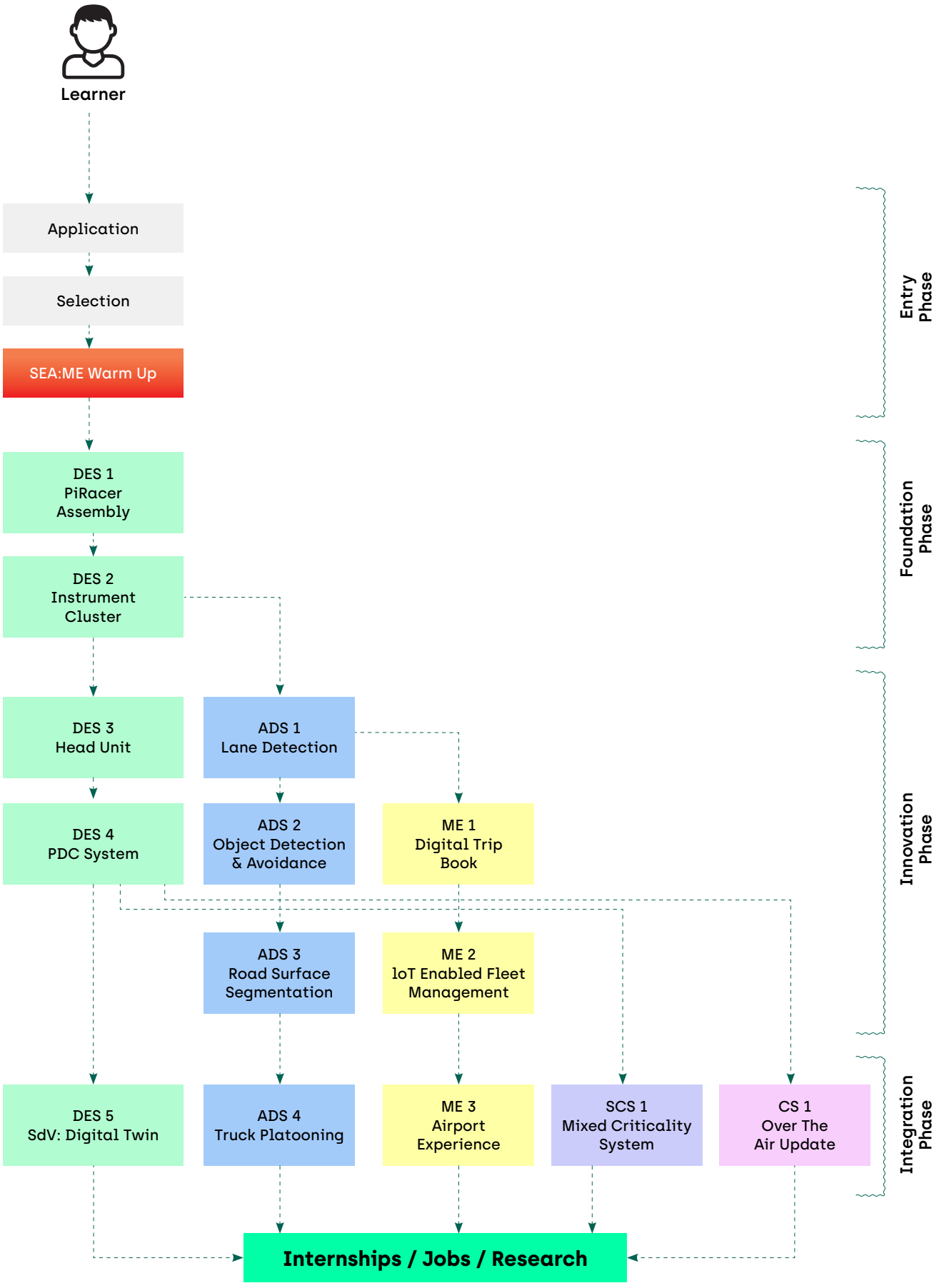
Hosted as Open Educational Resource (OER) on GitHub

SCS Project 1 - Mixed Criticality System

In the context of automotive safety-critical embedded systems, mixed-criticality systems refer to systems that integrate functions with different levels of criticality—some are safety-critical, while others are non-critical—onto the same hardware platform. This approach helps optimize resources, reduce costs, and improve efficiency but introduces challenges in ensuring that the critical functions are not compromised by the less critical ones.

.

Study Roadmap



Soft Curriculum

Mobility ecosystems are part of a rapidly growing and evolving industry that offers a variety of career paths and opportunities for software engineers. We provide you with insights on possible career paths - and open up opportunities for you to participate in workshops and events related to your journey into work as a mobility IT professional.

Building your Professional Network / Learning from Experts

The SEA:ME Curriculum features a diverse range of experts from the mobility industry, science, advocacy, consultancy, and politics as guest speakers, workshops leads and individually approachable mentors / fellows. They share their knowledge and advice and engage in discussion with us - providing students with insights into how mobility is shaped and why, into innovation and new ideas, and into best practices and opportunities for software development for mobility. Guest speakers and workshop leads will introduce different perspectives on mobility, as well as different ways of thinking about and imagining mobility and the role(s) it plays in our lives and in how we live together and relate to each other. They will provide our students with opportunities to re-imagine mobility and their own career and life opportunities.

Mobility Knowledge-Base

For the SEA:ME Curriculum, we are building an online knowledge-base of learning materials around mobility, together with experts. Once you have started in our curriculum, you will be part of growing this 360° compendium of expert knowledge around mobility. In addition to the tech fundamentals, this learning ecosystem provides you with multi-angle insights on what mobility is, how it has evolved, how it currently presents itself, and what are ideas for upcoming innovation and re-shaping of mobility solutions that will change how all of us use, access and benefit from mobility.

You will be able to access this learning ecosystem 24/7 - and to contribute by suggesting new content that would be beneficial to our learning community. We want to actively inspire and empower you to contribute to the shared learning content and to the spectrum of ideas and inspirations that make up our community.

At the same time, any content that copyright allows will be made available open source online, so that anyone anywhere can benefit from the information and insight that we collect and share in our community. In this way, we all become creators together and share with the world at large.

Plan your Path - Career Opportunities and Professional Development

Mobility ecosystems are part of a rapidly growing and evolving industry that offers a variety of career paths and opportunities for software engineers. We will provide you with insights on possible career paths - and open up opportunities for you to participate in workshops / trainings / events that are provided at 42 Wolfsburg.

Some of the most common career paths and opportunities in mobility ecosystems include:

1. Embedded Systems Software Engineer: Design, development, and testing of software for embedded systems in vehicles, such as infotainment systems, powertrains, and safety systems.
2. Vehicle Cybersecurity Software Engineer: Design, development, and testing of software to ensure the security and privacy of connected vehicles.
3. Electric Vehicle Software Engineer: Design, development, and testing of software for electric vehicles, including power management, battery management, and vehicle control systems.
4. Software Test Engineer: Testing software for vehicles to ensure that it meets quality standards and requirements, including functional testing, performance testing, and security testing.

5. Software Consultant: Providing expertise and advice on software development and technology to organizations in the automotive and mobility industry, including helping organizations to adopt new technologies and best practices.

6. Software Architecture Engineer: Define the structure and organization of software systems in vehicles. These software engineers work on defining the high-level design of software systems, including the organization of components, interfaces, and data flows.

7. Software Integration Engineer: Integrate software systems in vehicles to work seamlessly with each other. These software engineers work on integrating software systems in vehicles to ensure they function correctly and communicate effectively with other systems.

8. Human-Machine Interface (HMI) Software Engineer: Design and develop software systems for the interaction between vehicles and their occupants. These software engineers work on systems that allow drivers and passengers to interact with vehicles through displays, buttons, and other input devices.

9. Advanced Driver Assistance Systems (ADAS) Software Engineer: Design and develop software systems for advanced driver assistance features in vehicles. These software engineers work on systems that assist drivers in tasks such as lane keeping, automatic braking, and adaptive cruise control.

10. Automotive Cloud Software Engineer: Design and develop cloud-based software systems for the automotive industry. These software engineers work on systems that allow vehicles to store and process data in the cloud.

11. In-Vehicle Infotainment (IVI) Software Engineer: Design and develop software systems for in-vehicle entertainment and information systems. These software engineers work on systems that provide drivers and passengers with audio, video, and other forms of entertainment and information.

12. Vehicle Telematics Software Engineer: Design and develop software systems for vehicle telematics, including GPS tracking, remote diagnostics, and over-the-air updates. These software engineers work on systems that allow vehicles to communicate with remote servers for various purposes.

13. Mobile Application Developer for Mobility: Develop mobile applications for the mobility industry. These software engineers work on developing applications that allow drivers and passengers to interact with their vehicles, access information, and perform various functions from their mobile devices.

14. Vehicle Data Analyst: Analyze data generated by vehicles and their systems. These analysts work on analyzing data to gain insights into vehicle performance, usage patterns, and customer preferences.

15. Vehicle Software Support Engineer: Provide technical support for software systems in vehicles. These engineers work on providing support to customers and users of software systems in vehicles, resolving issues, and answering questions.

16. Software Development Operations (DevOps) Engineer: Ensure the efficient and effective operation of software development processes. These engineers work on automating software development processes, ensuring software quality, and managing software delivery.

These are just a few of the many career paths and opportunities available to software engineers in mobility ecosystems. The industry is constantly evolving, and new opportunities are emerging as technology advances and new business models emerge. Software engineers with a strong foundation in software engineering principles, a passion for technology, and a desire to work in a fast-paced and dynamic industry are well positioned to succeed in this field.

Explore - Emerging Technologies and Trends

There are several new developments and innovations in mobility ecosystems that our students will explore and prepare to contribute to, including:

1. Electric Vehicles (EVs) - The increasing focus on sustainability and reducing carbon emissions has led to a growing demand for electric vehicles (EVs). Software engineers play a crucial role in the development of EVs, working on battery management systems, charging infrastructure, and the vehicle's power electronics. They also develop software for the electric drivetrain, which includes the motor and the inverter, as well as the overall vehicle control systems.

2. Autonomous Vehicles - Autonomous vehicles have been a hot topic in recent years, and software engineers play a critical role in their development. They work on developing the perception, decision-making, and control systems that allow the vehicle to operate autonomously. This involves developing algorithms to detect and classify objects in the environment, make decisions based on the data collected, and control the vehicle's movements.

3. Connected Cars - The trend of connected cars is growing rapidly, and software engineers are working to develop new technologies to improve the driving experience. Connected cars use the internet to provide real-time traffic updates, entertainment systems, and other features, and software engineers work on developing the infotainment systems, the vehicle's internet connectivity, and the over-the-air update systems.

4. Vehicle Cybersecurity - With the increasing use of software in vehicles, cybersecurity has become a major concern. Software engineers are working on developing new solutions to protect vehicles against hacking, data theft, and other cyber attacks. This includes developing secure communication systems, secure boot processes, and secure data storage systems.

5. Advanced Driver Assistance Systems (ADAS) - Advanced Driver Assistance Systems (ADAS) are becoming increasingly common in vehicles, and software engineers play a key role in their development. ADAS systems, such as lane departure warning, adaptive cruise control, and automated emergency braking, use sensors and software algorithms to enhance safety and improve the driving experience.

6. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication - Software engineers are developing new technologies to enable communication between vehicles and between vehicles and infrastructure. This includes developing communication protocols, security solutions, and software systems for the exchange of information. The goal is to improve safety, reduce traffic congestion, and enhance the driving experience.

7. Artificial Intelligence (AI) - Artificial Intelligence (AI) is being used in the automotive industry to enhance the driving experience, improve safety, and optimize the performance of vehicles. Software engineers work on developing AI algorithms for applications such as computer vision, natural language processing, and machine learning, as well as integrating these algorithms into the vehicle's control systems.

8. Internet of Things (IoT) - The Internet of Things (IoT) is being used to connect vehicles and other devices, enabling new services and experiences for drivers and passengers. Software engineers play a crucial role in developing IoT-based solutions, working on developing the communication protocols, security solutions, and software systems for connecting devices.

These are just a few examples of the latest developments and innovations in mobility ecosystems. As technology continues to evolve, the role of software engineers in this field will continue to grow and change.

Societal Context, Change, and Future of Mobility

This section explores the ways in which mobility affects our lives. What current mobility modes / offers / limitations look like. And what innovations, changes and improvements the future of mobility can hold. It introduces why it's both exciting and valuable to work on the future of mobility and to help improve all of our lives in this way.

Here, students are able to explore their own learning paths around how mobility is designed and how it could be improved. And they can be part of shaping our communal learning by adding their own research, inviting experts as guest speakers and inspiring each other by co-organizing talks and workshops around the future of mobility. We want the SEA:ME learning community to be an open forum and lab for innovation and the building of ideas around how software solutions can be used for future mobility. And our students will co-drive this.

Mobility as we currently experience it is defined by long standing characteristics and continuing trends. These form the current baseline from which our approach to mobility sets out, and they include the following examples and solution approaches:

1. A sharp divide between the availability of mobility offers / choices between urban and rural environments. With cities having been shaped around automobile traffic since the 1950s, and rural areas in many countries

having been increasingly cut off from public transport offers. Smart networks of various mobility services can offer access to mobility in urban and rural environments and begin to connect these by making use of the intermodal spaces between them.

2. Dominance of individual traffic in personally owned cars, as well as of freight transport on roads. Since the second half of the 20th Century, living environments across the globe have increasingly been shaped around individual vehicles on roads, with personal movement on roads defined along the lines of freedom. In smart software-driven environments of the 21st Century, thinking and action along the lines of freedom can be re-imagined and re-shaped as freedom of access and freedom of choice for everyone who needs or wants to consume movement options.

3. Continued growth of car ownership in urban areas, and ongoing extension of road space. While science has identified this dynamic as one of "induced traffic", where increased road space repeatedly leads to more vehicles and traffic - and thus to new calls for yet more road space. Innovative 21st Century thinking and planning can make use of the powers of software to create efficient models and analyses of traffic flow, predictions of which effects which mobility modes can have, and help shape traffic in ways that adapts to the needs of residents more than residents continuing to adapt to assumed needs of traffic.

4. Preventative pricing levels and complicated booking models for public transport. Software-driven, networked public transport offers - across vehicle types / sizes, roads and rail - can boost ease of use, speed and comfort of shared transportation offers. Integrated booking, navigation, and authentication apps can help users to easily combine different mobility modes, types of vehicles, and services - by identifying what is most efficient and comfortable for each single trip from Point A to Point B.

5. Low availability of space for non-automotive vehicles, and large amounts of space dedicated to parked vehicles that have been averaged as being parked for 96% of time per day. More choice and easier access to different mobility modes can contribute to freeing up public spaces that have been increasingly used for parking over the last decades. Smart shared vehicles can also help avoid issues around parking in spaces that are not legal, where parked cars block weakest participants in mobility such as wheelchair users, blind citizens, physically frail / limited individuals or children on bikes.

6. Lack of barrier-free access to public transport, which leads to exclusion even where mobility choice generally exists. Modern networked and user-oriented transportation networks can offer different types of vehicles and access modes, and make these easily available and identifiable to users with special needs. They thereby can boost freedom of movement and freedom of choice for individuals that so far are strongly limited in where they can move at which time and in which ways.

7. Globally, there remains a constant increase in levels of emissions generated by traffic on ground, in water and in air. With these emissions (fine particles) being generated not just by combustion engines, but in large part by vehicle parts such as tires and brakes, overall volume of traffic / vehicles and efficient solutions for passenger and freight traffic are of importance. Smart mobility solutions can address emission levels, for example by easing use of low emission services (shared / rail) and by making trips in higher emission vehicles (individual / road) more efficient by use of shortest possible routes and efficient driving styles.

8. Effects of globalized supply chains and of consumer trends toward online ordering and returning of products are continuing to fuel increases in freight transport.. Innovation in mobility management can help alleviate some of the effects. For example by addressing the "last mile" where shipments are brought to final recipients after long-haul transport, or by building efficient long-haul transport networks consisting of road and rail, in which goods are easily switched between vehicle types.

Given that established characteristics and ongoing trends in mobility currently leave much to be desired, and given that traffic circumstances are getting ever more intense and difficult around us, it is only fortunate that at this point in time there is a whole set of new opportunities starting to open up. The opportunities provided by software are beginning to offer completely new ways of imagining and shaping mobility solutions. New ways of thinking about the kinds of mobility that would give more of us access,

and all of us more choice. So that each one of us could choose where and when we want to personally drive a car, or to sleep or work while riding in an autonomously driving vehicle, or to connect to public transport or ride-sharing, pick up a rental cargo bike, etc. etc. And how we could easily and cheaply combine any one of these modes and vehicles in one comfortable, quick and seamless trip from Point A to Point B.

At the same time, all of us living in cities will be able to benefit from more shared and diverse mobility options also in ways that transcend our own trips. For example, where space for individual parked vehicles becomes available again to all of us for use as an enjoyable public space for leisure, sports and play. Where more mobility choice and more shared mobility evolve, our shared urban spaces increase in size and the liveability of our environments increases. And where mobility choice increases in rural areas, people have more opportunities to connect with each other and to live in environments that are still different from cities but also easily accessible anytime.

Finally, with more mobility offers and choices available, all of us will be able to move anywhere anytime - including those of us who are limited in their mobility by health status, age, disability or other circumstances. And those of us who currently have to make the choice between a type of mobility that they feel uncomfortable with or no mobility at all can receive actual choice, safety and comfort.

The future of mobility looks bright because of the new potential that is emerging now for the first time in decades - a potential that is largely driven by what software can offer. Therefore anyone who becomes a software expert for mobility today not only has plenty of choice in terms of the direction / work area to go into, but also has plenty of opportunity to be a part of something that improves their own life and the lives of us all. Makes our lives both easier and more enjoyable - by leveraging the strengths of code.

Join SEA:ME

Applying to SEA:ME is easy. You can use the dedicated form on our website at https://seame.space/study-seame/

In order to join, you need basic coding skills and / or experience in mobility engineering. A background in mobility studies plus basic coding skills is especially appreciated.

We aim to build a learning community of diverse personal backgrounds and diverse focus areas. By having students with a strong interest in either embedded systems, autonomous driving, or mobility ecosystems - or any combination of these - we can build a community of knowledge sharing and mutual inspiration that maximizes your learning results and overall development as a professional in the field of mobility programming.

If you are interested in studying SEA:ME, check the curriculum website at www.seame.space for information on the locations and institutions where SEA:ME is currently offered. For each location, a link to its further information is included along with a short description. Each local installation of SEA:ME has its own admissions requirements - but all of them have to offer SEA:ME to the students for free.

For general information about SEA:ME directly from its inventors, please write to **contact@seame.space.**

Offer SEA:ME at your institution

SEA:ME is designed to be shared - for maximum impact on improving mobility - and for improving the lives of more students. We want to put SEA:ME on the global map as an example of how to inspire and qualify students both for good career / life opportunities and for becoming impactful in shaping better mobility for the future. Including mobility options beyond individual use of cars and trucks. And very much including the joint innovation potential of the two great pillars of current mobility trends: software and electric engines.

This is why we have designed it as a non-profit offer to other institutions that want to educate students for free. It can be offered to existing students of an educational institution who want to specialize in mobility software development, or as an open admissions process where anyone can apply. The creators of SEA:ME in Wolfsburg have successfully trialed an open admissions structure. By vetting candidates from any earlier education background in a "rolling admissions" process of individual assessment and continuous admission throughout the year. So that students can join a running program season at any time if they are sufficiently pre-qualified. With experienced students then supporting the newcomers toward a successful study start. At the same time, of course internal student selection or competitive admissions toward a fixed group starting date may work just as well.

Why use SEA:ME?

By using the SEA:ME curriculum, students can be educated in a very short time of 6-12 months, with current learning content that keeps getting updated to stay relevant to industry needs and ahead of emerging trends. Students are trained in practical problem-solving skills, in team work abilities, and in the structured agile workflow of real-life software developer teams. By training them in this way, educational institutions can both provide new opportunity to students as a charitable effort and at the same time work with industry partners in need of new talent.

How to set up SEA:ME

Institutions wanting to offer this curriculum set up a lab environment with learning hardware and furnishings, as well as basic pedagogical support staff (learning facilitators, usually re-

cent IT graduates). Each replication should also have an experienced coordinator who comes from the mobility coding domain and has work experience in designing software for relevant applications. Students learning in SEA:ME can only successfully solve the advanced-level learning projects if they a) work in teams, b) follow a structured agile workflow and reporting requirements that are reviewed by staff, and c) can call on advice if they get stuck. Then - given such a lab learning environment with agile processes and basic staff support - students can thrive on their own. Due to the strength of the curriculum design and the testing and improvement it has already undergone.

How to offer SEA:ME to students

Sharing SEA:ME is done in a non-profit / non-commercial setting. Meaning that institutions using it do also offer it free of charge to students, and do not charge them or third parties direct fees. Of course SEA:ME deployment can be supported by sponsors / donors, as long as these do not tie direct conditions to their support (such as: "How many graduates do we "get" for sum x?").

By definition, SEA:ME students remain free to choose how and where to employ their new skills. However: Sponsors / donors can be given special exposure to students, and can be encouraged to actively engage with students. By providing workshops, industry briefings, "knowledge-sharing" sessions with professionals working in the jobs that students train for, company visits, hackathon challenges, etc. In such an organic way, employers inspire SEA:ME students to apply to their jobs or internships. Because the students see the employers and their staff members as great places and great people to work with.

How to get started

Toward the goal of spreading SEA:ME learning opportunities, its owners offer advice and support to institutions interested in replicating the curriculum. We are proud of our product, and happy to inspire others and help them get started.

So: If you are interested in using SEA:ME or learning whether it can work for you as an institution, just approach us at **contact@seame.space**.

sea|me

software engineering in automotive
and mobility ecosystems

42 Wolfsburg/Berlin e.V.

Sitz:
Wolfsburg

Registergericht:
Amtsgericht Braunschweig, VR Nr.: 201961

Porschestraße 2 c, 38440 Wolfsburg
contact@42wolfsburg.de

Präsidium (Vorstand i.S.d. § 26 BGB):
Ralph Linde, Andrea Morgan-Schönwetter

Direktorin (Besondere Vertreterin i.S.d. § 30 BGB):
Stephanie Levy

42 Wolfsburg/ Berlin e.V
Porschestraße 2c
D-38440 Wolfsburg, Germany
E-Mail: contact@42wolfsburg.de

seame.space