

Lehrveranstaltung

GTI - Grundlagen der Technischen Informatik

Version: 2 | Letzte Änderung: 18.09.2019 12:11 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

^ Allgemeine Informationen

Langname	Grundlagen der Technischen Informatik
Anerkennende LModule	<u>GTI_BaET</u>
Verantwortlich	Prof. Dr. Markus Stockmann Professor Fakultät IME
Niveau	Bachelor
Semester im Jahr	Sommersemester
Dauer	Semester
Stunden im Selbststudium	60
ECTS	5
Dozenten	Prof. Dr. Markus Stockmann Professor Fakultät IME Kellersohn
Voraussetzungen	Studierende haben in den Vorlesungen PI1 und IP Grundlagen in der Programmierung (vorzugsweise in C) erworben, dazu zählen unter anderem: Aufbau von Algorithmen, Unterschied Programmiersprache und Maschinensprache, Variablendeklaration, Zeiger, Datentypen, Funktionen, Felder und Werterepräsentation in digitalen Systemen.
Unterrichtssprache	deutsch
separate Abschlussprüfung	Ja

Abschlussprüfung

Details

Summarische schriftliche Abschlussprüfung, um im Sinne des LO1 verstärkt die Kompetenzen K1, K2, K4, K5, K6, K11 zu prüfen. Aufgrund der Tatsache, dass die Prüfung im Rahmen des Praktikums keine Individualprüfung ist, werden die restlichen Kompetenzen im weniger detaillierten Umfang

ebenfalls geprüft.

Mindeststandard

Sichere Beherrschung von Standardtechniken zur Umsetzung von Automaten in C Programmen. Anwendung von Boolescher Algebra auf praktische Aufgabenstellungen. Praktische Aufgabenstellungen abstrahieren, um ereignisdiskrete Systeme (Automaten) modellieren zu können. Die Einbindung von Mikrocontrollern in Systemen über eine abstrahierte Schnittstelle durchführen zu können.

Prüfungstyp

Summarische schriftliche Abschlussprüfung, um im Sinne des LO1 verstärkt die Kompetenzen K1, K2, K4, K5, K6, K11 zu prüfen. Aufgrund der Tatsache, dass die Prüfung im Rahmen des Praktikums keine Individualprüfung ist, werden die restlichen Kompetenzen im weniger detaillierten Umfang ebenfalls geprüft.

^ Vorlesung / Übungen

Lernziele

Kenntnisse

elementare Automatentheorie

[Boole'sche Algebra kennen (PFK.2, PFK.4, PFK.5), Boolesche Funktionsnetzwerke, Grundrechenarten mit Zahlen (PFK.11), Codes zur Informationsdarstellung im Computer (PFK.5, PFK.8, PFK.9, PFK.10), Endliche Diskrete Automaten (FSM) (PFK.5, PFK.7, PFK.8, PFK.9, PFK.10)]

Grundlagen der Technologie digitaler Systeme

[Beschreibungsformen (PFK.8, PFK.9), Schaltplan, Beschreibungssprache (VHDL), Bausteine (PFK.9, PFK.10), Digitale Standard-ICs, wie Gatter AND, OR, NOT, XOR oder

Decoder, Multiplexer, Konfigurierbare Bausteine]

Grundlagen der C-Programmierung für hardwarenahe Programmierung (PFK.9)

[Zeiger und Zeigerarithmetik, Standardbibliotheken (stdio, string)]

hardwarenahe I/O-Programmierung in C (PFK.9)

[Aufbau digitaler I/O-Ports, Zugriff auf I/O-Ports mittels Zeiger, Zugriff auf I/O-Ports mittels Treiberbibliotheken, Bitbasierte Ein-Ausgabe und Verarbeitung mittels C]

Software-Entwicklungsumgebung (PFK.6, PFK.9)

Programmierung von Aufgaben des Messens, Steuerns und Regelns in C (PFK.8, PFK.9, PFK.10) [Realisierung von FSM in C, Aufbau einer anwendungsorientierten IO-Bibliothek auf Basis eines Treibers]

Aufbau und Funktionsweise eines dedizierten Kleinrechnersystems (z.B. Mikrocontroller)

[Architekturübersicht (Register, Rechenwerk, Steuerwerk, Speicher, Busstruktur, I/O-Komponenten) (PFK.12), Funktionsweise, d.h. Ablauf einer Programmabarbeitung auf Basis von Registertransfers (PFK.11)]

I/O-Schnittstellen eines Rechnersystems und deren Nutzung mittels C (am Beispiel des dedizierten Kleinrechnersystems) (PFK.9)

[digitale Ports (siehe oben), Timer/Counter]

Ereignisorientierte Programmierung in C (PFK.8, PFK.9, PFK.11)

Fertigkeiten

Systemverhalten aus spezifizierenden Texten herleiten (PFK.1, PFK.2, PFK.4, PFK.7)

[technische Texte erfassen, implizite Angaben erkennen und verstehen, fehlende Angaben erkennen, ableiten und erfragen]

Nutzung von Beschreibungsverfahren

[Einfache Umrechnungen Boolescher Funktionen (PFK.2, PFK.11), Umsetzung einer FSM in eine C-Programmstruktur (PFK.8)]

Aufbau eines digitaltechnischen Systems (PFK.6, PFK.8, PFK.9, PFK.10)

[Nutzung eines Werkzeugs für Spezifikation, Synthese aus Modell, Systemat. Test mit Testvektoren]

Aufbau eines Steuerungssystems mit Computer (PFK.6, PFK.7, PFK.8, PFK.9, PFK.10)

[Verstehen und Erläutern der Arbeitsweise eines Kleinrechnersystems inkl. einfacher I/O-Schnittstellen, Nutzung von Treiberbibliotheken in C für verschiedene I/O-Schnittstellen mit Unterstützung ihrer Interruptfähigkeit, digitale Ports, Timer/Counter, Programmierung des Systems mit C, Systemverhalten aus spezifizierenden Text herleiten, Aufstellen des Zustandsüberführungsdiagramms, Implementierung mittels C unter Verwendung von Treiberbibliothek]

Aufwand Präsenzlehre

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

Separate Prüfung

keine

^ Praktikum

Lernziele

Kenntnisse

Aufbau eines digitaltechnischen Systems (PFK.6, PFK.8, PFK.9, PFK.10)

[Nutzung eines Werkzeugs für Spezifikation, Synthese, Systemat. Test mit Testvektoren, Realisierung, Konfiguration aus Werkzeug, Test am realen System]

Aufbau eines Steuerungssystems mit Computer (PFK.6, PFK.7, PFK.8, PFK.9, PFK.10)

[Einfache technische Spezifikationen von I/O-Schnittstellen interpretieren und nutzen, Nutzung von Treiberbibliotheken in C für verschiedene I/O-Schnittstellen mit Unterstützung ihrer Interruptfähigkeit, digitale Ports, Timer/Counter, Programmierung des Systems mit C, Systemverhalten aus spezifizierenden Text herleiten, Aufstellen des Zustandsüberführungsdiagramms, Implementierung mittels C unter Verwendung von Treiberbibliothek]

Fertigkeiten

komplexere Aufgaben in einem Kleinteam bewältigen (PSK.1, PSK.6)

Erarbeitung eines digitalen Steuersystems

[übersichtliche Problemstellungen verstehen und analysieren (PFK.2, PFK.7), Systemverhalten aus spezifizierenden Texten herleiten, System strukturiert analysieren]

sinnvolle Teilsysteme erkennen, Schnittstellen zwischen Teilsystemen erfassen,

Problemlösung mittels digitalem Entwurfswerkzeug spezifizieren, testen und am Zielsystem in Betrieb nehmen (PFK.8, PFK.9, PFK.10)]

Erarbeitung eines Steuersystems mit Mikrocontroller und C-Programmen

[übersichtliche Problemstellungen verstehen und analysieren (PFK.2, PFK.7), Systemverhalten aus spezifizierenden Texten herleiten, System strukturiert analysieren

sinnvolle Teilsysteme erkennen, Schnittstellen zwischen Teilsystemen erfassen,

Problemlösung mittels Software-Entwicklungsumgebung in C implementieren, testen und am Zielsystem in Betrieb nehmen (PFK.8, PFK.9, PFK.10)]

Aufwand Präsenzlehre

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

Separate Prüfung

keine