Description du module

Informatique temps-réel embarquée

Généralités

Nombres de crédits ECTS

3

Sigle du module

TSM_EmbReal

Version

30 août 2009

Responsable du module

Hans Buchmann, FHNW

Langue

	Lausanne	Berne	Zurich
Enseignement	☒ E ☒ F	☐ D ☐ E ☐ F	☒ D ☒ E
Documentation	☒ E ☐ F	☐ D ☐ E ☐ F	☐ D ☒ E
Questions d'examen	☒ E ☒ F	☐ D ☐ E ☐ F	☒ D ☒ E

Catégorie du module

☐ Bases théoriques élargies

☒ Approfondissement technique et scientifique

☐ Modules de savoirs contextuels

Périodes

☒ 2 périodes d'enseignement frontal et une période d'exercice par semaine

☐ 2 périodes d'enseignement frontal par semaine

Brève description /Explication des objectifs et du contenu du module en quelques phrases

Le module «**informatique temps-réel embarquée**» réunit les principaux sujets, concepts et problématiques du développement d'applications embarquées. Les systèmes embarqués comprennent les environnements exempts de système d'exploitation ainsi que des environnements plus complexes intégrant un noyau temps réel et des fonctions supplémentaires. Ce module traitera des étapes de développement importantes pour ce genre d'applications, depuis les phases de conception, de développement et d'implémentation, jusqu'au déploiement, au déboguage et à leur maintenance.

Objectifs, contenu et méthodes

Objectifs d'apprentissage et compétences visées

Au terme de ce module, l'étudiant(e) sera capable de :

- décrire les différentes architectures des systèmes embarqués sur lesquels les solutions logicielles peuvent être déployées;
- reconnaître les différents types d'interactions synchrone/asynchrone entre le matériel et le logiciel;
- déduire à partir des besoins spécifiques à un domaine, les architectures logicielles et les spécifications des interfaces;
- concevoir une application embarquée en prenant en compte les contraintes et limitations liées au matériel ainsi qu'à son temps de réaction;
- comprendre comment les modèles comportementaux sont vérifiés et implémentés;
- expliquer les bases des systèmes d'exploitation temps réel;
- discuter des propriétés et des conséquences liées à l'utilisation des différents ordonnanceurs (multitâches et temps réel);
- discuter des avantages et des inconvénients des systèmes d'exploitation du marché et de choisir le plus adéquat pour une application spécifique;
- expliquer de quelle manière les applications sont développées à l'aide de formalismes de programmation et d'environnements de modélisation, de façon à pouvoir les utiliser sur différents environnements d'exécution (avec/sans système d'exploitation, exécutif, moniteur, etc.);
- écrire des applications embarquées comportant la gestion du matériel (*drivers*, *bootstrap*, etc.);
- déverminer les applications à l'aide d'outils de mise au point (*debugging*) et de traçage;
- porter des applications d'une architecture à une autre;
- éviter les pièges caractéristiques des logiciels temps réel embarqués.

Contenu du module avec pondération des contenus d'enseignement

Développement de l'application

- Analyse de la problématique:
  - Ciblage du domaine, processus et équipement, fonctionnalité et temps de réponse, isolation des problématiques par domaine, architecture logicielle générique et process de développement.
- Modélisation de composants réactifs:
  - Abstraction basée sur le modèle, formalisme et méthode, automates finis interactifs, architecture et comportement, composition synchrone et asynchrone, correspondance comportementale, vérification modèle et génération de code.
- Connexion input/output de composants réactifs:
  - Interaction de processus externes et composants réactifs, spécification de couche d'interaction, fonction d'interaction telles que services inter-couches, exécution des fonctions d'interaction basée sur un cycle.
- Concepts de programmation en temps réel:
  - Algorithmes de programmation statiques et dynamiques, priorités et délais, architecture d'exécution basée sur un cycle, serveurs de tâches en temps réel, hiérarchies de programmation, vérification de la programmabilité.

Environnement d'exécution pour système en temps réel

- Concurrence
  - Multiprogrammation (multithreading), processus vs threads (processus légers), problèmes de course (racing), solutions par attente active, instructions atomiques (TAS, CAS, spin-locks), sans attente (masquage d'interruptions, wait-free), par attente passive (verrous, sémaphores, conditions, barrières, etc.), mécanismes spécifiques (BKL, RCU, etc.).
- Système d'exploitation temps réel
  - Architecture des systèmes d'exploitation et sous-systèmes, espaces utilisateurs et noyau (kernel), appels système, gestion de la mémoire, ordonnanceur de tâches, communication et synchronisation entre les tâches, systèmes d'exploitation du marché.
- Multicœur et virtualisation
  - Architecture et processeurs multicœur, SMP (Symmetric MultiProcessing), Hyper-Threading, moniteur de machine virtuelle, full-/para-virtualisation, séparation des espaces d'adressage virtuels et physique.
- Gestion des défaillances

Interactions liées au matériel/logiciel

- Architecture du système
  - Architecture générique, machines à états finis, monoprocesseur et multiprocesseur, coprocesseurs, multitraitement (multiprocessing) symétrique, accès mémoire non-uniforme.
- Conception matérielle avec contraintes temps réel
  - Interactions entre bus, architecture de la mémoire, mémoire hiérarchique, cache, unité de gestion de la mémoire, contrôle d'interruptions.
- Accès logiciel aux ressources matérielles
  - BSP (Board Support Package), bootloader, démarrage du système, déboguage matériel, JTAG, systèmes de fichiers embarqués (flash, RAM, etc.).
- Drivers (pilotes logiciels)
  - Architecture de drivers, types de drivers (caractère, bloc, réseau, etc.), modes et niveaux de priorité du processeur, communication inter-modes, concurrence dans le noyau, interactions entre espaces utilisateurs et noyau, DMA, routines de service d'interruptions et traitements différés.

Méthodes d'enseignement et d'apprentissage

- Enseignement magistral
- Exercices
- Etude personnelle (études de documents, études de cas)

Connaissances et compétences prérequises

Les étudiants ont des connaissances professionnelles en :

- Programmation
- Architecture des ordinateurs
- Fondements des systèmes d'exploitation

Bibliographie

- Burns, A. Wellings. Real-Time Systems and Programming Languages. Addison-Wesley, third edition, 2001.
- F. Cottet, J. Delacroix, C. Kaiser, Z. Mammeri. /Scheduling in Real-Time Systems/. Wiley, 2002.
- G. C. Buttazzo. Hard Real-Time Computing Systems. Predictable Scheduling Algorithms and Applications. Second Edition, Kluwer Academic Publishers, 2005.

- L. Zaffalon. Programmation synchrone de systèmes réactifs avec Esterel et les SyncCharts. PPUR, 2005. (English Version planned for late 2008)
- Colin Walls. Embedded Software: The Works. Newnes, 2006
- Jack Ganssle. The Firmware Handbook. Newnes, 2004

Mode d'évaluation

Conditions d'admission aux examens de fin de module (tests exigés)

Aucune

Examen écrit de fin de module

Durée de l'examen:	120 minutes
Moyens autorisés:	Résumé personnel