

# Introduction aux paramètres de SigmaStudio pour DSPiY

v1.1

1. Notion de cellule dans SigmaStudio
  2. Description de quelques types de cellules
    - 2.1 Filtre général 2<sup>nd</sup> ordre double précision
    - 2.2 Filtre général 2<sup>nd</sup> ordre simple précision
    - 2.3 Filtre général 1<sup>er</sup> ordre double précision
    - 2.4 Parametric EQ
    - 2.5 Crossover
    - 2.6 Volume
    - 2.7 Inversion
3. Variables dans sigmastudio
4. Ressources et consommation
5. Simulation
6. Utilisation pour DSPiY
7. Correspondances E/S SigmaStudio – DSPiY

## Evolutions :

V1.0 17/02/2013 : version initiale

### **1. Notion de cellule dans SigmaStudio**

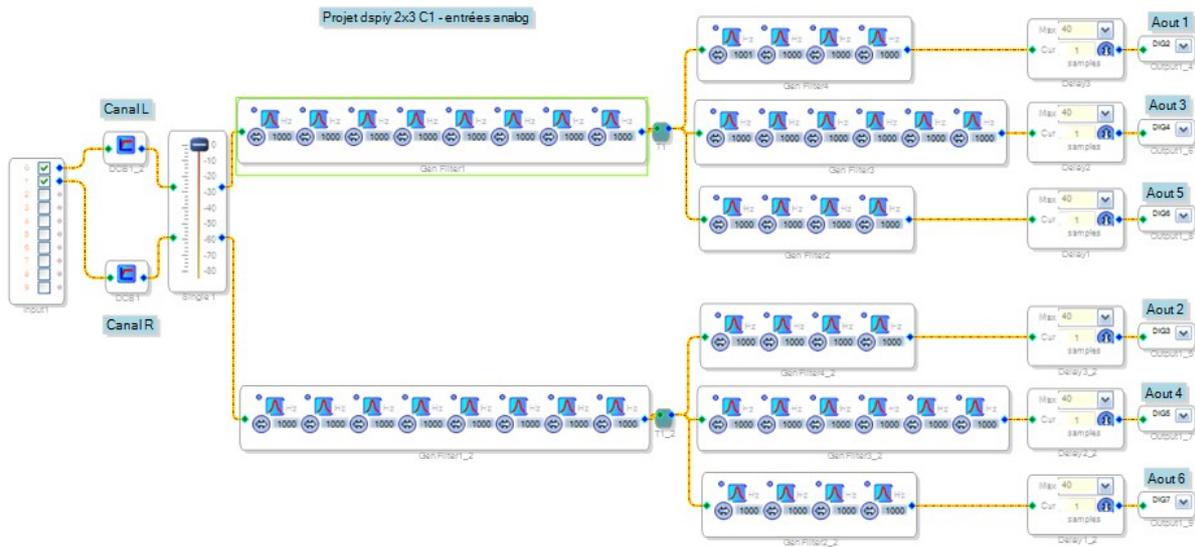
Dans l'interface graphique, un projet est une succession de **cellules**. Chaque algorithme de traitement appliqué au signal se fait dans une cellule.

Une cellule peut être simple comme un filtre général, ou complexe comme un crossover complet de 3 voies. SigmaStudio dispose d'un grand nombre de traitements que l'on n'utilisera pas dans le contexte DSPiY.

Une cellule peut, selon son type comprendre plusieurs algorithmes identiques. Par exemple 6 filtres génériques successifs dans une même cellule. C'est équivalent à utiliser 6 cellules de filtre simple en série. Ces filtres sont en série et peuvent être de types différents. On utilise le terme « filtre » de manière générale, en terme de traitement audio cela peut être un filtre PB, PH ou PB, un point d'équalization, etc.

Pour l'architecture du projet SigmaStudio pour le DSPiY on utilisera essentiellement des filtres de type general. Ce sont des filtres biquads génériques dont on affectera les coefficients pour qu'ils exécutent la fonction désirée.

Exemple :



Description de cette architecture: (projet 2 x 3 )

Premier canal : entrée ; suivi d'un filtre composante continue (DCB) ; volume ; cellule General Filter global à toutes les voies comportant 8 filtres en série ; suivi d'un T pour répartir sur 3 voies.

Chaque voie a une cellule General Filter comportant 4 ou 6 filtres, suivi d'un délai et enfin relié à une sortie digitale. Le deuxième canal est idem.

## 2. Description de quelques types de cellules

### 2.1 Filtre général 2<sup>nd</sup> ordre double précision

Dans un projet SigmaStudio ces filtres ont un Parameter Name de la forme: EQ1940Single10B1

Qui se décompose :

EQ1940 : nom de la cellule (c'est un filtre générique 2<sup>nd</sup> ordre)

Single : 1 voie (Dual pour 2 voies)

1 : numéro de la cellule

0 : numéro de coef

B : type de coef.

1 : numéro du filtre en série dans la cellule. (maximum 15)

NB coefficients : 5

Dans le fichier .params un filtre complet de ce type est une succession de 5 paragraphes, un par coefficient biquad, avec chacun le nom du paramètre, l'adresse et valeur.

Noter que DSPiYStudio reformate ces données et les range dans la section « parametres DSP » du fichier APPLI.apd. Chaque instance devient une ligne « Nom Adresse Valeur ».

Un filtre biquad de ce type complet devient 5 lignes dans un fichier Appli.apd :

EQ1940Single10B1 5 1,00365306543459

EQ1940Single11B1 6 -1,98635040364354

EQ1940Single12B1 7 0,986959392283122

EQ1940Single11A1 8 -1,98635040364354

EQ1940Single12A1 9 0,99061245771771

Ce qui correspond aux Nom, Adresse, valeur des 5 coefficients B0, B1, B2, A1, A2 de ce biquad.

## **2.2 Filtre général 2<sup>nd</sup> ordre simple précision**

Exemple :

Param Name: EQ1940SingleS10B1

Single devient SingleS. Le reste est idem à double précision.

**Note sur la précision:** Most of these filters will let you select between double- and single-precision computation. If there available system resources, double-precision should normally be used: it uses 56 bits for each calculation and takes 10 instructions per filter. Single-precision uses 28 bits for calculations, takes 6 instructions per filter, but saves you 3 RAM spaces over the double-precision algorithm. Single-precision should not be used for signal content below 1/10 the sampling frequency or for high-Q filters.

## **2.3 Filtre général 1<sup>er</sup> ordre double précision**

Exemple :

Param Name: FistOrderSingle101

Types : highpass, lowpass, and flat

limites : +20db -40db

NB coefficients: 3

## **2.4 Parametric EQ**

Au lieu d'utiliser une cellule avec plusieurs filtres, SigmaStudio propose une cellule complète pré-définie pour créer un filtre paramétrique 1 voie.

Très utile pour utilisation avec SigmaStudio seul puisque permet la modification des filtres à la volée avec représentation graphique.

*The Parametric EQ block extends the functionality of the General (2nd-Order) filter with an enhanced graphical filter design tool. In addition, multiple 2-nd order filter stages can be cascaded to create complex filter responses.*

*This filter provides:*

- *Graphical design of filter response curves.*
- *High order filtering, via cascaded 2nd order filters (maximum of 15)*
- *Independent filter type and settings for each filter stage.*

*This block is implemented using cascaded biquad filters. See [General 2nd-Order Filters](#) (in Algorithm Information) for details about the biquad filter design techniques used by this block.*

Exemple :

Param Name: PEQ1Chan10B1

PEQ nom de la cellule

1 apparemment toujours 1 donc le nom de la cellule est PEQ1

Chan1 :numéro de la cellule filtre param. (Exemple : un filtre paramétrique 2 voies aura

Chan1 et Chan2)

0 numéro du coef

B type de coef

1 numéro du filtre en série dans la cellule (max 15)

*Variables via sigmastudio :* (valeurs min max pour Peaking Filter)

Fréquence, Qfactor de 0.05 à 20, Boost de -100 à +30db, gain 1 de -10 à +10db

## **2.5 Crossover :**

*The Crossover block includes 2-way and 3-way crossover filters, typically used in loudspeaker systems*

to split the audio signal into separate frequency bands.

This filter provides:

- 2-way or 3-way crossover filtering.
- Graphical design of crossover response.
- Selectable crossover types: Linkwitz-Riley, Butterworth, Bessel.
- Selectable filter orders: 2nd, 3rd, 4th, 6th, and 8th.

This block is implemented using cascaded filters. See [General 2nd-Order Filters](#) (in Algorithm Information) for details about the biquad filter design techniques used in this block.

On dispose de :

Flow pour sortie Low, Flow et FHigh pour sortie Mid et Fhigh pour sortie High

Type de filtre et pente (**nécessite une nouvelle compilation lors d'un changement**)

Polarité pour sorties Low et High

Gain par sortie de +-10db max.

Exemple :

Parameter Name = CrossoverFilter3WayAlgDP1Bo\_o

CrossoverFilter3WayAlgDP nom de la cellule

1 numéro de la cellule Crossover

B type de coef

0 numéro du coef (donc différente logique que pour les filtres généraux)

\_o numéro du filtre (8 pour 3 voies 24db/o)

(0 et 1 pour low, 2 3 4 et 5 pour mid, 6 et 7 pour high)

Parameter Name = CrossoverFilter3WayAlgDP1LowInvert

CrossoverFilter3WayAlgDP nom de la cellule

1 numéro de la cellule Crossover

LowInvert sortie low (HighInvert pour sortie high)

Valeur 1 pour normal; -1 pour inversion

## **2.6 Volume**

Exemple :

Param Name: Gain1940AlgNS1

Gain1940AlgN nom du parameter

S volume simple (1 voie, mais apparemment un 2 voies = 2 simple, donc toujours S )

1 numéro de la cellule Gain

Valeurs : 0 pour -80db à 1 pour -0db

## **2.7 Inversion**

Exemple :

Param Name: EQ1940Invert1gain

Valeur 1 ou -1 pour inversion.

## **3. Variables dans sigmastudio**

- Parametrique : gain général (scale gain)+- 15db, gain du filtre (appelé aussi boost) +-10db, fréquence et Q de 0 à 16

- Shelving : Low ou high, gain général (scale gain)+- 15db, gain du filtre (appelé aussi boost) +-10db, fréquence et Slope ? de 0 à 2

- Général HP/LP/BP/BS : gain +-15db, fréquence et Q de 0 à 16 (pour les filtre passe bande : bandwidth de 0 à 11 octaves)
- Peaking : gain +-15db, fréquence et Q de 0 à 16
- Bessel Low-Pass / High-Pass
- Quelques autres types disponibles : Tone Control, IIR Coefficient (direct coefficient entry), 1st-Order Low-Pass / High-Pass, All-pass, Notch, Chebyshev Low-Pass / High-Pass

Consulter la doc de SigmaStudio pour plus d'information.

#### 4. Ressources

Le nombre d'instructions maximal du DSP est limité. Pour chaque sample, l'ensemble de la chaîne de traitement est appliqué. Le DSP a une puissance de 50mips. Pour 96k samples par seconde, cela fait environ 512 instructions maximum au total.

- Instructions : En 96kHz on dispose de **512** instructions. Le double, **1024** en 48kHz, la moitié en 192kHz.
- Data ram : **2048**
- Paramètre ram : **1024**

Exemples de consommation :

- 1 filtre général double précision 2eme ordre (biquad) consomme 10 instructions, 6 data et 5 paramètres
- 1 commande de volume sur 1 voie : 3 instructions, 1 data et 1 paramètre.
- 1 inversion de 180° sur 1 voie : 2 instructions, 1 data et 1 paramètre.
- 1 délai : 5 instructions
- 1 readback : 3 instructions

Le projet complet 2x3 présent au chapitre 1. consomme 506 instructions pour deux canaux.

#### **Pour déterminer les ressources consommées par un projet SigmaStudio :**

Compiler le projet en cliquant « Link Compile Connect ». SigmaStudio crée des répertoires et fichiers. Rechercher dans les fichiers créés un fichier texte nommé *compiler\_output.txt* Je le trouve dans le dossier un répertoire *IC1\_nomprojet* sous dossier *net\_list\_out2*.

Attention au bug SigmaStudio qui affiche toujours 1024 instructions disponibles dans ce fichier à la place de 512. Il se croit toujours en 48kHz !

#### 5. Simulation

SigmaStudio permet de simuler la réponse en fréquence du projet.

- insérer en début de chaîne de traitement une cellule « Simulation Stimuli ».
- insérer une cellule « Simulation Probe » à l'endroit où l'on souhaite prendre le signal de simulation (a priori en sortie !). Noter qu'une Probe peut prendre plusieurs entrées et sorties pour afficher plusieurs voies sur un seul graphe par exemple
- cliquer sur probe pour ouvrir l'affichage de la simulation
- cliquer sur l'icône Propagate Sampling Rate pour lancer la simulation. (après avoir fait Link & Compile Connect)

## 6. Utilisation avec DSPiy

Le mode d'utilisation privilégié du DSPiy est d'utiliser un projet SigmaStudio ayant l'architecture des canaux, voies et cellules biquad souhaitée, mais sans aucun paramétrage. C'est le rôle de dsEQ ou jEQ de simuler et calculer les coefficients des biquads et modifier les paramètres du projet en conséquence.

Si on a la licence SigmaStudio, on peut aussi réaliser l'intégralité d'un projet dans SigmaStudio en utilisant toutes les fonctions disponibles, dans la limite des ressources du DSPiy évidemment. Dans ce cas, il faut paramétrer entièrement chaque cellule dans SigmaStudio.

Pour cela :

- créer son projet dans SigmaStudio
- Link compile connect
- Export system files (conseillé de le faire dans un répertoire dédié)
- Vérifier que les ressources utilisées sont compatibles avec le DSPiy
- Dans DSPiyStudio, onglet avancé, créer le fichier Appli.apd en pointant le fichier .params qui se trouve dans le dossier d'export.
- Ne pas faire de modification dans dsEQ ou jEQ ou autre
- Transférer l'Appli.apd vers le DSPiy

Attention : Pour un biquad, le DSP s'attend à voir les 5 coefficients B0, B1, B2, A1, A2 avec le signe de A1 et A2 inversés !

SigmaStudio en tient compte : pour un biquad paramétré dans SigmaStudio, le signe des coefficients A1 et A2 est inversé dans le fichier .params

Comme les informations dans le fichier Appli peuvent provenir de plusieurs sources (dsEQ, jEQ, maj manuelle,...) et doivent être interprétables normalement, DSPiyStudio inverse les signes de ces deux coefficients en provenance de SigmaStudio lors de la création du fichier Appli.apd

Lors du transfert des paramètres entre le pc et le DSPiy, DSPiyStudio inverse les signes de ces coefficients afin que le biquad soit correctement interprété par le DSP.

## 7. Correspondances Entrées-Sorties SigmaStudio avec DSPiY

INPUT	
SigmaStudio	DSPiY
0	ANALOG IN1 – L
1	ANALOG IN2 – R
2	DIGITAL IN L
3	DIGITAL IN R

ANALOG OUT	
SigmaStudio	DSPiY (J4,J5,J6)
DIG 0	I2S out
DIG 1	I2S out
DIG 2	AOUT1
DIG 3	AOUT2
DIG 4	AOUT 3
DIG 5	AOUT 4
DIG 6	AOUT 5
DIG 7	AOUT 6

Aux ANALOG OUT	
SigmaStudio	DSPiY (J25)
DAC0	Aux1
DAC1	Aux2

