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

Systèmes Embarqués 1 & 2

tp.04 - Introduction au C

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

Daniel Gachet | HEIA-FR/TIC

tp.04 | 09.11.2016

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz

Objectifs

- A la fin du laboratoire, les étudiant-e-s seront capables de
  - Faire leurs premiers pas en programmation C
  - Développer (concevoir, coder et tester) un programme en C
  - Utiliser des bibliothèques standard C
  - Concevoir un programme modulaire (plusieurs fichiers) en C
  - Etudier un schéma élémentaire d'un circuit électronique et en comprendre son contenu
  - Concevoir un programme capable de piloter l'encodeur rotatif et l'afficheur 7-segment de la carte d'extension HEIA-FR du Beaglebone par l'intermédiaire des ports d'entrée/sortie du µP
- Durée
  - 1 séance de laboratoire (4 heures)
- Rapport
  - Rapport de laboratoire avec le code source

Travail à réaliser

- Développez une application implémentant un compteur allant de -99 à 99
- Pour réaliser cette fonctionnalité, les interfaces de la carte d'extension du Beaglebone seront utilisées comme suit
  - L'afficheur 7-segments servira à afficher la valeur du compteur
  - L'encodeur rotatif permettra d'incrémenter et de décrémenter la valeur du compteur
  - Une pression sur le bouton poussoir intégré dans l'encodeur mettra le compteur à zéro (0)

Travail à réaliser (II)

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

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

- Le code et le rapport seront rendus au travers du dépôt Git centralisé
  - *sources* : *.../tp/tp.04*
  - *rapport* : *.../tp/tp.04/doc/report.pdf*
- Délai
  - Le journal et le code doivent être rendus au plus tard 6 jours après le TP à minuit

Mise à jour

- Pour mettre à jour la bibliothèque spécialisée du Beaglebone

```
$ cd ~/workspace/se12
$ git pull
$ make -C ~/workspace/se12/bbb/source
```

- Pour permettre l'utilisation de l'outil "merge" de Git

```
$ sudo dnf install y meld
$ git config --global merge.tool meld
```

- Pour choisir votre dépôt personnel sur Gitlab comme dépôt primaire sur votre machine virtuelle

```
$ cd ~/workspace/se12/tp
$ git branch -u origin/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 `~/workspace/se12/bbb/source`

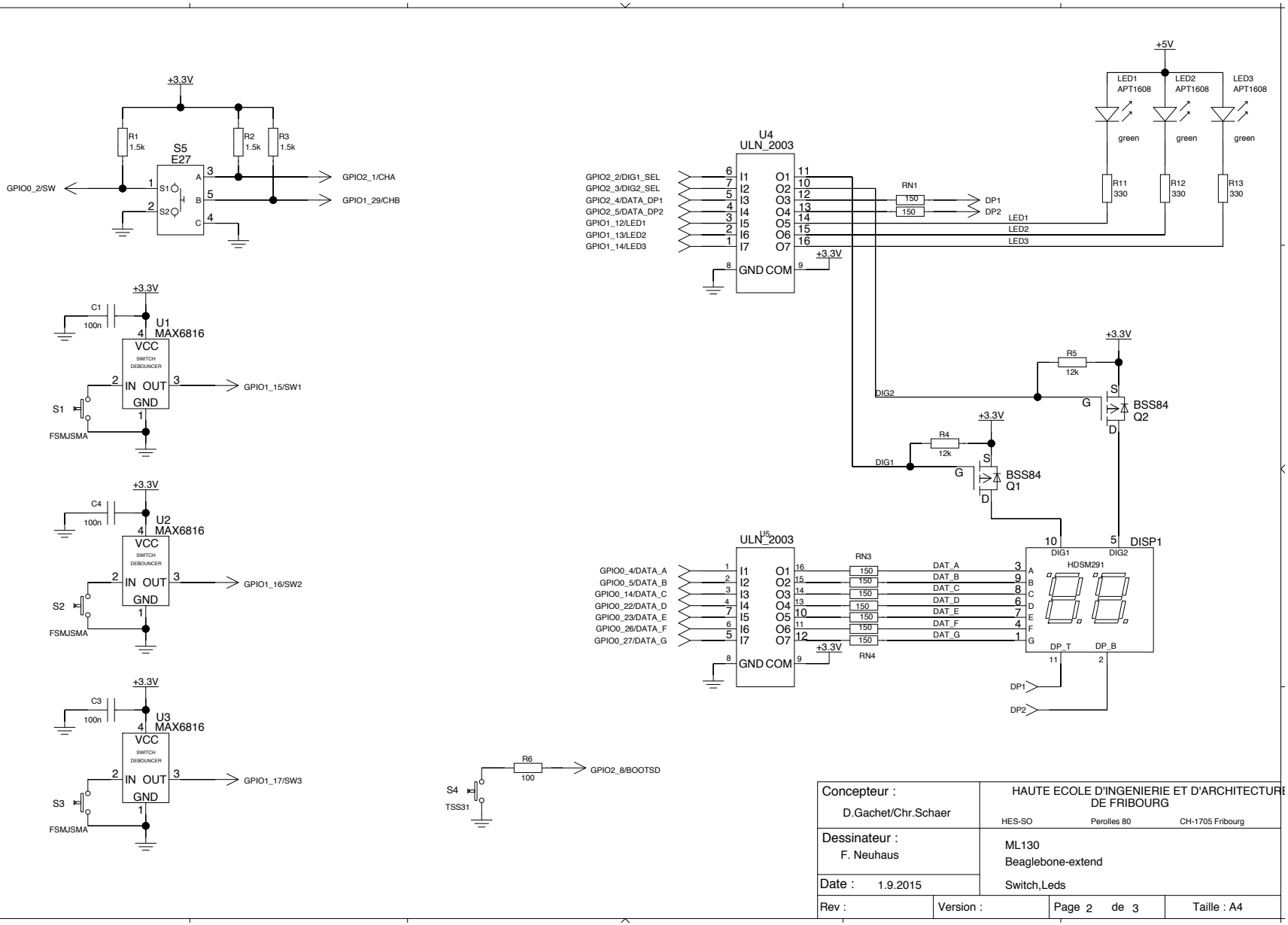
Les questions...

- Pourrait-on se passer des fichiers d'entête en C ? Si oui, comment ? Si non, pour quelle raison ?
- Quelle est l'utilité du pragma `#pragma once` dans les fichiers d'entête ? Doit-il être accompagné d'une autre directive ? Si oui, laquelle ?
- Que faut-il placer dans un fichier d'entête ?
- Quelle est l'utilité des mots-clef `extern` et `static` ?
- Comment faut-il procéder pour définir une constante en C ?
- Quelle(s) différence(s) existe-t-il entre les instructions `#define MAX 10` et `const int MAX=10;` ?

Les questions...(II)

- Comment peut-on définir une énumération en C ? Quelle est son utilité ?
- Quelle(s) différence(s) existe-t-il entre une structure en C (`struct S{}`) et une classe en Java (`class C{}`) ?
- Comment faut-il procéder pour définir un tableau en C ? Peut-on lui donner des valeurs initiales lors de sa définition ?
- Comment faut-il procéder pour obtenir le nombre d'éléments contenus dans un tableau ?

Schéma de la carte d'extension

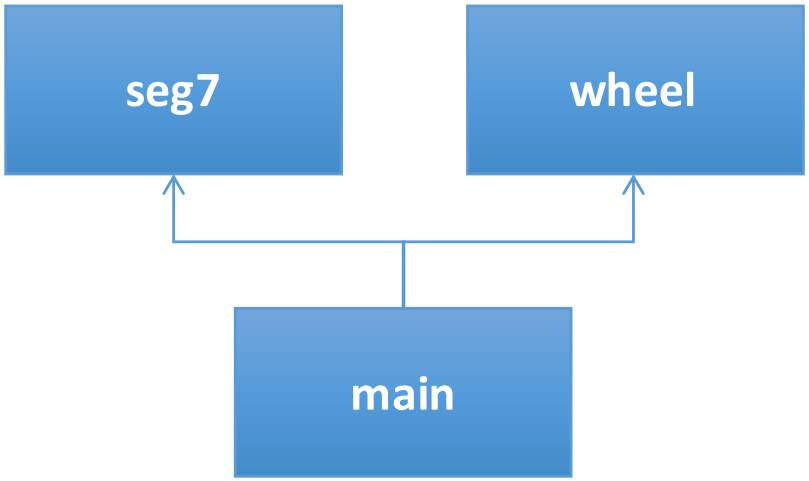


Concepteur : D.Gachet/Chr.Schaer	HAUTE ECOLE D'INGENIERIE ET D'ARCHITECTURE DE FRIBOURG HES-SO Perolles 80 CH-1705 Fribourg		
Dessinateur : F. Neuhaus	ML130 Beaglebone-extend		
Date : 1.9.2015	Switch,Leds		
Rev :	Version :	Page 2 de 3	Taille : A4

Contrôleur GPIO

- Les portes d'entrée/sortie utilisées pour contrôler l'encodeur rotatif et l'afficheur 7-segments sont pilotées par l'intermédiaire des contrôleurs GPIO du µP
- Le module "`am335x_gpio.h`" de la bibliothèque "`libbbb.a`" offrent des services pour
  - initialiser les contrôleurs "`am335x_gpio_init(…)`"
  - configurer les portes en entrée "`am335x_gpio_setup_pin(…)`"
  - configurer les portes en sortie avec un état initial "`am335x_gpio_setup_pin_out(…)`"
  - lire l'état d'une porte d'entrée "`am335x_gpio_get_state(…)`"
  - changer l'état de portes de sortie "`am335x_gpio_change_states(…)`"

Structure générale de l'application



- "`main`" module principal implémentant l'algorithme principal de l'application
- "`seg7`" module pour le pilotage de l'afficheur 7-segments et l'affichage de value du compteur
- "`wheel`" module pour la lecture de l'état de l'encodeur rotatif et de son bouton poussoir