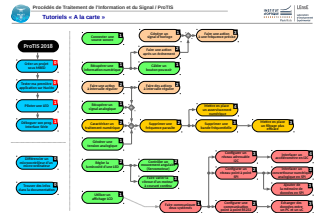
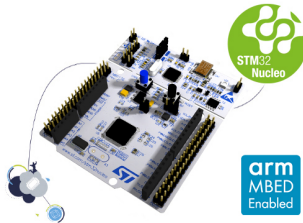


## Prototypage ProTIS

Polycopié complet

### PROcédés de Traitement de l'Information et du Signal



Cycle Ingénieur - 2<sup>ème</sup> année - Palaiseau  
Filières classique et CFA  
Année 2017-2018

Version du 20 mars 2018

Fabien ADAM  
Henri BENISTY  
Pierre GODET  
Julien VILLEMEJANE  
Fabienne BERNARD



# Table des matières

<b>Calendrier</b>	<b>v</b>
<b>Règles de fonctionnement du LEnsE</b>	<b>vii</b>
<b>I Modalités</b>	<b>1</b>
1 Objectifs et démarche	3
2 Livrables et évaluation	7
<b>II Catalogue des tutoriels.</b>	<b>11</b>
<b>Comment utiliser ce catalogue ?</b>	<b>13</b>
1 Premiers pas	15
2 Interfaçage	25
3 Gestion du temps & Interruptions	39
4 Traitement/Asservissement	47
5 Communication	59
<b>Index</b>	<b>69</b>



# Calendrier

Semaine	Date	Horaire	Gr1	Gr2	Gr3.1 / 1/2
7	lun. 12 févr.	13h30		TP0	
	mar. 13 févr.	13h30	TP 0		
	mer. 14 févr.	13h30			TP 0
	jeu. 15 févr.	08h30	Pr 1 - Sujet		
8	lun. 19 févr.	VACANCES			
9	lun. 26 févr.	SPECIFIQUE / PIMS			
10	lun. 5 mars	13h30		Pr 1 - Sujet	
	mar. 6 mars	13h30	Pr2 - Planning		
	mer. 7 mars	13h30			Pr 1 - Sujet
11	lun. 12 mars	AUTOMATIQUE			
12	lun. 19 mars	13h30		Pr2 - Planning	
	mar. 20 mars	13h30	Pr3		
	mer. 21 mars	13h30			Pr2 - Planning
	jeu. 22 mars	08h30		Pr3	
13	lun. 26 mars	13h30		Pr4 - Sché Fonc	
	mar. 27 mars	13h30	Pr4 - Sché Fonc		
	mer. 28 mars	13h30			Pr3
	jeu. 29 mars	08h30			Pr4 - Sché Fonc
14	lun. 2 avr.	FERIE			
	mar. 3 avr.	13h30	Pr5 - EXAM		
	mer. 4 avr.	13h30			Pr5 - EXAM
	jeu. 5 avr.	08h30		Pr5 - EXAM	
15	lun. 9 avr.	13h30		Pr6 - EXAM	
	mar. 10 avr.	13h30	Pr6 - EXAM		
	mer. 11 avr.	13h30			Pr6 - EXAM
	jeu. 12 avr.	08h30	Pr7		
16	lun. 16 avr.	VACANCES			
17	lun. 23 avr.	VACANCES			
18	lun. 30 avr.	13h30		Pr7	
	mar. 1 mai	FERIE			
	mer. 2 mai	13h30			Pr7
	jeu. 3 mai	13h30	Pr8 - Résultats		
19	lun. 7 mai	13h30		Pr8 - Résultats	
	mar. 8 mai	FERIE			
	mer. 9 mai	13h30			Pr8 - Résultats
	jeu. 10 mai	FERIE			
20	jeu. 17 mai	Rendu Livrables Finaux			

Version du 20 mars 2018, il sera mis à jour sur le site du LEnsE : [hebergement.u-psud.fr/lense/](http://hebergement.u-psud.fr/lense/)



# Règles de fonctionnement du LEnsE

---

## Absences

---

La présence des étudiant·e·s à toutes les séances de travaux pratiques prévues à l'emploi du temps est obligatoire et impérative. En cas de difficulté majeure, **si un membre d'un binôme est toutefois absent, l'autre doit venir à la séance et faire le TP**. Et, en Optique, chacun des membres du binôme rendra un compte-rendu individuel.

**Absence excusée. Justificatif** Le justificatif d'absence doit être déposé au secrétariat, les élèves concerné·e·s doivent aussi prévenir directement les responsables du LEsE du motif de l'absence (à l'avance, si l'absence est prévisible).

**Absence excusée. Rattrapage** L'élève doit impérativement prendre contact avec les enseignant·e·s de TP pour étudier la possibilité de rattrapage (suivant la disponibilité des enseignant·e·s, du matériel et des salles). L'élève rattrape alors le TP et :

**En optique**, l'élève rédige un CR qui sera noté. S'il n'est pas possible de trouver une date de rattrapage suite à une impossibilité du service des TP, le TP ne sera ni rattrapé ni noté (la moyenne sera faite sur les notes restantes). Ce TP restera néanmoins au programme de l'examen et l'étudiant·e pourra être interrogé·e sur ce TP lors de l'examen de TP.

**En ETI et ProTIS**, la synthèse du thème concerné, rédigée par le binôme, devra contenir des résultats des deux séances individuelles (la séance normale et celle de rattrapage).

Si l'élève refuse la date de rattrapage proposée, il ou elle sera considéré-e comme absent-e non excusé-e.

**Absence non excusée** Toute absence non justifiée entraîne :

**En optique**, un zéro pour la séance et l'impossibilité de travailler sur ce TP avant la période de révision. En cas d'absences répétées, le responsable d'année interdira à l'étudiant-e de passer l'examen en fin d'année.

**En ETI et ProTIS**, un zéro pour la note de synthèse concernée.

---

### Retards

---

Aucun retard n'est acceptable et en cas de retard important (ou de retards fréquents) d'un-e étudiant-e, celui-ci ou celle-ci se verra refuser l'accès au laboratoire. Les conséquences en seront identiques à celles d'une absence non excusée (voir plus haut).

---

### Plagiats

---

Le plagiat est le fait de s'approprier un texte ou partie de texte, image, photo, données... réalisé par quelqu'un d'autre sans préciser qu'il ne s'agit pas de son travail personnel. On plagie quand on ne cite pas l'auteur des sources que l'on utilise. Exemples de plagiat :

- Copier textuellement un passage d'un livre ou d'une page Web sans le mettre entre guillemets et/ou sans en mentionner la source.
- Insérer dans un travail des images, des graphiques provenant de sources externes (hors énoncé du TP) sans en indiquer la provenance.
- Utiliser le travail d'un-e autre élève et le présenter comme le sien (et ce, même si cette personne a donné son accord!).
- Résumer l'idée originale d'un auteur en l'exprimant dans ses propres mots, mais en omettant d'en indiquer la source.
- Traduire partiellement ou totalement un texte sans en mentionner la provenance.



Tout binôme convaincu de plagiat dans un compte-rendu ou une synthèse de TP se verra attribuer la note de 0/20 à ce TP ou cette synthèse et encourt les sanctions disciplinaires prévues au règlement intérieur.

---

### Respect du matériel et des locaux

---

Le LEnsE met à votre disposition une très grande quantité de matériel scientifique.

Ces matériels sont très fragiles, sensibles à la poussière, aux traces de doigts, aux rayures, etc. Merci d'en prendre le plus grand soin.

**Il est donc formellement interdit d'apporter de la nourriture ou des boissons dans l'ensemble du service (couloirs compris).** Merci de veiller aussi à laisser les locaux particulièrement propres (si vos chaussures sont sales, retirez-les et laissez-les à l'entrée !)

Pour toute demande d'accès en dehors des séances de TP, vous devez impérativement (et à l'avance) vous adresser au responsable technique du LEnsE, Thierry AVIGNON ou à Cédric LEJEUNE (bureau S1.18).



**Première partie**

**Modalités**



# 1

# Objectifs et démarche

## Sommaire

---

1	Objectifs pédagogiques . . . . .	3
2	Organisation . . . . .	3
3	Déroulement des séances . . . . .	4
4	L'espace numérique de travail . . . . .	4

---

## 1 Objectifs pédagogiques

A l'issue de ce module, vous serez capables de :

- concevoir un dispositif d'interfaçage et de traitement du signal,
- mettre en œuvre un tel dispositif en associant un microcontrôleur sur carte de type `Nucleo` et des circuits électroniques de mise en forme.

## 2 Organisation

### 2.1 Objectif technique global

Les 9 séances sont consacrées à la réalisation d'un prototype fonctionnel et documenté d'un dispositif à microcontrôleur, par équipe de 4 élèves.

## 2.2 Démarche

**La séance 00** a pour objectif de réunir les élèves par équipe de 4 autour d'un projet de prototypage. Avant cela, une première étape est de découvrir la cible matérielle. La séance est découpée en 4 parties :

1. Découverte de l'outil de prototypage (2h - travail en binôme).
2. Formation des équipes et choix des sujets de projets (1h30 - travail en groupe).
3. Découpage fonctionnel du projet (1h00 - travail en équipe).

**Les séances 01 à 08** sont des créneaux de TP | projet à la carte selon une répartition et un rythme choisis par chaque équipe de 2 binômes.

## 3 Déroulement des séances

Pour vous guider dans la prise en main technique du microcontrôleur, un catalogue de modules de TP (tutoriels) vous est proposé (à partir de la page 13 de ce document), dans lequel vous pourrez piocher en fonction de vos besoins et /ou de votre curiosité.

Le nombre et le choix des modules sont laissés libres, il vous est cependant demandé de tenir à jour une grille personnelle de compétences indiquant les notions que vous maîtrisez.

L'objectif est de savoir mettre en œuvre, à l'issue de la séance 04, au moins une expérience de niveau ③, au choix. Un test individuel validera cette compétence au cours des séances 05 et 06.

## 4 L'espace numérique de travail

Il est indispensable d'utiliser un outil de travail collaboratif en ligne. Cette version moderne et interactive du « cahier de manipulation » gardera en particulier la trace de tous les programmes, circuits et mesures que vous aurez réalisés. Ce document reste un document interne à l'équipe. L'équipe enseignante s'assurera que vous disposez bien d'un tel outil à chaque séance et pourra vous aider à sa mise en œuvre.

Vous devrez donc organiser un espace de travail partagé (au choix) :

- sur le "nuage" de l'Institut d'Optique (<https://cloud.institutoptique.fr>),
- sur la plateforme Trello (<https://www.trello.com>),

- dans une sphère Whaller ,
- dans un espace Slack,
- dans un répertoire GoogleDrive,
- ou sur toute autre plateforme de travail collaboratif en ligne adaptée à un environnement professionnel.

Il est en particulier important que **chaque membre de l'équipe puisse accéder à l'ensemble des documents lors de chaque séance.**

L'accès aux documents est autorisé (voire indispensable) lors des tests individuels d'évaluation.





## 2

# Livrables et évaluation

## Sommaire

---

1	<b>Livrables de projet (par équipe)</b> . . . . .	7
2	<b>Validation</b> . . . . .	8
3	<b>Vue d'ensemble</b> . . . . .	10

---

## 1 Livrables de projet (par équipe)

Des livrables intermédiaires sont attendus à la fin de la séance 03, documents qui seront complétés en fin de projet par des livrables finaux.

### 1.1 Livrables intermédiaires

Par équipe, il est attendu à l'issue de la séance 03 :

- le cahier des charges précis, donnant plusieurs versions possibles.
- le schéma des différentes fonctions techniques réalisées par le prototype
- le plan de formation, argumenté, des deux binômes (1 page max)

### 1.2 Livrables finaux

A l'issue du projet, sont attendus :

1. Un dossier technique (pdf ou site web/blog/etc.) comprenant :
  - (a) la description de la problématique à laquelle le prototype répond.
  - (b) un schéma complet de l'architecture du prototype où on trouvera :
    - Les entrées-sorties,
    - Le découpage fonctionnel du projet (fonctions matérielles et logicielles),
  - (c) la nomenclature et le coût de votre projet
  - (d) un mode d'emploi succinct (moins de 2 pages). Sur ce mode d'emploi figurera des photos commentées du projet. On connaîtra ainsi tous les branchements nécessaires ainsi que le mode opératoire.
2. Une page d'analyse du planning effectivement tenu et de retour de votre expérience de travail de projet.
3. Tous les programmes (commentés) et les schémas électriques des circuits réalisés dans une archive zip.
4. Vous nous remettrez de plus un document présentant le projet sous forme soit soit d'un mini-site web, d'un poster, soit d'un montage vidéo. Le public auquel devra s'adresser ce document correspond aux élèves de 1ère année.

## 2 Validation

Les notes pondérées de ProTIS constituent la note de l'UE 8P-151-SCI ProTIS. La pondération détaillée est la suivante :

- 25 %** pour une note de séance individuelle. Au cours des 8 séances seront évalués par l'équipe enseignante :
- le soin apporté à la réalisation des prototypes (qualité du câblage, choix des instruments de mesure et leur réglage, qualité du code réalisé, lisibilité, modularité),
  - l'autonomie (conception : choix de la solution, des composants et des algorithmes, réalisation : capacité à dépanner, utilisation des tests **d'autoévaluation**) et l'utilisation efficace d'un espace de travail collaboratif,
  - le respect du cahier des charges.

**30 %** pour la note d'examen pratique individuel. Une échelle **indicative** de notation est la suivante :

0/20	Note attribuée en cas d'absence non justifiée
5/20	Pas de circuit ni de programme fonctionnel, même avec aide.
8/20	Une partie des fonctionnalités visées est réalisée, avec aide. Démarche incorrecte.
11/20	Une partie des fonctionnalités visées est réalisée, avec aide. Démarche correcte.
15/20	Les fonctionnalités (circuits et/ou programmes) visées ont été réalisées de manière autonome. Quelques erreurs dans la démarche.
20/20	L'ensemble des fonctionnalités visées a été réalisé de manière autonome dans le temps imparti. Démarche rigoureuse.

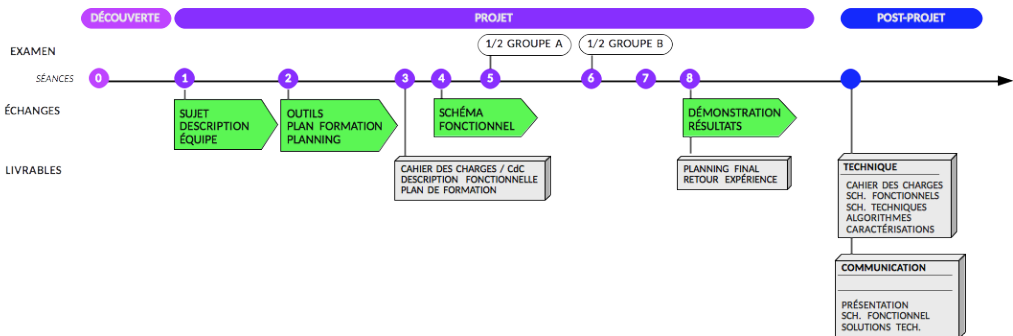
**15 %** pour la note des livrables intermédiaires (6 points pour le cahier des charges, 8 points pour le schéma fonctionnel et 6 points pour le plan de formation)

**30 %** pour la note des livrables finaux (6 points pour les documents techniques, 2 points pour l'analyse, 2 points pour les programmes et schémas et enfin 10 points pour la qualité du support de présentation)

## Points de pénalité

Des points de pénalité peuvent être attribués aux binômes n'ayant pas rangé leur poste de travail à l'issue des séances de TP | projet.

### 3 Vue d'ensemble



ÉVALUATION DU MODULE			
SÉANCES	LIVR. INTERMÉDIAIRES	EXAMEN INDIVIDUEL	LIVRABLES FINAUX
<b>SOIN 25 %</b> - Qualité câblage / prog. - Choix instruments <b>AUTONOMIE</b> - Conception : solution, algo - Réalisation : débogage <b>RESULTATS</b> - Respect CdC	<b>15 %</b> <b>CAHIER DES CHARGES (6)</b> - Fonctionnalité principale <b>DESC. FONCTIONNELLE (8)</b> - Schéma des fonctions et liens entre elles <b>PLAN FORMATION (6)</b> - Argumentaire choix tuto.	<b>30 %</b> 00/20 Absence 05/20 Pas de circuit ou programme fonctionnel 08/20 Démarche incorrecte 11/20 Démarche correcte avec quelques fonctions 15/20 Réalisation autonome 20/20 Démarche rigoureuse	<b>30 %</b> <b>ARCHIVES (2)</b> - Programmes Commentés - Schémas électriques <b>ANALYSE / REX (2)</b> <b>RAPPORT TECHNIQUE (6)</b> - Schémas électriques <b>COMMUNICATION (10)</b> - Site Web / Vidéo / Poster

FIGURE 2.1 – Une vue d'ensemble du déroulement et des évaluations de ProTIS

**Deuxième partie**

**Catalogue des tutoriels.**



# Comment utiliser ce catalogue ?

Les tutoriels sont disponibles sur le site :

<http://hebergement.u-psud.fr/villemejane/eiti/index.php/protis/protis-auto-formation/>

Ils sont organisés en 5 grandes catégories et classés selon leur niveau de complexité (voir figure 2).

CATÉGORIES				
	NIVEAUX			
	1	2	3	4
PREMIERS PAS AVEC NUCLÉO				
INTERFAÇAGE				
GESTION DU TEMPS / INTERRUPTIONS				
TRAITEMENT / ASSERVISSEMENT				
COMMUNICATION				

FIGURE 2 – Les catégories de tutoriels et leurs niveaux

Vous ne trouverez ici que la liste des tutoriels des 3 premières catégories : Premier Pas, Interfaçage et Gestion du temps. Le catalogue des deux autres catégories (Traitement/Asservissement et Communication) vous sera présenté dans un document séparé.

Ce document se veut une aide pour construire votre plan de forma-

tion en fonction des besoins de votre projet et/ou de votre curiosité.

Vous trouverez ainsi pour chaque tutoriel :

- la description de ses objectifs,
- un bref test d'autoévaluation sous forme de QCM dont les solutions sont disponibles dans la salle de TP

Enfin, vous trouverez un index des termes techniques à la fin de ce document.



# Catégorie 1

## Premiers pas

### Sommaire

---

1	Créer un projet sous MBED . . . . .	15
2	Tester un premier programme sur Nucléo . . .	16
3	Piloter une LED . . . . .	16
4	Déboguer son programme et utiliser l'affichage série . . . . .	16
5	Différencier un microcontrôleur d'un micro-ordinateur	17
6	Trouver des informations dans la documenta- tion . . . . .	17
	Pour se tester . . . . .	18

---

## 1 Créer un projet sous MBED

Niveau 0

Pré-requis :-

**A l'issue de ce tutoriel, vous saurez :**

- Créer un compte sur MBED Compiler
- Associer une carte de développement de type Nucléo
- Créer un projet pour cette carte de développement

**Mots clés :** Interface MBED

---

## 2 Tester un premier programme sur Nucléo

Niveau ①

Pré-requis : Créer un projet sous MBED

A l'issue de ce tutoriel, vous saurez :

- Reconnaître les différents connecteurs de la carte
  - Compiler et téléverser un programme sur la carte
  - Tester une première application
- 

## 3 Piloter une LED

Niveau ①

Pré-requis : Tester un premier programme sur Nucléo

A l'issue de ce tutoriel, vous saurez :

- Utiliser une sortie numérique pour piloter un composant externe
- Câbler une LED pour afficher une information binaire

Mots clés : Sortie numérique / LED

---

## 4 Débugger son programme et utiliser l'affichage série

Niveau ①

Pré-requis : Piloter une LED

A l'issue de ce tutoriel, vous saurez :

- Utiliser une sortie numérique pour déboguer un code source
- Afficher l'état d'une variable via la console série

Mots clés : Débogage , Oscilloscope, Console Série

---

## 5 Différencier un microcontrôleur d'un micro-ordinateur

Niveau 0

Pré-requis : -

**A l'issue de ce tutoriel, vous saurez :**

- Trouver l'information sur les différentes fonctions disponibles en ligne.
- Importer de nouvelles bibliothèques de fonctions.
- Décrire la structure simplifiée d'un microcontrôleur.

**Mots clés :** Microcontrôleurs et microprocesseurs

---

## 6 Trouver des informations dans la documentation

Niveau 1 **Pré-requis :** Tester un premier programme sur Nucléo

**A l'issue de ce tutoriel, vous saurez :**

- Décrire la structure d'un microcontrôleur.
- Définir ses performances et ses limites.
- Ecrire un code structuré pour un microcontrôleur.

**Mot clé :** Documentation

---

## Pour se tester

### 1 Créer un projet sous MBED

<p><b>1.</b> L'interface de développement MBED</p>	<p>(a) est un logiciel payant</p> <p>(b) est accessible en ligne</p> <p>(c) est installé sur tous les ordinateurs de l'IOGS</p>
<p><b>2.</b> Ce logiciel</p>	<p>(a) est une variante des compilateurs de C/C++ habituels</p> <p>(b) a une fonction de compilateur C/C++ nécessite que l'on précise la cible matérielle sur laquelle le programme va fonctionner</p> <p>(c) matérielle sur laquelle le programme va fonctionner</p>

## 2 Tester un premier programme sur Nucléo

<p><b>1.</b> La LED câblée sur la carte et appelée LED1</p>	<ul style="list-style-type: none"> <li>(a) est connectée à la broche D13 du microcontrôleur</li> <li>(b) n'est connectée à aucune broche du microcontrôleur</li> <li>(c) est connectée à la broche PA_5 du microcontrôleur</li> </ul>
<p><b>2.</b> Le programme « Blinky LED »</p>	<ul style="list-style-type: none"> <li>(a) permet d'afficher « Hello World! » sur l'ordinateur</li> <li>(b) fait clignoter la LED LED1 de la carte</li> <li>(c) allume la LED LED1 pendant 2s puis l'éteint.</li> </ul>
<p><b>3.</b> Remplacer la ligne 8 du programme « Blinky LED » par : wait (2) a pour conséquence :</p>	<ul style="list-style-type: none"> <li>(a) d'augmenter la durée pendant laquelle la LED LED1 est éclairée</li> <li>(b) de réduire la durée pendant laquelle la LED LED1 est éteinte</li> <li>(c) de modifier la fréquence de clignotement de la LED LED1.</li> </ul>
<p><b>4.</b> Un programme pour une application embarquée :</p>	<ul style="list-style-type: none"> <li>(a) s'écrit toujours en langage C</li> <li>(b) a besoin d'un ordinateur pour être exécuté</li> <li>(c) est téléversé après compilation sur le microcontrôleur pour fonctionner en autonomie</li> <li>(d) possède une boucle infinie qui ne s'arrête jamais</li> </ul>

### 3 Piloter une LED

<p><b>1. La ligne :</b>  <code>include "mbed.h"</code></p>	<p>permet d'importer une bibliothèque  <b>(a)</b> décrivant les entrées et les sorties de la carte Nucleo          contient une ligne du type  <b>(b)</b> <code>DigitalOut myled(D13);</code> permettant de définir quelle broche est utilisée en sortie.  <b>(c)</b> doit être présente dans l'en-tête de tous les programmes</p>
<p><b>2. La ligne de code</b>  <code>DigitalOut toto(D10);</code></p>	<p>est incorrecte, on doit écrire  <b>(a)</b> <code>DigitalOut toto(D10,1);</code> ou <code>DigitalOut toto(D10,0);</code>  <b>(b)</b> permet de déclarer la broche D10 comme une sortie numérique  <b>(c)</b> permet de déclarer la broche <code>toto</code> comme une sortie numérique</p>
<p><b>3. La tension d'alimentation de la carte Nucleo est égale à :</b></p>	<p><b>(a)</b> 5 V  <b>(b)</b> 3.3 V  <b>(c)</b> 5 A</p>
<p><b>4. Le montage 1 du tutoriel</b></p>	<p><b>(a)</b> doit être préféré au montage 2  <b>(b)</b> permet d'allumer la LED quand une tension de 3.3 V est appliquée à la broche D10  <b>(c)</b> permet d'allumer la LED quand une tension de 0 V est appliquée à la broche D10</p>

## 6. TROUVER DES INFORMATIONS DANS LA DOCUMENTATION 21

<b>5.</b> Le montage 2 du tutoriel	<b>(a)</b> doit être préféré au montage 1  comprend une résistance $R_{LED}$ que <b>(b)</b> l'on peut calculer par la même formule que lors du montage 1.  permet d'allumer la LED quand une <b>(c)</b> tension de 0V est appliquée à la broche D10
Dans le montage 1 du tutoriel, on souhaite limiter le courant à 20 mA. La LED possède un seuil de 2.3 V.	
<b>6.</b> On doit donc choisir une valeur $R_{LED}$	<b>(a)</b> inférieure à 10 k $\Omega$ <b>(b)</b> supérieure à 10 $\Omega$ <b>(c)</b> supérieure à 50 $\Omega$

## 4 Débugger son programme et utiliser l'affichage série

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.	
<b>1.</b> Pour cela il faut ajouter des commandes du type :	<b>(a)</b> <code>yourtest = 1; avant le code des instructions et yourtest = 0; après.</code> <b>(b)</b> <code>yourtest = 0; avant le code des instructions et yourtest = 1; après.</code> <b>(c)</b> <code>char c = 300;</code> à la suite du programme <code>DigitalOut yourtest(PA_4, 0);</code> <b>(d)</b> dans la partie initialisation du programme

<b>2.</b> Connaître la valeur d'une variable entière interne au microcontrôleur, à un instant donné :	<b>(a)</b> n'est pas possible. <b>(b)</b> est possible en utilisant un ordinateur relié au microcontrôleur par une liaison série. <b>(c)</b> est réalisable en utilisant un oscilloscope.
<b>3.</b> Le logiciel TeraTerm :	<b>(a)</b> établit une communication de protocole RS232 côté microcontrôleur. <b>(b)</b> établit une communication de protocole RS232 côté ordinateur. <b>(c)</b> est conçu pour fonctionner avec la carte Nucleo seulement.



## 5 Différencier un microcontrôleur d'un micro-ordinateur

<p><b>1.</b> Les éléments indispensables au bon fonctionnement d'un microcontrôleur sont :</p>	<p>(a) des espaces mémoires RAM/ROM</p> <p>(b) des entrées analogiques</p> <p>(c) un gestionnaire d'interface graphique</p>
<p><b>2.</b> Un microprocesseur :</p>	<p>(a) équipe les systèmes embarqués des automobiles</p> <p>(b) est un calculateur généraliste</p> <p>(c) permet d'exécuter uniquement des programmes en C est capable d'exécuter 2000 instructions élémentaires différentes (Intel i7)</p> <p>(d) est capable d'exécuter 2000 instructions élémentaires différentes (Intel i7)</p>
<p><b>3.</b> Un microcontrôleur :</p>	<p>(a) intègre tous les éléments nécessaires à l'exécution d'une tâche spécifique</p> <p>(b) possède des ressources matérielles illimitées</p> <p>(c) possède systématiquement une unité de calcul en virgule flottante</p> <p>(d) contient une zone de mémoire de données</p>
<p><b>4.</b> Les microcontrôleurs :</p>	<p>(a) ne peuvent être programmés qu'en langage machine</p> <p>(b) permettent de répondre rapidement à des événements extérieurs</p> <p>(c) intègrent une matrice de portes logiques</p>
<p><b>5.</b> Un programme pour système embarqué :</p>	<p>(a) doit s'arrêter systématiquement doit en permanence scruter les entrées</p> <p>(b) et mettre à jour les sorties en conséquence</p> <p>(c) doit pouvoir répondre dans un temps limité à toute sollicitation extérieure</p>

## 6 Trouver des informations dans la documentation

<p><b>1.</b> Les pages de documentation des fonctions présentes dans l'API de MBED</p>	<p>(a) ne sont pas utiles pour apprendre à utiliser les fonctions</p> <p>(b) sont écrites en anglais</p> <p>(c) sont disponibles en ligne</p> <p>(d) font uniquement la liste des classes</p>
<p><b>2.</b> Sous MBED Compiler, je peux :</p>	<p>(a) importer des bibliothèques que j'ai moi-même réalisées</p> <p>(b) changer de carte de développement, sans avoir à modifier le code</p> <p>(c) partager mon code avec d'autres</p> <p>(d) vendre mon code</p>
<p><b>3.</b> La bibliothèque RN41 :</p>	<p>(a) contient 1 classe</p> <p>(b) contient 2 classes</p> <p>(c) contient la fonction <code>getLine()</code></p> <p>(d) ne contient aucune fonction</p>

# Catégorie 2

# Interfaçage

## Sommaire

---

1	Récupérer une information numérique . . . . .	25
2	Câbler un bouton-poussoir . . . . .	26
3	Utiliser un écran LCD . . . . .	26
4	Récupérer un signal analogique . . . . .	26
5	Régler la luminosité d'une LED . . . . .	27
6	Générer une tension analogique . . . . .	27
7	Contrôler un mouvement angulaire à l'aide s'un servomoteur . . . . .	27
8	Faire varier la vitesse d'un moteur à CC . . . . .	28
9	Connecter un capteur sonore et un haut-parleur au microcontrôleur . . . . .	28
	Pour se tester . . . . .	29

---

## 1 Récupérer une information numérique

**Niveau ①**      **Pré-requis :** Piloter une LED  
Débuguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Configurer une broche du microcontrôleur en entrée numérique.
- Lire l'état d'une entrée numérique.

**Mots clés :** Entrée numérique

---

## 2 Câbler un bouton-poussoir

**Niveau ①** **Pré-requis :** Récupérer une information numérique

**A l'issue de ce tutoriel, vous saurez :**

- Réaliser le circuit de câblage d'un bouton-poussoir connecté à une entrée numérique

**Mots clés :** Bouton poussoir

---

## 3 Utiliser un écran LCD

**Niveau ②** **Pré-requis :** Débuguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Connecter un écran LCD de type EA DOG 163 à la carte Nucleo
- Afficher des caractères ou des chaînes de caractères sur cet écran

**Mots clés :** Écran LCD , Protocole SPI , Affichage

---

## 4 Récupérer un signal analogique

**Niveau ①** **Pré-requis :** Débuguer son programme et utiliser l'affichage série  
Piloter une LED

**A l'issue de ce tutoriel, vous saurez :**

- Configurer et utiliser les entrées analogiques du microcontrôleur
- Mesurer la valeur d'une tension appliquée à une broche d'entrée

**Mots clés :** Entrée analogique

---

## 5 Régler la luminosité d'une LED

**Niveau** ②

**Pré-requis :** Déboguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Expliquer le principe de la modulation en largeur d'impulsion (ou *Pulse Width Modulation* : *PWM*).
- Configurer le microcontrôleur pour piloter un signal de ce type.
- Utiliser un tel signal pour piloter l'intensité lumineuse d'une LED.

**Mot clé :** Sortie PWM, LED

---

## 6 Générer une tension analogique

**Niveau** ①

**Pré-requis :** Déboguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Configurer et utiliser les sorties analogiques du microcontrôleur
- Générer une tension variable au cours du temps (signal analogique) à l'aide du microcontrôleur

**Mots clés :** Sortie analogique

---

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

**Niveau** ②

**Pré-requis :** Régler la luminosité d'une LED

**A l'issue de ce tutoriel, vous saurez :**

- Définir le type d'application typique d'un servo-moteur.
- Définir quels signaux envoyer à un servomoteur pour le piloter.
- Utiliser un microcontrôleur pour le contrôler.

**Mot clé :** Sortie PWM, Servomoteur

---

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

**Niveau** ③

**Pré-requis :** Régler la luminosité d'une LED

**A l'issue de ce tutoriel, vous saurez :**

- Définir le type d'application typique d'un moteur à courant continu
- Définir quels signaux envoyer à un moteur à courant continu pour le piloter.
- Utiliser un microcontrôleur pour le contrôler.

**Mot clé :** Sortie PWM, Servomoteur

---

## 9 Connecter un capteur sonore et un haut-parleur au microcontrôleur

**Niveau** ①

**Pré-requis** Débugger son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Câbler le circuit d'un microphone afin d'enregistrer le signal sur le microcontrôleur
- Câbler le circuit d'un haut-parleur afin de le piloter avec le microcontrôleur

**Mots clés :** Microphone

---

## Pour se tester

### 1 Récupérer une information numérique

<p><b>1.</b> Pour récupérer une information numérique sur une entrée, il faut :</p>	<p>configurer la broche associée en entrée en la déclarant via l'instruction <code>DigitalIn</code></p> <p>(b) s'assurer que la tension appliquée sur cette broche soit égale à 0 V ou 5 V</p> <p>(c) lui associer un nom de variable incluant le préfixe <code>in**</code></p>
<p><b>2.</b> Une entrée numérique est bien adaptée pour :</p>	<p>(a) réaliser le filtrage numérique d'un signal</p> <p>(b) tester l'état d'un interrupteur fin de course</p> <p>(c) récupérer l'information issue d'un comparateur à ALI</p> <p>(d) récupérer l'information d'un capteur de température</p>
<p><b>3.</b> Le bouton poussoir <code>USER_BUTTON</code> :</p>	<p>(a) est connecté physiquement à l'entrée numérique <code>PC_13</code></p> <p>(b) est connecté au niveau haut (3,3 V) par une résistance de 4,7 k<math>\Omega</math></p> <p>(c) est connecté au niveau bas (0 V) par une résistance de 4,7 k<math>\Omega</math></p>

### 2 Câbler un bouton-poussoir

<p><b>1.</b> Dans le montage proposé dans le tutoriel, quelle est la valeur de la tension appliquée à la broche <code>D7</code> au repos ?</p>	<p>(a) 0</p> <p>(b) 1</p> <p>(c) 0 V</p> <p>(d) 5 V</p>
--	---

<p><b>2.</b> 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 ?</p>	<p>(a) 0</p> <p>(b) 1</p> <p>(c) 0 V</p> <p>(d) 5 V</p>
<p><b>3.</b> Un bouton poussoir :</p>	<p>doit être câblé sur l'entrée d'un ADC</p> <p>(a) car la tension qu'il délivre est comprise entre 0 V et 3,3 V</p> <p>(b) est un capteur de type tout ou rien (TOR)</p> <p>(c) est un composant électronique passif</p>
<p><b>4.</b> Lors de l'appui sur le bouton poussoir BP :</p>	<p>(a) l'entrée numérique D7 voit un niveau logique haut</p> <p>(b) l'entrée numérique D7 voit un niveau logique bas</p> <p>(c) le courant qui circule au travers de la résistance <math>R_{BP}</math> est égale à 0,33 mA</p> <p>(d) le courant qui circule au travers de la résistance <math>R_{BP}</math> est nul</p>
<p><b>5.</b> La ligne de code</p> <pre>a = mon_bouton.read();</pre>	<p>(a) est automatiquement appelée lors de l'appui sur le bouton poussoir</p> <p>(b) permet de lire une entrée numérique et de stocker sa valeur dans la variable a</p> <p>(c) doit être écrite à chaque endroit du programme où l'état du bouton poussoir doit être testé</p> <p>(d) peut précéder l'instruction <code>DigitalIn mon_bouton(D7);</code></p>



### 3 Utiliser un écran LCD

<p><b>1.</b> L'écran LCD est relié au microcontrôleur :</p>	<ul style="list-style-type: none"> <li>(a) par l'intermédiaire de 3 fils (et l'alimentation)</li> <li>(b) par l'intermédiaire d'une liaison série</li> <li>(c) par l'intermédiaire d'une liaison Bluetooth</li> <li>(d) par l'intermédiaire d'un protocole parallèle</li> </ul>
<p><b>2.</b> Lorsqu'on exécute la ligne suivante, <code>spi.write('a');</code>, de l'exemple Ecriture d'un caractère du tutoriel :</p>	<ul style="list-style-type: none"> <li>(a) la valeur de la variable <code>a</code> est affichée sur l'écran LCD</li> <li>(b) la série binaire <code>01100001</code> est transmise sur la broche MOSI de la carte Nucleo</li> <li>(c) le caractère 'a' est affiché à l'endroit où se trouve le curseur sur l'écran LCD</li> <li>(d) l'écran LCD est rempli du caractère 'a'</li> </ul>
<p><b>3.</b> Lorsqu'on exécute la ligne suivante :</p> <pre>LCD_DOG_setPosition(2, 3);</pre>	<ul style="list-style-type: none"> <li>(a) la série binaire <code>10010011</code> est transmise sur la broche MOSI de la carte Nucléo</li> <li>(b) le curseur de l'écran LCD se place à la ligne 3 et à la colonne 2</li> <li>(c) le curseur de l'écran LCD se place à la ligne 2 et à la colonne 3</li> <li>(d) l'écran LCD est effacé</li> </ul>
<p><b>4.</b> Dans l'exemple complet du tutoriel :</p>	<ul style="list-style-type: none"> <li>(a) la série binaire <code>10010011</code> est transmise sur la broche MOSI de la carte Nucléo</li> <li>(b) le curseur de l'écran LCD se place à la ligne 3 et à la colonne 2</li> <li>(c) le curseur de l'écran LCD se place à la ligne 2 et à la colonne 3</li> <li>(d) l'écran LCD est effacé</li> </ul>

## 4 Récupérer un signal analogique

<p><b>1.</b> Les convertisseurs analogique-numérique du microcontrôleur de la carte <code>Nucleo</code> :</p>	<p>quantifie la tension appliquée à une</p> <p><b>(a)</b> entrée analogique avec un quantum de <math>3,3/(2^{12} - 1) \text{ V}</math></p> <p>quantifie la tension appliquée à une</p> <p><b>(b)</b> entrée analogique avec un quantum de <math>3,3/(2^{16} - 1) \text{ V}</math></p> <p>échantillonnent la tension appliquée à</p> <p><b>(c)</b> une entrée analogique à la fréquence maximale de 5 MHz</p> <p><b>(d)</b> peuvent se voir appliquer une tension sinusoïdale d'amplitude 3,3 V</p>
<p><b>2.</b> Les entrées analogiques :</p>	<p><b>(a)</b> sont au nombre de 6 sur le connecteur arduino de la carte <code>Nucleo</code></p> <p>sont reliées à 3 ADC 16bits fonctionnant à 5 MHz de fréquence d'échantillonnage maximale</p> <p><b>(b)</b></p> <p><b>(c)</b> sont adaptées à la lecture de l'état d'un bouton poussoir</p> <p><b>(d)</b> acceptent des tensions comprises entre -3,3 V et 3,3 V</p>
<p><b>3.</b> La fonction <code>analog_in.read_u16()</code></p>	<p><b>(a)</b> renvoie un entier compris entre 0 et 65536</p> <p><b>(b)</b> renvoie un entier compris entre 0 et <math>2^{16} - 16</math></p> <p>permet de récupérer le résultat de la</p> <p><b>(c)</b> conversion analogique-numérique de l'entrée <code>u16</code></p>

## 5 Régler la luminosité d'une LED

<p><b>1.</b> La principale variable qui détermine la luminosité d'une LED perçue par un œil humain est</p>	<p>(a) La tension moyenne à ses bornes.</p> <p>(b) La puissance moyenne qu'elle consomme.</p> <p>(c) Le courant moyen qui la traverse.</p>
<p><b>2.</b> La modulation par largeur d'impulsion est la méthode utilisée pour faire varier la luminosité d'une LED</p>	<p>Car il suffit de moduler à une fréquence supérieure à 40 Hz pour mettre à profit la persistance rétinienne</p> <p>(a) Car il est plus facile de générer une tension de ce type qu'une tension analogique continue avec un microcontrôleur</p> <p>(b) Car on s'assure ainsi de la proportionnalité de la luminosité moyenne obtenue avec la commande</p> <p>(c)</p>
<p><b>3.</b> La syntaxe d'utilisation d'une sortie modulée en largeur d'impulsion nécessite des instructions du type</p>	<p>PwmOut Ma_Led_Pwm (D3) dans l'en-tête avant le main et Ma_Led_Pwm.period(RC) dans la boucle</p> <p>(a)</p> <p>PwmOut (D3) dans l'en-tête avant le main et PwmOut.write(value) dans la boucle</p> <p>(b)</p> <p>Ma_Led_Pwm.pulsewidth_us(Tpulse) ou Ma_Led_Pwm.pulsewidth(Tpulse)</p> <p>(c)</p>

## 6 Générer une tension analogique

<p><b>1.</b> La carte Nucleo a un système de conversion DAC sur 12 bits qui permet d'obtenir une tension</p>	<p>réglable entre 0 et 3,3 V par pas de</p> <p>(a) <math>\frac{3,3V}{1024}</math> sur les trois broches PA_5, PA_6 et PA_7</p> <p>(b) réglable entre 0 et 1 V au pas de 240 <math>\mu V</math> sur la seule broche PA_5.</p> <p>(c) réglable entre 0 et 3,3 V par pas de 240 <math>\mu V</math> sur la seule broche PA_5</p> <p>(d) de rapport cyclique variable entre 0 à 4095 sur la broche PA_4</p>
--	--

<p><b>2.</b> L'écriture du mot de 12 bits du DAC dans un mot de 16 bits se fait</p>	<p>En commençant à gauche par le MSB</p> <p>(a) à l'aide d'un décalage quadruple c'est à dire une commande du type «4</p> <p>(b) Directement, sans décalage.</p> <p>En commençant à gauche par le MSB et en remplissant les 4 LSB de 1 avec une instruction du type</p> <p>DAC = 0x10*a+0x0F</p>
<p><b>3.</b> La mise en œuvre logicielle du DAC nécessite des instructions du type :</p>	<p> AnalogOut (PA_5) dans</p> <p>(a) l'en-tête avant le main et AnalogOut.write (value) dans la boucle</p> <p> AnalogOut MyVar (PA_5) dans</p> <p>(b) l'en-tête avant le main et AnalogOut.write (MyVar (value)) dans la boucle</p> <p> AnalogOut MyVar (PA_5) dans</p> <p>(c) l'en-tête avant le main et MyVar.write (value) dans la boucle</p>

## 7 Contrôler un mouvement angulaire à l'aide s'un servomoteur

<p><b>1.</b> Appliquer une tension à un servomoteur standard sous forme d'impulsions de période 20 ms a pour conséquence :</p>	<p>de rendre sa vitesse de rotation proportionnelle au rapport cyclique des impulsions</p> <p>(a)</p> <p>de modifier sa position angulaire</p> <p>(b) lorsque le rapport cyclique est compris entre 0 et 100%</p> <p>de modifier sa position angulaire entre</p> <p>(c) <math>-90 \text{ deg}</math> à <math>+90 \text{ deg}</math>, selon le rapport cyclique, selon une fonction affine</p>
<p><b>2.</b> Parmi les propositions suivantes, lesquelles sont correctes ?</p>	<p>Un servomoteur standard doit être</p> <p>(a) alimenté à l'aide d'une alimentation continue de 5 à 6 V</p> <p>Le contrôle de l'angle d'un servomoteur standard est fait à l'aide d'une tension continue variable entre 0 et 5 V</p> <p>(b)</p> <p>Le signal de commande doit avoir une période de 1 ms</p> <p>(c)</p> <p>Le signal de commande doit avoir un rapport cyclique compris entre 0 et 0.1, pour une période de 20 ms.</p> <p>(d)</p>
<p><b>3.</b> Parmi les propositions suivantes, lesquelles sont correctes ?</p>	<p>Lorsqu'on supprime l'alimentation</p> <p>(a) d'un servomoteur, il reste dans sa position</p> <p>Lorsqu'on supprime le signal de commande d'un servomoteur, il reste dans sa position</p> <p>(b)</p> <p>Lorsqu'on applique un signal de commande de rapport cyclique de 0.15, le servomoteur revient à un angle de <math>0^\circ</math></p> <p>(c)</p>

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

<p><b>1.</b> Le principe de la modulation de largeur d'impulsion :</p>	<p>(a) ne peut pas s'appliquer à un moteur à CC sans filtrage</p> <p>(b) peut s'appliquer à un moteur à CC du fait de son inertie</p> <p>(c) peut directement être appliqué via la sortie du microcontrôleur au moteur à CC</p> <p>(d) nécessite un transistor (ou autre circuit) qui "booste" la commande PWM</p>
<p><b>2.</b> Dans le cas de la Commande réversible en sens de rotation du tutoriel, quelle(s) proposition(s) est (sont) correcte(s) ?</p>	<p>Commander T1 et T3 en même temps</p> <p>(a) (sans commander T2 et T4) permet d'arrêter le moteur</p> <p>(b) Commander T1 et T2 en même temps n'a aucun effet sur le montage</p> <p>(c) Ce type de commande nécessite 2 signaux PWM indépendants</p> <p>(d) Ce type de commande nécessite 2 signaux PWM complémentaires</p>

## 9 Connecter un capteur sonore et un haut-parleur au microcontrôleur

<p><b>1.</b> Le signal électrique issu d'un microphone :</p>	<p>(a) est directement applicable sur une entrée analogique de la carte Nucleo</p> <p>doit être amplifié pour atteindre les</p> <p>(b) 3,3 V de tension d'entrée maximale de l'ADC</p> <p>est à valeur moyenne nulle et doit donc</p> <p>(c) être décalé de 1,65 V pour s'adapter à l'ADC</p>
--	---

## 9. CONNECTER UN CAPTEUR SONORE ET UN HAUT-PARLEUR AU MICR

<p><b>2.</b> Sur le schéma de câblage proposé dans le tutoriel :</p>	<p>l'ensemble <math>C = 10 \mu\text{F}; R = 47 \text{k}\Omega</math> est placé en amont de l'échantillonneur et constitue donc un filtre anti-repliement</p> <p>le filtre <math>R = 1,5 \text{k}\Omega; C = 4,7 \text{nF}</math> a une</p> <p>(a) fréquence de coupure compatible avec le signal sonore</p> <p>(b) la tension de sortie du CNA (DAC) est à valeur moyenne nulle.</p> <p>(c) la tension <math>V_s</math> peut directement être appliquée à un haut parleur</p> <p>(d)</p>
<p><b>3.</b> Le code de test fourni :</p>	<p>peut servir sans modification à traiter le signal issu de l'enregistrement d'une chauve souris</p> <p>(a)</p> <p>est adapté au filtrage numérique de tout signal audio car la fréquence d'échantillonnage de 33 kHz est supérieure à 20 kHz</p> <p>(b)</p> <p>gère la fréquence d'échantillonnage au travers d'une interruption associée à un timer</p> <p>(c)</p> <p>peut servir de point de départ à la mise en œuvre d'un filtre numérique</p> <p>(d)</p>





# Catégorie 3

## Gestion du temps & Interruptions

### Sommaire

---

1	Faire une action à intervalle régulier . . . . .	39
2	Faire des actions à intervalle régulier . . . . .	40
3	Générer un signal d'horloge sans GBF . . . . .	40
4	Faire une action instantanément suite à un événement externe . . . . .	40
5	Faire une action périodique à une fréquence précise . . . . .	41
	Pour se tester . . . . .	42

---

### 1 Faire une action à intervalle régulier

**Niveau ②**      **Pré-requis :** Débuguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre la gestion du temps par le `Timer` matériel du microcontrôleur

**Mots clés :** Timer, Interruptions

---

## 2 Faire des actions à intervalle régulier

**Niveau ②** **Pré-requis :** Faire une action à intervalle régulier

**A l'issue de ce tutoriel, vous saurez :**

- Programmer une gestion du temps logicielle sur le microcontrôleur

**Mots clés :** Timer, Modes monostable et astable

---

## 3 Générer un signal d'horloge sans GBF

**Niveau ②** **Pré-requis :** Déboguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Réaliser un circuit oscillateur pour générer un signal d'horloge à une fréquence particulière.
- Mettre en oeuvre un quartz pour générer un signal d'horloge à une fréquence précise.

**Mots clés :** Horloge , Circuit oscillateur , Quartz

---

## 4 Faire une action instantanément suite à un événement externe

**Niveau ②** **Pré-requis :** Récupérer une information numérique

## 5. FAIRE UNE ACTION PÉRIODIQUE À UNE FRÉQUENCE PRÉCISE<sup>41</sup>

### A l'issue de ce tutoriel, vous saurez :

- Motiver la présence de modes d'interruption externe sur un microcontrôleur
- Configurer leur fonctionnement sur un microcontrôleur STM32

**Mots clés :** Interruptions

---

## 5 Faire une action périodique à une fréquence précise

**Niveau** ③

**Pré-requis :**

Générer un signal d'horloge sans GBF  
Faire une action instantanément suite à un événement externe

### A l'issue de ce tutoriel, vous saurez :

- utiliser un oscillateur externe combiné à une interruption du microcontrôleur

**Mots clés :** Interruptions , Circuit oscillateur

---

## Pour se tester

### 1 Faire une action à intervalle régulier

<p><b>1.</b> Qu'appelle-t-on timer ?</p>	<p>(a) l'horloge interne du microcontrôleur un module générant une interruption</p> <p>(b) au bout d'un certain nombre de coups d'horloge</p> <p>(c) Un oscillateur à une fréquence très précise</p>
<p>On considère le code suivant :</p> <pre>Ticker toggle_led_ticker; void toggle_led(void); int main() { toggle_led_ticker.attach(&amp;toggle_led, 1); while (1) {} }</pre>	
<p><b>2.</b> toggle_led désigne :</p>	<p>(a) une fonction</p> <p>(b) un pointeur</p> <p>(c) un signal d'interruption</p>
<p><b>3.</b> Dans ce code, la routine d'interruption est appelée</p>	<p>(a) à chaque front montant de l'horloge interne</p> <p>(b) toutes les secondes</p> <p>(c) toutes les millisecondes</p> <p>(d) jamais</p>

## 2 Faire des actions à intervalle régulier

<p><b>1.</b> Quelle est la proposition correcte ?</p>	<p>(a) Un timer logiciel peut fonctionner sans timer matériel</p> <p>(b) Un timer matériel peut fonctionner sans timer logiciel.</p>
<p><b>2.</b> Dans les exemples de code de ce tutoriel, <code>TIMER1</code> est :</p>	<p>(a) une variable</p> <p>(b) une fonction</p> <p>(c) une constante symbolique</p>
<p><b>3.</b> En ayant déclaré <code>InterruptIn bp_int (PA_8)</code>, que faut-il mettre dans le main pour appeler la fonction <code>routine_bp</code> à chaque front descendant de la broche <code>PA_8</code> ?</p>	<p>(a) <code>bp_int.fall(routine_bp);</code></p> <p>(b) <code>PA_8.fall(routine_bp);</code></p> <p>(c) <code>bp_int.fall(&amp;routine_bp);</code></p> <p>(d) <code>if (bp_int.fall) {routine_bp();}</code></p>

### 3 Générer un signal d'horloge sans GBF

<p><b>1. Le composant NE555 :</b></p>	<p>(a) permet de générer des signaux triangulaires</p> <p>(b) permet de générer des signaux rectangulaires</p> <p>(c) permet de générer des signaux rectangulaires de rapport cyclique inférieur à 50%</p> <p>(d) permet de générer des signaux de fréquence de 3 à 10 MHz</p>
<p><b>2. Lorsque le composant NE555 est en mode monostable :</b></p>	<p>(a) un front descendant sur l'entrée TRIGGER déclenche le passage à '1' de la sortie</p> <p>(b) le temps haut du signal de sortie ne dépend pas du temps à l'état bas de la broche TRIGGER</p> <p>(c) la plus courte impulsion est de 200 ns sur sa sortie</p> <p>(d) 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</p>
<p><b>3. Un quartz à 32,768 kHz :</b></p>	<p>(a) est un composant qui n'a pas besoin d'une alimentation externe</p> <p>(b) est souvent utilisé dans les montres pour obtenir une fréquence de 1 Hz</p> <p>(c) est associé à des composants actifs permettant de le faire osciller</p> <p>(d) 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</p>

#### 4 Faire une action instantanément suite à un événement externe

<p><b>1.</b> Le fonctionnement par interruption permet d'éviter de vérifier en permanence l'état des différents composants/capteurs</p>	<p><input type="checkbox"/> <b>V</b></p> <p><input type="checkbox"/> <b>F</b></p>
<p><b>2.</b> Une interruption bloque l'exécution du programme principal pour effectuer une tâche particulière</p>	<p><input type="checkbox"/> <b>V</b></p> <p><input type="checkbox"/> <b>F</b></p>
<p><b>3.</b> Une routine d'interruption est :</p>	<p>(a) un signal qui interrompt l'exécution du programme principal</p> <p>(b) un événement extérieur au microcontrôleur</p> <p>(c) une fonction appelée lorsqu'une interruption est générée</p>
<p><b>4.</b> Ayant déclaré <code>InterruptIn test(PA_12)</code> au préalable, pour effectuer une action lors du passage de 0 à 1 de la broche <code>PA_12</code> on utilise :</p>	<p>(a) <code>test.attach(&amp;ma_fct, UP);</code></p> <p>(b) <code>test.rise(&amp;ma_fct);</code></p> <p>(c) <code>test.fall(&amp;ma_fct);</code></p>
<p><b>5.</b> Quelle bibliothèque faut-il inclure pour gérer les interruptions :</p>	<p>(a) <code>mbed.h</code></p> <p>(b) <code>interrupts.h</code></p> <p>(c) Aucune, on utilise des fonctions du langage C</p>

**5 Faire une action périodique à une fréquence précise**

<p><b>1.</b> Parmi les propositions suivantes, lesquelles sont correctes ?</p>	<p><b>(a)</b> Il est possible d'utiliser autant d'objets de type <code>Ticker</code> que l'on souhaite</p> <p><b>(b)</b> Il est possible d'utiliser autant d'entrées d'interruption que l'on souhaite, associées à des oscillateurs différents</p> <p><b>(c)</b> Les <code>Ticker</code> permettent de descendre à des périodes de 100 ns</p> <p><b>(d)</b> Les <code>Ticker</code> permettent de descendre à des périodes de <math>1\ \mu\text{s}</math></p>
--	---



# Catégorie 4

# Traitement/Asservissement

## Sommaire

---

1	Caractériser un traitement numérique . . . . .	47
2	Supprimer une fréquence parasite . . . . .	48
3	Filtrer une bande fréquentielle . . . . .	48
4	Filtrer une bande fréquentielle plus efficace- ment . . . . .	49
5	Corriger un asservissement . . . . .	49
	Pour se tester . . . . .	50

---

## 1 Caractériser un traitement numérique

Niveau ②

Pré-requis : -

A l'issue de ce tutoriel, vous saurez :

- Donner la définition d'un traitement numérique.
- Mesurer le temps de calcul d'une itération d'un tel traitement.

**Mots clés :** Échantillonnage, Sortie analogique, Entrée analogique, Calcul, Algorithme , Temps de calcul.

---

## 2 Supprimer une fréquence parasite

<b>Niveau 2</b>	<b>Pré-requis :</b>	Caractériser un traitement numérique
		Récupérer un signal analogique
		Générer une tension analogique
		Faire une action à intervalle régulier

### A l'issue de ce tutoriel, vous saurez :

- définir ce qu'est un traitement linéaire du signal,
- différencier un filtre à moyenne mobile (FIR) d'un filtre récursif (IIR),
- régler les paramètres d'un filtre à moyenne mobile simple,
- mettre en œuvre un dispositif de filtrage temps réel d'un signal parasite.

**Mots clés :** Filtrage numérique, Algorithme linéaire, Réponse en fréquence

---

## 3 Filtrer une bande fréquentielle

**Niveau 3**      **Pré-requis :** Supprimer une fréquence parasite

### A l'issue de ce tutoriel, vous saurez :

- Utiliser `Matlab` pour concevoir un filtre FIR à partir d'un gabarit fréquentiel,
- Mettre en œuvre en temps réel le filtre conçu sur microcontrôleur.

**Mots clés :** `Filter Designer` (`Matlab`), Filtrage numérique, Réponse en fréquence, Filtre FIR

---

## 4 Filtrer une bande fréquentielle plus efficacement

Niveau ③

Pré-requis : Filtrer une bande fréquentielle

**A l'issue de ce tutoriel, vous saurez :**

- Utiliser `Matlab` pour concevoir un filtre IIR à partir d'un gabarit fréquentiel,
- Mettre en œuvre en temps réel le filtre conçu sur microcontrôleur.

**Mots clés :** IIR Filter Design (Matlab), Filtrage numérique, Réponse en fréquence, Filtre IIR

---

## 5 Corriger un asservissement

Niveau ④

Pré-requis : Filtrer une bande fréquentielle

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre un correcteur PID

**Mots clés :** Correcteur PID

---

## Pour se tester

### 1 Caractériser un traitement numérique

<p>On souhaite mesurer le temps d'exécution de l'instruction suivante  <code>myled = !myled;</code>          L'objet <code>myled</code> a été déclaré comme une sortie via l'instruction  <code>DigitalOut myled(D13);</code></p>	
<p><b>1.</b> A l'aide d'un Timer :</p>	<p>(a) il est possible de mesurer directement ce temps</p> <p>(b) il n'est pas possible de mesurer directement ce temps il faut mesurer l'exécution d'une certaine d'opérations de ce type pour avoir une bonne évaluation du temps d'exécution</p> <p>(c) on mesure un temps d'environ 300 ns environ pour cette instruction</p>
<p>Soit le code suivant :</p> <pre>for(int i = 0; i &lt; 100; i++)</pre>	
<p><b>2.</b> Quelles sont les propositions justes :</p>	<p>(a) Même s'il ne fait rien, ce code prend un certain temps pour être exécuté. Il faut obligatoirement mettre des instructions dans les accolades pour que ce code compile.</p> <p>(b) Cette boucle prend moins de 10 <math>\mu</math>s pour être exécutée. Il faut obligatoirement utiliser un objet <code>Timer</code> pour mesurer le temps d'exécution.</p>

```
Soit le code suivant, inclus dans une boucle infinie, avec t défini par
Timer t :
t.reset();
t.start();
t.stop();
pc.printf("Exec = %d usec \r\n", t.read_us());
wait(1);
```

**3.** Quelles sont les propositions justes :

**(a)** Ce code ne s'exécute pas.

**(b)** Ce code affiche Exec = 2 usec.

**(c)** Ce code affiche Exec = 1000000 usec.

**(d)** Ce code permet de connaître le temps d'exécution des instructions start et stop du TIMER.

## 2 Supprimer une fréquence parasite

<p><b>1.</b> Si on inclut l'ensemble du code présent dans la fonction <code>convert</code> dans la boucle principale <code>while(1)</code> (en supprimant la ligne <code>tik.attach(&amp;convert, TE);</code> :</p>	<p>(a) ce ne change pas le comportement du microcontrôleur</p> <p>(b) les opérations correspondant au code déplacé ne sont pas réalisées</p> <p>les opérations correspondant au code</p> <p>(c) déplacé ne sont pas réalisées à des instants précis et réguliers</p>
<p><b>2.</b> Entre les deux méthodes proposées pour réaliser le filtre à moyenne mobile :</p>	<p>(a) le code 1 est plus rapide en temps de calcul que le code 2</p> <p>(b) le code 2 est plus rapide en temps de calcul que le code 1</p> <p>(c) le code 1 a un temps de calcul constant</p> <p>(d) le code 2 a un temps de calcul constant</p>
<p><b>3.</b> En utilisant le code 2 pour le filtre à moyenne mobile :</p>	<p>on peut avoir une fréquence d'échantillonnage plus grande que pour le code 1</p> <p>(a)</p> <p>on doit réaliser la moyenne sur un nombre de points plus faible que dans le code 1 pour une même fréquence d'échantillonnage</p> <p>(b)</p> <p>on doit nécessairement initialiser le tableau de stockage des échantillons avant un premier appel à la fonction <code>convert</code></p> <p>(c)</p>

<p><b>4.</b> Pour le filtre moyeneur :</p>	<p>(a) le signal de sortie est en avance sur le signal d'entrée</p> <p>(b) le signal de sortie est en retard sur le signal d'entrée</p> <p>(c) le signal <code>clk_test</code> a pour période <math>T_E</math> et pour temps de calcul <math>\Delta T = cte</math> le signal <code>clk_test</code> permet de</p> <p>(d) connaître le temps de calcul d'un échantillon</p>
<p><b>5.</b> Si on applique une fréquence d'échantillonnage de 10 kHz pour réaliser le filtre moyeneur :</p>	<p>(a) il faut 100 points pour supprimer un signal à 100 Hz</p> <p>(b) toutes les fréquences du signal d'entrée jusqu'à 5 kHz sont conservées</p> <p>(c) il faut nécessairement un filtre anti-repliement à 20 kHz 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</p> <p>(d)</p>

### 3 Filtrer une bande fréquentielle

<p>Dans les 2 prochaines questions, on considère un filtre FIR dont la fréquence d'échantillonnage est <math>F_e</math> et d'ordre <math>n</math>.</p>	
<p><b>1. Quelles sont les propositions justes :</b></p>	<p>On peut réaliser un filtre numérique</p> <p>(a) passe-bas de fréquence de coupure supérieure à <math>F_e/2</math></p> <p>(b) Pour réaliser un filtre numérique passe-haut, il est indispensable de mettre un filtre anti-repliement passe-bas devant le filtre numérique</p> <p>(c) Un tel filtre numérique possédera <math>n + 1</math> coefficients réels.</p> <p>(d) Un tel filtre numérique possédera <math>n - 1</math> coefficients réels.</p>
<p><b>2. Quelles sont les propositions justes :</b></p>	<p>Augmenter l'ordre du filtre ne modifie pas le temps d'exécution du calcul du filtre</p> <p>(a) Augmenter l'ordre du filtre augmente le temps d'exécution du calcul du filtre</p> <p>(b) Diminuer l'ordre du filtre augmente le temps de conversion des échantillons</p> <p>(c) Diminuer la fréquence d'échantillonnage modifie l'ordre du filtre</p> <p>(d)</p>
<p><b>3. La fréquence réduite :</b></p>	<p>(a) va de 0 à 1</p> <p>(b) va de -0.5 à 0.5</p> <p>(c) va de 0 à 0.5</p> <p>(d) peut être affichée que de 0 à 0.5</p>



<p>On souhaite réaliser un filtre FIR de type <b>passé-bande</b>, avec les caractéristiques suivantes :</p> <ul style="list-style-type: none"> <li>— <math>F_{ech} = 10</math> kHz</li> <li>— <math>F_{stop1} = 1</math> kHz, <math>F_{pass1} = 1.5</math> kHz</li> <li>— <math>F_{stop2} = 3</math> kHz, <math>F_{pass2} = 2.5</math> kHz</li> <li>— Atténuation de 60 dB hors bande-passante et 1 dB dans la bande-passante</li> </ul>	
<p><b>4.</b> Quelles sont les propositions justes :</p>	<p>(a) Un filtre fenêtré de type <b>Kaiser</b> est d'ordre supérieur à 70.</p> <p>Un filtre <i>Equiripple</i> est d'ordre plus</p> <p>(b) élevé qu'un filtre fenêtré de type <b>Kaiser</b>.</p> <p>Un filtre <i>Equiripple</i> possède 40 coefficients, dont le premier vaut (environ) :</p> <p>(c) 0.0011</p> <p>Un filtre <i>Equiripple</i> possède 40 coefficients, dont le premier vaut (environ) :</p> <p>(d) 1.5038</p>
<p><b>5.</b> En reprenant l'exemple du filtre passe-haut proposé dans le tutoriel, quelles sont les propositions justes :</p>	<p>L'instruction :</p> <p>(a) <code>arm_fir_f32(&amp;fir, &amp;in, &amp;out, 1);</code> s'exécute en moins de 10 <math>\mu</math>s</p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.</p> <p>(b)</p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.</p> <p>(c)</p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus basse.</p> <p>(d)</p>

## 4 Filtrer une bande fréquentielle plus efficacement

<p>On considère un filtre IIR dont la fréquence d'échantillonnage est <math>F_e</math> et possédant <math>N</math> coefficients <math>\alpha</math> et <math>M</math> coefficients <math>\beta</math>.</p>	
<p><b>1.</b> Quelles sont les propositions justes :</p>	<p>(a) <math>N</math> et <math>M</math> sont nécessairement égaux</p> <p>(b) Le calcul de la sortie nécessitera <math>N+M</math> opérations élémentaires.</p> <p>(c) Un tel filtre sera nécessairement instable.</p>
<p><b>2.</b> Quelles sont les propositions justes :</p>	<p>(a) Des filtres de Butterworth passe-bas et passe-haut, définis avec le même gabarit, auront le même nombre de coefficients</p> <p>(b) A gabarit identique, un filtre IIR sera moins performant qu'un filtre FIR</p> <p>(c) A gabarit identique, un filtre IIR pourra utiliser une fréquence d'échantillonnage plus importante qu'un filtre FIR</p> <p>(d)</p>
<p><b>3.</b> On s'intéresse aux <b>filtres passe-bas</b> du paragraphe <i>Gabarit et conception</i> de la section <i>Conception du filtre sous Matlab</i> proposés dans le tutoriel. On obtient :</p>	<p>(a) un filtre d'ordre 14 pour les 2 modes de Chebyshev</p> <p>(b) un filtre d'ordre 48 pour le filtre de Butterworth</p> <p>(c) un ordre supérieur pour le filtre elliptique que pour les filtres de Chebyshev</p> <p>(d) un ordre inférieur pour le filtre elliptique que pour le filtre de Butterworth</p>

<p><b>4.</b> En reprenant l'exemple du filtre passe-haut proposé dans le tutoriel, quelles sont les propositions justes :</p>	<p>L'instruction :</p> <p>(a) <code>arm_iir_lattice_f32(&amp;my_iir, &amp;in, &amp;out, 1);</code> s'exécute en plus de <math>10\ \mu\text{s}</math></p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.</p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus basse.</p> <p>(c) Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.</p> <p>(d) Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.</p>
---	---



# Catégorie 5

# Communication

## Sommaire

---

1	Faire communiquer deux systèmes . . . . .	59
2	Configurer un réseau point à point RS232 . . . . .	60
3	Configurer un réseau point à point SPI . . . . .	60
4	Interfacer un convertisseur analogique-numérique en SPI . . . . .	60
5	Ajouter de la mémoire de données en SPI . . . . .	61
6	Configurer un réseau adressable I <sup>2</sup> C . . . . .	61
7	Interfacer un accéléromètre en I <sup>2</sup> C . . . . .	61
	Pour se tester . . . . .	62

---

## 1 Faire communiquer deux systèmes

Niveau ①

Utiliser un écran LCD  
**Pré-requis :** Débuguer son programme et utiliser l'affichage série

**A l'issue de ce tutoriel, vous saurez :**

- Expliquer le principe d'une communication numérique entre dispositifs électroniques
- Différencier les 3 types courants de bus de communication

**Mots clés :** Protocole et bus SPI, Protocole et bus RS232, Protocole série, Protocole et bus I<sup>2</sup>C

---

## 2 Configurer un réseau point à point RS232

**Niveau ②**      **Pré-requis :** Faire communiquer deux systèmes

**A l'issue de ce tutoriel, vous saurez :**

- mettre en œuvre une liaison RS232 bidirectionnelle entre le microcontrôleur et l'ordinateur

**Mots clés :** Protocole et bus RS232, Protocole série.

## 3 Configurer un réseau point à point SPI

**Niveau ②**      **Pré-requis :** Faire communiquer deux systèmes

**A l'issue de ce tutoriel, vous saurez :**

- Connecter le microcontrôleur à un autre circuit via une liaison SPI

**Mots clés :** Protocole et bus SPI

---

## 4 Interfacer un convertisseur analogique-numérique en SPI

**Niveau ③**      **Pré-requis :** Configurer un réseau point à point SPI

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre un convertisseur numérique analogique externe via une liaison SPI

**Mots clés :** Convertisseur numérique-analogique, CNA, DAC, , Protocole et bus SPI.

---

## 5 Ajouter de la mémoire de données en SPI

**Niveau ③ Pré-requis :** Configurer un réseau point à point SPI

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre un circuit externe de mémoire SRAM via une liaison SPI

**Mots clés :** Mémoire externe, Protocole et bus SPI

---

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

**Niveau ③ Pré-requis :** Faire communiquer deux systèmes

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre une liaison I<sup>2</sup>C
- Analyser les signaux de communication

**Mots clés :** Protocole et bus I<sup>2</sup>C

---

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

**Niveau ④ Pré-requis :** Configurer un réseau adressable I<sup>2</sup>C

**A l'issue de ce tutoriel, vous saurez :**

- Mettre en œuvre un accéléromètre 3 axes via une liaison I<sup>2</sup>C.

**Mots clés :** Protocole et bus I<sup>2</sup>C, Accéléromètre 3 axes I<sup>2</sup>C

---

## Pour se tester

### 1 Faire communiquer deux systèmes

<p><b>1. Quelles sont les propositions justes :</b></p>	<p>Une liaison parallèle de 8 bits cadencée à 10 MHz sera moins rapide qu'une liaison série de 8 bits cadencée à 100 MHz</p> <p>(a)</p> <p>Une liaison asynchrone est plus fiable qu'une liaison synchrone</p> <p>(b)</p> <p>Une liaison Half-Duplex permet d'échanger des données dans les deux directions en même temps</p> <p>(c)</p> <p>Les liaisons synchrones ont au plus 2 fils de transmission</p> <p>(d)</p>
<p><b>2. Quelles sont les propositions justes :</b></p>	<p>Un bus SPI entre un maître et 4 esclaves nécessite qu'une seule ligne de transmission CS depuis le maître</p> <p>(a)</p> <p>A fréquence d'horloge équivalente, un bus SPI sera plus rapide qu'une liaison I2C pour transmettre des données utiles</p> <p>(b)</p> <p>Une trame I2C est plus courte qu'une trame SPI</p> <p>(c)</p> <p>Une trame SPI est limitée à 1 octet à chaque fois</p> <p>(d)</p>



## 2 Configurer un réseau point à point RS232

<p><b>1.</b> Pour une transmission à 9600 bauds :</p>	<p>(a) Il faut environ 1 ms pour transmettre un caractère avec la fonction <code>putc</code></p> <p>(b) Il faut un peu moins de 100 <math>\mu</math>s pour transmettre un bit</p> <p>(c) Il faut entre 7 ms et 8 ms pour transmettre la chaîne de caractères "Bonjour" avec la fonction <code>printf</code></p> <p>(d) On ne peut pas transmettre plus de 5 caractères par seconde.</p>
<p><b>2.</b> Pour une transmission à 115200 bauds :</p>	<p>(a) l'envoi d'un caractère avec la fonction <code>putc</code> prend plus de 100 <math>\mu</math>s</p> <p>(b) Il faut un peu moins de 10 <math>\mu</math>s pour transmettre un bit</p> <p>(c) Il faut plus de 1 ms pour transmettre la chaîne de caractères "ProTIS 2018" avec la fonction <code>printf</code></p>

### 3 Configurer un réseau point à point SPI

<p>On utilise le code suivant pour configurer une liaison SPI : SPI</p> <pre>spi(D11, D12, D13);</pre> <p>Puis dans le programme principal :</p> <pre>spi.format(8, 0);</pre> <pre>spi.frequency(100000);</pre>	
<p><b>1. Quelles propositions sont justes :</b></p>	<p>(a) On visualisera une horloge à 100 kHz sur la broche D13 uniquement lors de l'utilisation de la fonction <code>spi.write(x)</code> ;</p> <p>(b) On visualisera en permanence une horloge à 100 kHz sur la broche D13</p> <p>(c) Les données transmises du maître vers l'esclave sont visualisables sur la broche D11</p> <p>(d) Les données transmises du maître vers l'esclave sont visualisables sur la broche D12</p>
<p><b>2. Quelles propositions sont justes :</b></p>	<p>(a) Il faut environ <math>25 \mu s</math> pour transmettre un caractère par la fonction <code>spi.write</code></p> <p>(b) Le signal d'horloge est à l'état bas lorsqu'il n'y a pas de transmission</p> <p>(c) Les deux fils de données sont à l'état haut lorsqu'il n'y pas de transmission</p> <p>(d) Un '1' logique est transmis par une différence de potentiel de 0V par rapport à la masse</p>

## 4 Interfacer un convertisseur analogique-numérique en SPI

<p><b>1.</b> Dans l'exemple de génération d'une rampe de tension (pour une alimentation de 3.3 V), pour la méthode standard :</p>	<p>(a) la fréquence du signal résultant est d'environ 4 Hz</p> <p>(b) le délai entre chacun des échantillons est constant et égal à 1 ms</p> <p>(c) la tension minimale en sortie du DAC est de 1 V</p> <p>(d) la tension maximale de sortie du DAC est de 3.3 V</p>
<p><b>2.</b> Dans l'exemple de génération d'une rampe de tension (pour une alimentation de 3.3 V) :</p>	<p>la méthode par interruption est plus</p> <p>(a) longue en temps de calcul que la méthode standard</p> <p>la méthode par interruption permet</p> <p>(b) une mise en place des échantillons plus régulière que la méthode standard</p> <p>(c)</p> <p>(d)</p>
<p><b>3.</b> Dans l'exemple de génération d'un signal sinusoïdal (pour une alimentation de 3.3 V) :</p>	<p>(a) Il n'est pas possible d'utiliser la méthode par interruption</p> <p>(b) La méthode par interruption est préférable à la méthode standard</p> <p>(c) La fréquence du signal résultant est d'environ 16 Hz</p> <p>La fréquence maximale du signal résultant que l'on peut obtenir est supérieure à 200 Hz</p> <p>(d)</p>

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

<p>On utilise le code suivant pour configurer une liaison SPI : I2C</p> <pre>my_i2c(I2C_SDA, I2C_SCL);</pre> <p>Puis dans le programme principal :</p> <pre>my_i2c.frequency(1000000);</pre>	
<p><b>1.</b> Quelles propositions sont justes :</p>	<p>On visualisera une horloge à 1MHz sur la broche D14 uniquement lors de l'utilisation de la fonction <code>my_i2c.write(...)</code> ;</p> <p>(b) Il n'est pas nécessaire de connecter un périphérique I2C pour visualiser les trames I2C sortant de la carte MBED</p> <p>Il est nécessaire de connecter un périphérique I2C pour visualiser les trames I2C sortant de la carte MBED</p> <p>(c) et de transmettre à l'adresse du périphérique pour obtenir la trame complète</p> <p>Les données transmises dans les deux</p> <p>(d) sens sont visualisables sur la broche D15</p>
<p><b>2.</b> Il est possible dans un même réseau I2C :</p>	<p>(a) de mettre deux périphériques identiques et de pouvoir les adresser indépendamment</p> <p>(b) de mettre deux périphériques identiques et de pouvoir les adresser indépendamment s'il est possible de modifier localement leurs adresses</p> <p>(c) de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses identiques</p> <p>(d) de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses différentes</p>

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

TO DO	
<p><b>1.</b> On s'intéresse à un accéléromètre de type ADXL350 :</p>	<p>(a) Il est possible de communiquer en SPI uniquement</p> <p>(b) Il est possible de communiquer en SPI et en I2C</p> <p>(c) Pour communiquer en I2C avec ce composant, il faut forcer la broche CS à 0 V</p> <p>(d) Pour communiquer en I2C avec ce composant, il faut forcer la broche CS à 3.3 V</p>
<p><b>2.</b> Avec l'accéléromètre de type ADXL350 :</p>	<p>(a) Il n'est pas possible de communiquer en I2C en High-Speed Mode</p> <p>Il faut obligatoirement utiliser une entrée d'interruption sur le microcontrôleur, cablée sur INT1 ou INT2 de l'accéléromètre,</p> <p>(b) pour récupérer les données</p> <p>Il n'est pas possible de récupérer les 6</p> <p>(c) octets de données en une seule trame de réponse</p> <p>Il est possible de récupérer les 6 octets</p> <p>(d) de données en une seule trame de réponse</p> <p>(e)</p>

# Index

- I<sup>2</sup>C, 60, 61
- RS232, 60
- SPI, 26, 60, 61
  
- Affichage, 26
- Algorithme, 47
- Algorithme linéaire, 48
- astable, 40
  
- Bouton poussoir, 26
  
- Calcul, 47
- Circuit oscillateur, 40
- CNA, DAC, 61
- Console Série, 16
- Correcteur, 49
  
- Débogage, 16
- Documentation, 17
  
- Échantillonnage, 47
- Entrée analogique, 27, 47
- Entrée numérique, 26
  
- Filter Designer (Matlab), 48
- Filtrage numérique, 48, 49
- Filtre FIR, 48
- Filtre IIR, 49
  
- Horloge, 40
  
- IIR Filter Design (Matlab), 49
- Interruptions, 40, 41
  
- LCD, 26
- LED, 16, 27
  
- MBED Interface, 15
- Memoire externe, Mémoire externe, 61
- Microcontrôleurs et microprocesseurs , comparaison17
- Microphone, 28
- monostable, 40
- Moteur à courant continu, 28
  
- Nucléo , Premiers pas sur 16
  
- Oscillateur, 41
- Oscilloscope, 16
  
- PID, 49
- Protocole série, 60
  
- Quartz, 40
  
- Réponse en fréquence, 48, 49
  
- Servomoteur, 28
- Sortie analogique, 27, 47
- Sortie numérique, 16
- Sortie PWM, 27, 28
- SRAM, 61
  
- Temps de calcul, 47
- Timer, 40

.



Procédés de Traitement de l'Information et du Signal / ProTIS

Tutoriels « A la carte »

