

Lehrveranstaltungshandbuch STE

Steuerungstechnik

Version: 3 | Letzte Änderung: 30.09.2019 14:20 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

– Allgemeine Informationen

Langname Steuerungstechnik

**Anerkennende
LModule** STE BaET

Verantwortlich Prof. Dr. Stefan Kreiser
Professor Fakultät IME

Gültig ab Wintersemester
2022/23

Niveau Bachelor

Semester im Jahr Wintersemester

Dauer Semester

**Stunden im
Selbststudium** 78

ECTS 5

Dozenten Prof. Dr. Stefan Kreiser
Professor Fakultät IME
Kellersohn

Voraussetzungen Grundlegende
prozedurale
Programmierkenntnisse
Shannon'sches
Abtasttheorem
Boole'sche Algebra
Datendiskretisierung
Datenkodierung
Endliche Automaten
(FSM)

Unterrichtssprache deutsch

Literatur

Lauber, Göhner: Prozessautomatisierung Bd. 1 u. 2
(Springer)

John, Tiegelkamp: SPS-Progr. mit IEC 61131-3
(Springer)

Wellenreuther, Zastrow: Automatisieren m. SPS
Theorie u. Praxis (Vieweg)

B. Baumgarten: Petri-Netze (Spektrum Akad.)

Priese, Wimmel: Theoretische Informatik - Petri
Netze (Springer)

Abschlussprüfung

separate
Abschlussprüfung

Ja

Details

Mündliche Prüfung nach schriftlicher Vorbereitung. Anhand einer realitätsnahen automatisierungstechnischen Aufgabenstellung angemessener Komplexität entwickeln die Studierenden ein geeignetes Modell für ein nebenläufiges ereignisdiskretes Steuerungssystem. Sie begründen die essenziellen Strukturen ihres Modells unter Bezugnahme auf typische automatisierungstechnische System-, Entwicklungs- und Wartungsanforderungen sowie aufgabenspezifische Vorgaben und weisen nach, dass das Modell das geforderte Verhalten und die geforderte Qualität zeigt, auf einem Steuerungsgerät implementierbar und dann als Steuerungssystem für die gegebene automatisierungstechnische Aufgabenstellung einsetzbar ist.

Mindeststandard

- Studierende extrahieren die wesentlichen relevanten Informationen und Lösungseinschränkungen aus der Aufgabenspezifikation und entwerfen ein begründetes, steuerungstechnisch interpretiertes Petri-Netz-Modell der Steuerung unter Berücksichtigung essenzieller automatisierungstechnischer Qualitätskriterien. - Studierende sind fähig, wesentliche Modellausschnitte im Gedankenexperiment zu simulieren und damit nachzuweisen, dass das betrachtete Modell spezielle, geforderte Verhaltensanteile realisiert. - Studierende sind fähig, ein angemessenes Implementierungskonzept für ihr spezifisches Modell auf einem industriellen Steuerungsgerät in seinen wesentlichen Strukturen und Eigenschaften zu beschreiben und zu begründen. Dabei zeigen sie, wie die einzelnen Modellelemente und Strukturen auf das Implementierungskonzept abgebildet werden.

Prüfungstyp

mündliche Prüfung,
strukturierte Befragung

– Vorlesung / Übungen

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Modellbildung
	Strukturierung
	Systemgrenzen
	Systemzerlegung
	Schnittstellen
	Systemfunktionen
	Verhaltensmodellierung
	Statecharts (SC)
	hybride Netze
	Nebenläufigkeit
	Hierarchie und Historie
	Aktionskonzept
	Petrinetze (PN)
	S/T-Netze
	Netzelemente
	Netzmatrix
	Vorbereichsmatrix
	Nachbereichsmatrix
	B/E-Netze
	Verhaltensanalyse
	Schaltsequenzen
	E-Graph
	Überdeckungsgraph
	Invarianten
	Verhaltensbewertung
	Lebendigkeit
	Reversibilität
	Beschränktheit
	Determiniertheit
	Steuerungstechnisch Interpretierte
	Petrinetze (SIPN)
	Modellierungsmuster
	Komplementstelle / Reservierung
	Kanten
	Test
	Inhibitor
	Event
	Hierarchie
	zeitbehaftete Transitionen
	Transitionsunternetze
	Stellenunternetze
	Seiten
	variables Kantengewicht
Kenntnisse	Steuerungssysteme
	Signalverarbeitung
	Echtzeit
	Arten
	Herkunft von Zeitbedingungen
	Diskretisierung
	Wert
	Zeit
Sensorik	

Besondere Voraussetzungen

keine

Begleitmaterial

digitale Vortragsfolien zur Vorlesung, digitale Übungsaufgabensammlung, Entwicklungswerkzeuge für Petrinetzentwurf, digitale Tutorials für Selbststudium
Themenscripte
Hilfsblätter
Videos

Separate Prüfung

Nein

Signaltechnischer Aufbau
Sensorsysteme
Kalibrierung
Aktorik
Signaltechnischer Aufbau
Aktorsysteme
Steuerungsgeräte
IPC
Programmorganisation
Ressourcen
RTOS
Tasks und Threads
Scheduling
Gerätebeispiele
µC-Boards
Prozessrechner
PAC
RTU
SPS
EN61131
Konfiguration
Ressourcen
zyklische Tasks
EA-Variable
Programmorganisation
POE
Datentypen
Funktionsbausteine
Programmiersprachen
vergleichende Übersicht
prozedural (ST)
grafische Sprachen (FB)
musterbasierte Implementierung
von SIPN auf SPS
Gerätebeispiele
verteilte Automatisierungssysteme
Kommunikation
Strukturen
Stern
Bus
Ring
Redundanz
Verfahren
Shared Memory
Message Passing
asynchron
synchron
Rendezvous
Futures
OSI-Modell
Protokollschichten
MAC-Verfahren
deterministisch
nicht deterministisch
Feldbusse
Industrie (EN61158)
Interbus
Profibus
Profinet
Automotive
CAN
Flexray
Netze

Protokollschichten
IEEE802
IP
Transportprotokolle
UDP
TCP
SCTP
Industrial Ethernet
Hardware
QoS
Redundanz (RSTP)
Virtuelle Netze (VLAN)
Leitsysteme
EN 61499
Architektur
Programmierung
Sicherheit
Gerätesicherheit
Netzwerksicherheit
MES und ERP
Stückgutverfolgung
Automatische Objektidentifikation
(AutoID)
Objekthistorie
Protokolle

Fertigkeiten

Verhalten ereignisdiskreter
Systeme modellieren
Systemverhalten aus Texten
verstehen
technische Textabschnitte
vollständig erfassen
implizite Angaben erkennen und
verstehen
fehlende Angaben erkennen und
ableiten bzw. erfragen
als State Chart (SC) modellieren
FSM als Spezialfall erkennen
Steuerungstechnisch
Interpretiertes Netz
als Petrinetz modellieren
BE-Netz
ST-Netz
Syntax beherrschen
Muster und Makros erkennen und
zielführend anwenden
hierarchisches Netz
tiefe Hierarchien verwenden
flache Hierarchie verwenden
Steuerungstechnisch
Interpretiertes Netz
Petrinetz-Entwicklungswerkzeug
verstehen und zielgerichtet
einsetzen
Modelle verifizieren
Bewertungskriterien definieren
Äquivalenz
Vollständigkeit
Determiniertheit
Lebendigkeit
Reversibilität
Beschränktheit
Einhalten von
Modellierungsvorgaben
...
Testfälle definieren
statische Reviews durchführen und
dokumentieren
Selbst
mit Peer
grafische Analyse
(mathematische Analyse)
dynamische Tests im Simulator
durchführen
Modelle anhand der
Testergebnisse korrigieren und
optimieren

Fertigkeiten Steuerungssysteme entwerfen
Echtzeit
Echtzeitbedingungen ableiten
geeignete Steuerungsgeräte
auswählen
geeignete Bussysteme auswählen
Echtzeitfähigkeit von
Steuerungssystemen nachweisen
SPS in ST programmieren
(EN61131-3)
Syntax beherrschen
Funktionsbausteine einsetzen
Implementierungsmuster für SIPN
herleiten und nutzen
Codegenerator für SIPN
konzipieren
für B/E-Netze
für S/T-Netze
Kontrollfluss in Leitsystemen nach
EN61499 modellieren

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	0
Tutorium (freiwillig)	1

– Projekt

Lernziele

Zieltyp	Beschreibung
Fertigkeiten	Steuerung programmieren kommerzielles SPS- Entwicklungswerkzeug verstehen und zielgerichtet einsetzen wesentliche Eigenschaften einer SPS konfigurieren Programmiersprache ST beherrschen synchrones Message Passing anwenden Funktionsbausteine in der Programmierung anwenden
Fertigkeiten	Simulator für Zielsystem im Zusammenspiel mit SPS- Entwicklungswerkzeug nutzen
Fertigkeiten	komplexe Aufgaben im Team bewältigen einfache Projekte planen und steuern Absprachen und Termine einhalten Reviews planen und durchführen
Fertigkeiten	Realweltsysteme modellieren System analysieren umfangreiche technische Texte erfassen und zielgerichtet auswerten Außenschnittstellen erkennen und korrekt nutzen System strukturieren sinnvolle Teilsysteme definieren Teilsystemfunktionen definieren Schnittstellen definieren Modell der Steuerung entwerfen hierarchisches Steuerungsmodell konzipieren Teilsystemsteuerungen als SIPN modellieren Teilsystemsteuerungen prüfen Funktion im Petrinetzsimulator testen im Peer-Review verifizieren, bewerten und freigeben Teilsystemsteuerungen integrieren Gesamtmodell der Steuerung im Simulator verifizieren

Besondere Voraussetzungen

keine

Begleitmaterial

digital vorgegebene
Projektaufgabe
(Lastenheft),
Entwicklungswerkzeuge
für Petrinetzentwurf
und SPS-
Programmierung,
Tutorials (Script, Video)
Zielsystem
Emulator für Zielsystem
Programmgerüst für
SPS

Separate Prüfung

Ja

Separate Prüfung

Prüfungstyp

Projektaufgabe im
Team bearbeiten (z.B.
im Praktikum)

Details

3 Präsenztermine je 4h
je Projektgruppe,
Abschlusspräsentation

Mindeststandard

Finden sinnvoller
Systemgrenzen und
Modellierung eines
hierarchischen
Gesamtsystems und der
konzipierten
Teilsysteme.
Implementierung der
Steuerung auf einem
professionellen
Steuerungsgerät

Fertigkeiten Steuerungsprogramm für SPS
entwerfen
SPS konfigurieren
zyklische Tasks definieren
vordefinierte EA-Variablen nutzen
vordefinierte Bedienoberfläche
nutzen
Modelltransformationen anwenden
Modelle der
Teilsystemsteuerungen
musterbasiert auf SPS
implementieren
hierarchische
Gesamtsystemsteuerung
integrieren
Implementierung verifizieren
Teilsystemtest am Emulator für
Zielsystem
Integrationsstest am Emulator für
Zielsystem

Fertigkeiten Steuerung am Zielsystem in
Betrieb nehmen

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Projekt	1
Tutorium (freiwillig)	0