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

# Systèmes Embarqués 1 & 2

## tp.03 - Pilotes de périphériques

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

Daniel Gachet | HEIA-FR/TIC

tp.03 | 19.10.2016

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz

# Objectifs

- A la fin du laboratoire, les étudiant-e-s seront capables de
  - Développer (concevoir, coder et tester) un petit programme capable de piloter un des 7-segments de la carte d'extension par l'intermédiaire des portes d'entrée/sortie du µP
  - Etudier un datasheet élémentaire d'un chip et en comprendre son contenu
- Durée
  - 1 séance de laboratoire (4 heures)
- Rapport
  - Journal de laboratoire avec le code source

# Travail à réaliser

- Développez une petite application réalisant un serpentin lumineux sur un des 7-segments de la carte d'extension du Beaglebone Black en allumant et éteignant successivement les LEDs de ses segments lumineux

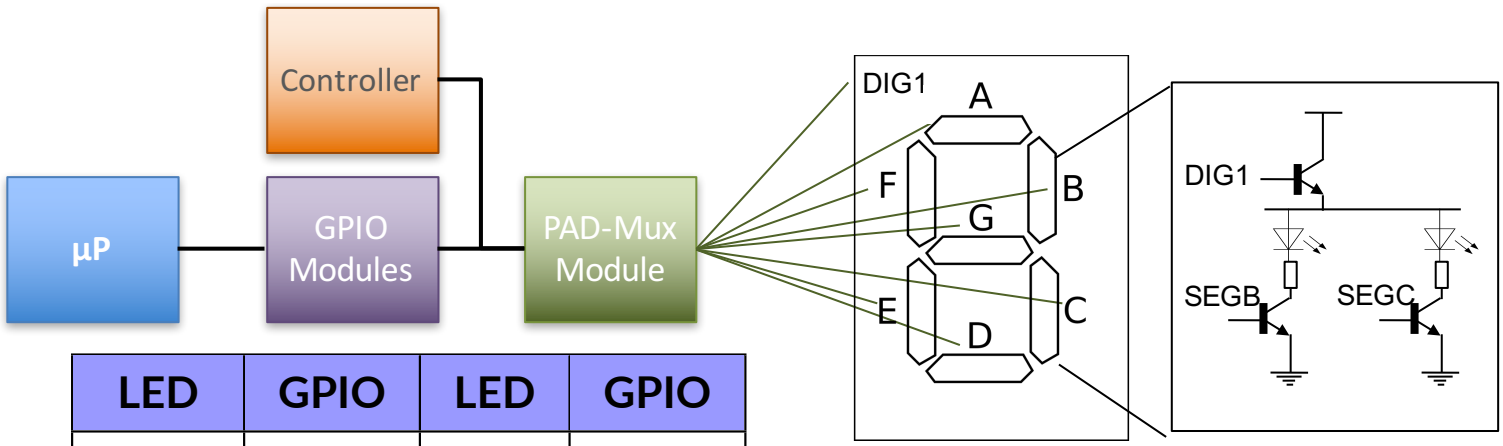
*http ://florelle.over-blog.org.over-blog.com/article-35786916.html*

- Les conditions d'exécution sont les suivantes
  - Le squelette du projet se trouve se trouve sur le dépôt centralisé
  - Pour le télécharger, tapez les commandes suivantes

    ```
    $ cd ~/workspace/se12/tp
    $ git pull upstream master
    ```
  - L'application sera intégrée dans le fichier « main.S »
- Le code et le rapport seront rendus au travers du dépôt Git centralisé
  - *sources : .../tp/tp.03*
  - *rapport : .../tp/tp.03/doc/report.pdf*
- Délai
  - Le journal et le code doivent être rendus au plus tard 6 jours après le TP à minuit

# Infrastructure matérielle

- Pour piloter les LEDs du display 7-segment, le µP dispose de contrôleurs d'entrée/sortie (GPIO), lesquels sont interfacés aux portes physiques d'entrée/sortie par l'intermédiaire de multiplexeurs (PAD-Mux Module)




| LED | GPIO | LED | GPIO |
|---|---|---|---|
| DIG1 | 2_2 | D | 0_22 |
| A | 0_4 | E | 0_23 |
| B | 0_5 | F | 0_26 |
| C | 0_14 | G | 0_27 |

# Contrôleur GPIO

- Les contrôleurs GPIO permettent au µP de contrôler l'état des portes d'entrée/sortie utilisées pour piloter les LEDs

Base addresses :
- GPIO0 : `0x481a'c000`
- GPIO2 : `0x44e0'7000`

| 31 | 30 | … | 1 | 0 | offset |
|---|---|---|---|---|---|
| | … | | | | |
| | oe | | | | 0x134 |
| | …. | | | | |
| | clear | | | | 0x190 |
| | set | | | | 0x194 |
| | … | | | | |

- Le registre *set* permet de mettre la porte à l'état haut (high ⇔ 1)
- Le registre *clear* permet de mettre la porte à l'état bas (low ⇔ 0)
- Le registre *oe* permet au µP de choisir le mode de fonctionnement de la porte d'entrée/sortie
  - bit mis à 1, la porte fonctionne en entrée
  - bit mis à 0, la porte fonctionne en sortie

# Multiplexeur des portes d'entrée/sortie

- Pour connecter au contrôleur GPIO les portes d'entrée/sortie correspondant aux LEDs du Beaglebone, il suffit de configurer le multiplexeur de chaque GPIO avec la valeur 0x4f aux adresses suivantes

gpio_0.4 : 0x44e1'0000 + 0x958
gpio_0.5 : 0x44e1'0000 + 0x95c
gpio_0.14 : 0x44e1'0000 + 0x980
gpio_0.22 : 0x44e1'0000 + 0x820
gpio_0.23 : 0x44e1'0000 + 0x824
gpio_0.26 : 0x44e1'0000 + 0x828
gpio_0.27 : 0x44e1'0000 + 0x82c
gpio_2.2 : 0x44e1'0000 + 0x890

# Initialisation matérielle

- Avant de pouvoir piloter les LEDs, le µP devra impérativement configurer les différents contrôleurs
  - Initialiser les contrôleurs GPIO0 et GPIO2
  - Configurer l'état des portes d'entrée/sortie à l'état souhaité
  - Configurer les portes des GPIO en mode sortie
  - Connecter les portes d'entrée/sortie des multiplexeurs aux contrôleurs GPIO et les mettre en mode sortie
- Le code ci-dessous permet d'initialiser les contrôleurs GPIO0 et GPIO2

```
mov     r0, #0
bl      am335x_gpio_init
mov     r0, #2
bl      am335x_gpio_init
```