

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique



Département d'Électronique

Projet de Fin d'Études

Présenté par

BOUALEM Abdelbassit
EDJEKOUANE Idir

Pour l'obtention du diplôme

INGÉNIEUR D'ÉTAT EN ÉLECTRONIQUE

Thème

Etude et réalisation d'un contrôleur embarqué à base de DSP

Devant le jury composé de :

Dr M.S. Ait Cheikh	Président
Mr M. Taghi	Examineur
Pr C. Larbes	Promoteur
Mr K. Boudjit	Co-promoteur

Promotion Juin 2011

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الملخص:

لقد تمكنا من خلال هذا العمل من دراسة وإنجاز متحكم في سرعة مدمج لمحرك لا تزامني ثلاثي الطور بواسطة معالج الإشارة الرقمي "DSP TMS320F2812" و اخترنا كطريقة للتحكم, التحكم السلمي. بدأنا الدراسة التجريبية بإنجاز بطاقة الاستطاعة التي تحوي موج ثلاثي الطور ذو مستويين بواسطة المركب المدمج IRAM136-0461G ثم قمنا بإنجاز بطاقة التوصيل بين المعالج الرقمي و بطاقة الاستطاعة. التطوير البرمجي يهدف إلى دمج خوارزمية تعديل عرض النبضة الشعاعية, باستعمال برنامج Code Composer Studio(CCS) و بطاقة التطوير eZdsp™ F2812 DSK.

الكلمات المفتاحية : محرك لا تزامني , موج ثلاثي الطور, معالج الإشارة الرقمي, مغير السرعة, خوارزمية تعديل عرض النبضة الشعاعية.

Résumé :

L'objectif de ce mémoire est l'étude et l'implémentation d'un variateur de vitesse embarqué à base de DSP TMS320F2812 de Texas Instrument pour un moteur asynchrone triphasé. La commande utilisée est une commande scalaire. Nous avons commencé l'implémentation hardware de notre système par la réalisation de la carte de puissance (onduleur triphasé à deux niveaux) à base du module IRAM136-0461G, puis une carte d'interface et d'adaptation entre le DSP et la carte de puissance. L'implémentation software consiste à implémenter l'algorithme MLI vectorielle sur le DSP TMS320F2812 en utilisant le *Code Composer Studio (CCS)* et la carte de développement *eZdsp™ F2812 DSK*.

Mots clés : Moteur asynchrone, Onduleur triphasé, DSP, Variateur de vitesse, Algorithme SVM.

Abstract :

The purpose of this thesis is the design and the implimentation of an embedded variable speed drive based on digital signal processor DSP TMS320F2812 from Texas Instrument for an inductor motor. The technique used is a scalar control. We started the hardware implementation by the power board which is based based on an Integrated Power Module IRAM136-0461G then an interface board between the DSP and the power board. The software implementation consists on implement the space vector modulation SVM algorithm on the DSP TMS320F2812 using the *Code Composer Studio (CCS)* and the development card *eZdsp™ F2812 DSK*.

Keywords : Inductor motor, Three-phase Inverter, DSP, Variable Speed Drive, SVM algorithm.

Remerciements

Nous remercions Allah de nous avoir prêté vie, santé et volonté pour achever ce travail.

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à Monsieur Cherif LARBES, Professeur à l'Ecole Nationale Polytechnique, et à Monsieur Kamel BOUDJIT, Chargé de Cours à l'Université des Sciences et de la Technologie Houari Boumédiène, de nous avoir confié ce sujet et de l'aide précieuse qu'ils nous ont toujours apportée jusqu'à l'achèvement de ce travail. Leurs insignes observations et suggestions ont été d'un grand apport pour son aboutissement. Qu'ils trouvent ici l'expression de notre sincère reconnaissance.

Nous remercions vivement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'être rapporteurs de notre mémoire.

Des remerciements vont également à tous ceux qui nous ont apporté, de près ou de loin, orientation, soutien et aide dans la réalisation et la concrétisation de ce travail.

Nos vifs remerciements à tous nos enseignants du primaire à l'université, ainsi que les électroniciens de la promotion 2011.

Nous tenons à remercier toutes personnes qui nous ont aidés dans notre cursus.

Dédicace

Je dédie ce travail à :
Mes parents,
Mes frères et mes sœurs,
Ainsi qu'à tous mes amis.
Idir

Je dédie ce modeste travail à :
Mes parents qui sont ma fierté,
Mes sœurs et mes frères pour leur soutien,
Mes Neveux : Bochra, Fatima, Mohamed Amine, Amel et Saif-el-Islam.
Ma grande famille,
Mes amis...
Abdelbassit

Table des matières

Résumé	1
Table des figures	9
Liste des tableaux	10
Abréviations	11
Introduction générale	12
1 Etat de l'art du moteur asynchrone et l'onduleur	14
1.1 Introduction	14
1.2 Le moteur asynchrone	14
1.2.1 Présentation de moteur asynchrone	15
1.2.2 Principe de fonctionnement et caractéristiques	16
1.3 La commande du moteur asynchrone	17
1.3.1 La commande scalaire	17
1.3.2 La commande vectorielle	18
1.3.3 Inversion du sens de rotation du moteur	20
1.4 Les onduleurs	20
1.4.1 Les onduleurs de tension monophasés	21
1.4.2 Les onduleurs de tension triphasés	23
1.4.3 Caractérisation de la tension fournie par l'onduleur	25
1.4.4 Domaines d'application des onduleurs	26
1.5 La commande des onduleurs	27
1.5.1 Techniques analogiques de génération de MLI	28
1.5.2 Techniques numériques de génération de MLI	30
1.6 Conclusion	32

2	Etude de la carte DSP TMS320F2812	33
2.1	Introduction	33
2.2	Généralités sur les DSPs	33
2.2.1	Historique	34
2.2.2	Les spécificités d'un DSP	35
2.2.3	Domaines d'application	38
2.2.4	La famille de DSP de Texas Instruments TMS320	39
2.3	Le DSP TMS320F2812	39
2.3.1	Descriptions et caractéristiques générales :	39
2.3.2	Le CPU (core)	42
2.3.3	Memory-Map	43
2.3.4	Les périphériques	44
2.3.5	L'outil de développement matériel (eZdsp™ F2812 DSK)	49
2.4	Conclusion	52
3	Implémentation Hardware	53
3.1	Introduction	53
3.2	Description générale du banc d'essai	54
3.3	Le module IRAM136-0461G	55
3.3.1	Caractéristiques du module IRAM136-0461G	56
3.4	Conception et réalisation de la carte de puissance	57
3.4.1	La technique de bootstrap et le choix de condensateur de bootstrap	60
3.4.2	Calcul du condensateur de bus continu	63
3.4.3	Alimentation de la carte de puissance	64
3.5	Conception et réalisation de la carte d'interface pour le DSP	64
3.5.1	Etage d'isolation	64
3.5.2	Les entrées CAP (unité de capture)	67
3.5.3	Les entrées ADC (Convertisseur Analogique Numérique)	68
3.6	Conclusion	69
4	Implémentation Software	70
4.1	Introduction	70
4.2	Outil de développement software	71
4.2.1	Code Composer Studio	71
4.3	Essais préliminaires sur la carte DSP	76
4.3.1	Génération d'un signal MLI	76

4.3.2	La conversion Analogique/Numérique	79
4.4	L'implémentation de l'algorithme de SVM	80
4.4.1	Représentation vectorielle des variables	80
4.4.2	Transformation triphasée biphasé	81
4.4.3	Principe de la modulation vectorielle	83
4.4.4	Algorithme d'implémentation de la modulation vectorielle	85
4.5	Résultats expérimentaux	86
4.6	Conclusion	90
	Conclusion générale et perspectives	91
	Bibliographie	93
	Annexe A	95
	Annexe B	98
	Annexe C	100

Table des figures

1.1	Coupe axiale d'un moteur à Induction	15
1.2	Symbole de la machine AS à cage et la photo de son rotor	15
1.3	Symbole de la machine asynchrone à rotor bobiné	16
1.4	Principe du champ tournant	17
1.5	Schéma équivalent du moteur asynchrone	18
1.6	Principe de la commande vectorielle	19
1.7	Principe de l'orientation du flux rotorique	20
1.8	Onduleur monophasé en demi-pont	21
1.9	Différentes formes des signaux	22
1.10	Onduleur en pont monophasé	22
1.11	Tensions et leurs spectres	23
1.12	Onduleur triphasé	23
1.13	Les signaux de commande, les tensions des 3 bras de l'onduleur et la tension de la phase (a) du récepteur	25
1.14	Schéma de principe d'un onduleur pour une alimentation de secours	26
1.15	Schéma de principe d'un onduleur pour la conduite d'un moteur asynchrone	27
1.16	PWM à Hystérésis, controle de courant et logique de commutation	28
1.17	Principe de la génération d'une MLI triangulo-sinusoïdale triphasée [4]	29
1.18	Formes d'ondes des commandes et des tensions de sorties en triphaé	30
2.1	L'architecture de Von Neumann	35
2.2	L'architecture de Harvard	35
2.3	L'architecture de Harvard modifié	36
2.4	Représentation simplifiée de l'unité MAC	37
2.5	La structure du Pipeline	38
2.6	Schéma bloc simplifié de TMS320F2812	41
2.7	Schéma bloc détaillé de TMS320F2812 [16]	42

2.8	Schéma bloc de C28x	43
2.9	Memory Map du TMS320F2812	44
2.10	Schéma blocs de Event Manager A	45
2.11	Les sources d'interruptions	47
2.12	Les vecteurs d'interruptions vers le CPU	48
2.13	schéma synoptique du module DSK eZdsp F2812	50
2.14	Vue supérieure de la carte SDK eZdsp F2812	51
3.1	Photo du banc d'essai	54
3.2	Schéma synoptique du banc d'essai	54
3.3	Le contrôleur embarqué	55
3.4	Le module IRAM136-0461G	56
3.5	Schéma interne du module IRAM136-0461G	57
3.6	Schéma synoptique de carte de puissance	58
3.7	Schéma électrique de la carte de puissance	58
3.8	La carte de puissance	59
3.9	Schéma de circuit de bootstrap pour une phase	61
3.10	Les valeurs de C_{bs} recommandées en fonction des fréquences de porteuse de MLI appliquées	63
3.11	Onde de tension redressée montre les périodes de conduction de V_{min} à V_{max}	63
3.12	Circuit d'alimentation de 15V et 12V	64
3.13	Interface d'isolation entre le DSP et la carte de puissance	65
3.14	L'optocoupleur et son schéma interne	66
3.15	Circuit d'isolation	66
3.16	Circuit d'alimentation de 5V	67
3.17	Les signaux PWM1 et PWM2	67
3.18	Schéma électrique CAP/QEP	68
3.19	Schema électrique de l'entrée de l'ADC	68
4.1	Workspace du logiciel code composer	72
4.2	Les étapes de développement	73
4.3	La fenêtre de création de projet	74
4.4	Le Watch Window	76
4.5	Génération de deux PWM en mode asymétrique	77
4.6	Génération de deux PWM en mode symétrique	77
4.7	Les signaux PWM1 et PWM2	79

4.8	Comparaison entre les valeurs mesurées et les valeurs calculées	80
4.9	Représentation vectorielle des variables	81
4.10	Les différentes combinaisons de commande des interrupteurs dans la commande SVM des onduleurs à 2 niveaux	81
4.11	Représentation des 8 états de l'onduleur dans le repère (α, β) et définition des secteurs	82
4.12	Décomposition d'un vecteur de consigne	83
4.13	Notion d'accessibilité des vecteurs	84
4.14	SVM avec motifs MLI centrés	85
4.15	Algorithme d'implémentation de la SVM	86
4.16	Signaux MLI S1 à S6 avec $f_{PWM}=5\text{kHz}$	87
4.17	Tensions simples V_a , V_b et V_c	88
4.18	Tensions composées V_{ab} , V_{bc} et V_{ca}	89
4.19	Spectre fréquentiel de la tension simple et composée	90

Liste des tableaux

2.1	Les principales caractéristiques de F2812	40
2.2	Connecteurs de l'eZdsp F2812	51
2.3	Les pins du connecteur P8.	52
3.1	Liste des composants de la carte de puissance	59
4.1	Valeurs numériques mesurées	80
4.2	Table d'excitation des interrupteurs de l'onduleur à 2 niveaux	82

Abréviation

AC : Alternating Current.
ADC : Analog-to-Digital Converter.
ALU : Arithmetic Logic Unit.
EV : Event Manager.
CCS : Code Composer Studio.
DC : Direct Current.
DSP : Digital Signal Processor.
eCAN : Enhanced Controller Area Network.
GP : General Purpose.
GPIO : General Purpose Input Output.
I/O : Input/Output.
JTAG : Joint Test Action Group.
LED : Light Emitting Diode.
MAC : Multiply and Accumulate Operation.
MAS : Moteur ASynchrone.
MIPS : Millions of Instructions Per Second.
McBSP : Multichannel Buffered Serial Port.
MLI : Modulation de Largeur d'Impulsion.
PIE : Peripheral Interrupt Expansion.
PWM : Pulse Width Modulation.
QEP : Quadrature-Encoder Pulse.
RAM : Random Access Memory.
ROM : Read Only Memory.
SARAM : Single Access RAM.
SCI : Serial Communications Interface.
SPI : Serial Peripheral Interface.
SVM : Space Vector Modulation.
TI : Texas Instruments.

Introduction générale

La commande à vitesse variable des entraînements électriques, a bénéficié ses dernières années d'avancées méthodologique et technologique significatives. En effet, les progrès de l'électronique numérique et le développement des composants de l'électronique de puissance permet aujourd'hui de mettre en œuvre des algorithmes de commande non envisageables il y a une dizaine d'années.

Dans diverses applications industrielles modernes, la machine asynchrone bénéficie d'une attention particulière due essentiellement à sa fiabilité, à sa robustesse, son coût relativement faible et sa simplicité de construction. Le travail envisagé porte sur un moteur asynchrone alimenté à partir d'un onduleur de tension variable en tension et en fréquence, avec le rapport V/f constant à Modulation de Largeur d'Impulsion (MLI).

Une large gamme de microprocesseurs et microcontrôleurs sont actuellement dédiés spécialement à la commande des machines, parmi eux les DSP (Digital Signal Processor). Leur puissance de calcul et l'efficacité de leur contrôle permettent de commander avec une haute résolution et mettre en œuvre des commandes complexes.

Ainsi, l'objectif de notre mémoire est d'étudier et de réaliser d'un contrôleur embarqué de vitesse pour un moteur asynchrone triphasé, à base d'un onduleur de tension piloté par une commande MLI vectorielle générée par un DSP.

Notre mémoire est organisé en quatre chapitres :

Dans le premier chapitre, nous avons donné quelques principes sur le moteur asynchrone, son principe de fonctionnement et les commandes utilisées. Les onduleurs de tension à leur tour sont étudiés, en donnant leur principe, et en citant quelques caractéristiques puis on s'intéresse aux stratégies de commandes utilisées dans les onduleurs triphasés à deux niveaux. Parmi elles, deux variantes de commande se distinguent : la MLI triangulo-sinusoïdale et la MLI vectorielle.

Le deuxième chapitre est consacré aux DSPs, une introduction générale aux DSPs en premier lieu puis une description architecturale du DSP TMS320F2812, et l'on termine le chapitre par une description de la carte de développement eZdsp™ F2812 DSK qu'on a utilisé pour l'implémentation.

Dans le troisième chapitre, on aborde l'implémentation Hardware. La première étape de notre réalisation concerne la carte de puissance, cette carte est basée sur un module de puissance IRAM136-0461G, nous allons expliquer le fonctionnement de ce module et comment choisir les composants de son montage. La deuxième étape est la réalisation d'une carte d'interface générale pour le DSP, cette carte contient un étage d'isolation en utilisant les optocoupleurs, un étage pour les entrées analogiques et un étage pour adapter les signaux des capteurs.

Au dernier chapitre, on aborde l'implémentation Software. On commence par la présentation de l'outil de développement logiciel, le Code Composer Studio CCS, ensuite, un essai concret de la génération de 6 signaux PWM (Pulse Width Modulation) nous permettra de mettre en évidence tous les paramètres devant être pris en compte, et on termine le chapitre par l'élaboration de l'algorithme de MLI vectorielle (SVM).

Enfin, le travail sera clôturé par une conclusion générale et des perspectives d'avenir.

Chapitre 1

Etat de l'art du moteur asynchrone et l'onduleur

1.1 Introduction

Dans la première partie de ce chapitre, nous allons présenter brièvement le moteur asynchrone, le principe de fonctionnement et les caractéristiques ainsi que ces deux types de commande qui sont la commande scalaire et la commande vectorielle. La deuxième partie est consacrée pour exposer les différents types des onduleurs, on s'intéresse particulièrement à l'onduleur triphasé à deux niveaux qui sera utilisé dans l'implémentation Hardware, leurs caractéristiques et les stratégies de commande.

1.2 Le moteur asynchrone

Le moteur asynchrone, ou moteur à induction, est actuellement le moteur électrique dont l'usage est le plus répandu dans l'industrie. Son principal avantage réside dans l'absence de contacts électriques glissants, ce qui conduit à une structure simple, robuste et facile à construire. Relié directement au réseau industriel à tension et fréquence constantes, il tourne à vitesse peu différente de la vitesse synchrone ; c'est lui qui est utilisé pour la réalisation de la quasi-totalité des entraînements à vitesse constante. Grâce à l'électronique de puissance le moteur asynchrone permet aussi la réalisation d'entraînements à vitesse variable et la place qu'il occupe dans ce domaine ne cesse de croître.

1.2.1 Présentation de moteur asynchrone

La machine asynchrone est constituée des principaux éléments suivants [1] :

- **Stator** : feuilleté, support d'un bobinage polyphasé générant un champ tournant a $2p$ poles.
- **Rotor** : il y a deux types :
 - Rotor à cage : Rotor feuilleté comprenant des encoches contenant des barres non isolées (insérées en Cu ou injection en Al), court-circuitées aux extrémités par deux anneaux.
 - Rotor bobiné : Rotor feuilleté, comprenant des encoches contenant un bobinage triphasé, isolé. Ce dernier est relié a 3 bagues sur lesquelles frottent des charbons reliés a un rhéostat.

La figure 1.1 illustre une coupe axiale d'un moteur à induction.

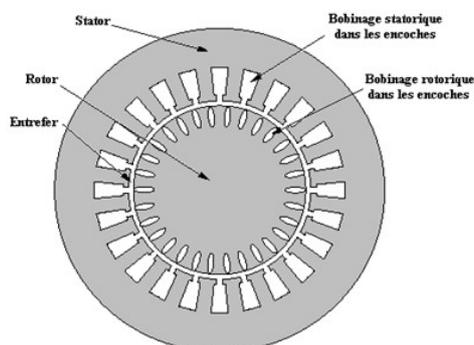


FIGURE 1.1 – Coupe axiale d'un moteur à Induction

La figure 1.2 montre un rotor à cage d'écureuil et le symbole de la machine asynchrone à rotor à cage.

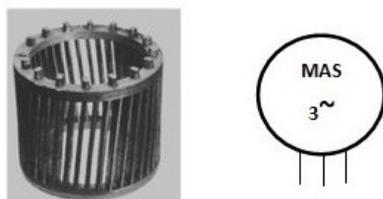


FIGURE 1.2 – Symbole de la machine AS à cage et la photo de son rotor

La figure 1.3 montre un rotor bobiné et le symbole de la machine asynchrone à rotor bobiné.



FIGURE 1.3 – Symbole de la machine asynchrone à rotor bobiné

1.2.2 Principe de fonctionnement et caractéristiques

Lorsque les enroulements statoriques sont alimentés, ils produisent un champ tournant (figure 1.4). Les lignes de flux qui traversent le rotor se déplacent par rapport aux barres conductrices ; un courant induit prend naissance dans ces barres mises en court-circuit par leurs extrémités. L'action du champ magnétique sur ce courant génère des forces (Loi de Laplace), donc un couple qui entraîne le rotor dans le sens du champ tournant. Le couple produit par la machine est un couple moteur car il s'oppose au couple appliqué par la charge qui est à l'origine du glissement. [2]

Soit un moteur asynchrone à $2p$ pôles alimenté à partir du réseau alternatif de fréquence f . Le stator crée un flux tournant à la vitesse angulaire synchrone Ω :

$$\Omega = \frac{2\pi f}{p} = \frac{\omega}{p} \quad (1.1)$$

Le rotor tourne à une vitesse Ω' plus petite que Ω . L'écart entre Ω' et Ω augmente lorsque le couple résistant sur l'arbre du rotor augmente. On appelle glissement l'écart des vitesses angulaires synchrone Ω et réelle Ω' rapporté à la vitesse synchrone Ω' :

$$g = \frac{\Omega - \Omega'}{\Omega} = \frac{\omega - \omega'}{\omega} = \frac{N - N'}{N} \quad (1.2)$$

Ou $N = \frac{\Omega}{2\pi}$: vitesse synchrone et $N' = \frac{\Omega'}{2\pi}$: vitesse de rotor.

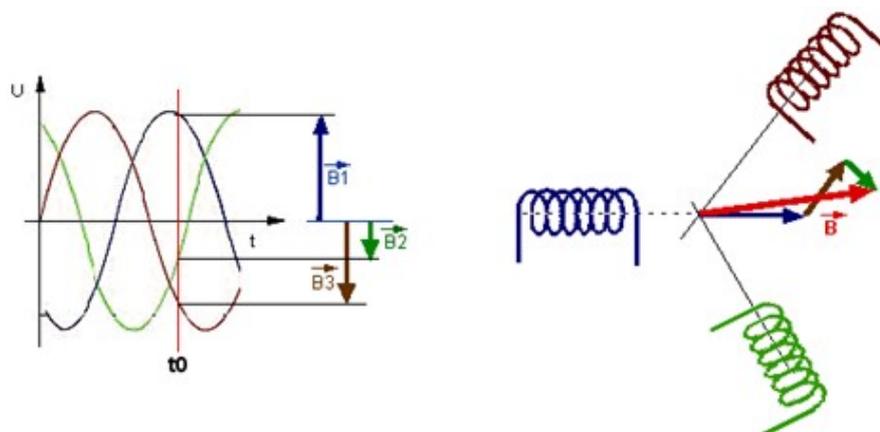


FIGURE 1.4 – Principe du champ tournant

1.3 La commande du moteur asynchrone

On distingue deux types de commandes : la commande scalaire et la commande vectorielle.

- La commande scalaire : basée sur le modèle en régime permanent, elle est simple à implanter avec une dynamique lente. Elle contrôle les grandeurs en amplitude.
- La commande vectorielle : basée sur le modèle transitoire, précise et rapide elle permet le contrôle du couple. Elle est chère car nécessitant souvent des capteurs, encodeur incrémental ou estimateur de vitesse, DSP... Elle contrôle les grandeurs en amplitude et en phase. [4]

1.3.1 La commande scalaire

Plusieurs commandes scalaires existent selon que l'on agit sur le courant ou sur la tension. Elles dépendent surtout de la topologie de l'actionneur utilisé (onduleur de tension ou de courant). L'onduleur de tension étant maintenant le plus utilisé en petite et moyenne puissance, c'est la commande en V/f qui est la plus utilisée. [2]

Contrôle en V/f de la machine asynchrone

La figure 1.5 représente le schéma équivalent par phase d'une machine asynchrone. Le flux est créé par le courant circulant dans l'inductance magnétisante L_m . Les performances optimales du moteur sont obtenues si le flux, et donc le courant magnétisant, est maintenu sensiblement constant sur toute la gamme de vitesse. [2]

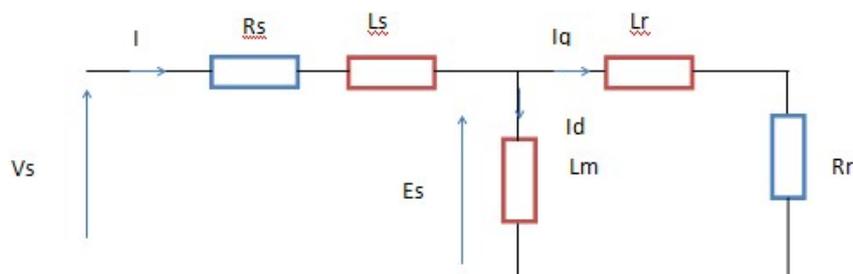


FIGURE 1.5 – Schéma équivalent du moteur asynchrone

Tel que :

L_s : inductance de fuite statorique.

R_s : résistance statorique.

L_m : inductance mutuelle, inductance magnétisante.

R_r : résistance rotorique.

L_r : inductance de fuite rotorique.

Le courant magnétisant peut être calculé par l'expression : $I_d = \frac{E_s}{L_m \omega}$ avec

$E_s = V_s - (R_s + L_s \omega)I$, en négligeant $(R_s + L_s \omega)I$, on obtient :

$$I_d = \frac{V_s}{f} \frac{1}{2\pi L_m} \quad (1.3)$$

Le courant magnétisant peut donc être maintenu constant en maintenant le rapport $\frac{V_s}{f}$ sensiblement constant. Toutefois, à des fréquences et des tensions faibles, le terme $(R_s + L_s \omega)I$ ne peut plus être négligé, et en particulier le terme $R_s I$. Pour maintenir le flux constant, quel que soit la charge du moteur et donc quel que soit le courant absorbé, le maintien d'un rapport $\frac{V_s}{f}$ constant n'est pas suffisant. [2]

1.3.2 La commande vectorielle

Le contrôle vectoriel consiste à générer les tensions MLI requises de façon à maintenir en tout temps un flux constant dans l'entrefer et obtenir un décalage aussi proche que possible de 90° entre les vecteurs d'espaces du flux et de la force magnétomotrice des courants rotoriques, et aussi un couple maximal. [3]

Le principe du découplage permet de modéliser la machine asynchrone comme une machine à courant continu, par application de cette nouvelle technique de commande.

Pour réaliser un contrôle similaire à celui des machines à excitation séparée, il est nécessaire d'orienter le flux en quadrature avec le couple d'où la méthode du flux

orientée, et ainsi la composante directe du courant statorique i_{ds} prend la direction du flux rotorique. Le couple dépend alors uniquement de la composante en quadrature i_{qs} . [5]

Le contrôle vectoriel de flux avec un moteur asynchrone permet de maîtriser séparément le courant magnétisant et le courant actif. Le variateur vectoriel reçoit les informations sur la vitesse et la position angulaire du rotor grâce à un codeur incrémental accouplé au moteur. Ces valeurs sont traitées numériquement dans un calculateur en fonction de la vitesse et du couple désiré. Ceci va agir sur les courants d'onduleur de manière à positionner le flux stator en quadrature avec le courant induit dans le rotor comme dans un moteur à courant continu. [5]

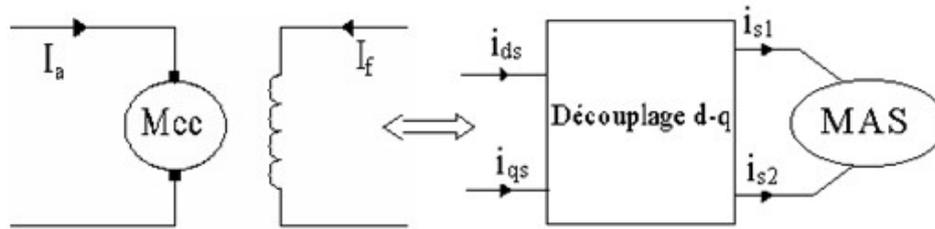


FIGURE 1.6 – Principe de la commande vectorielle

Le couple électromagnétique de la machine à courant continu :

$$C_{em} = k \cdot \phi_a \cdot I_a \quad (1.4)$$

avec $\phi_a = k_f I_f$.

Le couple électromagnétique de la machine asynchrone :

$$C_{em} = \frac{p \cdot M}{L_r} (\phi_{dr} \cdot i_{qs} - \phi_{qr} \cdot i_{ds}) \quad (1.5)$$

Dans le modèle de PARK de la machine asynchrone, nous choisissons un référentiel lié au champ tournant tel que l'axe 'd' coïncide avec la direction du flux (rotorique, statorique, ou d'entrefer), pour maintenir la composante directe constante et à annuler la composante en quadrature, (figure1.7).

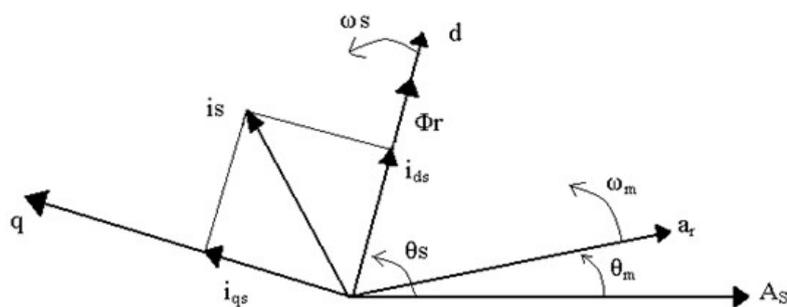


FIGURE 1.7 – Principe de l'orientation du flux rotorique

1.3.3 Inversion du sens de rotation du moteur

Le sens de rotation du moteur asynchrone triphasé s'inverse en inversant le sens de rotation du flux tournant statorique. Il suffit pour cela de permuter les entrées de deux phases. L'inversion du sens du flux tournant pendant la rotation du moteur entraîne d'abord le freinage puis l'arrêt et enfin la rotation en sens inverse du moteur. [3]

1.4 Les onduleurs

L'onduleur est un convertisseur statique continu/alternatif qui assure des échanges d'énergie entre une source et une charge. Il permet d'alimenter la charge en alternatif, à une fréquence fixe ou variable à partir d'une source de puissance continue (bus continu soit une ou des batterie(s), des redresseurs, des panneaux solaires etc...).

Les onduleurs de tension : sont alimentés par une source de tension continue, d'une impédance interne négligeable et de tension constante peu affectée par les variations du courant qui la traverse. Les onduleurs de courant sont alimentés par une source de courant. [4]

Un onduleur sert à convertir une puissance continue en une puissance alternative. Il existe deux grandes catégories d'onduleurs :

1. Les onduleurs autonomes (ou oscillateurs) qui génèrent leur propre fréquence et leur propre tension alternative. Dans ces onduleurs, la commutation est forcée.
2. Les onduleurs non autonomes dont la fréquence et la tension alternative sont imposées par le réseau qu'ils alimentent. Dans ces onduleurs, la commutation est

naturelle. Elle est naturelle en ce sens que ce sont les tensions alternatives du réseau qui effectuent le transfert du courant d'un bras de commande à l'autre. [3]

1.4.1 Les onduleurs de tension monophasés

Ce type d'onduleurs est destiné à alimenter des charges alternatives monophasées. On distingue deux configurations de base :

Onduleur en demi-pont

Le principe de l'onduleur monophasé en demi-pont peut être expliqué à partir de la figure 1.8. On dispose d'une source de tension continue à point milieu et de 2 interrupteurs (S_1 et S_2).

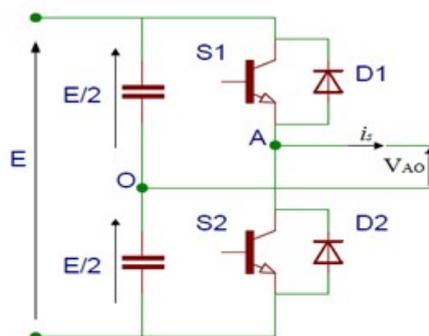


FIGURE 1.8 – Onduleur monophasé en demi-pont

Quand seul l'interrupteur S_1 est fermé pendant une demi-période, la tension de sortie V_s prend la valeur $E/2$. Quand seul l'interrupteur S_2 est conducteur la tension V_s prend la valeur $-E/2$. La logique du circuit de commande est conçue tel que S_1 et S_2 ne soient jamais allumés en même temps.

Le courant $i_s(t)$ dépend de la forme de la tension $V_{ao}(t)$ et du type de récepteur. Sur une charge résistive pure R, les diodes D_1 et D_2 ne jouent aucun rôle. Tandis que sur une charge (R et L) les diodes D_1 et D_2 assurent un retour d'énergie à la source continue, elles sont appelées les diodes de récupération.

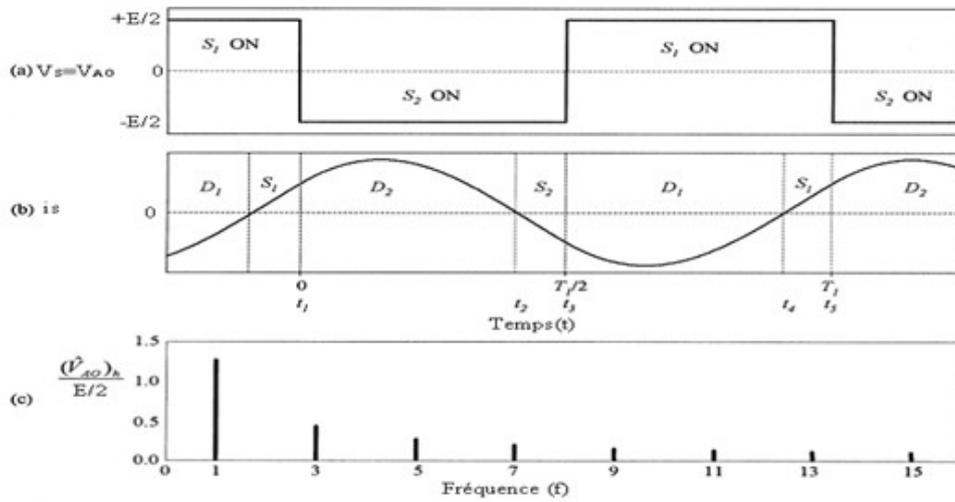


FIGURE 1.9 – Différentes formes des signaux

Onduleur monophasé en pont

Pour éviter la nécessité d'une source à point milieu et doubler la tension de sortie $V(t)$ on utilise un onduleur en pont. Dans ce cas on commande sur une demi-période les interrupteurs (S_1, S_4) et les interrupteurs (S_2, S_3) sur le reste de période. Les figures 1.10 et 1.11 montrent respectivement la structure d'un onduleur monophasé en pont et les formes des différentes grandeurs.

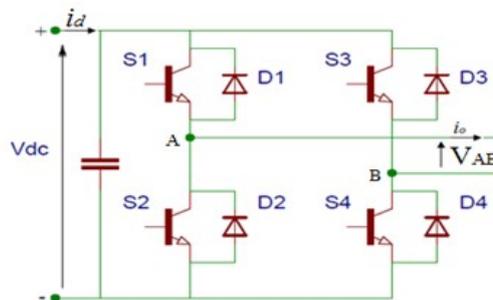


FIGURE 1.10 – Onduleur en pont monophasé

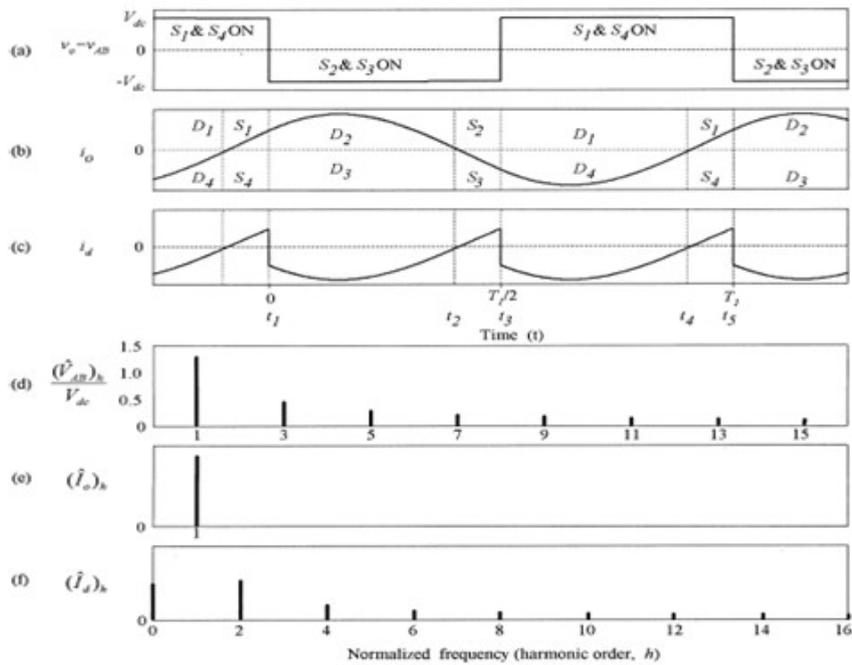


FIGURE 1.11 – Tensions et leurs spectres

1.4.2 Les onduleurs de tension triphasés

Les onduleurs monophasés sont utilisés pour des applications de faible puissance, alors que les onduleurs triphasés couvrent la gamme de moyenne et de forte puissance.

Trois onduleurs monophasés en demi-pont (ou en pont) peuvent être connectés en parallèle pour constituer un onduleur triphasé. La figure 1.12 montre un schéma de principe d'un onduleur triphasé de tension.

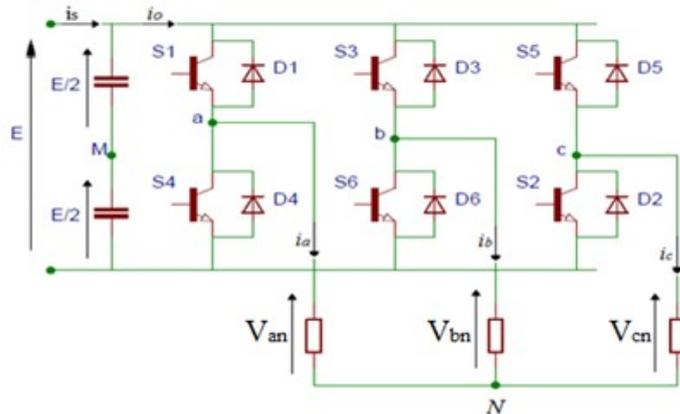


FIGURE 1.12 – Onduleur triphasé

L'objectif de cette topologie est de fournir une source de tension triphasée, dont l'amplitude, la phase et la fréquence sont contrôlables. L'onduleur de tension triphasé est constitué de trois cellules de base. Une charge triphasée est raccordée aux points milieux a, b et c. Les impulsions d'allumage des onduleurs monophasés doivent être avancées ou retardées de 120° les uns par rapport aux autres afin d'obtenir 3 tensions triphasées équilibrées (pour le fondamental).

Le récepteur étant couplé en étoile, les tensions de phase (V_{an} , V_{bn} et V_{cn}) sont liées aux tensions des trois bras de l'onduleur (V_{aM} , V_{bM} et V_{cM}) par :

$$V_{aM} = V_{an} + V_{nM} \quad (1.6)$$

$$V_{bM} = V_{bn} + V_{nM} \quad (1.7)$$

$$V_{cM} = V_{cn} + V_{nM} \quad (1.8)$$

En sommant les trois tensions V_{aM} , V_{bM} et V_{cM} , il vient :

$$V_{aM} + V_{bM} + V_{cM} = 3V_{nM} + (V_{an} + V_{bn} + V_{cn}) \quad (1.9)$$

Comme le récepteur est triphasé équilibré par conséquent :

$$i_a = \frac{V_{an}}{Z}, i_b = \frac{V_{bn}}{Z}, i_c = \frac{V_{cn}}{Z} \quad (1.10)$$

De plus ce récepteur est à neutre isolé donc la somme des courant est nulle :

$$i_a + i_b + i_c = \frac{(V_{an} + V_{bn} + V_{cn})}{Z} = 0 \quad (1.11)$$

D'où :

$$V_{nM} = \frac{V_{aM} + V_{bM} + V_{cM}}{3} \quad (1.12)$$

Et les tensions simples sont alors :

$$V_{an} = +\frac{2}{3}V_{aM} - \frac{1}{3}V_{bM} - \frac{1}{3}V_{cM} \quad (1.13)$$

$$V_{bn} = -\frac{1}{3}V_{aM} + \frac{2}{3}V_{bM} - \frac{1}{3}V_{cM} \quad (1.14)$$

$$V_{cn} = -\frac{1}{3}V_{aM} - \frac{1}{3}V_{bM} + \frac{2}{3}V_{cM} \quad (1.15)$$

Il est à noter que les commandes des interrupteurs S_4 , S_6 , S_2 sont complémentaires dans cet ordre aux commandes des interrupteurs S_1 , S_3 , S_5 .

Les tensions (V_{an} , V_{bn} et V_{cn}), ont la même forme mais elles sont décalées entre elles par 120° . Les tensions : $U_{ab} = V_{an} - V_{bn}$, $U_{bc} = V_{bn} - V_{cn}$, $U_{ca} = V_{cn} - V_{an}$ forment les tensions entre phases.

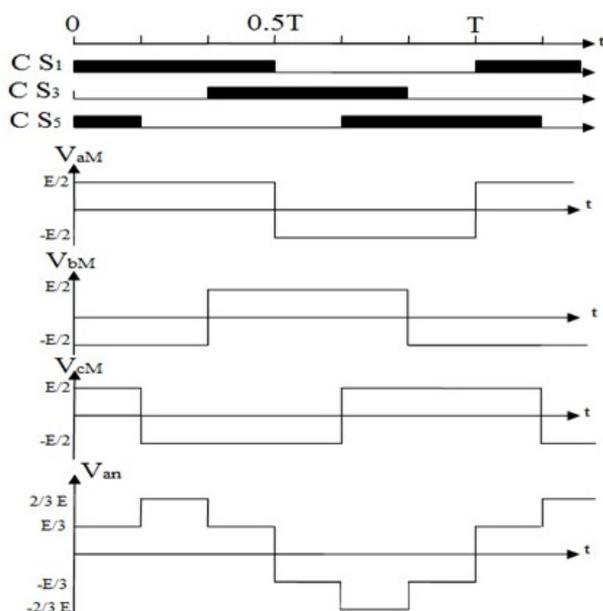


FIGURE 1.13 – Les signaux de commande, les tensions des 3 bras de l'onduleur et la tension de la phase (a) du récepteur

1.4.3 Caractérisation de la tension fournie par l'onduleur

Les onduleurs sont caractérisés par la qualité de la tension alternative qu'ils fournissent à leurs sorties. Celle-ci est évaluée par 3 paramètres : le facteur d'harmonique, la distorsion totale d'harmonique et l'harmonique de plus bas d'ordre. [7]

Le facteur d'harmonique HF

Il représente la mesure de la contribution individuelle d'une harmonique (par exemple celle de la nième harmonique).

$$HF_n = \frac{V_n}{V_1} \quad (1.16)$$

Avec :

V_1 : Valeur efficace du fondamentale.

V_n : Valeur efficace du nième harmonique.

La distorsion totale d'harmonique THD

Il représente la mesure du rapprochement dans la forme entre l'onde de tension et son fondamental :

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} (V_n)^2}}{V} = \frac{\sqrt{V^2 - (V_1)^2}}{V} \leq 1 \quad (1.17)$$

Ou encore :

$$THD = \frac{V}{V_1} \geq 1 \quad (1.18)$$

L'harmonique d'ordre le plus bas (LoH)

C'est celui dont la fréquence est la plus proche du fondamental et dont l'amplitude est supérieure ou égale de 3% de celle du fondamentale.

1.4.4 Domaines d'application des onduleurs

Les deux grands domaines d'applications classiques des onduleurs de tension sont les alimentations de secours et la commande des moteurs alternatifs. Ils sont également caractéristiques de deux grandes familles, respectivement celle des systèmes à fréquence fixe et celle des systèmes à fréquence variable. [4]

Domaine des fréquences fixes

Dans ce domaine, l'onduleur de tension est destiné à pallier les défaillances ou même l'absence d'un réseau alternatif en recréant la tension correspondante à partir d'une batterie. Le schéma de principe d'une telle application est donné par la figure 1.14. Cette application requiert généralement la fourniture d'une tension de sortie très pure, donc sans les harmoniques de découpage, d'où la nécessité d'un filtre de sortie. [4]

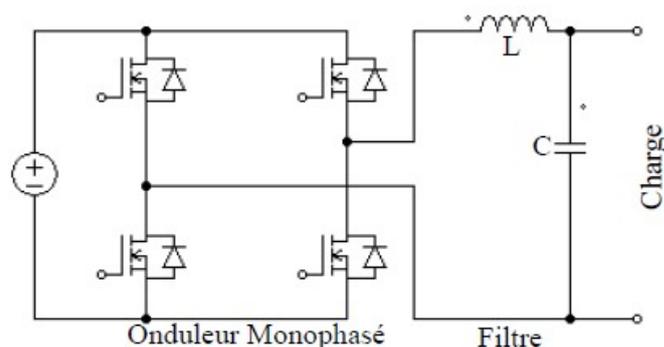


FIGURE 1.14 – Schéma de principe d'un onduleur pour une alimentation de secours

Domaine des fréquences variables

Dans ce domaine, l'onduleur est généralement alimenté par l'intermédiaire d'un réseau alternatif. La source continue est donc un redresseur suivi d'un filtre d'entrée qui peut avoir une double fonction, à savoir éliminer les composantes harmoniques

de courant issues de l'onduleur, et les composantes harmoniques de tension dues au redressement.

Une autre différence notable par rapport au cas précédent réside dans le domaine de fonctionnement beaucoup plus étendu tant en fréquence fondamentale (de quelques hertz à quelques centaine de hertz) qu'en amplitude.

Le domaine d'applications des onduleurs de tension le plus connu est sans doute celui de la variation de vitesse des machines à courant alternatif, ce objet de notre projet. Le schéma de principe d'une telle application est donné par la figure 1.15. [4]

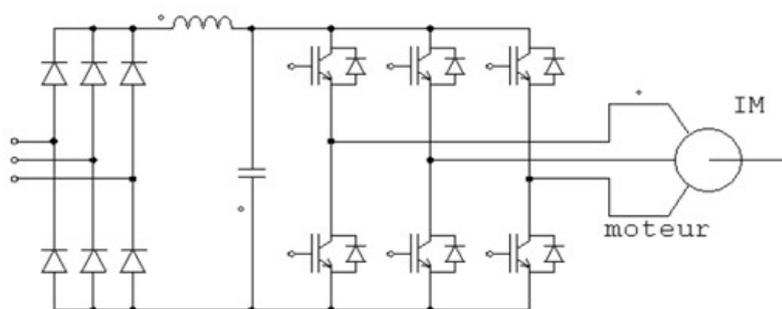


FIGURE 1.15 – Schéma de principe d'un onduleur pour la conduite d'un moteur asynchrone

1.5 La commande des onduleurs

A l'âge de la révolution des interrupteurs semi-conducteurs de puissance à commutation très rapide, l'utilisation des convertisseurs de type onduleur de tension à MLI (Modulation de Largeur d'Impulsion), trouve énormément d'intérêt, en particulier dans l'entraînement des machines à courant alternatif.

La commande de l'onduleur par la modulation de largeur d'impulsion (MLI) permet de générer, par alternance de la tension alternative, une tension composée de plusieurs créneaux de largeurs variables. Elle permet ainsi, d'obtenir un fondamental de tension variable en amplitude et en fréquence.

La modulation est le processus de variation d'une caractéristique d'une porteuse à l'aide d'une information. En Electronique de Puissance, cette information est l'ampli-

tude, la fréquence et la phase d'une tension ou d'un courant. Le contrôle de l'impulsion du courant ou de la tension permet d'obtenir un signal d'amplitude donnée. [8]

Plusieurs stratégies de commande des onduleurs ont été développées dans la littérature dont les principes consistent soit :

1.5.1 Techniques analogiques de génération de MLI

Consistent à la génération des signaux de commande des interrupteurs de puissance par l'asservissement de la tension de sortie de l'onduleur à une référence de tension sinusoïdale : c'est la commande dite implicite. Il existe plusieurs techniques analogiques telle que principalement la MLI triangulo-sinusoïdale (engendrée), la modulation Delta et le contrôle de courant à hystérésis. [4]

Le contrôle de courant à hystérésis

Le contrôle de courant à hystérésis est une technique MLI très simple à implémenter et elle a un contrôle direct de courant. La logique de commutation est réalisée à l'aide de trois contrôleurs d'hystérésis, un pour chaque phase (figure 1.16). Le contrôle de courant à hystérésis, connue encore comme le bang-bang control, est appliqué sur les trois phases séparément. Chaque contrôleur détermine l'état de commutation d'un demi-pont de l'onduleur d'une façon à maintenir le courant correspondant dans une bande hystérésis de courant.

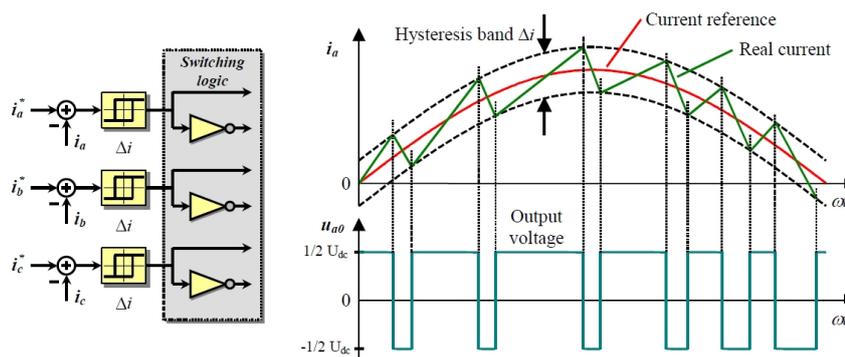


FIGURE 1.16 – PWM à Hystérésis, contrôle de courant et logique de commutation

La technique MLI triangulo-sinusoïdale

La simplicité d'implémentation de cette technique par des moyens analogiques lui a permis d'être la plus utilisée jusqu'à l'apparition récente des microprocesseurs qui a

rendu possible l'implémentation d'autres techniques numériques compétitives donnant ainsi un large choix.

Le principe de cette technique consiste à comparer un signal triangulaire $V_p(t)$ d'amplitude fixe appelé porteuse à un signal $V_{ref}(t)$ d'amplitude et fréquence variable appelé référence ou modulante (figure 1.17). L'intersection de ces signaux donne les instants de commutation des interrupteurs.

Le signal de référence représente l'image de la sinusoïde qu'on désire à la sortie de l'onduleur. Quant à la porteuse, elle définit la cadence de la commutation des interrupteurs statiques de l'onduleur, c'est un signal de haute fréquence par rapport au signal de référence. [7]

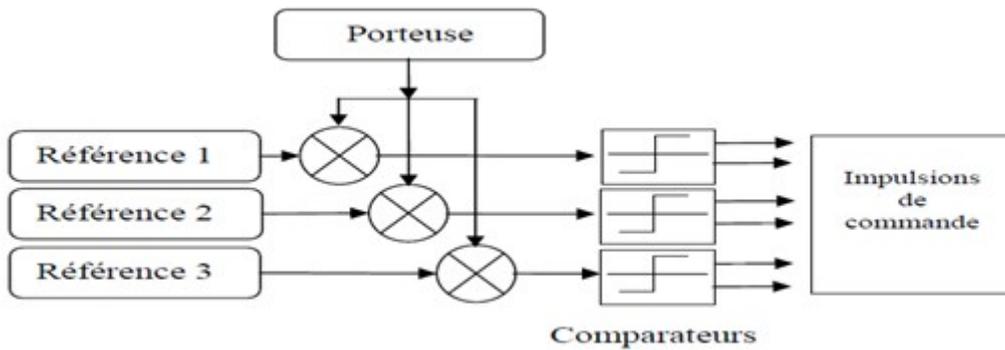


FIGURE 1.17 – Principe de la génération d'une MLI triangulo-sinusoïdale triphasée [4]

La technique MLI est caractérisée par deux paramètres :

- L'indice de modulation m égal au rapport de la fréquence de la porteuse f_p à celle de la référence f_{ref} .

$$m = \frac{f_p}{f_{ref}} \quad (1.19)$$

- Le coefficient de réglage r , égal au rapport de l'amplitude de référence \widehat{V}_{ref} à celle de la porteuse \widehat{V}_p .

$$r = \frac{\widehat{V}_{ref}}{\widehat{V}_p} \quad (1.20)$$

La modulation est dite synchrone si m est entier, elle est dite asynchrone dans l'autre cas. En modulation synchrone, si le maximum de la référence correspond à un pic de la porteuse, on dit que le calage est optimal et la tension alternative bénéficie d'une double symétrie par rapport au quart et au milieu de la période.

La figure 1.18 illustre la forme d'onde des commandes des interrupteurs et les tensions de la sortie.

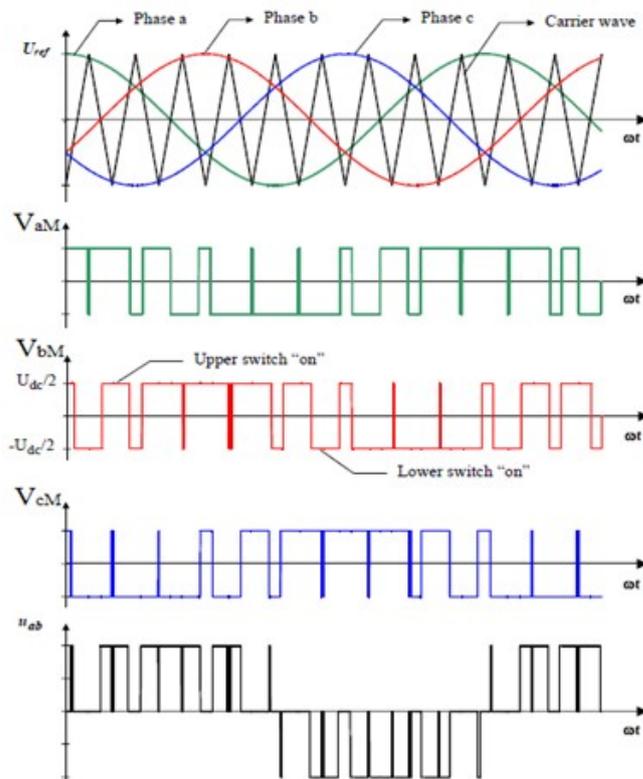


FIGURE 1.18 – Formes d'ondes des commandes et des tensions de sorties en triphasé

1.5.2 Techniques numériques de génération de MLI

Consistent à la détermination des instants de commutation des composants de puissance formant l'onduleur par le biais du développement en série de Fourier des formes d'onde souhaitées en sortie répondant à des critères bien définies (Taux d'harmoniques, valeur du terme fondamental,..) : C'est la commande dite explicite où la commande des interrupteurs se fait d'une façon numérique. Il existe plusieurs techniques numériques telle que la technique MLI calculée (programmée) et la technique MLI vectorielle qui est l'objet de notre travail.

La MLI calculée de PATEL et HOFT

Le principe de la technique de modulation par élimination d'harmoniques a été introduit pour la première fois par Turnbull en 1964 puis développé par Patel et Hoft. Cette technique consiste à former l'onde de sortie d'une succession de créneaux de

largeurs variables et contrôlables. Les angles de commutation sont déterminés de façon à éliminer certains harmoniques gênants dans l'onde de sortie. Améliorant ainsi le rendement du système onduleur–machine par la réduction des ondulations du couple, ainsi que des pointes de courant et des pertes dans la machine.

La technique MLI vectorielle

Actuellement, les stratégies de commandes numériques sont appliqués dans les systèmes pour cette raison les techniques de modulation numériques sont également disponibles. La technique de modulation vectorielle est une technique numérique pour laquelle l'objectif est de générer une onde PWM (Pulse Width Modulation) de tension $V(t)$ à la sortie de l'onduleur dont la valeur moyenne sur chaque période de découpage T_d est égale à celle de la tension de référence $V_{ref}(t)$ sur cette période. Ceci est effectué à chaque période d'échantillonnage en sélectionnant les états appropriés des interrupteurs parmi la table d'excitation de l'onduleur à 2 niveaux et en déterminant la durée d'application de chacun des états. La sélection des états et la détermination des durées sont basées sur la transformation triphasée (abc) - biphasé (α, β) [7]. Cette méthode sera détaillé dans le chapitre de l'implémentation software.

Avantages de la MLI vectorielle

Les avantages de la MLI vectorielle vis-à-vis la MLI sinusoïdale sont [2] :

- La MLI vectorielle offre comme principal avantage de minimiser les harmoniques de tension et d'augmenter le rendement de l'onduleur de 15% par rapport à la MLI sinusoïdale.
- L'amplitude de la tension composée peut atteindre la valeur U_{dc} . Donc, une utilisation maximale de la tension d'entrée est possible dans la zone de fonctionnement linéaire.
- Seulement un vecteur de tension de référence est commandé pour générer trois ondes sinusoïdales.
- L'implantation de la MLI vectorielle en temps réel est plus facile.
- La flexibilité de sélectionner les états inactifs (état 0) et leurs distributions sur une période d'échantillonnage nous donnent deux degrés de liberté.
- Comme le vecteur tension de référence est une quantité à deux dimensions (V_α et V_β), il est faisable d'implanter la MLI vectorielle avec les techniques de commande vectorielle avancées des machines à courant alternatif.

1.6 Conclusion

Dans la première partie de ce chapitre, nous avons présenté brièvement le moteur asynchrone ainsi que ses deux types de commande qui sont la commande scalaire et la commande vectorielle. La deuxième partie a été consacrée pour exposer les différents types des onduleurs et leurs caractéristiques et les stratégies de commande. Tenu compte des avantages de MLI vectorielle par rapport au MLI scalaire, nous avons opté pour l'implémentation de cette technique sur la carte DSP. Le chapitre suivant est consacré pour expliquer le fonctionnement de cette carte.

Chapitre 2

Etude de la carte DSP TMS320F2812

2.1 Introduction

Ce chapitre est consacré à une introduction générale aux DSPs en premier lieu puis une description architecturale détaillée de DSP TMS320F2812 (on se réfère à F2812 dans la suite du mémoire) de Texas Instrument que nous avons utilisé dans notre projet, issu de la génération 28x et appartenant à la famille TMS320 conçu pour le contrôle numérique des signaux ainsi qu'à d'autres applications de commande (robotique) et de traitement de signaux.

Ce processeur de 176 broches travaille sur des mots de 32bit à virgule fixe, possède une horloge interne qui peut atteindre les 150MHz. Ce processeur est intégré dans un kit de développement eZdsp F2812 DSK construite par Spectrum Digital, intègre toutes les fonctionnalités du processeur F2812.

2.2 Généralités sur les DSPs

Le DSP (Digital Signal Processor) fait partie d'un type particulier de microprocesseurs. Ses fonctions spéciales ainsi que son architecture le rendent performant dans le domaine du traitement du signal et d'automatisme.

Dans la conception des systèmes de traitement numérique, contrairement aux processeurs classiques, tout système fondé autour d'un DSP bénéficie des avantages dérivants de ses particularités architecturale et de programmation ; dont on peut citer :

- Une unité de calcul puissante et rapide (augmentation des performances du système).
- Traitement de données en temps réel : cette capacité rend le DSP un élément parfait pour les applications qui ne tolère pas les retards.
- Flexibilité : si des erreurs apparaissent dans le fonctionnement du système, la re-conception ne sera qu'au niveau soft.
- Fiabilité : un système de traitement numérique notamment à base de DSP présente une immunité au bruit. Et donc un calcul précis.
- Implémentation d'algorithmes adaptatifs.
- Réduction du coût...

2.2.1 Historique

La première apparition des DSP était au début des années 70. Et cela pour des applications militaires (radars et télécommunications cryptées). Les DSP étaient destinés à réaliser des applications particulières bien précises, le plus souvent définitives. En 1978, Texas Instruments a introduit un DSP pour la synthèse de la voix pour des applications grand public. Ce premier pas a ouvert la concurrence, vers la conception des DSP polyvalents, entre plusieurs firmes (Texas Instruments, Analog Devices, Motorola, Lucent et autres). [11]

En 1982 Texas Instruments introduit le TMS32010, premier membre de ce qui allait devenir la famille de DSP la plus répandue, en grande partie grâce aux efforts déployés par ce constructeur pour fournir des outils de développement aussi bien matériels que logiciels.

A la fin des années 80, apparition de nombreux autres DSP comprenant des circuits de calcul plus rapides en virgule fixe comme DSPi d'Hitachi et le DSP16A de ATT et d'autres DSP virgule flottante comme la famille TMS320C3x de TI et les DSP96001 et DSP96002 de Motorola.

Aujourd'hui, toutes les grandes firmes de fabrication des dispositifs semi-conducteurs ont déjà leur produit de DSP ou sont encore en développement, réfléchissant leur confiance à l'énorme demande à la fin des années 80s, 90s et maintenant au 21^{ème} siècle, en parallèle avec le succès spectaculaire des microprocesseurs. [10]

2.2.2 Les spécificités d'un DSP

L'Architecture

L'architecture de Von Neumann : consiste à mettre les données et le programme dans la même zone mémoire, et donc utiliser un seul bus de données et d'adresses pour les données et le programme (les instructions).

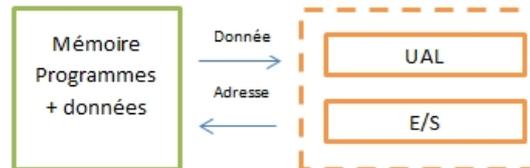


FIGURE 2.1 – L'architecture de Von Neumann

Il est bien clair qu'on ne peut pas lire une instruction ou une donnée qu'en un seul cycle d'horloge, donc pour l'exécution d'une instruction il faut souvent plusieurs cycles d'horloge. Ceci est un inconvénient majeur pour les applications dont le temps d'exécution est primordial. Un DSP dédié pour des applications à temps réel ne peut être servi par une telle structure. [11]

L'architecture de Harvard : caractérisée par la séparation entre mémoire données et mémoire des programmes. Cette structure offre la possibilité d'accéder aux données et aux instructions du programme en même temps (en un cycle d'horloge), grâce à deux bus distincts de données et deux bus distincts d'adresses, cela permet une grande souplesse pour l'enregistrement et l'utilisation des données.

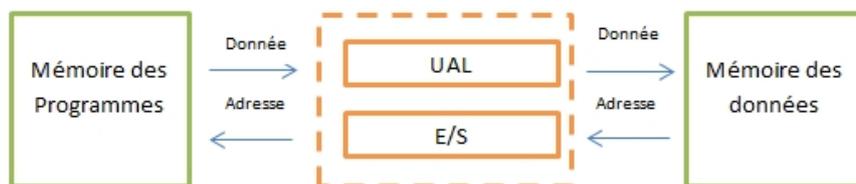


FIGURE 2.2 – L'architecture de Harvard

Ce type de structure, utilisé par les microcontrôleurs, permet d'avoir de grandes vitesses d'exécution. Cependant, l'augmentation du nombre de boîtiers mémoires et donc l'augmentation du nombre de bus rend le processeur très coûteux. [11]

L'architecture d'un DSP : A priori, pour augmenter le rapport performance/prix, les concepteurs de DSP ont utilisé une structure de Harvard dite 'structure de Harvard

modifiée'. Il s'agit de réaliser deux bus distincts de données et deux bus distincts d'adresse pour les accès en mémoires internes (architecture de Harvard), et pour les accès en mémoires externes deux bus l'un pour les données et l'autre pour les adresses (architecture de Von Neumann). Les transferts de données entre les bus internes et externes s'effectuent par multiplexage temporel. C'est le cas du TMS320F2812. [11]

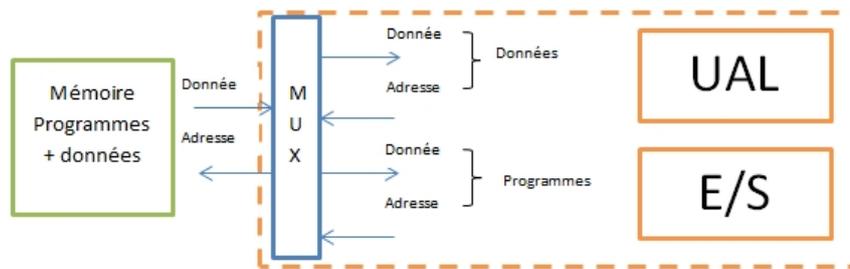


FIGURE 2.3 – L'architecture de Harvard modifié

Mémoire et mode d'adressage

Un DSP a la capacité de manipuler deux opérandes, ou même plus, via une seule instruction. Donc il doit réaliser au minimum un double accès mémoire en un cycle d'horloge. Pour les faire, en plus de la RAM (deux boîtiers de mémoires vives au minimum) et de la ROM, et dans certains DSP, une autre mémoire est utilisée pour stocker les instructions répétitives appelée 'Cache Memory', permettant de libérer les deux bus de données internes pour véhiculer les opérandes et de compenser le temps d'accès en ROM qui est critique par rapport à celui d'une RAM. [5][11]

Les modes d'adressages des données sont un point particulier des DSP. Un DSP peut posséder plusieurs unités logiques de génération d'adresse, travaillant en parallèle avec la logique du cœur du DSP.

On peut accéder aux données à partir d'une mémoire, des registres, et des instructions en utilisant les modes d'adressages suivants :

- Adressage registre : l'opérande est dans l'un des registres du CPU.
- Adressage direct : l'opérande est dans l'adresse mémoire spécifiée par l'instruction
- Adressage indirect : l'adresse de l'opérande est rangée dans un registre auxiliaire.
- Adressage immédiat : opérande encodé dans l'instruction.
- Adressage relatif au PC : utilisé lors des branchements, le déplacement vers la destination est généré par l'assembleur.

En plus de ces modes d'adressages standards, et tout dépend du type de DSP, d'autres modes d'adressages spéciaux existent, destinés à optimiser l'exécution de certains algorithmes répandus tels que les filtres, FFT, convolution, corrélation....etc. On peut citer :

- Adressage circulaire : un registre auxiliaire est incrémenté/décroché en respectant les limites du buffer circulaire. Selon ces limites deux versions existent : (@départ, @arrivée) et (@départ, longueur).
- Adressage par inversion de l'ordre des bits : utilisé dans le calcul de la FFT.

L'unité MAC

Le DSP a été inventé plusieurs années après le processeur standard. Le besoin d'un processeur numérique de signal est dû aux nouvelles spécificités des algorithmes de traitement de signal numérique, ces algorithmes contiennent toujours des MAC (multiplication and accumulation) [10], cette unité est représentée dans la figure 2.4.

Cette unité de calcul fait la différence essentielle du DSP avec tous les autres processeurs. Elle est optimisée de manière à réaliser une multiplication et une addition en un cycle d'horloge [11]. Effectuer une opération MAC en un seul cycle n'est malgré tout pas satisfaisant si le cycle d'horloge est trop long . Le principal objectif d'évolution des DSP a toujours été d'améliorer le temps de calcul d'un MAC [14].

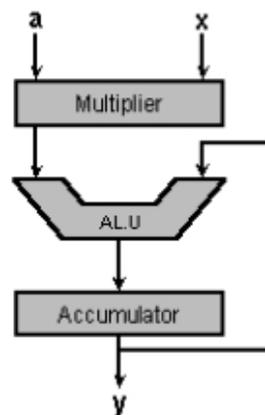


FIGURE 2.4 – Représentation simplifiée de l'unité MAC

Le Pipeline

Afin de gagner du temps lors de l'exécution de séries d'instructions, il est donc nécessaire d'optimiser ces différentes étapes en les parallélisant ou en les "pipeline". Le principe retenu est celui utilisé dans les usines de production et qui consiste à découper le travail en tâches élémentaires [14]. La figure 2.5 montre la structure du pipeline.

CYCLE	Fetch	Decode	Read	Execute
m-3	w	-	-	-
m-2	x	w	-	-
m-1	y	x	w	-
M	z	y	x	w
m+1	-	z	y	x
m+2	-	-	z	y
m+3	-	-	-	z

FIGURE 2.5 – La structure du Pipeline

Arithmétique d'un DSP

Il existe deux types de DSP, DSP à virgule fixe et DSP à virgule flottante [14] :

- Les DSP à virgule fixe : les données sont représentées 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 sur 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.
- Les 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 : $n = mantisse \times 2^{exposant}$. 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).

2.2.3 Domaines d'application

Le traitement numérique du signal est une méthode, ou encore mieux, une science qui traite les signaux du monde réel. Chose qui a rendu le domaine d'application des processeurs de traitement numérique du signal très vaste. Les applications courantes des DSP sont : [11]

- En général : filtres numériques, filtres adaptatifs, corrélation, FFT, DCT, PLL et génération de signaux. . .
- Instrumentation : analyse spectrale, suppression de bruits, séismographe. . .
- Médical : monitoring, imagerie médicale. . .
- Contrôle : asservissement, robotique, vision artificielle. . .
- Télécommunications : modems, multiplexeurs, FAX, annulation d'écho, répéteurs de ligne, égaliseurs auto adaptatifs, cryptage de données. . .
- Militaire : radar, sonar, guidage de missile, calculateurs de trajectoires, communication et sécurité, navigation. . .
- Grand public : automobile, électroménager, jeux, synthétiseurs musicaux, navigation, téléphone. . .

2.2.4 La famille de DSP de Texas Instruments TMS320

En 1982, Texas Instruments a lancé son premier DSP : le TMS3210. Depuis, Texas instruments s'est imposé comme le principal fabricant des processeurs de traitement de signal. La famille de DSP TMS320 est organisée aujourd'hui en cinq générations principales qui correspondent chacune à une classe de performances et d'applications. Elles sont représentées par les sigles suivants : [11] • TMS320C1X • TMS320C2X • TMS320C3X • TMS320C4X • TMS320C5X • TMS320C6X • TMS320C8X

2.3 Le DSP TMS320F2812

2.3.1 Descriptions et caractéristiques générales :

Le DSP F2812, membre de la famille TMS320C28x, a une très haute intégration des composants et de fonctionnalité, plus une performance très remarquable, il offre des solutions adéquates aux applications de contrôle. Il est le premier DSP 32 bits et 150MIPS avec une mémoire flash on-chip et un périphérique analogique de très haute précision on-chip, avec une impressionnante 32x32bit MAC, durant seulement un cycle d'horloge 6.67ns. En plus leur architecture est spécialement optimisée pour C/C++. En outre, ce DSP permet aux utilisateurs de développer leur code en virgule flottante virtuelle via la technique IQ math. [10]

Le F2812 est constitué d'un noyau dit CPU (core processor) associé à une mémoire et des périphériques. La communication entre eux se fait à travers des bus (bus de périphérique pour la communication noyau/périphérique et bus de mémoire pour la

communication noyau/ mémoire) [15], comme le montre la figure 2.6 du schéma simplifié de F2812. Il est construit selon l'architecture Harvard modifiée (voir la section 2.2.2.1).

Les principales caractéristiques de ce DSP sont résumées dans le tableau suivant :

Génération	TMS320F281x Controllers
Technologie	High-Performance Static CMOS Technology
Horloge	150 MHz (6.67-ns Cycle)
Niveau du signal	1.8-V Core @135 MHz
	1.9-V Core @150 MHz
	3.3-V Entrées/sorties
	0-3 V convertisseur A/N
Mémoire interne (on-chip)	Single-Access RAM (SARAM). 18K Words mémoire Flash de 128k words
Gestionnaires des évènements	EVA et EVB
General-Purpose (GP) Timers	4
Comparateurs (CMP)/PWM	16
Captures (CAP)/canaux QEP	6/2
Mémoire externe	Jusqu'à 1M words
Signaux MLI (PWM)	16 canaux
	possibilité d'une MLI vectorielle SVPWM
Convertisseur A/N	16 canaux
	Résolution de 12-bits
	Temps de conversion 80 ns (12.5 MSPS)
Broches d'entrée/sortie	Jusqu'à 56 broches

TABLE 2.1 – Les principales caractéristiques de F2812

La figure 2.6 illustre un schéma simplifié de DSP TMS320F2812.

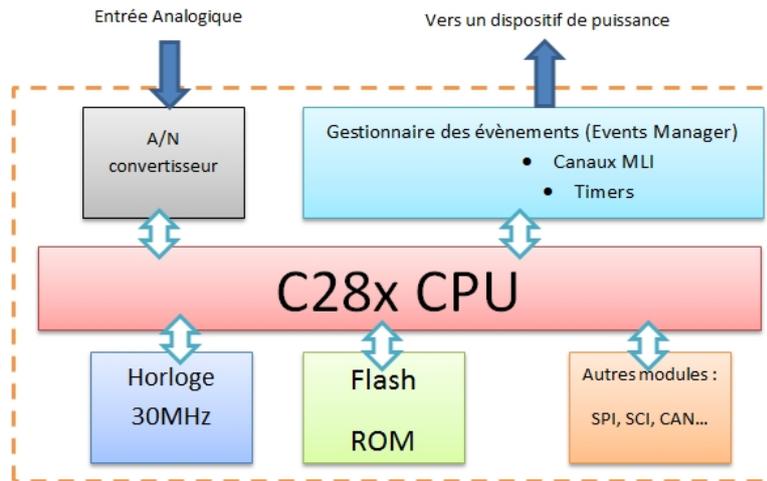


FIGURE 2.6 – Schéma bloc simplifié de TMS320F2812

Comme le cerveau de F2812 est un C28x core, le F2812 contient plusieurs périphériques orientés contrôle. Ces périphériques peuvent faire pratiquement n'importe quel contrôle numérique possible exigé, leur application s'étend de la conversion analogique-numérique à la génération d'une MLI. Les périphériques de communication permettent la possibilité d'une communication avec un périphérique externe, ordinateur ou un autre DSP [6]. Comme le montre la figure 2.7, plusieurs périphériques sont disponibles :

- Système de contrôle et d'Interruption, (System Control and Interrupts) ;
- Interface externe (XINTF), (External Interface) ;
- Contrôleur CAN amélioré (Enhanced Controller Area Network) (eCAN) ;
- Port série multi canal tamponné (McBSP), (Multichannel Buffered Serial Port) ;
- Périphérique d'interface série (SPI), (Serial Peripheral Interface) ;
- Interface de communication série (SCI), (Serial Communication Interface) ;
- Rom de démarrage (Boot Rom) ;
- Convertisseur analogique numérique (ADC) ;
- Gestionnaire d'événement (EV), Event Manager.

La figure 2.7 illustre les différents blocs et périphériques reliés au cœur de DSP C28x CPU.

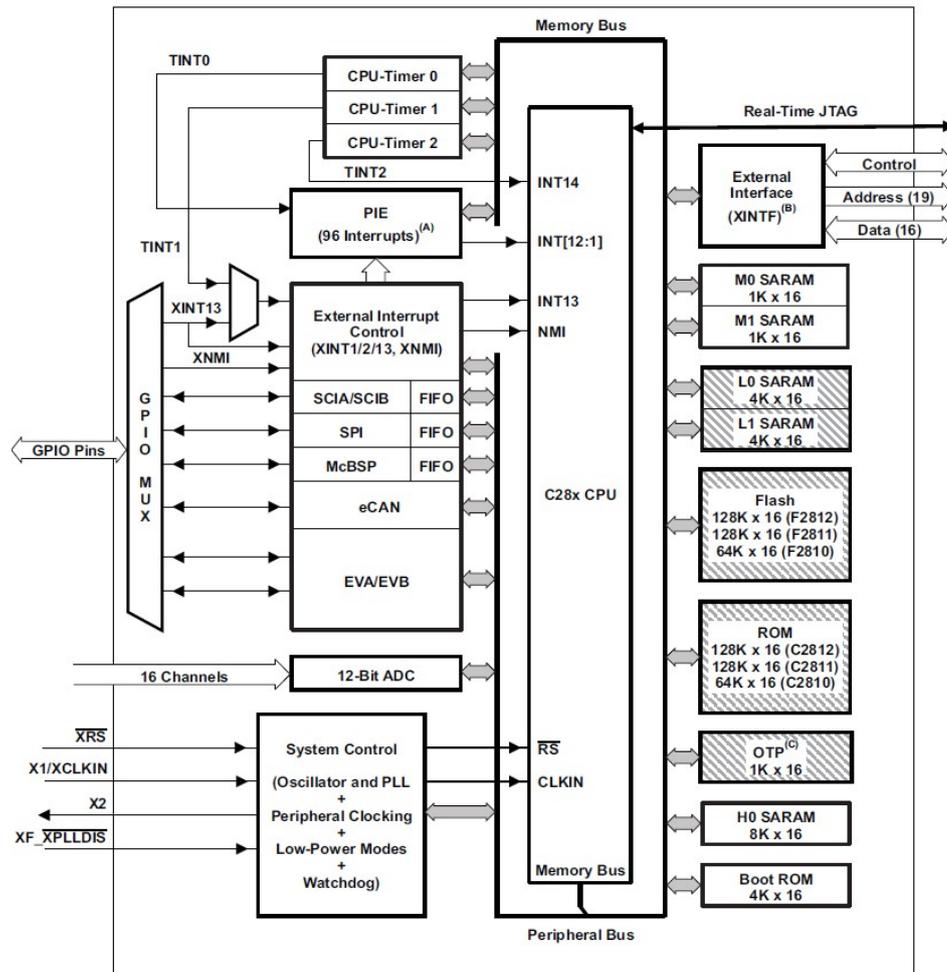


FIGURE 2.7 – Schéma bloc détaillé de TMS320F2812 [16]

2.3.2 Le CPU (core)

Le cœur de F2812 est le C28x DSP core, ce noyau est de 32 bits à virgule fixe, on peut assimiler ce core à un microprocesseur dans un microordinateur personnel. [10]

Le C28x a un compilateur C/C++ très efficace, il permet à l'utilisateur de développer son software de contrôle en langage de haut niveau en C/C++. Le C28x est très efficace dans les opérations mathématiques et dans les tâches des systèmes de contrôle typiquement réalisées par des microcontrôleurs. La capacité de faire des MAC 32 x 32-bit permet au C28x de manipuler des opérations numériques avec une grande résolution. [15][16]

Le CPU est principalement caractérisé par un pipeline de 8 phases. Une unité arithmétique et logique (ALU : arithmetic and logic unit) sert à traiter les opérations lo-

giques et arithmétiques. Address register arithmetic unit (ARAU) sert à générer les adresses pour la mémoire de données ou pour incrémenter et décrémenter les pointeurs. Le Barrel shifter sert à faire un décalage à gauche ou à droite de 16 bit. Un multiplieur permet de réaliser des multiplications de 32x32 bits avec un résultat sur 64 bits.

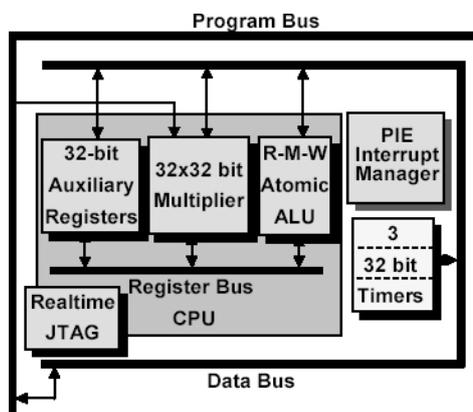


FIGURE 2.8 – Schéma bloc de C28x

2.3.3 Memory-Map

Dans le F2812, l'espace mémoire est divisé entre les programmes et les données. Il y a plusieurs types de mémoire disponible qui peuvent être utilisées à la fois pour les programmes et les données comme la mémoire Flash, SARAM, expanded SARAM et Boot ROM. La mémoire est composée par des mot-mémoire de 16 bits. [18]

Le F2812 a un accès mémoire aux deux types de mémoire, on-chip et off-chip. Il utilise 32-bit pour les adresses des données et 22-bit pour les adresses des programmes, ce qui permettra au F2812 d'adresser au totale 4G mot-mémoire pour les données et 4M mot-mémoire pour les programmes.

Tous les registres des périphériques sont rassemblés dans les "Peripheral Frames" (cadres des périphériques) : PFO, PF1 et PF2. Ces cadres sont tout simplement des mémoires de données adressées.

Le cadre périphérique PF0 inclut les ensembles des registres pour contrôler la vitesse interne de la mémoire FLASH, aussi bien que l'accès au calendrier SARAM interne. SARAM signifie "Single Access RAM", cela veut dire que nous pouvons faire un accès

à ce type de mémoire par cycle d'horloge. Flash interne est la mémoire non-volatile, le plus souvent utilisée pour le stockage du code et de données qui doivent être présente au moment du démarrage. Le cadre périphérique PF1 contient la plupart des registres des unités périphériques, alors que le cadre PF2 est réservé pour le bloc du registre CAN. [22]

Certaines des zones mémoires sont protégées par un mot de passe avec "Code Security Modele". Il s'agit d'une fonctionnalité permettant d'empêcher le retour de l'ingénierie. Une fois le mot de passe programmé, tout accès aux zones sécurisées est accordé seulement si le bon mot de passe est entré dans une zone spéciale de PF0.

La memory-map de la figure 2.9 montre les différents blocs de mémoire disponibles des programmes et des données.

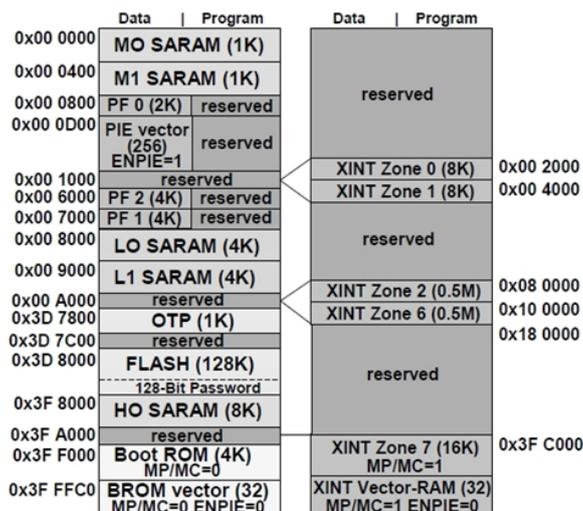
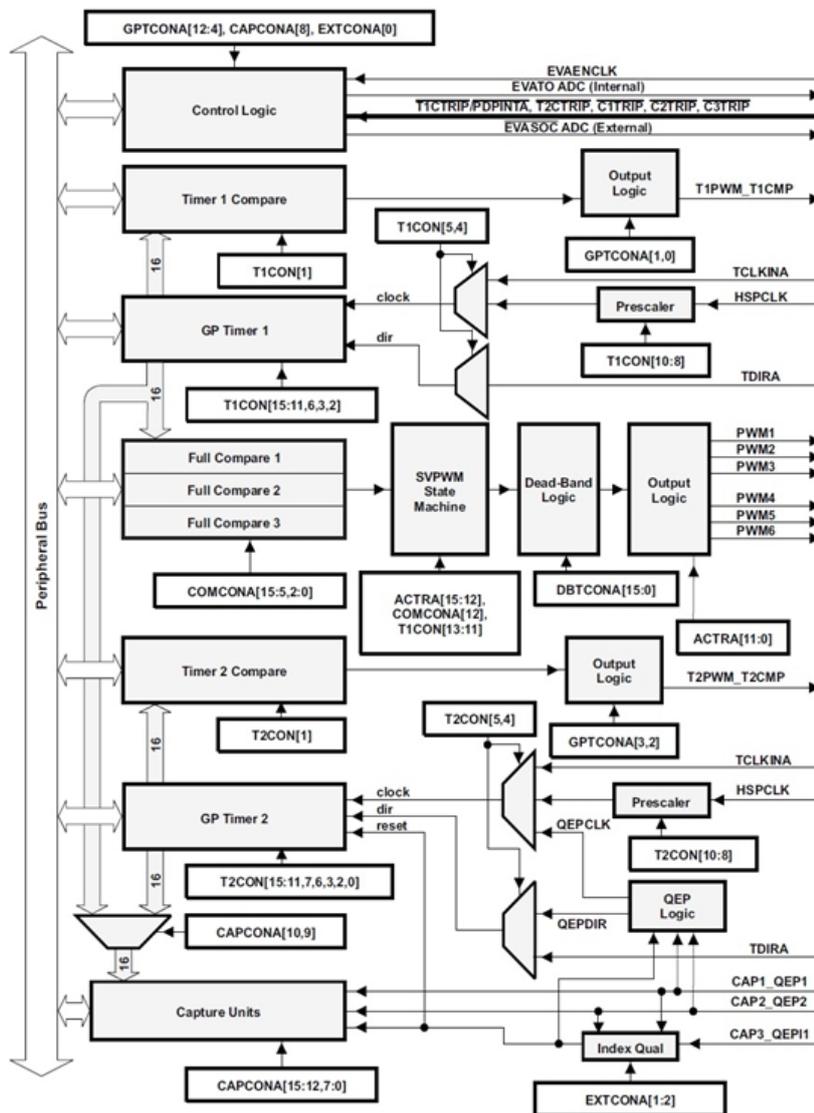


FIGURE 2.9 – Memory Map du TMS320F2812

2.3.4 Les périphériques

L'Event Manager

Il y a deux Event Managers dans F2812, le EVA et EVB. L'Event Manager est le plus important périphérique dans le contrôle numérique des Moteurs. Il a les fonctions nécessaires pour faire un contrôle des dispositifs électromécaniques. Chaque Event Manager est composé des " blocs " fonctionnels y compris les Timers, comparateurs, une unité pour les capteurs, circuit logique pour MLI, le circuit quadrature-encoder-pulse (QEP), les interruptions. Voir la figure 2.10.



A. The EVB module is similar to the EVA module.

FIGURE 2.10 – Schéma blocs de Event Manager A

General purpose timers (GP) : Chaque EV contient deux GP (Timer 1, 2 pour le EVA et Timer 3, 4 pour le EVB) chacun d'eux contient :

- un 16-bit timer, up-/down-counter, TxCNT, en mode Ecrire/Lire
- un 16-bit timer-compare register, TxCMPR, en mode Ecrire/Lire
- un 16-bit timer-period register, TxPR, en mode Ecrire/Lire
- un 16-bit timer-control register, TxCON, en mode Ecrire/Lire
- entrée sélectionnable d'horloge interne/externe.
- un prescaler programmable pour une entrée d'horloge interne/externe.

- Control and interrupt logic, pour quatre interruptions masquables : underflow, overflow, timer compare, and period interrupts
- input pin pour sélectionner la direction (TDIRx)

On a la possibilité de sélectionner l'usage d'une horloge interne ou externe du DSP pour base de temps et un choix du coefficient à choisir pour l'horloge (possibilité de choisir différentes fréquences allant jusqu'à 150MHz). Les GP Timers peuvent être utilisés indépendamment ou synchronisés les uns avec les autres. Le registre compare associé à chaque GP Timer peut être utilisé pour générer des signaux PWM. GP Timer peut être configuré aussi comme base de temps pour QEP et les capture units. [15][21]

Unité de Full-Compare : Il y a trois unités de full compare sur chaque gestionnaire d'événement. Ces unités de full compare utilisent GP Timer 1 comme base de temps pour générer six sorties PWM et la génération de formes d'ondes en utilisant le temps - mort programmable. L'état de chacun des six sorties est configuré indépendamment.

Le Temps Mort Programmable(DeadBand) : Le circuit générateur de Deadband comporte trois compteurs de 4-bits et un registre compare de 16-bits. La valeur désirée du Deadband peut être programmée dans le registre compare et le résultat sera transféré dans les full-compares. La génération du Deadband peut être activée, désactivée pour chaque sortie individuellement. Le circuit générateur du Deadband produit deux sorties (avec ou sans zone morte) pour chaque sortie d'unité de compare. Les états de sortie du générateur du Deadband sont configurables et variables selon les besoins par le biais du tampon double ACTRx.

Unité pour les capteurs (capture unit) : Le capture unit décode les signaux acquis à travers les pins CAP1, CAP2, CAP3 pour EVA et CAP4, CAP5, CAP6 pour EVB provenant de capteurs externes (à effet Hall par exemple). Les entrées sont stockées dans des piles FIFO à deux niveaux. Le contrôle des CAP se fait à travers les registres CAPCON. L'unité pour les capteurs a les caractéristiques suivantes [21] :

- un 16-bit capture control register, CAPCONx (R/W).
- un 16-bit capture FIFO status register, CAPFIFOx.
- Sélection de GP timer 1/2 (pour EVA) or 3/4 (pour EVB) comme la base de temps.
- Trois 16-bit 2-level-deep FIFO stacks, une pour chaque capture unit.

- trois pins d’entrées pour les capteurs X (CAP1/2/3 for EVA, CAP4/5/6 for EVB).
- Une entrée pour chaque capture unit (toutes les entrées sont synchronisées avec l’horloge de CPU. Pour que la transition soit capturée, l’entrée doit atteindre son niveau de courant pendant le front montant d’horloge. Les entrées CAP1/2/3 et CAP 4/5/6 peuvent être utilisées comme des entrées QEP pour le circuit de QEP.
- L’utilisateur peut régler la détection des transitions (rising edge, falling edge).
- Trois flags d’interruption masquable, un pour chaque capture unit.

Quadrature encoder pulse : Le QEP est une unité qui permet d’utiliser des encodeurs tel qu’un capteur optique placé sur l’arbre d’un moteur. Cette unité possède deux entrées sur chaque EV (QEP1, 2 et QEP3, 4).

Peripheral Interrupt Expansion (PIE)

Le PIE est un bloc qui multiplexe plusieurs sources interruptions en quelques entrées d’interruptions. Le PIE supporte 96 interruptions qui sont regroupées en 8 blocs, et chaque groupe est introduit dans l’une des 12 lignes de CPU-Interrupt (INT1 à INT12). Chacune des 96 interruptions est soutenue par son propre vecteur stocké dans une mémoire vive dédiée qui peut être écrasée par l’utilisateur. Le vecteur d’interruption est automatiquement récupéré par le CPU pour le service de routine. Par conséquent, le CPU peut répondre rapidement à des événements d’interruption. La hiérarchisation des interruptions est contrôlée par le hardware ou le software. [15]

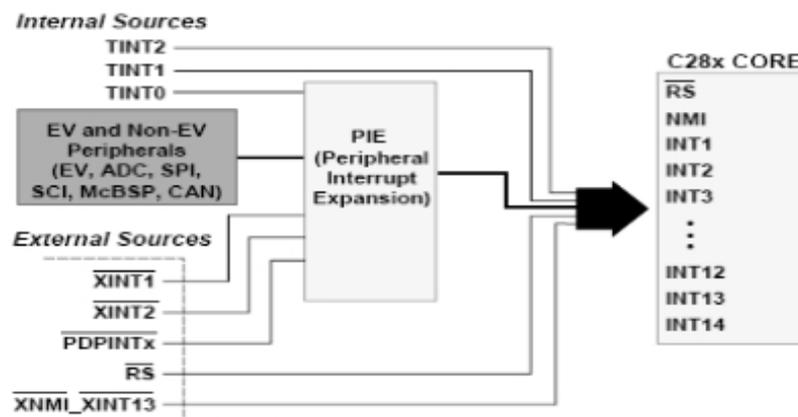


FIGURE 2.11 – Les sources d’interruptions

La figure 2.12 montre les 16 lignes d'interruptions vers le cœur processeur, 14 d'entre elles sont masquables (l'utilisateur peut les activer ou les désactiver à travers des registres dédiés) et 2 non masquable. [22]

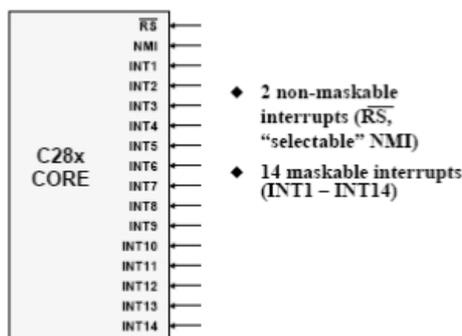


FIGURE 2.12 – Les vecteurs d'interruptions vers le CPU

Watchdog Timer(WD)

Le chien de garde " Watchdog Timer " est un périphérique qui contrôle les opérations hardware et software et assurer un reset du system lorsque son compteur interne est débordé (overflow). Le WD (lorsque il est activé) va compter jusque un certaine temps, si nécessaire l'utilisateur doit initialiser le WD périodiquement, donc éviter un reset non voulu du système. Si pour une raison le CPU est bloqué, le WD va générer un reset de system, par exemple si le programme entre dans une boucle infinie le WD sera débordé et le DSP sera initialisé et le programme est rebranché vers leur début. La plus part des erreurs qui se produisent temporellement et qui causent un blocage ou un mal fonctionnement de CPU peuvent être éliminées par la fonction de WD. Le WD augmente la sûreté de CPU. [6]

Un module ADC (Analog to Digital Converter)

L'ADC contient 16 canaux de 12 bits, configurés comme étant deux modules de 8 canaux. Les deux modules de 8 canaux peuvent être configurés en cascade pour former un module de 16 canaux. Il contient deux unités échantillonnage et arrêt (Sample-and-hold) pour l'échantillonnage simultané.

L'ADC a plusieurs applications importantes, elles commencent par le contrôle des performances des moteurs par le monitoring de feedback, comme la fonction de current sensing juste avec un shunt sans l'utilisation d'un capteur de courant très chère.

Un module eCAN (enhanced Controller Area Network)

Ceci est la version améliorée du périphérique CAN. Il supporte 32 boîtes aux lettres et il est compatible avec CAN2.0B.

Un module McBSP (Multichannel Buffered Serial Port)

Le McBSP est utilisé pour se connecter à des lignes E1 / T1, il est utilisé généralement pour les applications modem de haute résolution. Les McBSP reçoivent et transmettent les informations par le biais des registres de 16-niveau FIFO. Cela réduit considérablement les frais pour le service de ce périphérique.

Deux modules SCI (Serial Communication Interface A, B)

Le SCI est un port série asynchrone. Sur le F2812, le port supporte un niveau de 16, il reçoit et transmet les informations sur des registres FIFO.

Un module SPI (Serial Peripheral Interface)

Le SPI à une vitesse élevée, les ports séries synchrones E/S de longueur 16-bit permet d'échanger des données (16-bit) à l'intérieur comme à l'extérieur de l'appareil. Le SPI est utilisé pour les communications entre le DSP et les périphériques externes ou à un autre processeur.

Entrées et sorties à usage général GPIO (General-purpose Input/Output)

La plupart des signaux sont multiplexés pour un usage général E/O (GPIO). Cela permet d'utiliser un pin comme GPIO si le signal périphérique ou la fonction n'est pas utilisée. A l'initialisation, tous les pins GPIO sont configurés comme des entrées, on peut ensuite programmer individuellement chaque pin GPIO comme mode signal. [22]

2.3.5 L'outil de développement matériel (eZdsp™ F2812 DSK)

L'eZdsp F2812 DSK est une carte autonome conçue pour le développement et l'exécution des programmes pour le processeur TMS320F2812. C'est une carte électronique de circuit imprimé multicouche, de dimension 13.4x7.6 cm et alimentée par 5Volts fournis par une alimentation externe.

L'eZdsp F2812 est embarqué avec un DSP TMS320F2812. Il permet de vérifier le code F2812 complètement et rapidement. Deux connecteurs d'expansion sont fournis pour n'importe quel circuit d'évaluation nécessaire non fournis sur la configuration embarquée. [10]

Les caractéristiques principales du DSK eZdsp F2812 sont :[17]

- DSP TMS320F2812 fonctionne à 150MIPS.
- Horloge de 30MHz.
- 18K words on-chip RAM.
- 128K words Flash ROM.
- 64K words off-chip de mémoire SRAM.
- Deux expansions de connexion (analogique, I/O).
- IEEE 1149.Q JTAG emulator.
- 5Volts fournis par un adaptateur AC.

La figure 2.13 montre un schéma fonctionnel de la configuration de base pour l'eZdsp F2812. Les principales interfaces de l'eZdsp sont : l'interface de JTAG, l'interface d'expansion, l'interface analogique et l'interface des entrées/sorties (I/O).

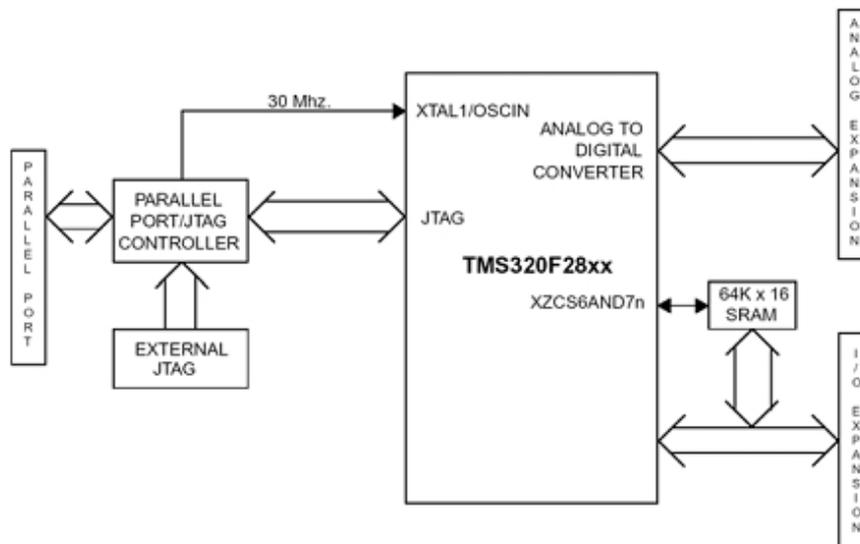


FIGURE 2.13 – schéma synoptique du module DSK eZdsp F2812

L'eZdsp inclut les mémoires on-chip suivantes :

- 128K*16 Flash.
- 2 blocs de 4K*16 Single Access RAM (SARAM).
- 1 bloc de 8K*16 SARAM.

– 2 blocs de 1K*16 SARAM.

En plus, une mémoire de 64K*16 off-chip SRAM est insérée. Le processeur sur l'eZdsp peut être configuré en mode boot-loader ou en mode non- boot-loader. L'eZdsp a cinq connecteurs. Le Pin 1 de chaque connecteur est identifié par garniture carrée de soudure.

La figure 2.14 illustre une vue supérieure de la carte SDK eZdsp F2812.



FIGURE 2.14 – Vue supérieure de la carte SDK eZdsp F2812

La fonction de chaque connecteur est montrée dans la table ci-dessous :

Connecteur	Fonction
P1	Interface JTAG
P2	Expansion
P3	Interface parallèle de contrôleur de port/JTAG
P4/P7/P8	Interface des I/O
P5/P9	Interface analogique
P6	Connecteur d'alimentation

TABLE 2.2 – Connecteurs de l'eZdsp F2812

Dans ce travail, on s'intéresse aux connecteurs des entrées/sorties : P4/P7/P8 et surtout le P8 et le connecteur d'interface analogique.

Le tableau suivant montre la définition des pins de connecteur P8.

P8 Pin#	Signale	P8 Pin#	Signale
1	+3.3V/+5V/NC	2	+3.3V/+5V/NC
3	SCITXDA	4	SCIRXDA
5	XINT1n/XBIO _n	6	CAP1/QEP1
7	CAP2/QEP2	8	CAP3/QEP1
9	PWM1	10	PWM2
11	PWM3	12	PWM4
13	PWM5	14	PWM6
15	T1PWM/T1CMP	16	T2PWM/T2CMP
17	TDIRA	18	TCLKINA
19	GND	20	GND
21	No connect	22	XINT1N/XBIO _n
23	SPISIMOA	24	SPISOMIA
25	SPICLKA	26	SPISTEA
27	CANTXA	28	
29	XCLKOUT	30	PWM7
31	PWM8	32	PWM9
33	PWM10	34	PWM11
35	PWM12	36	CAP4/QEP3
37	T1CTRIIP/PDPINTA _n	38	T3CTRIIP/PDPINTB _n
39	GND	40	GND

TABLE 2.3 – Les pins du connecteur P8.

2.4 Conclusion

A travers ce chapitre, nous avons eu une introduction générale aux DSPs puis une description architecturale de DSP TMS320F2812, d'après cette description nous avons constaté que ce DSP est une solution adéquate pour les applications de commande numérique des moteurs, ce qui convient parfaitement à notre projet.

Chapitre 3

Implémentation Hardware

3.1 Introduction

Ce chapitre concerne la conception et la réalisation d'un contrôleur embarqué de vitesse pour un moteur asynchrone triphasé, à base d'un onduleur de tension piloté par une commande MLI vectorielle générée par un DSP.

La première étape de notre réalisation concerne la carte de puissance. Cette carte est basée sur un module de haute intégration, ce qui nous permet d'obtenir un circuit de puissance très simple qui nécessite un minimum de composants. Notre choix s'est porté rapidement sur le module IRAM136-0461G qui est une solution adéquate qui respecte plusieurs contraintes imposées par un système embarqué (réduction de volume, cout, fiabilité...). Nous allons expliquer le fonctionnement de ce module et la façon de choisir les composants de son montage.

La deuxième étape est la réalisation d'une carte d'interface entre la carte de puissance et le DSP, celle-ci est indispensable pour le bon fonctionnement en toute sécurité de notre système.

3.2 Description générale du banc d'essai

La plateforme expérimentale est constituée d'un moteur asynchrone triphasé connecté à la carte de puissance qui comporte un onduleur de tension à deux niveaux (300W, 4A), une carte d'interface qui contient l'étage d'isolation, une carte de développement DSP, et des appareils de mesure et un PC. La figure 3.1 montre une photo du banc d'essai.

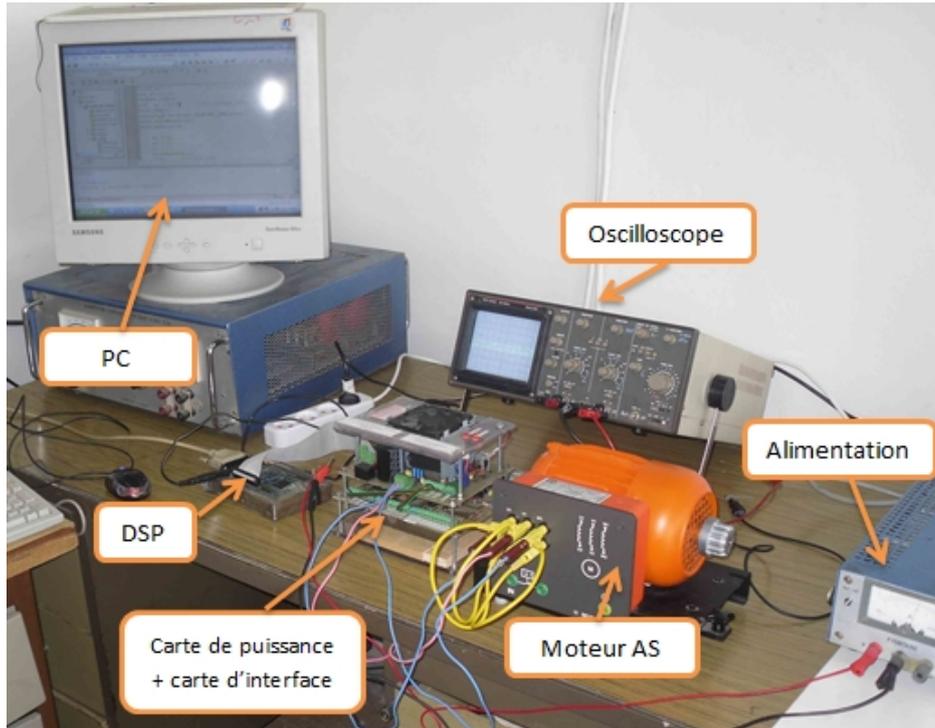


FIGURE 3.1 – Photo du banc d'essai

La figure 3.2 illustre le schéma fonctionnel du dispositif expérimental.

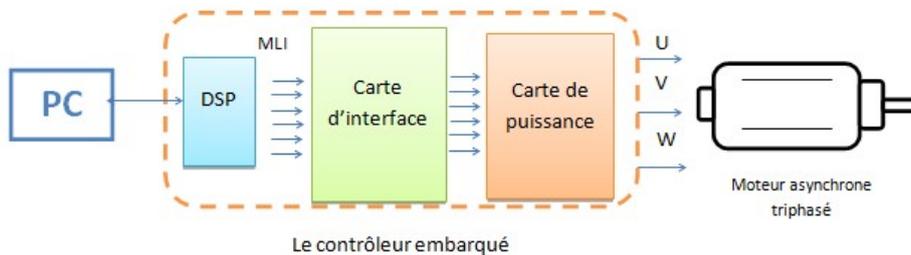


FIGURE 3.2 – Schéma synoptique du banc d'essai

Comme il est illustré dans le schéma synoptique de la figure 3.2, notre implémentation Hardware se décompose en trois parties :

- Un DSP F2812 pour générer les signaux MLI.
- Une carte d'interface pour protéger le DSP.
- Une carte de puissance pour générer les tensions (U, V, W), cette carte sera branchée au moteur asynchrone triphasé.

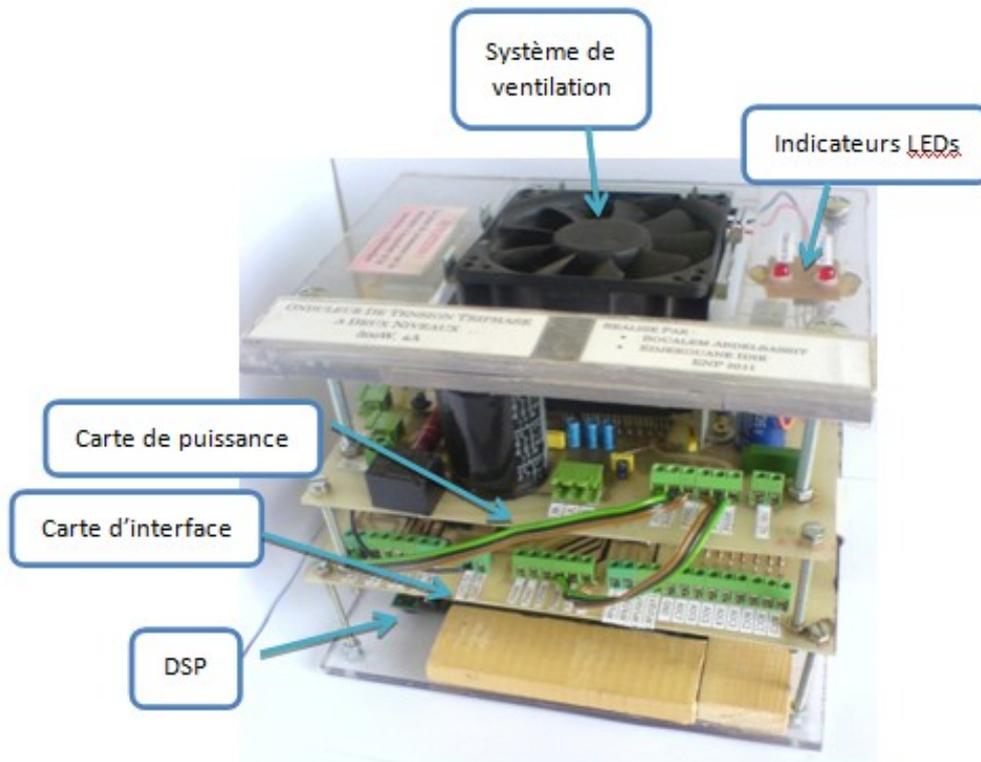


FIGURE 3.3 – Le contrôleur embarqué

3.3 Le module IRAM136-0461G

Le composant IRAM136-0461G développé par l'entreprise américaine International Rectifier, est un module de puissance avec une technologie d'intégration très avancée. Il est destiné et optimisé pour des applications de contrôle des moteurs asynchrones triphasés de puissance inférieure à 300W.

Ce module offre l'intégration la plus élevée disponible dans le marché, il comporte un pont de redressement, un onduleur triphasé complet avec les diodes de bootstrap

et le driver IC, une résistance shunt pour mesurer le courant absorbé et un capteur de température.

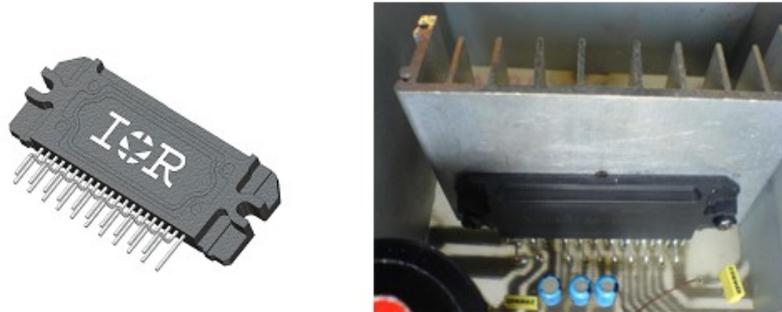


FIGURE 3.4 – Le module IRAM136-0461G

3.3.1 Caractéristiques du module IRAM136-0461G

Les caractéristiques du module IRAM136-0461G sont : [19]

- Pont de redressement incorporé.
- Résistance Shunt intégrée.
- Gates Drivers et diodes bootstrap intégrées.
- Température monitoring.
- Entrées logiques avec Schmitt-triggered.
- Forte immunité contre les bruits.
- Puissance du moteur 0.1–0.3kW / jusqu'à 253V, 50/60Hz.
- Isolation 2000VRMS /1min.
- Retard de propagation égale pour tous les canaux.
- La fonction under-voltage lockout.
- La protection over-temperature/over-current.

Le schéma de la figure 3.5 représente la structure interne du module utilisé.

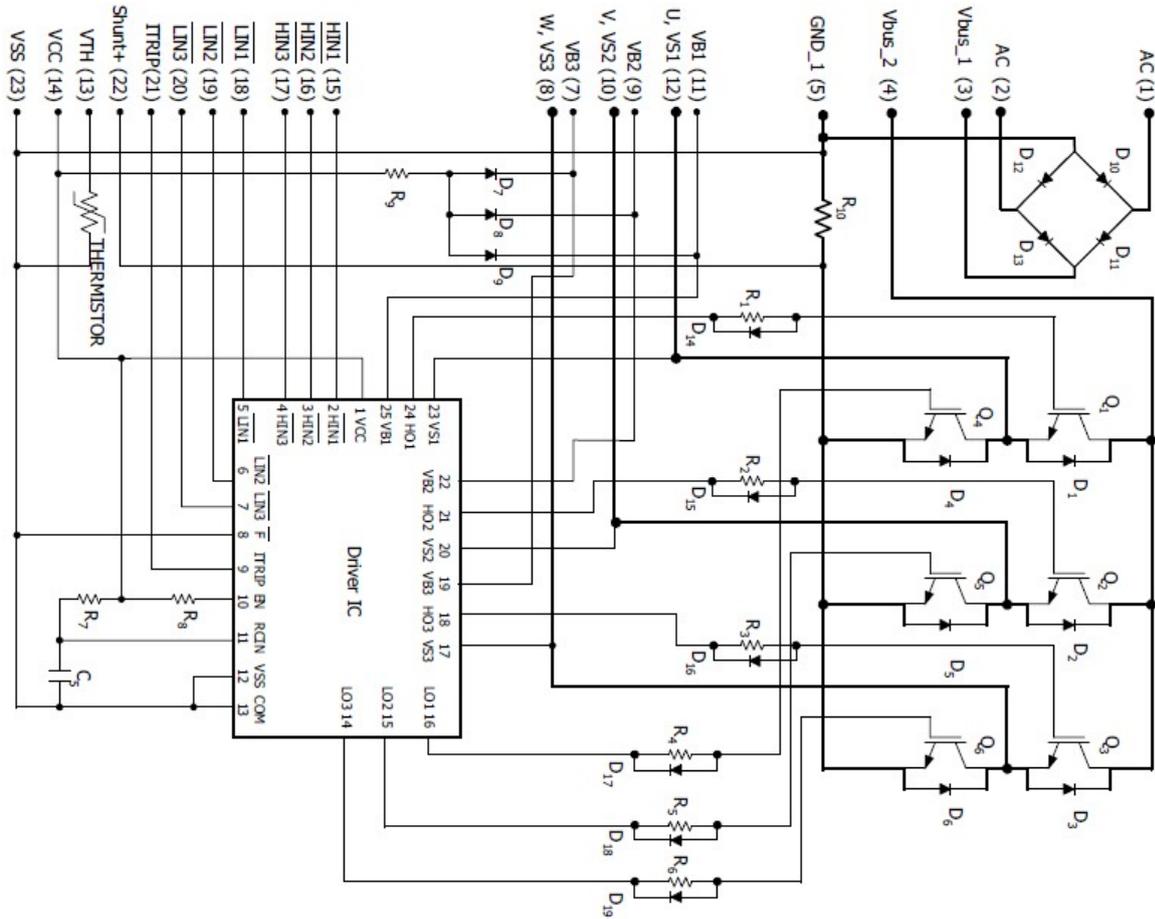


FIGURE 3.5 – Schéma interne du module IRAM136-0461G

3.4 Conception et réalisation de la carte de puissance

La conception de la carte de puissance est basée sur le module IRAM136-0461G. A cause de la haute intégrité de module, le schéma de carte de puissance devient très simple, il suffit d'ajouter un condensateur de filtrage de bus continu, condensateur de bootstrap et une alimentation stabilisée de 15V pour alimenter le driver IC.

Comme il est illustré dans la figure 3.6, la carte de puissance comporte 6 entrées MLI. Les signaux MLI sont numérotés 1,3 et 5 pour la commande des IGBT de high-side et 2,4 et 6 pour la commande des IGBT de low-side. Cette carte est alimentée directement par la tension de secteur. Le schéma électrique de la carte de puissance est illustré dans la figure 3.7.

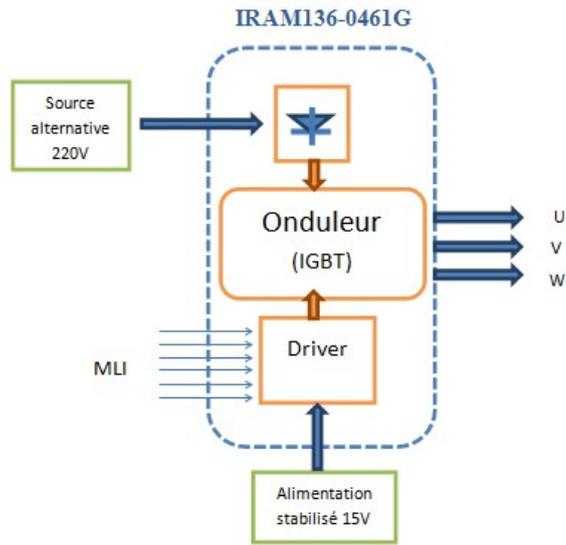


FIGURE 3.6 – Schéma synoptique de carte de puissance

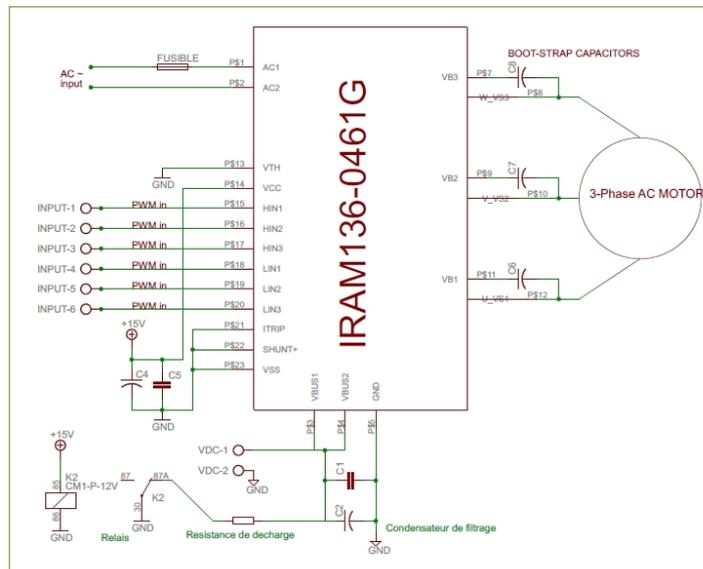


FIGURE 3.7 – Schéma électrique de la carte de puissance

le tableau ci-dessous reprente la liste des valeurs des composants utilisés dans la carte de puissance :

C1 (filtrage de bus continue)	100nF
C2	470 μ F
C6,C7,C8 (Capacités de Bootstrap)	10 μ F
C4	100 μ F
C5 (filtrage de l'alimentation 15V)	100nF
Résistance de décharge	1k Ω
Relais 4 poles	12V

TABLE 3.1 – Liste des composants de la carte de puissance

La figure 3.8 montre une vue supérieure de la carte de puissance.

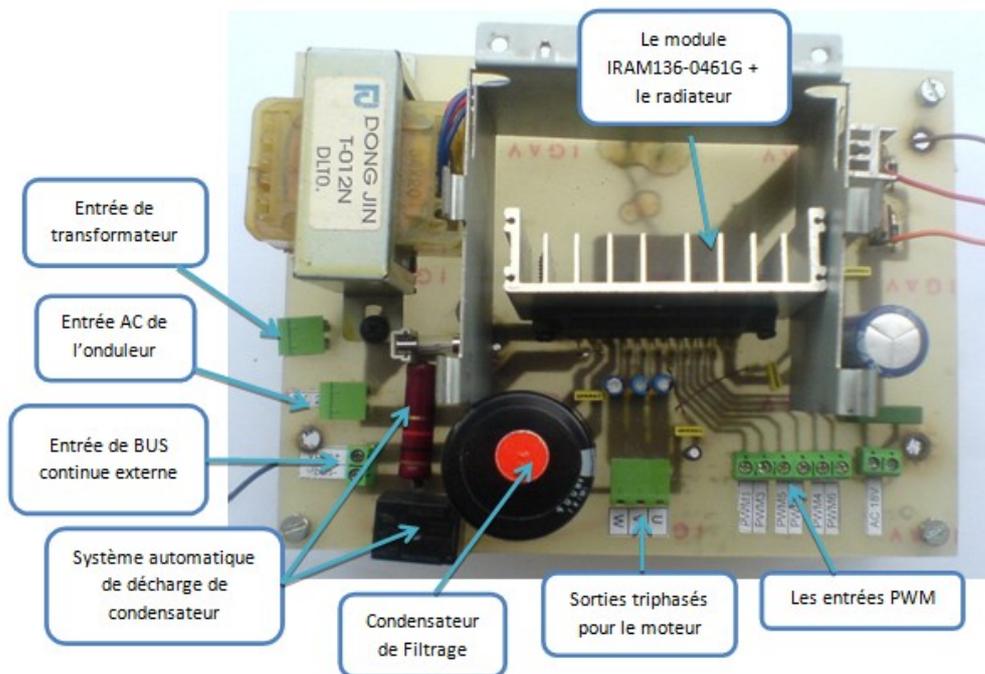


FIGURE 3.8 – La carte de puissance

Pendant la réalisation de la carte de puissance, nous avons appris à prendre en considération plusieurs précautions avant n'importe quel essai. Le non respect de ces précautions nous a coûté plusieurs modules :

1. Mettre les condensateurs de bootstrap le plus proche possible du module pour éviter des effets des oscillations qui peuvent avoir lieu à cause des longues pistes de connexion dans le circuit imprimé.

2. à cause de la forte capacité de condensateur de filtrage, et la forte tension appliquée (jusqu'à 340V), il fallait prévoir un circuit de décharge pour le condensateur après chaque utilisation de la carte de puissance afin d'avoir plus de sécurité pour l'utilisateur et les composants du circuit. Pour automatiser cette décharge, on a utilisé un relais avec une résistance de forte puissance.
3. Il est fortement déconseillé d'utiliser directement la tension de secteur surtout lors du premier essai. Il est préférable d'utiliser une alimentation stabilisée externe branché au bus continu de l'onduleur, le courant de cette alimentation est limité ce qui nous évite les dommages liées à un court-circuit ou une fausse commande.
4. Pour arrêter le moteur, il faut impérativement débrancher la puissance délivrée à l'onduleur avant d'arrêter les PWM. Le non respect de cette règle va endommager directement le module, parce que lorsqu'on arrête les PWM, les signaux de commande envoyée aux IGBT ne s'annulent pas instantanément, alors on risque d'avoir pendant un laps de temps une fausse commande qui va court-circuiter les bras de l'onduleur.
5. Mettre un fusible de 5A avant le module.
6. Mise en place d'un système de refroidissement pour le module, en utilisant un radiateur d'Aluminium avec un système de ventilation.

3.4.1 La technique de bootstrap et le choix de condensateur de bootstrap

Le driver IC des IGBTs de high et low-side suggère une alimentation flottante pour chaque un des trois circuits de high-side qui vont produire les impulsions d'allumage. Une technique très commode pour obtenir une telle alimentation est l'utilisation de la technique de bootstrap. [4][20]

La figure 3.9 montre l'implémentation de cette technique dans le module pour l'un des trois bras de l'onduleur. Le circuit est répété pour chaque bras.

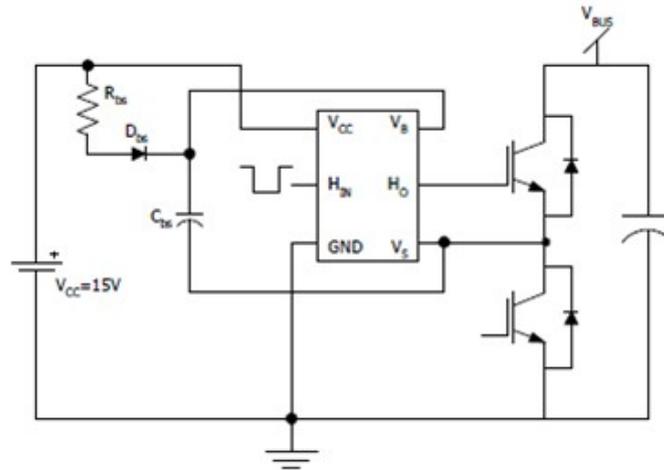


FIGURE 3.9 – Schéma de circuit de bootstrap pour une phase

Le principe de cette technique consiste à profiter de la conduction de IGBT low-side pour charger la capacité de bootstrap C_{bs} sous la tension $V_{cc} = 15V$ à travers la diode D_{bs} . Lorsque l'IGBT low-side se bloque et que la tension réapparaît à ses bornes, la diode D_{bs} se bloque à son tour, et elle doit donc être capable de supporter la haute tension V_{bus} . La charge alors portée par C_{bs} sert à l'alimentation du circuit de commande de IGBT high-side.

La durée de conduction de IGBT low-side doit être suffisante pour que le condensateur de bootstrap puisse se charger totalement. De la même façon, la durée de conduction du transistor high-side est limitée pour éviter la décharge complète du condensateur de bootstrap, compte tenu de la consommation du circuit de commande ; pour la même raison, la fréquence de découpage ne peut être trop basse.

Le dimensionnement de la capacité du condensateur de bootstrap est imposé par plusieurs contraintes :

1. La tension V_{bs} doit être supérieure au niveau under-voltage lockout level pour IC driver.
2. La capacité C_{bs} ne se charge pas exactement à 15V lorsque l'IGBT de low-side est allumé car il faut tenir compte des chutes de tension V_{bs} due à la diode et $V_{ceon(S2)}$ due à l'IGBT low-side.

3. Lorsque l'IGBT high-side est allumé, le condensateur C_{bs} se décharge comme suit :

- Une charge Q_g pour allumer l'IGBT.
- Un courant I_{qbs} consommée par le circuit de high-side dans IC driver.
- Une charge Q_{ls} consommé par le circuit level-shifters dans le IC driver.
- Courant de fuite I_{dl} dans les diodes de bootstrap.
- Courant de fuite I_{cbs} du condensateur de fuite.
- Une charge Q_{rrbs} de recouvrement inverse de la diode de bootstrap.

La charge perdue par la capacité de bootstrap est donnée par la relation :

$$\Delta Q_{bs} = Q_g + Q_{rrbs} + Q_{ls} + \frac{I_{qbs} + I_{dl}}{f_{sw}} \quad (3.1)$$

Où f_{sw} est la fréquence de porteuse de MLI appliquée.

Cette perte de charge dans le condensateur de bootstrap est traduite par une chute de tension par V_{bs} . La valeur de C_{bs} peut être calculée en fonction de la chute de tension désirée dans V_{bs} comme suit :

$$C_{bs} = \frac{\Delta Q_{bs}}{\Delta V_{bs}} \quad (3.2)$$

La chute de tension dans V_{bs} peut être évaluée comme étant le pourcentage de la valeur de V_{bs} avant la fermeture des interrupteurs de high-side. La valeur minimale de V_{bs} dans un cycle de modulation est donnée par la relation :

$$V_{bs} = V_{cc} - V_{fbs} - V_{ceon(S2)} \quad (3.3)$$

Combinant les équations 3.1, 3.2 et 3.3 avec $V_{bs}=1\%$, on obtient :

$$C_{bs} = \frac{Q_g + Q_{rrbs} + Q_{ls} + \frac{I_{qbs} + I_{dl}}{f_{sw}}}{0.01(V_{cc} - V_{fbs} - V_{ceon(S2)})} \quad (3.4)$$

La figure 3.10 représente les valeurs du condensateur de bootstrap recommandé en fonction des fréquences de MLI appliqué, Pour notre projet, on a choisi une fréquence de 10kHz avec $C_{bs}=1.5\mu F$.

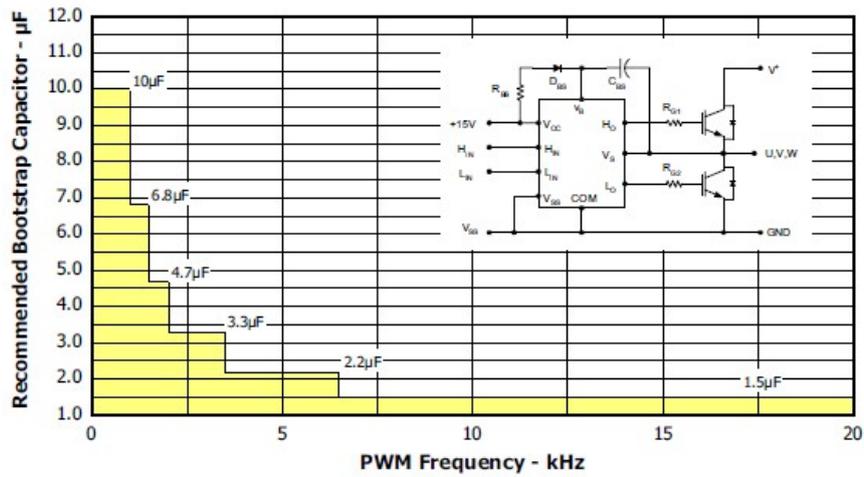


FIGURE 3.10 – Les valeurs de C_{bs} recommandées en fonction des fréquences de porteuse de MLI appliquées

3.4.2 Calcul du condensateur de bus continu

Un condensateur électrolytique est utilisé pour lisser la tension redressée par le pont de redressement incorporé, sa valeur est une fonction inverse de taux d'ondulations autorisé, on peut la calculer par formule suivante donnée par le constructeur : [19]

$$C_{min} = \frac{2P_{in}}{(V_{max}^2 - V_{min}^2)f_{rect}} \quad (3.5)$$

Où P_{in} est la puissance de la charge utilisée.

Dans notre cas, $V_{max}=240\sqrt{2}=340V$ et on prend $V_{min}=300V$ avec $P_{in}=250W$ et $f_{rect}=50*2=100Hz$. Alors $C_{min}=\frac{2*250}{(342^2-325^2)*100}=440\mu F$.

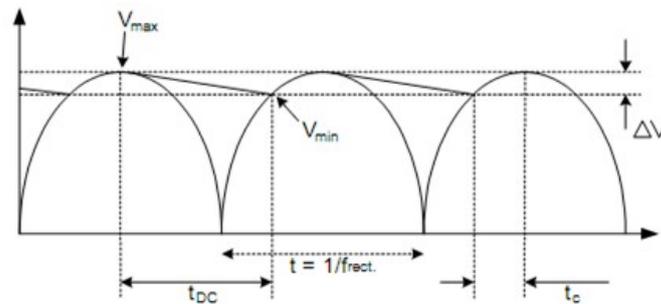


FIGURE 3.11 – Onde de tension redressée montre les périodes de conduction de V_{min} à V_{max}

3.4.3 Alimentation de la carte de puissance

Pour notre montage, le module ne nécessite que une tension de +15V. Nous avons réalisé cette alimentation stabilisée à l'aide d'un régulateur de tension de type LM7815. On a ajouté une alimentation auxiliaire de 12V pour alimenter le ventilateur par un régulateur de tension de type LM7812. Le schéma de cette alimentation est illustré dans la figure 3.12.

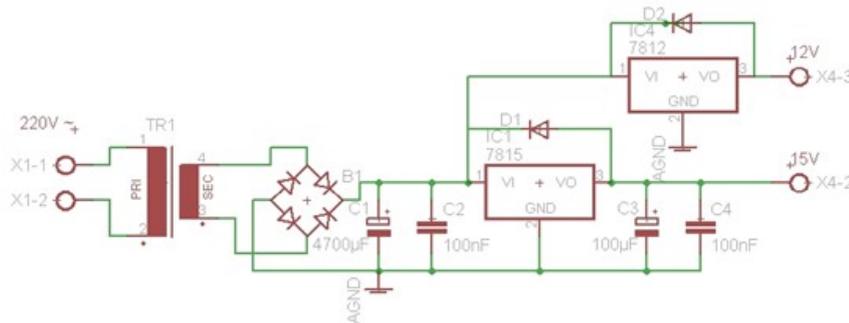


FIGURE 3.12 – Circuit d'alimentation de 15V et 12V

3.5 Conception et réalisation de la carte d'interface pour le DSP

Nous avons réalisé une carte d'interface qui contient non seulement un étage d'isolation pour les sorties PWM de DSP, mais aussi pour les entrées analogique (ADC0-7) et les entrées (CAP1-4/QEP1-4).

A cause de la contrainte de temps, on'a réalisé le contrôleur de vitesse sans la boucle de retour, donc on n'a pas besoin d'utiliser pour le moment les ADC et les CAP/QEP, mais cette interface va être utile pour des travaux ultérieures qui comportent une régulation de vitesse.

3.5.1 Etage d'isolation

Afin de pouvoir utiliser le DSP en toute sécurité, il faut prévoir un circuit d'isolation entre la carte DSP et la carte de puissance, car le DSP est très vulnérable aux perturbations électriques, en plus c'est l'élément le plus coûteux dans notre système de contrôle. La solution convenable pour remédier à ce problème est d'utiliser des optocoupleurs, qui permettent la transmission optique de la commande éloignée.

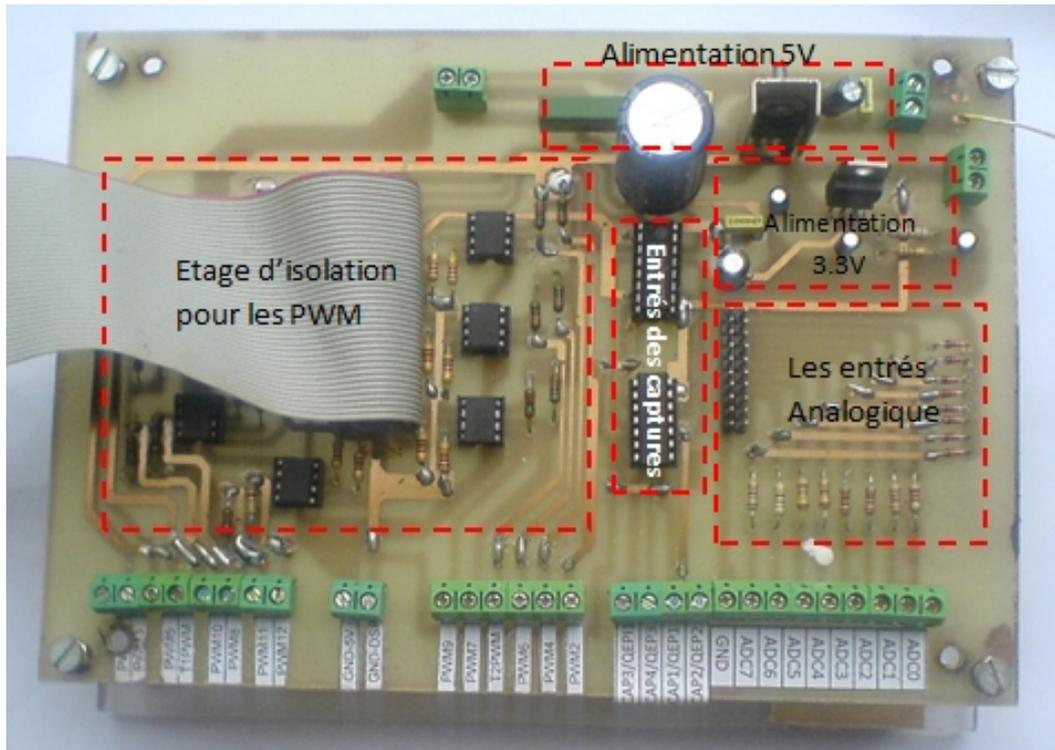


FIGURE 3.13 – Interface d'isolation entre le DSP et la carte de puissance

Les optocoupleurs transmettent l'information de commande et non l'énergie. L'énergie de commande est prélevée sur l'alimentation V_{cc} qui doit être isolée du DSP, donc on doit avoir deux plans de masse dans le circuit d'interface, un GND pour le DSP et l'autre pour l'alimentation.

Notre choix s'est porté sur les optocoupleurs HCPL-2530 de la société HEWLETT PACKARD (voir l'annexe B), vue leur rapidité et leur disponibilité, ce circuit a les caractéristiques suivantes :

- Vitesse rapide jusqu'à 1Mbit/s.
- Double canaux.
- Compatible avec les signaux LSTTL, TTL et CMOS Logique.
- Un faible courant d'entrée (25 mA).
- Temps de montée (rise time) $t_r = 1.3\mu s$.
- Temps de décroissance (fall time) $t_f = 0.2\mu s$.
- $V_{cc}=5V$.

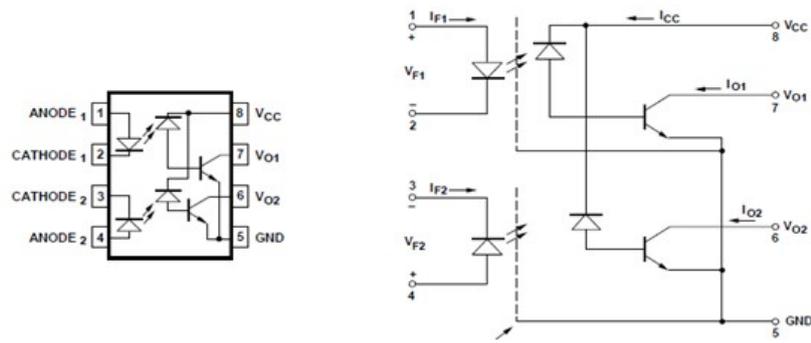


FIGURE 3.14 – L’optocoupleur et son schéma interne

Afin de déterminer les valeurs des deux résistances placées à l’entrée et à la sortie, nous avons réalisé le schéma de la figure 3.15 sur une plaque d’essai, après l’injection d’un signal carré de 5V à l’aide de GBF et la visualisation sur l’oscilloscope, nous avons calibré les résistances à $R_e = 1k\Omega$ et $R_s = 10k\Omega$.

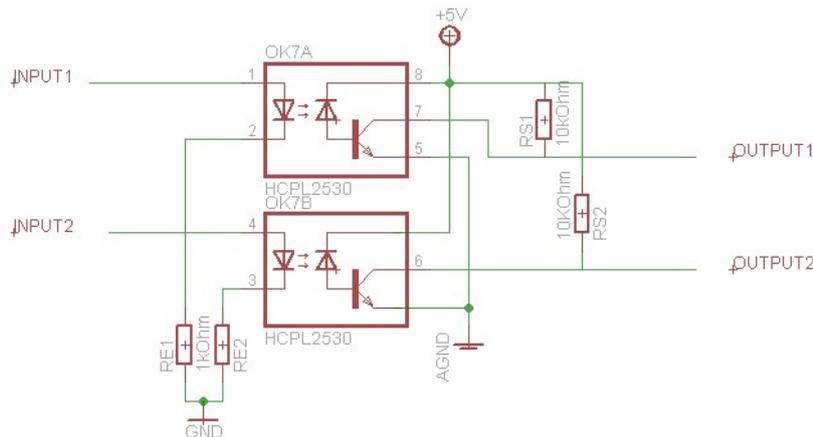


FIGURE 3.15 – Circuit d’isolation

Ce circuit nécessite une tension stabilisée de 5V. On a utilisé le LM7805 comme un régulateur de tension.

La figure 3.16 représente le schéma électrique de cette alimentation, les valeurs des composants sont indiqués sur le schéma.

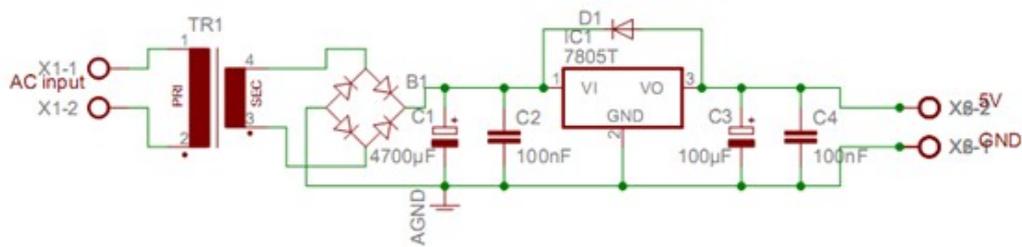


FIGURE 3.16 – Circuit d'alimentation de 5V

La figure 3.17 montre les MLI issues de la carte DSP par le biais de carte d'interface (étage d'isolation). Nous remarquons que les amplitudes des signaux montent jusqu'à 5V et descendent jusqu'à 0V.

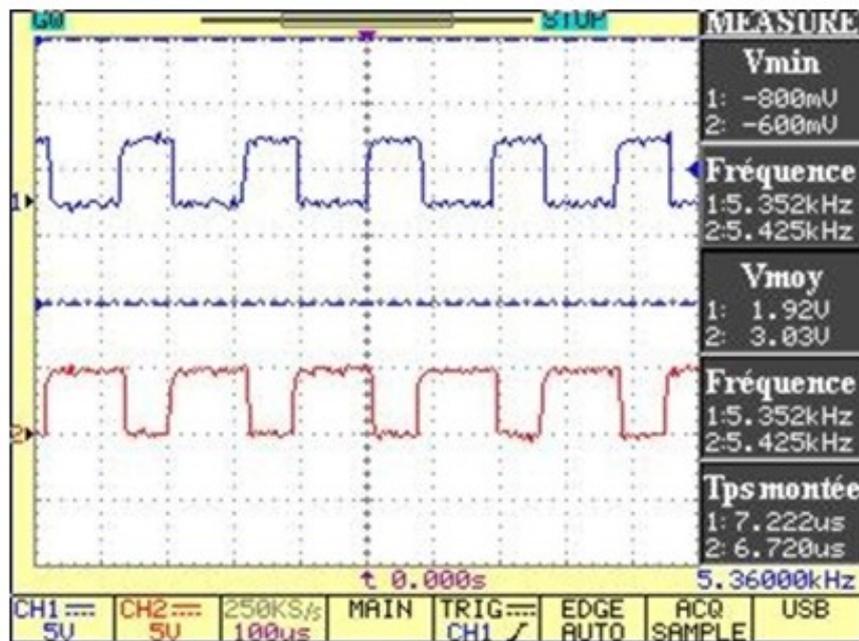


FIGURE 3.17 – Les signaux PWM1 et PWM2

3.5.2 Les entrées CAP (unité de capture)

Pour les entrées CAP du DSP nous avons utilisé un circuit intégré HEF4050B, qui est un Non-Inverting Buffers, parmi les caractéristiques de ce circuit, lorsque il reçoit des signaux de 5V ou plus les sorties suivent toujours la tension de l'alimentation. Pour cette raison l'alimentation de ce circuit sera de 3.3V qui a été réalisée à cet effet. Ainsi nous avons ajouté un Trigger de Schmidt SN74LS14 pour avoir des signaux TTL exploitables par le DSP. La figure 3.18 montre le schéma électrique pour les entrées CAP. [22]

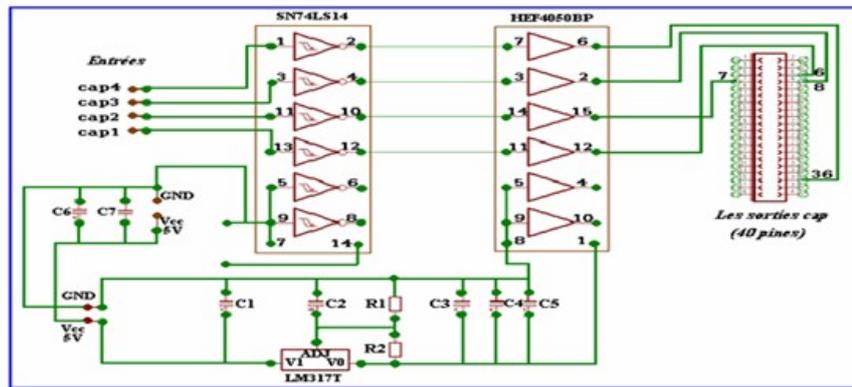


FIGURE 3.18 – Schéma électrique CAP/QEP

Pour pouvoir alimenter le Non-Inverting Buffers HEF4050B, nous avons réalisé une alimentation de 3.3V à partir d'un régulateur de tension qui est un LM317, la figure 3.18 montre le schéma électrique réalisé à cet effet.

La tension à la sortie du LM317 est régie par l'équation :

$$V_{OUT} = 1.25 \times (1 + R_2/R_1) + I_{ADJ} \times R_2 \quad (3.6)$$

D'où : $R_1 = 1k\Omega$ et $R_2 = 620\Omega$.

3.5.3 Les entrées ADC (Convertisseur Analogique Numérique)

Les entrées ADC du DSP (eZdsp F2812) sont de 3V, cela veut dire que tout dépassement au-delà de cette valeur causera la destruction de l'entrée des pins de l'ADC [22]. Pour cette raison, nous avons conçu un circuit de protection pour protéger les entrées ADC du DSP des signaux qui proviennent des capteurs de courant et qui sont déjà amplifiés, et les réduire à une valeur inférieure ou égale à 3V. A cet effet, nous avons utilisé des diodes Zener de type BZX3V montées en série avec des résistances de 220Ω .

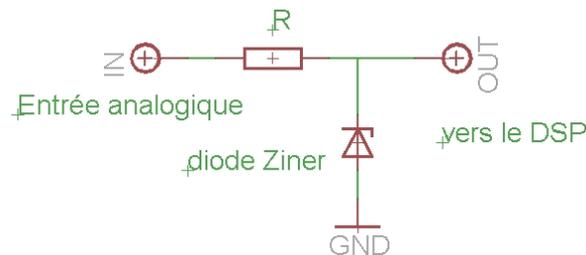


FIGURE 3.19 – Schema électrique de l'entrée de l'ADC

3.6 Conclusion

Dans ce chapitre, nous avons mis en œuvre les différentes étapes pour l'implémentation Hardware de notre système. L'utilisation d'un module à la place d'un onduleur discret nous permet de réaliser un système embarqué relativement simple sans l'utilisation de plusieurs composants, mais d'autre part, le manque de documentation sur ce module nous a causé plusieurs problèmes durant la réalisation. Le module en question est très efficace mais très sensible, il nécessite des systèmes de protection.

L'utilisation des optocoupleurs nous ont permet de réaliser un étage d'isolation très efficace, en plus, l'interface conçue inclue des CAP/QEP et des ADC accessibles facilement.

Les résultats expérimentaux obtenus sont très encourageants pour faire de futurs travaux.

Chapitre 4

Implémentation Software

4.1 Introduction

Suite à l'implémentation Hardware, une implémentation Software va compléter la conception de notre système.

Pour aborder l'implémentation Software, on commence par la présentation de l'environnement de développement intégré qui est conçu pour les plates-formes des DSP TMS320Cx de Texas instruments, le Code Composer Studio CCS.

Ensuite et dans le but de nous familiariser avec la programmation de F2812, nous allons faire des essais préliminaires sur la carte DSP, on commence par la génération de 6 signaux PWM qui nous permettra de mettre en évidence tous les paramètres devant être pris en compte, puis des essais sur l'unité de conversion analogique/numérique.

La commande choisie pour l'implémentation sur la carte DSP est la technique de MLI vectorielle (SVM). Le principe et l'algorithme de cette technique seront expliqués dans la ssection 4.4.

4.2 Outil de développement software

Afin de commencer le développement logiciel avec le DSP, un certain nombre d'outils de base seront nécessaires. Ceux-ci permettront à des routines de logiciel d'être écrites, converties en forme compréhensible par le DSP et alors téléchargées au dispositif de cible. Ces outils sont fournis avec le système de développement en tant qu'élément du paquet complet (Code Composer Studio). Les éléments suivants décrivent les outils de base exigés :

- **éditeur de texte** : Tous les modules de l'algorithme de commande sont écrits en langage C.
- **Compilateur** : permet de détecter les erreurs de syntaxe.
- **Assembleur** : permet générer le code machine correspondant au processeur.
- **éditeur de liens** : Ce code est ensuite " lié " avec les codes nécessaires à la configuration matérielle du DSP et à la configuration des communications avec l'extérieur. A ce stade, le programme est converti en codes directement exécutables par le DSP.
- **Environnement de débogage** : Permet au programme d'être examiné pour le dispositif particulier de DSP ; l'environnement de correction peut être sous forme de simulateur ou d'émulateur.
- **Downloader** : transférer le programme à la carte de développement de DSP.

4.2.1 Code Composer Studio

Le Code Composer Studio (CCS) est un environnement de développement intégré conçu pour les plates-formes des DSP TMS320Cx de Texas instruments. Il est fourni en standard avec la carte de développement pour le DSP[10]. Le Code Composer Studio intègre les outils suivants pour faciliter la construction et la mise au point des programmes de DSP [22] :

- Editeur de code source, debugger, manager de projet,...
- Un compilateur de langage C/C++, un assembleur de code relocalisation, un éditeur de liens,
- Un Simulateur.
- un environnement d'exécution qui permet de télécharger un programme exécutable sur une carte cible, de l'exécuter et de le déboguer au besoin.
- L'analyse d'un programme en cours d'exécution, l'échange des données entre le PC et la carte de développement de DSP et la visualisation des données, tout ça en temps réel.

La figure 4.1 illustre la fenêtre du travail du code composer studio.

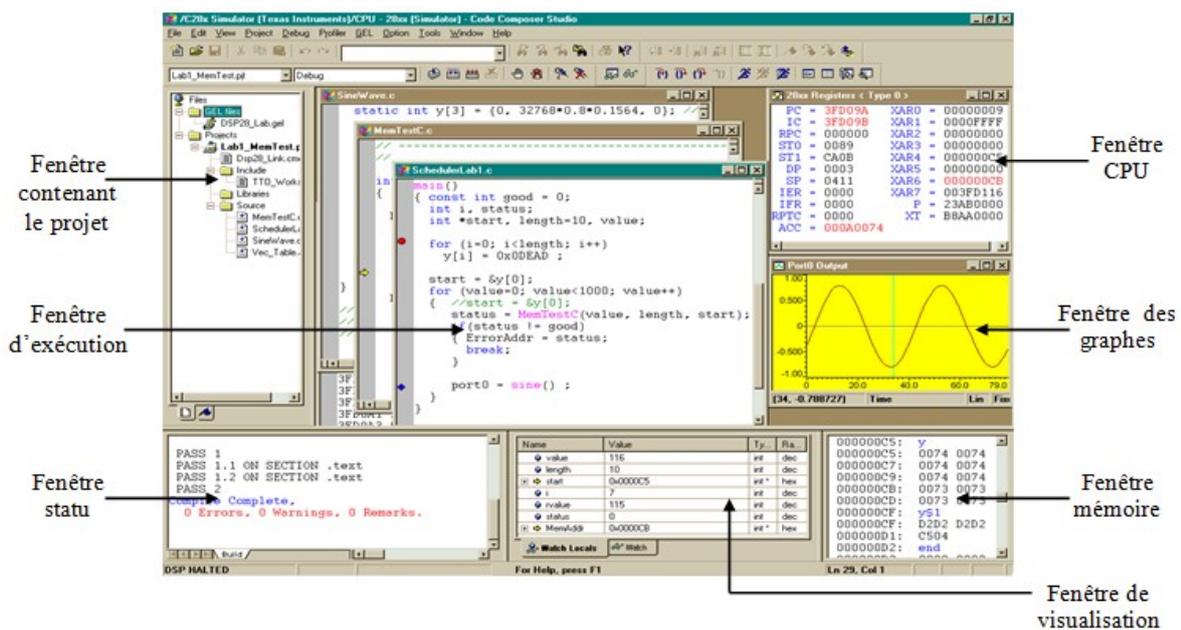


FIGURE 4.1 – Workspace du logiciel code composer

Développement d'un programme avec le CCS

La figure 4.2 montre le flot de design software avec le Code Composer Studio. Les étapes de base sont : édition, compilation, build et debug.

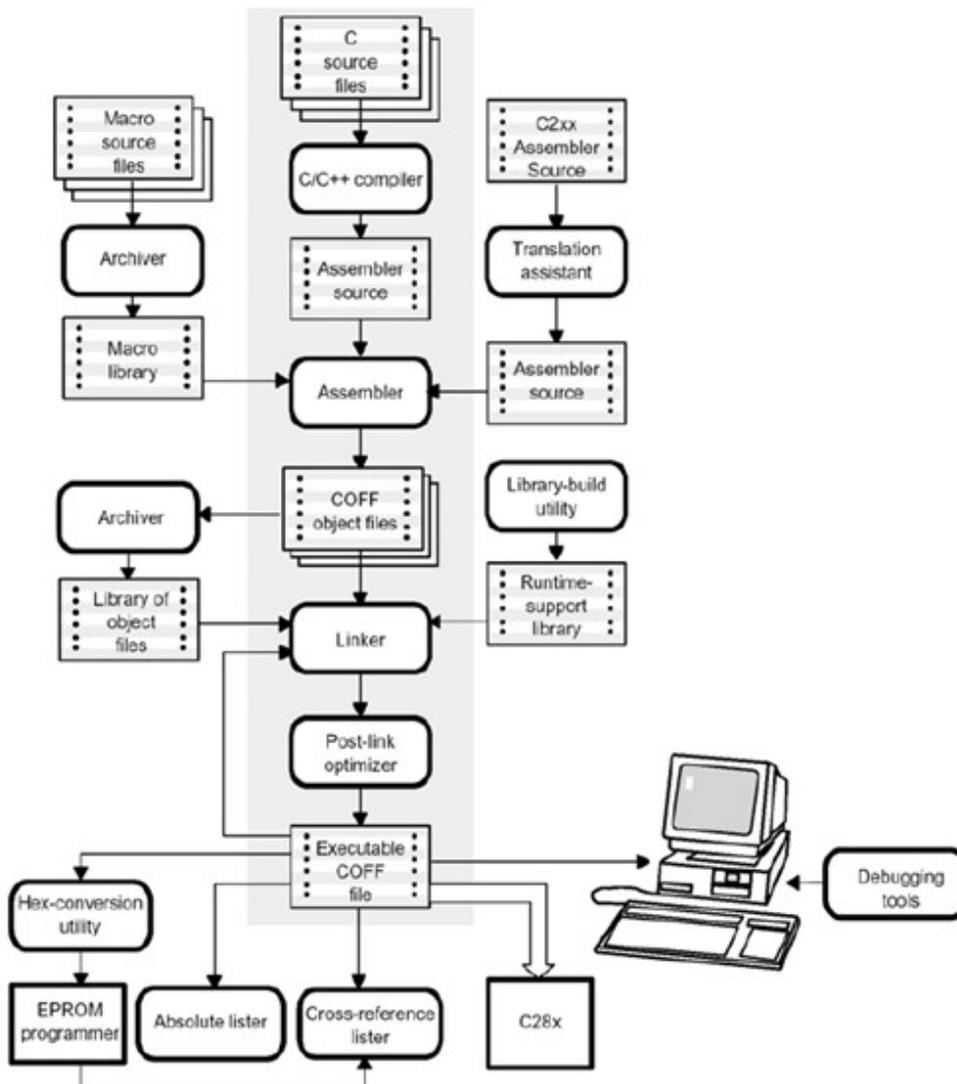


FIGURE 4.2 – Les étapes de développement

Les fichiers DSP281x peripheral header et les exemples des projets [23], donnés par Texas Instruments peuvent être utilisés comme une base pour une plateforme de développement. Tous les programmes suivent un flot de design similaire et recommandé pour la configuration de F2812.

Les procédures suivantes permettent de créer un nouveau projet. Les informations pour un projet sont stockées dans un simple fichier-projet (*.pjt). [10]

1. Dans le menu **Project**, choisir **New**, la fenêtre de création de projet s'apparaît.

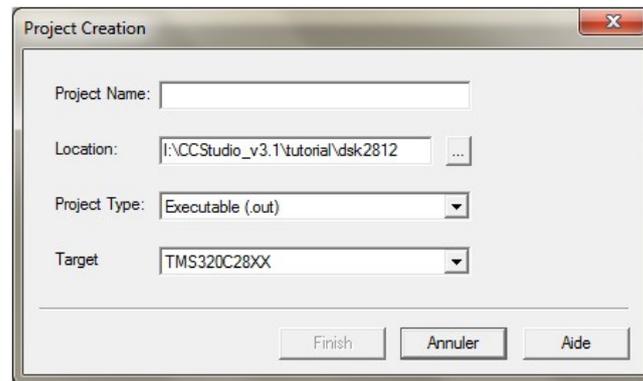


FIGURE 4.3 – La fenêtre de création de projet

Dans le champ **Project Name**, entrez le nom de votre projet. Chaque projet doit avoir un nom unique.

Dans le champ **Location**, spécifier le répertoire où vous voulez enregistrer les fichiers de projet et les fichiers objets générés par le compilateur et l'assembleur. Il est conseillé d'utiliser de nouveau répertoire de chaque nouveau projet.

Choisir le type de projet exécutable (.out) ou Librairie (.lib) dans le champ **Project Type**. Utiliser le champ **Target** pour spécifier la famille du DSP. Cette information est nécessaire quand les outils sont installés pour les cibles multiples. Pour finir cliquer sur **Finish**.

2. Après la création d'un projet, on ajoute les fichiers sources, les bibliothèques, les fichiers header et les fichiers de commande linker à la liste de projet.

Dans le menu **Project**, choisir **Add Files to Project**, la boîte de dialogue Add Files to Project apparaît. Spécifier le fichier à ajouter, utiliser la liste des types de fichier pour sélectionner le type de fichier. Cliquer sur **ouvrir** pour ajouter le fichier spécifié au projet. N'essayez pas d'ajouter manuellement les fichiers header/include (*.h) au projet. Ces fichiers sont ajoutés automatiquement.

3. Pour établir et lancer un programme, les étapes suivantes doivent être effectuées :

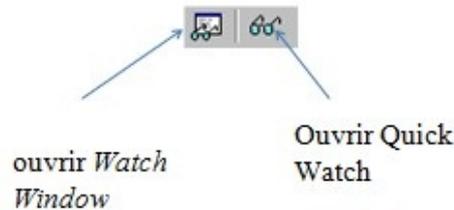
Dans le menu **Project**, choisir **Rebuild All**. Le CCStudio recompile, réassemble, et reconnecte tous les fichiers du projet. Des messages au sujet de ce

processus sont montrés dans une armature au fond de la fenêtre. Par défaut, le fichier (.out) est construit dans un répertoire de débogage situé sous le répertoire courant de projet. Pour changer cet endroit, choisir un autre à partir de la barre d'outils de CCStudio.

Charger le programme sur le DSP en choisissons **Load Program** dans le menu **File**. Le CCStudio ouvre une fenêtre de démontage qui montre les instructions démontées qui composent le programme. Dans le menu **Debug**, choisir **Run** pour lancer l'exécution de programme et choisir **Halt** pour quitter l'exécution de programme.

Watch Window

Lors du débogage d'un programme, il est très utile de voir les changements des valeurs des variables pendant l'exécution. La fonction Watch Window permet le contrôle des valeurs des variables locales et globales et les expressions de C/C++. Pour ouvrir le Watch Window : View→Watch Window, ou cliquer directement sur le bouton Watch Window dans le Watch toolbar.



- Le Watch window (figure 4.4) comporte deux tabulations : Watch Locals et Watch.
- Dans le Watch Locals, le debugger affiche les noms, valeurs et les types des variables dont ils sont locaux dans la fonction en cours d'exécution
 - Dans le Watch, le debugger affiche les noms, valeurs et les types des variables dont ils sont choisis par l'utilisateur.

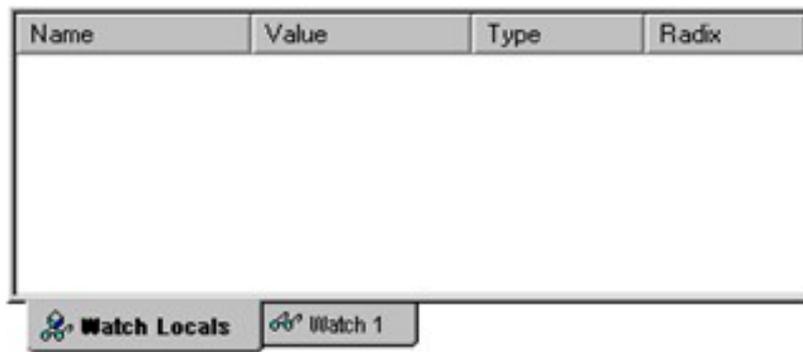


FIGURE 4.4 – Le Watch Window

4.3 Essais préliminaires sur la carte DSP

4.3.1 Génération d'un signal MLI

Le F2812 permet de générer jusqu'à huit formes d'onde PWM (sorties) par chaque Event Manager (voir la section 3.3.4.1). Trois paires indépendantes (six sorties PWM) par les trois unités de Full Compare avec DeadBands programmables, et deux PWM indépendants par GP Timers Compare sans DeadBand.

Pour générer un signal PWM, un Timer approprié est nécessaire pour répéter une période de comptage qui est la même période PWM. Un registre compare (CMPRx) est utilisé pour stocker les valeurs modulantes. La valeur mise dans le registre compare est constamment comparée à la valeur du compteur à rebours. Lorsque les valeurs sont égales, une transition (active haut ou active bas) passe sur la sortie, et lorsqu'une deuxième égalité est faite entre les valeurs, ou quand la fin d'une période du Timer est atteinte, une transition se déclenche sur la sortie. [22]

De cette manière, une sortie d'impulsion est générée dont la durée est proportionnelle à la valeur dans le registre compare. Ce processus est répété pour chaque période du Timer avec différentes valeurs (modulants) dans le registre compare. En conséquence, un signal PWM est généré sur la sortie associée.

Les deux modes d'onde PWM symétrique et asymétrique peuvent être générés par chaque compare unit dans le Event Manager. Ces deux modes de génération exigent la configuration des mêmes registres de Event Manager, le processus de configuration inclue les étapes suivantes : [10]

- Configurer et charger le ACTRx.

- Configurer et charger le DBTCONx, si le dead-band est utilisé.
- Initialiser CMPRx.
- Configurer et charger le COMCONx.
- Configurer et charger le T1CON (pour EVA) ou T3CON (pour EVB) pour commencer l'opération.
- Réécrire le CMPRx par de nouvelles valeurs déterminées.

La forme d'onde PWM asymétrique est caractérisée par une non symétrie par rapport à la période de PWM. Comme il est montré dans la figure 4.5, la largeur de chaque impulsion peut être changée d'un seul côté. Une telle flexibilité de contrôle est particulièrement utile dans la commande de moteur à reluctance variable, dans notre projet, on s'intéresse plus au mode symétrique.

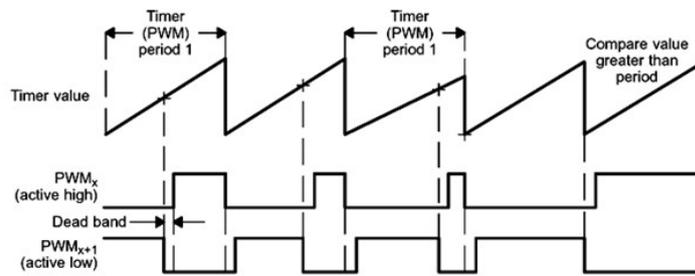


FIGURE 4.5 – Génération de deux PWM en mode asymétrique

La forme d'onde PWM symétrique est caractérisée par une symétrie par rapport à la période de PWM. Les avantages de la forme symétrique par rapport à la forme asymétrique est qu'elle a deux zones inactives de même durée, au début et à fin de la période de PWM. Avec cette symétrie, on a moins de harmoniques dans le courant du moteur asynchrone que l'autre forme [10]. La figure 4.6 montre deux exemples d'une onde PWM symétrique.

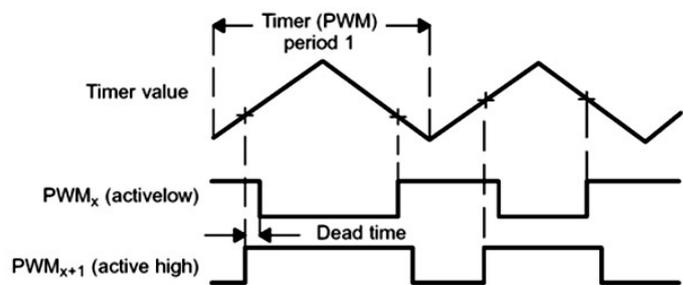


FIGURE 4.6 – Génération de deux PWM en mode symétrique

Un exemple de programme

Un essai concret pour la génération de 6 signaux PWM nous permettra de mettre en évidence tous les paramètres devant être pris en compte.

1. Initialiser GPIO Port A, active les PWM pins pour les utiliser comme des signaux de périphérique.

`GpioMuxRegs.GPAMUX.all = 0x00FF;`

2. Configuration de EVA GP Timer 1 : T1PWM, PWM1-PWM6.

Nous devons configurer plusieurs registres. Pour cela, le registre d'un Timer doit être mis en place. Les bits 8 à 10 du registre control TxCON sont les entrées d'horloge pré diviseur (prescaler) qui définit un autre facteur de division d'horloge. Rappelons que la fréquence est dérivée de :

- L'oscillateur externe (30 MHZ) ;
- Le statut interne PLL (PLLCR : multiplier par 10 / 2 = 150 MHZ) ;
- La haute vitesse d'horloge pré diviseur (HSPCLK = diviser par 2 = 75 MHZ) ;
- Le facteur pré diviseur (1 à 128).

Cela nous donne la possibilité de préciser la période souhaitée pour l'horloge. Pour notre cas, configurer un signal PWM symétrique avec une fréquence de 5 KHZ, nous ramène à utiliser ces calculs :

$$T1_PWM_Freq = \frac{ex_clock_freq * PLL}{HISPCC * TPS * T1PR * 2} \quad (4.1)$$

$$10kHz = \frac{30MHz * 5}{2 * 1 * T1PR * 2} \rightarrow T1PR = 7500.$$

Charger TxPR avec 7500 pour régler la période du Timer à 200µs et pour avoir une largeur d'impulsion de 25%, il suffit juste de charger le registre TxCMPR ou CMPRx avec la valeur 1875 par simple calcul. Le résultat logique est contrôlé au moyen des registres appelés " Registre d'Action de Contrôle ACTRA " et le registre "COMCONA", avec l'aide de ces deux registres, nous pouvons ajuster la forme du signal de sortie PWM à nos besoins.

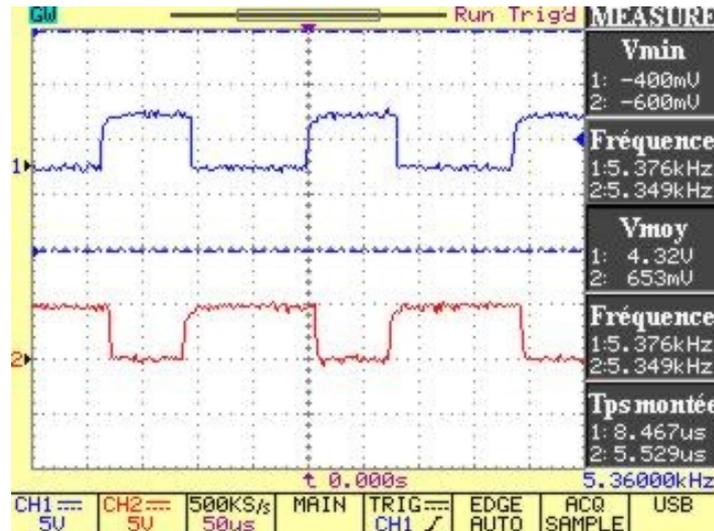


FIGURE 4.7 – Les signaux PWM1 et PWM2

4.3.2 La conversion Analogique/Numérique

Si le système contient une boucle de retour par exemple : mesure de courant, nous devons programmer le convertisseur Analogique / Numérique du DSP pour pouvoir convertir les signaux analogiques en provenance des capteurs.

L'une des plus importantes unités périphériques du TMS320F2812 est le convertisseur Analogique / Numérique (ADC). Cette unité fournit une interface importante entre le contrôleur et le mode réel. La plupart des signaux physiques tel que la vitesse, le courant,... sont des signaux analogiques. Presque tous ils peuvent être représentés comme une tension électrique entre V_{min} et V_{max} , par exemple, 0-3V, qui est proportionnelle au signal original. Le C.A.N (Convertisseur Analogique Numérique) du DSP (TMS320F2812) utilisé est un convertisseur de 12 bits avec échantillonneur bloqueur intégré, l'erreur de quantification est donc de 0,73 millivolts (tension maximale captée 3V). La formule donnant le résultat de la conversion est donnée par :

$$V_{in} = \frac{D * 3.0V}{4095} \quad (4.2)$$

Pour différentes valeurs de tensions injectées, nous mesurons la valeur numérique de la conversion, d'où le résultat est de 12-bits, d'après la formule suivante :

$$Valeur\ Numérique = (Tension\ à\ l'entrée) * \frac{4095}{3.0} \quad (4.3)$$

Toutes les mesures ont été prises par la suite sur Code Composer Studio. Il offre la possibilité de visualiser en temps réel le contenu de la conversion du signal analogique

en valeur numérique. Cela en utilisant l'option sur l'outil de développement appelé " Watch Window ".

Le tableau suivant, montre les résultats obtenus pour différentes valeurs de tensions :

Tension à l'entrée (V)	Valeur numérique théorique	Valeur numérique mesurée
0	683	506
1	1365	1404
1.5	2048	1802
2	2730	2512

TABLE 4.1 – Valeurs numériques mesurées

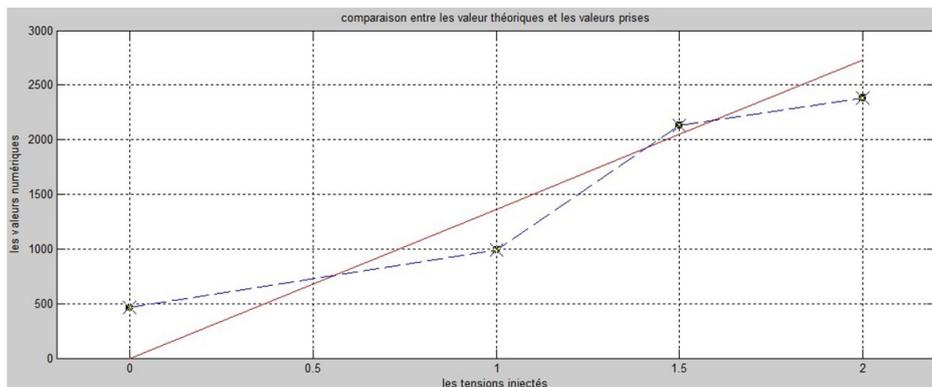


FIGURE 4.8 – Comparaison entre les valeurs mesurées et les valeurs calculées

4.4 L'implémentation de l'algorithme de SVM

4.4.1 Représentation vectorielle des variables

Dans tout système triphasé équilibré, les variables peuvent être représentées par un vecteur dont les composantes, disposées sur 3 axes à 120° , sont les valeurs de ces variables sur chaque phase. [2]

La figure 4.9 représente des variables triphasées sinusoïdales.

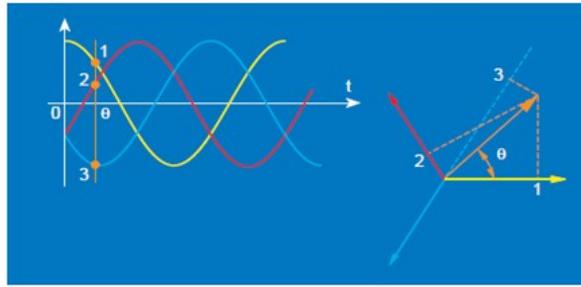


FIGURE 4.9 – Représentation vectorielle des variables

4.4.2 Transformation triphasée biphasé

Les grandeurs triphasés dont la somme des composantes est nulles dans les référentiel fixe (abc) peuvent être représentées par une grandeur biphasé dans un repère fixe (α, β). Par exemple, le vecteur des 3 tensions de références $[V_{ref}]_{abc} = [v_a^* v_b^* v_c^*]^T$ peut être représenté par le vecteur $[V_{ref}]_{\alpha\beta} = [V_\alpha^* V_\beta^*]^T$ en utilisant la transformation de Clarke :

$$v_\alpha^* = \frac{2}{3}(v_a^* - \frac{1}{2}(v_b^* + v_c^*)) \quad (4.4)$$

$$v_\beta^* = \frac{\sqrt{3}}{3}(v_b^* - v_c^*) \quad (4.5)$$

Si les références forment un système sinusoïdal triphasé équilibré caractérisé par \hat{V}_{ref} et la pulsation ω , au vecteur $[V_{ref}]_{abc}$ correspond dans le repère fixe (α, β) au vecteur $\vec{V}_{ref} = [V_{ref}]_{\alpha\beta}$ d'amplitude \hat{V}_{ref} constant et tournant à la vitesse angulaire ω . [7]

Les différentes combinaisons de commande des interrupteurs sont représentées sur la figure 4.10. Il existe huit états.

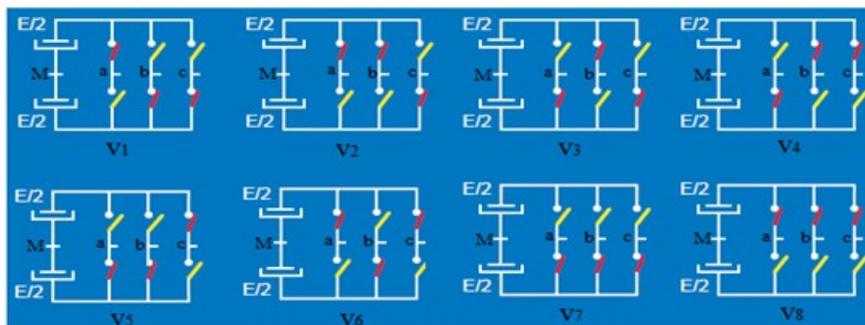


FIGURE 4.10 – Les différentes combinaisons de commande des interrupteurs dans la commande SVM des onduleurs à 2 niveaux

En sortie de l'onduleur les tensions : V_{aM} , V_{bM} et V_{cM} ne peuvent prendre que deux valeurs distinctes ($E/2$ ou $-E/2$). Les tensions : V_{an} , V_{bn} et V_{cn} sont décrites par la relation suivante :

$$\begin{pmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{pmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{pmatrix} V_{aM} \\ V_{bM} \\ V_{cM} \end{pmatrix} \quad (4.6)$$

De même la transformation est appliquée aux huit états de l'onduleur à 2 niveaux ce qui correspond dans le repère (α, β) à huit vecteurs \vec{V}_1 à \vec{V}_8 . Les extrémités des vecteurs \vec{V}_1 à \vec{V}_6 forment les sommets d'un hexagone et les deux vecteurs \vec{V}_7 et \vec{V}_8 forment l'origine du repère (α, β) (figure 4.11).

T_{abc}	v_{an}	v_{bn}	v_{cn}	v_{α}	v_{β}	V	θ_i	\vec{V}_i
1 0 0	$\frac{2E}{3}$	$-\frac{E}{3}$	$-\frac{E}{3}$	$\frac{2E}{3}$	0	$\frac{2E}{3}$	0	\vec{V}_1
1 1 0	$\frac{E}{3}$	$\frac{E}{3}$	$-\frac{2E}{3}$	$\frac{E}{3}$	$\frac{E}{\sqrt{3}}$	$\frac{2E}{3}$	60°	\vec{V}_2
0 1 0	$-\frac{E}{3}$	$\frac{2E}{3}$	$-\frac{E}{3}$	$-\frac{E}{3}$	$\frac{E}{\sqrt{3}}$	$\frac{2E}{3}$	120°	\vec{V}_3
0 1 1	$-\frac{2E}{3}$	$\frac{E}{3}$	$\frac{E}{3}$	$-\frac{2E}{3}$	0	$\frac{2E}{3}$	180°	\vec{V}_4
0 0 1	$-\frac{E}{3}$	$-\frac{E}{3}$	$\frac{2E}{3}$	$-\frac{E}{3}$	$-\frac{E}{\sqrt{3}}$	$\frac{2E}{3}$	240°	\vec{V}_5
1 0 1	$\frac{E}{3}$	$-\frac{2E}{3}$	$\frac{E}{3}$	$\frac{E}{3}$	$-\frac{E}{\sqrt{3}}$	$\frac{2E}{3}$	300°	\vec{V}_6
0 0 0	0	0	0	0	0	0	0	\vec{V}_7
1 1 1	0	0	0	0	0	0	0	\vec{V}_8

TABLE 4.2 – Table d'excitation des interrupteurs de l'onduleur à 2 niveaux

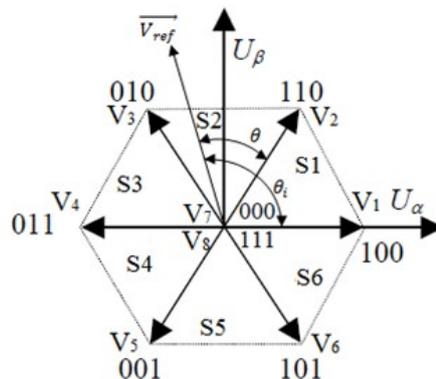


FIGURE 4.11 – Représentation des 8 états de l'onduleur dans le repère (α, β) et définition des secteurs

S_1 : secteur 1, S_2 : secteur 2, ..., S_6 : secteur 6. Deux vecteurs d'espace successifs \vec{V}_i et \vec{V}_{i+1} définissent le secteur i .

Le module V et l'argument θ_i du vecteur \vec{V}_i dans le repère (α, β) homologue du vecteur $[V_{ref}]_{abc}$.

4.4.3 Principe de la modulation vectorielle

L'objectif de la modulation vectorielle est d'approcher, sur une période de découpage T_d , le vecteur de référence \vec{V}_{ref} par les 8 vecteurs (\vec{V}_i , $i=1,8$) disponibles pour l'onduleur à 2 niveaux. Cependant, si le vecteur \vec{V}_{ref} est situé dans un des 6 secteurs de l'hexagonale (S_1 à S_6) on utilise les 2 plus proches vecteurs \vec{V}_i et \vec{V}_{i+1} et le vecteurs zéro $\vec{V}_z = \vec{V}_7$ et \vec{V}_8 [7].

Pour que la valeur moyenne de la tension de référence \vec{V}_{ref} sur une période de découpage T_d soit la même que celle due à l'application des tensions \vec{V}_i et \vec{V}_{i+1} et \vec{V}_z pendant les durée T_i , T_{i+1} et T_z respectivement l'équation vectorielle suivante doit être vérifiée :

$$\vec{V}_{ref} * T_d = \vec{V}_i * T_i + \vec{V}_{i+1} * T_{i+1} + \vec{V}_z * T_z \quad (4.7)$$

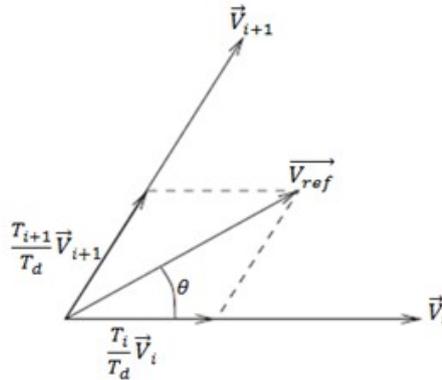


FIGURE 4.12 – Décomposition d'un vecteur de consigne

Comme le vecteur \vec{V}_z est nul, la projection de cette équation sur l'axe \vec{V}_i puis sur l'axe \vec{V}_{i+1} donne :

$$V_{ref} * T_d * \cos(\theta) = V * T_i + V * T_{i+1} * \cos\left(\frac{\pi}{3}\right) \quad (4.8)$$

$$V_{ref} * T_d * \cos\left(\frac{\pi}{3} - \theta\right) = V * T_i * \cos\left(\frac{\pi}{3}\right) + V * T_{i+1} \quad (4.9)$$

Comme $V = \frac{2E}{3}$ et $\cos(\frac{\pi}{3}) = \frac{1}{2}$ la solution après développement est la suivante :

$$T_i = \sqrt{3} \frac{V_{ref}}{E} T_d * \sin(\frac{\pi}{3} - \theta) \quad (4.10)$$

$$T_{i+1} = \sqrt{3} \frac{V_{ref}}{E} T_d * \sin(\theta) \quad (4.11)$$

Si la durée $T_i + T_{i+1} < T_d$, la période est complétée par le vecteur zéro pendant le reste de la période :

$$T_z = T_d - (T_i + T_{i+1}) \quad (4.12)$$

Donc la séquence (000) est appliquée sur une durée de $T_7 = T_z/2$ et la séquence (111) est appliquée sur une durée de $T_8 = T_z/2$.

Avec cette stratégie de modulation, les degrés de liberté sont : [9]

- Le choix des séquences de roue libre par répartition des temps T_7 et T_8 .
- Le choix de la séquence d'apparition des vecteurs dans la période T_d .

En général, le temps de roue libre est réparti uniformément c'est à dire que $T_7 = T_8 = T_z/2$. Afin d'harmoniser les commutations sur une période T_d , le prochain vecteur est choisi tel qu'il ne provoque qu'une commutation sur un seul bras de pont. Ceci traduit la notion d'accessibilité d'un vecteur. Un vecteur est dit accessible depuis le vecteur actuel s'il ne fait commuter qu'un seul bras de pont pour y accéder. La figure 4.13 montre les vecteurs accessibles en fonction du vecteur initial.

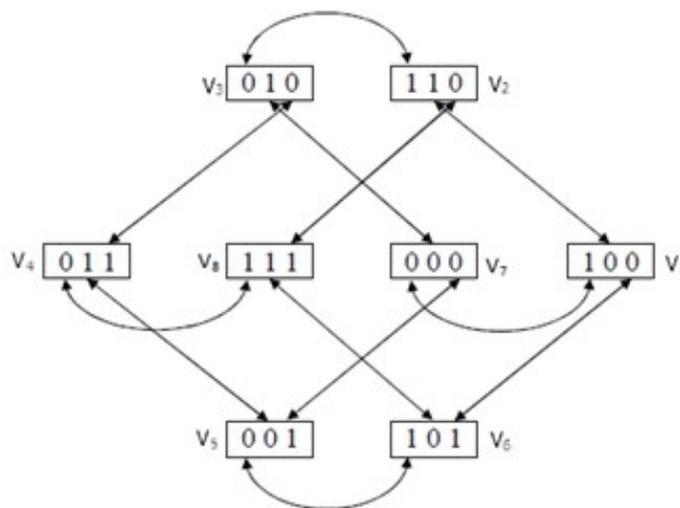


FIGURE 4.13 – Notion d'accessibilité des vecteurs

La figure 4.14 illustre la SVM pour $\vec{V}_i = \vec{V}_1$ et $\vec{V}_{i+1} = \vec{V}_2$.

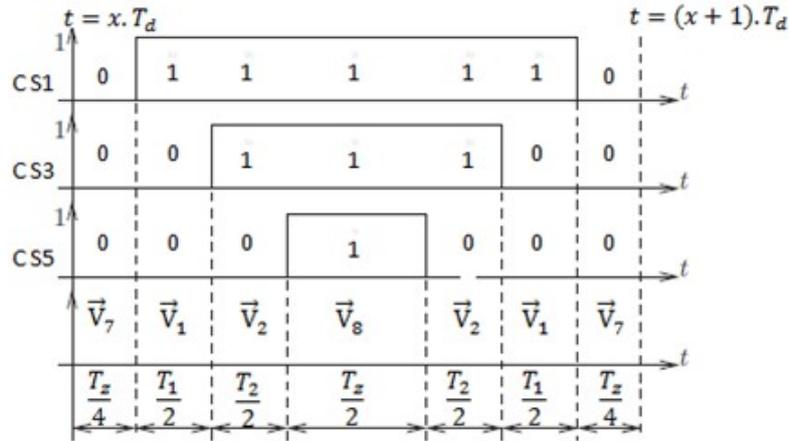


FIGURE 4.14 – SVM avec motifs MLI centrés

4.4.4 Algorithme d'implémentation de la modulation vectorielle

1. Le vecteur de référence $[V_{ref}]_{abc} = [v_a^* v_b^* v_c^*]^T$ est transformé dans le repère (α, β) pour obtenir le vecteur de référence $\vec{V}_{ref}(k)$.
2. On détecte le secteur où se trouve le vecteur $\vec{V}_{ref}(k)$ par :

$$\sin(\theta_i(k)) = \frac{V_{ref\alpha}(k)}{\sqrt{(V_{ref\alpha}(k))^2 + (V_{ref\beta}(k))^2}} \quad (4.13)$$

$$\cos(\theta_i(k)) = \frac{V_{ref\beta}(k)}{\sqrt{(V_{ref\alpha}(k))^2 + (V_{ref\beta}(k))^2}} \quad (4.14)$$

Ce qui permet de déterminer $\theta_i(k)$ dans l'intervalle $[0 \ 2\pi]$.

3. A partir du secteur où se trouve $\vec{V}_{ref}(k)$, on isole les deux configurations \vec{V}_i et \vec{V}_{i+1} qui encadrent \vec{V}_{ref} .
4. Enfin, on exploite les relations qui donnent : T_i, T_{i+1} et T_z .

L'algorithme est récapitulé dans le schéma synoptique ci-dessous

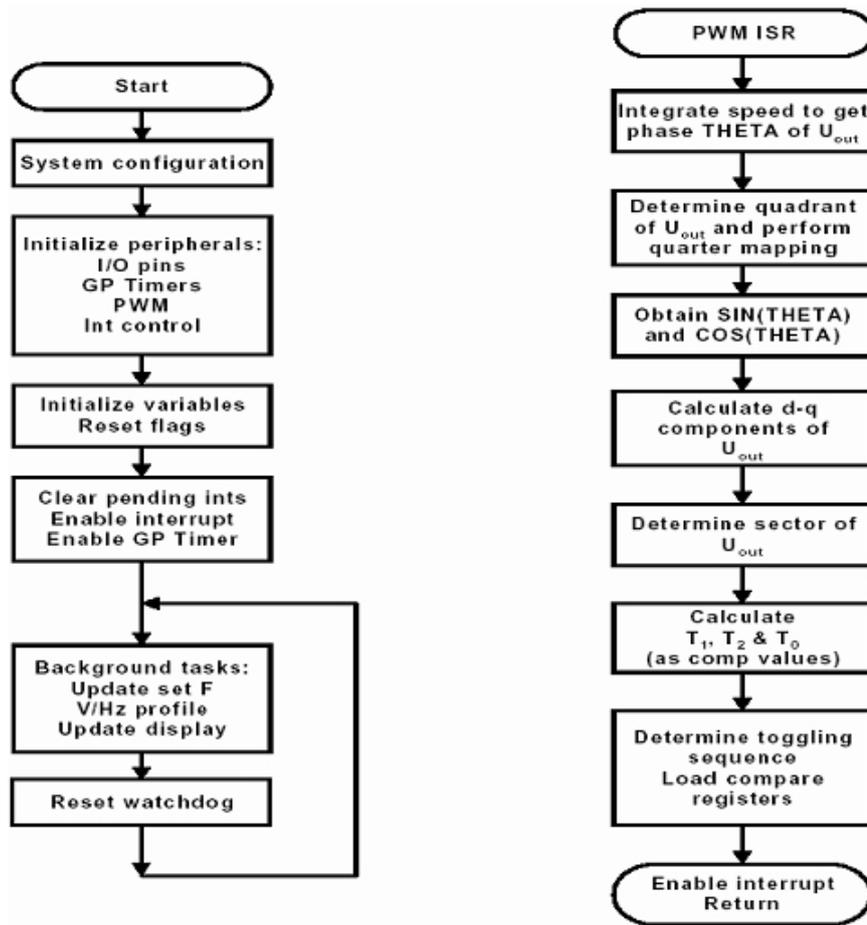


FIGURE 4.15 – Algorithme d’implémentation de la SVM

4.5 Résultats expérimentaux

Après l’implémentation de la commande SVM sur le DSP et la mise en place de la carte d’interface, nous exécutons le programme qui va générer les signaux MLI. Les résultats expérimentaux sont visualisés sur un oscilloscope numérique, puis récupérés sur un ordinateur par une carte d’acquisition du constructeur InstruNet.

InstruNet est un produit matériel et logiciel qui sert à faire une interface entre l’ordinateur et les équipements du laboratoire dans le but de faire l’acquisition des données et le contrôle. Il offre des entrées/sorties analogiques et numériques.

La figure 4.16 représente les signaux MLI sortants de la carte de d'interface, on constate que les signaux de commande montent jusqu'à 5V à cause des optocoupleurs alimentés par 5V.

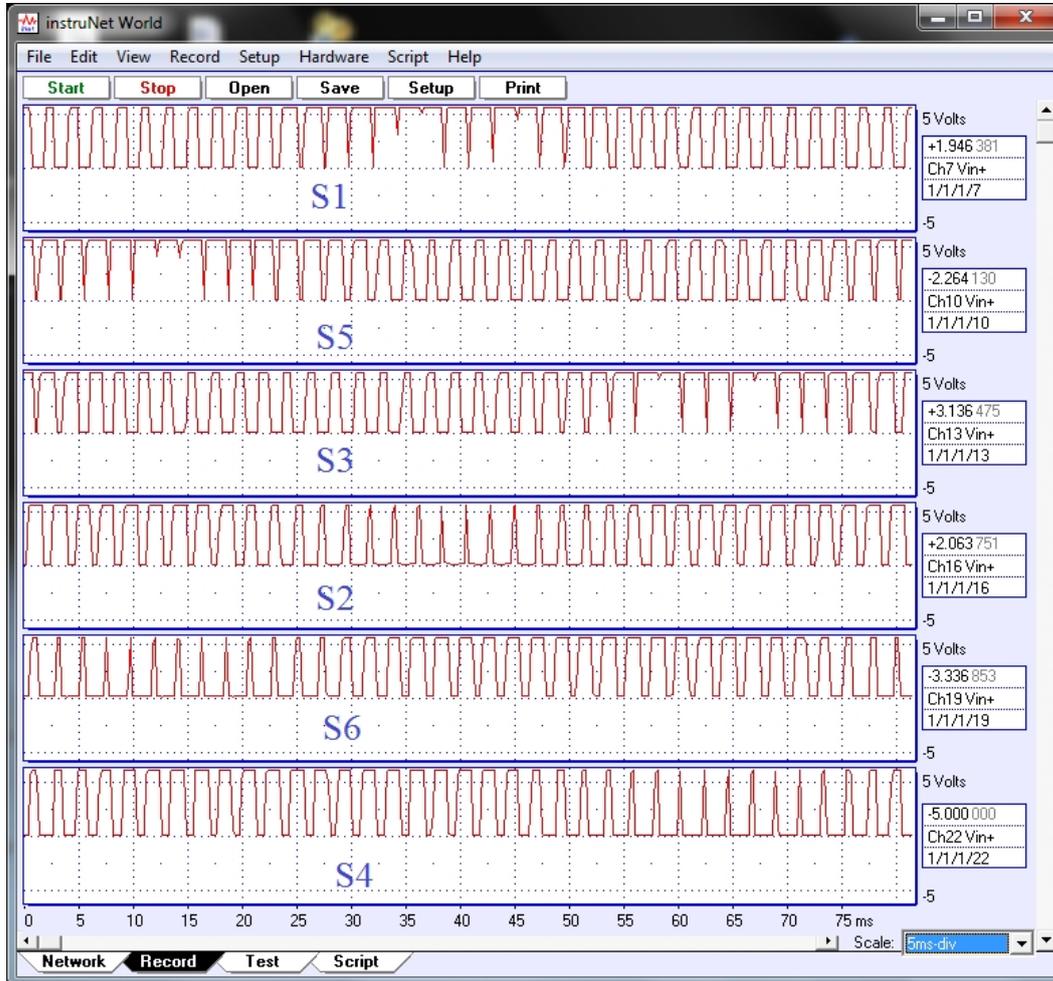


FIGURE 4.16 – Signaux MLI S1 à S6 avec $f_{PWM}=5\text{kHz}$

La figure 4.17 illustre les trois tensions simples sortantes de la carte de puissance. nous constatons que ces tensions sont bien décalées entre elles de 120° . Le bus continu de l'onduleur est égal à $E=315\text{V}$, alors nous remarquons que les amplitudes montent jusqu'à 205V qui est proche de la valeur théorique $2 \cdot E/3=210$.

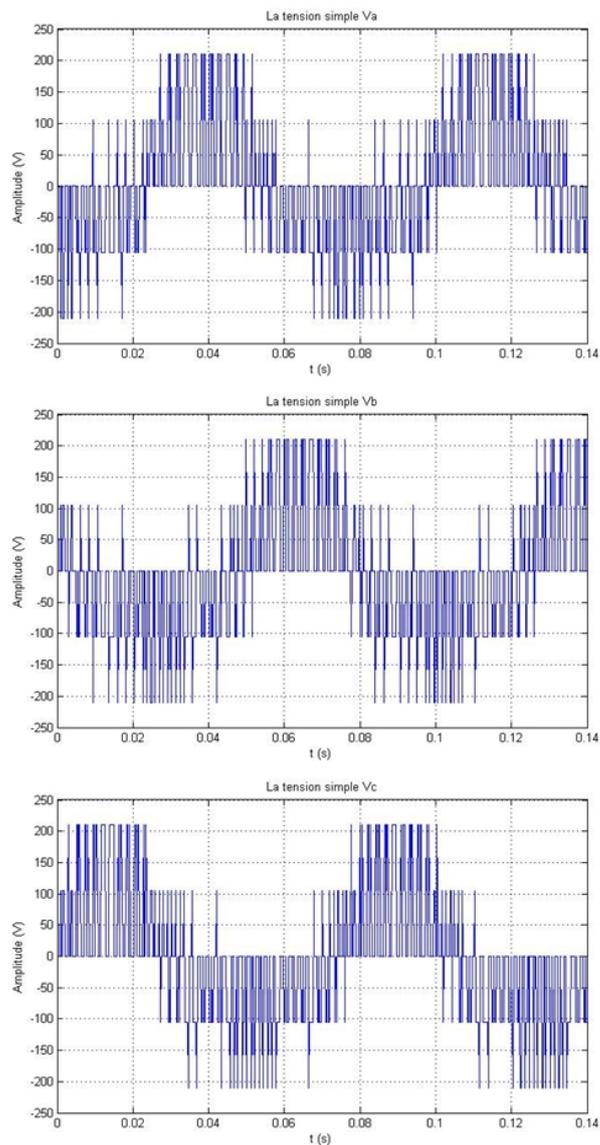


FIGURE 4.17 – Tensions simples V_a , V_b et V_c

La figure 4.18 montre les trois tensions composées, elles sont bien décalées entre elles de 120° . La fréquence des signaux est de 12.25 Hz.

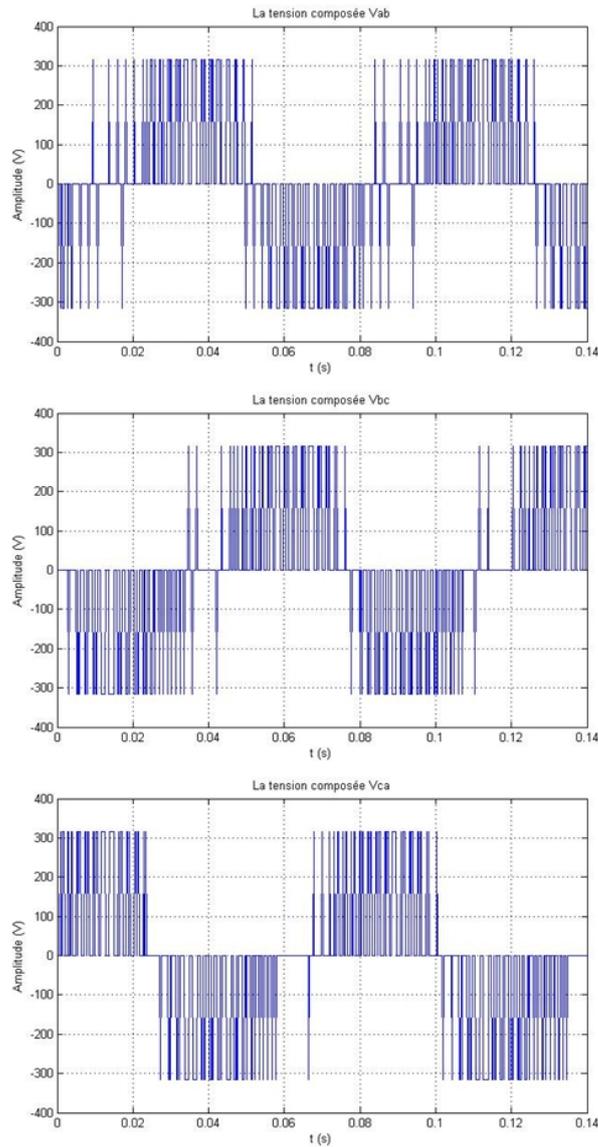


FIGURE 4.18 – Tensions composées V_{ab} , V_{bc} et V_{ca}

La figure 4.19 illustre le spectre des harmoniques des tensions simples et composées. Il est clair que les harmoniques d'ordre multiple de 3 sont nulles, ceci est due au système triphasé équilibré qui élimine les harmoniques d'ordre trois. Les harmoniques d'ordre multiple de 2 sont nulles.

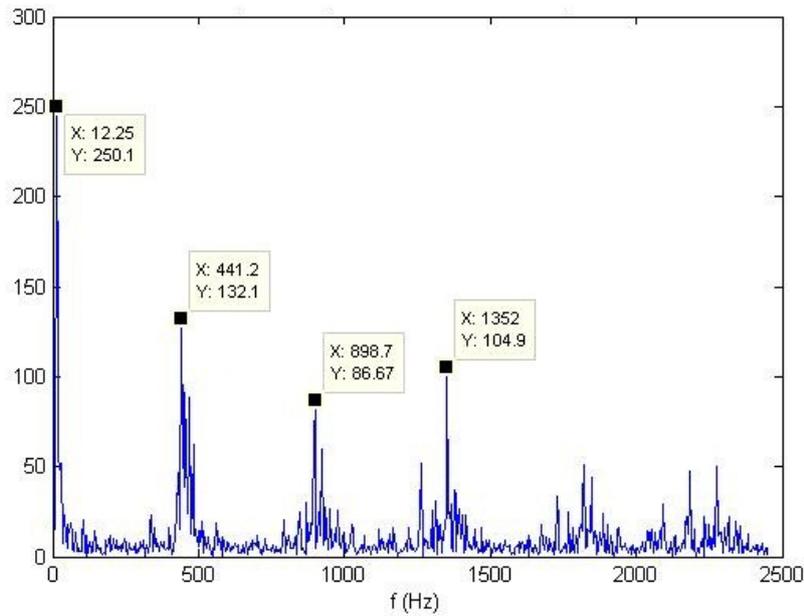


FIGURE 4.19 – Spectre fréquentiel de la tension simple et composée

4.6 Conclusion

Dans ce chapitre nous avons commencé par la présentation de l'outil de développement logiciel, le Code Composer Studio CCS, ensuite, nous avons effectué des essais préliminaires sur le banc d'essai. Pour se familiariser avec la programmation de F2812, nous avons généré un simple signal MLI, en plus nous avons effectué des essais sur l'unité de la conversion Analogique/Numérique. Dans la deuxième partie de ce chapitre, nous avons expliqué le principe de la commande MLI vectorielle, puis élaboré l'algorithme de cette commande. Nous avons constaté que notre commande nécessite une bonne maîtrise des interruptions de ce processeur.

Conclusion générale et perspectives

Le travail que nous avons entrepris concerne l'étude et la réalisation d'un contrôleur embarqué de vitesse à base de DSP TMS320F2812 pour un moteur asynchrone triphasé. Pour cela nous avons commencé par introduire des notions de base sur ce type de moteurs et sur les onduleurs de tension.

Ensuite, nous avons étudié l'onduleur de tension à deux niveaux. Deux types de commande de l'onduleur ont été évoqués : la MLI triangulo-sinusoïdale et la MLI vectorielle. La comparaison entre ces deux techniques nous a montré les avantages apportés par l'utilisation de la technique SVM qui sont : la minimisation des harmoniques de tension et l'augmentation de rendement de l'onduleur.

Lors du passage à l'implémentation nous avons choisi d'utiliser le DSP TMS320F2812, qui est optimisé pour le contrôle numérique des moteurs. Ce processeur est intégré dans un kit de développement eZdsp F2812 DSK construit par Spectrum Digital. Il intègre toutes les fonctionnalités du processeur F2812. Dans l'implémentation hardware on 'a utilisé le module de puissance IRAM136-0461G à la place d'un onduleur discret ce qu'à permis de réaliser un système embarqué relativement simple avec l'utilisation d'un minimum de composants, mais d'autre part, le manque de documentation sur ce module nous a causé plusieurs problèmes durant la réalisation.

Entre l'onduleur et le DSP, nous avons réalisé une carte d'interface, elle contient non seulement l'étage d'isolation pour toutes les PWM1-14 mais aussi les entrées des capteurs CAP/QEP et les entrées analogiques ADC0-7.

Dans la dernière partie de ce mémoire, on a abordé l'implémentation Software de notre système. On a fait une brève présentation de l'outil de développement logiciel, le Code Composer Studio CCS. En suite, pour se familiariser avec la programmation de DSP nous avons exposé un essai concret de la génération d'un simple signal MLI. On

a terminé le chapitre par l'élaboration de l'algorithme de MLI vectorielle (SVPWM).

Enfin, ce projet nous a permis de comprendre les différentes étapes de développement d'un système embarqué en se basant sur un DSP.

On termine par proposer quelques perspectives pour l'avenir :

- Nous espérons que cette étude sera suivie par l'étude d'une commande avec une boucle de retour ce qui nous permet d'améliorer la qualité de contrôle, par exemple une commande vectorielle de moteur asynchrone.
- Utilisation d'un circuit de temporisation entre l'alimentation de bus continu et le lancement des signaux MLI.
- Utilisation de la thermistance et du shunt intégrés dans le module pour faire une protection contre le sur-échauffement et sur-courant dans module, en utilisant la fonction de déclenchement I_{TRIP} .

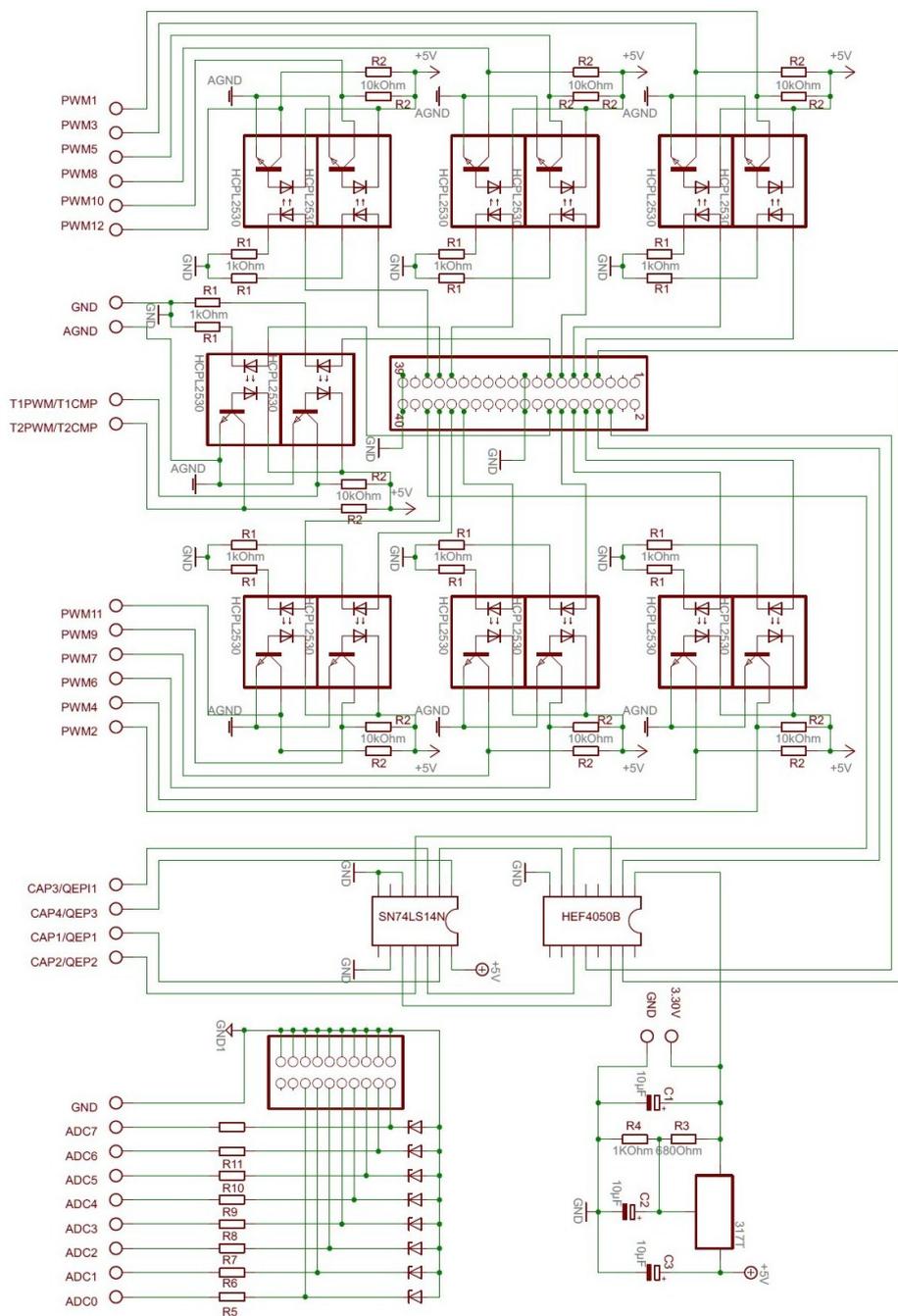
Bibliographie

- [1] ACHOURI Radouane et HIDOUCHE Mohamed, "*Commande vectorielle de la machine asynchrone*", Projet de fin d'étude, Ecole Nationale Polytechnique d'Alger, Juin 2007.
- [2] "*Intersection*", Le magazine Schneider de l'enseignement technologique et professionnelle, Juin 1998.
- [3] WILDI Theodor, "L'électrotechnique", 3^{ème} édition, éditions de Boek Université, Bruxelles 2000, 1196p.
- [4] BENDIB Douadi, "*Etude et réalisation d'une commande MLI on-line sur circuit FPGA*", mémoire de Magister, Ecole Nationale Polytechnique d'Alger, Juin 2009.
- [5] REZAOUI Med Mounir, "*Etude et commande d'un moteur asynchrone monophasé par convertisseurs classiques et matriciels*", Projet de Fin d'Etude, Ecole Nationale Polytechnique d'Alger, Juin 2005.
- [6] A.TOLIYAT Hamid et CAMPBELL Steven, "*DSP-based Electromechanical Motion Control*", CRC Press, 2004.
- [7] CHEKIREB, "*cours d'Electronique de puissance, 4^{ème} année EA*", cours non publié, Ecole Nationale Polytechnique d'Alger.
- [8] CAPITANEANU Stefan Laurentiu, "*Optimisation de la fonction MLI d'un onduleur de tension deux-niveaux*", These de Doctorat, Institut National Polytechnique de TOULOUSE, Novembre 2002.
- [9] HOBRAICHE Julien, "*Contribution à l'étude des machines synchrones à aimants permanents internes à large espace de fonctionnement. Application à l'alternodémarrreur*", Thèse de Doctorat, Université De Technologie De Compiègne, Novembre 2004.
- [10] BARIS Bagci, "*Programming and Use of TMS320F2812 DSP to Control and Regulate Power Electronic Converters*", Master Thesis, Fachhochule KOLN University of applied sciences Cologne, October 2003.

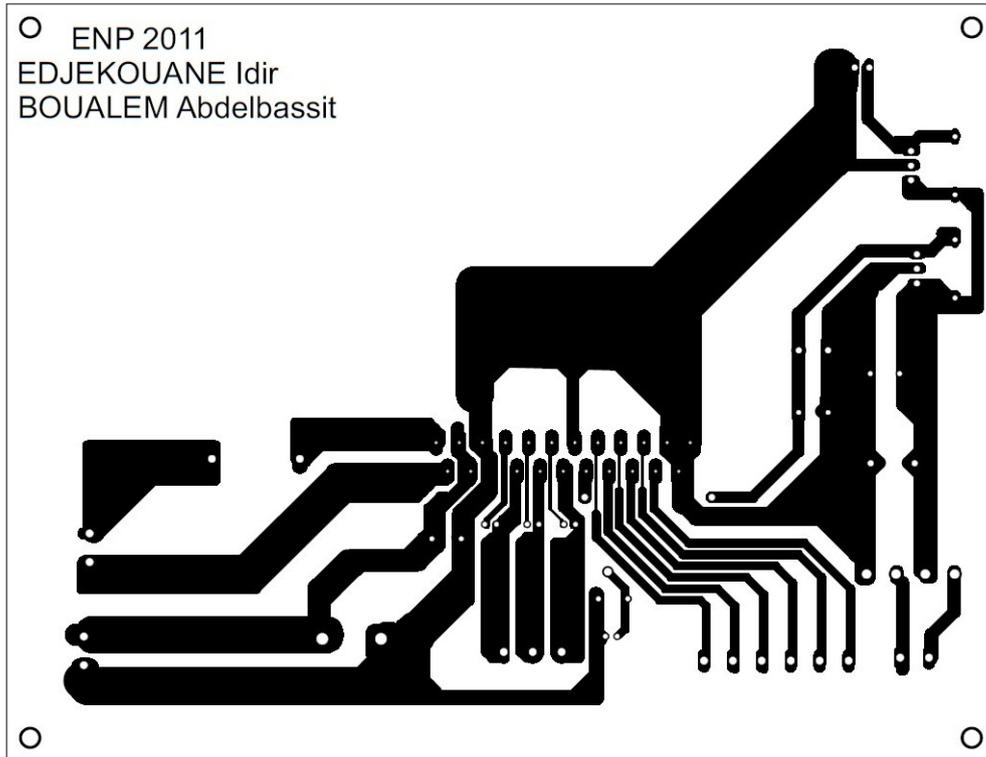
-
- [11] HAMAMI Latifa, "*DSP et famille C6000*", cours non publié, Ecole Nationale polytechnique d'Alger.
- [12] KIATTISIN Kanjanawanishkul, "*Development of a hardware interface between a DSP (TMS320F2812) and a Converter*", Project work, Institut für Leistungselektronik und Elektrische Antriebe, Septembre 2005.
- [13] RAMAN Nair Harish Gopala Pillai, "*Design and development of embedded DSP controllers for power electronic applications*", Master Thesis, The University of Texas at Arlington, Mai 2006.
- [14] BOUDJIT Kamel, "*Introduction aux processeurs de signaux DSP*", cours à l'Université Saad Dahlab de Blida.
- [15] TADRIST Nadia, "*Implementation par DSP d'une commande en vitesse d'une machine à courant continu sans collecteur (BDCM)*", mémoire de Magistère de l'USTHB, Novembre 2009.
- [16] Data Manual, "*TMS320F2808 Digital Signal Processors*", Texas Instruments, Literature Number : SPRS174R, Avril 2001 Revised, Mai 2010, www.ti.com.
- [17] Spectrum Digital Inc. "*eZdsp F2812 Technical Reference Rev D*", Février 2003, www.spectrumdigital.com.
- [18] BORMANN Frank, "*Digital Signal Controller Digital Signal Controller TMS320F2812*", University of Applied Sciences Zwickau (FH).
- [19] Datasheet de module "*IRAM136-0461G*", Literature Number : PD-97271, International Rectifier, www.irf.com.
- [20] "*Application note AN-1044 revA*", International Rectifier, www.irf.com.
- [21] "*TMS320x281x DSP Event Manager (EV) Reference Guide*", Literature Number : SPRU065C, November 2004, www.ti.com.
- [22] BOUDJIT Kamel, "*Synchronisation en vitesse de deux moteurs à courant continu à base de DSP (TMS320F2812)*", mémoire de Magister, l'USTHB, 2008.

Annexe A

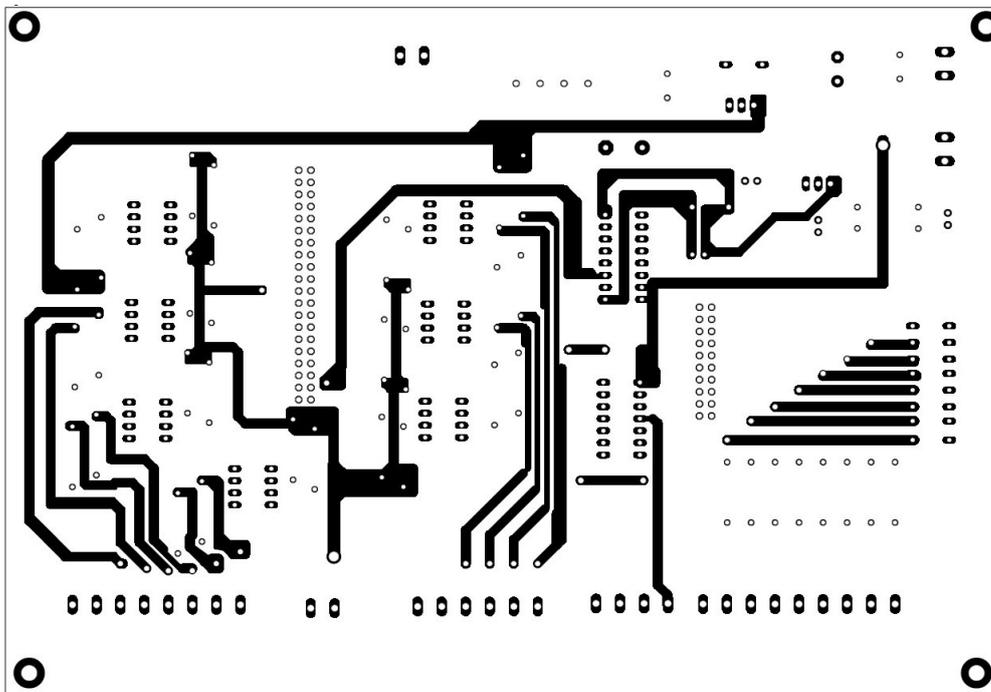
Schéma électrique de la carte d'interface de DSP.



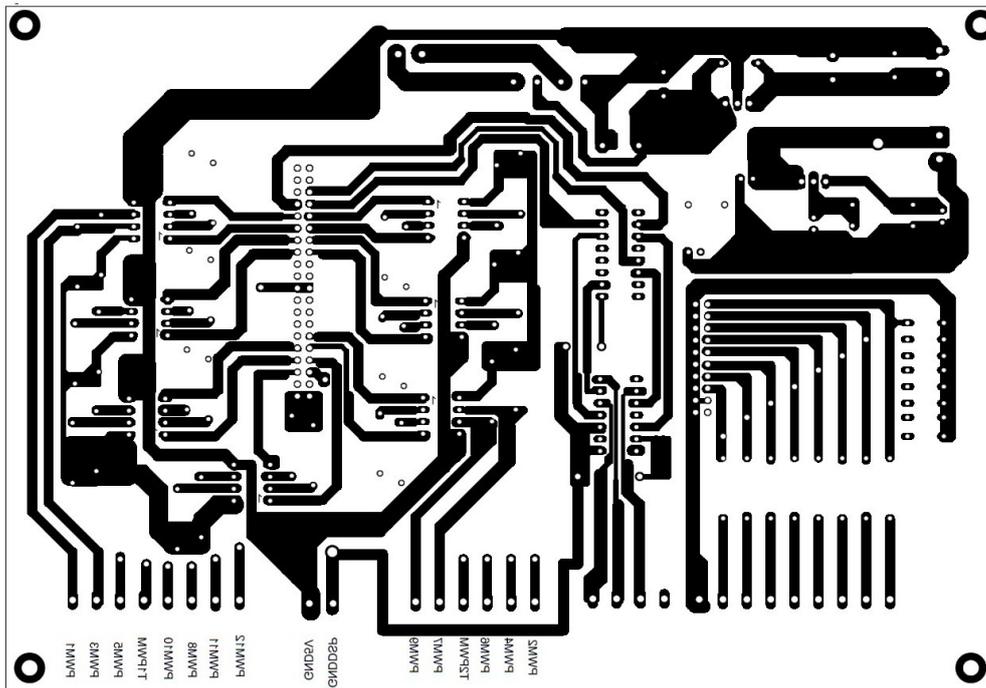
Circuit imprimé de la carte de puissance (côté cuivre).



Circuit imprimé de la carte d'interface (côté composants).



Circuit imprimé de la carte d'interface (côté cuivre).



Annexe B

PD-97271 RevA

International
IR Rectifier

Plug N Drive™ Integrated Power
Module for Energy Efficient Motor Drives

IRAM136-0461G
iMOTION™ Series
4A, 600V

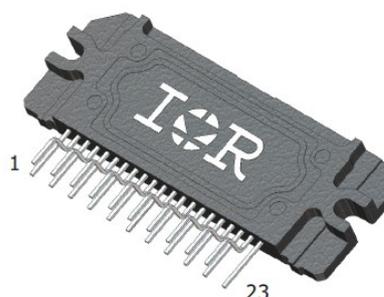
Description

International Rectifier's IRAM136-0461G is an Integrated Power Module developed and optimized for electronic motor control in energy saving applications. Targeting the sub 300W three-phase motor drive applications, such as fan or refrigerator compressor drives, this module offers the highest level of integration available in the market today. It features an input diode rectification bridge and a three-phase inverter, complete with bootstrap diodes, high voltage gate driver IC, current shunt resistor and temperature sensor. This high performance AC motor-driver is housed in a compact single-in-line isolated package for a very simple design.

The internal shunt offers easy current feedback for precise control and safe operation. A built-in temperature monitor and logic level shut-down function, along with the short-circuit rated IGBTs and integrated under-voltage lockout function, deliver high level of protection and fail-safe operation.

Features

- Internal Rectifier Diode Bridge
- Internal Shunt Resistor
- Integrated Gate Drivers and Bootstrap Diodes
- Temperature Monitor
- Undervoltage lockout for all channels
- Matched propagation delay for all channels
- Schmitt-triggered input logic
- Cross-conduction prevention logic
- Lower di/dt gate driver for better noise immunity
- Motor Power range 0.1~0.3kW / up to 253V, 50/60Hz
- Isolation 2000V_{RMS}/1min



Absolute Maximum Ratings

Parameter	Description	Max. Value	Units
V _{RRM}	Input Bridge Blocking Voltage	600	V
V _{CES}	IGBT Blocking Voltage	600	
V ⁺	Positive Bus Input Voltage	450	
I _O @ T _C =25°C	RMS Phase Current (F _{PWM} =20kHz)	3.6	
I _O @ T _C =100°C	RMS Phase Current (F _{PWM} =20kHz)	2	
I _{O PK}	Pulsed RMS Phase Current (t _p <100ms, F _{PWM} =20kHz)	5	
F _{PWM}	PWM Carrier Frequency	20	kHz
P _d	Power dissipation per IGBT @ T _C =25°C	16	W
V _{ISO}	Isolation Voltage (1min)	2000	V _{RMS}
T _J (IGBT & Diodes)	Operating Junction temperature Range	-40 to +150	°C
T _J (Driver IC)	Operating Junction temperature Range	-40 to +150	
T	Mounting torque (M3 screw)	1.0	Nm

Dual Channel, High Speed Optocouplers

Technical Data

HCPL-2530 HCPL-0530
 HCPL-2531 HCPL-0531
 HCPL-4534 HCPL-0534

Features

- 15 kV/ μ s Minimum Common Mode Transient Immunity at $V_{CM} = 1500$ V (HCPL-4534/0534)
- High Speed: 1 Mb/s
- TTL Compatible
- Available in 8 Pin DIP, SO-8, and 8 Pin DIP – Gull Wing Surface Mount (Option 020) Packages
- High Density Packaging
- 3 MHz Bandwidth
- Open Collector Outputs
- Guaranteed Performance from 0°C to 70°C
- Safety Approval
 UL Recognized – 2500 V rms for 1 minute (5000 V rms for 1 minute for Option 020) per UL1577
 CSA Approved
- Single Channel Version Available (4502/3, 0452/3)
- MIL-STD-1772 Version Available (55XX/65XX/4N55)

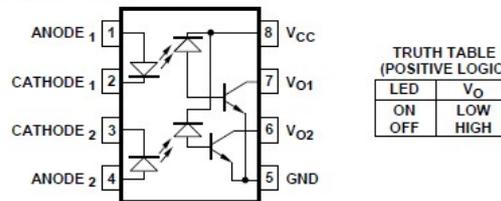
Applications

- Line Receivers – High Common Mode Transient Immunity (> 1000 V/ μ s) and Low Input-Output Capacitance (0.6 pF)
- High Speed Logic Ground Isolation – TTL/TTL, TTL/LTTL, TTL/CMOS, TTL/LSTTL
- Replace Pulse Transformers – Save Board Space and Weight
- Analog Signal Ground Isolation – Integrated Photon Detector Provides Improved Linearity over Phototransistor Type
- Polarity Sensing
- Isolated Analog Amplifier – Dual Channel Packaging Enhances Thermal Tracking

Description

These dual channel optocouplers contain a pair of light emitting diodes and integrated photodetectors with electrical insulation between input and output. Separate connection for the photodiode bias and output transistor collectors increase the speed up to a hundred times that of a conventional phototransistor coupler by reducing the base-collector capacitance.

Functional Diagram



TRUTH TABLE
(POSITIVE LOGIC)

LED	V _O
ON	LOW
OFF	HIGH

A 0.1 μ F bypass capacitor between pins 5 and 8 is recommended.

CAUTION: It is advised that normal static precautions be taken in handling and assembly of this component to prevent damage and/or degradation which may be induced by ESD.

Annexe C

Code source de MLI

```
// Etape 1 : Entrer les header files nécessaires
// DSP28_Device.h : spécifier la bibliothèque de DSP choisi
include "DSP28_Device.h"
void main(void)
{
// Etape 2 : Initialiser le système de contrôle des registres, PLL, chien de garde, Hor-
loges à l'état par défaut :
// cette fonction est situé dans le fichier DSP28_SysCtrl.c file.
InitSysCtrl();
// Etape 3 : sélectionner les GPIO à utiliser dans l'application :
// cette fonction est situé dans le fichier DSP28_Gpio.c file.
// Initialiser les GPIO
EALLOW;
// Activer les PWM pins
GpioMuxRegs.GPAMUX.all = 0x00FF; // EVA PWM 1-6 pins
EDIS; // Désactiver les PWM pins
// Etape 4 : Initialiser le tableau de vecteur PIE :
// le PIE vecteur est initialise par des pointeurs à le shell Interrupt
// Service Routines (ISR). le shell routines est situé dans le fichier DSP28_DefaultIsr.c.
// désactiver et clear tous les interruptions de CPU :
DINT;
IER = 0x0000;
IFR = 0x0000;
// Initialiser le Pie Control Registers à la l'état par défaut :
// cette fonction est situé dans le fichier DSP28_PieCtrl.c file.
InitPieCtrl();
```

```

// Initialiser le tableau de vecteur PIE à un état connu :
// cette fonction est situé dans le fichier DSP28_PieVect.c.
InitPieVectTable();
// Etape 5 : Configuration de Event Manager A
// 1) Initialiser le EVA Timer1
EvaRegs.T1PR = 0xFFFF; // Timer1 period
EvaRegs.T1CMPR = 0x3C00; // Timer1 compare
EvaRegs.T1CNT = 0x0000; // Timer1 counter
// 2) configurer le registre de contrôle T1CON
// TMODE = continuous up/down
// Activer le Timer
// Activer le Timer compare
EvaRegs.T1CON.all = 0x1042;
// 3) charger les valeurs au compare registre pour les PWM1-PWM6
EvaRegs.CMPR1 = 0x0C00;
EvaRegs.CMPR2 = 0x3C00;
EvaRegs.CMPR3 = 0xFC00;
// 4) Choisir le mode d'activation de sorites PWM (active haut/active bas) par la
configuration de registre compare action control register ACTRA.
// output pin 1 CMPR1 - active high
// output pin 2 CMPR1 - active low
// output pin 3 CMPR2 - active high
// output pin 4 CMPR2 - active low
// output pin 5 CMPR3 - active high
// output pin 6 CMPR3 - active low
EvaRegs.ACTRA.all = 0x0666;
// 5) configurer le registre de dead-band
EvaRegs.DBTCONA.all = 0x0000; // désactiver le deadband
// 6) configurer les compares registres
EvaRegs.COMCONA.all = 0xA600;
//Etape 6 : lancer une boucle infinie
for(;;);
}

```