

Logiciels DSPiy

DSPiyStudio logiciels et format des fichiers

incomplet

1. Organisation et nommage
2. Création d'une Application DSPiy
3. Modification d'une Application
4. Fonctionnement de DSPiyStudio
5. Description des paramètres configuration (parametres uc)

Evolutions :

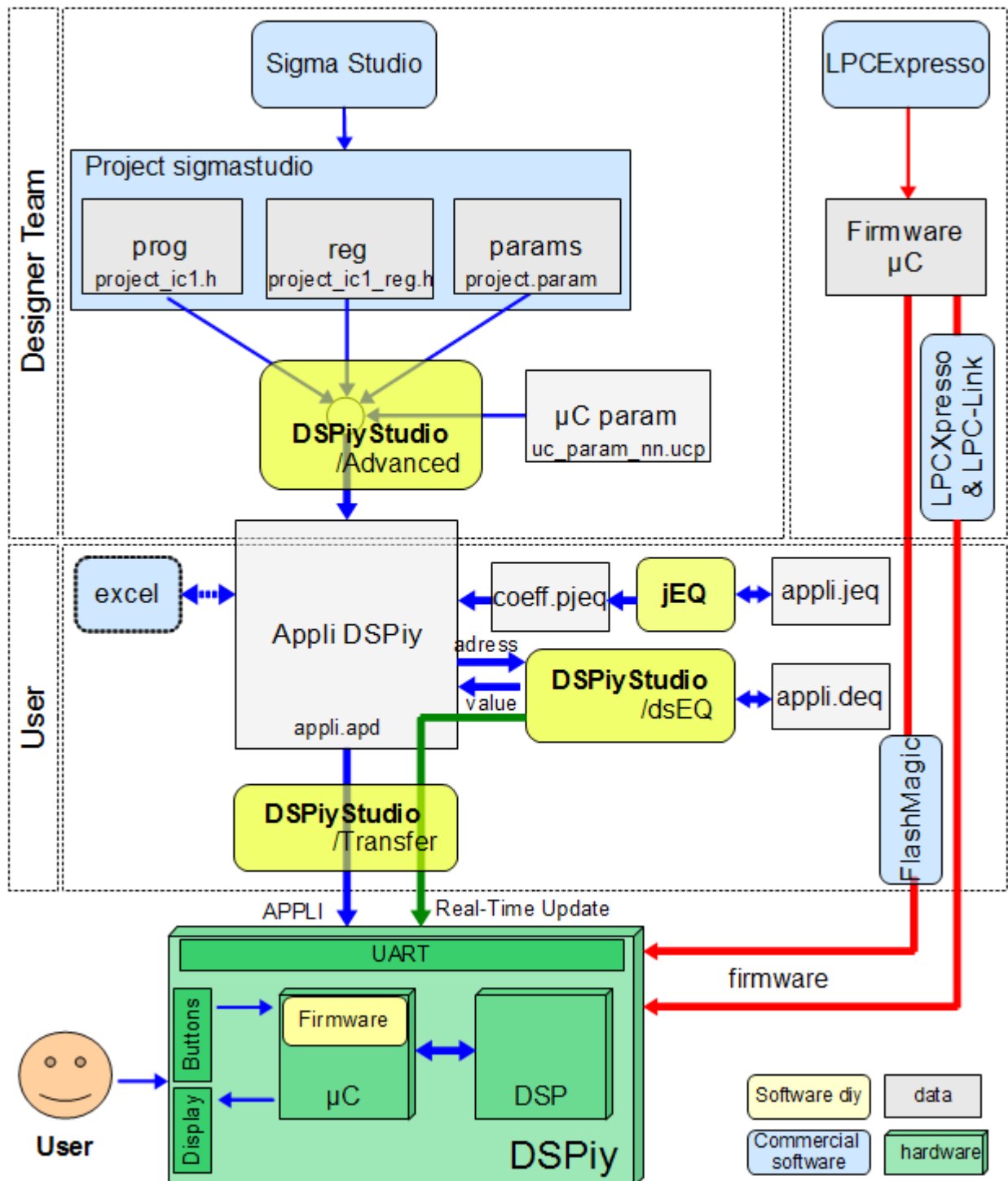
V1 du 17/02/2013 : création

v2 du 04/09/2014 : compléments sur le format du contenu Appli.apd et adaptation .params si composant inhabituel utilisé

1. Organisation et nommage

1.1 Architecture globale des logiciels pour le DSPiy

DSPiy Software block diagram



1.2. Fichiers

	Nom	Contenu	Créé avec
0	uC_firmware_nn.exe	Firmware pour le μ C (exécutable par lui)	LCPEXpresso
1	uC_param_nn.ucp	Paramètres par défaut pour le μ C	Éditeur texte
2	project_IC1.H	Programme DSP	SigmaStudio
3	project_IC1.REG.H	Adresses & valeurs par défaut des registres	SigmaStudio
4	project.params	Adresses & valeurs par défaut des paramètres DSP	SigmaStudio
5	APPLI.apd	Configuration complète d'une APPLI pour le DSPiY. modifié par l'utilisateur.	DSPiY studio
6	APPLI.deq	Description en termes humainement lisibles des paramètres audio de l'APPLI	DSPiYStudio

(0) firmware pour le micro-contrôleur. Développé par l'équipe de Design (thierryvalk essentiellement), il ne sera que peu modifié une fois la phase de développement et stabilisation terminée. Il est transféré dans le μ C où il devient un programme exécutable qui régit le fonctionnement du DSPiY.

Pour fonctionner, le DSPiY n'aura besoin que d'un firmware et d'un fichier Application.

Le DSPiY sera livré avec un firmware installé et une Appli de test. Cette dernière sera à remplacer à la première mise en marche.

L'utilisateur pourra installer un nouveau firmware dans le DSPiY grâce au logiciel Flashmagic (gratuit) depuis son pc via le port USB du DSPiY et/ou avec Dstudio à partir de la v3.

L'utilisateur avancé pourra aussi installer et déboguer le firmware grâce à lcp-expresso et une carte Lpc-link de NXP branchée sur le connecteur EMU du DSPiY. (carte disponible pour environ 30€ auprès de Lextronic).

Description succincte des fichiers

(1) **uC_param_nn.ucp** : paramètres pour le micro-contrôleur exploités dans le firmware. Il faudra s'assurer lors de la création d'une Appli de la compatibilité du fichier uc_params avec la version de firmware installé.

Créé par Thierryvalk, il ne sera pas modifié par l'utilisateur.

(2) (3) (4) **fichiers .h et .params** générés par SigmaStudio après compilation, par click de Export System Files.

Il s'agit d'un ensemble de fichiers par projet. Chaque modification d'architecture de projet dans SigmaStudio génère un contenu différent de ces fichiers.

Ces fichiers servent à créer l'APPLI initiale et ne sont pas modifiés par DSPiYStudio (lecture seule).

Avec un usage standard de SigmaStudio où le Designer fait un projet non modifiable par l'utilisateur, ces trois fichiers suffisent pour faire fonctionner le DSP.

Comme avec le DSPiy on rend le paramétrage modifiable par l'utilisateur, une approche différente a été choisie : on n'exploite que l'ossature (*l'architecture*) du projet SigmaStudio qui est insérée dans un fichier APPLI. Les *valeurs* des filtres quant a elles ne sont pas saisies dans SigmaStudio mais sont calculées et insérées par un logiciel tiers, comme DSPiyStudio (onglets Filtres et Crossover) ou jEQ.

(5) fichier **Application APPLI.apd** (aussi dénommé Application ou APPLI)

Ce fichier contient l'intégralité d'une configuration du DSPiy.

Il est formé de 4 sections :

- paramètres du micro-contrôleur (la configuration)
- paramètres DSP, reformatage du fichier mon_projet.params créé par SigmaStudio
- registres DSP , reformatage du fichier my_project_IC1_REG.H créé par SigmaStudio
- programme DSP, partie data du fichier my_project_IC1.H créé par SigmaStudio

Chaque section dans ce fichier, sauf la partie programme, a le même format : Nom Adresse Valeur.

Il est créé par *DSPiyStudio* (onglet Avancé) ou fourni par l'équipe de Design. Il contient alors les valeurs par défaut de la configuration produite par l'équipe de Design.

Il peut être modifié par *DSPiyStudio* lui même (onglets dsEQ) ou par *jEQ* ou par import/export dans un tableur ou même modifié manuellement si l'utilisateur est expert.

Transféré au DSPiy, c'est le μ Contrôleur qui se charge de mettre au bon endroit les données dans la RAM du DSP.

Une ou plusieurs APPLIs peuvent aussi être enregistrées dans la mémoire flash et devenir des Presets. Le micro-contrôleur saura charger un Preset dans la RAM du DSP sur appui de bouton ou de télécommande par exemple.

DSPiyStudio modifie seulement les valeurs dans le fichier Appli. Il sera nécessaire de créer une routine qui va vérifier l'intégrité du fichier APPLI.

De même il faudra bien prévenir les utilisateurs qu'en phase de réglage, il y a un risque pour les tweeter et que nous ne sommes pas responsables.

(6) Fichier **Appli.deq**

C'est un fichier utilisé par Dstudio pour décrire l'application en langage humainement compréhensible. C'est le fichier *source*. Il est enregistré en même temps que le fichier APPLI .apd et contient les descriptions des traitements en termes utilisateur, alors que dans le fichier APPLI.apd les descriptions sont dans le format pour le DSP.

Par exemple : pour un filtre biquad de type PEQ, à 1000Hz, gain 5,2 et Q=5,2

APPLI.deq enregistre :

```
GLOBALBIQUAD_1_1_F 1000
GLOBALBIQUAD_1_1_Q 5,2
GLOBALBIQUAD_1_1_G 5,2
GLOBALBIQUAD_1_1_T PEQ
```

alors que dans APPLI.pad on trouvera dans la section Paramètres DSP :

Exemple :

```
EQ1940Single10B1 5 1,00365306543459
EQ1940Single11B1 6 -1,98635040364354
EQ1940Single12B1 7 0,986959392283122
EQ1940Single11A1 8 -1,98635040364354
EQ1940Single12A1 9 0,99061245771771
```

Pour un filtre de type Linkwitz-Riley à 24dB consommant 4 filtres sigmastudio, on trouverait 4x5lignes dans appli.apd pour seulement une description dans APPL.deq.

En réouvrant une application, c'est grace à ce fichier .deq que DS peut renseigner les différents champs dans les onglets.

2. Structure fichier Appli.apd

Comme dit précédemment, Il est formé de 4 sections avec une entete qui contient quelques informations d'ordre général.

- paramètres uc : liste des paramètres du micro-contrôleur (la configuration) sous la forme Nom Adresse Valeur
- paramètre DSP, reformatage du fichier mon_projet.params créé par SigmaStudio. sous la forme Nom Adresse Valeur , adresse et valeur en décimal
- registres DSP , reformatage du fichier my_project_IC1_REG.H créé par SigmaStudio. sous la forme Nom Adresse Valeur , adresse et valeur en hex
- programme DSP, partie data du fichier my_project_IC1.H créé par SigmaStudio. Du Hex
-

Attardons nous sur la section Paramètres DSP.

Prenons un exemple.

Dans le fichiers .params généré par SigmaStudio, un biquad aura la forme en 5 blocs:

```
Cell Name      = Gen Filter2
Parameter Name = EQ1940Single20B1
Parameter Address = 134
Parameter Value = 0,4001829624176025390625
Parameter Data :
0x00, 0x33, 0x39, 0x32
```

```
Cell Name      = Gen Filter2
```

Parameter Name = EQ1940Single21B1
Parameter Address = 135
Parameter Value = -0,4001829624176025390625
Parameter Data :
0xFF, 0xCC, 0xC6, 0xCE

Cell Name = Gen Filter2
Parameter Name = EQ1940Single22B1
Parameter Address = 136
Parameter Value = 0
Parameter Data :
0x00 , 0x00 , 0x00 , 0x00 ,

Cell Name = Gen Filter2
Parameter Name = EQ1940Single21A1
Parameter Address = 137
Parameter Value = 0
Parameter Data :
0x00 , 0x00 , 0x00 , 0x00 ,

Cell Name = Gen Filter2
Parameter Name = EQ1940Single22A1
Parameter Address = 138
Parameter Value = 0
Parameter Data :
0x00 , 0x00 , 0x00 , 0x00 ,

Dans le fichier APPLI.APD, ce même biquad devient 5 lignes. C'est plus compact !

```
EQ1940Single20B1 134 0,000264395749179163  
EQ1940Single21B1 135 0,000528791498358326  
EQ1940Single22B1 136 0,000264395749179163  
EQ1940Single21A1 137 -1,97420999113081  
EQ1940Single22A1 138 0,975267574127527
```

La forme est Nom Adresse Valeur

- Le Nom est celui du Parameter Name.
- L'adresse est Parameter Adress
- La valeur est Parameter Value, qui est la conversion en décimal de la valeur Hexadécimale de Parameter Data.

Pour détailler le nom de ce biquad, la forme c'est : EQ1940 Single 2 0B 1

- EQ1940 pour un biquad générique dans SigmaStudio
- Single je sais pas pourquoi

- 2 est le numéro de la cellule dans le projet
- puis 0B, 1B, 2B, 1A, ou 2A qui sont les 5 coefficients habituels d'un biquad dans l'ordre
- 1 qui est le numéro de séquence de ce biquad dans la cellule. Une cellule comporte plusieurs biquads en série.

A propos de la conversion de Parameter Data de Hexa à décimal

Le DSP du DSPiY fonctionne en format 5.23 (Un nombre est représenté avec 5 chiffres pour la partie entière et 23 décimales)

Pour trouver parameter value, il faut d'abord convertir l'hex en décimal, puis diviser par 2^{23} , puis si le nombre obtenu est >256 , retirer 512.

Exemple :

parameter data : 0x00, 0x33, 0x39, 0x32

devient en Hexa : 00333932

converti en décimal = 3356978 (avec la calculatrice windows par exemple)

divisé par 2^{23} (8388608) : $3356978/8388608 = 0,4001829624176025390625$

autre exemple :

parameter data : 0xFF, 0xCC, 0xC6, 0xCE

FFCC6CE en décimal = 4291610318

$4291610318/2^{23} = 511,5998170375823974609375$

Comme > 256 on retire 512

$511,5998170375823974609375 - 512 = -0,4001829624176025390625$

Quelques soucis avec certaines cellules SigmaStudio

Quand on sort des sentiers battus, et qu'on utilise des composants non prévus initialement il peut arriver que SigmaStudio génère dans le .params des informations indigestes pour DStudio !

DStudio v3 lors de la fabrication de l'appli.apd s'attend à trouver dans .params des blocs bien formés, avec une seule valeur.

Or le composant Loudness par exemple est généré ainsi :

```

Cell Name      = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi
Parameter Address = 4
Parameter Data :
0x00, 0x33, 0x39, 0x32,
0xFF, 0xCC, 0xC6, 0xCE,
0x00, 0x33, 0x39, 0x32,
0x00, 0x00, 0x15, 0x6E,
0x00, 0x7F, 0xEA, 0x92,

```

Pour que la fabrication d'appli fonctionne, il faut donc reformater le .params à la main, en découpant cette description de cellule en 5 blocs séparés .

Ce qui devient : (Notez le nommage modifié de chaque Parameter Name et l'incrémentation de Parameter Address, en plus de l'ajout de Parameter Value converti en décimal 5.23 selon méthode décrite précédemment)

Cell Name = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi1
Parameter Address = 4
Parameter Value = 0,4001829624176025390625
Parameter Data :

0x00, 0x33, 0x39, 0x32,

Cell Name = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi2
Parameter Address = 5
Parameter Value = -0,4001829624176025390625
Parameter Data :

0xFF, 0xCC, 0xC6, 0xCE,

Cell Name = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi3
Parameter Address = 6
Parameter Value = 0,4001829624176025390625
Parameter Data :

0x00, 0x33, 0x39, 0x32,

Cell Name = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi4
Parameter Address = 7
Parameter Value = 0,0006539822
Parameter Data :

0x00, 0x00, 0x15, 0x6E,

Cell Name = Loudness L&H1
Parameter Name = LoudnessLH1001iir_coeff_bo_hi5
Parameter Address = 8
Parameter Value = 0,9993460178
Parameter Data :

0x00, 0x7F, 0xEA, 0x92,

4. Section Parametre DSP (issu du Fichier uC_Param.ucp)

Format de la version b1 du firmware. Chaque nouvelle version apporte ses nouveautés, a suivre sur le site dspiy.be

La plupart des valeurs sont éditables dans l'onglet

On regrettera que le format de fichier ne permette pas de mettre en commentaire la description sur chaque ligne. Un jour peut être;

Contient les noms de paramètres adresse et valeurs par défaut pour toutes les fonctions du μ C.

La ou les premières lignes comprendront l'entête avec des infos telles que versions des fichiers, date et heure de création ...

Pour faciliter les évolutions futures tous les noms commencent par DO1_

Tous les noms en majuscule. La première valeur est la valeur par défaut.

Nom	adresse	Valeurs	commentaires	nom repris dans le fichier
DO1_VOLUME	1	0	niveau max -0db	
		255	niveau min -80db	
DO1_VOLUME_CTL	2	0	pas de contrôle de volume	
		1	potentiomètre sur BP1	
		2	encodeur sur BP1 et BP2	
		3	volume + sur BP1 volume – sur BP2	
DO1_VOLUME_MAX	3	0	limite du volume : niveau max -0db 255 niveau min -80db	
DO1_VOLUME_ON	4	0	volume à la mise sous tension : 0 dernier utilisé	
		1	volume de DO1_VOLUME (on ne mémorise pas le volume)	
DO1_BOUTONP1	5	0	pas d'action (pas utilisé ou utilisé pour commande volume)	
		1-30	charge nouveau APPLI	
		253	charge APPLI précédent (si existe)	
		254	charge APPLI suivant (si existe)	
		255	Mute / UnMute	
Idem pour BP2 à BP5 et Iso_in 6 -> 15 (réserve)				
DO1_RELAI	16	0	off	
		1	on	
DO1_ISO_OUT	17	0	off	
		1	on	
DO1_LED1	18	0	off	
		1	on	

2 volume

Idem led2 led3 + réserve

Do1_IR_CODE 27 0 pas de télécommande

1 Code Sony

2 Code RC5 (philips)

Do1_IR_ADR 28 0->255 adresse IR

Do1_IR_SUB_ADR 29 0->255 sub-adresse IR (selon codage)

Do1_IR_USER_PLUS 30 0->255 code pour charger APPLI suivant (si existe)

Do1_IR_USER_MOIN 31 0->255 code pour charger APPLI précédent (si existe)

Do1_IR_USER_1 32 0->255 code pour charger APPLI1

Idem pour APPLI 2 à APPLI 10

Do1_IR_MUTE 33 0->255 code pour mute / un_mute

Do1_IR_VOLUME_PLUS 34 0->255 code pour augmenter volume (sauf pour potentiomètre)

Do1_IR_VOLUME_MOIN 35 0->255 code pour diminuer volume (sauf pour potentiomètre)

Do1_IR_OFF 36 0->255 code pour mise en standby

Do1_IR_ON 37 0->255 code pour mise sous tension.

Réserve

Do1_DSP 58 0->4 sélection du DSP

Do1_USERPROG_ON 59 0 chargement dernier APPLI à la mise sous tension

1->10 numéro d' APPLI

Infos firmware

DSP_Write : écrit un ou plusieurs paramètres depuis le μ C vers la RAM du DSP.

DSP_Write_Safe : écrit un paramètre de manière synchrone avec le traitement des samples, afin de ne pas créer des plocs. On le voit à l'écran du DSPiY quand on modifie un paramètre dans l'onglet Filtres. (affichage fugitif de W safe)

Write_flash : écriture de la RAM du DSP dans la mémoire flash (cas des Preset)