

Sommaire :

Partie I : Concepts fondamentaux et problématique du temps réel

Cette introduction pose les bases et définit ce qui différencie l'informatique classique de l'informatique industrielle sous contraintes de temps.

- **Chapitre 1 : Définition des systèmes temps réel de contrôle-commande**
 - Caractéristiques d'une application temps réel.
 - Notion de temps réel strict (*hard*) et souple (*soft*).
 - Spécificités des systèmes embarqués et de contrôle-commande.
- **Chapitre 2 : Contraintes temporelles et déterminisme**
 - Échéances, périodicité, temps de réponse, gigue (*jitter*).
 - Importance de la prédictibilité du système.

Partie II : Méthodologies de spécification et de conception

Cette partie détaille comment traduire un besoin industriel en une architecture de fonctions et de données logiques exploitables.

- **Chapitre 3 : La méthode SA-RT (Structured Analysis for Real-Time)**
 - Modélisation des flux de données (DFD) et des flux de contrôle (CFD).
 - Spécifications de contrôle (Spécifications d'état/transitions).
- **Chapitre 4 : La méthode DARTS (Design Approach for Real-Time Systems)**
 - Passage du modèle d'analyse au modèle d'architecture logicielle.
 - Structuration de l'application en un ensemble de tâches concurrentes.

Partie III : Noyaux temps réel et ordonnancement (Implémentation)

Une fois l'architecture en tâches définie, cette section explique comment un système d'exploitation temps réel (RTOS) gère l'exécution de ces tâches.

- **Chapitre 5 : Programmation multitâche et exécutifs temps réel**
 - Structure d'un noyau temps réel (RTOS).
 - Mécanismes de synchronisation (sémaphores, mutex) et de communication (boîtes aux lettres, files de messages).
- **Chapitre 6 : Politiques d'ordonnancement**
 - Ordonnancement à priorités fixes : RMS (*Rate Monotonic Scheduling*), DM (*Deadline Monotonic*).
 - Ordonnancement dynamique : EDF (*Earliest Deadline First*).
 - Tests de faisabilité et d'ordonnancabilité (calcul des bornes de charge CPU).
- **Chapitre 7 : Partage de ressources et inversion de priorité**
 - Le problème de l'inversion de priorité.
 - Protocoles de résolution : PIP (*Priority Inheritance Protocol*) et PCP (*Priority Ceiling Protocol*).

Partie IV : Environnements technologiques et études de cas

La dernière partie est dédiée à la mise en pratique à travers des langages spécifiques et des cas concrets du monde industriel.

- **Chapitre 8 : Langages de programmation pour le temps réel**
 - Utilisation du **Langage C** (programmation système).
 - Spécificités du langage **Ada** pour le temps réel strict.
 - Approche graphique avec **LabVIEW**.
- **Chapitre 9 : Études de cas industrielles**
 - Conception complète d'une application de contrôle-commande depuis le cahier des charges jusqu'à l'implémentation multitâche.
 - Validation temporelle de l'implémentation.