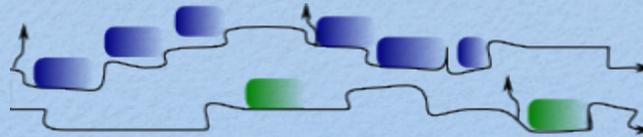


LEJOS RT

MACHINE VIRTUELLE JAVA TEMPS RÉEL



MOTIVATIONS POUR LEJOS RT



- Disposer d'une **plateforme ludique** pour l'enseignement de la **programmation temps réel en Java** (www.lejosRT.org). Version actuelle 2.1
- Permettre la **mise en oeuvre de cas concrets** grâce aux différents capteurs (contrôler le déplacement d'un robot, réagir à un contact, à une élévation de température, éviter un obstacle).
- Permet le **développement et l'utilisation des mécanismes (algorithmes) RTSJ** (ordonnancement / moniteurs de synchronisation) non disponibles dans les plateformes classiques libres (LinuxRT/RI).



Lego NXT

Processeur principal: Atmel 32-bit ARM processor,
AT91SAM7S256

256 KB FLASH, 64 KB RAM, 48 MHz

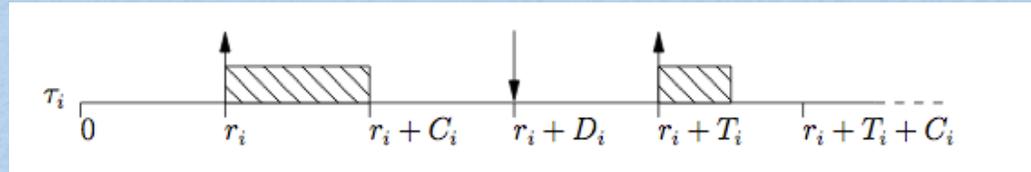
Coprocesseur: Atmel 8-bit AVR processor, ATmega48

4 KB FLASH, 512 Byte RAM, 8 MHz

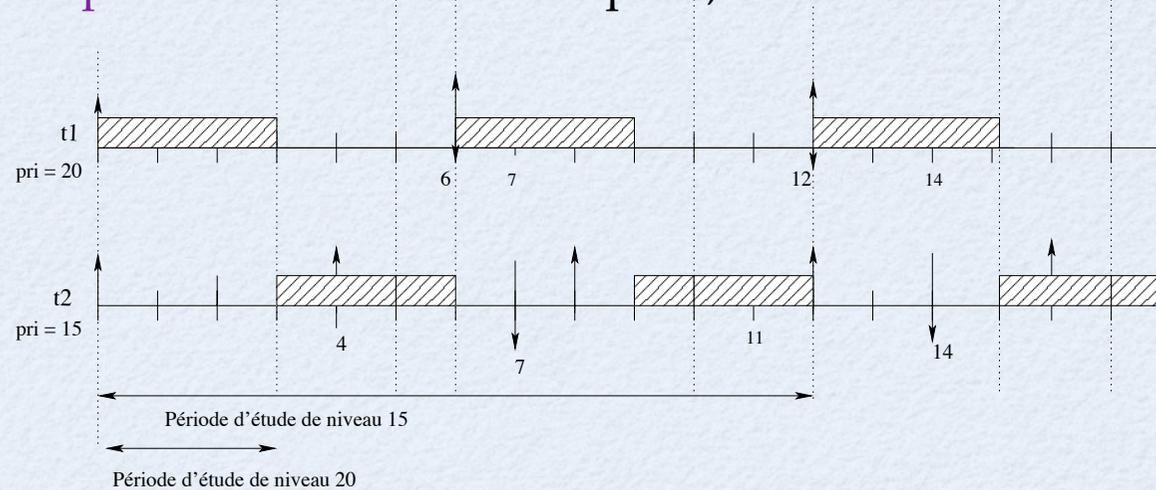
I/Os: 4 ports d'entrée / 3 ports de sortie / bluetooth / USB 2.0

écran LCD (100x64pixels) / haut parleurs (8bit/8khz)

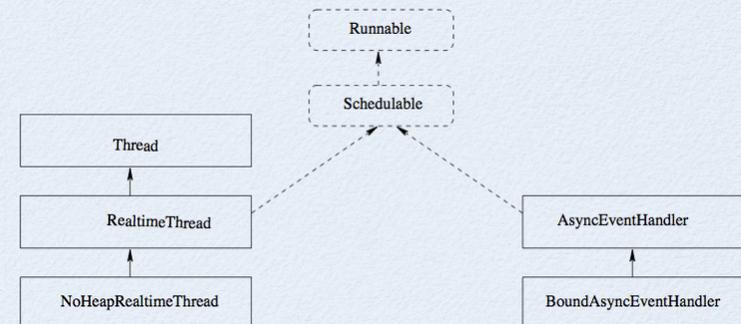
MODÈLES DE TÂCHE



- trafic périodique / apériodique / sporadique
- modèles de contrainte:
 - contraintes temporelles: échéance sur requête, échéance contrainte et échéance arbitraire

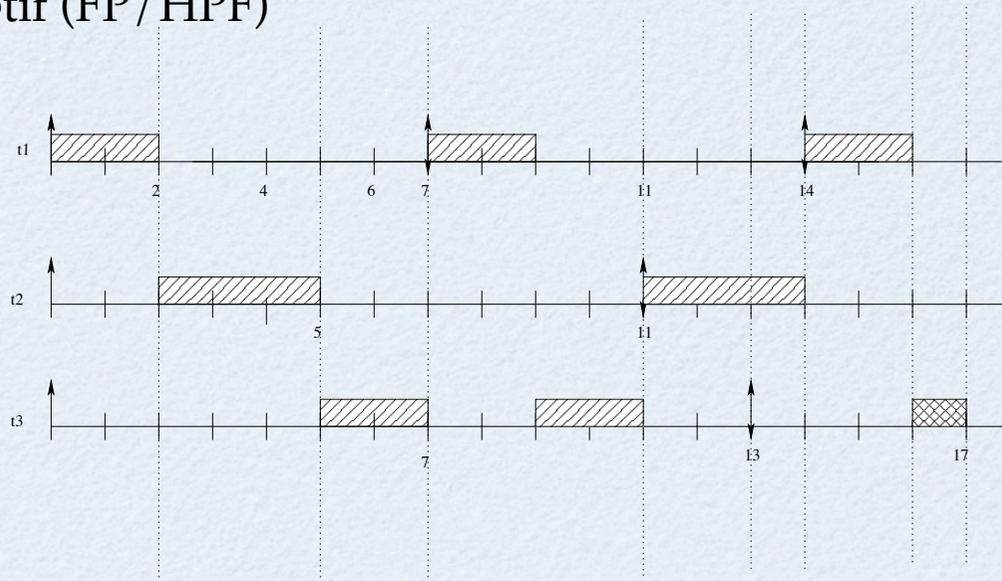


- contraintes de précedence: tâche en graphe
 - (prochaine version)
- entités ordonnables (RTT et AEH)

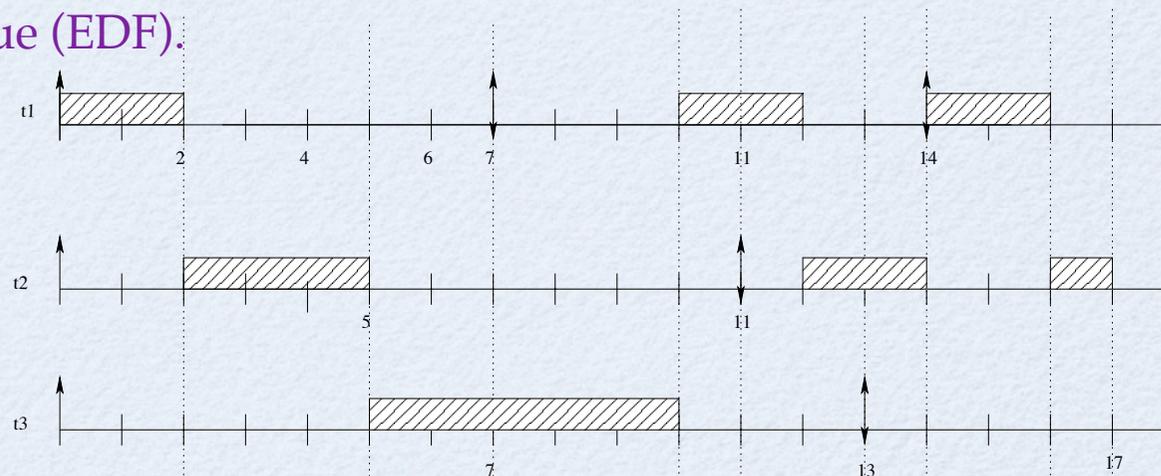


MODÈLE D'ORDONNANCEMENT - LEJOS RT

- **Priorité Fixe Préemptif (FP / HPF)**

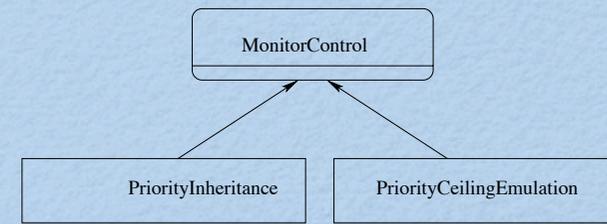


- **et Priorité Dynamique (EDF).**

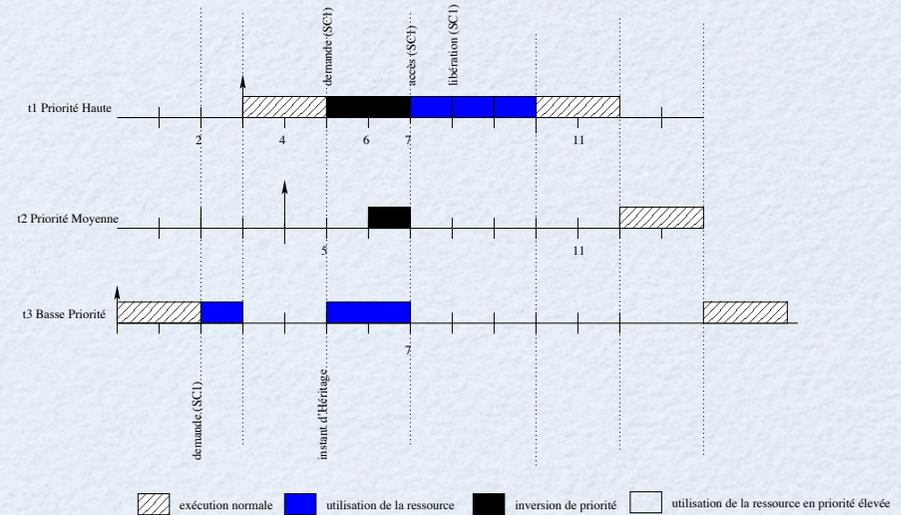
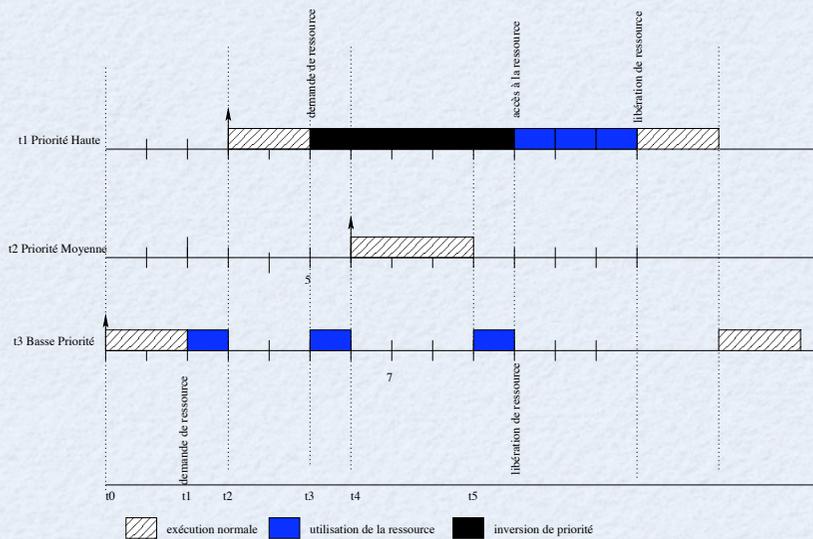


- **à venir: RateMonotonic / DeadlineMonotonic / FIFO (pour les tâches de même priorité) / Ordonnancement des tâches en graphe**

MONITEURS DE SYNCHRONISATIONS

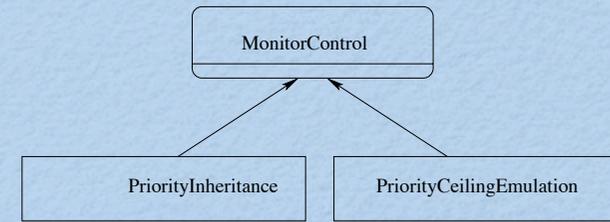


- **Priorité Inheritance Protocole (PIP)**

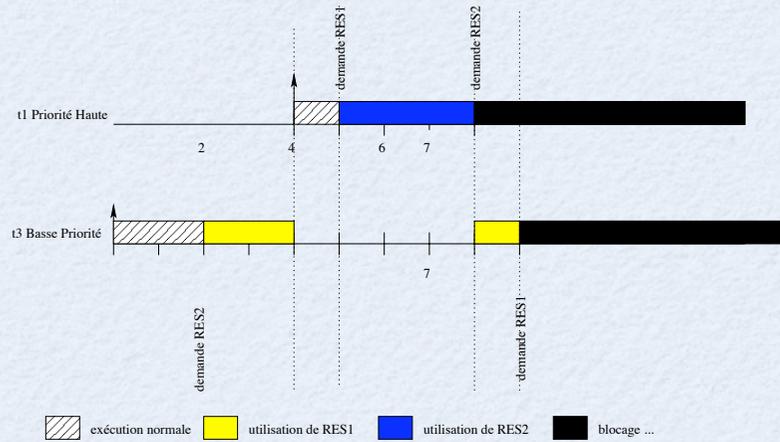


- **moniteur standard Java (non temps réel) (a venir)**

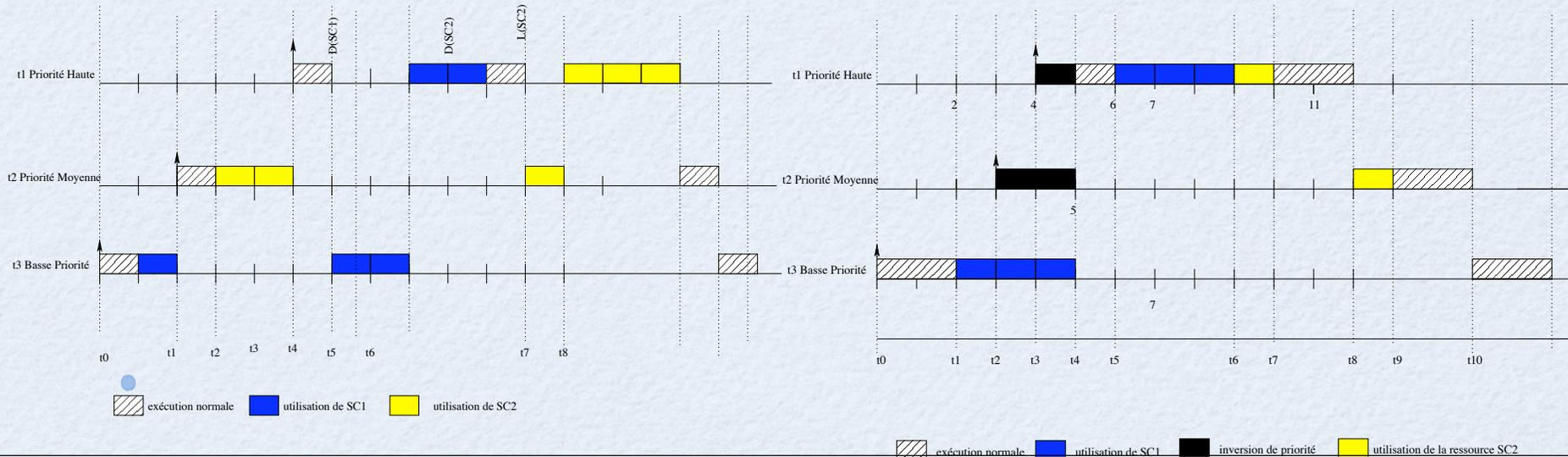
MONITEURS DE SYNCHRONISATIONS



- et **Priorité Ceiling Emulation (PCE)**. pas d'interblocage /



- **pas de blocages en chaîne**



ANALYSE DE FAISABILITÉ

- intégration de **méthodes** correctes **d'analyse d'ordonnançabilité** respectant de l'état de l'art (disponibles).
- **prochaine version:**
 - Prise en compte de l'analyse des tâches à échéance arbitraire (busy period)
 - Analyse des **tâches en graphe**.
 - Proposition de 2 modes d'analyse pour les graphes (lié au mode d'ordonnancement)
 - respect de précédences
 - répartition des échéances

ACTUALITÉ LEJOS RT



- Développement et **intégration d'algorithmes issus de travaux de recherche** :
 - tolérance aux fautes (Calcul de Slack DASS/MASS), gestion de l'énergie (en cours), graphes de tâches (en cours).
- **Plateforme de mesure** (évaluation du coût des algorithmes).
- Utilisation en 2010/2011:
 - **DUT Info 2** (cours programmation concurrente & introduction au temps réel)
 - **Master 1** (cours Temps réel).
 - **Master 2** Projets de recherche (graphes de tâches / Tolérance aux fautes (algorithmes statiques: Allowance en fréquence et en durée)
 - **ESIEE**: Mesure d'énergie
 - **Stage 2A** ENSTA (Tolérance aux fautes)