

# Microprogrammed Embedded Systems

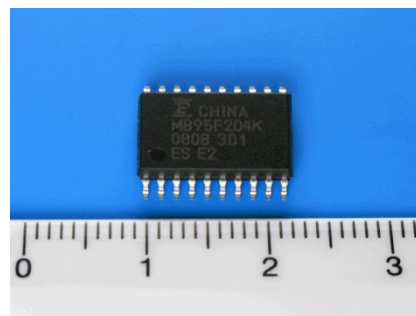
## Systemes Embarqués Microprogrammés

*Cours de 3<sup>ème</sup> année, Section d'Electricité  
Crédits: 4/semestre, Heures/Semaine: 4*

Prof. David Atienza Alonso  
Section de Génie Électrique et Électronique  
Faculty of Engineering (STI)

# Qu'est ce qu'un système embarqué?

- Traditionnellement:
  - Contrôleurs de machines à laver
  - Régulateurs de température du frigo
  - Vidéocassettes
  - ...



ASICs ou microcontrôleurs simples  
(8-bit PIC MB95F204K ou famille de 16-bit PIC)

- Récemment les systèmes embarqués sont microprogrammés, et ils sont... partout
  - Caméras, consoles de jeux, Smartphones, etc....



# Pourquoi microprogrammés?

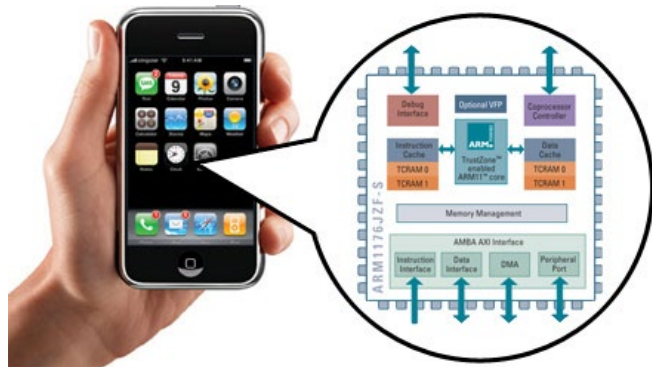
- Le business ultime: la convergence digitale
  - Un dispositif avec plusieurs fonctions!



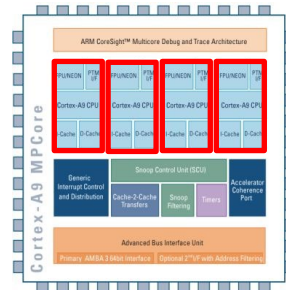
- Microprocesseurs de 32-bit
  - Plus de mémoire, plus de versatilité
  - 75% de tout le marché sont des cœurs **Advanced RISC Machines (ARM)** (entreprise fondée en novembre 1990, par Acorn, Apple and VLSI Tech.)

# Conception des systèmes embarqués microprogrammés actuels

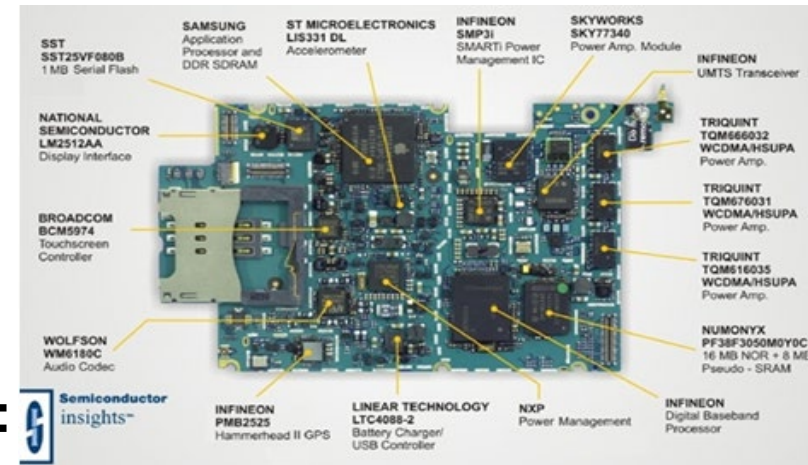
- Les systèmes embarqués sont très complexes
  - Microprocesseurs 32 bit, plusieurs types de mémoires, etc.,



**iPhone 3:  
ARM 1176JZF-S**



**iPhone 3GS/4S/5:  
ARM Cortex  
A9 MPCore**



**Système iPhone complet**

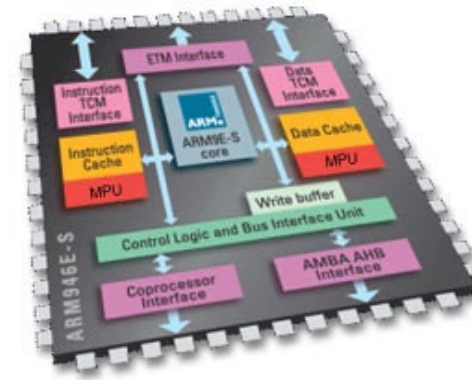
- Les ingénieurs EE doivent travailler avec tous les composants de ces nouveaux systèmes
  - Limités en capacités de mémoire et de performance
  - Valeur dans l'ensemble: compréhension et optimisation globale du système complet!

# Enseignement SEL à l'EPFL avec derniers systèmes embarqués microprogrammés

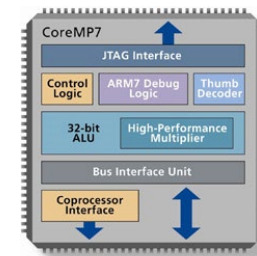
- Étudier les systèmes embarqués de hautes prestations en détail:
  - Microprocesseurs de 32 bit, mémoires, Bluetooth, boutons, touch-pad, écran tactile, wifi, son, etc.
  
- Ou utiliser un « système réel »: Nintendo **D**evelopment **S**ystem (DS)



**ARM946E-S**



**ARM7TDMI-S**

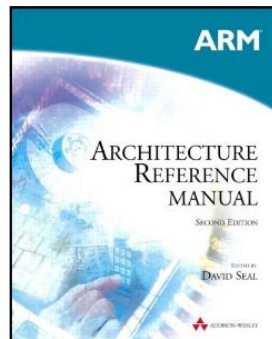


1. Introduction aux systèmes embarqués microprogrammés
  - Plate-forme physique de la Nintendo DS (NDS)
  - Introduction au logiciel de compilation et GUI
2. Microprocesseur et hiérarchie de mémoire
  - Architecture des microprocesseurs et mémoire dans la Nintendo DS
  - Programmation combinée C-assembleur
3. Gestion des E/S et des circuits périphériques
  - Description des périphériques et des E/S dans la Nintendo DS
  - Contrôleurs d'interruption simple
  - Interface et gestion des boutons, clavier, écrans (LCD et tactile) et son
4. Développement de « solutions complètes » sur la NDS
5. Développement en groupe (de deux): jeu, calendrier, etc.
  - Utilisation du logiciel pour programmation collaborative: GitLab

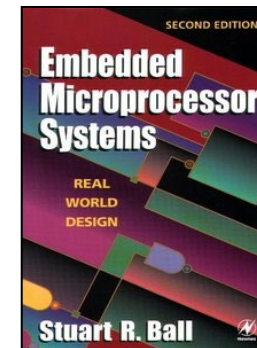
- Nintendo n'a pas de livre officiel: documentation en ligne et en anglais
  - Sites en ligne/GNU (NDS Homebrew, DevKitPro, DSDev, etc.)

- Bibliographie (en Anglais)

David Seal, "**ARM Architecture Reference Manual**". Addison Wesley, 2<sup>nd</sup> ed., 2001.



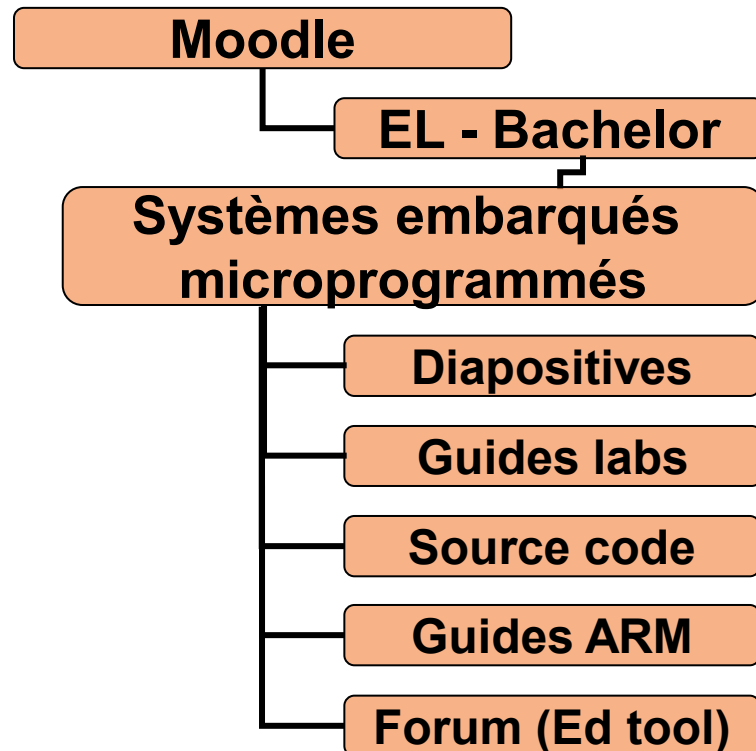
Stuart R. Ball, "**Embedded Microprocessor Systems**". Newness, 2<sup>nd</sup> ed., 2000.



- Bibliographie (en Français)

- Bernard Cassagne, « Introduction au langage C », 1998.
- Website: [www-clips.imag.fr/commun/bernard.cassagne/Introduction\\_ANSI\\_C.html](http://www-clips.imag.fr/commun/bernard.cassagne/Introduction_ANSI_C.html)

- Diapositives de théorie et guides de laboratoires
- Matériel supplémentaire (codes, binaries NDS, etc.)
- Questions? Forum: Ed tool




<https://edstem.org/eu/courses/2446/discussion>

<http://moodle.epfl.ch>

Electrical and Electronic Engineering (EL) / EL - Bachelor  
**Systèmes embarqués microprogrammés**  
 Course Participants Grades More ▾

▼ General Collapse all



**Objectifs:**  
 L'étudiant comprendra les architectures des systèmes embarqués microprogrammés, les architectures des microprocesseurs, hiérarchie de mémoire et les différents périphériques de Entrée/Sortie (E/S) inclus, utilisant comme étude de cas la plate-forme portable Nintendo DS.

**Objectives:**  
 The student will get to know the architecture of microprogrammed embedded systems, including the microprocessor architecture, memory hierarchy and different input/output peripherals, using as case study the Nintendo DS portable platform.

**Horaires:**

- Cours: Vendredi de 9h00 à 11h00 au BS 290
- Laboratoire: Vendredi de 14h00 à 16h00 au MED 2 2524

**Heures pour poser des questions:**

- Vendredi, 11h00 - 13h00, office ELG 130 (David Alierza)
- Mardi, 11h00 - 13h00, office ELG 121 (Stefano Albin)
- Mercredi, 11h00 - 13h00, office ELG 121 (Stasa Kostio)
- Jeudi, 11h00 - 13h00, office ELG 121 (Juan Sapritza)
- Jeudi, 14h00 - 16h00, office ELG 121 (Kai Zhu)
- Vendredi, 12h00 - 14h00, office ELG 134 (Dimitrios Samakzovis)

Vous pouvez également poser des questions online en utilisant le Forum "Questions et Réponses (via Ed tool)"

**Support du cours:**

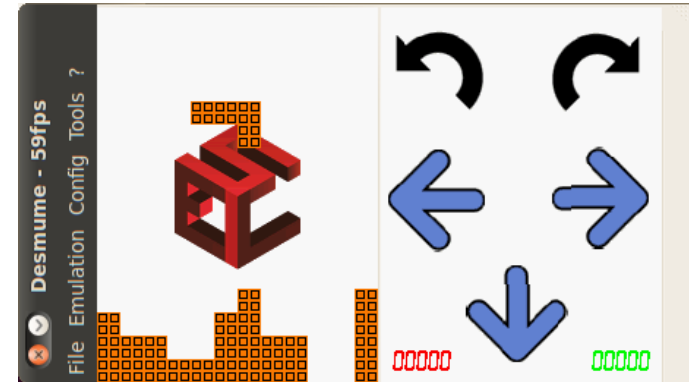
- Diapositives du cours et séries de laboratoire sur Moodle, et livres de référence à la bibliothèque indiqués pendant le cours.

Stable libnds version (Ubuntu) Mark as done

Stable libnds version (MacOS) Mark as done

- Il y aura deux sessions par semaine
  - Théorie (Vendredi 9h00 - 11h00), salle BS 260
  - Laboratoire (Vendredi 14h00 - 16h00), salle MED22524
- Suivi des séances de théorie et de laboratoire
  - En personne sur le campus.
  - Questions
    - Théorie et Laboratoire: Vous pouvez demander pendant la séance chaque chaque semaine.
    - Tutorat: Vous pouvez demander une séance par e-mail pendant les horaires disponibles sur le site Moodle

- La théorie et les séances de laboratoire sont entrelacés
  - Apprentissage des concepts théoriques à travers leur application sur la Nintendo DS (consulter impérativement Moodle chaque semaine)
- Séances: théorie (vendredi matin) + laboratoire (vendredi après midi)
  - Apprentissage des concepts de théorie (anglais)
  - Séances de laboratoires pour développer un jeu comme exemple de système (français ou anglais)



Un Kit de Nintendo DS individuel vous est prêté pour le travail à la maison

- L'évaluation est faite sur un examen et un projet final
  - Examen pendant les séances du cours (35%)
  - Développer en groupe de deux (ou seul(e)) une solution complète: jeux, agenda, système de contrôle, etc. (65%)

- Théorie:
  - Prof. David Atienza Alonso ([david.atienza@epfl.ch](mailto:david.atienza@epfl.ch))
  - Dr. Miguel Péon-Quiros ([miguel.peon@epfl.ch](mailto:miguel.peon@epfl.ch))
  - Dr. Darong Huang ([darong.huang@epfl.ch](mailto:darong.huang@epfl.ch))
- Laboratoire:
  - Mr. Dimitrios Samakovlis ([dimitrios.samakovlis@epfl.ch](mailto:dimitrios.samakovlis@epfl.ch))
  - Mr. Stefano Albini ([stefano.albini@epfl.ch](mailto:stefano.albini@epfl.ch))
  - Ms. Stasa Kostic ([stasa.Kostic@epfl.ch](mailto:stasa.Kostic@epfl.ch))
  - Mr. Juan Sapriza ([juan.sapriza@epfl.ch](mailto:juan.sapriza@epfl.ch))
  - Mr. Kai Zhu ([kai.zhu@epfl.ch](mailto:kai.zhu@epfl.ch))
- Horaire au tutorat (avec préavis préalable par e-mail):
  - Vendredi, 12h00 - 14h00, office ELG 134 (Dimitrios Samakovlis)
  - Mardi, 11h00 - 13h00, office ELG 121 (Stefano Albini)
  - Mercredi, 11h00 - 13h00, office ELG 121 (Stasa Kostic)
  - Jeudi, 11h00 - 13h00, office ELG 121 (Juan Sapriza)
  - Jeudi, 14h00 - 16h00, office ELG 121 (Kai Zhu)
  - Vendredi, 11h00 - 13h00, office ELG 130 (David Atienza)

# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- Pong (Sébastien Baumann, Salim Sunier)



# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- Kongey Donk (Niloufar Alipour Therani, Matthias Lambert)



# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

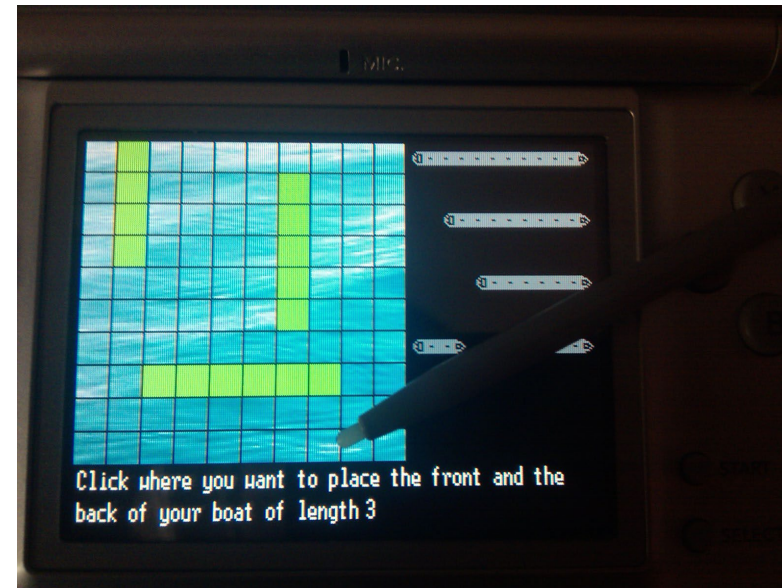
- Galaxy Flight (Gain Kim, Emilien Coulinge)



See video at: <https://www.youtube.com/user/eslepfl>

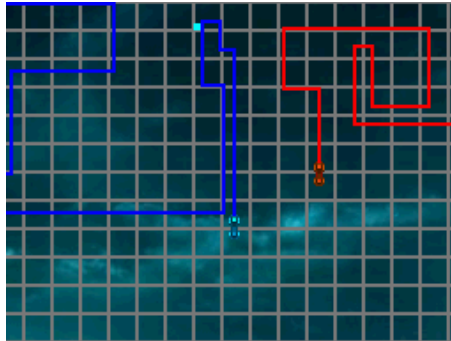
# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- Bataille Navale (Bastien Mösching, César Laurent)



# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- DS Tron (Niklaus Lehmann, Adrian Hauswirth)



EPFL

## Des étudiants en électronique se font la main sur des Nintendos

Par Jérôme Ducret. Mis à jour le 06.07.2012

En guise de travaux pratiques, une trentaine de participants au cours du professeur David Atienza ont créé des jeux et des applications maison pour la petite console.



1/5 Niklaus Lehmann, l'une des vainqueurs de la compétition de programmation sur Nintendo DS à l'EPFL. Lui et son collègue Adrian Hauswirth, absent ce jour-là, ont mis au point le DS-Tron, une course de motos, vue d'en haut à



# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- Digital Effects Synthesizer (Raffael Hochreutener)



# Exemple de projets avec Nintendo DS comme systèmes embarqués microprogrammés

- OSC-based Sequencer (Luis Gaemperle et Michael Tschannen)



Sequencer GUI



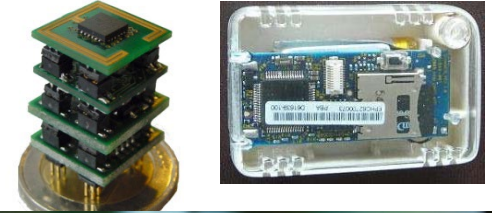
Wireless  
OSC – MIDI Translator



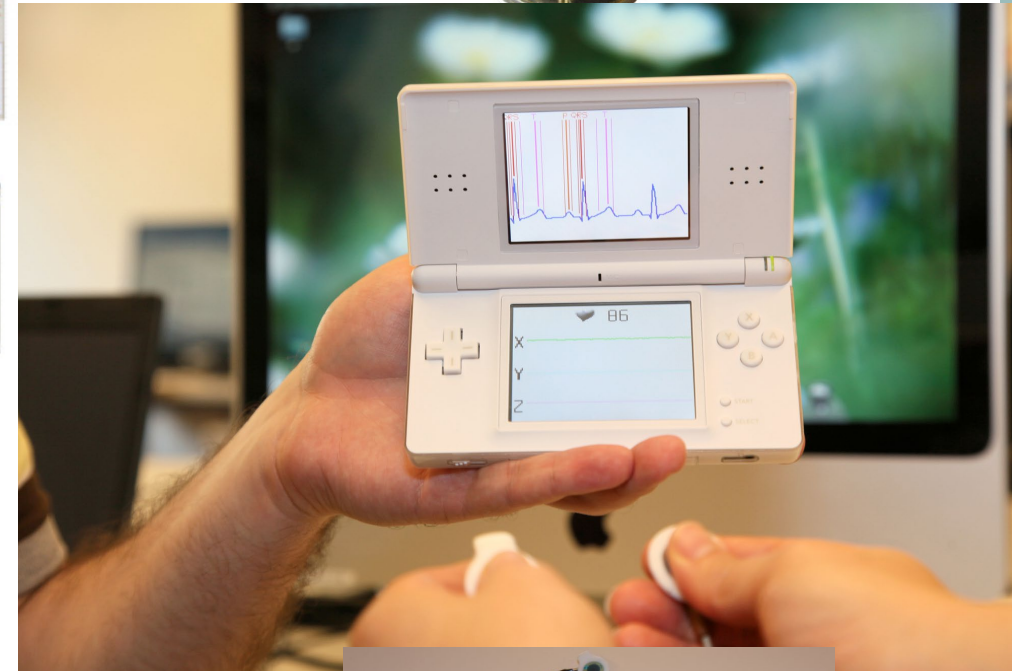
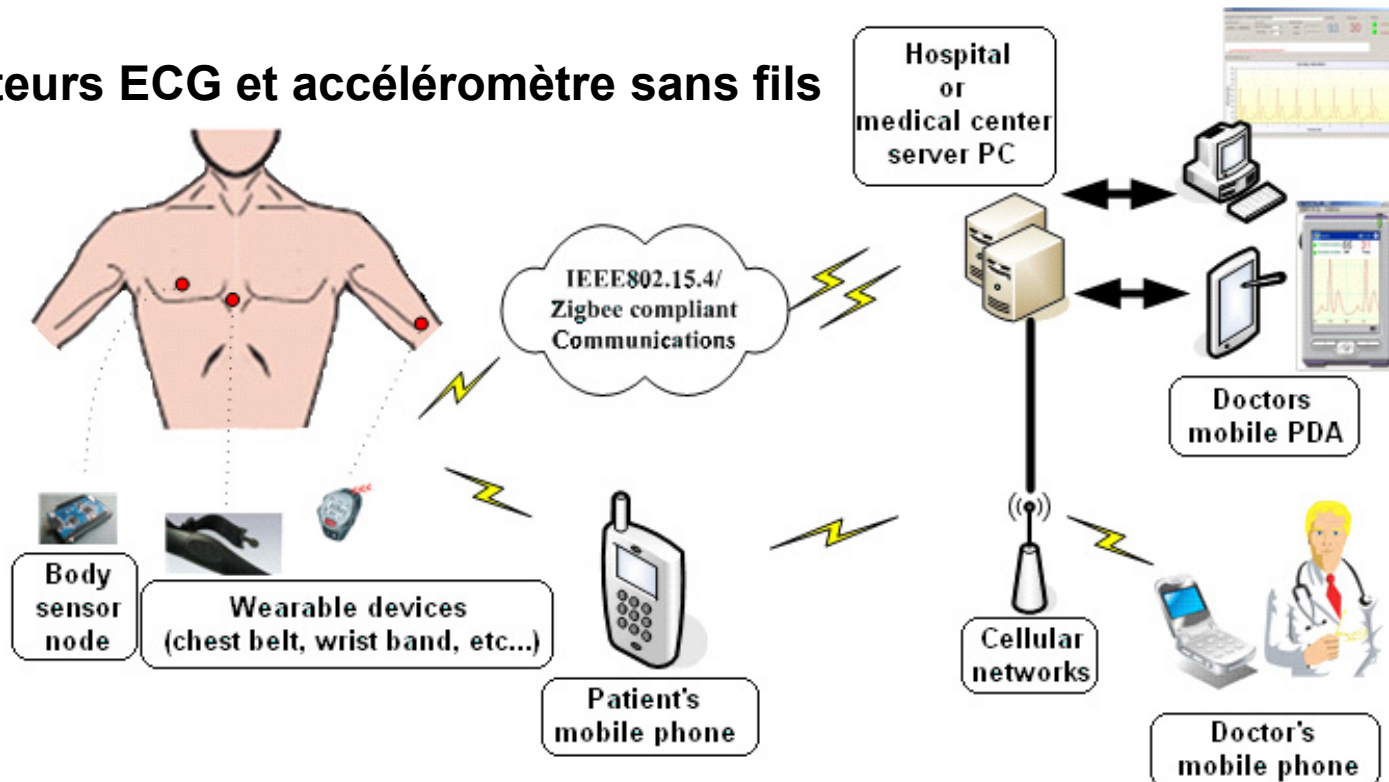
MIDI Instruments

# Projets à l'EPFL par des ingénieurs électriciens en systèmes embarqués microprogrammés

- Projet d'un système embarqué pour "healthy lifestyle" et fitness avec capteurs et la Nintendo DS
- Et aussi possible avec l'iPhone plus tard!



## Capteurs ECG et accéléromètre sans fils

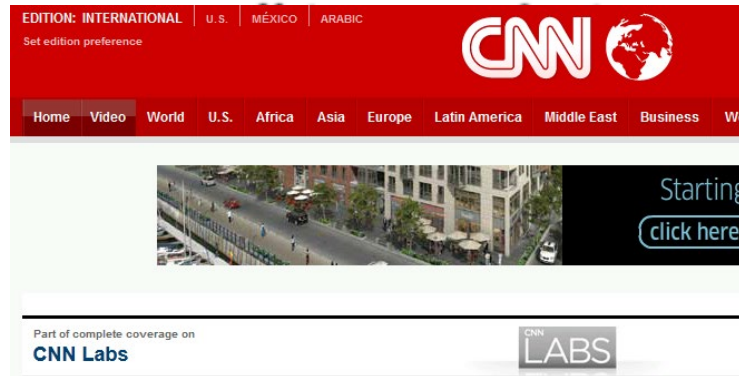


# New smart ECG monitors have considerable resonance in the media and the medical community

18 octobre 2011  
Le résumé de l'actualité romande

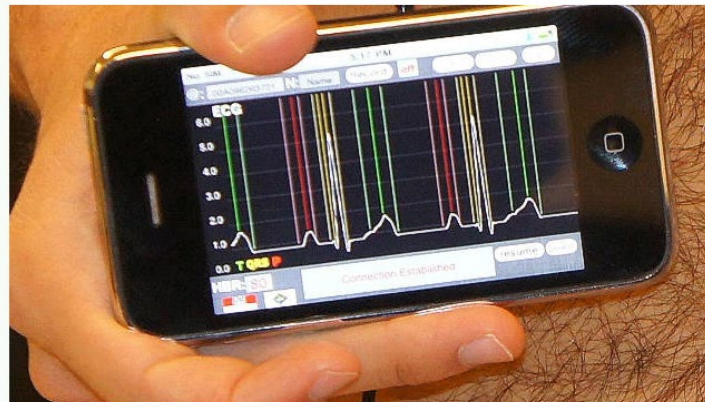


**Non-intrusive, light and operates for 4 weeks for a single battery recharge**

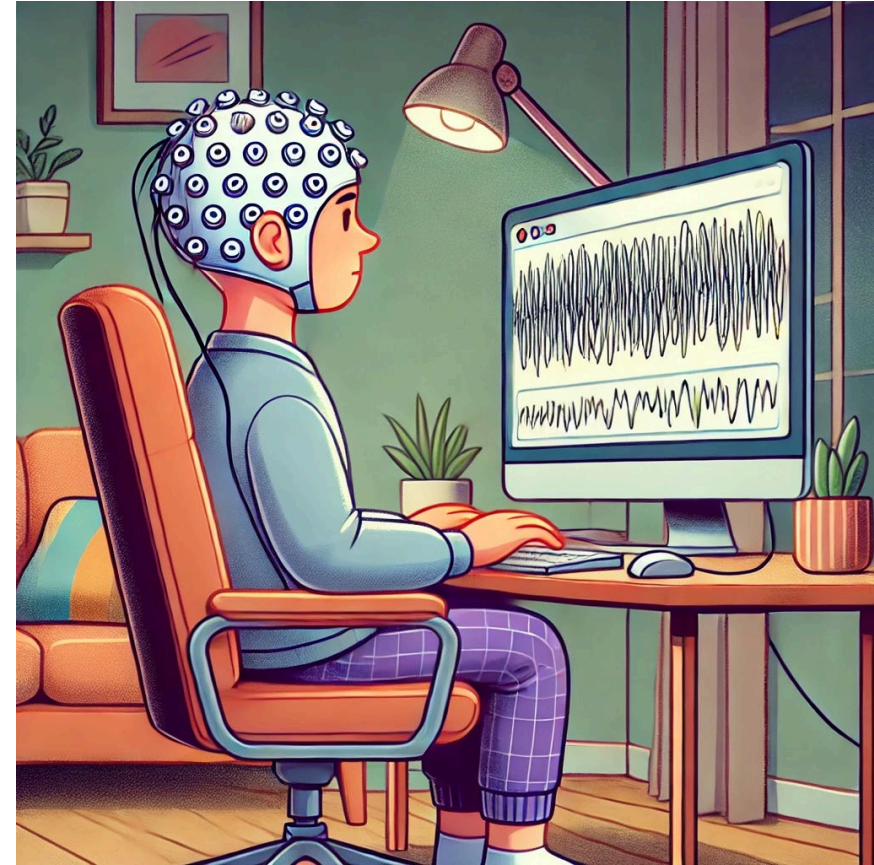


## Smartphone detects danger in a heartbeat

By Matthew Knight, CNN  
October 24, 2011 -- Updated 16:14 GMT (00:14 HKT) | Filed under: Mobile

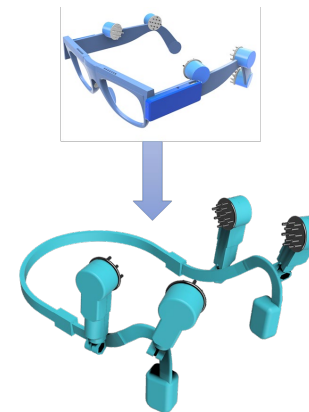
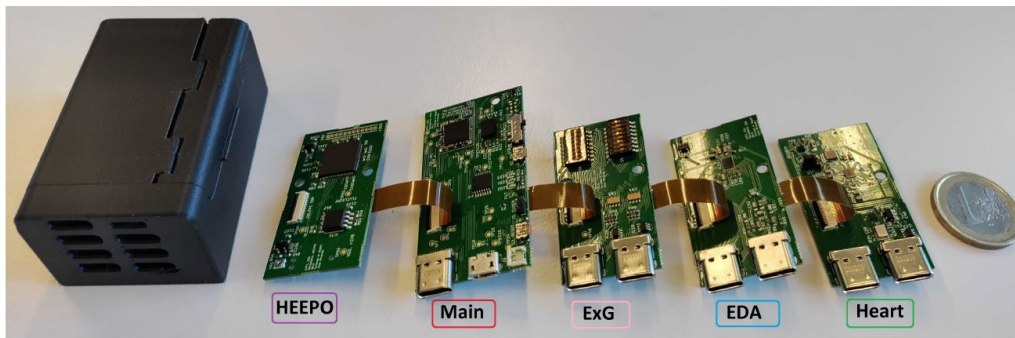
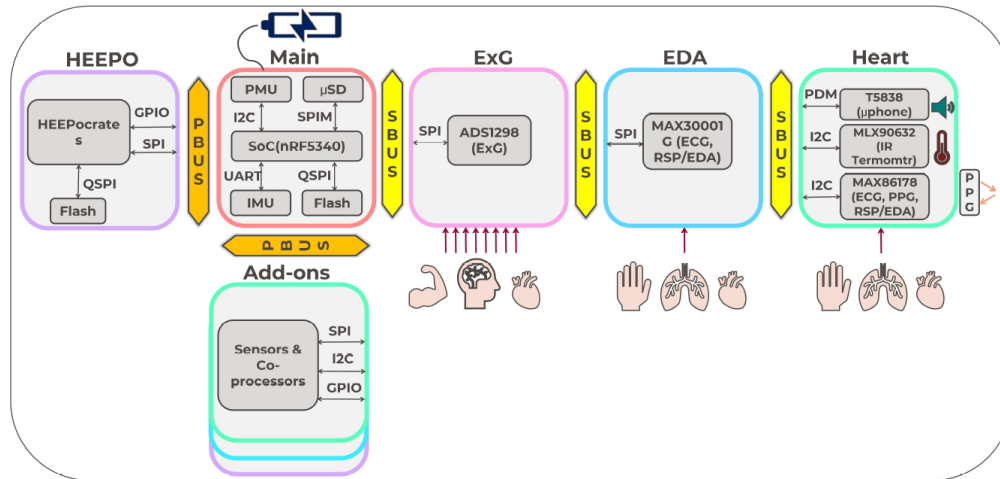
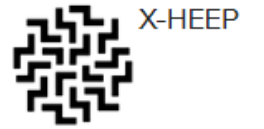


# What's Next - Continuous Monitoring at Home: Smart Embedded Systems



**New open-source platform for the complete EPFL ecosystem – Based on RISC-V**

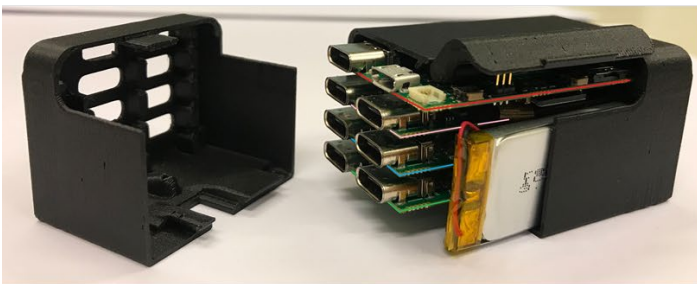
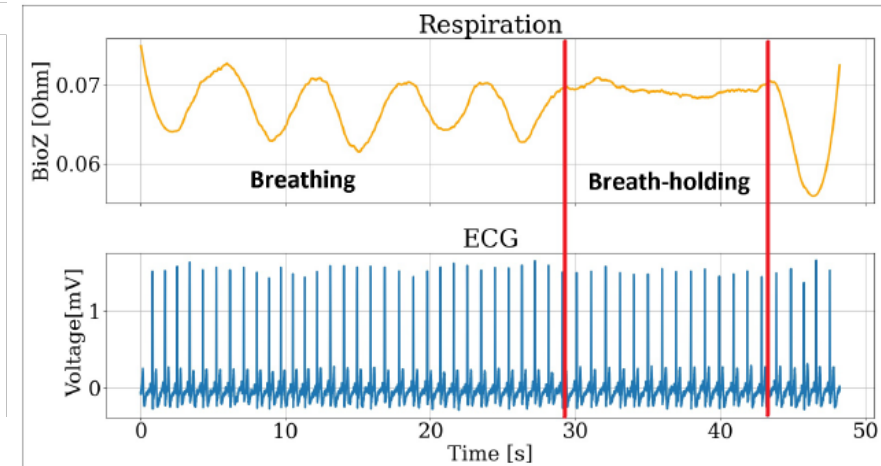
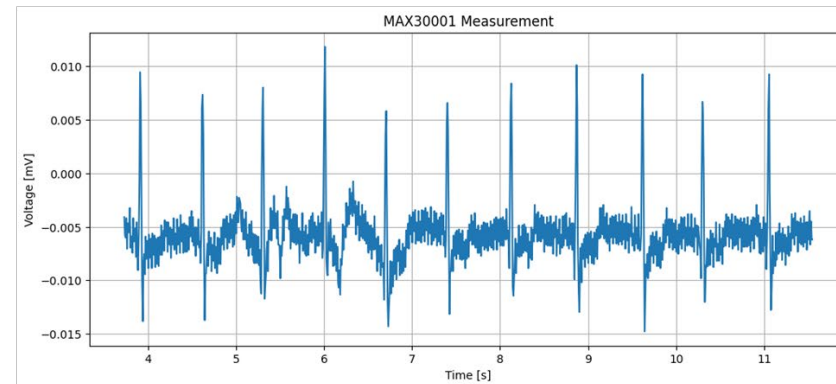
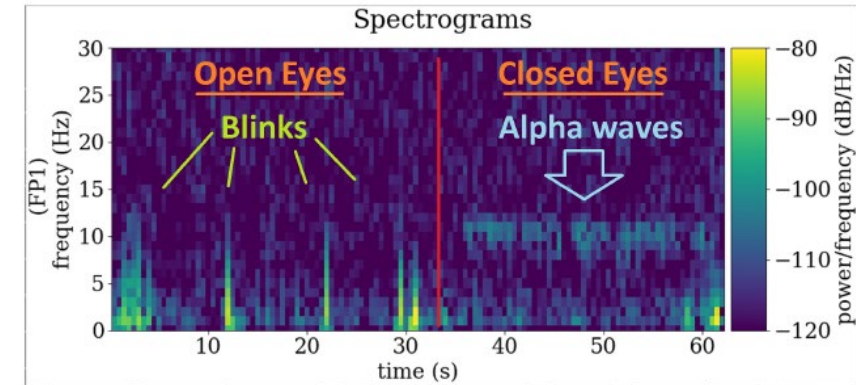
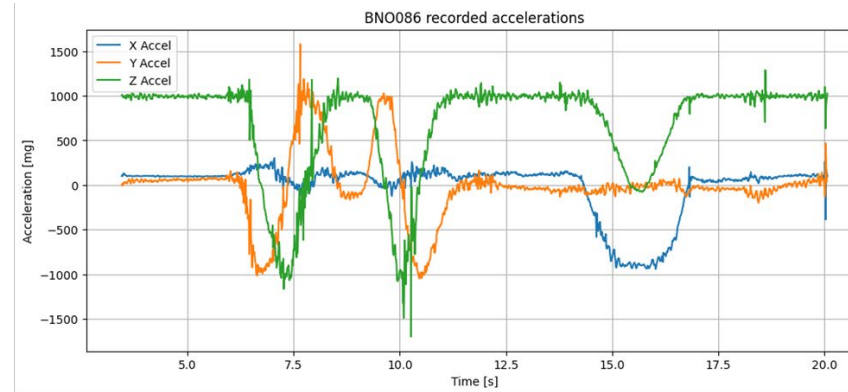
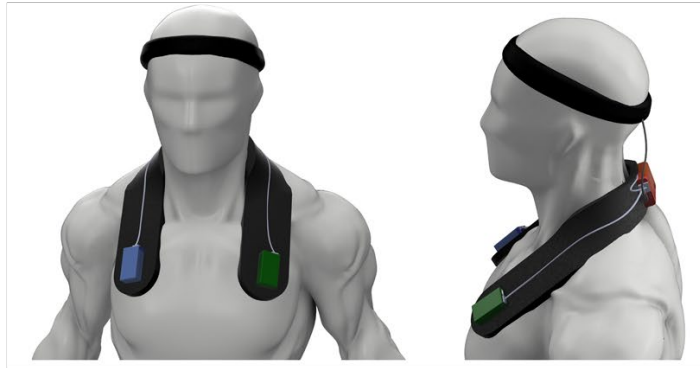
- New X-HEEP Open-Source Multi-Core Platform (RISC-V)



### Additional Information:

1. X-HEEP: <https://www.epfl.ch/labs/esl/research/systems-on-chip/x-heap/>
2. Versasens: <https://www.epfl.ch/labs/esl/research/smart-wearables/versasens/>

# VersaSens: Multi-Parametric System Able to Measure Many Bio-Signals in Real-Time



Source: <https://www.epfl.ch/labs/esl/research/smart-wearables/versasens/>

# Resources to learn more...

## X-HEEP Repository



<https://github.com/esl-epfl/x-heep>

## X-HEEP Documentation



<https://x-heep.readthedocs.io/en/latest/>

## FEMU Repository

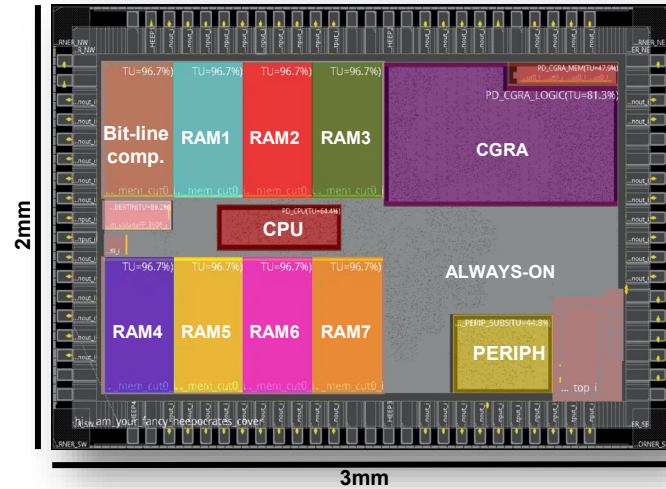


<https://github.com/esl-epti/x-heep-femu>

## FEMU SDK Repository

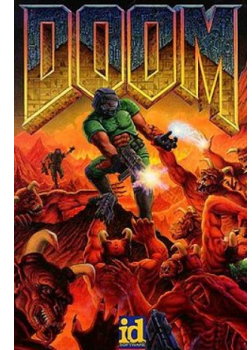


<https://github.com/esl-epfl/x-heep-femu-sdk>



```
[17:43:58:372] Z_Init: Init zone memory allocation daemon.
[17:43:58:436] zone memory: Using native C allocator.
[17:43:58:501] V_Init: allocate screens.
[17:43:58:533] M_LoadDefaults: Load system defaults.
[17:43:58:581] NRFD-TODO: I_BindInputVariables
[17:43:58:629] NRFD-TODO: I_BindJoyStickVariables
[17:43:58:677] NRFD-TODO: M_BindStrikeControls
[17:43:58:724] NRFD-TODO: M_BindWeaponControls
[17:43:58:772] NRFD-TODO: M_BindMagControls
[17:43:58:821] NRFD-TODO: M_BindMenuControls
[17:43:58:852] NRFD-TODO: M_BindChatControls
[17:43:58:900] NRFD-TODO: NET_BindVariables
[17:43:58:948] NRFD-TODO: LoadDefaultCollection
[17:43:58:996] NRFD-TODO: LoadDefaultCollection
[17:43:59:029] WAD File: doom.wad
[17:43:59:060] W_Init: Init WADfiles.
[17:43:59:092] adding doom.wad
[17:43:59:124] W_CheckCorrectWAD
[17:43:59:140] D_IdentifyVersion
[17:43:59:172] InitGameVersion
[17:43:59:204] NRFD-TODO: game detect
[17:43:59:236] Game version:
[17:43:59:268] Emulating the behavior of the 'Doom 1.9' executable.
[17:43:59:332] NRFD-TODO? W_GenerateHashTable
[17:43:59:380] D_SetGameDescription
[17:43:59:412] NRFD-TODO: M_GetSaveGameDir
[17:43:59:460]
=====
DOOM Shareware
=====
[17:43:59:556]
[17:43:59:620]
[17:43:59:716] NRFD-TODO: NRFD Doom is free software, covered by the GNU General Public
[17:43:59:812] License. There is NO warranty, not even for MERCHANTABILITY or FITNESS
[17:43:59:908] FOR A PARTICULAR PURPOSE. You are welcome to change and distribute
[17:44:00:004] copies under certain conditions. See the source for more information.
[17:44:00:011]
=====
[17:44:00:197] I_Init: Setting up machine state.
[17:44:00:245] M_Init: Init miscellaneous info.
[17:44:00:292] R_Init: Init DOOM refresh daemon
[17:44:00:324] R_InitTextures
[17:44:00:363] W_GetNumForName: P_NAMES not found!
```

## Nice, but...Can it run Doom of Pentium age?!



[Source: Wikipedia]

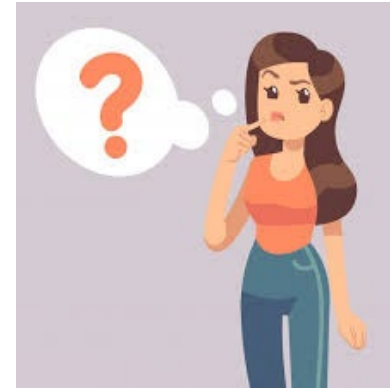
Processor	Intel Pentium P54C	EPFL/ESL HEEPocrates
Architecture	X86 (32 bits)	RISC-V (32 bits)
Release year	1994	2023
Purpose	General purpose, HPC, gaming	Ultra-low power, healthcare, wearables
Frequency	100 MHz	Up to 470 MHz
Technology	Intel 600 nm	TSMC 65 nm
Voltage (V)	3.3	0.8 – 1.2
Area (mm <sup>2</sup> )	~ 150	6
On-chip SRAM	16 KB (8+8)	256 KB (8x 32 KB)
Max Power (W)	10 – 15	0.048 (< 0.5 %)

**X-HEEP designed in ESL and other EPFL labs!**  
<https://www.epfl.ch/labs/esl/research/systems-on-chip/x-heep/>

**A first attempt...  
Doom start-up done,  
missing rendering!**

**New Bachelor semester project possible to complete it, let us know if interested!**

# Questions?



**Aujourd'hui: Essayer la Nintendo DS...**