



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèmes Embarqués 1 & 2

tp.09 - Interruptions matérielles

Classes T-2/I-2 // 2016-2017

Daniel Gachet | HEIA-FR/TIC
tp.09 | 20.03.2017



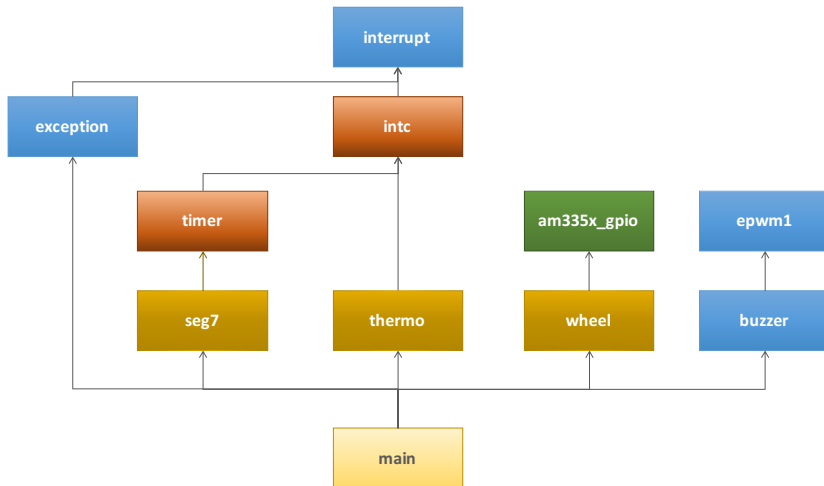
- A la fin du laboratoire, les étudiant-e-s seront capables de
 - ▶ Décrire le traitement des interruptions matérielles des μ P ARM
 - ▶ Développer les pilotes permettant de gérer les interruptions matérielles du μ P TI AM335x
 - ▶ Migrer une application basée sur la scrutation pour la rendre plus réactive en mode interruptif
 - ▶ Etudier le datasheet du μ P ARM Cortex-A8 et du μ P TI AM335x
- Durée
 - ▶ 1 séance de laboratoire (4 heures)
- Rapport
 - ▶ Rapport de laboratoire avec le code source



- Adapter l'application Thermo-Buzzer d'un précédent TP basée sur de la scrutation et la transformer en mode interruptif
- Le encodeur rotatif et son bouton poussoir serviront d'interface
 - ▶ affichage de la température actuelle sur le display 7-segments
 - ▶ réglage de la valeur supérieure de supervision de température
 - ▶ réglage de la valeur inférieure de supervision de température
- Pour choisir le menu et la fonction
 - ▶ une première pression pour entrer dans le menu (affichage du menu sur le display 7-segments)
 - ▶ le bouton rotatif pour choisir la fonction
 - ▶ une deuxième pression pour activer la fonction choisie



Structure générale de l'application





- Développer les modules suivants
 - ▶ "main" module principal de l'application
 - ▶ "intc" module pour la gestion des interruptions matérielles
 - ▶ "timer" module pour le pilotage des timers 2 à 7 du μ P TI AM335x
- Adapter les modules suivants pour un traitement interruptif
 - ▶ "wheel" gestion de l'encodeur rotatif
 - ▶ "seg7" gestion du display 7-segments
 - ▶ "thermo" gestion du thermomètre i2c



- Pour télécharger le squelette du projet du dépôt centralisé

```
$ cd ~/workspace/se12/tp  
$ git pull upstream master
```

- Pour mettre à jour les paths des includes dans eclipse

- ▶ ouvrir **Properties** pour votre projet
- ▶ aller **C/C++ General** → **Paths and Symbols**
- ▶ ouvrir **Includes** → **GNU C**
- ▶ ajouter `/home/lmi/workspace/se12/bbb/source`

- Le code et le rapport seront rendus au travers du dépôt Git centralisé

- ▶ *sources : .../tp/tp.09*
- ▶ *rapport : .../tp/tp.09/doc/report.pdf*

- Délai

- ▶ Le journal et le code doivent être rendus au plus tard 6 jours après le TP à 23h59