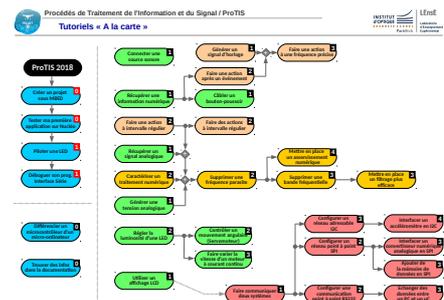
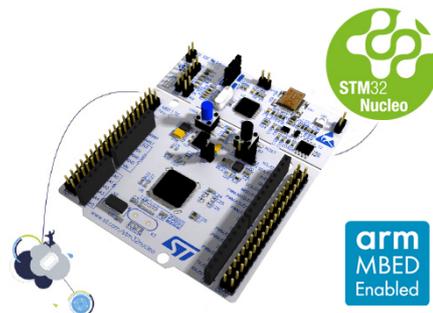


Prototypage ProTIS

Polycopié n° 2

PROcédés de Traitement de l'Information et du Signal



Cycle Ingénieur - 2^{ème} 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

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

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 efficacement	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

Niveau ②

Pré-requis :

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</p> <pre>myled = !myled;</pre> <p>L'objet myled a été déclaré comme une sortie via l'instruction</p> <pre>DigitalOut myled(D13);</pre>	
<p>1. 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</p> <p>(c) il faut mesurer l'exécution d'une certaine d'opérations de ce type pour avoir une bonne évaluation du temps d'exécution</p> <p>(d) 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 < 100; i++)</pre>	
<p>2. Quelles sont les propositions justes :</p>	<p>(a) Même s'il ne fait rien, ce code prend un certain temps pour être exécuté.</p> <p>(b) Il faut obligatoirement mettre des instructions dans les accolades pour que ce code compile.</p> <p>(c) Cette boucle prend moins de 10 μs pour être exécutée.</p> <p>(d) Il faut obligatoirement utiliser un objet Timer 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>1. 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(&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>2. 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>3. 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) 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) 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>4. 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 T_E et pour temps de calcul $\Delta T = cte$</p> <p>(d) le signal <code>clk_test</code> permet de connaître le temps de calcul d'un échantillon</p>
<p>5. 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</p> <p>(d) 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>

3 Filtrer une bande fréquentielle

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

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

4. Quelles sont les propositions justes :

(a) Un filtre fenêtré de type **Kaiser** est d'ordre supérieur à 70.

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 *Equiripple* possède 40 coefficients, dont le premier vaut (environ) : 1.5038

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

L'instruction :

(a) `arm_fir_f32(&fir, &in, &out, 1);`
s'exécute en moins de 10 μ s

(b) Le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.

(c) Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus élevée.

(d) Le filtre peut être utilisé à une fréquence d'échantillonnage inférieure à 10 kHz mais la fréquence de coupure sera plus basse.

4 Filtrer une bande fréquentielle plus efficacement

<p>On considère un filtre IIR dont la fréquence d'échantillonnage est F_e et possédant N coefficients α et M coefficients β.</p>	
<p>1. Quelles sont les propositions justes :</p>	<p>(a) N et M sont nécessairement égaux</p> <p>(b) Le calcul de la sortie nécessitera $N+M$ opérations élémentaires.</p> <p>(c) Un tel filtre sera nécessairement instable.</p>
<p>2. Quelles sont les propositions justes :</p>	<p>Des filtres de Butterworth passe-bas et passe-haut, définis avec le même gabarit, auront le même nombre de coefficients</p> <p>(a) A gabarit identique, un filtre IIR sera moins performant qu'un filtre FIR</p> <p>A gabarit identique, un filtre IIR</p> <p>(c) pourra utiliser une fréquence d'échantillonnage plus importante qu'un filtre FIR</p> <p>(d)</p>
<p>3. On s'intéresse aux filtres passe-bas 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>4. 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(&my_iir, &in, &out, 1);</code> s'exécute en plus de $10\ \mu\text{s}$</p> <p>Le filtre peut être utilisé à une fréquence d'échantillonnage différente de 10 kHz en conservant les mêmes caractéristiques.</p> <p>(b) 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)</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 ² C	61
7	Interfacer un accéléromètre en I ² C	61
	Pour se tester	62

1 Faire communiquer deux systèmes

Niveau ① Utiliser un écran LCD
Pré-requis : Déboguer 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 I²C

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 I²C

Niveau ③ Pré-requis : Faire communiquer deux systèmes

A l'issue de ce tutoriel, vous saurez :

- Mettre en œuvre une liaison I²C
- Analyser les signaux de communication

Mots clés : Protocole et bus I²C

7 Interfacer un accéléromètre en I²C

Niveau ④ Pré-requis : Configurer un réseau adressable I²C

A l'issue de ce tutoriel, vous saurez :

- Mettre en œuvre un accéléromètre 3 axes via une liaison I²C.

Mots clés : Protocole et bus I²C, Accéléromètre 3 axes I²C

Pour se tester

1 Faire communiquer deux systèmes

<p>1. Quelles sont les propositions justes :</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>2. Quelles sont les propositions justes :</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>1. 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 μs 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>2. Pour une transmission à 115200 bauds :</p>	<p>(a) l'envoi d'un caractère avec la fonction <code>putc</code> prend plus de 100 μs</p> <p>(b) Il faut un peu moins de 10 μs 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>1. Quelles propositions sont justes :</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>2. Quelles propositions sont justes :</p>	<p>(a) Il faut environ $25\mu s$ 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>1. 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>2. 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>3. 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 I²C

<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>1. 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>(a)</p> <p>Il n'est pas nécessaire de connecter un périphérique I2C pour visualiser les trames I2C sortant de la carte MBED</p> <p>(b)</p> <p>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</p> <p>(c)</p> <p>Les données transmises dans les deux sens sont visualisables sur la broche D15</p> <p>(d)</p>
<p>2. Il est possible dans un même réseau I2C :</p>	<p>de mettre deux périphériques identiques et de pouvoir les adresser indépendamment</p> <p>(a)</p> <p>de mettre deux périphériques identiques et de pouvoir les adresser indépendamment s'il est possible de modifier localement leurs adresses</p> <p>(b)</p> <p>de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses identiques</p> <p>(c)</p> <p>de mettre autant de périphériques que l'on souhaite s'ils ont tous des adresses différentes</p> <p>(d)</p>

7 Interfacer un accéléromètre en I²C

TO DO	
<p>1. 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>2. 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, câblée 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

- I²C, 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 »

