

# Elektronische Systeme

## Student Group

First Name	Surname	Matrikel Nr.

## Table of Contents

- Elektronische Systeme** ..... 2
- Einführung*** ..... 2
- Bisherige Kurse*** ..... 2
- Themen für Kurzvorträge*** ..... 3
- Themen für Projekte*** ..... 3
- Terminplanung*** ..... 6
- weiterführende Links ..... 6

# Elektronische Systeme

## Einführung



Source: Pixabay (CC 0 Lizenz)



Source: Pixabay (CC 0 Lizenz)



Source: Pixabay (CC 0 Lizenz)

Unterschiedlichste elektronische System umgeben uns Tag täglich: Mobiltelefone, Laptops, Fahrzeugsteuerung... In diesem Kurs wollen wir eine Systemidee von Ihnen in elektronischer Hard- und Software umsetzen, um dabei die Einblicke in die Elektronikentwicklung zu erweitern. Ziel ist vor dem Jahresabschluss bereits die Systeme zu präsentieren.

Weiterhin ergeben sich vor den Prüfungen 4 Termine, bei welchen Experten aus der Industrie interessante Einblicke in die Elektronik darstellen. Hier bin ich auf Ihr Interesse angewiesen und offen für Ideen.

## Bisherige Kurse

Im letzten Kurs wurden folgende Präsentationen gehalten:

- **Wie wird eine Platine entwickelt?** (externer Vortragender, Würth/WEdirekt)  
Nutzentypen, Multilayer, Flex-Systeme, Vias
- **Weitere Tipps und Tricks zum Layouting von Platinen** (Prof. Gruhler)  
parasitäre Induktivitäten und Kapazitäten, Führung von analoger und digitaler Masse, Aufbau von Multilayerplatinen, Kunst des Layoutings
- **Mikrocontroller-Technologien** (externer Vortragender, Microchip)

Hardware-Software-Codesign, typische Haken und Ösen bei der Hardwareentwicklung

- **Software- und Systementwicklung** (*externer Vortragender, Bosch*)  
 Prozesse, industrielle Tools, Tagesablauf

Gegebenenfalls könnten folgende Themen interessant sein:

- **Wie kann ich Kommunikation und Datenablage sicher machen**  
 Checksummen, Zyklische Redundanzprüfung, Hash-Funktionen, Kryptographie
- **Wie wandelt man - z.B. im Elektroauto, im Netzteil, in der Endstufe - Wechselfrequenz in Gleichspannung und umgekehrt**  
 Halbbrücke, Buck/Boost-Converter, Vollbrücke, B6-Brücke, Transistortypen, Gegentaktendstufe
- \* **Warum brennen Prozessoren nicht durch?**  
 Thermomanagement, Wärmestromkreis, transiente thermische Impedanz
- **komplexere Filter**  
 Bandsperre, Bessel-, Butterworth-, Chebyshev-Filter, Ausnutzung von Resonanz
- Künstliche Intelligenz

Projektvorschläge folgen bis zum Semesterbeginn. Eigene Projektvorschläge sind gerne gesehen.

## Themen für Kurzvorträge

1. Transistoren - Transistorstrukturen im Chip: von Standardzellen zu 3D-Integration und „Forksheets“-Transistoren (FinFETs, vertikale Transistoren, ...)
2. Transistoren - Einführung in CMOS
3. Energiespeicher - Miniaturbatterien in Dünnschicht
4. Energiespeicher - Supercaps und Hybrid-Supercaps
5. Energiespeicher - Impedanzspektroskopie bei Batterien
6. Controller - Multicore, Pipelining und Vectoring in Prozessoren
7. Controller - DSPs
8. Controller - Hardware-integrierte Neuronale Netze (analoge Neuronen über Resistive RAM, Compute in Memory, mathematische MAC Operationen, FPGAs, ...)
9. Einführung in den thermischen Stromkreis (z.B. Berechnungen mit Zth)
10. Einführung in Photonik (optische Kopplung, Polarisationsmanagement, SiN-Wellenleiter)

## Themen für Projekte

Nr.	Thema	Beschreibung	Umfang	sinnvolle Interessen		
				Softwareentwicklung?	Hardwareentwicklung?	Systementwicklung?
1	Simulide Systemübersicht / Weiterentwicklung	Das Simulationstool Simulide ist in Cpp programmiert. Aktuell ist nicht bekannt, dass es eine Systemübersicht gibt (welche einzelnen SW-Module gibt es? wie interagieren diese?). Ein Ziel ist diese zu entwickeln. Als Start können verschiedene Forumsbeiträge genutzt werden, z.B. <a href="#">Kompilieren unter Win10</a> . Weiterhin können kleine Komponenten umgesetzt werden (z.B. Motor mit "Spannungsausgang" für Drehzahl, Verwendung des VCC Eingangs zum Start des Microcontrollers etc.)	2	ja, Cpp (in Windows)		Ja, Systemübersicht

2	Einarbeitung in Falstad circuitjs	Ziel ist eine Systemübersicht über das Programm zu erstellen und kleine Verbesserungen vorzunehmen. z.B. neuer Verbindungs-/Datentyp ohne phys. Einheit und Konvertierungsmodule dafür (z.B. zur Eingabe von Zahlenwerten wie 0x025, welche dann über einen Bus auf einzelne Bits und Pegel heruntergebrochen werden können)	2	Ja, Java Script		Ja, Systemübersicht
3	WLAN-Hookup MEXLE (als standalone Basis Board)	OTA-Flashing für AVR-Chips / Mexle 328PB über ESP32	1...2	Ja, C++ / C (embedded)	Ja, Schaltung/Layout in eagle	
4	eine "einfache" Programmiersprache: Scratch auf dem MEXLE 328PB Board	erste Tests zur Verwendung von Scratch mit einem AVR Chip (siehe S4A). Ziele sind suchen eines Anwendungsbeispiels, Implementierung, Tests, Beschreiben der Vorgehensweise,	1...2	Ja, Scratch		Ja
5	(einfacher) Prozessornachbau in Falstad / digital	Ziel ist das Erstellen eines einfachen funktionsfähigen Prozessors in Simulationen. Verschiedene Beispiele existieren bereits (z.B. RISC CPU, kleine ALU, oder Simulationen in Logic Ordner Examples\processors\...). Die Aufgabe dient dazu das Verständnis aufzubauen und einen kompakten Prozessor zu entwickeln (z.B. 4bit).	2...4		Ja, Simulation	
6	Modellierung eines Operationsverstärkers	(z.B. einfaches Modell, komplexeres Modell)	1...2			
7	Elektronik-Management über Bar-, QR-Code (z.B. quaggajs, automatischer Erstellung für eagle)	Auf allen zukünftigen Platinen sollen 2D-Codes abgedruckt werden, um weiterführende Infos im Netz bereitzustellen und um eine Seriennummer abzudrucken. Ziel ist eine Implementierung eines ULP-Makros in eagle, welche diese 2D-Codes erstellt. Weiterhin soll auf einem Server ein PHP / JS Skript erstellt werden, welches mit der Smartphone-Kamera Fotos aufnehmen kann, daraus den 2D-Code und damit die Seriennummer / Typ ermittelt und auf die weiterführenden Infos weiterleitet. Hierzu sind einige Frameworks im Netz auffindbar	1...2	Ja, in eagle und ggf. PHP, Java Script		
8	(weitere) Anpassung MEXLE ↔ Fischertechnik	Motor(treiber)output und Sensordaten-Input auf als Hookup auf Basis des AVR32DB. Konzeptionierung eines Demo-Fischertechnikaufbaus sinnvoll (z.B. kleines Fahrzeug mit 2 Motoren und div. Sensoren)	2...3		Ja, Schaltung/Layout in eagle	Ja, Fischertechnik-Demosystem

9	„Leit“-Projekte entwickeln	Für die Fächer Grundlagen der Digitaltechnik und Elektronik-Labor sind Leitschaltungen zu entwickeln. Diese sollen den Studierenden zu Beginn des Semesters bereits eine Anwendung zeigen von dem, was Sie nach und nach erst lernen. Bei "Grundlagen der Digitaltechnik" könnte dies z.B. eine einfache Implementierung der TWI Hardware sein. Diesen sollen in geeignete MEXLE2020 Module umgewandelt werden. Beim "Elektronik-Labor" könnte dies z.B. eine Ansteuerung eines 3-phasigen Motors mit einem DRV88xx, einem AVR-uC und einem Lagesensor als Analoginput sein. (Ggf. ist dies auch für ET1, ET2, Microcontrollertechnik und ET-Labor möglich)	1...2		Ja, Schaltung/Layout in eagle und Simulation	Ja, Suche nach geeigneten Schaltungen und Systemen, Simulation in Falstad / Tina / Logic
10	Magnetkreise mit FEMM auslegen (z.B. Elektromotor, Reluktanzmotor, Lautsprecher etc. )	Das Programm <a href="#">FEMM</a> ermöglicht die Simulation von elektrischen und magnetischen Effekten per finite Elemente. Ziel der Aufgabe ist ein Aufbau verschiedener Magnetkreise (Gleichstrommotor, Reluktanzmotor, Synchronmaschine, Lautsprecher etc. ) und eine Beschreibung wie dies umsetzbar ist zu erstellen. Die Struktur ist über ein CAD-Programm (z.B. AutoCAD) zu erstellen. Optimal ist ein Output als Animation	1...2		Ja, Simulation	
11	Entwicklung von diversen Layouts	(1) Intelligentes Displaymodul mit Tasten: Anbindung der Tasten und des Displays an I2C (2) Funktionsgenerator-Hookup (auf AVR32DB Basisboard, welches bereits DAC und OPV hat). Umwandlung von unipolarem Signal (0..3,3V) in bipolares (-3,3V...+3,3V) (3) Template für zukünftige Sensormodule (4) Alternative Basisplatten auf Basis des TI MSP430 oder PIC18 (ggf. auch STM8)	3		Ja, Schaltung/Layout in eagle	Ja, Konzepterstellung für Funktion, Pinning und Anforderung an Software
12	Software I2C	I2C ohne die TWI-Register eines AVR-Chips über direkte Ausgabe von Bits implementiert werden. Hierfür ist bereits eine library von <a href="#">Peter Fleury</a> vorhanden. Diese soll so erweitert werden, dass die I2C-Slaves dynamisch hinzugefügt werden können (ARP oder Address Resolution Protocol für "hot plug").	2...3	Ja, C (embedded)		

# Terminplanung

Semester-woche	Termin	Modus	Projekt
1	01.03	Einführung	Auswahl des Projekts, Komponentensuche, Einlesen in Datenblätter, Projektplanung: - Mindmap - Gantt-Chart - Ressourcenplanung
2	08.03		
3	15.03	Eigenständige Arbeit / Coaching	
4	22.03		
5	29.03		
6	05.04	ENTFÄLLT (Ostern)	
7	12.04	Eigenständige Arbeit / Coaching	
8	19.04		
-	26.04	ENTFÄLLT (Prüfungen)	
-	03.05		
-	10.05		
-	17.05		
-	24.05		
9	31.05	Präsentationsslot 1-4, Coaching	
10	07.06	Präsentationsslot 5-8, Coaching	
11	14.06	Präsentationsslot 9-12, Coaching	
12	21.06	Präsentationsslot 13-16, Coaching	
13	28.06	Präsentationsslot 7-20, Coaching	

Bei allen Veranstaltungen ist die Möglichkeit zu Rücksprachen zu den eigenen Projekten gegeben.

## weiterführende Links

[Theorie paralleler und verteilter Systeme](#) von Hr. Prof. Tantau an der [Uni Lübeck](#)

[Dimensionierung von Schaltnetzteilen](#)

[iPES: interaktives Power Electronics Seminar](#)

[diverse Skripte für Elektronik](#) der ZHAW (Schweiz)

From:

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

Permanent link:

[https://wiki.mexle.org/elektronische\\_systeme/start?rev=1615165317](https://wiki.mexle.org/elektronische_systeme/start?rev=1615165317)

Last update: **2021/05/09 10:06**

