

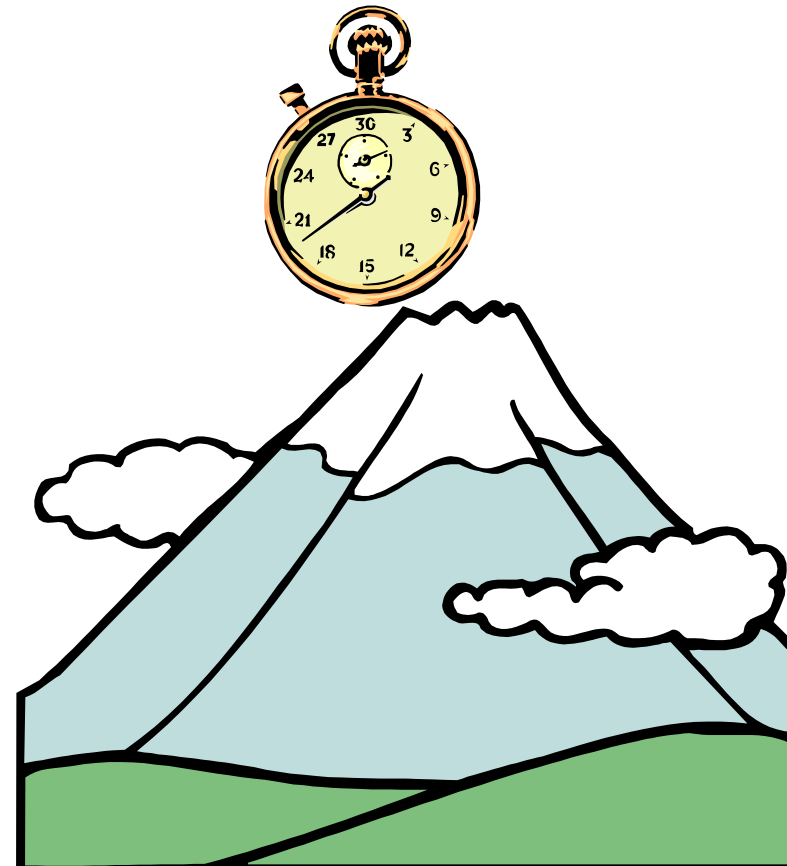
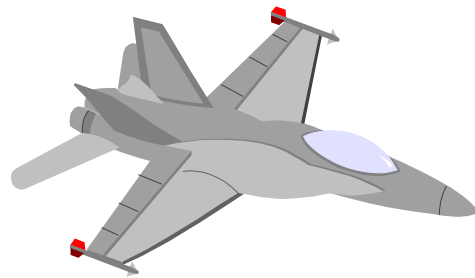
La méthode DARTS et la programmation multitâche en LabVIEW

Khanh Hieu NGO, Emmanuel GROLLEAU
LISI/ENSMA

Plan de la présentation

- Systèmes temps réel
 - Problématique
 - Cycle de vie
- Méthodes de spécification/conception
 - SA/RT
 - DARTS
- Implémentation de systèmes temps réel
 - Comparaison de différents langages de programmation
 - Outils LabVIEW pour le temps réel
- La bibliothèque DARTSVIEW
- Perspectives

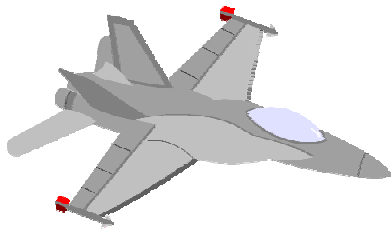
Introduction au temps réel



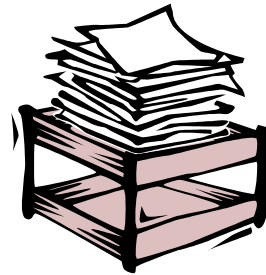
Correction algorithmique

Correction temporelle

Systèmes temps réel



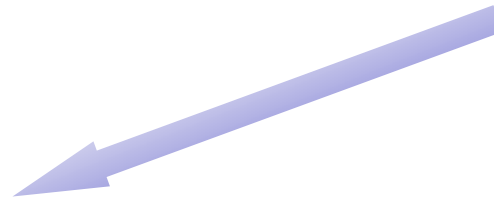
Procédé



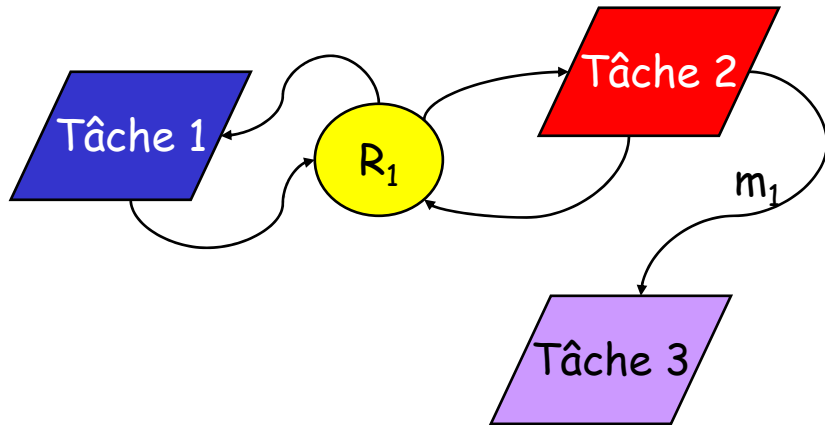
Cahier des charges



Spécification



Contraintes temporelles



Découpage en tâches

Cycle de vie d'un système temps réel

Spécification
(SA-RT, UML...)

Validation temporelle
(WCET, ordonnancement)

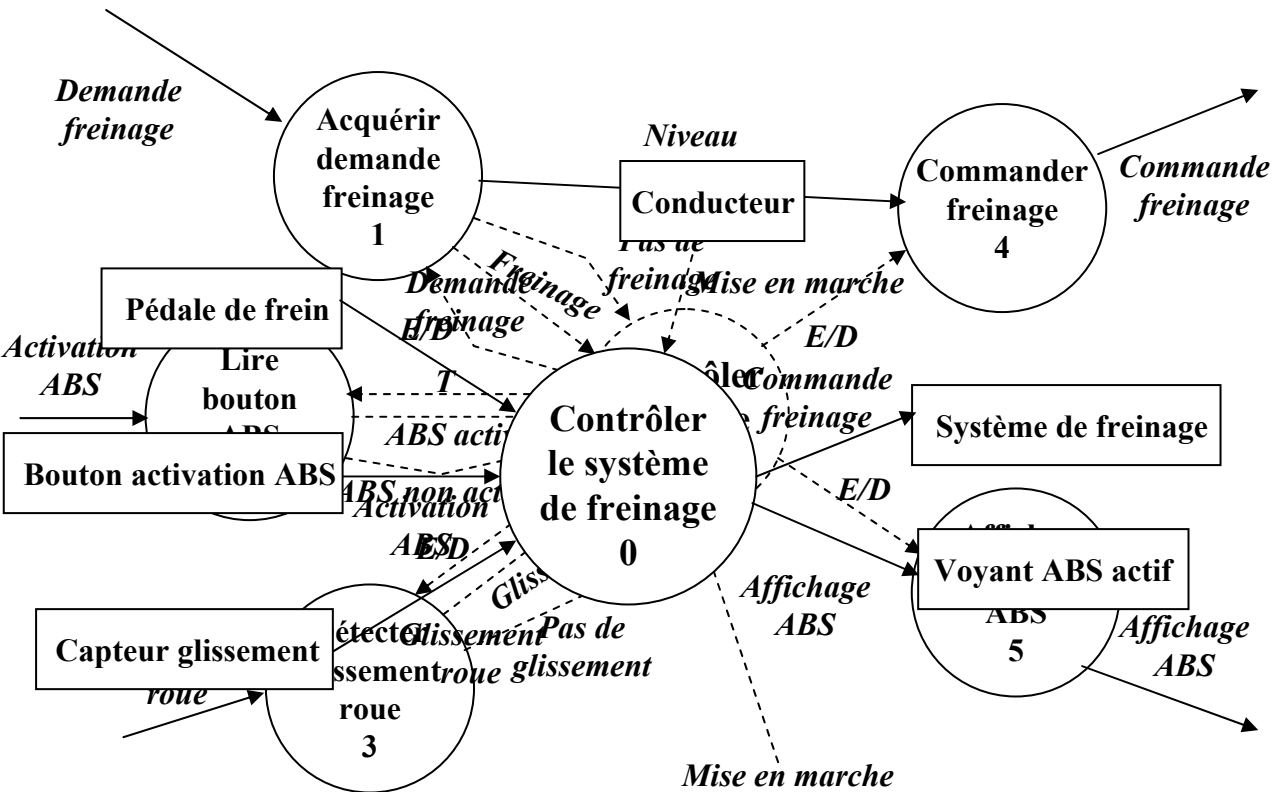
Conception
(DARTS,...)

Tests d'intégration

Implémentation
(Ada, C, LabVIEW,...)

Tests unitaires

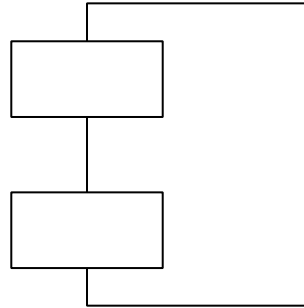
Système exemple: freinage ABS (version académique)



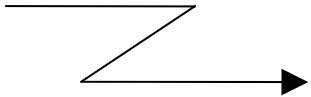
Eléments d'un diagramme DARTS



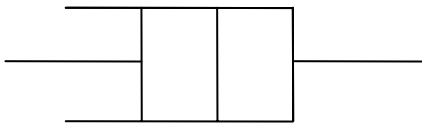
Tâche



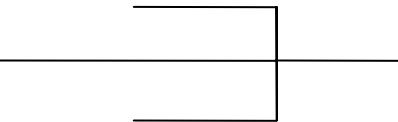
Module de données



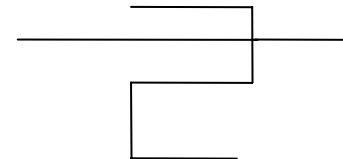
Evénement (*interruption_IT, gestion de temps_HTR...*)



Boîte aux lettres
(*Bal*)

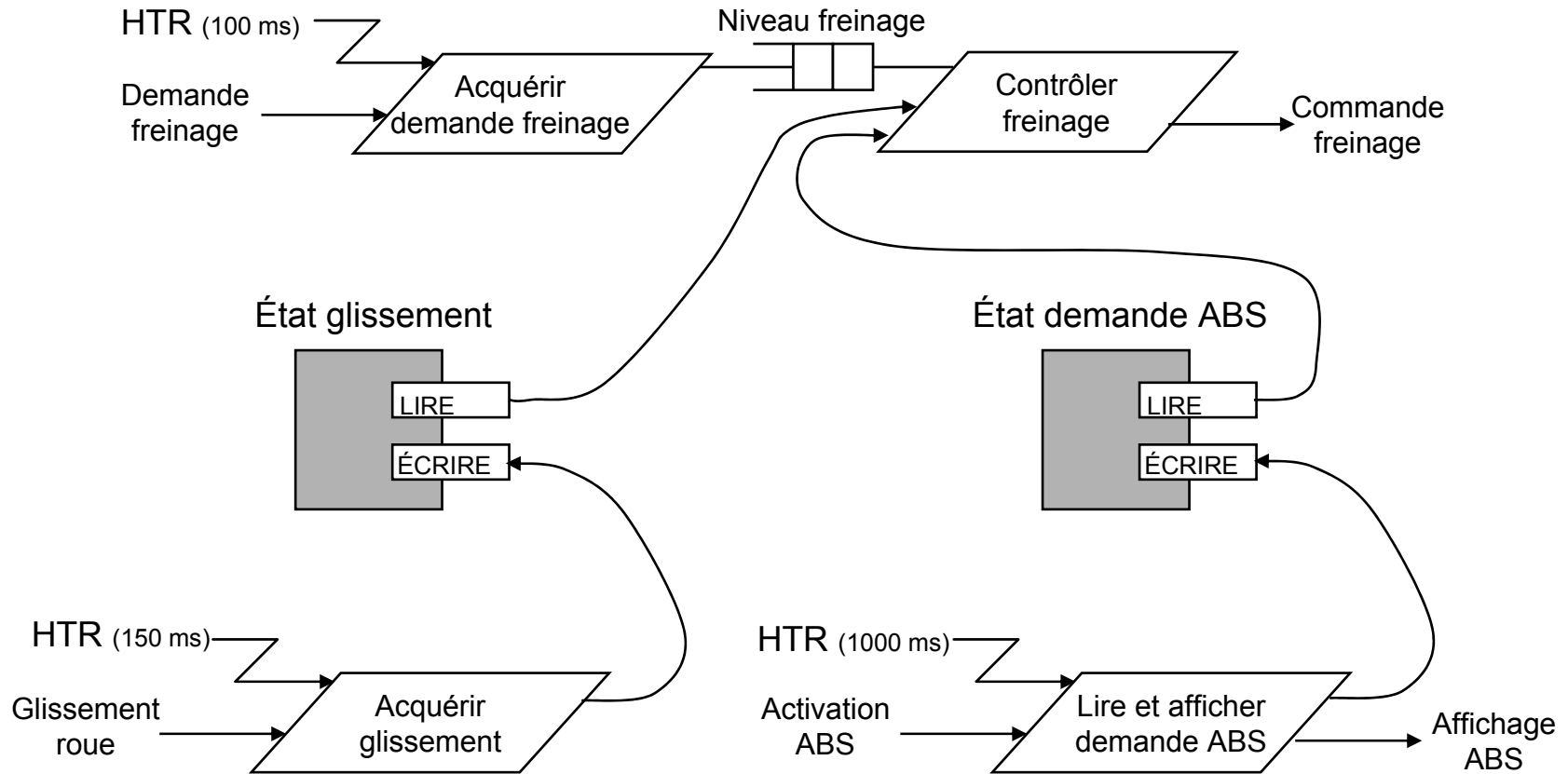


Synchronisation



Rendez-vous

Conception DARTS



LabVIEW: un langage pour le multitâche

DARTS

- ★ Tâche
- ★ Synchronisation
- ★ Communication
- ★ Gestion du temps
- ★ Module de données
- ★ Rendez-vous

C

- ➡ Tâche
- ➡ Sémaphore
- ➡ Queue
- ➡ Delay

LabVIEW

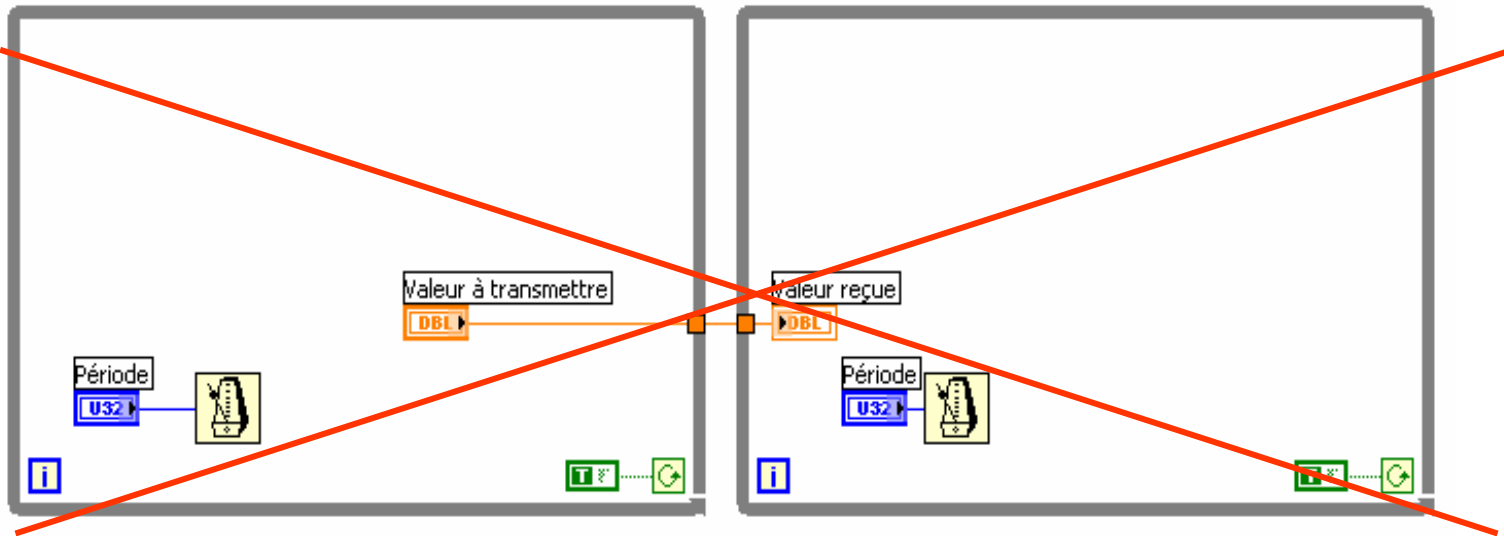
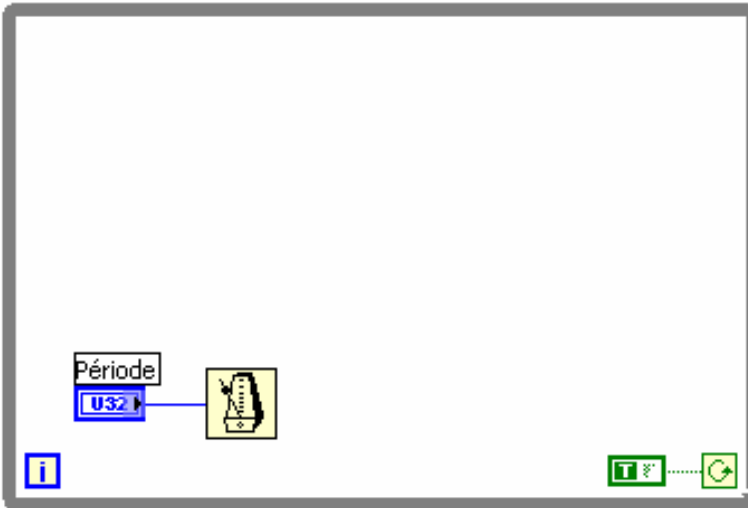
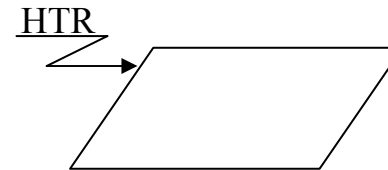
- ➡ Boucle
- ➡ Sémaphore
- ➡ Queue/ RT FIFO
- ➡ Delay, delay until...
- ➡ vi non réentrants
- ➡ rendez-vous

ADA

- ➡ Tâche
- ➡ Objets protégés
- ➡ Delay, delay until
- ➡ Rendez-vous

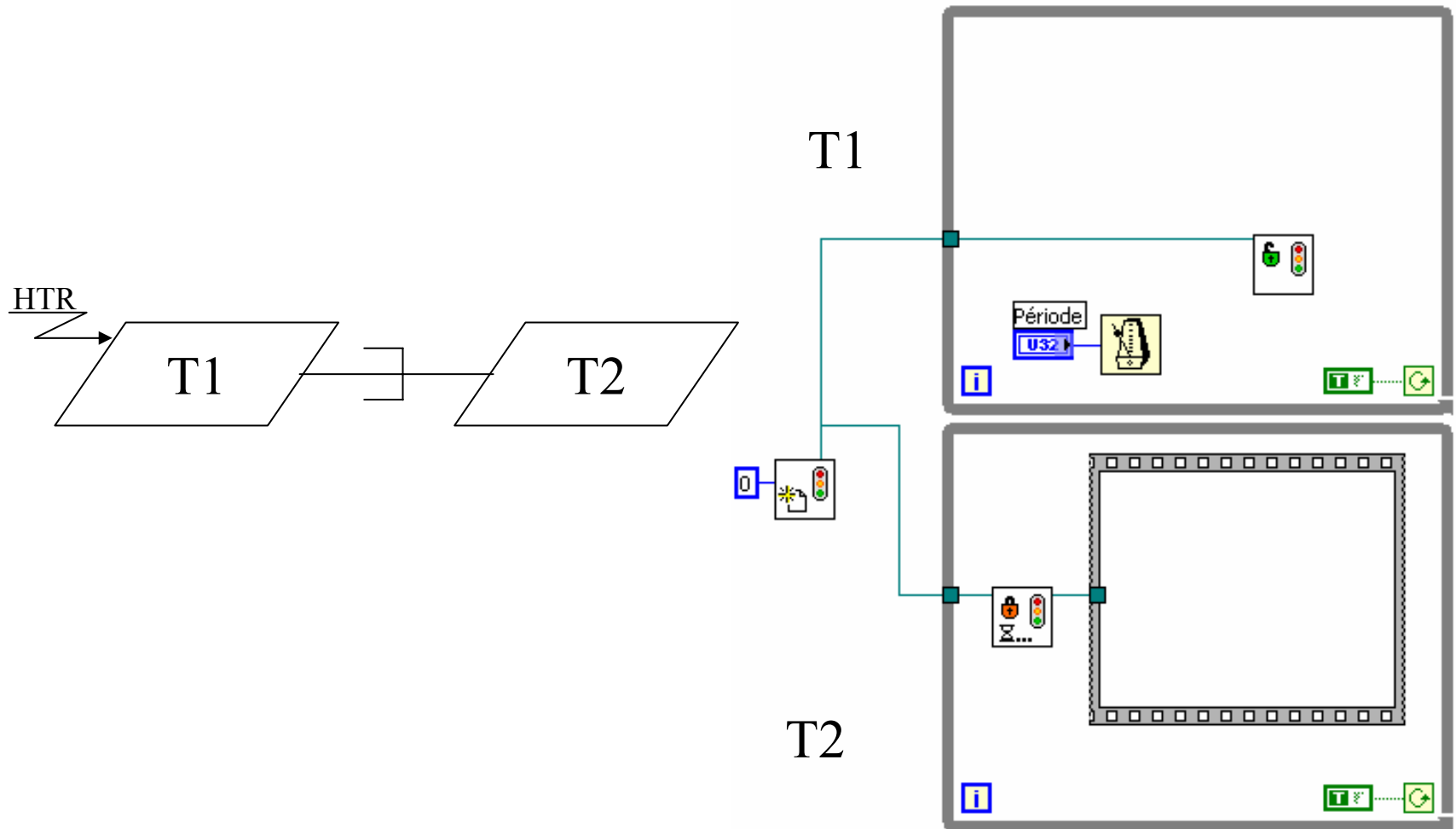
Tâches

► Tâche périodique indépendante



► Tâches périodiques **non** communicantes

Synchronisation



Taxonomie des modes de communication

Boîte aux lettres

Bornée

Non bornée
(non bloquante en écriture)

Avec écrasement
(non bloquante en écriture)

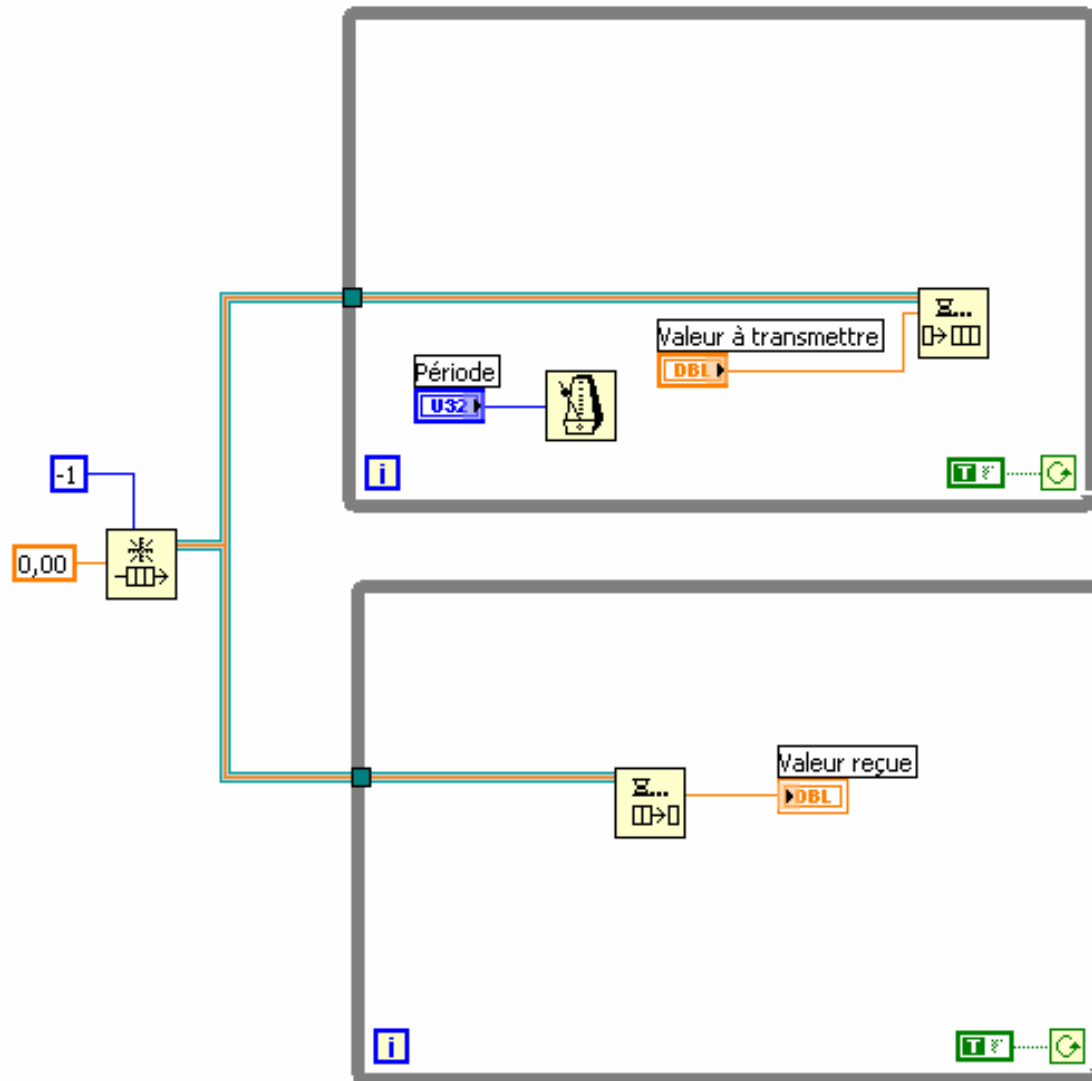
Sans écrasement
(bloquante en écriture)

De taille 1 sans donnée
=>synchronisation


- ▶ Si la boîte aux lettres est non bloquante en lecture (i.e. si aucune nouvelle donnée, l'ancienne donnée est lue), c'est un module de données
- ▶ Rendez-vous (mais avec paramètres) -> peu utilisé car difficile à valider -> on recommandera l'utilisation de 2 boîtes aux lettres

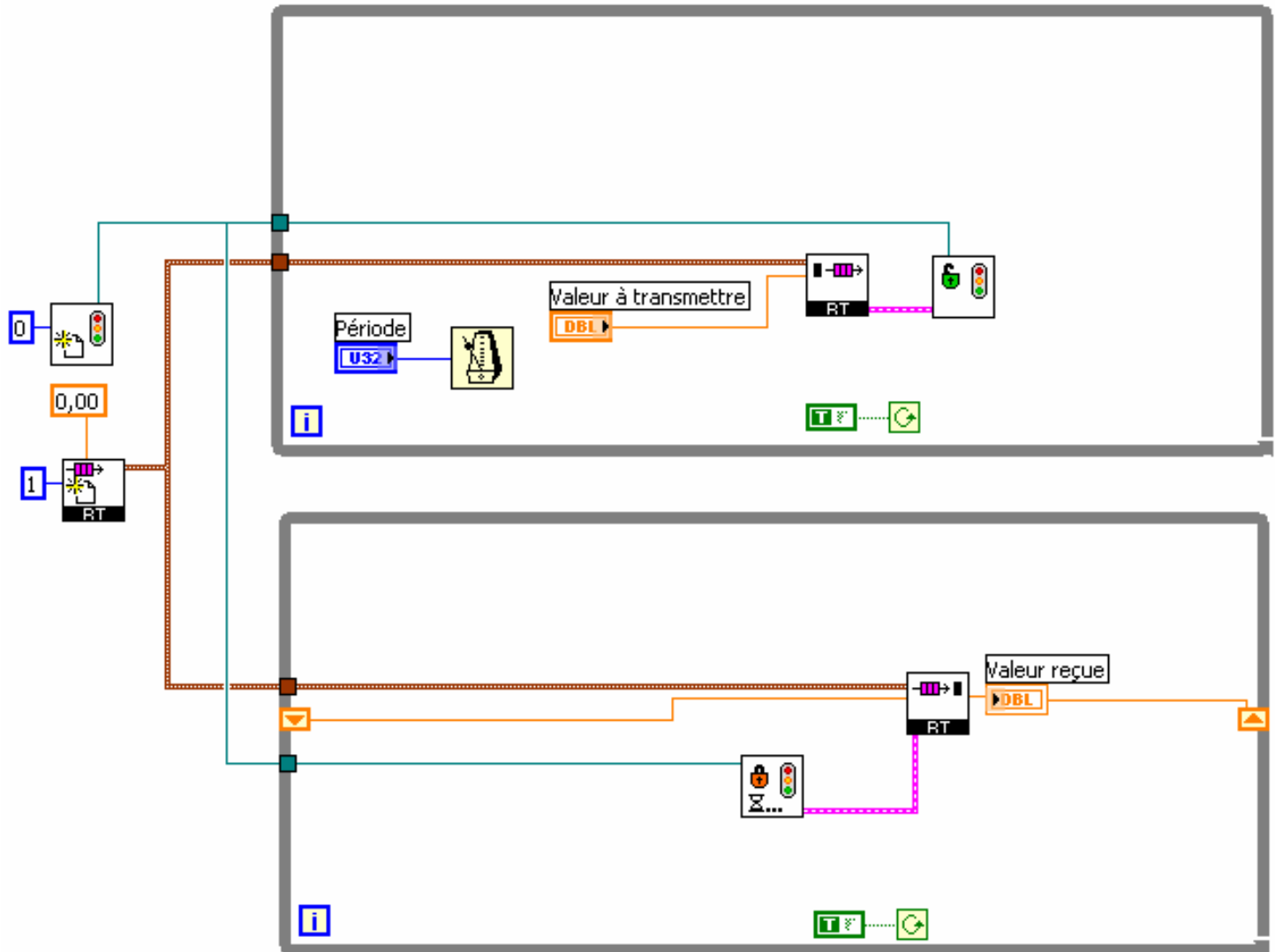
Communication par boîtes aux lettres: sans écrasement

- Non bornée (sans écrasement) ou bien bornée et bloquante en écriture → queue

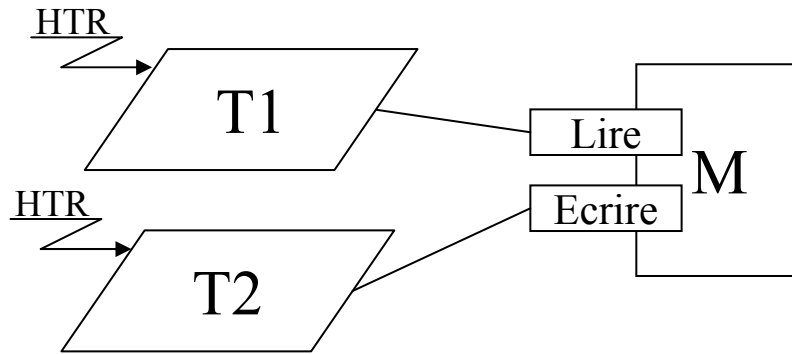


Communication par boîtes aux lettres : avec écrasement

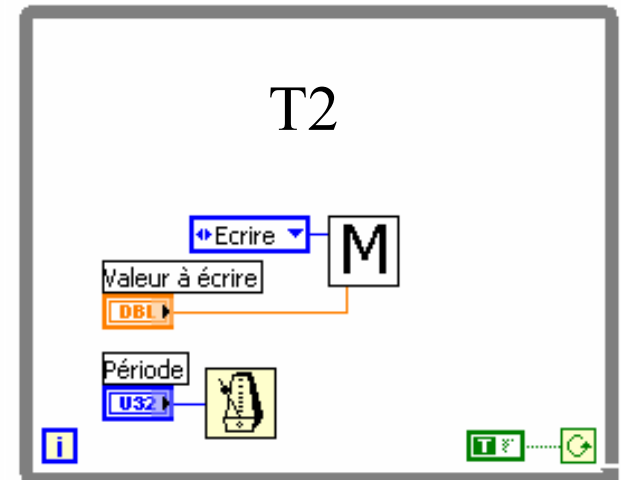
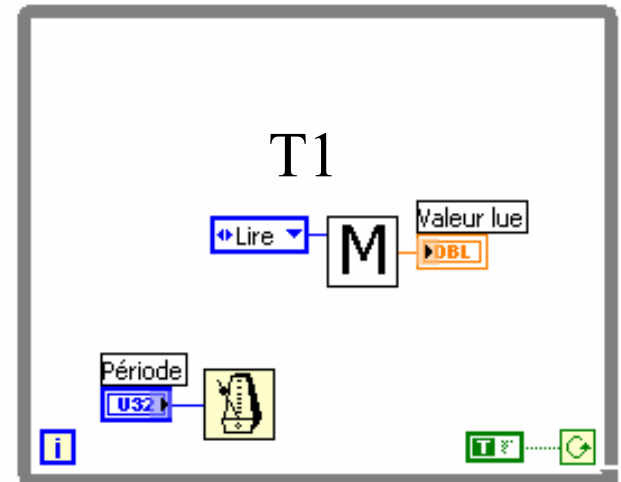
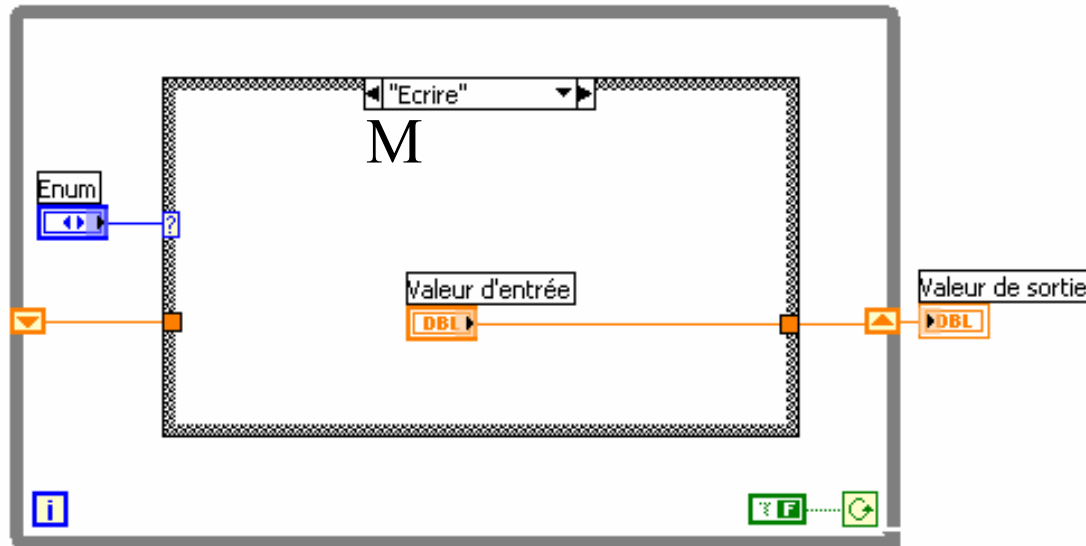
- Bornée avec écrasement → RT FIFO
-  La RT FIFO est non bloquante en lecture



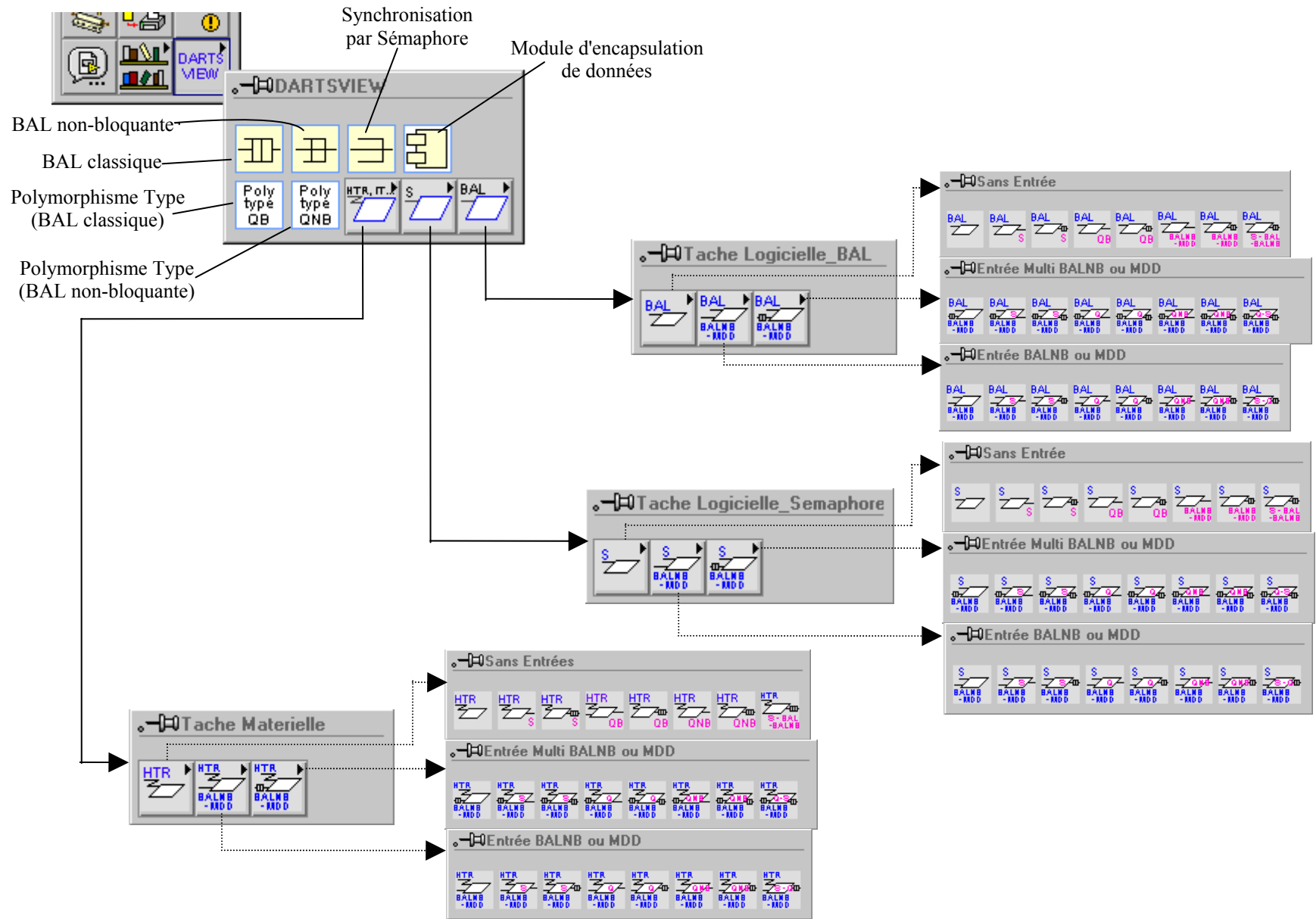
Communication par module de données



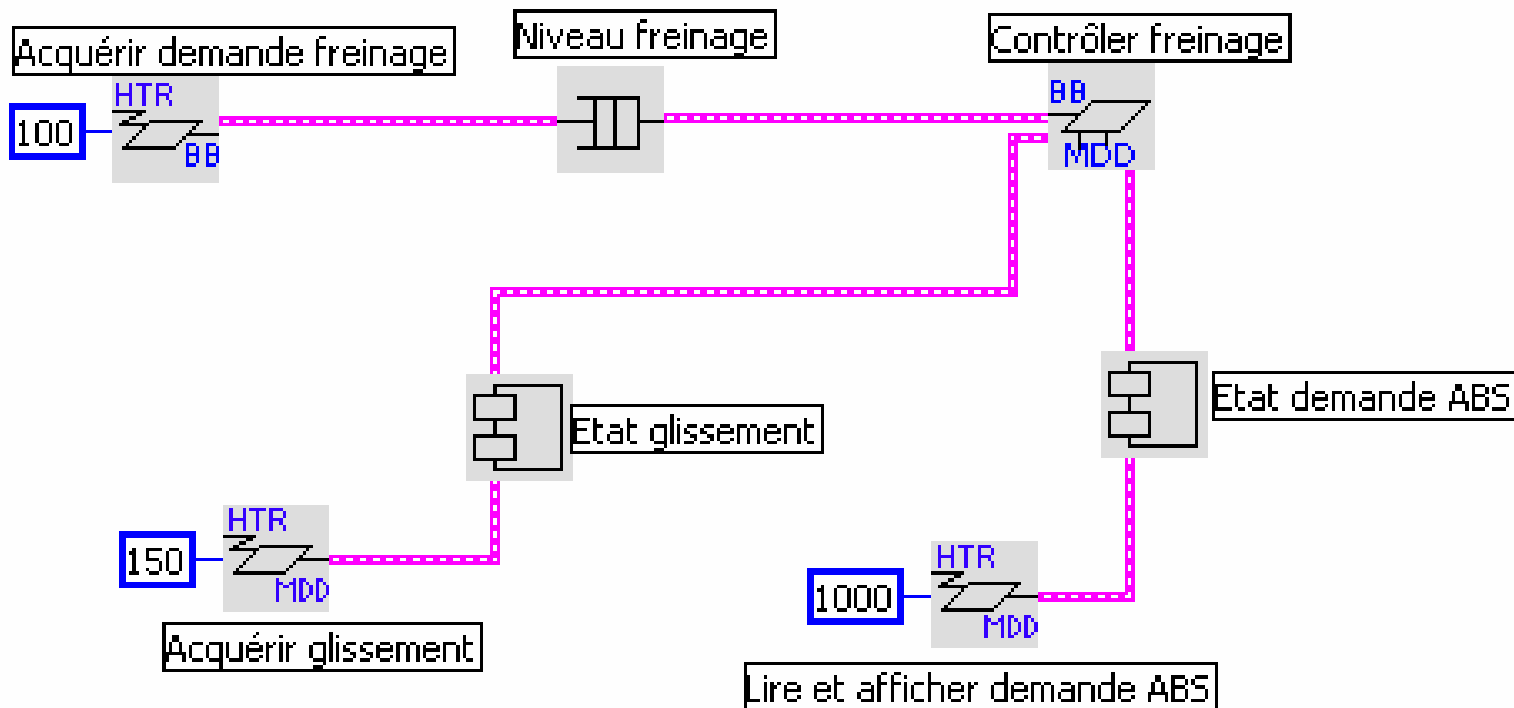
- ▶ VI non réentrant et registre à décalage



La bibliothèque DARTSVIEW



Utilisation de DARTSVIEW



Conclusion & Perspectives

- ▶ DARTSVIEW permet simplement d'implémenter à partir de la conception
 - ▶ Utilisé en TP cette année
- ▶ Génération de code presque opérationnelle (C, Ada, niveau méta)
- ▶ Même si le langage cible est Ada ou C, facilite le cycle de vie de la Conception aux tests d'intégration

