Fachhochschule Nordwestschweiz
Hochschule für Technik

CAS Embedded System 2020

Kursort
Hochschule für Technik FHNW
Institut für Sensorik und Elektronik
Steinackerstrasse 5
5210 Windisch

Kursraum
4.227 (Gebäude 4)

Unterrichtszeiten
Vormittag 08.35-11.45
Nachmittag 13.00-16:15

Stundenplan (grafisch)

Feb 20			Mär 20			Apr 20			Mai 20			Jun 20		
1	Sa		1	So		1	Mi		1	Fr		1	Mo	Pfingstmontag
2	So		2	Mo		2	Do		2	Sa		2	Di	
3	Mo		3	Di		3	Fr	ESL (1)	3	So		3	Mi	
4	Di		4	Mi		4	Sa		4	Mo		4	Do	
5	Mi		5	Do		5	So		5	Di		5	Fr	Projekt 2
6	Do		6	Fr	SIV (3)	6	Mo		6	Mi		6	Sa	
7	Fr		7	Sa		7	Di		7	Do		7	So	
8	Sa		8	So		8	Mi		8	Fr	ESS (1)	8	Mo	
9	So		9	Mo		9	Do		9	Sa		9	Di	
10	Mo		10	Di		10	Fr	Karfreitag	10	So		10	Mi	
11	Di		11	Mi		11	Sa		11	Mo		11	Do	
12	Mi		12	Do		12	So		12	Di		12	Fr	Projekt 2
13	Do		13	Fr	ESH (1)	13	Mo		13	Mi		13	Sa	
14	Fr		14	Sa		14	Di		14	Do		14	So	
15	Sa		15	So		15	Mi		15	Fr	ESS (2)	15	Mo	
16	So		16	Mo		16	Do		16	Sa		16	Di	
17	Mo		17	Di		17	Fr	ESL (2)	17	So		17	Mi	
18	Di		18	Mi		18	Sa		18	Mo		18	Do	
19	Mi		19	Do		19	So		19	Di		19	Fr	Projekt 2
20	Do		20	Fr	ESH (2)	20	Mo		20	Mi		20	Sa	
21	Fr	SIV (1)	21	Sa		21	Di		21	Do	Auffahrt	21	So	
22	Sa		22	So		22	Mi		22	Fr		22	Mo	
23	So		23	Mo		23	Do		23	Sa		23	Di	
24	Mo		24	Di		24	Fr	ESL (3)	24	So		24	Mi	
25	Di		25	Mi		25	Sa		25	Mo		25	Do	
26	Mi		26	Do		26	So		26	Di		26	Fr	Präsentation
27	Do		27	Fr	ESH (2)	27	Mo		27	Mi		27	Sa	
28	Fr	SIV (2)	28	Sa		28	Di		28	Do		28	So	
29	Sa		29	So		29	Mi		29	Fr	ESS (3)	29	Mo	
			30	Mo		30	Do		30	Sa		30	Di	
			31	Di					31	So				

Stundenplan (tabellarisch)

Kurs-Woche	Kalender-Woche	Datum	Dozent	Themen
Signalverarbeitung				
1	8	21.02.2020 22.02.2020	Hufschmid	Grundlagen der Signalverarbeitung
2	9	28.02.2020 29.02.2020	Hufschmid	Anwendungen der Signalverarbeitung
3	10	06.03.2020 07.03.2020	Hufschmid	Schriftliche Prüfung 1 Internet of Things (IoT)
Embedded System				
4	11	13.03.2020 14.03.2020	Buchmann	C/C++ Brushup
5	12	20.03.2020 21.03.2020	Buchmann	Hardwarenahe Programmierung mit C++
6	13	27.03.2020 28.03.2020	Buchmann	Hardwarenahe Programmierung mit C++
7	14	03.04.2020 04.04.2020	Di Cerbo	Auswahl eines Embedded Systems
8	16	17.04.2020 18.04.2020	Di Cerbo	Linux from Scratch
9	17	24.04.2020 25.04.2020	Di Cerbo	Hardwarenahe Programmierung unter Linux
10	19	08.05.2020 09.05.2020	Scheier	Einführung in UML und CIRO Generische Software Architektur
11	20	15.05.2020 16.05.2020	Scheier	Modellieren der funktionalen Anforderungen Aspekte der Code-Generierung
12	22	29.05.2020 30.05.2020	Scheier	Schriftliche Prüfung 2 Verbindungssoftware, Scheduling
Gruppenarbeit				
13	23	05.06.2020 06.06.2020		*Gruppenarbeit*
14	24	12.06.2020 13.06.2020		*Gruppenarbeit*
15	25	19.06.2020 20.06.2020		*Gruppenarbeit*
16	26	26.06.2020	Alle	Präsentation der Gruppenarbeiten

Modulbeschreibung

Signalverarbeitung (SIV)	
Leitung	Prof. Dr. Markus Hufschmid markus.hufschmid@fhnw.ch
Umfang	48 Kontaktlektionen
Unterrichtssprache	Deutsch
Anspruchsniveau	Anwendung / Analyse
Lernziele	Die Studierenden können digitale Signale auf einem Embedded System aufbereiten und auswerten. Sie sind zudem in der Lage, die Verbindung zu externen Sensoren oder Datenbanken über aktuelle Wireless Technologien zu realisieren.
Inhaltsübersicht	Grundlagen der Signalverarbeitung • Abtasttheorem • Digitale Filter (FIR, IIR) • Fast Fourier Transformation (FFT) Anwendungen der Signalverarbeitung • Adaptive Filter (LMS, RLS, MMSE) • Fehlerkorrektur (Parity, Reed Solomon, Viterbi) Internet of Things (IoT) • Kabellose Übertragungsverfahren (WLAN, BT) • ESP32 Plattform
Empfohlene Vorkenntnisse	Grundkenntnisse in Mathematik und Digitaler Signalverarbeitung
Leistungsbewertung	Schriftliche Modulschlussprüfung

Embedded System Hardware (ESH)	
Leitung	Prof. Hans Buchmann hans.buchmann@fhnw.ch
Umfang	48 Kontaktlektionen
Unterrichtssprache	Deutsch
Anspruchsniveau	Anwendung / Analyse
Lernziele	Die Studierenden können C/C++ Programme für kleinere Systeme ohne Betriebssystem (Barebones) entwickeln.
Inhaltsübersicht	C/C++ brush up • Pointer/Arrays, struct/class, call-back Barebones • Nützliche C++ Konstrukte für die hardwarenahe Programmierung: Constructor/Destructor, Resource Acquisition Is Initialization (RAII), Templates • Zugriff auf die Hardware • Interrupts: Hardware call-back, front-/backend, der gemeinsame Zugriff
Empfohlene Vorkenntnisse	Grundkenntnisse in C Verwendung eines Systems zur Versionenverwaltung (svn, git etc.)
Leistungsbewertung	Schriftliche Modulschlussprüfung

Embedded System Linux (ESL)	
Leitung	Manuel Di Cerbo manuel.dicerbo@fhnw.ch
Umfang	48 Kontaktlektionen
Unterrichtssprache	Deutsch
Anspruchsniveau	Anwendung / Analyse
Lernziele	Die Studierenden können C/C++ Programme für grössere Systeme mit Betriebssystem (Linux) entwickeln.
Inhaltsübersicht	Auswahl eines Embedded Systems • Aspekte der Hardwareauswahl • Optionen für Betriebssysteme (General Purpose, RTOS oder ohne OS) • Anforderungen an Realtimefähigkeit und hardwaremässige Segmentierung • SoC vs SoM Linux from Scratch • Hostsystem und Toolchain einrichten • Bootloader auswählen, konfigurieren und kompilieren • Bootprozess, ROM, First Stage, Second Stage, Linux • Initramdisk vs. Root File System • Verschiedene Möglichkeiten Linux zu booten (sftp, nfs, eMMC, SD-Karte) • Kernel auswählen, herunterladen und kompilieren • Kompatibilität zu Linux Realtime patches • Konfiguration des Kernels (menuconfig) • Minimaler Linux Userspace mit Buildroot (selektierbare features), Konfiguration und Kompilieren • Package Management: verschiedene Varianten Hardwarenahe Programmierung unter Linux • im Userspace, Zugriff auf die Hardware • im Kernelspace, Kernelmodules • Memory Management • Multi Threading • Callbacks
Empfohlene Vorkenntnisse	Vorgängermodul: Embedded System Hardware
Leistungsbewertung	Schriftliche Modulschlussprüfung

Embedded System Software (ESS)	
Leitung	Johannes Scheier scia@zhaw.ch
Umfang	48 Kontaktlektionen
Unterrichtssprache	Deutsch
Anspruchsniveau	Anwendung / Analyse
Lernziele	Die Studierenden kennen ein systematisches Vorgehen um Anwendungen für eingebettete Systeme zu entwickeln und sind in der Lage, dieses anzuwenden. Sie können funktionale Anforderungen an das System graphisch modellieren, um daraus C/C++ - Code für verschiedene Plattformen zu generieren. Sie sind in der Lage, die Verbindung zwischen plattformunabhängiger Anwendungssoftware und der Hardware zu definieren und kennen Strategien um die Einhaltung von Echtzeitanforderungen zu garantieren.
Inhaltsübersicht	UML • UML (Unified Modelling Language) • Einführung der relevanten Elemente CIRO • Einführung in CIRO (Communicating Interacting Reactive Objects) eine Erweiterung von UML für reaktive Systeme Generische-Software-Architektur • Überblick über die Software-Architektur Modellieren der funktionalen Anforderungen • Modellieren von reaktiven Systemen mit kooperierenden State-Machines Code-Generierung • Generierter Code • Strategien • Werkzeuge • Effizienz, Qualität, Verbindungssoftware • Definition der Verbindungssoftware zu Hardware-Plattform bzw. OS. • Lokal vs. verteilt Scheduling • Multi-Tasking • Verteilung • Task-Scheduling, Event-Scheduling
Empfohlene Vorkenntnisse	Vorgängermodule: Embedded System Hardware & Embedded System Linux
Leistungsbewertung	Schriftliche Modulschlussprüfung