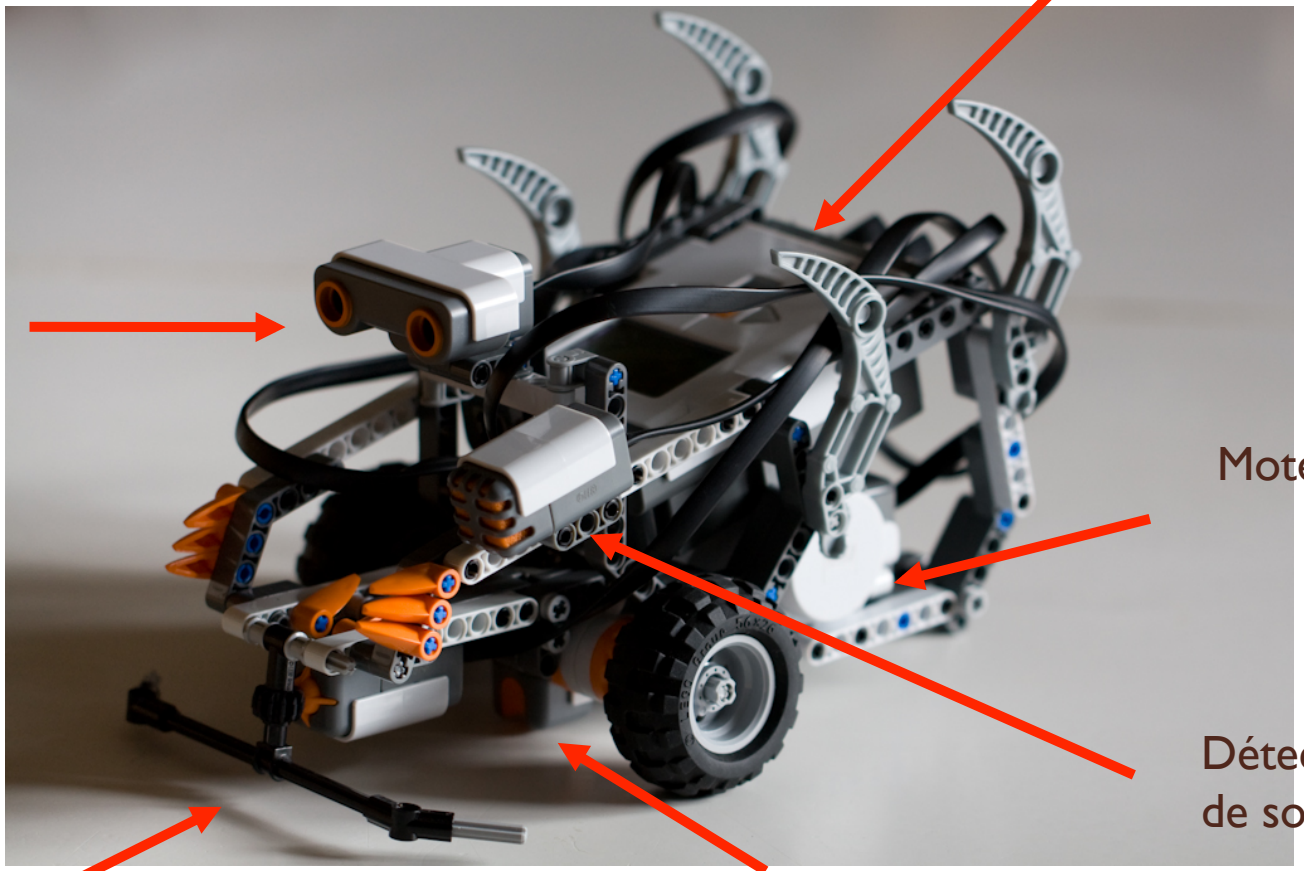
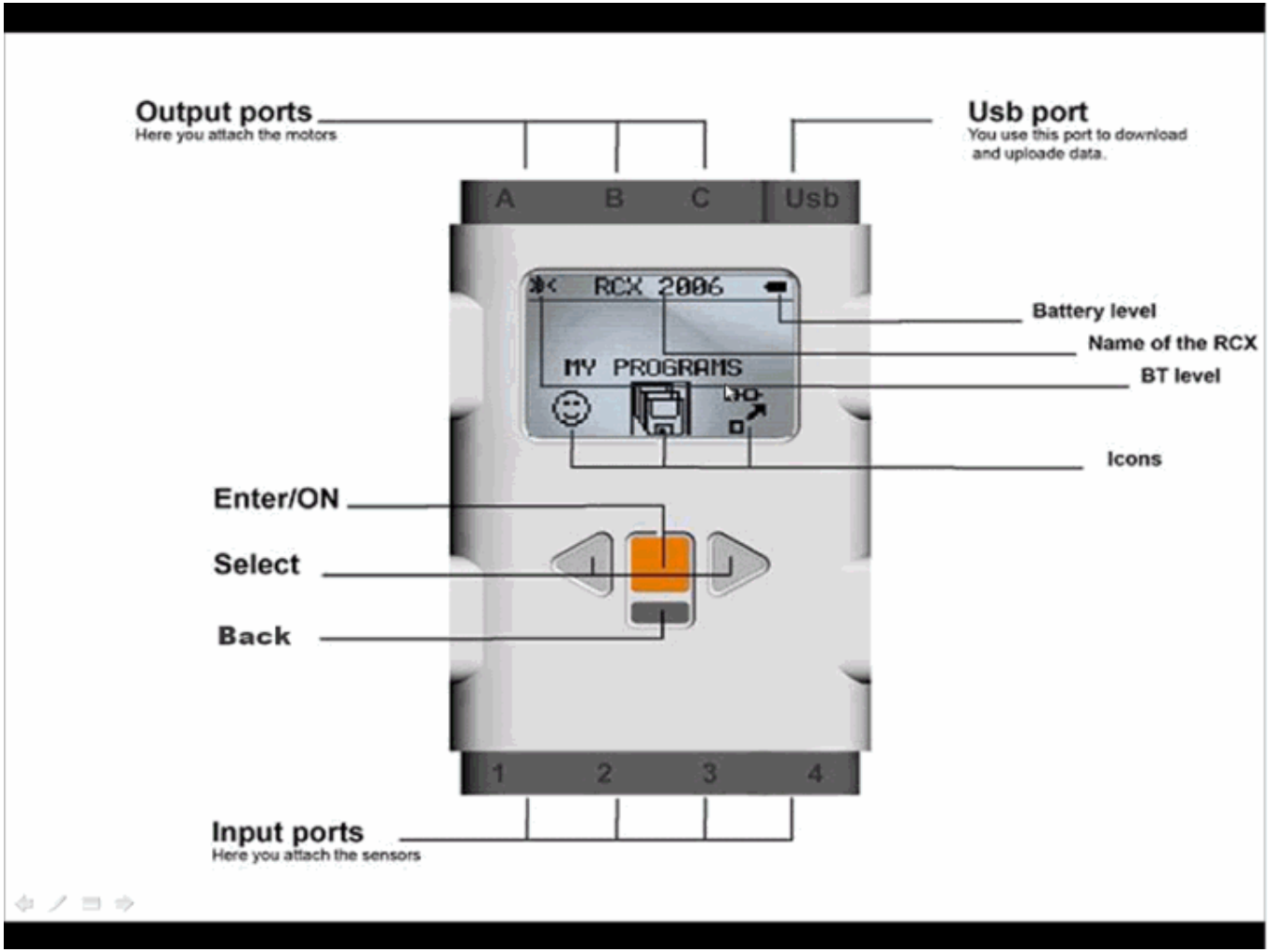
Travaux Pratiques de l'option STR

INTRODUCTION AUX NXT ET À OSEK/VDX

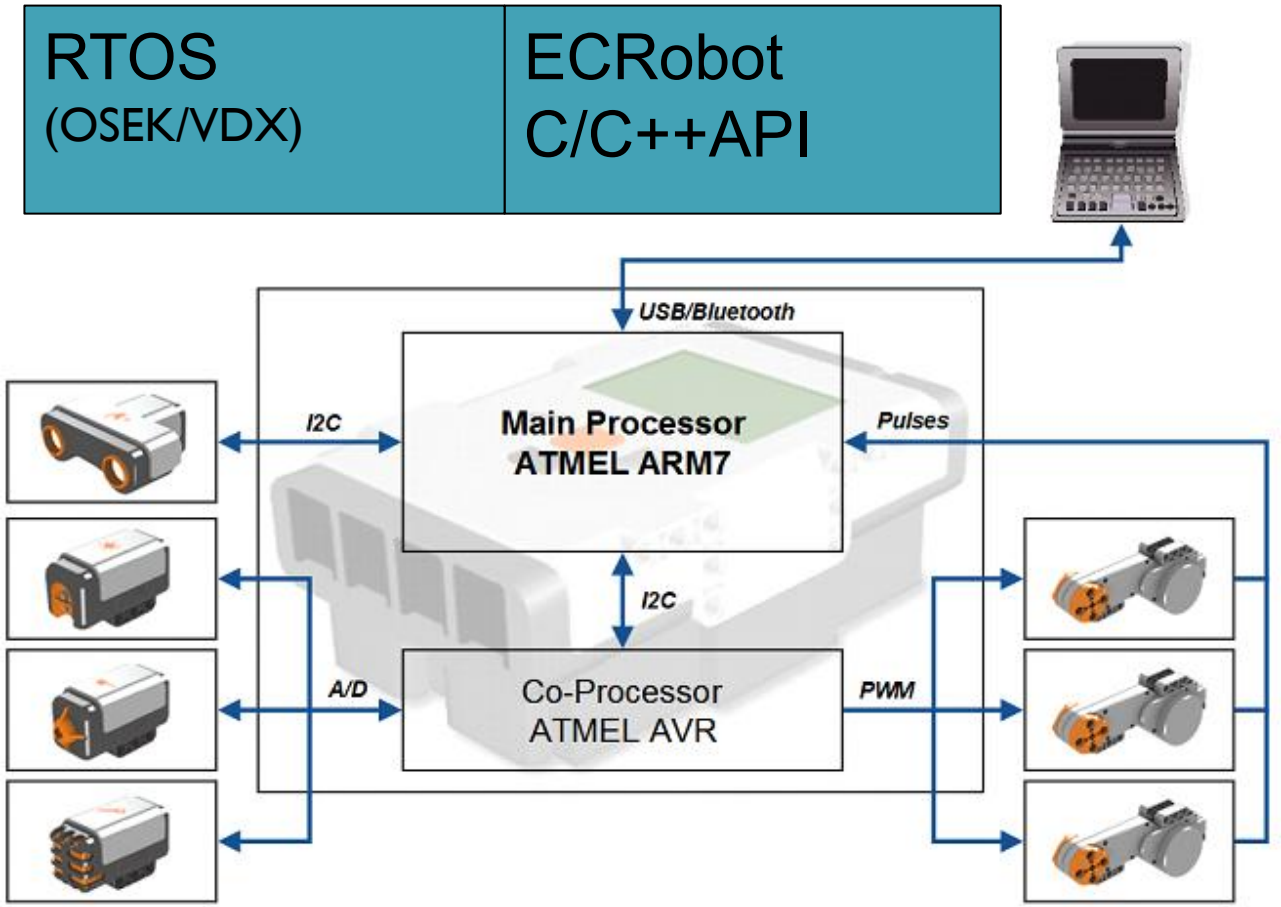
Les robots Mindstorm/NXT



La brique


Output ports
Here you attach the motors
Usb port
You use this port to download and uploade data.
A
B
C
Usb
RCX 2006
MY PROGRAMS
Battery level
Name of the RCX
BT level
Icons
Enter/ON
Select
Back
1
2
3
4
Input ports
Here you attach the sensors

Environnement logiciel



La bibliothèque ecrobot

- Gestion des moteurs
  - Vitesse, angle
- Capteur de lumière
  - Lecture valeur (0..1023), grande valeur = noir
- Touch sensor
  - Lecture valeur (on/off)
- Sound sensor
  - Lecture valeur (0..1023), grande valeur = fort
- Numéros de port en paramètre !

La bibliothèque ecrobot

- Sonar
  - Lecture distance de l'obstacle en cm

- Son
  - Joue des sons (notes individuelles, wav)

- Etat de la brique
  - Boutons (enter/run), état de la batterie, heure système
- Display
  - Différents formats, très "rustique"
  - Attention : appeler display_update !

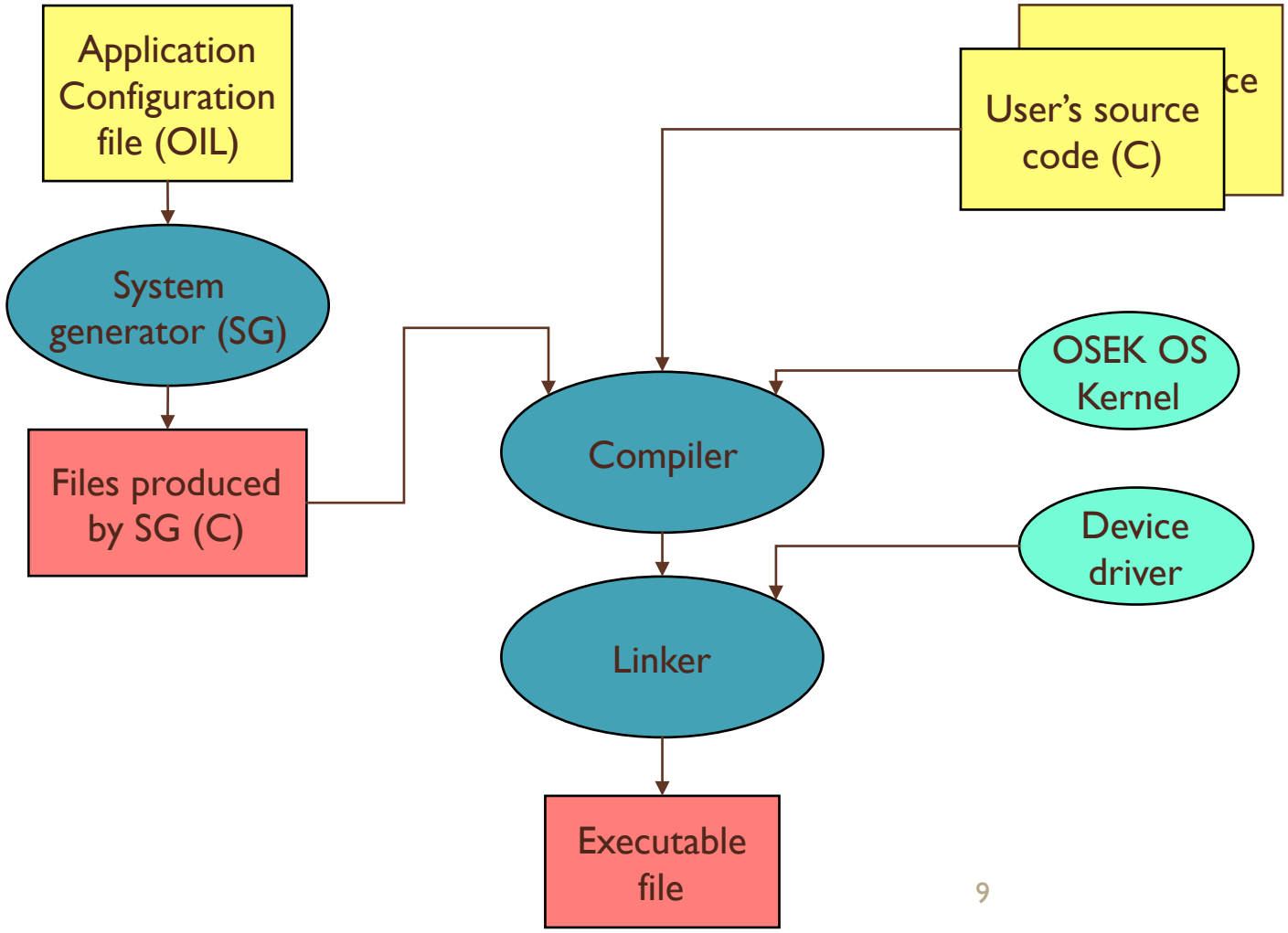
OSEK/VDX : introduction

- Créé dans les années 90 par un consortium de sociétés automobiles et d'universitaires
- Objectif : standard d'architecture ouverte pour l'électronique embarquée dans les véhicules
- Trois parties dans l'architecture
  - Système d'exploitation
  - Gestion du réseau
  - Communication (échanges de données entre U.C.)
- Standard d'interface, diverses implantations
- Dans les TP : Trampoline (open-source, Irccyn)

OSEK/VDX : un noyau statique

- Noyau statique
  - Objets du système définis à la compilation (tâches, messages, sémaphores, etc …)
  - Langage de définition des services : OIL (OSEK Implementation Language)
  - Faible consommation mémoire
- Classes de conformité: différentes « versions » du noyau par rapport aux appels système disponibles

OIL (OSEK Implementation Language)



Services OSEK/VDX

- Gestion de tâches
- Interface de mise au point (Hooks)
- Synchronisation
- Gestion du temps
- Interruptions
- Gestion mémoire : inexistante

Gestion de tâches

- Tâches basiques
  - N'ont pas accès aux services de synchronisation
- Tâches étendues
  - Ont accès aux services de synchronisation
  - Un état supplémentaire : waiting
- Attributs (statiques, liste non exhaustive)
  - PRIORITY (grande valeur = grande priorité)
  - SCHEDULE (NON ou FULL) : préemptif ou non
  - ACTIVATION : nombre max. d'exéc. simultanées
  - AUTOSTART : défini si activé au démarrage de l'OS
  - STACKSIZE : taille de pile

Gestion de tâches

- Ordonnancement non préemptif
  - Pour tâches non préemptibles (SCHEDULE = NON)
  - Conservation du processeur tant qu'une de ces tâches s'exécute
  - Passage en mode non préemptif possible en-ligne
- Ordonnancement préemptif
  - Pour tâches préemptibles (SCHEDULE = FULL)
  - En-ligne, préemptif
  - Par priorités (valeur la plus haute = tâche plus prioritaire)
  - Priorités statiques

Déclaration de tâche (OIL+C)

Fichier OIL (attributs de la tâche)

```
TASK nomtache {
  PRIORITY = entier;
  AUTOSTART = TRUE ou FALSE;
  ACTIVATION = entier;
  SCHEDULE = FULL ou NON;
  STACKSIZE = 4096;
  … (autres attributs non explicités ici) …
};
```

Fichier C (corps de la tâche)

```
TASK(nomtache) {
  /* Code C décrivant le comportement de la tâche */
}
```

Gestion de tâches : API

- ActivateTask : activation d'une tâche
- TerminateTask : terminaison d'une tâche (obligatoire)
- ChainTask : terminaison et activation combinée
- Schedule : appel explicite à l'ordonnançeur
- GetTaskID: retourne l'identificateur de tâche
- GetTaskState : retourne l'état d'une tâche (RUNNING, WAITING, READY, SUSPENDED, INVALID_TASK)
- Pas de création dynamique de tâche !

Fonctions de mise au point (Hooks)

- Hooks : définition
  - Fonctions écrites par l'utilisateur
  - Appelées par le système d'exploitation à des points clé de son exécution. Interface standardisée
- Propriétés
  - Plus haute priorité que les tâches
  - Non interruptibles par interruptions de catégorie 2 (voir plus loin)
  - Appels système disponibles restreints

Fonctions de mise au point (Hooks)

- Points d'appels des hooks
  - Démarrage du système (fin du démarrage du système, avant exécution de l'ordonnançeur, qui lance les tâches) : StartupHook
  - Arrêt du système : ShutdownHook
  - Erreur lors d'un appel système : ErrorHook
  - Changements de contexte : PreTaskHook (appelée avant de donner le processeur à une tâche) et PostTaskHook (appelée avant d'oter le processeur à une tâche)

Fonctions de mise au point (Hooks)

- Fichier OIL (déclaration de l'existence des hooks)

```
OS config {
    ERRORHOOK = TRUE ou FALSE;
    PRETASKHOOK = TRUE ou FALSE;
    POSTTASKHOOK = TRUE ou FALSE;
    STARTUPHOOK = TRUE ou FALSE;
    SHUTDOWNHOOK = TRUE ou FALSE;
    };
```

- Fichier C (code des hooks présents)

```
void PreTaskHook(void) {
    /* Code C du hook */
}
```

Synchronisation

- Evénements
  - Attente, Signalement, Effacement
  - API: SetEvent, WaitEvent, ClearEvent
- Ressources partagées
  - Equivalent à un mutex
  - API: GetResource() ReleaseResource()
  - Gère les inversions de priorité (temps de blocage borné, pas d'interblocage)
- Objets de synchronisation déclarés statiquement dans le OIL
- Pas de sémaphores à compteur !

Gestion du temps

- Consultation du temps système
- Compteurs et alarmes
  - Appel système unique pour déclenchement d'un nombre infini d'alarmes périodiques
  - Traitement d'alarme : activation de tâche, signalement d'événement, appel d'un callback
  - API : voir cours

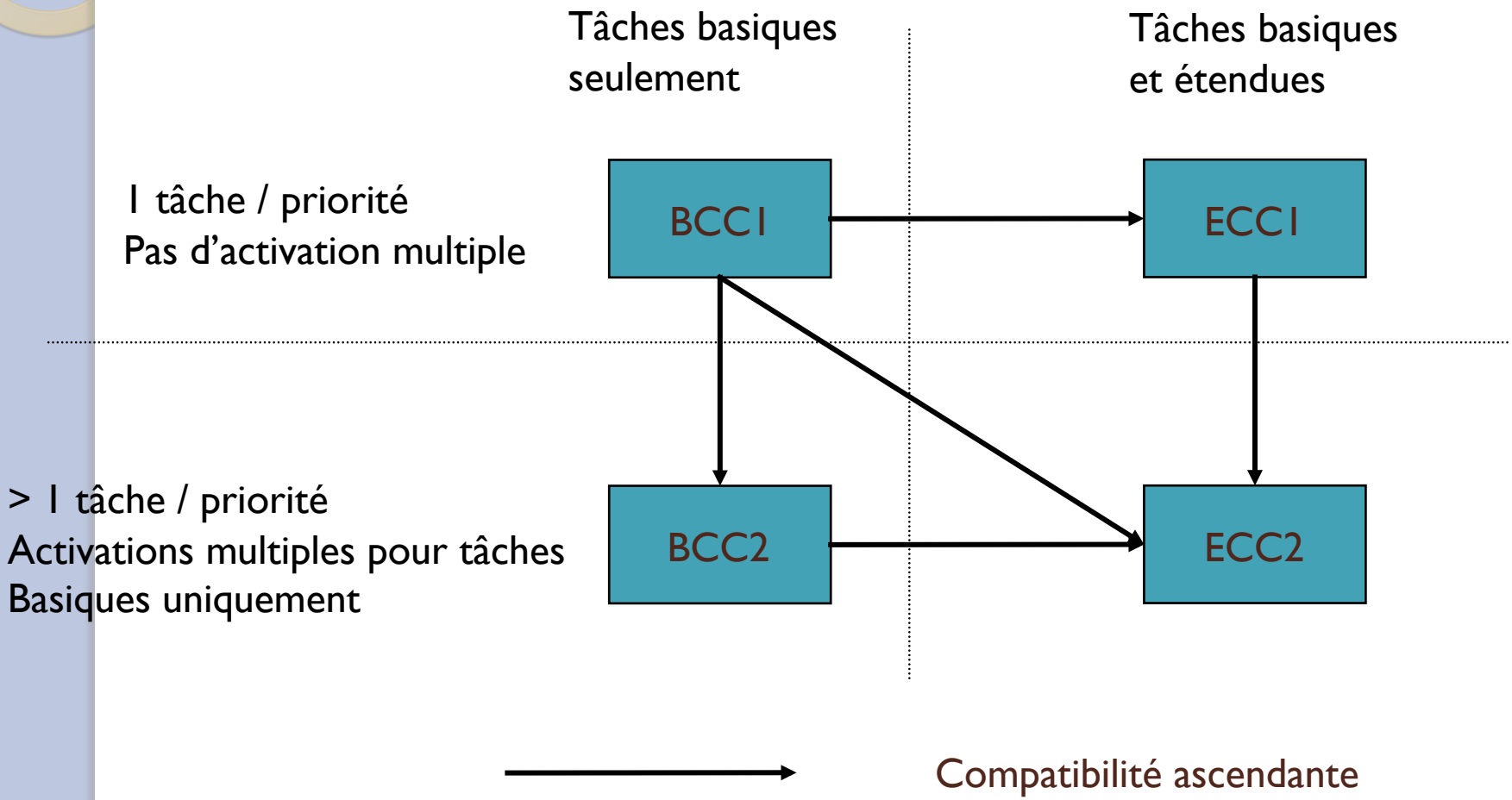
Interruptions

- Routines d'interruptions de niveau 1 (ISR1)
  - Interruptions matérielles (pas de OIL)
  - Appels système interdits
  - Pas de changement de contexte en fin d'ISR, rapide
- Routines d'interruptions de niveau 2
  - Autorisées à faire des appels système, ont un contexte
  - Plus prioritaires que les tâches
  - Plus lentes que ISR1

Classes de conformité

- Motivations
  - Adaptation aux caractéristiques des applications
  - Adaptation aux capacités des architectures (processeur, mémoire)
- Définition
  - Implantation d'un sous-ensemble clairement identifié des fonctionnalités OSEK
- Paramètres des classes de conformité
  - Nombre d'activations simultanées des tâches
  - Types de tâches (basiques/étendues)
  - Nombre de tâches par priorités

Classes de conformité



Classes de conformité

Besoins min. pour être conforme à une classe

	BCC1	BCC2	ECC1	ECC2
Exécutions multiples	Non	Oui	BT: non ET: oui	BT: oui ET: oui
Plus d'une tâche par priorité	Non	Oui	Non	oui
Nombre d'événements par tâche	---		8	
Nombre de priorités	8		16	
Ressources	RES_SCHEDULER		8 (incluant RES_SCHEDULER)	
Alarmes	1			

Retour aux TPs STR

- Disponibles en-ligne
  - Sujets de TP
  - Norme OSEK/VDX (OS) et résumé de l'API
  - Norme OSEK/VDX (OIL) et fichier exemple
  - API Ecrobot
- Non disponibles en continu
  - Les robots
  - (Sauf TP 1) : venez en TP avec du code écrit !