



**Hes·so**  
Haute Ecole Spécialisée  
de Suisse occidentale  
Fachhochschule Westschweiz



Haute école d'ingénierie et d'architecture Fribourg  
Hochschule für Technik und Architektur Freiburg

# Embedded Systems 1

## Einführung und Ziele

Daniel Gachet | Jacques Supcik | 2018/2019  
o.01 | T2-dg / I2-d | 2018-09-14

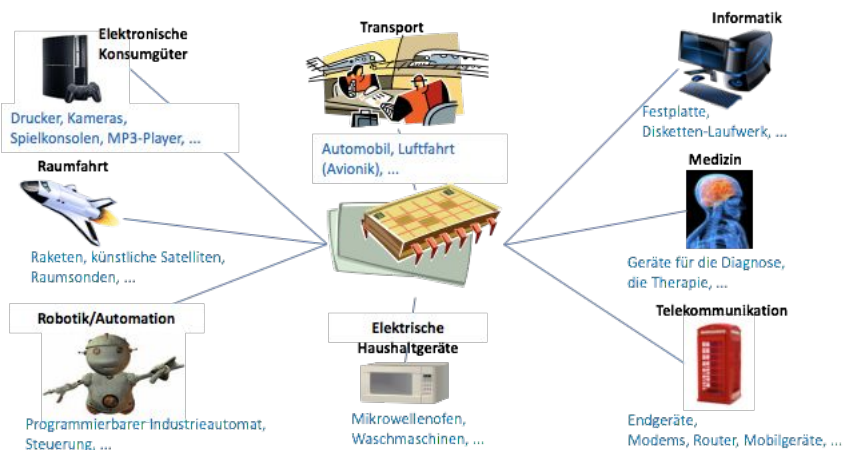
## Inhalt

- Einführung in die embedded systems
- Zweck und Zielsetzungen des Kurses
- Ablauf
- Bewertung
- Bibliografie und Referenzen

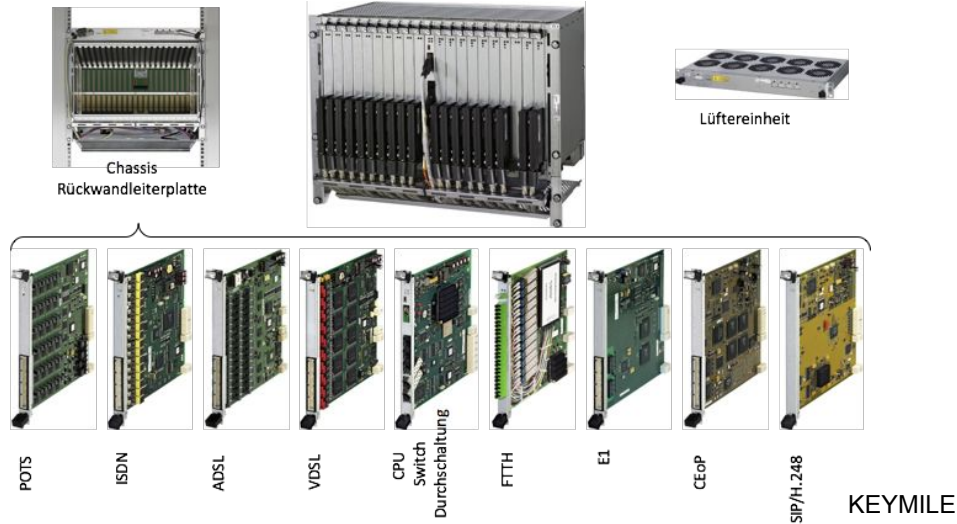
# Anwendungen der Embedded systems

- Die Anwendungsbereiche der embedded systems sind dank der Entwicklung des Siliziums weit und sehr vielfältig.
- Es existieren praktisch keine modernen Geräte ohne Mikroprozessor, der mit mehr oder weniger komplexer Software ausgestattet ist.

# Anwendungen der Embedded systems



# Beispiel einer Mehrdienste-Zugang Ausrüstung



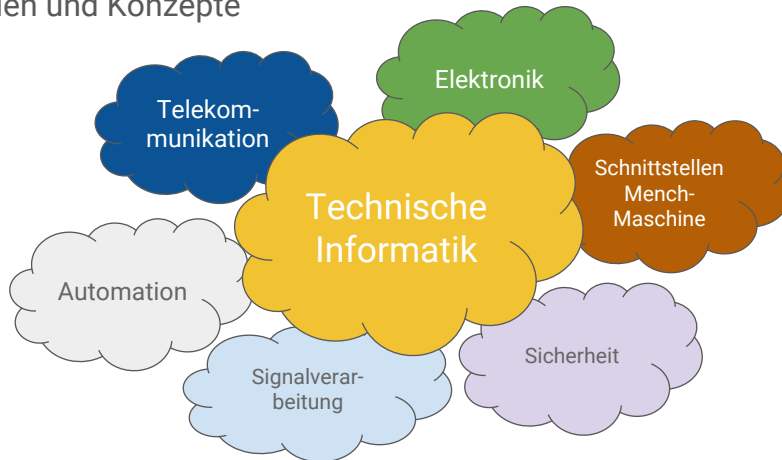
# Beispiele mobiler Anwendungen



Tablette Apple iPad Pro  
Apple A10X (hexa-core 64 bits ARMv8-a)

# Technologien

Die embedded systems verlangen den Einsatz sehr unterschiedlicher Technologien und Konzepte



## 8 grossartige Ideen in der Computerarchitektur

- Design für Moore's Gesetz
- Verwenden Sie Abstraktion, um das Design zu vereinfachen.
- Machen Sie den gemeinsamen Fall schnell.
- Leistung durch Parallelität
- Leistung durch Pipelining
- Leistung durch Vorhersage (Prediction)
- Speicherhierarchie
- Zuverlässigkeit durch Redundanz

# Mikroprozessor - Definitionen

## Prozessor

Name (m) (Englisch processor), Bauteil, das in einem Rechner dazu dient, Befehle zu interpretieren und auszuführen. Gesamtheit der Programme, die auf einem Rechner die Ausführung von Programmen ermöglichen, die in einer bestimmten Sprache geschrieben sind.

*Nach Larousse online (<http://www.larousse.fr/dictionnaires/francais/>)*

# Mikroprozessor - Definitionen

## Mikroprozessor

Name (m) (Englisch microprocessor, von to process, verarbeiten) Prozessor in Miniaturform, in dem alle Elemente in einem integrierten Schaltkreis zusammengefasst sind.

Die seit 1971 kommerziell verfügbaren Mikroprozessoren bilden die Grundkomponenten von Mikrocomputern und werden in zahlreichen Geräten eingesetzt, um diese mit intelligenten und automatischen Funktionen auszurüsten.

*Nach Larousse online (<http://www.larousse.fr/dictionnaires/francais/>)*

# Mikroprozessor - Definitionen

## Embedded system

Ein embedded system ist definiert als autonomes elektronisches Informatik-System, das oft in Echtzeit arbeitet und auf eine genau definierte Aufgabe spezialisiert ist. Der Begriff bezeichnet sowohl die Informatik-Hardware als auch die eingesetzte Software. Seine Ressourcen sind im Allgemeinen begrenzt. Diese Begrenzung bezieht sich allgemein auf den Platzbedarf (kleineres Gehäuse) und den Energieverbrauch (eingeschränkte Leistungsaufnahme).

*Nach Wikipedia: [http://fr.wikipedia.org/wiki/Système\\_embarqué](http://fr.wikipedia.org/wiki/Système_embarqué)*

# Mikroprozessor - Definitionen

## Embedded system @ iSIS/HEIA-FR

System, das sowohl Hardware als auch Software einsetzt, um wirtschaftlich einen technischen Aspekt zu lösen, ohne dass der Benutzer des besagten Systems sich dessen bewusst ist

# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- in der Sprache C zu programmieren
  - Kenntnis der Typen und der grundlegenden Deklarationen
  - Kenntnis komplexer Typen und der Funktionen
  - Kenntnis und Anwendung von Zeigern und Funktionszeigern
  - Schnittstelle Assembler – C
  - Einführung in die objektorientierte Programmierung
  - Entwicklungswerkzeuge (Programmierung, Überprüfung, Dokumentation)

# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- in der ARM-Assemblersprache zu programmieren
  - Kenntnis der internen Register und des Befehlssatzes
  - Verarbeitung von Zahlen und von Flags
  - Programmierung einfacher Anwendungen
  - Benutzung der  $\mu P$  unter Verwendung höherer Programmiersprachen

# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- die allgemeine Architektur der Mikroprozessoren und der  $\mu$ P-Systeme zu beschreiben
  - Architektur Von Neumann und Harvard
  - Interner Aufbau der Mikroprozessoren
  - Eingänge/Ausgänge, Zugriff auf den Speicher und die Peripheriegeräte

# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- Unterbrechungen (Interrupts) zu behandeln
  - Kenntnis der Grundprinzipien von Unterbrechungen
  - Behandlung von Unterbrechungen (Kontextwechsel)
  - Eingänge/Ausgänge (Gegenüberstellung Unterbrechung - Polling-Modus)
  - Betriebssysteme (grundlegende Mechanismen - Kern)



# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- die moderne Architektur der Mikroprozessoren zu verstehen
  - Programmierung eines direkten Speicherzugriffs (DMA)
  - Programmierung des Cache-Speichers
  - Programmierung der virtuellen Speicherverwaltung (MMU)

# Zielsetzungen

Am Ende des Kurses sind die Studierenden in der Lage,

- $\mu$ P-Systeme einzusetzen
  - Initialisierung des  $\mu$ P und seiner Peripheriegeräte
  - Starten der Softwareanwendungen
  - Implementierung einer Kommunikationsschnittstelle

# Zielsetzungen der Laborübungen

- Die Laborübungen dienen folgenden Hauptzielen
  - Vertiefung der in den Kursen vermittelten Lerninhalte
  - Einsatz von  $\mu$ P und embedded systems
  - Anwendung von plattformübergreifenden Entwicklungswerkzeugen für embedded systems
    - Werkzeugkette für das Programmieren von Software Anwendungen
    - Werkzeugkette für das Debuggen von Software Anwendungen
  - Realisierung eines kleinen Entwicklungsprojekts
    - Projektverwaltung
    - Spezifikationen, Projektierung
    - Realisierung, Überprüfung/Validierung
    - Dokumentation

# Ablauf

## Ablauf der Kurslektionen

- Tragbare Computer geschlossen, wenn sie nicht für den Kurs benötigt werden
- Erklärungen auf Folien, an der Wandtafel oder mit dem Projektor
- Ein Teil der Übungen wird während des Kurses gelöst
- Eine Reihe von Übungen wird zur Verfügung gestellt
- **Zögern Sie vor allem nicht, in den Lektionen Fragen zu stellen!**

# Ablauf

## Ablauf der Laborübungen

- industrielle Formel (alles steht zur Verfügung, Zusammenarbeit zwischen den Gruppen, ...)
- Hardware Beaglebone Black (TI AM3358 / ARM Cortex-A8)
- Ordnung und Disziplin (Material)
- Keine Esswaren oder Getränke (Material)
- Aufräumen/wegräumen des Materials nach den Laborübungen

# Kursunterlagen

Alle Kursunterlagen stehen unter folgende GIT-Repositories zur Verfügung:

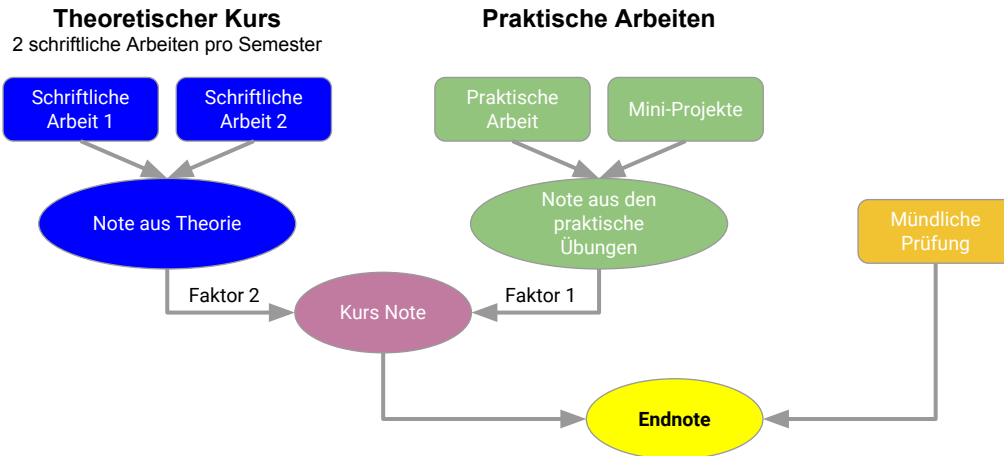
Kursunterlagen :

<https://gitlab.forge.hefr.ch/se12-1819/se12>

Praktikum :

<https://gitlab.forge.hefr.ch/se12-1819/tp>

# Bewertungen



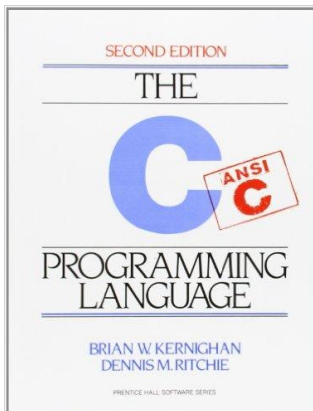
## Schriftliche Arbeiten

- **Bedingungen**
  - Tragbarer PC (Notebook, Tablet) NICHT erlaubt
  - Mobiltelefon NICHT erlaubt
  - Kursunterlagen und persönliche Notizen NICHT erlaubt
- **Antworten**
  - Muss auf Deutsch gegeben werden
  - Mehrdeutige Antworten werden systematisch als falsch bewertet
- **Absenzen**
  - Unbegründete Absenzen → Minimalnote 1
  - Kein Wiederholungstermin
- **Betrügereien**
  - Sofortiger Prüfungsausschluss → Minimalnote 1

# Praktische Arbeiten

- Bedingungen
  - Es steht alles Material zur Verfügung
  - Lerntagebuch oder Bericht mit Code im Anhang innerhalb von 7 Tagen
- Berichte
  - Müssen auf Deutsch verfasst werden (Ausnahmen sind für Studenten der deutschen Muttersprache möglich).
- Absenzen
  - Unbegründete Absenzen → Minimalnote 1

# Bibliografie

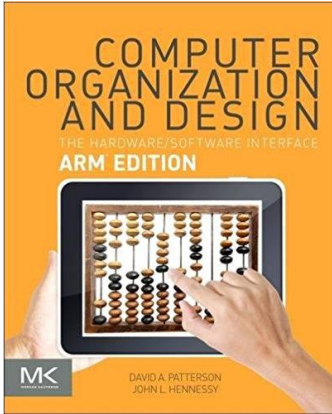


## The C Programming Language

*Brian W. Kernighan, Dennis M. Ritchie*

- 2<sup>nd</sup> edition (April 1, 1988)
- Verlag : Prentice Hall
- ISBN : 0131103628, 978-0131103627
- 272 Seiten

# Bibliografie

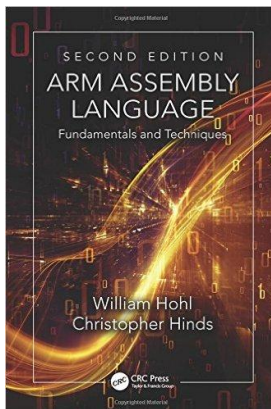


## Computer Organisation and Design

*David A. Patterson, John L. Hennessy*

- The Morgan Kaufmann Series in Computer Architecture and Design
- 1<sup>st</sup> edition (March 16, 2016)
- Verlag : Morgan Kaufmann
- ISBN : 0128017333, 978-0128017333
- 720 Seiten

# Bibliografie

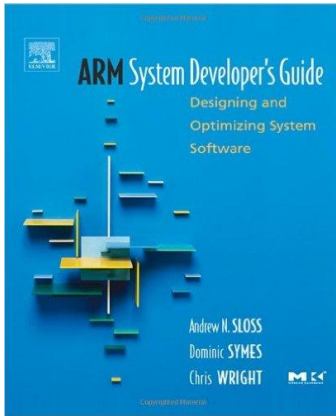


## ARM Assembly Language

*William Hohl, Christopher Hinds*

- Fundamentals and Techniques
- 2<sup>nd</sup> edition (October 20, 2014)
- Verlag : CRC Press
- ISBN : 1482229854, 978-1482229851
- 453 Seiten

# Bibliografie

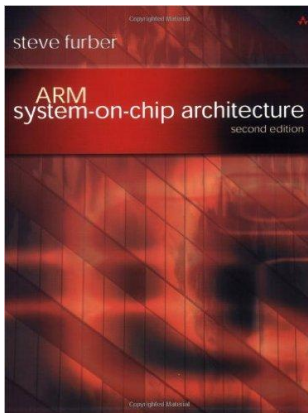


## ARM System Developer's Guide

*Andrew Sloss, Dominic Symes, Chris Wright*

- Designing and Optimizing System Software
- 1<sup>st</sup> edition (April 8, 2004)
- Verlag : Morgan Kaufmann
- ISBN : 1558608745, 978-1-55860-874-0
- 689 Seiten

# Bibliografie



## ARM System-on-Chip Architecture

*Steve Furber*

- 2<sup>nd</sup> edition (August 14, 2000)
- Verlag : Addison-WesleyProfessional
- ISBN : 0201675196, 978-0201675191
- 432 Seiten

## Einige zusätzliche Bücher

- Calculatrices, Traité d'électricité, volume xiv, Jean-Daniel Nicoud  
ISBN 2-604-00016-4
- Informatique industrielle I, Henri Nussbaumer ISBN 2- 88074-100-9
- Le matériel informatique : concepts et principes, Alfred Strohmeier  
ISBN 2-88074-092-4
- Microprocessor systems design, Alan Clements ISBN 0-534-94822-7
- Programmation concurrente, André Schiper ISBN 2-88074-093-2
- ARM Assembly Language : an Introduction, J.R. Gibson  
ISBN-13 : 978-1-84753-696-9
- Le langage C (Norme ANSI), B. W. Kernighan/D. M. Ritchie  
ISBN 2-10-048734-5
- Langage C, C. Delannoy ISBN 978-2-212-12445-3

## Datasheets, Reference Manuals

- Sitara TI AM335x Processor
  - ARMv7 Architecture Reference Manual
  - Cortex-A8 Technical Reference Manual
  - Cortex-A Programmer Guide
  - ARM Architecture Procedure Call Standard
  - ARM Instruction Set Summary
  - AM335x Technical Reference Manual
  - AM335x Sitara Processors
  - ARM Introducing NEON
- Beaglebone Black
  - BBB Reference Manual
  - BBB Black Schematics



# Toolchain

- GNU Toolchain
  - GNU as (assembler)
  - GNU bash (shell)
  - GNU binutils (binary utilities)
  - GNU gcc (compiler)
  - GNU ld (linker)
  - GNU make (maketool)
- OpenOCD
  - OpenOCD User's Guide
  - J-Link User Guide
  - GNU binutils (binary utilities)

# Toolchain

- Linux Utilities
  - Commandes de référence
- Git
  - Git Cheat Sheet (Tower)
  - Git Cheat Sheet (GitHub)
  - Git Cheat Sheet (git.or.cz)

# Referenzen

- Langage C, Christian Queinnec, Technique de l'ingénieur
- Le langage C, Peter Aitken et Bradley L. Jones, Edition Pearson
- C for Java Programmers, Niranjana Nagarajan, Department of Computer Science Cornell University
- Architecture des ordinateurs, IUP GEII – informatique et télécommunications, Patrick Marcel
- Architecture des systèmes à microprocesseurs, Maryam Siadat & Camille Diou
- Langage assembleur, exemple de l'assembleur ARM, Tarik Graba / Télécom ParisTech
- The "Clockwise/Spiral Rule", David Anderson

# Standards

- 754, IEEE Standard for Floating-Point Arithmetic, 29 August 2009
- International Standard, Programming Languages - C, N1570