

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY

UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2018

Faculté: Sciences de l'Ingénierat

Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :

Implémentation sur un DSP d'un filtre numérique passe-bas en utilisant la carte « TMS320C6713 platform ».filtrage des signaux ECG

Domaine : Sciences et Technologie

Filière : Electronique

Spécialité : Electronique des système Embarqué

Par :

Teguig Oussama

DEVANT Le JURY

Président : S.TOUMI

Grade UBM Annaba

Directeur de mémoire: R.HAMDI

Grade UBM Annaba

Examineurs: M.BOUTALBI

Grade UBM Annaba

M.FRIHI

Grade UBM Annaba

SOMMAIRE

REMERCIEMENTS	1
SOMMAIRE	2
GLOSSAIRE ET ABREVIATIONS.....	5
LISTE DES FIGURES	6
LISTE DES TABLEAUX.....	7
RESUME	8
INTRODUCTION GENERALE.....	8
CHAPITRE 1 – CONCEPTS DE BASE DES DSP	9
1.1 INTRODUCTION.....	10
1.2 AVANTAGES DES SYSTEMES A BASE DE DSP.....	11
1.3 DIFFERENCE ENTRE DSP ET MICROPROCESSEURS ORDINAIRES	12
1.4 TYPES ET FORMATS DES DONNEES MANIPULEES PAR LES DSP	13
1.4.1 DSP A VIRGULE FIXE.....	14
1.4.2 DSP A VIRGULE FLOTTANTE	15
1.5 SELECTION DU DSP LE PLUS ADAPTE	16
1.6 PRODUITS DE TEXAS INSTRUMENTS	17
1.8. La carte DSK TMS320C6713	18
A. INTRODUCTION	19
B. L'ARCHITECTURE DU TMS320C6713	20
C. UNITE CENTRALE DE TRAITEMENT (CPU)	21
D. LES PERIPHERIQUES DU TMS320C6713	22
E. LA STRUCTURE DE LA MEMOIRE	23
F. TMS320C6713 DSK (C6713 DSK)	24
G. ORGANISATION DE LA MEMOIRE DU C6713 DSK	25
CHAPITRE 2 – LES SIGNAUX ECG.....	26
1.Introduction.....	27
2.Electrocardiographie.....	28
2.1 Anatomie Cardiaque.....	29
2.2 Système de conduction électrique.....	30
3.Principe de l'électrocardiogramme.....	31
CHAPITRE 3 – LE FILTRE RIF	32
1. Introduction.....	33
2. Signal échantillonné.....	34
3. Filtres à réponse impulsionnelle finie.....	35
CHAPITRE 4-IMPLEMENTATION D'UN FILTRE FIR PASSE BAS DANS LA CARTE TMS320C7613 ..	36
1.Le Code Composer Studio.....	37
2.Lancement du CCS.....	38
3 .Programmation en C	39
4 .Création d'un filtre FIR passe-bas	40
5 .Implémentation du filtre sur le DSP	41
6.Exemple filtrage d'un signal ECG	42
CONCLUSION	43
ANNEXES	44
BIBLIOGRAPHIE	45
WEBOGRAPHIE	46

Résumé

DSP signifie selon le contexte Digital Signal Processor ou Digital Sound Processing. Un DSP est un processeur spécialisé dans le traitement numérique du signal. Son architecture est optimisée pour traiter une grande quantité de données en parallèle à chaque cycle 'horloge. Ce mode de fonctionnement est très efficace pour traiter des signaux numériques (filtrage, compression, extraction, etc.) comme de la vidéo ou de la musique.

Les DSP sont utilisés dans la plupart des applications du traitement numérique du signal en temps réel. On les trouve dans les modems (modem RTC, modem ADSL), les téléphones mobiles, les appareils multimédia (lecteur MP3), les récepteurs GPS... Ils sont également utilisés dans des systèmes vidéo, les chaînes de traitement de son, partout où l'on reçoit un signal que l'on doit modifier à l'aide du filtrage.

Ce projet de fin d'étude a pour but l'étude de l'architecture des DSP et plus particulièrement celui de TEXAS INSTRUMENT, le TMS320C6713, ainsi que l'implémentation sur la carte d'évaluation TMS320C6713DSK d'un filtre numérique à réponse impulsionnelle finie, afin de pouvoir étudier de façon pratique ce filtre.

Introduction générale

L'utilisation des techniques numériques a nettement pris le pas sur les techniques analogiques dans de nombreux domaines, parmi lesquels on peut citer,

- le traitement du signal classique (filtrage, transformée de Fourier rapide, génération de signaux,...),
- les télécommunications (codage-décodage, modulation-démodulation, égalisation adaptative, annulation d'écho, cryptage,...),
- le traitement de la parole (codage-compression, analyse, reconnaissance, synthèse...) et des images (codage-compression, reconnaissance de formes...),
- le radar (poursuite multimode, traitement anti réverbération, identification de cibles,...),
- les applications médicales : traitement de signaux EEG, EMG,..., imagerie biomédicale dans Les RMN...),
- et la commande (industrielle, avionique,...), etc.

Les exigences contradictoires de très basse consommation d'énergie et d'accroissement de la fonctionnalité du TNS ainsi que la rapidité de transfert et de traitement des données ont conduit à un certain nombre d'avancées dans les algorithmes, les technologies des semi-conducteurs et les architectures de système qui ont abouti au développement d'un système DSP (Digital Signal Processor) faisant partie de la famille des microprocesseurs, mais dont les jeux d'instructions et l'architecture le rendent performant dans le domaine du traitement numérique du signal.

D'un point de vue économique, on peut être assuré d'une croissance importante du marché des DSP eu égard les équipements dans lesquels ils

sont présents,

- les téléphones mobiles sous forme de circuits spécialisés à cœur de DSP,
- les modems sous une forme similaire,
- les terminaux DSL (Digital Subscriber Line), HDSL, ADSL... dans lesquels ils assurent l'égalisation, le brouillage, l'annulation d'écho, la suppression de la télédiaphonie, etc.,
- la télévision haute définition (TVHD) (codage-décodage du son et de l'image),
- la radiodiffusion numérique (DAB pour Direct Broadcast Audio),
- les DVD (Digital Video Disks) pour le décodage son Dolby AC-3 et le décodage vidéo MPEG-2,
- le contrôle des moteurs à courant alternatif,
- et le contrôle des disques durs : utilisation de techniques à maximum de vraisemblance (PRML Partial Response, Maximum Likelihood) pour augmenter la densité d'enregistrement, etc.

Les objectifs principaux de ce travail se résument aux nombres de deux,

- Découverte en pratique de l'architecture des processeurs de signaux (DSP) et les problèmes liés à leur programmation en implantant des algorithmes simples sur un processeur 32 bits à virgule flottante (TMS320C6713 DSK).

- Complètement des connaissances en traitement numérique du signal en implémentant un algorithme d'un filtre FIR passe-bas sur le DSP, objet de notre étude, ainsi que la vérification en pratique des résultats théoriques.

Le document est organisé de la manière suivante. Nous introduisons dans le premier chapitre des notions théoriques sur les processeurs dédiés au traitement du signal. Le second chapitre est consacré à la description de la carte TMS320C6713 DSK à base de DSP et qui sera utilisée pour implémenter

le filtre numérique. Nous présentons dans le troisième chapitre les deux modèles de filtrage numérique passe-bas que nous avons conçu et développé : le premier est le modèle MATLAB qui sert de référence et le deuxième est le modèle pratique pour DSP. Des résultats sont également présentés au niveau du troisième chapitre. Nous terminons ce document par une conclusion et des perspectives de ce travail.

Chapitre 1 – Concepts de base des DSP

1. Introduction

Les DSP sont utilisés dans de nombreux domaines d'applications comme cité au niveau de l'introduction générale. La façon dont un DSP s'insère dans une chaîne de traitement du signal est illustrée dans la figure 1.1. Cette chaîne comporte généralement trois étapes,

- la conversion analogique numérique,
- le traitement pour le DSP,
- et la conversion numérique analogique.

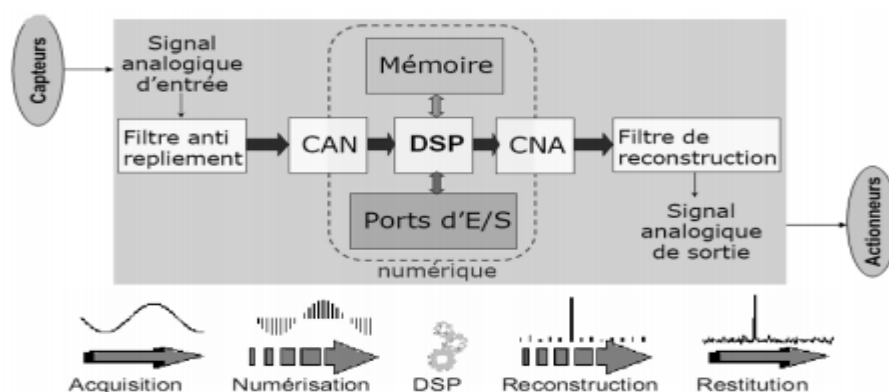


Figure 1.1 - Chaîne complète typique d'un système de traitement numérique de signal.

L'étape de conversion analogique-numérique a pour but de réaliser les opérations d'échantillonnage et de quantification du signal à traiter. Le DSP effectue le traitement désiré : filtrage, détection d'un signal, estimation, régulation, modulation, ... Le signal

traité est alors de nouveau converti (si nécessaire) en signal analogique lors de l'étape de conversion numérique-analogique.

Les étapes d'entrée et de sortie peuvent elles-mêmes être décomposées en plusieurs étapes. La première étape de conversion analogique-numérique est généralement un filtrage analogique effectué par un filtre anti repliement. Son but est de filtrer le signal qui doit être traité par le DSP de façon à ce que le critère d'échantillonnage de Shannon soit satisfait : le spectre du signal ainsi filtré exhibe donc une amplitude très faible pour les fréquences supérieures à la moitié de la fréquence d'échantillonnage : on évite ainsi le phénomène de recouvrement de spectre. L'échantillonnage est alors effectué en sortie de ce filtre. De façon pratique, il ne peut être réalisé à l'aide d'un Peigne de Dirac. On utilise un échantillonneur constitué par une suite périodique d'impulsions rectangulaires de durée Ω et de période T_e . L'échantillonnage est alors réalisé par un système bloquant son signal d'entrée pendant Ω . Généralement on choisit $\Omega = T_e$ pour des raisons de simplicité et le système s'appelle un échantillonneur bloqueur. En sortie de l'échantillonneur bloqueur, on place alors un convertisseur analogique-numérique dont le but est d'effectuer l'opération de quantification.

En sortie du DSP, on place un convertisseur numérique-analogique qui fournit en sortie un signal analogique de type fonction en escalier. En effet celui-ci se contente généralement de maintenir pendant une période d'échantillonnage la valeur qu'il a en entrée : on obtient ainsi un bloqueur d'ordre 0. Afin « d'adoucir » ce signal analogique, on place en sortie de ce convertisseur un filtre dit filtre de reconstruction [4].

2. Avantages des systèmes à base de dsp

Tous les systèmes à base de DSP bénéficient des avantages suivants,

- **Souplesse de la programmation** : un DSP est avant tout un processeur exécutant un programme de traitement du signal. Ceci signifie que le système bénéficie donc d'une grande souplesse de développement. De plus, les fonctions de traitement numérique peuvent évoluer en fonction des mises à jour des programmes, et cela pendant toute la durée de vie du produit incluant le système. La modification d'un filtre numérique ne nécessite pas un changement matériel.
- **Des possibilités propres au système de traitement numérique du signal**: certaines fonctions de traitement du signal sont difficiles à implanter en analogique, voire

irréalisables (exemple : un filtre à réponse en phase linéaire).

- **Stabilité** : en analogique, les composants sont toujours plus ou moins soumis à des variations de leurs caractéristiques en fonction de la température, de la tension d'alimentation, du vieillissement, etc. Une étude sérieuse doit tenir compte de ces phénomènes, ce qui complique et augmente le temps de développement. Ces inconvénients n'existent pas en numérique.

- **Répétitivité, reproductibilité** : les valeurs des composants analogiques sont définies avec une marge de précision plus ou moins grande. Dans ces conditions, aucun montage analogique n'est strictement reproductible à l'identique, il existe toujours des différences qu'il convient de maintenir dans des limites acceptables. Un programme réalisant un traitement numérique est par contre parfaitement reproductible, « à l'infini » [1].

3. Différence entre DSP et microprocesseurs ordinaires

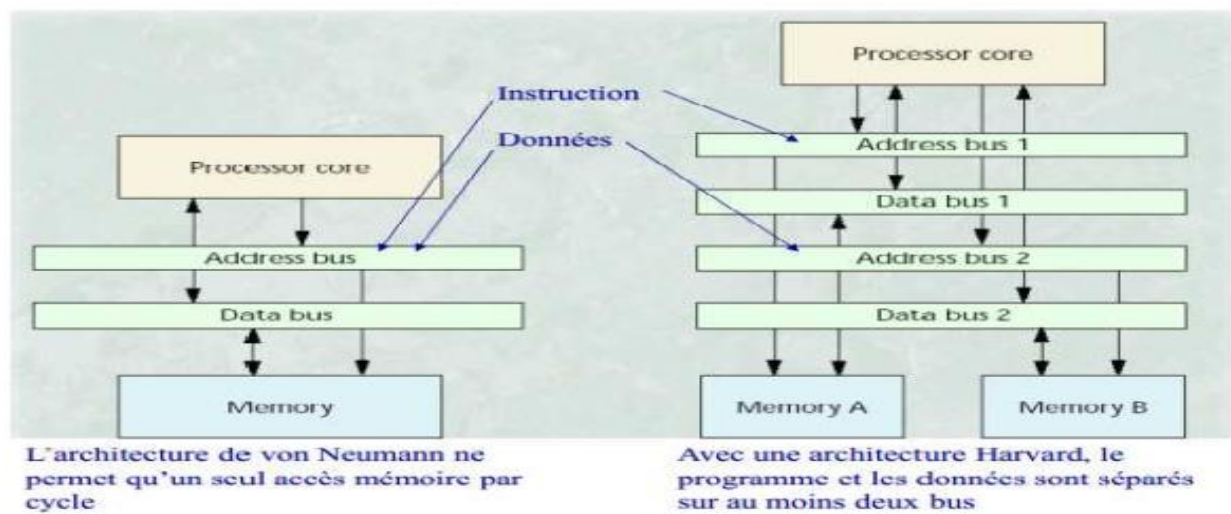
La différence fondamentale se situe sur le simple fait qu'un microprocesseur n'est pas destiné à une application spécifique alors que le DSP est destiné au traitement du signal, ce qui lui permet d'avoir une architecture optimisée pour ces types de traitement.

Généralement les principales distinctions se résument au nombre de deux,

- Contrairement aux microprocesseurs classiques, la plupart des DSP ont un jeu d'instructions spécialisé permettant de lire en mémoire une donnée, d'effectuer une multiplication puis une addition, et enfin d'écrire en mémoire le résultat, le tout en un seul cycle d'horloge. Ce type d'opération est nommé MAC, en anglais Multiply and Accumulat.

- Une autre particularité des DSP par rapport aux microprocesseurs ordinaires, c'est qu'ils ont une architecture Harvard, ceci est en opposition à celle de Von Newman.

Dans l'architecture Harvard, la mémoire des données et celle du programme sont



Physiquement différentes. Ces deux mémoires sont accessibles par l'intermédiaire de bus différents. Cette configuration interne permet d'avoir simultanément accès à la mémoire et d'exécuter une instruction. La vitesse d'exécution d'un programme s'en trouve donc doublé (en théorie). La figure 1.2 illustre la différence entre les deux architectures : Von Newman et Harvard.

Figure 1.2 - Architecture de Von Neumann vs architecture de Havard.

Outre l'opération MAC et l'architecture Havard, la plupart des DSP (dont le TMS320C6713 qui fait objet de notre étude) ont d'autres particularités importantes par rapport aux processeurs généraux (Intel, Cyrix, AMD9...), elles sont en général [1],

- Un adressage circulaire, permettant de ne pas faire des tests de modulo sur les pointeurs de tampons.
- Un mode d'adressage par inversion de bits, servant à réorganiser les échantillons de sortie de Transformée de Fourier Rapide (TFR ou FFT).
- De la mémoire interne minimisant les temps d'accès RAM.
- Des banques de mémoire, permettant de scinder par exemple partie réelle et partie imaginaire.
- Une architecture à pipe-line permettant de paralléliser le maximum d'opérations élémentaires (pré-chargement d'instruction, chargement des données, exécution d'opérations arithmétiques, stockage des résultats).
- Parfois des convertisseurs analogiques numériques et numériques analogiques.

- Certains DSP utilisent l'architecture dite « Structure de Harvard modifiée » (pour réduire le coût de la structure Harvard). À l'extérieur, le DSP ne propose qu'un bus de données et un bus d'adresse, comme la structure Von Neumann. Toutefois, à l'intérieur, la puce DSP dispose de deux bus distincts de données et de deux bus distincts d'adresses. Le transfert des données entre les bus externes et internes est effectué par multiplexage temporel (c'est le cas du TMS320C6713).

4. DSP à virgule fixe

Les DSP à virgule fixe : les données sont représentés comme étant des nombres fractionnaires à virgule fixe, (exemple -1.0 à +1.0), ou comme des entiers classiques. La représentation de ces nombres fractionnaires s'appuie la méthode du « complément à deux ». L'avantage de cette représentation (qui n'est qu'une convention des informaticiens) est de permettre facilement l'addition binaire de nombres aussi bien positifs que négatifs.

5. DSP à virgule flottante

Les données sont représentées en utilisant une mantisse et un exposant. La représentation de ces nombres s'effectue selon la formule suivante : $n = \text{mantisse} \times 2^{\text{exposant}}$. Généralement, la mantisse est un nombre fractionnaire (-1.0 à +1.0), et l'exposant est un entier indiquant la place de la virgule en base 2 (c'est le même mécanisme qu'en base 10).

La très grande dynamique proposée par les DSP à virgule flottante (comme TMS320C6713) permet virtuellement de ne pas se soucier des limites des résultats calculés lors de la conception d'un programme. Cet avantage a cependant un prix, à savoir qu'un système basé sur un DSP à virgule flottante a un coût de fabrication supérieur par rapport à un système basé sur DSP à virgule fixe. La puce d'un DSP à virgule flottante nécessite à la fois une surface de silicium plus importante (cœur plus complexe), et un nombre de broches supérieur, car la mémoire externe est elle aussi au format 32 bits. Le système revient donc plus cher (exemple : 2 x 32 broches juste pour les bus de données externes avec une architecture Harvard de base)

6. Sélection du DSP le plus adapté

La sélection d'un DSP se base avant tout sur la puissance de traitement nécessaire. Toutefois, la performance du DSP n'est pas le seul critère à prendre en compte, il faut

également tenir compte des impératifs suivants,

- Le type de DSP à utiliser (virgule fixe ou flottante) en fonction du domaine d'application.
- Les ressources mémoires utilisés.
- Les besoins d'un ou de plusieurs timers internes, de ports série synchrones ou asynchrone, etc.
- La nécessité éventuelle d'exécuter un système temps réel, qui s'avérera plus facile à implanter sur certains DSP.
- Le coût du DSP, son rapport « performance/prix » en fonction du volume de production envisagé.

D'autres éléments non négligeables interviennent dans le choix d'un DSP, il s'agit des moyens disponibles pour mener le développement en un temps donné, comme

- La qualité de la documentation (de préférence claire et abondante).
- La disponibilité de notes d'applications, d'un support technique.
- La qualité du système de développement utilisé.
- La possibilité d'utiliser un langage de haut niveau (Langage C).
- La présence de bibliothèques (du constructeur ou de tierces parties).
- La possibilité de réaliser facilement des prototypes et à faible coût.

Le choix n'est pas toujours simple et certains critères peuvent être contradictoires, certaines règles de choix se dégagent quand même. Ainsi pour des applications destinées à faire un fort volume de production, le critère déterminant est sans conteste le prix du DSP. Pour des applications à faible volume de production, le prix du DSP importe peu, le critère est alors la facilité de développement.

Dans tous les cas, la présence d'un bon support technique est un facteur à ne pas négliger, car un DSP est quand même plus complexe à mettre en œuvre qu'un microprocesseur classique .

7. produits de texas instruments

La famille de processeurs la plus répandue actuellement est sans conteste celle des DSP de Texas Instruments qui détient environ 70 % du marché, les 30 % restant étant partagés entre Motorola, Analog Devices, Lucent Technologies, Nec et Oki.

Les familles les plus récentes des DSP de Texas instrument sont,

- TMS320C54x, DSP format fixe ;
- TMS320C20x, DSP format fixe;
- TMS320C24x, DSP format fixe;
- TMS320C62x, DSP format fixe à architecture VLIW ; rchitecture VLIW;
- TMS320C64x, DSP format fixe;

Ces nouvelles familles sont regroupées en trois classes appelées plates-formes. Ces trois classes sont appelées:

- TMS320C6000, formée des familles C62x, C67x et C64x;
- TMS320C5000, formée de la famille C54x et des C54xx;
- TMS320C2000, formée des familles C20x et C24x;

Le tableau suivant résume les principales caractéristiques de ces trois classes,

Application	Type de DSP	Caractéristique
TMS30C6000 DSP hautes performances		
C64x	Application au traitement numérique de signal	32 bits virgule fixe 24000 MIPS -3,0 GHz
C62x	Applications exigeantes en vitesse: stations de base des réseaux de communications mobiles, équipement de radiodiffusion, réseaux informatiques	16 bits virgule fixe architecture VLIW 1200-2400 MIPS

C67x	Applications exigeantes en précision, dynamique et vitesse: Antennes adaptatives des stations de base, imagerie médicale, reconnaissance de parole, graphisme...	32 bits virgule flottante architecture VLIW	600MFLOPS-1GFLOPS
TMS320C5000 DSP optimisés en consommation			
C54x	Applications de télécommunications exigeantes en coût, consommation, vitesse: terminaux mobile, voix sur IP, alphapages...	16 bits virgule fixe	0.54 mWLMIPS 30-200 MIPS
TMS320C2000 DSP optimisés pour les applications de contrôle			
C20x	Applications de grand volume en téléphonie, électronique grand public, appareils photos numériques ou contrôleurs de disques durs...	16 bits virgule fixe	PLL, UART, timers, mémoire flash intégrée 20-40MIPS
C24x	Applications de contrôle moteur, automatisation, robotique, contrôle d'appareils électroménagers...Bon compromis prix/performance	16 bits virgule fixe	port série SCI, SPI et CAN, convertisseur Analogique/numérique 20MIPS

Tableau I.3- Caractéristiques des trois classes des DSP de Texas Instruments.

8. La carte DSK TMS320C6713

A. introduction

La plateforme TMS320C6000 des processeurs de signaux numériques fait partie de la

famille TMS320 de TEXAS INSTRUMENTS. Elle comporte les processeurs TMS320C62x à arithmétique fixe et TMS320C67x à arithmétique flottante. Le TMS320C6713 est considéré comme le membre le plus performant de la catégorie C67x, son architecture VLIW lui permet le traitement par paquets de huit instructions en parallèle par huit unités fonctionnelles. La discussion dans ce chapitre se focalisera sur le processeur TMS320C6713. L'architecture et les périphériques qui lui sont associés seront également discutés.

B.L'architecture du Tms320c6713

Le TMS320C6713 est constitué de trois parties principales (figure suivante),

- l'unité centrale de traitement CPU,
- les périphériques,
- et la mémoire.

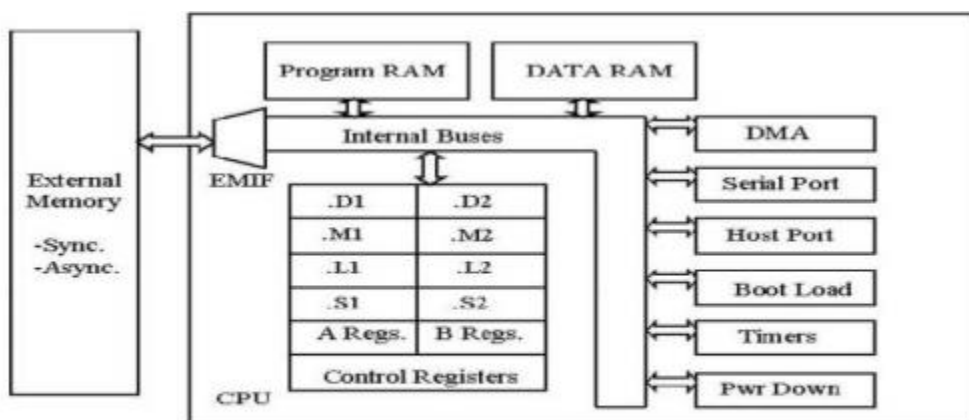


Figure 2.2 - Bloc diagramme simplifié de la famille TMS320C67xx.

C. Unité centrale de traitement (CPU)

Le CPU est constitué d'une unité de contrôle de programme, de deux unités fonctionnelles, de deux blocs de 16 registres de 32 bits, de contrôleurs d'interruptions et d'autres éléments.

C.1. Unité de contrôle de programme

Elle est constituée des éléments suivants,

- Unité "fetch" programme : Elle a pour rôle récupérer les programmes. Cette opération se déroule en quatre phases :
 - Phase PG: l'adresse du code est générée.
 - Phase PS : l'adresse est envoyée à la mémoire.
 - Phase PW: l'attente de lecture du code de la mémoire.

-Phases PR : la lecture du code.

- Unité "dispatche" de l'instruction: le code récupéré de la mémoire est affecté à l'unité fonctionnelle associée.

- Unité de décodage de l'instruction: elle a pour rôle de décoder l'instruction.

C.2. Unités fonctionnelles

Le CPU contient huit unités: fonctionnelles divisées en deux parties 1 et 2. Leurs fonctions sont les suivantes:

- Unités .M1 et .M2 : ces unités sont dédiées à la multiplication.

- Unités .L1 et .L2 : ces unités sont dédiées à l'arithmétique et la logique.

- Unités .D1 et .D2 : ces unités sont dédiées au chargement, la sauvegarde et calcul d'adresse.

- Unités .S1 et .S2: ces unités sont dédiées pour le décalage de bit, l'arithmétique, la logique et le branchement .

Functional Unit	Description
.L unit (.L1, .L2)	32/40-bit arithmetic and compare operations Left most 1, 0, bit counting for 32 bits Normalization count for 32 and 40 bits 32 bit logical operations 32/64-bit IEEE floating-point arithmetic Floating-point/fixed-point conversions
.S unit (.S1, .S2)	32-bit arithmetic operations 32/40 bit shifts and 32-bit bit-field operations 32 bit logical operations Branching Constant generation Register transfers to/from the control register file 32/64-bit IEEE floating-point compare operations 32/64-bit IEEE floating-point reciprocal and square root reciprocal approximation
.M unit (.M1, .M2)	16 x 16 bit multiplies 32 x 32-bit multiplies Single-precision (32-bit) floating-point IEEE multiplies Double-precision (64-bit) floating-point IEEE multiplies
.D unit (.D1, .D2)	32-bit add, subtract, linear and circular address calculation

Tableau 2.1- Description détaillée unités fonctionnelles du TMS320C6713

C.3. Registres

Le CPU contient 32 registres de 32 bits divisés en deux blocs égaux : registre fichier A (AO-A15) et registre fichier B (BO-B15), leurs fonctions sont réparties comme suit:

- Les registres A1-A2 et BO-B1-B2 : ils sont utilisés comme registres conditionnels.

- Les registres A4-A7 et B4-B7: ils sont utilisés pour adressage circulaire.

- Les registres AO-A9, BO-B2 et B4-B9 : ils sont utilisés comme registres temporaires.

- Les registres A10-A15 et B10-B15 : ils sont utilisés pour la sauvegarde et la restitution

de données d'un sous-programme.

À ces 32 registres s'ajoutent les registres de contrôles et d'interruptions.

D. Les périphériques du TMS320C6713

Le TMS320C6713 a plusieurs périphériques qui sont :

- Le contrôleur DMA : Il permet sans l'aide du CPU de transférer des données entre les espaces mémoire (interne, externe et des périphériques). Il a quatre canaux programmables et un autre canal auxiliaire.
- Le contrôleur EDMA : Il permet le transfert des données entre les espaces mémoire comme le DMA. Il a 16 canaux programmables.
- L'interface port hôte HPI. Il donne au processeur hôte un contrôle total pour un accès direct de l'espace mémoire du CPU et à la cartographie de la mémoire des périphériques du DSP.
- Deux McBSP qui sont des ports séries multi-canaux protégés. Ils permettent la communication avec les périphériques externes. Ils ont la même structure. Ils supportent une communication full-duplex.
- L'interface de mémoire externe EMIF : Il permet l'interface avec plusieurs éléments (mémoires) externes.
- Les compteurs : Le DSP possède deux compteurs qui peuvent être synchronisés par une source interne ou externe et ils sont utilisés comme générateurs de pulsations, compteurs d'événements externes, interrupteurs du CPU après l'exécution de tâches et déclencheur du DMA/EDMA.
- Les interruptions : l'ensemble des périphériques contient jusqu'à 32 sources d'interruptions.

E. La structure de la mémoire

Le TMS320C6713 basé sur l'architecture de Harvard modifiée utilise une mémoire externe et une mémoire interne (Figure 2.3).

- La mémoire externe occupe les espaces CEO, CE1, CE2, CE3.
- La mémoire interne a une taille de 260 KB qui est décomposée en deux niveaux :
 - Le niveau (L1) est constitué de deux mémoires caches de 4 KB chacune, (L1P) qui est utilisée pour les programmes et (L1D) qui est utilisée pour les données

□ Le Niveau (L2) est composé de 256 KB de mémoire partagée entre mémoire des données et mémoire de programmes.

F.

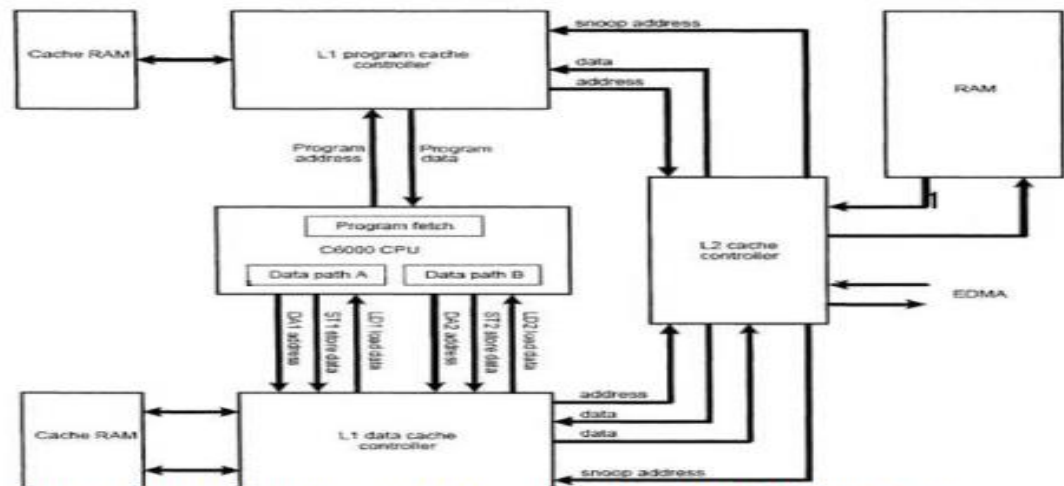


Figure 2.3- Organisation de la mémoire interne du TMS320C6713.

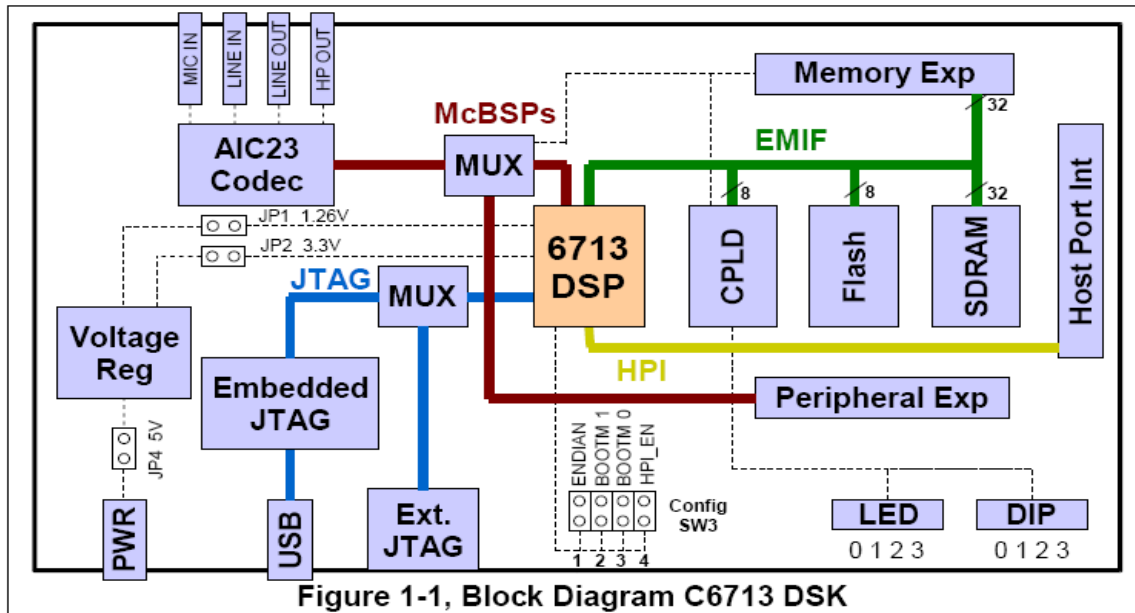
TMS320C6713 DSK (C6713 DSK)

La carte d'évaluation DSK est puissante, relativement peu coûteuse, avec les outils de support matériels et logiciels nécessaires pour le traitement du signal en temps réel.

Il s'agit d'un système DSP complet. Cette carte comprend le processeur de traitement numérique du signal, le TMS320C6713 à virgule flottante et un codec stéréo TLV320AIC23 (AIC23) de 32 bits pour l'entrée et la sortie (Figure 2.4). Le codec AIC23 « utilisant une technologie sigma-delta » fonctionne comme un CAN pour les entrées analogiques et comme un CNA pour les sorties numériques du DSP. Il se connecte à une horloge système de 12 MHz. Le taux d'échantillonnage variable de 8 à 96 kHz et peut être réglé facilement. La carte comporte un emplacement libre pour l'ajout d'un périphérique ou d'une carte additionnelle ("daughter card"), un emplacement pour ajouter de la mémoire, quatre LED ("Light-Emitting Diodes") et quatre DIP switches ("Dual In-line Pin") programmables.

La carte DSK comprend 16 MB de mémoire synchrone dynamique à accès aléatoire (DRAM) et 256 kB de mémoire flash. Le DSK fonctionne à 225 MHz, il intègre un régulateur de tension qui fournit 1,26 V pour le noyau C6713 et 3,3 V pour la mémoire et les périphériques.

Le DSK dispose également de quatre prises audio jacks de 3,5 mm, deux pour les entrées : microphone (mono) et "line in" (stéréo), et deux pour les sorties : "speaker" (stéréo) et "line out" (stéréo). En fait les deux entrées (respectivement les deux sorties) renvoient les signaux au même port physique, c'est-à-dire au même signal d'entrée (respectivement de sortie). La seule différence entre microphone et "line in" (respectivement "speaker" et "line out") réside dans les impédances des ports. Autrement dit, on a quatre prises audio, mais une seule entrée et une seule sortie, chacune disponible avec deux impédances différentes



9. Caractéristiques du DSK C6713

- DSP TMS320C6713 fonctionnant à 225 MHz.
- Codec stereo de 16 bits (AIC23)
- Quatre (4) connecteurs permettent des entrées et des sorties analogiques:
 - MIC IN : entrée de microphone,
 - LINE IN: entrée analogique "ligne"
 - LINE OUT: sortie analogique "ligne"
 - HEADPHONE: sortie écouteurs/casque (multiplexée avec l'entrée ligne).
- 16 Mbytes de SDRAM
- 512 Kbytes de mémoire flash (256 Kbytes pour une configuration par défaut)
- 4 LED et 4 interrupteurs (DIP switches)
- Configuration de la carte par programmation du CPLD
- Deux connecteurs de 80 broches permettent des extensions de périphériques et mémoire externes.
- Un émulateur JTAG intégré jumelé avec la connectivité USB

Address	C67x Family Memory Type	6713 DSK
0x00000000	Internal Memory	Internal Memory
0x00030000	Reserved Space or Peripheral Regs	Reserved or Peripheral
0x80000000	EMIF CE0	SDRAM
0x90000000	EMIF CE1	Flash
0xA0000000	EMIF CE2	CPLD
0xB0000000	EMIF CE3	Daughter Card

0x90080000

Cartographie de la mémoire (memory map) du DSK 6713

8 .Bloc diagramme du codec

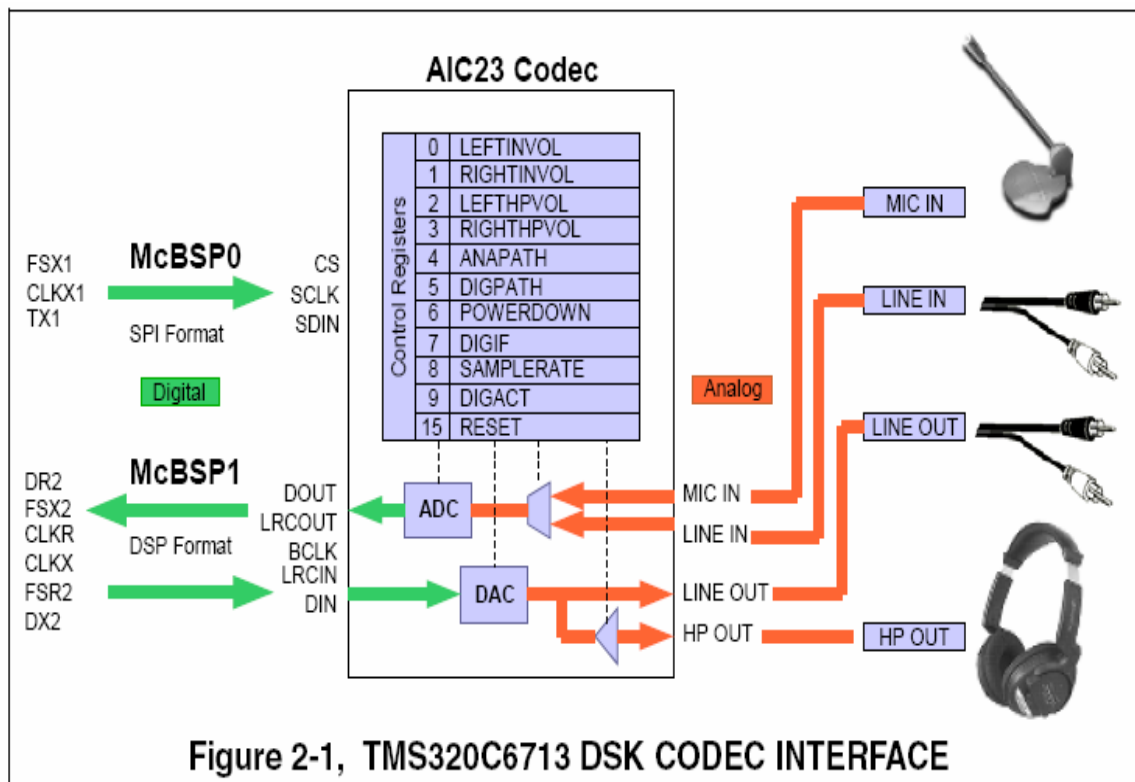


Figure 2-1, TMS320C6713 DSK CODEC INTERFACE

Chapitre 2 – les signaux ECG

1. Introduction

Les signaux physiologiques sont des signaux électriques délivrés par le corps humain au conséquent des activités faites par l'êtres humains lui-même, ils ont des caractéristiques comme tous les signaux électriques une amplitude (de rapport microvolts ou millivolts) et de dizaine de hertz (basse fréquence), il existe plusieurs sources de ces signaux le cœur, le cerveau et les muscle. On va parler dans ce chapitre sur les caractéristiques et les bruits qui influent sur ces signaux.

Nous présentons dans ce chapitre le fonctionnement général du système cardiovasculaire puis, de manière plus détaillée, le principe de l'électrocardiogramme (ECG). Cette présentation se limite au strict nécessaire pour une bonne compréhension du mémoire.

2. Electrocardiographie

Le cœur est l'élément central du système cardiovasculaire. Nous décrivons dans la suite du chapitre l'anatomie et le fonctionnement électrique d'un cœur sain.

Le potentiel électrique généré par l'activité musculaire est connu depuis les travaux de Carlo Matteucci en 1842. Les premières expérimentations sont réalisées en 1878 par John Burden Sanderson et Frederick Page qui détectent à l'aide d'un électromètre capillaire les phases QRS et T, définies par la suite. A cette même période, E. Marey, et Augustus Waller montrent que l'activité électrique cardiaque peut être suivie à partir de la peau. En 1887, le premier électrocardiogramme humain est publié par Augustus Waller. En 1895, Willem Einthoven, met en évidence les cinq déflexions P, Q, R, S et T. Il utilise le galvanomètre à cordes en 1901 et publie les premières classifications d'électrocardiogrammes pathologiques en 1906. Il obtiendra en 1924 un prix Nobel pour ses travaux sur l'électrocardiographie. Les dérivations précordiales sont utilisées pour le diagnostic médical à partir de 1932 et les dérivations frontales unipolaires à partir de 1942, ce qui permet à Emanuel Goldberger de réaliser le premier tracé sur 12 voies. Aujourd'hui, l'électrocardiographie est une technique relativement peu coûteuse, permettant à l'aide d'un examen indolore et sans danger, de surveiller l'appareil cardiocirculatoire, notamment pour la détection des troubles du rythme et la prévention de l'infarctus du myocarde.

2.1 Anatomie Cardiaque

Le cœur est un organe musclé constitué de quatre chambres. Les deux chambres supérieures sont appelées les oreillettes, tandis que les deux chambres inférieures sont appelées les ventricules. Le cœur est aussi séparé en deux parties : le cœur droit et le cœur gauche. Chacune de ces deux parties comprend donc une oreillette et un ventricule. L'oreillette droite collecte le sang qui a parcouru tout le corps (le sang "bleu") et l'envoie vers le ventricule droit afin qu'il soit éjecté dans les poumons pour y être réoxygéné. De la même façon, l'oreillette gauche collecte passivement le sang qui a traversé les poumons et l'achemine au ventricule gauche qui éjecte le sang.

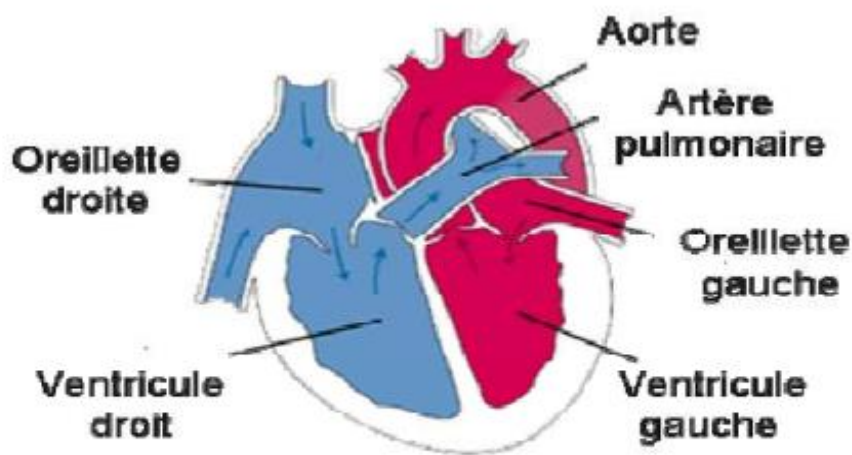


Figure II.1: Anatomie du cœur

2.2 Système de conduction électrique

Le cœur comprend un système de conduction électrique automatique qui assure chacun de ses battements. Le système de "pacemaker" interne est appelé le nœud sinusal (SA) qui se trouve dans la paroi de l'oreillette droite. Cette région produit une impulsion électrique qui gagne une région centrale du cœur appelée nœud auriculo-ventriculaire (AV) qui est stimulé en déclenchant une impulsion électrique qui descend le long du faisceau de His vers les branches droite et gauche et se termine dans les fines fibres de Purkinje comme le montre la figure II.2. Cette onde électrique produit la contraction du muscle cardiaque responsable de l'éjection du sang vers les poumons et l'aorte. Le nombre de contractions observées par minute correspond au rythme cardiaque ou "pouls". D'autres voies de conduction intracardiaque peuvent court-circuiter le parcours électrique normal. Il s'agit des faisceaux de Kent, des fibres de Mahaim et des fibres de James que l'on peut voir sur le schéma de la figure II.2. Leur existence est suspectée d'être due à la présence de certains symptômes pathologiques.

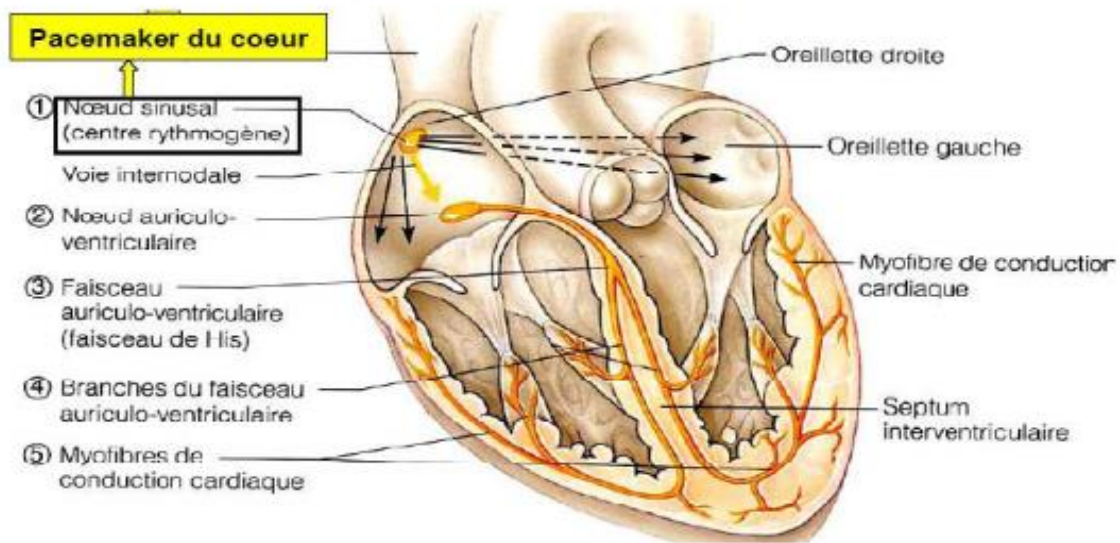


Figure II.2: Illustration schématique du système de conduction électrique du cœur.

L'électrocardiographie est l'art d'enregistrer l'activité électrique du cœur. Le corps humain étant électriquement conducteur, les potentiels d'actions générés lors de l'activité électrique cardiaque peuvent être recueillis par des électrodes placées sur la peau.

L'enregistrement de cette activité électrique du cœur, sur un plan frontal (par les dérivations des membres) et sur un plan horizontal (par les dérivations précordiales), est un électrocardiogramme (ECG).

3.Principe de l'électrocardiogramme

Le terme électrocardiogramme (ECG) désigne en général les techniques permettant de visualiser les différences de potentiel qui résultent de l'excitation du cœur. Ces potentiels naissent à la limite entre les zones excitées et celles non excitées du myocarde et sont mesurés entre deux points de la surface du corps. Une fibre cardiaque en cours de dépolarisation peut être assimilée à un dipôle électrique. A un instant donné le front de l'onde d'activation formé par l'ensemble des dipôles élémentaires crée un champ électrique qui est fonction des moments dipolaires.

L'enregistrement de l'évolution temporelle du champ électrique résultant, effectué au moyen d'électrodes cutanées, est nommé l'électrocardiogramme de surface et appelé simplement ECG.

3.1 Ondes et Intervalles de L'ECG

L'ECG est un outil de diagnostic qui permet de détecter les pathologies cardiaques rythmiques, musculaires, les problèmes extracardiaques métaboliques, médicamenteux, hémodynamiques et autres.

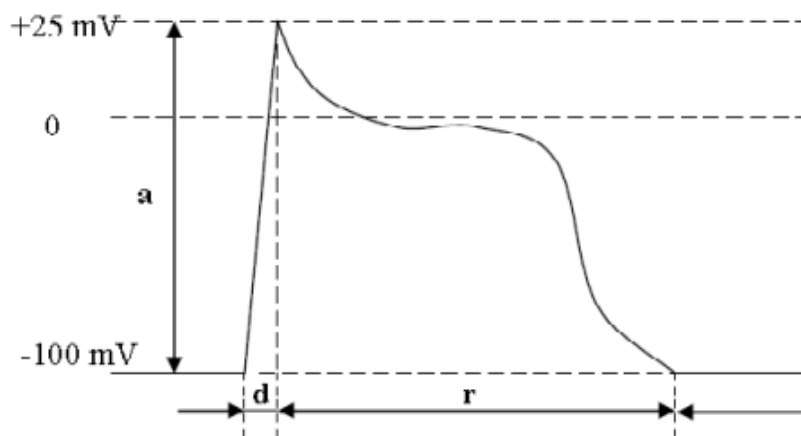


Figure II.3: Courbe du potentiel d'action : Dépolarisation d (rapide) et Repolarisation r (lente).

3.2 Electrophysiologie des cellules cardiaques

Une cellule au repos est polarisée. Le potentiel de la face interne de la cellule étant négatif par rapport à celui du milieu environnant. Cette différence de potentiel représente le potentiel de la membrane, il est entre 70 et 100 mV. Si on excite la surface de la cellule avec une charge négative telle que le potentiel de la membrane s'abaisse au moins d'un tiers, une décharge (dépolarisation) de celle-ci surviendra alors spontanément. La cellule cardiaque répond à ce phénomène par une contraction. La polarité à travers la membrane cellulaire est inversée. Le potentiel de la face interne est alors approximativement de 25mV plus élevé que celui de la surface externe de la cellule. Après la dépolarisation (phénomène initiale et rapide), la cellule se repolarise instinctivement (phénomène terminal et lent) et revient à l'état de repos. La figure II.3 montre la succession d'événements électriques par une courbe de variations de la différence de potentiel (ddp) entre les faces de la membrane cellulaire au cours d'un cycle. La variation de la ddp de -100 à 25 mV s'appelle potentiel d'action. Lorsqu'un phénomène électrique altère suffisamment le milieu au contact d'un tissu excitable, il se produit un état de déséquilibre qui se propage de proche en proche à toutes

les parties du tissu. Cette propagation s'accompagne d'une onde électrique enregistrable avec des appareils adéquats

3.3 Ondes Enregistrées

On enregistre, au moyen d'électrodes cutanées, l'évolution temporelle du vecteur champ électrique. Dans l'ECG normal, le rythme est d'origine sinusale fait d'une succession régulière de séquences P-QRS-T-U.

Pour chaque battement cardiaque, l'électrocardiogramme enregistre quatre ondes successives Figure II.4:

Le processus de dépolarisation et de repolarisation des structures myocardiques se présente dans l'ECG comme une séquence de déflexions ou ondes superposées à une ligne de potentiel zéro, appelée ligne isoélectrique ou ligne de base. Ces déflexions sont dites positives si elles sont situées au-dessus de la ligne isoélectrique sinon elles sont dites négatives.

L'onde P : Elle représente la dépolarisation auriculaire. Cette onde peut être positive ou négative avec une durée de l'ordre de 90 ms. Généralement son observation est difficile, spécialement dans des conditions bruitées. Il faut noter que la repolarisation auriculaire n'est pas visible sur l'ECG car elle coïncide avec le complexe QRS d'amplitude plus importante.

Intervalle PR (ou PQ) est le temps entre le début de P et le début du QRS.

Il est le témoin du temps nécessaire à la transmission de l'influx électrique du nœud sinusal des oreillettes au tissu myocardique des ventricules (conduction auriculo-ventriculaire). Sa durée normale, mesurée du début de l'onde P au début du complexe QRS est de 0,12 à 0,20 seconde. La durée de l'espace PR diminue lorsque la fréquence cardiaque augmente.

Il est normalement isoélectrique.

Le complexe QRS : Il correspond à la dépolarisation ventriculaire précédant l'effet mécanique de contraction et il possède la plus grande amplitude de l'ECG. Il est constitué de trois ondes consécutives : l'onde Q qui est négative, l'onde R qui est positive dans un ECG normal et l'onde S qui est négative. Sa durée normale est comprise entre 85 et 95 ms.

Le segment ST : Il représente l'intervalle durant lequel les ventricules restent dans un état de dépolarisation actif. Il est aussi défini comme la durée entre la fin de l'onde S et le début de l'onde T.

L'intervalle QT : Il représente la durée entre le début du complexe QRS et la fin de l'onde T. Cet intervalle reflète la durée de la dépolarisation et repolarisation ventriculaire.

En effet sa dynamique peut être associée à des risques d'arythmie ventriculaire et de mort cardiaque soudaine.

L'onde T : Elle correspond à la repolarisation des ventricules, qui peut être négative, positive ou bi phasique et qui a normalement une amplitude plus faible que le complexe QRS. Bien que la dépolarisation et la repolarisation des ventricules soient des événements opposés, l'onde T est normalement du même signe que l'onde R, ce qui indique que la dépolarisation et la repolarisation ne sont pas symétriques.

Onde U : est une petite déflexion parfois observée après l'onde T dans les dérivations précordiales V à V4. Elle est positive dans toutes les dérivations sauf en aVR, son origine est discutée.

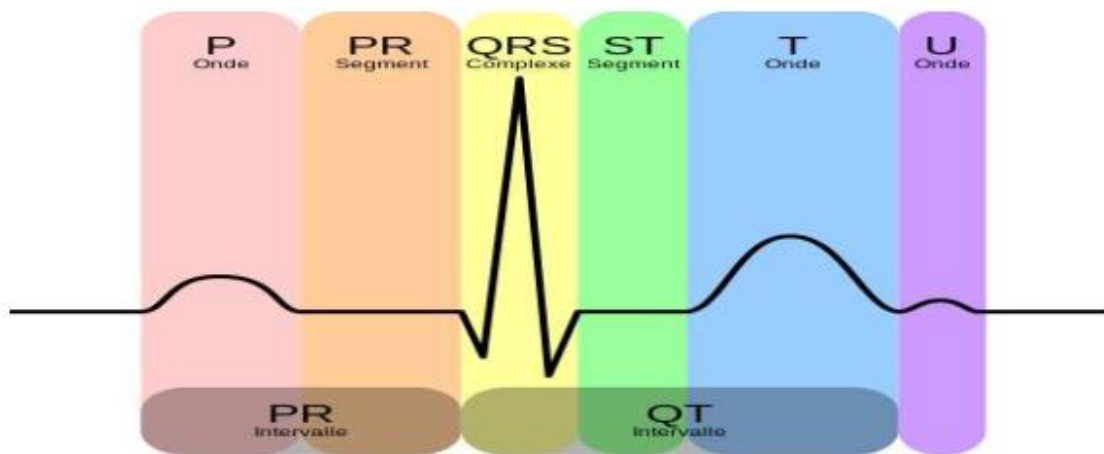


Figure II.4: ECG normal (intervalles temporels et différentes ondes)

3.4 L'utilisation du L'ECG

Les maladies cardiovasculaires représentent la cause la plus fréquente de décès dans le monde selon les études statistiques annuelles faites au niveau de l'organisation mondiale de la santé (OMS). Par conséquent, le diagnostic de ces maladies dangereuses semble une tâche vitale.

Dans les services de cardiologie au niveau des hôpitaux, le signal électrocardiogramme (ECG) reste encore l'un des outils prédominants et les plus largement utilisés pour le diagnostic et l'analyse des arythmies cardiaques.

Notamment, les médecins utilisent l'électrocardiogramme, qui est une représentation graphique temporelle des différences de potentiels des forces électriques qui conduisent à la contraction musculaire cardiaque.

L'électrocardiogramme contient énormément d'informations sur le fonctionnement et les éventuelles pathologies du cœur.

Chapitre 3 – le filtre RIF

1. Introduction

Le filtrage du signal numérique intervient dans de nombreux domaines : transmission du signal, traitement du son, des images, traitement des données scientifiques, etc.

On se limite au cas des signaux unidimensionnels. On considère principalement des filtres à réponse impulsionnelle finie (filtres RIF). Une étude plus détaillée des filtres RIF est présentée dans Filtres à réponse impulsionnelle finie.

Un signal est la représentation physique de l'information qu'il transporte de sa source à son destinataire. Il sert de vecteur à une information. Il constitue la manifestation physique d'une grandeur mesurable (courant, tension, force, température, pression, etc.). Les signaux sont des grandeurs électriques variant en fonction du temps $x(t)$ obtenues à l'aide de capteurs. Sur le plan analytique : Un signal sera une fonction d'une variable réelle, en général le temps.

Exemples :

- ♣ Onde acoustique : délivré par un microphone (parole, musique, ...)
- ♣ Signaux biologiques : EEG, ECG
- ♣ Tension aux bornes composant électronique
- ♣ Signaux géophysiques : vibrations sismiques
- ♣ Finances : cours du pétrole
- ♣ Images, Vidéos

Remarque : Tout signal physique comporte une *composante* aléatoire (perturbation externe, bruit, erreur de mesure, etc. ...). **Le bruit** est défini comme tout phénomène perturbateur gênant la perception ou l'interprétation d'un signal, par analogie avec les nuisances acoustiques (interférence, bruit de fond, etc.). La différenciation entre le signal et le bruit est artificielle et dépend de l'intérêt de l'utilisateur : les ondes électromagnétiques d'origine galactique sont du bruit pour un ingénieur des télécommunications par satellites et un signal utile pour les radioastronomes.

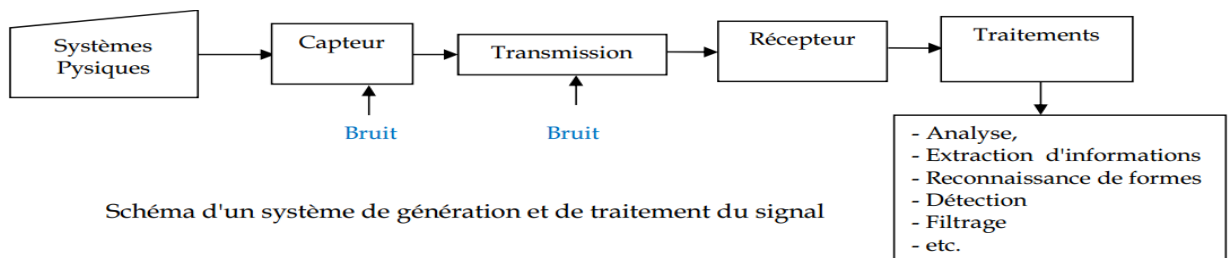


FIGURE III : schéma d'un système de génération et traitement du signal

2. Signal échantillonné

Lorsqu'un signal analogique $x(t)$ est échantillonné avec une période d'échantillonnage T_e , on obtient la suite suivante :

$$x_n = x(nT_e) \quad (1)$$

Où n est un entier. La suite x_n constitue un signal numérique.

Lorsque le spectre de $x(t)$ a une fréquence maximale f_{\max} , la condition de Nyquist-Shannon permet de s'assurer que toute l'information du signal analogique est présente dans le signal échantillonné : la fréquence d'échantillonnage $f_e=1/T_e$ doit être supérieure à $2f_{\max}$.

3. Filtrage à réponse impulsionnelle finie

Introduction

Les filtres numériques peuvent être classés selon plusieurs critères :

la longueur de la réponse impulsionnelle implique deux types de filtres :

RIF, i.e. **Filtres à réponse impulsionnelle finie**

RII, i.e. **Filtres à réponse impulsionnelle infinie**

le type de représentation, ou de structure, implique deux types de filtres récursifs ($a_i \neq 0$) et non récursifs.

A l'exception de cas particuliers, les filtres récursifs et non récursifs sont respectivement équivalents aux filtres RII et RIF.

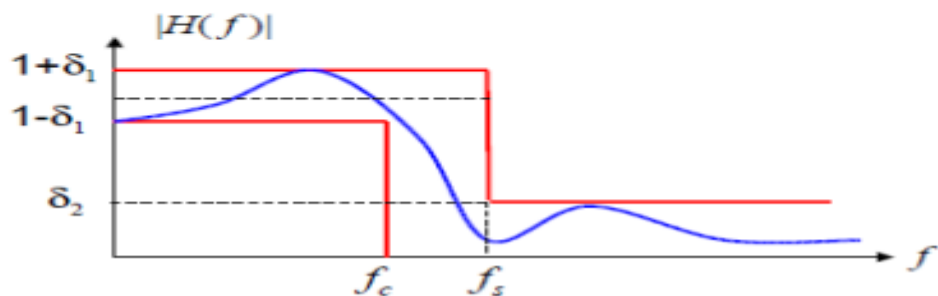
3.a. Définition

Un filtre numérique à réponse impulsionnelle finie réalise le calcul d'un signal de sortie y_n en fonction du signal d'entrée x_n de la manière suivante :

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad |h(n)| < \infty,$$

où, $h(n)$ est la réponse impulsionnelle du filtre et N est sa longueur.

Le gabarit réel d'un filtre FIR est le suivant,

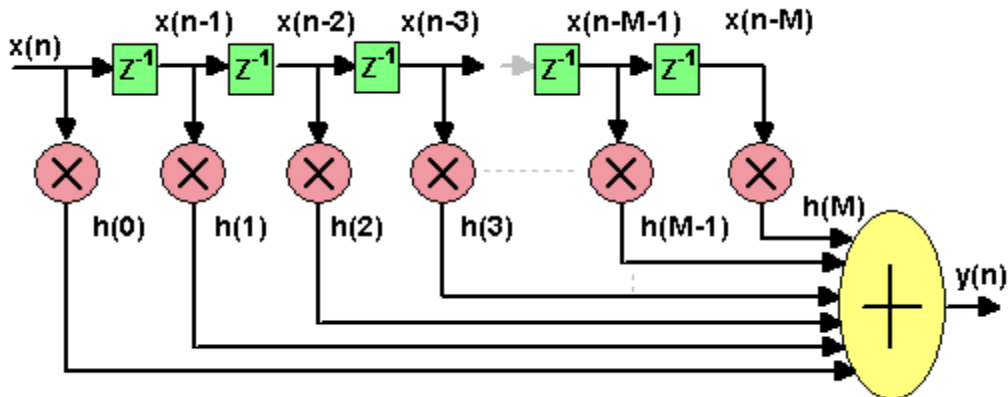


3.b. Réalisation

Les filtres numériques peuvent être réalisés à l'aide de trois éléments ou opérations de base. Soit l'élément gain, l'élément de sommation et le retard unitaire. Ces éléments sont suffisants pour réaliser tous les filtres

numériques linéaires possibles. La réalisation présentée dans la figure ci-contre est une réalisation directe de type 1 du filtre RIF.

Un filtre RIF peut également être représenté par un graphe :



3.c. Propriétés

Les remarques générales suivantes peuvent être portées sur les filtres RIF.

- Les filtres RIF sont forcément stables, peu importe les coefficients utilisés.
- La complexité d'un filtre RIF est moindre que celle d'un filtre RII du même ordre. Cette propriété peut être utile sur les plateformes limitées en puissance de calcul.
- Généralement, les filtres RIF sont moins sensibles aux erreurs de quantification que les filtres RII. L'absence de récursivité empêche les erreurs cumulatives.
- Un filtre RIF est moins sélectif qu'un filtre RII du même ordre. C'est-à-dire que la transition entre la bande passante et la bande rejetée est moins rapide que dans le cas du filtre RII.
- Contrairement à un RII, un filtre RIF peut avoir une réponse impulsionnelle symétrique et introduire un retard sur le signal mais aucun déphasage.

Le filtre est caractérisé par,

- La bande passante BP
- La bande atténuée (ou coupée).
- La largeur $\Delta F = f_s - f_c$ de la zone de transition.
- L'amplitude des oscillations en bande passante δ_1 .
- L'amplitude des ondulations en bande atténuée δ_2 .

Les filtres numériques présentent par rapport aux filtres analogiques les avantages-inconvénients suivants :

Avantages :

- Les filtres numériques sont insensibles aux conditions extérieures (chaleur, humidité, etc.)
- Certains filtres numériques sont impossibles à réaliser de manière analogique (exemple: les filtres RIF).
- Les filtres numériques ne sont pas sensibles aux non idéalités d'un 10ème amplificateur opérationnel. Ainsi, un filtre RII du 10 ordre est tout à fait envisageable (attention quand même au bruit de calcul !).
- La problématique du bruit change d'aspect: dans le filtrage numérique on parle de 'bruit de quantification' et de 'bruit de calcul'. Le premier est lié au nombre de bits employés pour la quantification (8 bits, 16 bits, etc.). Le second est négligeable si l'unité de calcul est de type 'floating point'. De toute manière, le bruit numérique est localisé: on sait d'où il vient et il reste stable.

Inconvénients :

- Les filtres numériques nécessitent un filtrage analogique anti-repliement à l'échantillonnage et à la restitution.
- Les performances d'un filtre sont directement proportionnelles à la puissance de l'unité de calcul (processeur ou DSP).
- Beaucoup de problèmes peuvent apparaître si l'unité de calcul est de type 'fixed point'.

Les paramètres d'un filtre nécessitent parfois une double précision pour être opérationnels, ce qui ralentit les performances.

On trouve essentiellement deux grandes familles de filtres numériques :

A. Filtres RIF : Ces filtres non récurrents n'ont pas de contre-réaction.

Avantages :

- Toujours stables.
- Phase linéaire si symétrie des coefficients
- Possibilité de réaliser toutes sortes de filtres en dessinant simplement des gabarits de réponse en amplitude

Inconvénients :

- Beaucoup de calculs par rapport à un RII
- Le retard du filtre (de groupe ou de phase) peut être important si le nombre

de 'taps' est élevé.

B. Filtres RII : Ces filtres récursifs ont une contre

Avantage :

- Beaucoup moins de calculs par rapport à un RIF équivalent au niveau des performances.

Inconvénients :

- Il faut vérifier la stabilité.

- Phase non linéaire

Chapitre 4 – Implémentation d'un filtre FIR passe bas dans la carte tms320c7613 :

1. Le Code Composer Studio

L'évaluation des performances des algorithmes sur le DSP est effectuée en utilisant Code Composer Studio (CCS), son fonctionnement est illustré par la figure suivante,

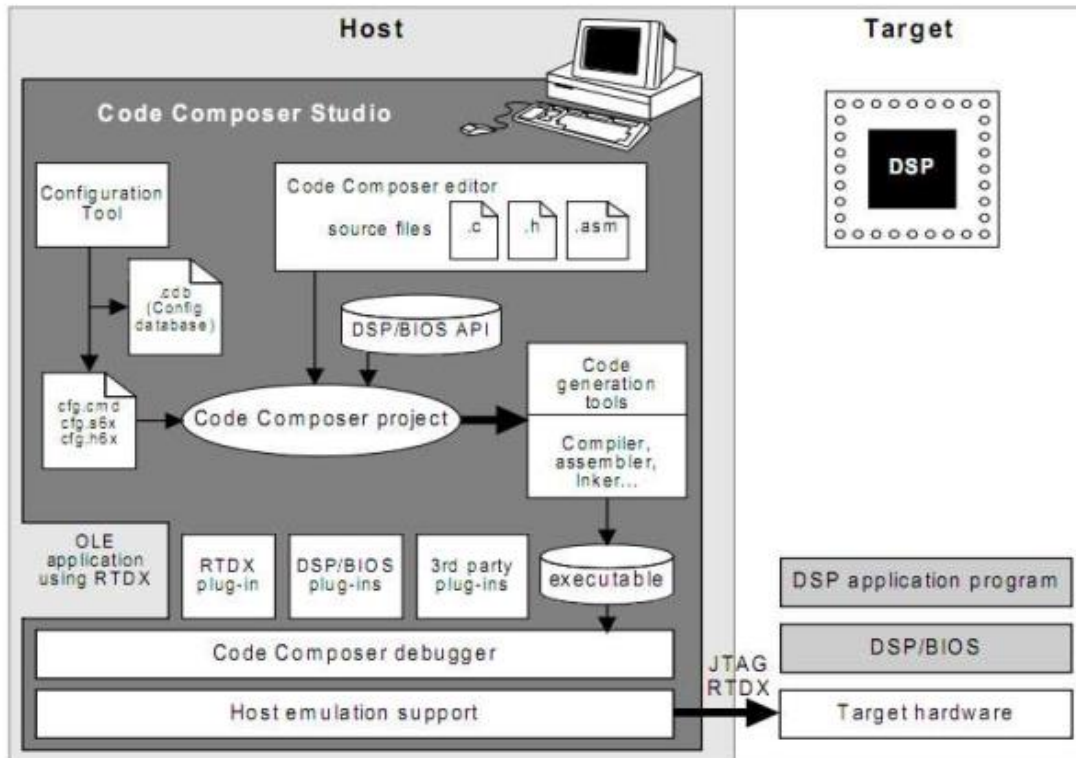


Figure 4.1- Structure du système de développement du TMS320C6713.

Le logiciel Code Composer Studio est une plate-forme de développement qui inclut les éléments suivants,

- Environnement de développement intégré (IDE) : il permet l'édition ('built'), et la correction ('debug') des programmes destinés au DSP.
- Outils de génération du code pour le TMS320C6000 : ces outils sont le compilateur, l'assembleur et l'éditeur de lien. Le compilateur C/C++ permet de compiler le programme source (xxx.c) pour le convertir en assembleur (xxx.asm), l'assembleur reçoit le fichier xxx.asm et le convertit en langage machine ou fichier objet (xxx.obj), enfin l'éditeur de liens (linker) qui combine les fichiers objet et les fichiers bibliothèques et le fichier xxx.cmd pour produire un fichier exécutable avec une extension .out, c'est ce fichier qui sera chargé sur le processeur C6713 pour être exécuter.
- DSP/BIOS: c'est un outil d'analyse en temps réel, pour s'en servir, on doit créer un fichier de configuration 'xxx.cdb', où seront définis les objets utilisés par l'outil DSP/BIOS. Ce fichier permet aussi de faciliter l'organisation de la mémoire et la gestion du vecteur des interruptions, en offrant la possibilité de les faire sur un environnement visuel via la section de gestion de la mémoire MEM (Memory Section Manager), et via HWI (Hardware Interrupt Service Routine Manager) pour les interruptions.
- JTAG (Joint Team Action Group) et RTDX (Real Time Data Exchange) (voir Figure

suivante): le RTDX permet un échange de données en temps réel entre l'hôte (PC par exemple) et la destination (la carte DSK dans notre cas), il permet aussi l'analyse et la visualisation des données au cours de l'exécution du programme, alors que le lien JTAG est utilisé pour atteindre l'émulateur (qui se trouve à l'intérieure du DSP), c'est ce dernier qui permet au CCS de contrôler en temps réel l'exécution du programme .

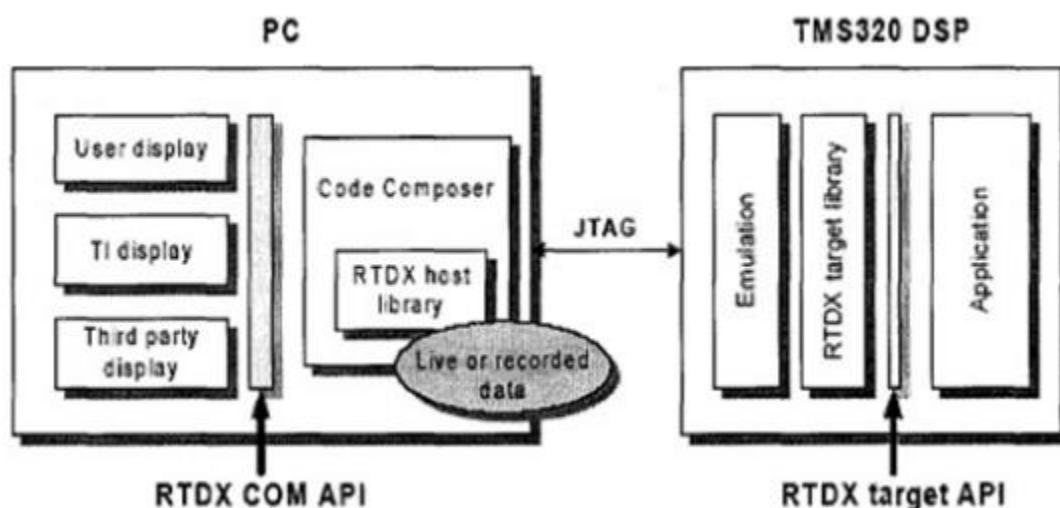


Figure 4.2- L'outil JTAG

- Simulateur intégré : le logiciel Code Composer Studio offre la possibilité de tester des programmes pour DSP sans utiliser la carte DSK à l'aide d'un simulateur intégré.

Remarque : Grâce à une collaboration entre Texas Instruments Inc. et MathWorks Corp, on pourrait de plus configurer la carte à l'aide le Matlab/Simulink. Cette collaboration a réduit immensément la complexité de programmation de la carte. Dans le cadre de ce projet, nous avons préféré d'utiliser le langage C grâce à sa portabilité par les différents fabricants des cartes DSP (ce n'est pas le cas pour la programmation avec Matlab/Simulink).

2.Lancement du CCS

Avant de lancer le CCS, il est impératif de suivre dans l'ordre les étapes suivantes,

1. S'assurer que le Code Composer Studio (CCS) est fermé.
2. Mettre sous tension la carte en connectant le jack d'alimentation. Si elle était déjà sous tension, débrancher puis rebrancher le jack d'alimentation.
3. Vérifier que la carte s'initialise correctement : l'ordinateur doit signaler la connexion USB et les 4 diodes groupées présentes sur la carte doivent clignoter après une quinzaine de secondes et rester allumées.
4. Utilisez l'utilitaire de diagnostic pour tester la fonctionnalité de la carte, pour le faire, on procède comme suit :
 - Appuyer sur l'icône « 6713 DSK Diagnostics Utility v3.1 » qui se trouve sur le bureau.
 - Appuyer sur START pour lancer le diagnostic.
 - Vérifier qu'à la fin du diagnostic que tous les indicateurs soient en vert comme le montre la figure suivante,

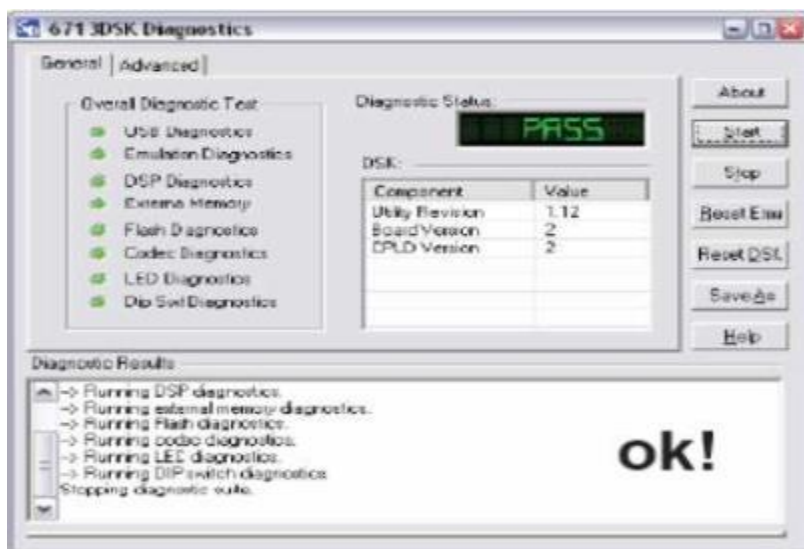


Figure 4.3– la fenêtre de la fin du diagnostic de la carte

5. Lancer le Code Composer Studio en cliquant sur « 6713 DSK CC Studio 3.1 » qui se trouve par défaut sur le bureau.

6. Établir la connexion avec la carte en cliquant sur « Debug » dans le menu principal et en choisissant « Connect ». On peut vérifier le statut de la connexion qui s’affiche dans le coin gauche de la fenêtre.

3 .Programmation en C

La réalisation d'une application avec CCS se fait par la création d'un projet (un fichier avec extension .pj) suivi de la configuration des différentes options nécessaires à son exécution. Dans un projet, on retrouve plusieurs fichiers regroupés selon leurs types dans des répertoires différents, ces répertoires sont,

- Include: contient les fichiers de l'en-tête *.h.
- Libraries : contient les fichiers librairies *.lib.
- Source : contient les fichiers sources du projet, ces fichiers peuvent être des programmes en langage C, C++, ou/et des programmes en assembleur *.asm.

Dans ce travail, nous nous sommes contentés de l’utilisation du langage C.

3.1 Création du fichier exécutable

Pour créer le fichier exécutable qui sera chargé sur le DSP, il faut passer par plusieurs étapes :

1. Dans le menu principal, on clique sur « Project », et on choisit « New ». Dans cette fenêtre, on donne un nom pour notre projet, sa location sur le disque dur de l’ordinateur, son type (exécutable .out ou librairie .lib) et la carte DSP (Target) utilisée (C6713 pour notre cas).
2. Il faut écrire le programme principal en C ou en Assembleur en cliquant dans le menu principal sur « File » puis « new ».
3. Il faut joindre tous les fichiers indispensables (ne pas oublier le programme principal) pour le bon fonctionnement du programme, pour le faire, on doit cliquer dans le menu principal sur « Project », puis « add files to project », et finalement sur « open ».
4. Joindre aussi le fichier de commande (.cmd ou .lcf). Ce type de fichier divise la

mémoire de la carte en des sections.

5. Avant de générer le code, il faut configurer les options de compilation et de lien (Project build options) tel qu'elle est illustrée par la figure suivante,

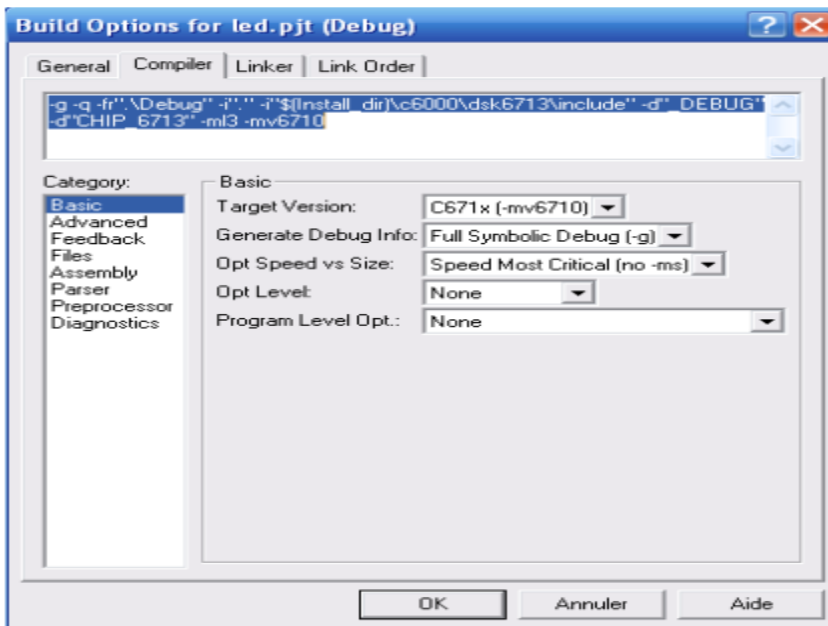


Figure 4.4- Fenêtre de configuration des options de compilation et de lien.

6. Après l'étape de configuration, on construit le projet (Project Rebuild All) qui génère le fichier exécutable avec l'extension .out. Ce dernier est ensuite chargé sur le DSP (File Load) et finalement, il ne reste qu'à exécuter le programme (Debug Run).

4 .Création d'un filtre FIR passe-bas

1ere il faut calculer les coefficients

Pour calculer les coefficients de la réponse impulsionnelle, on a recours à deux méthodes : la première est celle du calcul manuel, qui demande beaucoup de temps, et que nous ne traitons pas dans ce travail, tandis que la deuxième méthode est celle du calcul à l'aide de l'outil 'fdatool' de Matlab.

Pour y arriver, il nous faut qu'à taper dans la fenêtre de commande Matlab 'fdatool', cette icommande permet d'ouvrir la fenêtre suivante,

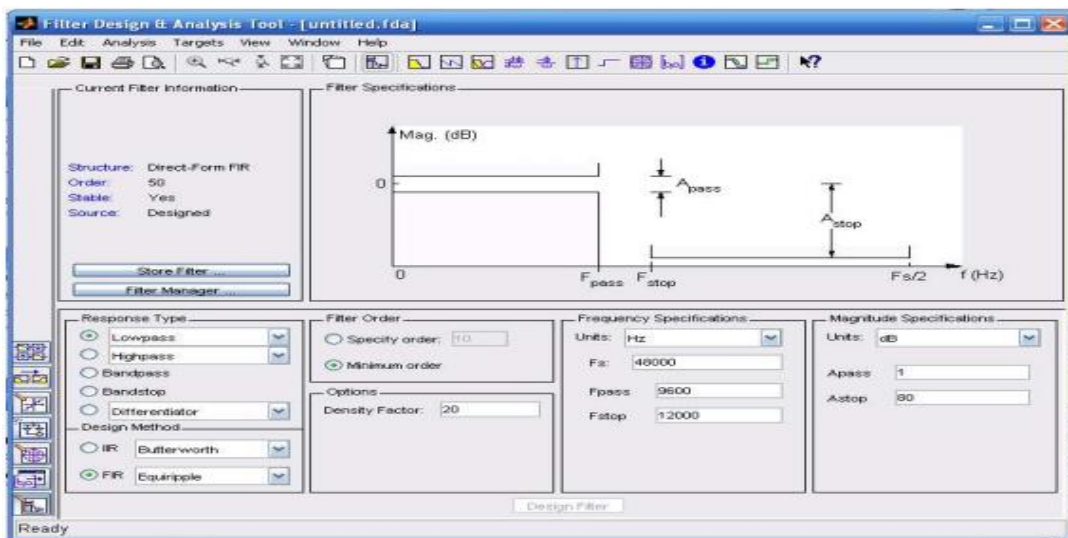


Figure 4.5 - la fenêtre 'fdatool' de MATLAB.

A ce niveau, nous précisons le type, les caractéristiques (citées ci-dessus), la méthode de conception du filtre, Sélectionner ensuite « Design Filter », puis (Target Generate C Header...) pour générer les coefficients de la réponse impulsionnelle dans un fichier en-tête.

5 .Implémentation du filtre sur le DSP

Pour implémenter ce filtre FIR sur le DSP qui fait objet de notre étude, il faut réserver une partie de la mémoire pour stocker les N coefficients de la réponse impulsionnelle, une autre partie sera consacrée pour les échantillons de l'entrée. Cette dernière devra mettre à jour à chaque réception d'une nouvelle donnée $x(n)$. En effet, chaque nouvel échantillon doit écraser l'avant dernière en gardant la même taille du tableau comme le montre le schéma suivant :

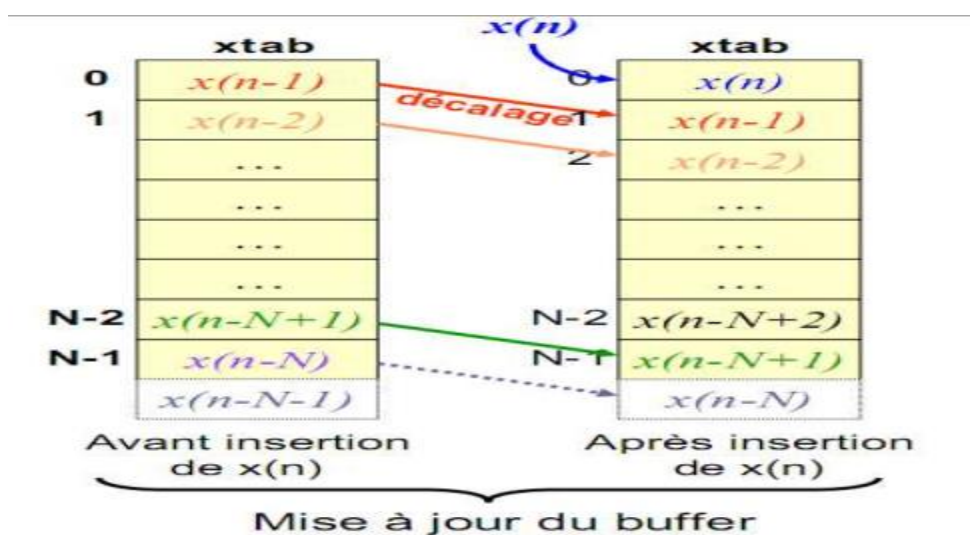


Figure 4.6- mémoire de donnée pour un filtre FIR.

Pour tester ce filtre et s'assurer de son bon fonctionnement, nous avons passé par les étapes suivantes :

- a) La création d'un projet nommé « filtre_fir.pjt » en CCS : pour y arriver, il faut joindre au programme principal en langage C « filtre_fir.c » (voir annexe C), les fichiers suivants (dont l'utilité est déjà mentionnée dans le chapitre précédent), c6713dskinit.c, Vectors_intr.asm, c6713dsk.cmd, rts6700.lib, dsk6713bsl.lib et

csl6713.lib.

Construire le projet (Project Rebuild All) qui génère le fichier exécutable 'filtre_fir.out' et finalement charger ce fichier exécutable sur le DSP (File Load).

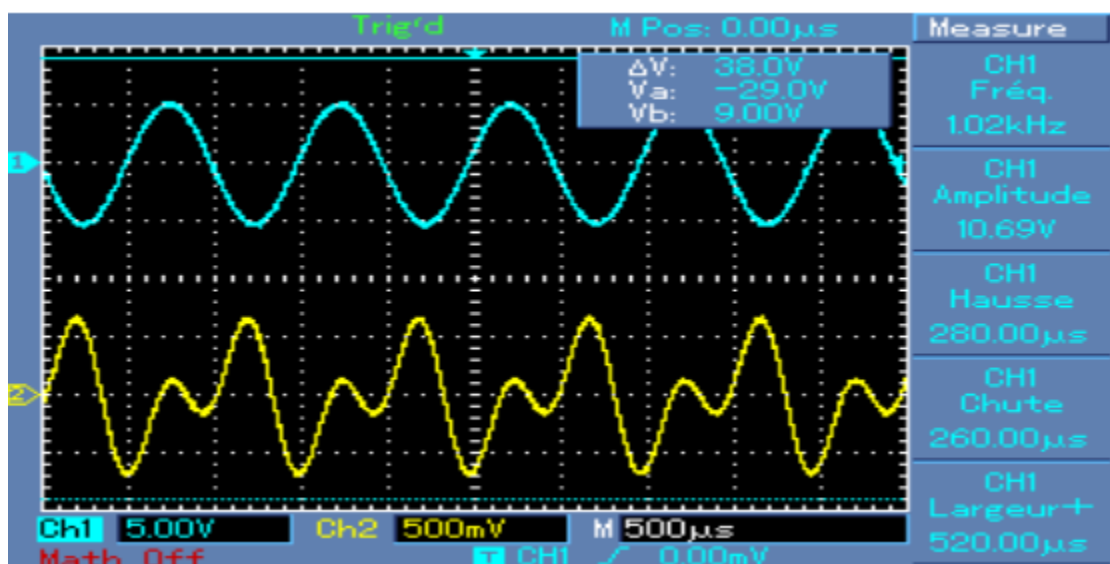
b) La création d'un script MATLAB pour la génération d'une somme de deux signaux sinusoïdaux, la première sinusoïde a une fréquence de 1 kHz et la deuxième de 2 kHz (le signal qui sera filtré) (voir script dans l'annexe).

c) Relier l'entrée 'LINE IN' du DSK avec le PC à l'aide d'un câble JACK.

d) Relier la sortie de la carte à un oscilloscope pour visualiser le signal de sortie.

e) Lancer le programme.

Le résultat obtenu sur l'oscilloscope est présenté au niveau de la figure 4.6



La sortie de la carte (signal en bleu) est un signal de fréquence de 1.02 kHz, d'où la validité du résultat escompté. En effet, le filtre a laissé passer le premier signal (de fréquence de 1KHz) et il a rejeté le deuxième (de 2 KHz).

Afin de s'assurer de la validité de la réponse du filtre implémenté sur le DSP, nous avons fait une comparaison entre sa réponse et celle obtenue théoriquement avec l'outil 'fdatoool' de MATLAB. En effet, nous avons attaqué la carte par un signal sinusoïdal à partir d'un GBF dont la fréquence varie autour de la fréquence de coupure, les résultats obtenus sont présentés dans le tableau et la figure suivants,

f (Hz)	400	700	1020	1120	1430	1480	1500	1520	1540	1560
V_e (volt)	2	2	2	2	2	2	2	2	2	2
V_s (Volt)	2.0	2.0	2.0	1.97	1.97	1.85	1.67	1.52	1.40	1.11
V_s / V_e	1.0	1.0	1.0	0.985	0.985	0.925	0.835	0.760	0.700	0.555

f (Hz)	1580	1590	1610	1620	1660	1680	1700	1720	1740	1760
V_e (volt)	2	2	2	2	2	2	2	2	2	2
V_s (Volt)	0.88	0.77	0.60	0.43	0.11	0.07	0.02	0.005	0.001	0.003
V_s / V_e	0.440	0.385	0.300	0.215	0.055	0.035	0.010	0.002	0.000	0.000

La réponse du filtre qui nous a généré les coefficients de la réponse impulsionnelle est la suivante,

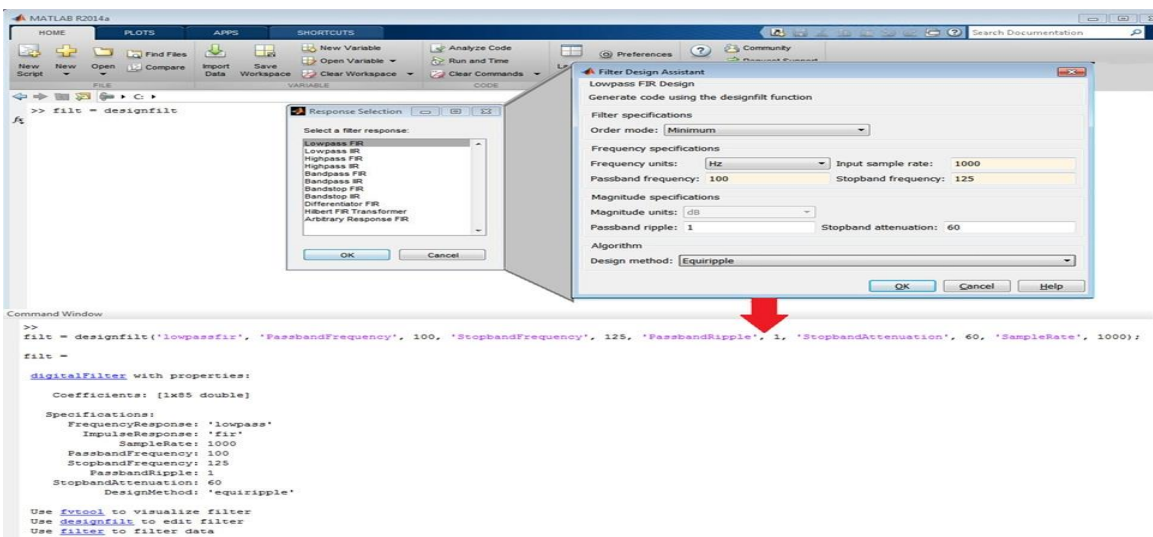


Figure 4.7-1 – la réponse du filtré désigné par ‘fdatool’ dans MATLAB

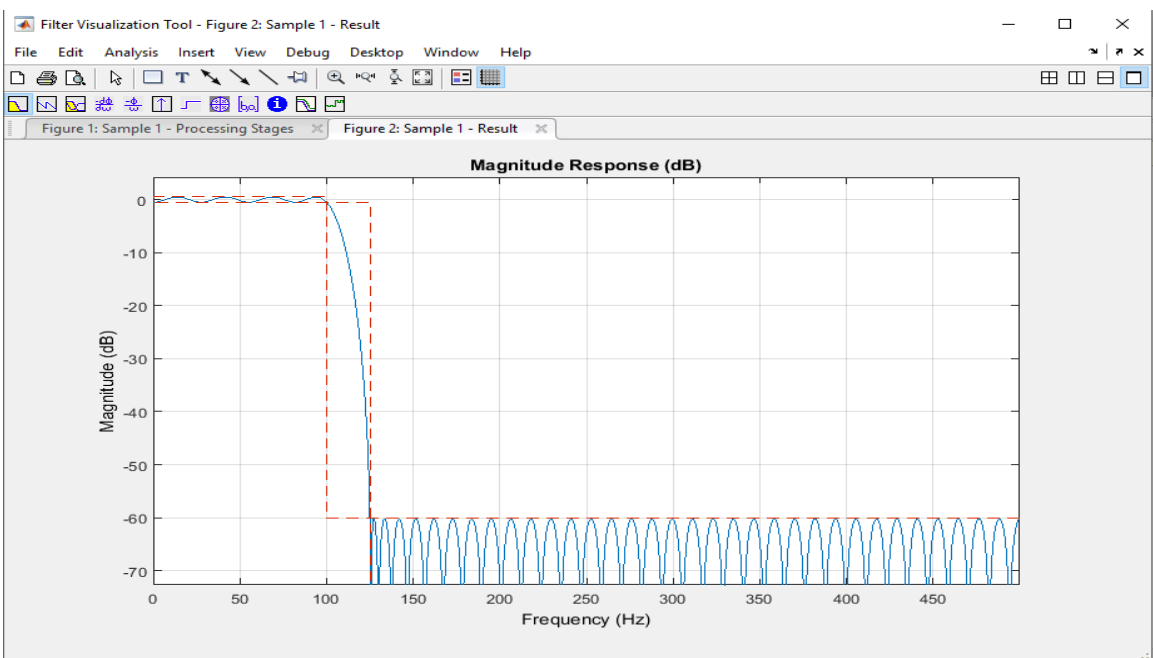


Figure 4.7-2 – la réponse du filtré désigné par ‘fdatool’ dans MATLAB

designfilt('lowpassfir', 'PassbandFrequency', 100,'StopbandFrequency', 125, 'PassbandRipple', 1, 'StopbandAttenuation', 60, 'SampleRate', 1000);

Exemple filtrage d'un signal ECG

```
for demo = 1:2:3
    % Clear our variables
    clear ecg samplingrate corrected filtered1 peaks1 filtered2 peaks2 fresult
    % Load data sample
    switch(demo)
        case 1,
            plotname = 'Sample 1';
            load ecgdemodata1;
        case 3,
            plotname = 'Sample 2';
            load ecgdemodata2;
    end
    % Remove lower frequencies
    fresult=fft(ecg);
    fresult(1 : round(length(fresult)*5/samplingrate))=0;
    fresult(end - round(length(fresult)*5/samplingrate) : end)=0;
    corrected=real(ifft(fresult));
    % Filter - first pass
    WinSize = floor(samplingrate * 571 / 1000);
    if rem(WinSize,2)==0
        WinSize = WinSize+1;
    end
    filtered1=ecgdemowinmax(corrected, WinSize);
    % Scale ecg
    peaks1=filtered1/(max(filtered1)/7);
    % Filter by threshold filter
    for data = 1:length(peaks1)
        if peaks1(data) < 4
            peaks1(data) = 0;
        else
            peaks1(data)=1;
        end
    end
    positions=find(peaks1);
    distance=positions(2)-positions(1);
    % Returns minimum distance between two peaks
    for data=1:length(positions)-1
        if positions(data+1)-positions(data)<distance
            distance=positions(data+1)-positions(data);
        end
    end
    % Optimize filter window size
    QRdistance=floor(0.04*samplingrate);
    if rem(QRdistance,2)==0
        QRdistance=QRdistance+1;
    end
    WinSize=2*distance-QRdistance;
    % Filter - second pass
    filtered2=ecgdemowinmax(corrected, WinSize);
    peaks2=filtered2;
    for data=1:length(peaks2)
        if peaks2(data)<4
            peaks2(data)=0;
        else
            peaks2(data)=1;
        end
    end
    %% This part of the code between the double comments is added by Surya Penmetsa
    positions2=find(peaks2);
```



```

distanceBetweenFirstAndLastPeaks = positions2(length(positions2))-positions2(1);
averageDistanceBetweenPeaks =
distanceBetweenFirstAndLastPeaks/length(positions2);
averageHeartRate = 60 * samplingrate/averageDistanceBetweenPeaks;
disp('Average Heart Rate = ');
disp(averageHeartRate);
% The code written by Surya Penmetsa Ends here.
%% % Create figure - stages of processing
figure(demo); set(demo, 'Name', strcat(plotname, ' - Processing Stages'));
% Original input ECG data
subplot(3, 2, 1); plot((ecg-min(ecg))/(max(ecg)-min(ecg)));
title('\bf1. Original ECG'); ylim([-0.2 1.2]);
% ECG with removed low-frequency component
subplot(3, 2, 2); plot((corrected-min(corrected))/(max(corrected)-min(corrected)));
title('\bf2. FFT Filtered ECG'); ylim([-0.2 1.2]);
% Filtered ECG (1-st pass) - filter has default window size
subplot(3, 2, 3); stem((filtered1-min(filtered1))/(max(filtered1)-min(filtered1)));
title('\bf3. Filtered ECG - 1st Pass'); ylim([0 1.4]);
% Detected peaks in filtered ECG
subplot(3, 2, 4); stem(peaks1);
title('\bf4. Detected Peaks'); ylim([0 1.4]);
% Filtered ECG (2-d pass) - now filter has optimized window size
subplot(3, 2, 5); stem((filtered2-min(filtered2))/(max(filtered2)-min(filtered2)));
title('\bf5. Filtered ECG - 2d Pass'); ylim([0 1.4]);
% Detected peaks - final result
subplot(3, 2, 6); stem(peaks2);
title('\bf6. Detected Peaks - Finally'); ylim([0 1.4]);
% Create figure - result
figure(demo+1); set(demo+1, 'Name', strcat(plotname, ' - Result'));
% Plotting ECG in green
plot((ecg-min(ecg))/(max(ecg)-min(ecg)), '-g'); title('\bf Comparative ECG R-Peak
Detection Plot');
% Show peaks in the same picture
hold on
% Stemming peaks in dashed black
stem(peaks2.'*((ecg-min(ecg))/(max(ecg)-min(ecg)))', ':k');
% Hold off the figure
hold off
end

```

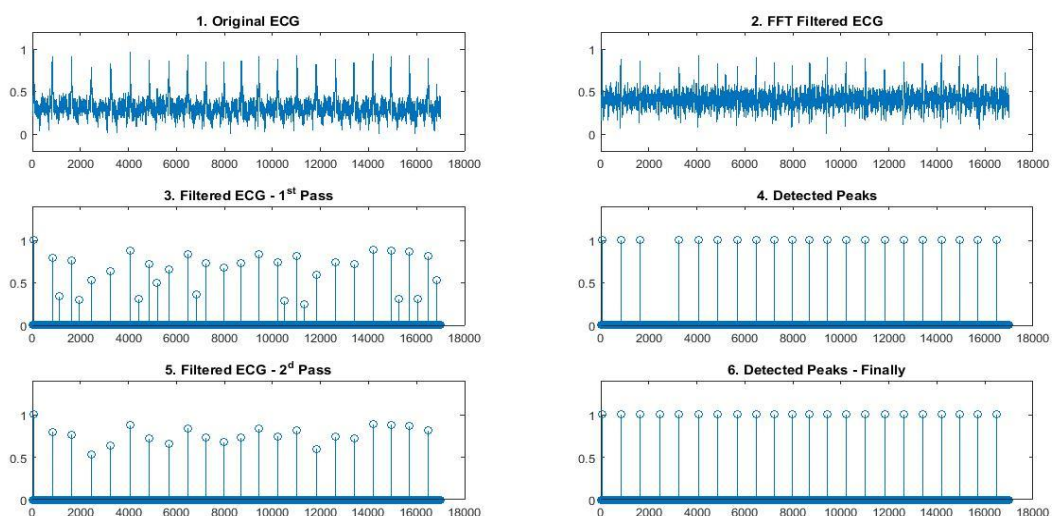


Figure 4.8 – un signal ECG original dans MATLAB

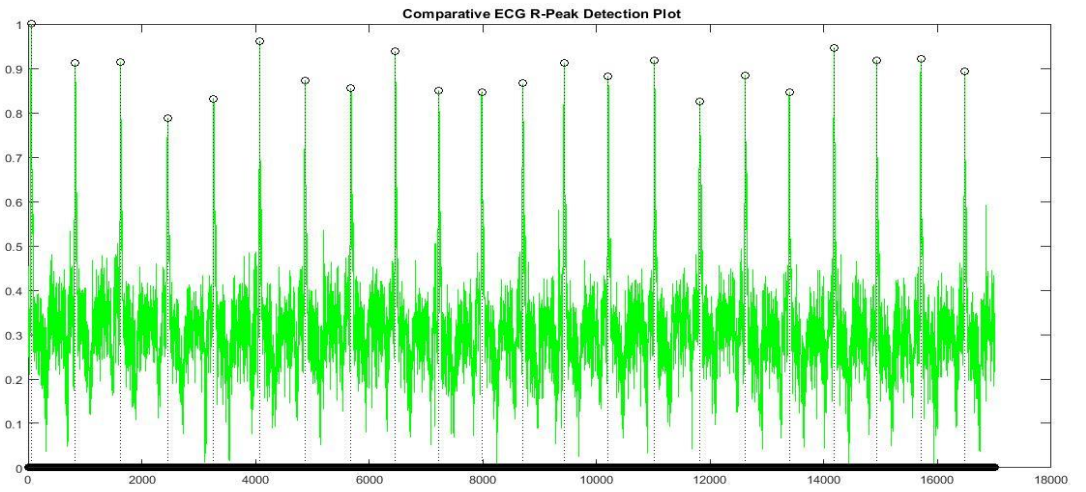


Figure 4.9 – un signal ECG original dans MATLAB

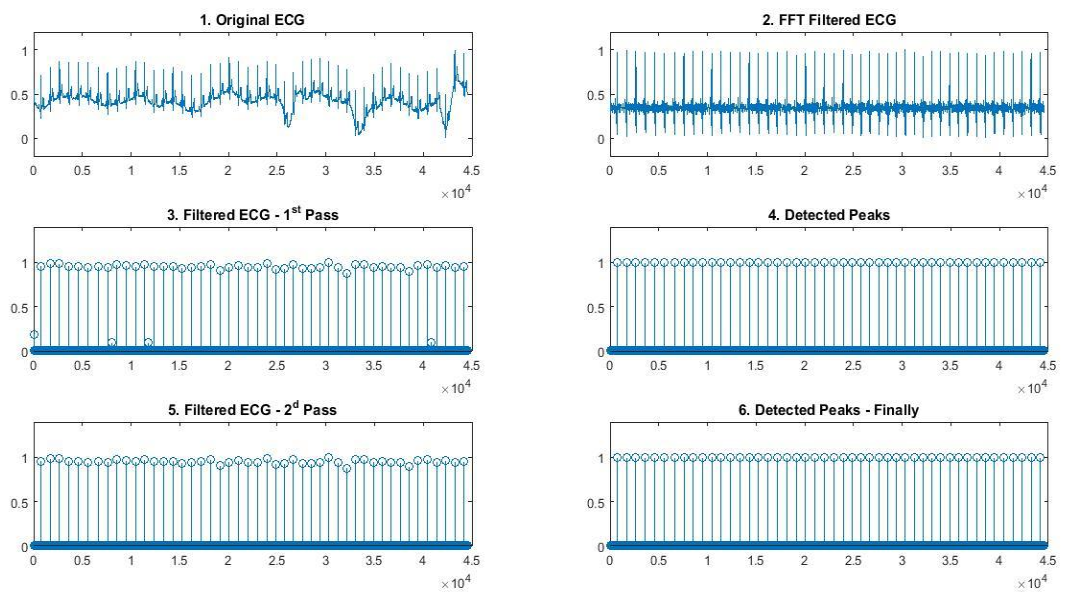


Figure 4.10 – un signal ECG laissé seulement QRS dans MATLAB

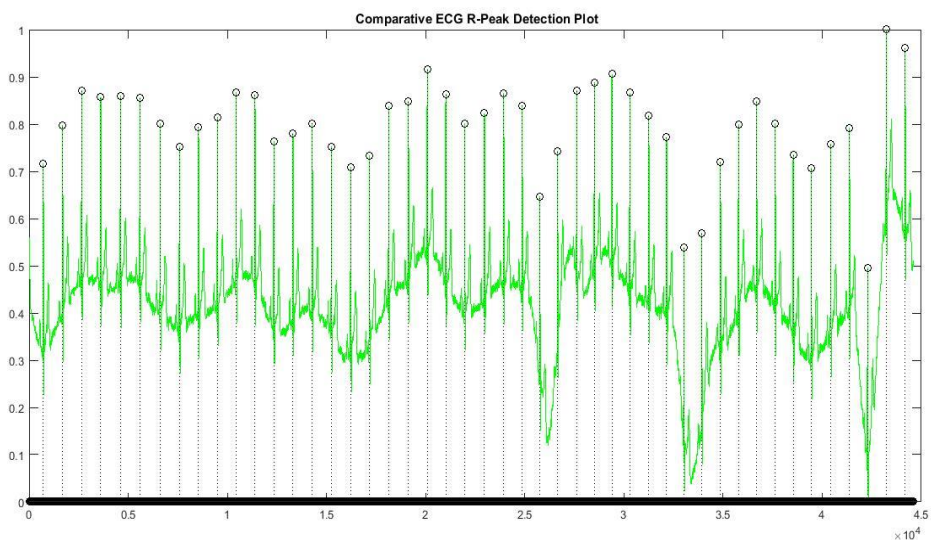


Figure 4.11 – un signal ECG filtré dans MATLAB

Conclusion

Durant notre projet, nous avons pu constater que les DSP ne représentent qu'une partie du traitement numérique du signal, mais il reste essentiel par son rôle important dans la chaîne de traitement.

Le travail réalisé au cours de cette période nous a permis d'élargir nos compétences dans le domaine du traitement numérique du signal, ainsi que la découverte du fonctionnement et de l'architecture des DSP constituant un élément clé dans le TNS.

Comme perspectives, nous recommandons de faire la programmation sur DSP en utilisant le langage assembleur, vu ses innombrables avantages par rapport à la programmation en langage C. nous recommandons également l'implémentation des algorithmes plus complexes sur le DSP, par exemple, Les algorithmes de codage canal, de compression et des modulateurs numériques : ASK, PSK, FSK, OFDM,...

Pour conclure, maîtriser ou avoir des bases sur l'utilisation du DSP et le traitement numérique du signal devient indispensable vu le nombre de domaines concernés de nos jours.

Bibliographie

- [1] M. Correvon(2010). Introduction aux DSP orientés & applications industrielle - *Notes du cours* [Présentation PDF]. Repéré dans le site www.iai.heig-vd.ch. [2] dsk6713_TechRef de TEXAS INSTRUMENT (2006), Datasheet du C6713 DSK.
- [3] Rulph Chassaing(2005). Digital Signal Processing and Applications with the C6713 and C6416. USA, New Jersey: John Wiley & Sons, Inc.
- [4] Sylvain MONTAGNY(2008). Microprocesseurs microcontrôleurs - *Notes du cours* [Présentation PDF]. Repéré dans le site www.sfa.univ-savoie.fr.
- [5] TMS320 datasheet pdf datenblatt - Texas Instruments

WEBOGRAPHIE

- [6] www.aix-mrs.iufm.fr/formations/filieres/ge/data/DSP
- [7] fr.wikipedia.org/wiki/Processeur_de_signal_num%C3%A9rique

<http://b.l.free.fr/Page9.html>

<http://www.medteq.info/med/ECGFilters>

Annexe A : Programme principal en langage C du filtre FIR

passé-bas de fréquence de coupure 1500Hz

```
#include "dsk6713_aic23.h" // fichier support pour le codec et le DSK

#define Nbr_coeff 81

Uint32 fs=DSK6713_AIC23_FREQ_8KHZ; // fréquence d'échantillonnage

int sortie = 0; //initialiser la sortie du filtre

short h[Nbr_coeff]=

{-6, 13, 42, 52, 14, -40, -45, 18, 74, 35, -70, -100, 12, 137, 89, -105, -192, -14, 228, 188, -
142, -338,-76,361, 361, -179, -579, -208, 569, 685, -209, -1040, 519, 992,1496,-229, -
2492, -1827, 3237, 9832, 12872, 9832, 3237, -1827, -2492, -229, 1496, 992, -519,-1040,
-209, 685, 569, -208, -579, -179, 361, 361, -76, -338, -142,188,228, 14, -192, -105, 89,
137, 12, -100, -70, 35, 74, 18, -45, -40, 14, 52, 42, 13, -6};

short X_retard[Nbr_coeff]; //échantillons retardés

interrupt void c_int11() //ISR

{short i;

X_retard [0]=input_sample(); //le nouvel échantillon d'entrée

sortie = 0; // initialiser la sortie du filtre

for (i = 0; i < N; i++)

sortie =sortie+ (h[i] * X_retard [i]); //sortie(n) += h(i)* x(n-i)

for (i = Nbr_coeff-1; i > 0; i--)

X_retard [i] = X_retard [i-1]; // déplacer les échantillons retardés

output_sample(sortie >> 15); //échantillon de sortie du filtre en format short

return;

}

void main()

{

comm_intr(); //init DSK, codec, McBSP

while(1); //infinite loop

}
```

Annexe D: Script MATLAB pour la génération de la somme
de deux signaux sinusoïdaux de fréquences 1kHz et 2kHz

```
F1=1000; % fréquence du premier signal
F2=2000; % fréquence du deuxième signal
fe=8000; % fréquence de l'échantillonnage
N= 2^16; % Nombre d'échantillons
temps=(0:N-1)/fe; %axe temporel en seconde
x1=sin(2*pi*F1*temps); %premier signal
x2=sin(2*pi*F2*temps); %deuxième signal
for u=1:100,
sound(x1+x2) ; %écouter le mélange des deux fréquences
end
```

Annexe A : Programme filtre FIR passe bas

```
#include "dsk6713_aic23.h" //fichier entête de support du codec
Uint32 fs = DSK6713_AIC23_FREQ_8KHZ; //fréquence d'échantillonnage
#define DSK6713_AIC23_INPUT_MIC 0x0015
#define DSK6713_AIC23_INPUT_LINE 0x0011
Uint16 inputsource=DSK6713_AIC23_INPUT_MIC; //selection de l'entrée
#define LOOPLength 8

short loopindex = 0; //indice de la table

short gain = 10; //gain

#include "dsk6713_aic23.h" // fichier support pour le codec et le DSK

#define Nbr_coeff 81

const lowpass[Nbr_coeff]= {-6, 13, 42, 52, 14, -40, -45, 18, 74, 35, -70, -100, 12, 137,
89, -105, -192, -14,228, 188, -142, -338,-76,361, 361, -179, -579, -208, 569, 685, -209, -
1040, 519, 992,1496,-229, -2492, -1827, 3237, 9832, 12872, 9832, 3237, -1827, -2492, -
229, 1496, 992, -519,-1040, -209, 685, 569, -208, -579, -179, 361, 361, -76, -338, -
142,188,228, 14, -192, -105, 89, 137, 12, -100, -70, 35, 74, 18, -45, -40, 14, 52, 42, 13, -
6};

union {Uint32 combo;short channel[2];}temp;

Uint32 Leftchannel,Rightchannel,data;

int a, i;

void main1() {comm_intr(); //init DSK, codec, McBSP
DSK6713_DIP_init();

hCodec= DSK6713_AIC23_openCodec(0,&config);//write audiob
while(1){ if(DSK6713_DIP8_get(0)==0){

float samples[50];

float result=0.0;

while(!DSK6713_AIC23_read(hCodec,&data)); //read audio
for (a = low -1; a>= 1; a--) samples[i]=samples[i-1];

temp.combo=data;

samples[0]=(float) temp.channel[0];

for (a = 0; a< low; a++) resultat+=lowpass[a]*samples[a];

while(!DSK6713_AIC23_write(hCodec,&data));} } }
```