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

Systèmes Embarqués 1 & 2

o.01 - Introduction et objectifs

Classes T-2/I-2 // 2018-2019

Daniel Gachet | HEIA-FR/TIC

o.01 | 13.09.2018

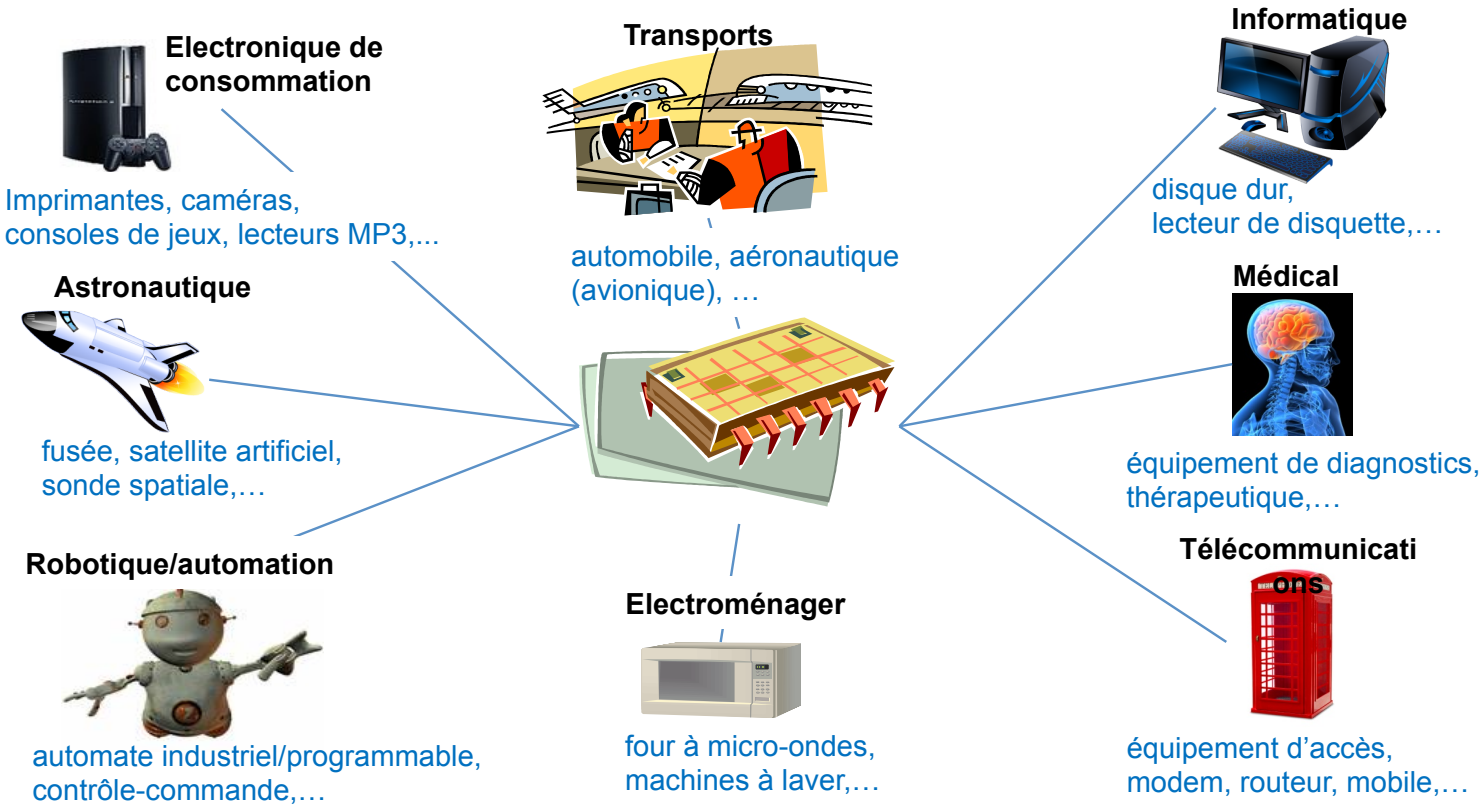
Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz

Contenu

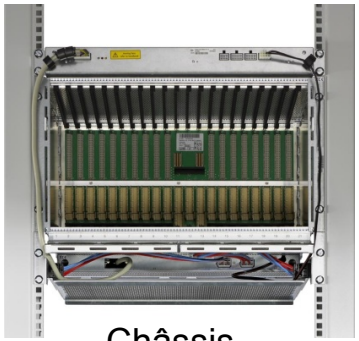
- Introduction
- Buts et objectifs
- Déroulement
- Evaluation
- Littérature et références

Applications des systèmes embarqués

- Grâce à l'évolution du silicium, les domaines d'applications des systèmes embarqués sont vastes et très variés
- Il n'existe pratiquement plus d'équipements modernes sans microprocesseur muni de logiciel plus ou moins complexe

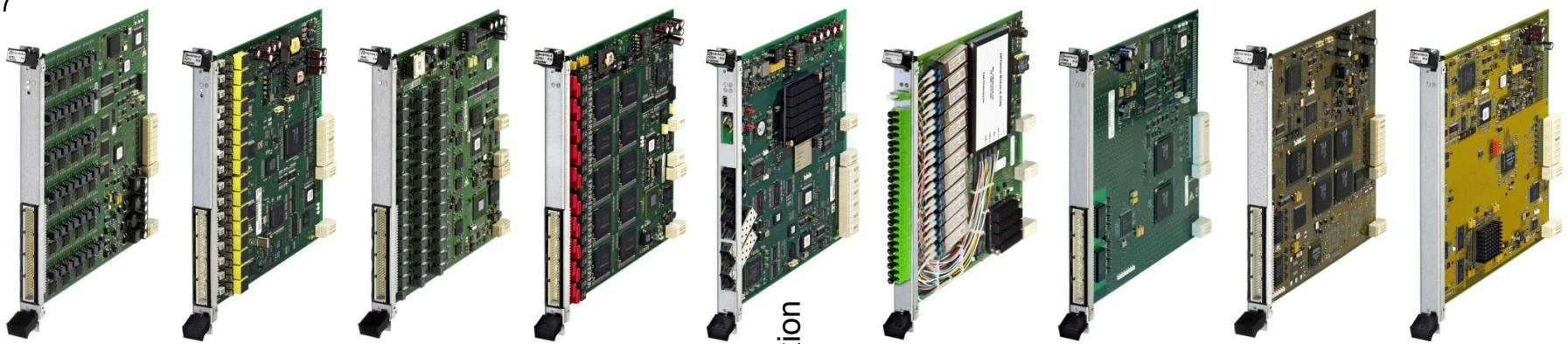


Exemple d'un équipement d'accès multiservices - KEYMILE



Châssis
Backplane

Ventilateurs



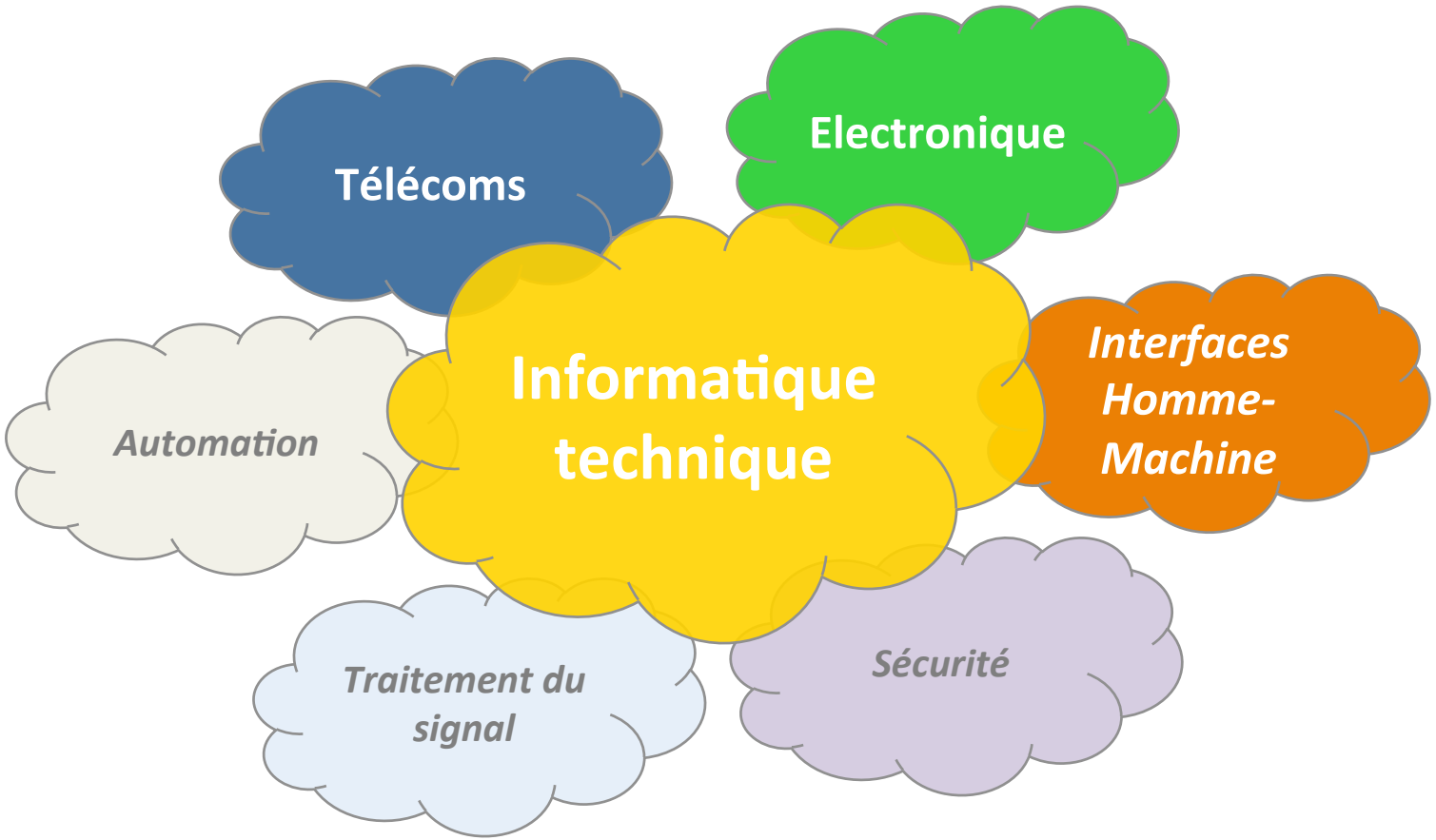
Exemple d'applications mobiles

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

Smart Phones Samung Galaxy S9
Samsung Exynos 9810 (octa-core 64 bits ARMv8-a)

Technologies

- Les systèmes embarqués exigent la mise en œuvre de technologies et de concepts très variés



8 grandes idées dans l'architecture des ordinateurs

- Conception pour les lois de Moore
- Utiliser des abstractions pour simplifier le design
- Rendre le cas général rapide
- Performance par le parallélisme
- Performance par le pipelining
- Performance par la prédiction
- Hiérarchie des mémoires
- Fiabilité par la redondance

[0] Réf. Computer Organization and Design, p. 11-12

Microprocesseur - Définition

- **Processeur**

  nom masculin (anglais processor).
  Organe destiné, dans un ordinateur, à interpréter et exécuter des instructions. Ensemble de programmes permettant d'exécuter sur un ordinateur des programmes écrits dans un certain langage.

  Selon le Larousse en ligne : http://www.larousse.fr/dictionnaires/francais/

- **Définitions (I)**

  nom masculin (anglais microprocessor, de to process, traiter).
  Processeur miniaturisé dont tous les éléments sont rassemblés en un seul circuit intégré.
  Commercialisés depuis 1971, les microprocesseurs sont les composants de base des micro-ordinateurs et sont utilisés dans de nombreux appareils pour doter ceux-ci de fonctionnalités évoluées et d'automatismes.

  Selon le Larousse en ligne : http://www.larousse.fr/dictionnaires/francais/

Définitions (II)

- **Système embarqué**

  Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. Ses ressources sont généralement limitées. Cette limitation est généralement d'ordre spatial (encombrement réduit) et énergétique (consommation restreinte).

  Selon wikipedia : https://en.wikipedia.org/wiki/Embedded_system

- **Système embarqué @ iSIS/HEIA-FR**

  Système mettant en œuvre du matériel et du logiciel afin de solutionner un aspect technique de manière économique, sans que l'utilisateur dudit système en soit conscient.

Objectifs

A la fin du cours, les étudiant-e-s seront capables de

- **Programmer en langage C**
  - Connaissance des types et déclarations de base
  - Connaissance des types complexes et des fonctions
  - Connaissance des pointeurs et des pointeurs de fonctions
  - Interfaçage assembleur – C
  - Introduction à la programmation orientée objet
  - Outils de développement (génération, validation, documentation)
- **Programmer en langage assembleur ARM**
  - Connaissance des registres internes et du jeu d'instructions
  - Traitement des nombres et des fanions (flags)
  - Programmation d'applications simples
  - Utilisation des µP par des langages de programmation évolués
- **Décrire l'architecture générale des microprocesseurs**
  - Architecture Von Neumann et Harvard
  - Architecture interne des microprocesseurs
  - Entrées/sorties, accès à la mémoire et aux périphériques

Objectifs (II)

- **Traiter les interruptions**
  - Connaissance du principe de base des interruptions
  - Traitement des interruptions (changement de contexte)
  - Entrées/Sorties (interruption vs. polling mode)
  - Systèmes d'exploitation (mécanismes de base - noyau)
- **Décrire l'architecture avancée des microprocesseurs**
  - Programmation du contrôleur d'accès direct à la mémoire (DMA)
  - Programmation de la mémoire cache
  - Programmation du gestionnaire de mémoire virtuelle (MMU)
- **Mettre en œuvre des systèmes à microprocesseurs**
  - Initialisation du µP et de ses périphériques
  - Lancement d'applications logicielles

Objectifs des travaux pratiques

Les laboratoires ont pour buts principaux

- **Assimilation de la matière étudiée dans les cours**
- **Mise en œuvre de µP et de systèmes embarqués**
- **Utilisation des outils de développement croisé pour des systèmes embarqués**
  - Chaîne d'outils pour la génération d'applications logicielles
  - Chaîne d'outils pour le débogage d'applications logicielles
- **Réalisation d'un petit projet de développement**
  - Gestion du projet
  - Spécifications, conception
  - Réalisation, vérification/validation
  - Documentation

Déroulement

- **Déroulement des séances de cours**
  - PC portables fermés s'ils ne sont pas nécessaires pour le cours
  - Explications sur transparent, au tableau et sur projecteur
  - Une partie des exercices sera résolue pendant le cours
  - Séries d'exercices à disposition
  - **Surtout, n'hésitez pas à interrompre le cours pour poser des questions !!!**
- **Déroulement des laboratoires**
  - Formule industrielle (tout à disposition, collaboration entre groupes, etc.)
  - Matériel Beaglebone Black (TI AM3358 / ARM Cortex-A8)
  - Ordre et discipline (matériel)
  - Pas de nourriture ou de boissons (matériel)
  - Remise en état/place du matériel après les séances

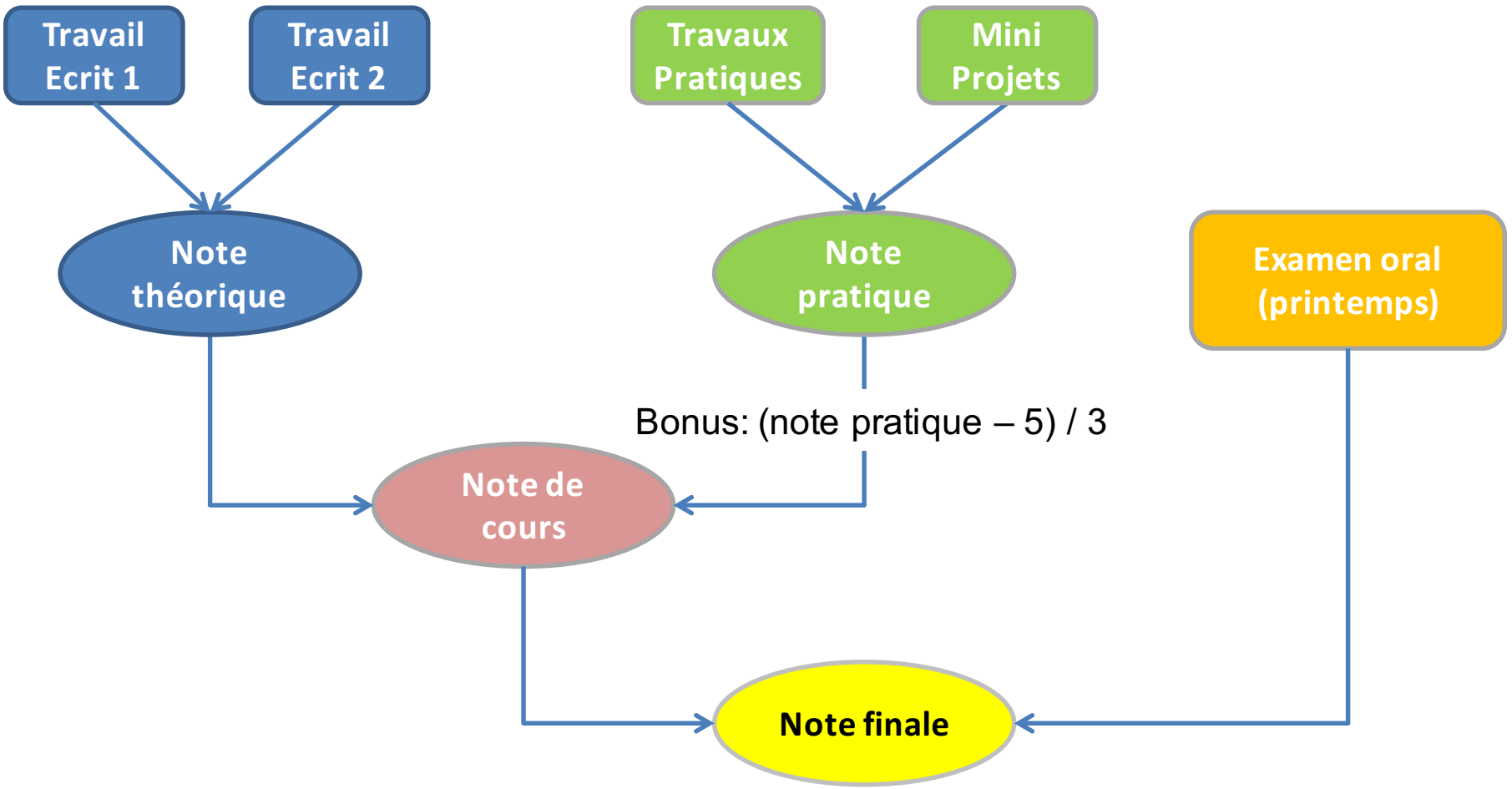
Tout le support de cours est disponible sous des dépôts Git de la HEIA-FR

- Support de cours
  https://gitlab.forge.hefr.ch/se12-1819/se12
- Travaux pratiques
  https://gitlab.forge.hefr.ch/se12-1819/tp

Evaluations

Cours théoriques
2 travaux écrits par semestre

Travaux pratiques



Travaux écrits

- **Conditions**
  - PC portable NON autorisé
  - Téléphone portable NON autorisé
  - Notes de cours ou personnelles NON autorisées
- **Réponses**
  - Peuvent être rédigés en français, allemand ou anglais
  - Réponses ambigües sont systématiquement mal interprétées
- **Absences**
  - Absence non justifiée → note minimale de 1
  - Pas de séance de rattrapage
- **Tricheries**
  - Examen terminé sur le champ → note minimale de 1

Travaux pratiques dirigés et projets intégrés

- **Conditions**
  - Tout le matériel est à disposition
  - Journal de laboratoire et code source sous Git, délai selon TP
- **Rapports**
  - Peuvent être rédigés en français, allemand ou anglais
- **Absences**
  - Absence non justifiée → note minimale de 1

Littérature

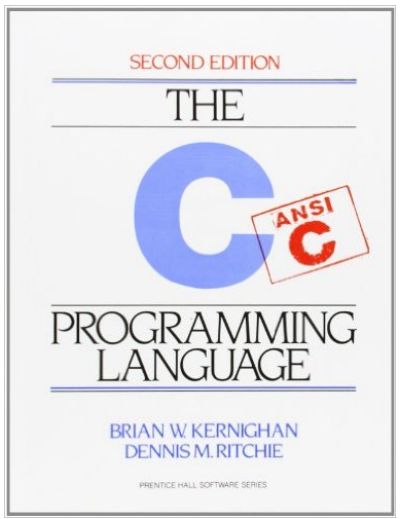
Quelques livres en ligne (E-books ENI)

http://www.mediapluspro.com/sites/hefr/bn

Quelques livres intéressants...

The C Programming Language

Brian W. Kernighan, Dennis M. Ritchie



Edition 2nd edition (April 1, 1988)

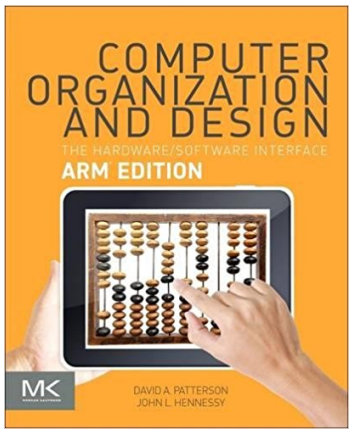
Publisher Prentice Hall

ISBN 0131103628, 978-0131103627

Length 272 pages

Computer Organization and Design

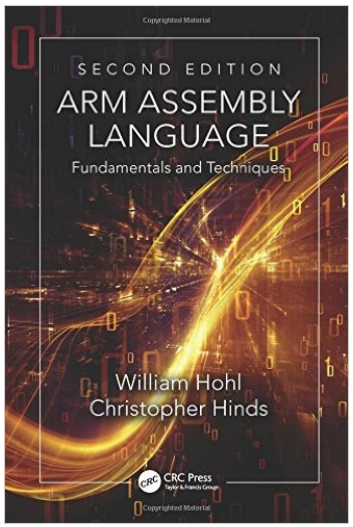
David A. Patterson, John L. Hennessy



Description	The Morgan Kaufmann Series in Computer Architecture and Design
Edition	1st edition (March 16, 2016)
Publisher	Morgan Kaufmann
ISBN	0128017333, 978-0128017333
Length	720 pages

ARM Assembly Language

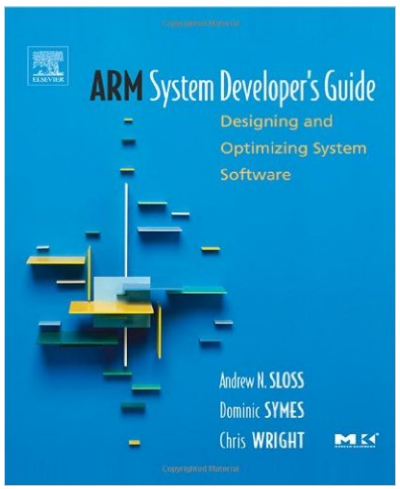
William Hohl, Christopher Hinds



Description	Fundamentals and Techniques
Edition	2nd edition (October 20, 2014)
Publisher	CRC Press
ISBN	1482229854, 978-1482229851
Length	453 pages

ARM System Developer's Guide

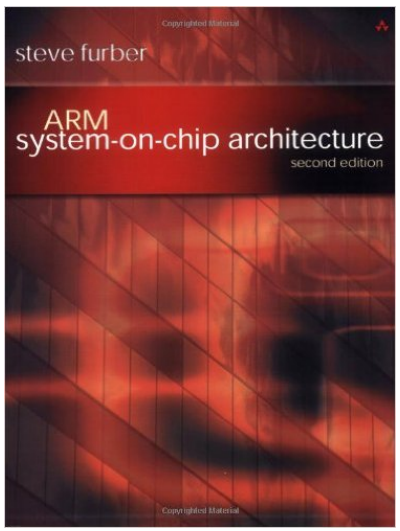
Andrew Sloss, Dominic Symes, Chris Wright



Description	Designing and Optimizing System Software
Edition	1st edition (April 8, 2004)
Publisher	Morgan Kaufmann
ISBN	1558608745, 978-1-55860-874-0
Length	689 pages

ARM System-on-Chip Architecture

Steve Furber



Edition	2nd edition (August 14, 2000)
Publisher	Addison-Wesley Professional
ISBN	0201675196, 978-0201675191
Length	432 pages

Quelques livres supplémentaires

- 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**
  - [01] ARMv7 Architecture Reference Manual
  - [02] Cortex-A8 Technical Reference Manual
  - [03] Cortex-A Programmer Guide
  - [04] ARM Architecture Procedure Call Standard
  - [05] ARM Instruction Set Summary
  - [06] AM335x Technical Reference Manual
  - [07] AM335x Sitara Processors
  - [08] ARM Introducing NEON
- **Beaglebone Black**
  - [01] BBB Reference Manual
  - [02] BBB Black Schematics

Toolchain

- **GNU Toolchain**
  - [01] GNU as (assembler)
  - [02] GNU bash (shell)
  - [03] GNU binutils (binary utilities)
  - [04] GNU gcc (compiler)
  - [05] GNU ld (linker)
  - [06] GNU make (make tool)
- **OpenOCD**
  - [01] OpenOCD User's Guide
  - [02] J-Link User Guide
  - [03] GNU binutils (binary utilities)
- **Linux Utilities**
  - [01] Commandes de référence
- **Git**
  - [01] Git Cheat Sheet (Tower)
  - [02] Git Cheat Sheet (GitHub)
  - [03] Git Cheat Sheet (git.or.cz)

Références et standards

Références

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

Standards

[01] 754, IEEE Standard for Floating-Point Arithmetic, 29 August 2009
[02] International Standard, Programming Languages - C, N1570