

# QCMs - Tutoriels Nucleo

4 mars 2019

## Table des matières

Premiers pas avec Nuc1eo	1
Interfaçage	4
Gestion du temps & Interruptions	8
Traitement/Asservissement	10
Communication	13

## Premiers pas avec Nuc1eo

### Créer un projet sous MBED

**Question 1** L'interface de développement MBED

- est accessible en ligne
- B est installé sur tous les ordinateurs de l'IOGS
- C est un logiciel payant

**Question 2** Ce logiciel

- a une fonction de compilateur C/C++
- nécessite que l'on précise la cible matérielle sur laquelle le programme va fonctionner
- est une variante des compilateurs de C/C++ habituels

### Tester une première application sur Nuc1eo

**Question 3** La LED câblée sur la carte et appelée LED1

- est connectée à la broche PA\_5 du microcontrôleur
- est connectée à la broche D13 du microcontrôleur
- C n'est connectée à aucune broche du microcontrôleur

**Question 4** Le programme « Blinky LED »

- A permet d'afficher « Hello World! » sur l'ordinateur
- fait clignoter la LED LED1 de la carte
- C allume la LED LED1 pendant 2s puis l'éteint.

**Question 5** Remplacer la ligne 8 du programme « Blinky LED » par :

`wait(2)`

a pour conséquence :

- de modifier la fréquence de clignotement de la LED `LED1`.
- de réduire la durée pendant laquelle la LED `LED1` est éteinte
- d'augmenter la durée pendant laquelle la LED `LED1` est éclairée

**Question 6** Un programme pour une application embarquée :

- a besoin d'un ordinateur pour être exécuté
- possède une boucle infinie qui ne s'arrête jamais
- est téléversé après compilation sur le microcontrôleur pour fonctionner en autonomie
- s'écrit toujours en langage C

## Piloter une LED

**Question 7** La ligne :

`include "mbed.h"`

- permet d'importer une bibliothèque décrivant les entrées et les sorties de la carte `Nucleo`
- doit être présente dans l'en-tête de tous les programmes
- contient une ligne du type `DigitalOut myled(D13)` ; permettant de définir quelle broche est utilisée en sortie.

**Question 8** La ligne de code `DigitalOut toto(D10);`

- permet de déclarer la broche `toto` comme une sortie numérique
- permet de déclarer la broche `D10` comme une sortie numérique
- est incorrecte, on doit écrire `DigitalOut toto(D10,1)` ; ou `DigitalOut toto(D10,0)` ;

**Question 9** La valeur maximale des tensions par la carte `Nucleo` est égale à :

- 5 A
- 5 V
- 3.3 V

**Question 10** Le montage 1 du tutoriel

- permet d'allumer la LED quand une tension de 3.3 V est appliquée à la broche `D10`
- permet d'allumer la LED quand une tension de 0 V est appliquée à la broche `D10`
- doit être préféré au montage 2

**Question 11** Le montage 2 du tutoriel

- comprend une résistance  $R_{LED}$  que l'on peut calculer par la même formule que lors du montage 1.
- doit être préféré au montage 1
- permet d'allumer la LED quand une tension de 0 V est appliquée à la broche `D10`

**Question 12** Dans le montage 1 du tutoriel, on souhaite limiter le courant à 20 mA. La LED possède un seuil de 2.3 V. On doit donc choisir une valeur  $R_{LED}$

- inférieure à 10  $\Omega$
- supérieure à 50  $\Omega$
- supérieure à 10 k $\Omega$

## Déboguer son programme et utiliser l'affichage série

**Question 13** On souhaite mesurer le temps d'exécution d'une série d'instruction à l'oscilloscope en affichant l'évolution de la tension aux bornes d'une broche. Pour cela il faut ajouter des commandes du type :

- `yourtest = 1;` avant le code des instructions et `yourtest = 0;` après.
- `DigitalOut yourtest (PA_4, 0);` dans la partie initialisation du programme
- `yourtest = 0;` avant le code des instructions et `yourtest = 1;` après.
- `char c = 300;` à la suite du programme

**Question 14** Connaître la valeur d'une variable entière interne au microcontrôleur, à un instant donné :

- n'est pas possible.
- est possible en utilisant un ordinateur relié au microcontrôleur par une liaison série.
- est réalisable en utilisant un oscilloscope.

**Question 15** Le logiciel TeraTerm :

- établit une communication de protocole RS232 côté ordinateur.
- est conçu pour fonctionner avec la carte Nucleo seulement.
- établit une communication de protocole RS232 côté microcontrôleur.

## Différencier un microcontrôleur d'un micro-ordinateur

**Question 16** Les éléments indispensables au bon fonctionnement d'un microcontrôleur sont :

- un gestionnaire d'interface graphique
- des entrées analogiques
- des espaces mémoires RAM/ROM

**Question 17** Un microprocesseur :

- permet d'exécuter uniquement des programmes en C
- est un calculateur généraliste
- est capable d'exécuter 2000 instructions élémentaires différentes (Intel i7)
- équipe les systèmes embarqués des automobiles

**Question 18** Un microcontrôleur :

- contient une zone de mémoire de données
- intègre tous les éléments nécessaires à l'exécution d'une tâche spécifique
- possède des ressources matérielles illimitées
- possède systématiquement une unité de calcul en virgule flottante

**Question 19** Les microcontrôleurs :

- permettent de répondre rapidement à des événements extérieurs
- ne peuvent être programmés qu'en langage machine
- intègrent une matrice de portes logiques

**Question 20** Un programme pour système embarqué :

- doit s'arrêter systématiquement
- doit pouvoir répondre dans un temps limité à toute sollicitation extérieure
- doit en permanence scruter les entrées et mettre à jour les sorties en conséquence

## Trouver des informations dans la documentation

**Question 21** Les pages de documentation des fonctions présentes dans l'API de MBED

- A ne sont pas utiles pour apprendre à utiliser les fonctions
- sont écrites en anglais
- sont disponibles en ligne
- font uniquement la liste des classes

**Question 22** Sous MBED `Compiler`, je peux :

- changer de carte de développement, sans avoir à modifier le code
- vendre mon code
- importer des bibliothèques que j'ai moi-même réalisées
- partager mon code avec d'autres

**Question 23** La bibliothèque `RN41` :

- contient 1 classe
- contient la fonction `getLine()`
- ne contient aucune fonction
- contient 2 classes

## Interfaçage

### Récupérer une information numérique

**Question 24** Pour récupérer une information numérique sur une entrée, il faut :

- configurer la broche associée en entrée en la déclarant via l'instruction `DigitalIn`
- s'assurer que la tension appliquée sur cette broche soit égale à 0 V ou 5 V
- lui associer un nom de variable incluant le préfixe `in**`

**Question 25** Une entrée numérique est bien adaptée pour :

- A réaliser le filtrage numérique d'un signal
- B récupérer l'information d'un capteur de température
- tester l'état d'un interrupteur fin de course
- récupérer l'information issue d'un comparateur à ALI

**Question 26** Le bouton poussoir `USER_BUTTON` :

- A est connecté au niveau bas (0 V) par une résistance de 4,7 k $\Omega$
- est connecté au niveau haut (3,3 V) par une résistance de 4,7 k $\Omega$
- est connecté physiquement à l'entrée numérique `PC_13`

### Câbler un bouton-poussoir

**Question 27** Dans le montage proposé dans le tutoriel, quelle est la valeur de la tension appliquée à la broche `D7` au repos ?

- A 0 V
- B 5 V
- C 1
- D 0

**Question 28** Dans le montage proposé dans le tutoriel, quelle est la valeur de la tension appliquée à la broche D7 quand on appuie sur le bouton ?

- A 1
- B 0
- C égale à la valeur de  $V_{DD}$
- D 0 V

**Question 29** Un bouton poussoir :

- A est un composant électronique passif
- B est un capteur de type tout ou rien (TOR)
- C doit être câblé sur l'entrée d'un ADC car la tension qu'il délivre est comprise entre 0 V et 3,3 V

**Question 30** Lors de l'appui sur le bouton poussoir BP :

- A le courant qui circule au travers de la résistance  $R_{BP}$  est nul
- B le courant qui circule au travers de la résistance  $R_{BP}$  est égale à 0,33 mA
- C l'entrée numérique D7 voit un niveau logique haut
- D l'entrée numérique D7 voit un niveau logique bas

**Question 31** La ligne de code

```
a = mon_bouton.read();
```

- A est automatiquement appelée lors de l'appui sur le bouton poussoir
- B peut précéder l'instruction `DigitalIn mon_bouton(D7);`
- C permet de lire une entrée numérique et de stocker sa valeur dans la variable a
- D doit être écrite à chaque endroit du programme où l'état du bouton poussoir doit être testé

## Utiliser un affichage LCD

**Question 32** L'écran LCD est relié au microcontrôleur :

- A par l'intermédiaire d'une liaison série
- B par l'intermédiaire de 3 fils (et l'alimentation)
- C par l'intermédiaire d'une liaison Bluetooth
- D par l'intermédiaire d'un protocole parallèle

**Question 33** Lorsqu'on exécute la ligne suivante, `spi.write('a');`, de l'exemple Ecriture d'un caractère du tutoriel :

- A la série binaire 01100001 est transmise sur la broche MOSI de la carte Nucleo
- B le caractère 'a' est affiché à l'endroit où se trouve le curseur sur l'écran LCD
- C l'écran LCD est rempli du caractère 'a'
- D la valeur de la variable a est affichée sur l'écran LCD

**Question 34** Lorsqu'on exécute la ligne suivante :

```
LCD_DOG_setPosition(2, 3);
```

- A le curseur de l'écran LCD se place à la ligne 3 et à la colonne 2
- B le curseur de l'écran LCD se place à la ligne 2 et à la colonne 3
- C l'écran LCD est effacé
- D la série binaire 10010011 est transmise sur la broche MOSI de la carte Nucléo

**Question 35** Dans l'exemple complet du tutoriel :

- A la série binaire 10010011 est transmise sur la broche MOSI de la carte Nucléo
- B le curseur de l'écran LCD se place à la ligne 2 et à la colonne 3
- C le curseur de l'écran LCD se place à la ligne 3 et à la colonne 2
- D l'écran LCD est effacé

## Récupérer un signal analogique

- Question 36** Les convertisseurs analogique-numérique du microcontrôleur de la carte `Nucleo` :
- A échantillonnent la tension appliquée à une entrée analogique à la fréquence maximale de 5 MHz
  - B peuvent se voir appliquer une tension sinusoïdale d'amplitude 3,3 V
  - C quantifient la tension appliquée à une entrée analogique avec un quantum de  $3,3/(2^{16} - 1)$  V
  - D quantifient la tension appliquée à une entrée analogique avec un quantum de  $3,3/(2^{12} - 1)$  V

- Question 37** Les entrées analogiques :
- A sont au nombre de 8 sur le connecteur arduino de la carte `Nucleo`
  - B sont reliées à 3 ADC 16bits fonctionnant à 5 MHz de fréquence d'échantillonnage maximale
  - C sont adaptées à la lecture de l'état d'un bouton poussoir
  - D acceptent des tensions comprises entre -3,3 V et 3,3 V

- Question 38** La fonction `analog_in.read_u16()`
- A permet de récupérer le résultat de la conversion analogique-numérique de l'entrée `u16`
  - B renvoie un entier compris entre 0 et  $2^{16} - 16$
  - C renvoie un entier compris entre 0 et 65535

## Régler la luminosité d'une LED

- Question 39** La principale variable qui détermine la luminosité d'une LED perçue par un œil humain est
- A La tension moyenne à ses bornes.
  - B La puissance moyenne qu'elle consomme.
  - C Le courant moyen qui la traverse.

- Question 40** La modulation par largeur d'impulsion est la méthode utilisée pour faire varier la luminosité d'une LED
- A Car il est plus facile de générer une tension de ce type qu'une tension analogique continue avec un microcontrôleur
  - B Car on s'assure ainsi de la proportionnalité de la luminosité moyenne obtenue avec la commande
  - C Car il suffit de moduler à une fréquence supérieure à 40 Hz pour mettre à profit la persistance rétinienne

- Question 41** La syntaxe d'utilisation d'une sortie modulée en largeur d'impulsion nécessite des instructions du type
- A `PwmOut Ma_Led_Pwm(D3)` dans l'en-tête avant le main et `Ma_Led_Pwm.period(RC)` dans la boucle
  - B `Ma_Led_Pwm.pulsewidth_us(Tpulse)` OU `Ma_Led_Pwm.pulsewidth(Tpulse)`
  - C `PwmOut(D3)` dans l'en-tête avant le main et `PwmOut.write(value)` dans la boucle

## Générer une tension analogique

- Question 42** La carte `Nucleo` a un système de conversion DAC sur 12 bits qui permet d'obtenir une tension
- A de rapport cyclique variable entre 0 à 4095 sur la broche `PA_4`
  - B réglable entre 0 et 3,3 V par pas de  $240 \mu\text{V}$  sur la seule broche `PA_5`
  - C réglable entre 0 et 1 V au pas de  $240 \mu\text{V}$  sur la seule broche `PA_5`.
  - D réglable entre 0 et 3,3 V par pas de  $\frac{3,3\text{V}}{1024}$  sur les trois broches `PA_5`, `PA_6` et `PA_7`

- Question 43** L'écriture du mot de 12 bits du DAC dans un mot de 16 bits se fait
- A Directement, sans décalage.
  - B En commençant à gauche par le MSB à l'aide d'un décalage quadruple c'est à dire une commande du type «4
  - C En commençant à gauche par le MSB et en remplissant les 4 LSB de 1 avec une instruction du type `DAC = 0x10*a+0x0F`

**Question 44** La mise en œuvre logicielle du DAC nécessite des instructions du type :

- A `AnalogOut MyVar (PA_5)` dans l'en-tête avant le main et `AnalogOut.write(MyVar (value))` dans la boucle
- B `AnalogOut (PA_5)` dans l'en-tête avant le main et `AnalogOut.write(value)` dans la boucle
- C `AnalogOut MyVar (PA_5)` dans l'en-tête avant le main et `MyVar.write(value)` dans la boucle

## Contrôler un mouvement angulaire à l'aide d'un servomoteur

**Question 45** Appliquer une tension à un servomoteur standard sous forme d'impulsions de période 20 ms a pour conséquence :

- A de modifier sa position angulaire entre  $-90 \text{ deg}$  à  $+90 \text{ deg}$ , selon le rapport cyclique, selon une fonction affine
- B de rendre sa vitesse de rotation proportionnelle au rapport cyclique des impulsions
- C de modifier sa position angulaire lorsque le rapport cyclique est compris entre 0 et 100%

**Question 46** Parmi les propositions suivantes, lesquelles sont correctes ?

- A Le signal de commande doit avoir un rapport cyclique compris entre 0 et 0,1, pour une période de 20 ms.
- B Un servomoteur standard doit être alimenté à l'aide d'une alimentation continue de 5 à 6 V
- C Le contrôle de l'angle d'un servomoteur standard est fait à l'aide d'une tension continue variable entre 0 et 5 V
- D Le signal de commande doit avoir une période de 1 ms

**Question 47** Parmi les propositions suivantes, lesquelles sont correctes ?

- A Lorsqu'on applique un signal de commande de rapport cyclique de 0,15, le servomoteur revient à un angle de  $0^\circ$
- B Lorsqu'on supprime l'alimentation d'un servomoteur, il reste dans sa position
- C Lorsqu'on supprime le signal de commande d'un servomoteur, il reste dans sa position

## Faire varier la vitesse d'un moteur à CC

**Question 48** Le principe de la modulation de largeur d'impulsion :

- A peut directement être appliqué via la sortie du microcontrôleur au moteur à CC
- B ne peut pas s'appliquer à un moteur à CC sans filtrage
- C peut s'appliquer à un moteur à CC du fait de son inertie
- D nécessite un transistor (ou autre circuit) qui "booste" la commande PWM

**Question 49** Dans le cas de la Commande réversible en sens de rotation du tutoriel, quelle(s) proposition(s) est (sont) correcte(s) ?

- A Ce type de commande nécessite 2 signaux PWM complémentaires
- B Commander T1 et T3 en même temps (sans commander T2 et T4) permet d'arrêter le moteur
- C Commander T1 et T2 en même temps n'a aucun effet sur le montage
- D Ce type de commande nécessite 2 signaux PWM indépendants

## Connecter une source sonore

**Question 50** Le signal électrique issu d'un microphone :

- A est à valeur moyenne nulle et doit donc être décalé de 1,65 V pour s'adapter à l'ADC
- B doit être amplifié
- C est directement applicable sur une entrée analogique de la carte Nucleo

**Question 51** Sur le schéma de câblage proposé dans le tutoriel :

- A l'ensemble  $C = 10 \mu\text{F}$ ;  $R = 47 \text{k}\Omega$  est placé en amont de l'échantillonneur et constitue donc un filtre anti-repliement
- B le filtre  $R = 1,5 \text{k}\Omega$ ;  $C = 4,7 \text{nF}$  a une fréquence de coupure compatible avec le signal sonore
- C la tension  $V_s$  peut directement être appliquée à un haut parleur
- D la tension de sortie du CNA (DAC) est à valeur moyenne nulle.

**Question 52** Le code de test fourni :

- A gère la fréquence d'échantillonnage au travers d'une interruption associée à un timer
- B peut servir sans modification à traiter le signal issu de l'enregistrement d'une chauve souris
- C est adapté au filtrage numérique de tout signal audio car la fréquence d'échantillonnage de 33 kHz est supérieure à 20 kHz
- D peut servir de point de départ à la mise en œuvre d'un filtre numérique

## Gestion du temps & Interruptions

### Faire une action à intervalle régulier

**Question 53** Qu'appelle-t-on timer ?

- A l'horloge interne du microcontrôleur
- B un module générant une interruption au bout d'un certain nombre de coups d'horloge
- C Un oscillateur à une fréquence très précise

On considère le code du listing 1

Listing 1 – Code

```

1 Ticker toggle_led_ticker;
2 void toggle_led(void);
3 int main() {
4 toggle_led_ticker.attach(&toggle_led, 1);
5 while (1) {}
6 }
```

**Question 54** `toggle_led` désigne :

- A un signal d'interruption
- B une fonction
- C un pointeur

**Question 55** Dans ce code, la routine d'interruption est appelée

- A toutes les secondes
- B toutes les millisecondes
- C à chaque front montant de l'horloge interne
- D jamais

### Faire des actions à intervalle régulier

**Question 56** Quelle est la proposition correcte ?

- A Un timer matériel peut fonctionner sans timer logiciel.
- B Un timer logiciel peut fonctionner sans timer matériel

**Question 57** Dans les exemples de code de ce tutoriel, `TIMER1` est :

- A une fonction
- B une constante symbolique
- C une variable

**Question 58** En ayant déclaré `InterruptIn bp_int (PA_8)`, que faut-il mettre dans le main pour appeler la fonction `routine_bp` à chaque front descendant de la broche `PA_8` ?

- A `bp_int.fall(&routine_bp);`
- B `PA_8.fall(routine_bp);`
- C `if (bp_int.fall) {routine_bp();}`
- D `bp_int.fall(routine_bp);`

## Générer un signal d'horloge sans GBF

**Question 59** Le composant `NE555` :

- A permet de générer des signaux rectangulaires
- B permet de générer des signaux rectangulaires de rapport cyclique inférieur à 50%
- C permet de générer des signaux triangulaires
- D permet de générer des signaux de fréquence de 3 à 10 MHz

**Question 60** Lorsque le composant `NE555` est en mode monostable :

- A un nouveau passage à l'état bas de l'entrée `TRIGGER`, alors que la sortie n'est toujours pas revenu à l'état bas, relance le délai du monostable
- B la plus courte impulsion est de 200 ns sur sa sortie
- C le temps haut du signal de sortie ne dépend pas du temps à l'état bas de la broche `TRIGGER`
- D un front descendant sur l'entrée `TRIGGER` déclenche le passage à '1' de la sortie

**Question 61** Un quartz à 32,768 kHz :

- A est associé à des composants actifs permettant de le faire osciller
- B permet d'obtenir un signal triangulaire lorsqu'il est mis dans le montage proposé dans la partie Montage oscillateur à base d'un quartz du tutoriel
- C est souvent utilisé dans les montres pour obtenir une fréquence de 1 Hz
- D est un composant qui n'a pas besoin d'une alimentation externe

## Faire une action instantanément suite à un événement externe

**Question 62** Le fonctionnement par interruption permet d'éviter de vérifier en permanence l'état des différents composants/capteurs

- A Faux
- B Vrai

**Question 63** Une interruption bloque l'exécution du programme principal pour effectuer une tâche particulière

- A Vrai
- B Faux

**Question 64** Une routine d'interruption est :

- A une fonction appelée lorsqu'une interruption est générée
- B un signal qui interrompt l'exécution du programme principal
- C un événement extérieur au microcontrôleur

**Question 65** Ayant déclaré `InterruptIn test (PA_12)` au préalable, pour effectuer une action lors du passage de 0 à 1 de la broche `PA_12` on utilise :

- `test.rise(&ma_fct);`
- `test.fall(&ma_fct);`
- `test.attach(&ma_fct, UP);`

**Question 66** Quelle bibliothèque faut-il inclure pour gérer les interruptions :

- Aucune, on utilise des fonctions du langage C
- `interrupts.h`
- `mbed.h`

## Faire une action périodique à une fréquence précise

**Question 67** Parmi les propositions suivantes, lesquelles sont correctes ?

- Il est possible d'utiliser autant d'objets de type `Ticker` que l'on souhaite
- Il est possible d'utiliser autant d'entrées d'interruption que l'on souhaite, associées à des oscillateurs différents
- Les `Ticker` permettent de descendre à des périodes de  $1 \mu s$
- Les `Ticker` permettent de descendre à des périodes de  $100 ns$

# Traitement/Asservissement

## Caractériser un traitement numérique

On souhaite mesurer le temps d'exécution de l'instruction suivante :

```
myled =!myled;
```

L'objet `myled` a été déclaré comme une sortie via l'instruction `DigitalOut myled(D13);`

**Question 68** A l'aide d'un `Timer` :

- il est possible de mesurer directement ce temps
- il n'est pas possible de mesurer directement ce temps
- on mesure un temps d'environ  $300 ns$  environ pour cette instruction
- il faut mesurer l'exécution d'une centaine d'opérations de ce type pour avoir une bonne évaluation du temps d'exécution

Soit le code suivant :

```
for(int i = 0; i < 100; i++){}
```

**Question 69** Quelles sont les propositions justes :

- Il faut obligatoirement mettre des instructions dans les accolades pour que ce code compile.
- Même s'il ne fait rien, ce code prend un certain temps pour être exécuté.
- Cette boucle prend moins de  $10 \mu s$  pour être exécutée.
- Il faut obligatoirement utiliser un objet `Timer` pour mesurer le temps d'exécution.

Soit le code suivant, inclus dans une boucle infinie, avec `t` défini par `Timer t` :

Listing 2 – Code

```
1 t.reset(); t.start(); t.stop();
2 pc.printf("Exec=%d_usec\r\n",t.read_us()); wait(1);
```

**Question 70** Quelles sont les propositions justes :

- A Ce code affiche `Exec = 1000000 usec.`
- B Ce code ne s'exécute pas.
- C Ce code affiche `Exec = 2 usec.`
- D Ce code permet de connaître le temps d'exécution des instructions `start` et `stop` du `TIMER`.

## Supprimer une fréquence parasite

**Question 71** Si on inclut l'ensemble du code présent dans la fonction `convert` dans la boucle principale `while(1)` (en supprimant la ligne `tik.attach(\&convert, TE);` :

- A ce ne change pas le comportement du microcontrôleur
- B les opérations correspondant au code déplacé ne sont pas réalisées
- C les opérations correspondant au code déplacé ne sont pas réalisées à des instants précis et réguliers

**Question 72** Entre les deux méthodes proposées pour réaliser le filtre à moyenne mobile :

- A le code 1 a un temps de calcul constant
- B le code 2 est plus rapide en temps de calcul que le code 1
- C le code 1 est plus rapide en temps de calcul que le code 2
- D le code 2 a un temps de calcul constant

**Question 73** En utilisant le code 2 pour le filtre à moyenne mobile :

- A on peut avoir une fréquence d'échantillonnage plus grande que pour le code 1
- B on doit nécessairement initialiser le tableau de stockage des échantillons avant un premier appel à la fonction `convert`
- C on doit réaliser la moyenne sur un nombre de points plus faible que dans le code 1 pour une même fréquence d'échantillonnage

**Question 74** Pour le filtre moyeneur :

- A le signal de sortie est en retard sur le signal d'entrée
- B le signal `clk_test` a pour période  $T_E$  et pour temps de calcul  $\Delta T = cte$
- C le signal de sortie est en avance sur le signal d'entrée
- D le signal `clk_test` permet de connaître le temps de calcul d'un échantillon

**Question 75** Si on applique une fréquence d'échantillonnage de 10 kHz pour réaliser le filtre moyeneur :

- A si le signal d'entrée possède une composante à 312,5 Hz, alors celle-ci est supprimée en sortie si on utilise un nombre de points égal à 64
- B il faut nécessairement un filtre anti-repliement à 20 kHz
- C toutes les fréquences du signal d'entrée jusqu'à 5 kHz sont conservées
- D il faut 100 points pour supprimer un signal à 100 Hz

## Supprimer une bande fréquentielle

Dans les 2 prochaines questions, on considère un filtre FIR d'ordre  $n$  dont la fréquence d'échantillonnage est notée  $F_e$ .

**Question 76** Quelles sont les propositions justes ?

- A On peut réaliser un filtre numérique passe-bas de fréquence de coupure supérieure à  $F_e/2$
- B Un tel filtre numérique possédera  $n + 1$  coefficients réels.
- C Pour réaliser un filtre numérique passe-haut, il est indispensable de mettre un filtre anti-repliement passe-bas devant le filtre numérique
- D Un tel filtre numérique possédera  $n - 1$  coefficients réels.

**Question 77** Quelles sont les propositions justes ?

- A Diminuer la fréquence d'échantillonnage modifie l'ordre du filtre
- B Augmenter l'ordre du filtre augmente le temps d'exécution du calcul du filtre
- C Diminuer l'ordre du filtre augmente le temps de conversion des échantillons
- D Augmenter l'ordre du filtre ne modifie pas le temps d'exécution du calcul du filtre

**Question 78** La valeur de la fréquence réduite :

- A est comprise entre 0 à 1
- B peut être affichée que de 0 à 0.5
- C est comprise entre 0 et 0.5
- D est comprise entre -0.5 et 0.5

On souhaite réaliser un filtre FIR de type **passé-bande**, avec les caractéristiques suivantes :

- $F_{ech} = 10$  kHz
- $F_{stop1} = 1$  kHz,  $F_{pass1} = 1.5$  kHz
- $F_{stop2} = 3$  kHz,  $F_{pass2} = 2.5$  kHz
- Atténuation de 60 dB hors bande-passante et 1 dB dans la bande-passante

**Question 79** Quelles sont les propositions justes ?

- A Un filtre *Equiripple* possède 40 coefficients, dont le premier vaut (environ) : 1.5038
- B Un filtre *Equiripple* est d'ordre plus élevé qu'un filtre fenêtré de type **Kaiser**.
- C Un filtre *Equiripple* possède 40 coefficients, dont le premier vaut (environ) : 0.0011
- D Un filtre fenêtré de type **Kaiser** est d'ordre supérieur à 70.

**Question 80** En reprenant l'exemple du filtre passe-haut proposé dans le tutoriel, quelles sont les propositions justes ?

- A Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus basse.
- B Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.
- C L'instruction : `arm_fir_f32(&fir, &in, &out, 1);` s'exécute en moins de 10  $\mu$ s
- D Le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.

## Mettre en œuvre un filtrage plus efficace

On considère un filtre IIR dont la fréquence d'échantillonnage est  $F_e$  et possédant  $N$  coefficients  $\alpha$  et  $M$  coefficients  $\beta$ .

**Question 81** Quelles sont les propositions justes ?

- A  $N$  et  $M$  sont nécessairement égaux
- B Le calcul de la sortie nécessitera  $N + M$  opérations élémentaires.
- C Un tel filtre sera nécessairement instable.

**Question 82** Quelles sont les propositions justes ?

- A A gabarit identique, un filtre IIR sera moins performant qu'un filtre FIR
- B A gabarit identique, un filtre IIR pourra utiliser une fréquence d'échantillonnage plus importante qu'un filtre FIR
- C Des filtres de Butterworth passe-bas et passe-haut, définis avec le même gabarit, auront le même nombre de coefficients

**Question 83** On s'intéresse aux **filtres passe-bas** du paragraphe *Gabarit et conception* de la section *Conception du filtre sous Matlab* proposés dans le tutoriel. On obtient :

- A un filtre d'ordre 14 pour les 2 modes de Chebyshev
- B un ordre inférieur pour le filtre elliptique que pour le filtre de Butterworth
- C un filtre d'ordre 48 pour le filtre de Butterworth
- D un ordre supérieur pour le filtre elliptique que pour les filtres de Chebyshev

**Question 84** En reprenant l'exemple du filtre passe-haut proposé dans le tutoriel, quelles sont les propositions justes :

- A Il le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.
- B Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus basse.
- C L'instruction :  

```
arm_iir_lattice_f32(&my_iir, &in, &out, 1);
```

s'exécute en plus de 10  $\mu$ s
- D Il le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.

## Corriger un asservissement

to do

## Communication

### Faire communiquer deux systèmes

**Question 85** Quelles sont les propositions justes ?

- A Une liaison parallèle de 8 bits cadencée à 10 MHz sera moins rapide qu'une liaison série de 8 bits cadencée à 100 MHz
- B Les liaisons synchrones ont au plus 2 fils de transmission
- C Une liaison asynchrone est plus fiable qu'une liaison synchrone
- D Une liaison Half-Duplex permet d'échanger des données dans les deux directions en même temps

**Question 86** Quelles sont les propositions justes ?

- A Un bus SPI entre un maître et 4 esclaves nécessite qu'une seule ligne de transmission CS depuis le maître
- B Une trame I2C est plus courte qu'une trame SPI
- C Une trame SPI est limitée à 1 octet à chaque fois
- D A fréquence d'horloge équivalente, un bus SPI sera plus rapide qu'une liaison I2C pour transmettre des données utiles

### Configurer une communication point à point RS232

**Question 87** Pour une transmission à 9600 bauds :

- A Il faut environ 1 ms pour transmettre un caractère avec la fonction `putc`
- B On ne peut pas transmettre plus de 5 caractères par seconde.
- C Il faut un peu moins de 100  $\mu$ s pour transmettre un bit
- D Il faut entre 7 ms et 8 ms pour transmettre la chaîne de caractères "Bonjour" avec la fonction `printf`

**Question 88** Pour une transmission à 115200 bauds :

- A l'envoi d'un caractère avec la fonction `putc` prend plus de 100  $\mu$ s
- B Il faut un peu moins de 10  $\mu$ s pour transmettre un bit
- C Il faut plus de 1 ms pour transmettre la chaîne de caractères "ProTIS 2019" avec la fonction `printf`

## Configurer un réseau point à point SPI

On utilise le code suivant pour configurer une liaison SPI :

```
SPI spi(D11, D12, D13);
```

Puis dans le programme principal :

```
spi.format(8, 0); spi.frequency(100000);
```

**Question 89** Quelles propositions sont justes ?

- A On visualisera en permanence une horloge à 100 kHz sur la broche D13
- B Les données transmises du maître vers l'esclave sont visualisables sur la broche D11
- C On visualisera une horloge à 100 kHz sur la broche D13 uniquement lors de l'utilisation de la fonction `spi.write(x)` ;
- D Les données transmises du maître vers l'esclave sont visualisables sur la broche D12

**Question 90** Quelles propositions sont justes ?

- A Un '1' logique est transmis par une différence de potentiel de 0V par rapport à la masse
- B Il faut environ 25  $\mu$ s pour transmettre un caractère par la fonction `spi.write`
- C Le signal d'horloge est à l'état bas lorsqu'il n'y a pas de transmission
- D Les deux fils de données sont à l'état haut lorsqu'il n'y a pas de transmission

## Interfacer un convertisseur numérique-analogique en SPI

**Question 91** Dans l'exemple de génération d'une rampe de tension (pour une alimentation de 3.3 V), pour la méthode standard :

- A le délai entre chacun des échantillons est constant et égal à 1 ms
- B la tension minimale en sortie du DAC est de 1 V
- C la fréquence du signal résultant est d'environ 4 Hz
- D la tension maximale de sortie du DAC est de 3.3 V

**Question 92** Dans l'exemple de génération d'une rampe de tension (pour une alimentation de 3.3 V) :

- A la méthode par interruption est plus longue en temps de calcul que la méthode standard
- B la méthode par interruption permet une mise en place des échantillons plus régulière que la méthode standard

**Question 93** Dans l'exemple de génération d'un signal sinusoïdal (pour une alimentation de 3.3 V) :

- A La fréquence maximale du signal résultant que l'on peut obtenir est supérieure à 200 Hz
- B La méthode par interruption est préférable à la méthode standard
- C Il n'est pas possible d'utiliser la méthode par interruption
- D La fréquence du signal résultant est d'environ 16 Hz

## Ajouter de la mémoire de données en SPI

TO DO

## Configurer un réseau adressable I<sup>2</sup>C

On utilise le code suivant pour configurer une liaison I2C :

```
I2C my_i2c(I2C_SDA, I2C_SCL);
```

Puis dans le programme principal :

```
my_i2c.frequency(1000000);
```

**Question 94** Quelles propositions sont justes ?

- On visualisera une horloge à 1 MHz sur la broche D14 uniquement lors de l'utilisation de la fonction `my_i2c.write(...)`;
- Les données transmises dans les deux sens sont visualisables sur la broche D15
- Il n'est pas nécessaire de connecter un périphérique I2C pour visualiser les trames I2C sortant de la carte MBED
- Il est nécessaire de connecter un périphérique I2C pour visualiser les trames I2C sortant de la carte MBED et de transmettre à l'adresse du périphérique pour obtenir la trame complète

**Question 95** Il est possible dans un même réseau I2C :

- de mettre deux périphériques identiques et de pouvoir les adresser indépendamment s'il est possible de modifier localement leurs adresses
- de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses différentes
- de mettre deux périphériques identiques et de pouvoir les adresser indépendamment
- de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses identiques

## Interfacer un accéléromètre en I<sup>2</sup>C

**Question 96** On s'intéresse à un accéléromètre de type ADXL350 :

- Il est possible de communiquer en SPI et en I2C
- Il est possible de communiquer en SPI uniquement
- Pour communiquer en I2C avec ce composant, il faut forcer la broche CS à 0 V
- Pour communiquer en I2C avec ce composant, il faut forcer la broche CS à 3.3 V

**Question 97** Avec l'accéléromètre de type ADXL350 :

- Il n'est pas possible de récupérer les 6 octets de données en une seule trame de réponse
- Il est possible de récupérer les 6 octets de données en une seule trame de réponse
- Il faut obligatoirement utiliser une entrée d'interruption sur le microcontrôleur, câblée sur INT1 ou INT2 de l'accéléromètre, pour récupérer les données
- Il n'est pas possible de communiquer en I2C en High-Speed Mode