

Microcontroller Programming

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

Mikrocontroller-Technik	2
<i>Einführung und Einordnung</i>	2
<i>Ablauf im Semester</i>	2
<i>Abgabe</i>	3
<i>Links zur Vorlesung</i>	3
<i>weiterführende Links</i>	4
C Programmierung	4
Embedded Systems Engineering und Hardware/Software Co-Design	4
sonstiges	4

Mikrocontroller-Technik

Einführung und Einordnung



Source: eigenes Foto (CC0 1.0)

Die Mikrocontroller-Technik befasst sich damit, wie man einem programmierbaren Bauteil ("Mikrocontroller") Leben einhaucht. Dabei wird die Software im Folgenden in der Programmiersprache C programmiert. Im Gegensatz zu der im 1. und 2. Semester im Kurs Informatik dargestellten Sprachkomponenten und Algorithmen wird hier Wert auf die Eigenheiten durch die Hardware-nähe gelegt. Es wird auf die Kenntnisse folgender Fächer zurückgegriffen:

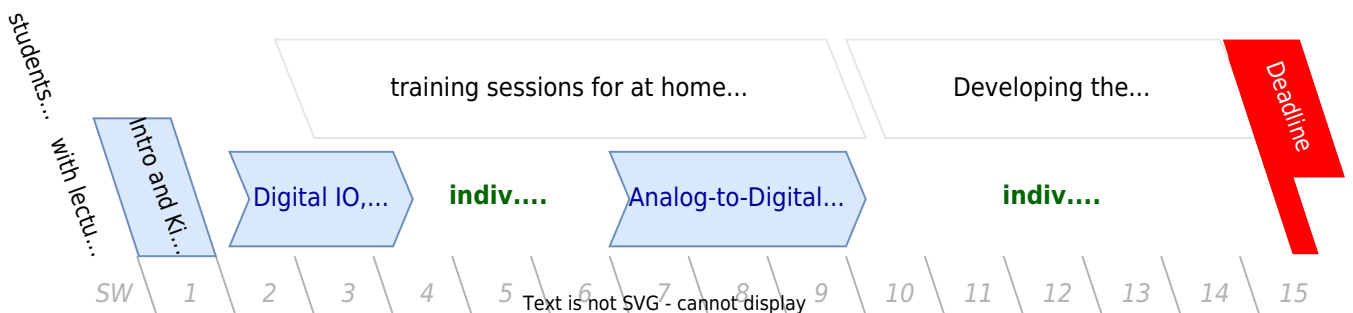
- Informatik I und II

Wichtig:

- Bleiben Sie mit uns in Kontakt! Trotzdem es zeitweise Wochen mit eigenständigen Bearbeitung gibt, ist der Austausch wichtig, damit zu den jeweiligen Abgaben ein sinnvolles Ergebnis abgeliefert wird.
- Suche Sie den Austausch mit den anderen Studierenden. Meistens haben Sie die Lösung, zum Problem eines anderen und umgekehrt.
- Wenn Sie erst eine Woche vor der Abgabe mit dem Programmieren beginnen, wird es in der Regel eng. Besonders, wenn es Unklarheiten und Nachfragen gibt. Beginnen Sie frühzeitig.
- Versuchen Sie Spaß am Programmieren zu finden. Es ist wie Puzzeln, nur dass Sie selbst das Bild bestimmen!

Ablauf im Semester

Der Kurs ist im Semester in verschiedene Schritte aufgeteilt:



Semester-woche	Titel	Beschreibung
SW1	Kickoff in der ersten Woche	hier werden die Details zum Kurs durchgegangen und die Vorbereitungen besprochen.

Semester-woche	Titel	Beschreibung
SW1	erste Übung	Bereits in der ersten Woche werden erste "Gehversuche" in der Entwicklungsumgebung umgesetzt um die Software und Hardware zu testen.
SW2 - SW10	Geführte Entwicklung von Code	Dies ist aufgeteilt in - Montags: Einführung in die Themen durch Tim Fischer mit Schwerpunkt auf die Theorie - Donnerstags: praktische Arbeit an der Hardware durch Tim Fischer, Petre Sora (ASE) und Ralf Ziegler (MR)
SW11 - SW15	eigenständige Arbeit	Hier sind Sie am Start und müssen vorgegebene Projekte bearbeiten Montags und Donnerstags: Mentorings mit Tim Fischer, Petre Sora (ASE) und Ralf Ziegler (MR)
SW15	Abgabe des Codes und der Doku	Die Inhalte müssen bis 17.01.2024 hochgeladen sein.
im Prüfungs-zeitraum	schriftliche Prüfung	Eine Beispielklausur ist in ILIAS zu finden

Abgabe

- Zur Abgabe der Softwarekomponenten
 - Bitte geben Sie folgenden Komponenten ab:
 - Projektordner aus Microchip Studio
 - Bericht mit Fluss/Zustandsdiagramm als PDF
 - Bitte entfernen Sie nicht notwendige Files, d.h. lassen Sie alte, nicht eingebundene Files und alte Versionen weg, sofern Sie solche angelegt haben
 - Die Abgabe erfolgt per GITlab
- Für die Bewertung des Fachs Microcontrollertechnik wird folgendes Schema verwendet: [bewertung_ws24_prj99.xlsx](#)
Die Bewertung setzt sich aus folgenden Punkten zusammen:
 - 60% Prüfung im Prüfungszeitraum
 - 30% Code-Qualität (u.a. Funktionsfähigkeit des Programms, Code-Struktur, konsistente Namensgebung, Inline Dokumentation)
 - 5% Bericht + Flussdiagramm
 - 5% Mitarbeit bei den Mentorings

Allgemeine Randbedingungen

1. Lesen Sie die Ihnen zugewiesene Projektbeschreibung gut durch
2. Programmiersprache ist C (Arduino ist nicht möglich)
3. Die Delay-Funktion darf nicht für die Zeit-Synchronisation im Millisekunden-Bereich genutzt werden!
Es ist eine Einteilung in Takte (10ms, 100ms, etc.) zu verwenden.
4. Beachten Sie die Infos unter [Tipps für Programmierung](#), insbesondere die Vorgaben für die Programmierung!

Links zur Vorlesung

Thema	Beschreibung	Sprache
Tutorial	Eine schöne Einführung in die Embedded Softwareentwicklung ist im Buch Sensornetzwerke in Theorie und Praxis - Embedded Systems-Projekte erfolgreich realisieren von Kollegen Meroth und Sora zu finden. Dort wird der Einstieg in das Feld die (in Hardware) eingebettete Softwareentwicklung erklärt. Aus dem Hochschulnetz bzw. mit VPN können Sie dieses direkt bei Springer Link betrachten.	Deutsch
Tutorial	Empfehlenswertes Tutorial von mikrocontroller.net	Deutsch
Tutorial	Also the AVR itself has a nice introduction into C programming on the AVR plattform as Video and textual Developer Help	English
Datenblatt	Das Data Sheet des Atmega88 ist ebenfalls ein gutes Nachschlagewerk	English
Datenblatt	Das Datenblatt des Atmega88 wurde auch auf Deutsch übersetzt: Online Datenblatt ATmega88	English
C-Support	Ein Rechner für Interrupt Timer und PreScaler	English
C-Support	Dokumentation der AVR Compilerbibliotheken (z.B. stdlib)	English

weiterführende Links

C Programmierung

- schönes Online [Open Source Buch](#) zum Erlernen der Programmiersprache C
- [C Code Referenz Card](#) ("C Cheatsheet")

Embedded Systems Engineering und Hardware/Software Co-Design

- [Handbuch Embedded Systems Engineering](#) der TU Clausthal / FH Nordhausen: Sehr ausführliches und tiefgehendes Buch
- Auch das Buch [Practical UML Statecharts in C/C++, 2nd Edition: Event-Driven Programming for Embedded Systems](#) eignet sich als ausführliches Nachschlagewerk
[中国語の翻訳](#)

sonstiges

- [Tabelle der ASCII Zeichen](#)
- [Bauanleitung des Mexle AVR-Proggis](#)

From:

<https://wiki.mexle.org/> - **MEXLE Wiki**

Permanent link:

<https://wiki.mexle.org/microcontrollertechnik/start?rev=1726705700>

Last update: **2024/09/19 02:28**

