



Année universitaire 2016/2017

NOM :

**Consignes relatives au déroulement de l'épreuve**

*A remplir obligatoirement par l'enseignant responsable du contrôle*

**Date : 16 novembre 2016**

**Devoir Module Bases des systèmes embarqués – Partie Cours**

**Durée: 1h30**

**Professeurs responsables : François Joly – Yasmina Layouni**

**Documents :** ☒ autorisés ☐ non autorisés

**Si oui : type(s) de documents autorisés : Photocopie Fiche technique du 8051F020 uniquement**

**Calculatrices :** ☒ autorisées ☐ non autorisées

**Si oui : type(s) de calculatrices autorisées : alphanumériques**

HL

**LES TELEPHONES PORTABLES ET AUTRES APPAREILS DE STOCKAGE DE DONNEES NUMERIQUES NE SONT PAS AUTORISES.**

**Les téléphones portables doivent être éteints pendant toute la durée de l'épreuve et rangés dans les cartables.**

**S'agissant de contrôle sans document, les trousseaux doivent être rangés dans les cartables.**

**Les cartables doivent être fermés et posés au sol.**

**Les oreilles des candidats doivent être dégagées.**

**Rappels importants sur la discipline lors des examens**

La présence à tous les examens est strictement obligatoire ; tout élève présent à une épreuve doit rendre une copie, même blanche, portant son nom, son prénom et la nature de l'épreuve.

Une absence non justifiée à un examen invalide automatiquement le module concerné.

Toute suspicion sur la régularité et le caractère équitable d'une épreuve est signalée à la direction des études qui pourra décider l'annulation de l'épreuve; tous les élèves concernés par l'épreuve sont alors convoqués à une épreuve de remplacement à une date fixée par le responsable d'année.

Toute fraude ou tentative de fraude est portée à la connaissance de la direction des études qui pourra réunir le Conseil de Discipline. Les sanctions prises peuvent aller jusqu'à l'exclusion définitive du (des) élève(s) mis en cause.

Merci d'indiquer sur chaque page votre nom et votre prénom

## Devoir Session 1 – BSE - Partie Cours

### Durée recommandée : 1H30

Questions relatives au cahier des charges : « *Module de surveillance de container* »

Remarque : Les questions peuvent être traitées indépendamment les unes par rapport aux autres.

Dans cette partie examen « cours », nous ne demandons pas de code C, nous voulons des algorithmes, du pseudo code afin de nous montrer votre bonne compréhension du problème.

Précisions sur l'évaluation :

Certains exercices peuvent être traités de différentes manières. Toutefois l'évaluation sera maximale pour des solutions qui sollicitent le moins possible le processeur et utilisent au mieux les périphériques.

### Gestion du capteur ADXL377 d'Analog Devices – Analyse de documentation

#### 1. Question – Analyse du fonctionnement de l'ADXL377 – 2 points.

- 1.1. Les informations produites par le capteur sont-elles transmises sous forme numérique ou analogique ?
- 1.2. Quelle est la plage de sécurité des signaux produits par le capteur ?
- 1.3. Quelle est la plage de mesure des signaux produits par le capteur pour cette application compte tenu du cahier des charges ?
- 1.4. Quelles est la résolution minimale du dispositif de conversion (CAN) à mettre en œuvre ?
- 1.5. Compte tenu de l'application, quel est le temps de conversion maximum du CAN ?

Les informations transmises par le capteur sont sous la forme analogique  
Le capteur fournit des signaux dans la plage 0,1 - 2,0 V  
Pour cette application la plage de mesure sera de -100g à 100g soit des signaux  
de 0,1V - 2,1V car le capteur est linéaire jusqu'à 180g

On souhaite avoir une résolution supérieure à 0,5g il faut donc au moins 260 points  
de mesure.

Le DAC du 8051 est un dac 12 bits fonctionnant à la tension de référence 12,43V  
Le DAC a donc une résolution maximale de  $5,3 \times 10^{-4} \text{ V}$  soit d'environ  
un dixième de g.

Nos signaux étant dans la plage 0,1V - 2,1V on peut utiliser la plage  
complète (à moins de conditionner le signal et de multiplier un gain 2).

On a donc environ 2000 points de mesure sur cette plage soit une résolution  
de  $\frac{1}{10} \text{ g}$  environ.

Si on souhaite une résolution de 0,5g  $\Rightarrow 3,25 \text{ mV}$  environ il faut donc utiliser au moins  
10 bits du dac ( $> 770$  points de mesure)

Il faut que le temps de conversion soit inférieur à 1  $\mu\text{s}$   
5 kHz

## 2. Question – Configuration du CAN (ADC0 ou ADC1) – 4 points.

- 2.1. Expliquez pourquoi et comment vous allez utiliser le ou les convertisseurs ADC0 et ADC1 du 8051F020.
- 2.2. Expliquez toutes les configurations requises pour la mise en œuvre de la solution de conversion.

Remarque : A ce stade on ne s'intéresse pas encore au mode de déclenchement de la conversion.

On utilisera le convertisseur ADC0 du 8051 pour lire la valeur du capteur ADXL 377.

La résolution et le temps de conversion de l'ADC est suffisants pour cette application.

Pour cela on définira dans un premier la référence de tension qui sera la référence du 8051 (2.3V typ)

L'ADC0 devra être configuré avec un gain de 1 et il lira toutes les différentes entrées analogiques.

Pour cela il faudra lancer trois conversions consécutives tout en changeant le registre ANUX0 afin de sélectionner l'axe du capteur.

On utilisera les 10 bits du <sup>ADC</sup> DAC et sa fréquence sera inférieure à 18.432 MHz

Le DAC sera utilisé en mode "Single-Ended" et on publiera le registre à gauche (convertisseur traite au moins 10 bits, ce sera plus facile)

## 3. Question – Branchement du ADXL377 sur le 8051F020 – 2 point.

Expliquez comment vous allez brancher ce capteur sur le microcontrôleur (GPIO ? broches dédiées ?) Justifiez.

On branchera le capteur sur les trois premières entrées analogiques qui sont des entrées dédiées.

Le capteur ne demande qu'un peu de courant, ce qui le rend compatible avec une liaison directe au microcontrôleur.



4. Question – Mise en œuvre de la conversion – Calcul de l'accélération -3 points.

A partir du résultat produit par la conversion A/N, expliquez comment vous transformez cette valeur (quel est son format ?) en une valeur signée exprimée en dixième de g (int Acc\_dixieme\_de\_g) (pas de code, simplement du pseudo-code, une formule de calcul, un algorithme...)

Récupérer `ADCOH/ADCOL`

Convertir en décimal, éliminer la justification dans le calcul

Convertir en g avec pour référence  $0g = 1.5V$  (mesurer à l'oscilloscope précédemment ?)  
↳ produit en croix  $1.5 \rightarrow x$   
 $2.43 \rightarrow 2.10$

Tronquer au dixième de g

5. Question – Mise en œuvre de la conversion – Mode mémorisation des valeurs maximales -5 points.

Dans ce mode, on fait fonctionner la conversion sur les 3 voies avec pour chaque voie une fréquence d'échantillonnage de 5 KHz, mais on souhaite uniquement conserver les valeurs maximales d'accélération positives et négatives.

Proposez une solution logicielle en mode foreground, permettant de répondre à ce besoin en consommant un minimum de temps processeur.

Les valeurs maximales d'accélération seront stockées dans 2 variables globales : unsigned int Max\_acc et unsigned int Min\_acc (ces variables seront utilisées pour stocker les valeurs maximales, quelles que soient les voies X, Y et Z).

→ Configuration Timer pour charger régulièrement la voie d'acquisition

↳ Timer 2 16 bit en autoreload

↳ Sysclock / 12 pour horloge du timer (CKCON2 = 0xEF)

↳ Valeur de préchargement ( $\sim 65535$ )

↳ interruption tout les  $0.06ms \sim \frac{2ms}{3}$

↳ Activation des interruptions

→ Configuration ADC en mode "Auto-Interrupt"

↳ Configuration de la source de tension de référence (référence interne)

↳ Définir AIN0, 1 et 2 en mode single ended

↳ Définir horloge du DAC tel que le temps d'acquisition est  $< \frac{0.2}{3} ms$

↳ justification à droite

↳ ~~READ\_ADC\_TH/ADACTL et ADCQTH/ADQCTL qui sont 0~~ (car on veut à l'instant donné la valeur actuelle des bits du registre)

↳ Gain = 1

⇒ Autorisation interruption générale + Timer 2 + ADC

→ ISR\_TIMER\_2

↳ if (AMXOSL == 3)

{ AMXOSL = 0 }

else { AMXOSL++; // ou AMXOSL = AMXOSL + 1; }

// Changement de la source de l'ADC

↳ Remise à zéro du flag  
→ Appel de la fonction ADC\_Read

~~ISR-ADC Priority 8~~

~~↳ if (Value - lue - ADC < Value - Min - Acc)~~

~~→ Min - Acc = Value - lue - ADC;~~

~~→ ADCOLTH : ADCOLTL prend la valeur de Min - Acc~~

~~↳ if (Value - lue - ADC > Min - Acc)~~

ADC - Read

↳ lancement conversion

↳ Attente conversion

↳ if (Value - lue > Value - max)

Value - max = Value - lue

↳ if (Value - lue < Value - min)

Value - min = Value - lue

# 6. Question – Mise en œuvre de la conversion – Mode dépassement de valeurs critiques -4 points.

Dans ce mode, on fait fonctionner la conversion sur une seule voie de mesure, la voie X, à sa vitesse maximale (5Khz), mais on souhaite déclencher une alarme si les valeurs d'accélération positives et négatives dépassent un seuil.

Proposez une solution logicielle en mode foreground, permettant de répondre à ce besoin en consommant un minimum de temps processeur.

Les valeurs seuil d'accélération seront stockées dans 2 variables globales : `unsigned int Seuil_max_acc` et `unsigned int Seuil_Min_acc`.

→ Configuration ADC en mode Windows interrupt (3)  
 ↳ Même configuration que précédemment pour ADC  
 + Paramétrage du mode windows interrupt  
 → `ADCOLTH & ADCOLT = Seuil_min_acc`  
 → `ADCOBTH / ADCOBTL = Seuil_max_acc` → `AMXOSL = 0, AMXOCF = 0, ADCIST = 0`  
 → Autoriser interruption générale + interruption ADC0, + `AD0WINT = 0`  
`ISR_ADC_0 Priority 0`  
 {  
   Activation - Alarme();  
   Remise à zéro du flag d'interruption `AD0WINT`  
 }

}

