

3.3.4 Software Engineering

Software Engineering
Modulübersicht
EDV-Bezeichnung: KIIB340S
Modulverantwortliche(r): Prof. Dr.-Ing. Reiner Kriesten
Modulumfang (ECTS): 6 CP
Einordnung (Semester): 3. Semester
Inhaltliche Voraussetzungen: Grundlagen der Informatik und Mikrocomputertechnik
Voraussetzungen nach SPO: Nach SPO sind keine formellen Voraussetzungen erforderlich.
Kompetenzen: Die Teilnehmenden können Anwendungsfunktionen cyber-physischer Systeme konzeptionieren, modellieren und implementieren, indem sie <ul style="list-style-type: none"> a) das Vorgehen zur modellbasierten Entwicklung und zum Rapid Prototyping von Anwendungsfunktionen anwenden, b) Techniken zur Modellierung von Software- und Umgebungsmodellen eigenständig anwenden und auf neue Problemstellungen transferieren, c) einfache Anwendungsfunktionen cyber-physischer Systeme modellieren, simulativ testen und diese graphischen Modelle in Code überführen, d) State-of-the-art Modellierungstools der Industrie verwenden, e) Fallstricke und Gefahren in Implementierungen erkennen und vermeiden, f) Zustandsautomaten, Flussdiagramme und Kennlinienfelder detailliert verstehen, anwenden und diese modellbasierten Diagramme in Code realisieren, g) Einfache Black-Box Tests einzelner Teilfunktionen und Gesamtfunktionen modellbasiert entwerfen, h) die Besonderheit von Echtzeitsystemen gegenüber normalen Betriebssystemen kennen und beschreiben, i) Sicherheitsaspekte bei Fahrzeugfunktionen benennen können.
Prüfungsleistungen: Klausur, 120 Minuten
Verwendbarkeit: In diesem Modul werden die Grundlagen zur Modellierung und Implementierung software-intensiver Systemen auf Basis konventioneller Modellierungstechniken gelegt. Damit wird insbesondere eine Abgrenzung dieser konventioneller Lösungskonzepte zu Ansätzen basierend auf Künstlicher Intelligenz gelegt und deren Eigenschaften wie Nachverfolgbarkeit und Zuverlässigkeit erlernt.

Lehrveranstaltung: Software Engineering
EDV-Bezeichnung: KIIB341S
Dozierende(r): Prof. Dr.-Ing. Reiner Kriesten
Umfang (SWS): 3
Turnus:
Art, Modus: Vorlesung, Pflichtfach
Lehrsprache: Deutsch
<p>Inhalte: Die Vorlesung mit integrierter Übung lehrt Studierende, automotiv bzw. autonome Funktionen gemäß moderner Simulationstechniken zu entwerfen und zu implementieren. Neben dem Blick auf die Modellierung legt diese Vorlesung ebenfalls ein Augenmerk auf typische Implementierungsaspekte sowie die konkrete Integration in verteilte Steuergeräte und deren Kommunikation. Die Vorlesung geht hierbei auf folgende Inhalte ein:</p> <ul style="list-style-type: none"> • Einführung in Graphentheorie, Flussdiagramme und Zustandsautomaten, • Übersetzung technischer Anforderungen in oben genannte Modellansätze, • Realisierung von Flussdiagrammen und Zustandsautomaten in Matlab/Simulink und in prozedurale, automotiv Programmiersprachen, • Validierung und Regression Testing von Logiken, • Software-Entwicklung mit verteilten Modulen und Abhängigkeiten, • Rapid-Prototyping der Funktionen. <p>Dabei wird anhand einer konkreten automotiven Funktion ein Modell zur Funktionsweise der Software und der notwendigen Umgebung realisiert, getestet und weiter unter Verwendung von S-Functions sukzessive in Code entwickelt und dieser auf einen Mikrocomputer mit passender Hardware, Sensorik und Aktorik portiert.</p>
<p>Empfohlene Literatur:</p> <ul style="list-style-type: none"> • Skriptum und Foliensatz zur Vorlesung. • Stateflow User's guide: Download unter www.mathworks.com. • Online-Hilfe und Tutorial's für Stateflow: siehe Hilfe innerhalb Matlab/ Simulink.

Lehrveranstaltung: Modellbasierte Entwicklung automobiler Steuergerätesoftware
EDV-Bezeichnung: KIIB342S
Dozierende(r): Dr. Peter Dencker
Umfang (SWS): 2
Turnus:
Art, Modus: Vorlesung, Pflichtfach
Lehrsprache: Deutsch
<p>Inhalte: Die Vorlesung mit integrierter Übung lehrt Studierende in einem ersten theoretischen Teil die charakteristischen Eigenschaften automotiver Software im Unterschied zu gängiger PC Software, insbesondere die Eigenschaften der eingebetteten Software, hoher</p>

Sicherheitsanforderungen, hoher Anforderungen an die Echtzeiteigenschaften, hoher Anforderungen an die Verfügbarkeit, verteilte Software, hoher Hardwarekostendruck und lange Produktlebenszyklen.

Hieraus werden weiter die sich daraus ergebenden Anforderungen an moderne Entwicklungsprozess-Modelle mit Hinführung zur Simulation, Rapid Prototyping und Hardware in the loop (HIL) erläutert. Die zusätzlichen Anforderungen von autonomen Systemen werden thematisch unter den Sicherheitsaspekten Vertrauen, Zuverlässigkeit und Manipulationsschutz diskutiert.

In einem zweiten praktischen Teil lehrt die Veranstaltung die Studierenden, (automotive) Funktionen gemäß einer modernen Modellierungssprache (ASCET) modellbasiert zu entwerfen und zu implementieren. Neben dem Blick auf die Modellierung legt die Vorlesung ein Augenmerk auf typische Implementierungsaspekte wie Ressourcenknappheit (Speichermangel, fehlende Fließpunktarithmetik) und Zeitanforderungen von Echtzeitsystemen, die in Steuergeräten realisiert werden müssen.

Die Vorlesung geht dabei auf insbesondere auf folgende Inhalte ein:

- Einführung in Objektorientierung, Flussdiagramme und Zustandsautomaten,
- Übersetzung technischer Anforderungen in oben genannte Modellansätze,
- Realisierung von Flussdiagrammen und Zustandsautomaten,
- Verteilung der Funktionen auf die verschiedenen Tasks des zugrundeliegenden Echtzeitbetriebssystems OSEK mit Priorisierung,
- Abbildung von Fließpunktarithmetik auf quantisierte Integerarithmetik,
- Rapid-Prototyping der Funktionen in der ASCET Experimentierumgebung.

Dabei werden anhand dreier konkreter automotiver Funktionen (Tachometer, Blinker-Steuerung, ABS) Modelle zur Funktionsweise der Software realisiert, und in der Experimentierumgebung getestet.

Empfohlene Literatur:

- Foliensatz zur Vorlesung
- Jörg Schäuffele; Thomas Zurawka: Automotive Software Engineering. Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen. 5. Auflage, Springer Vieweg Verlag 2013, Wiesbaden.