



الجمهورية الجزائرية الديمقراطية الشعبية  
Republique Algerienne Democratique Et Populaire  
وزارة التعليم العالي والبحث العلمي



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة العربي التبسي - تبسة

Université Larbi Tébessi – Tébessa –

Faculté des Sciences et de la Technologie

Département de Génie électrique

## MEMOIRE

Présenté pour l'obtention du **diplôme de Master Académique**

En : **Électronique**

Spécialité : **Instrumentations**

Par : Fares Chahinaz  
Chouchane Tahani

Sujet

# COMMANDE D'UN CONVERTISSEUR DC/DC PAR UN MICRO CONTROLEUR

Déposé et évaluation, le / / , devant le jury composé de :

Mr Djari Abdelhamid  
Mr Lemita Abdallah  
Melle Gouder Soraya

MCA  
MAA  
MCB

Président  
Rapporteur  
Examineur 1

Promotion : 2020/2021

## اهداء

الحمد لله الذي اثار لنا درج العلم واماننا على اداء هذا الواجب ووفقنا الى  
انجاز هذا العمل.

مررت قاطرة البحث بكثير من العوائق, ومع ذلك حاولت ان اتخطاها بثبات بفضل من الله  
وما انا ذا اختتم بحثي بكل همّة ونشاط .

الى من افضلها على نفسي ولم لا فلقد ضحت من اجلي , ولم تدخر جهدا في سبيل  
اسعادي على الدوام (امي الحبيبة) .

الى خالد الذكر الذي وفاته المنية وكان خير مثال لرجل الاسرة, والذي لم يتهاون يوم  
في توفير سبيل الخير والسعادة لي (والدي العزيز) رحمك الله يا ابي

الى رباحين حياتي وسبب فرحتي ودعمي في هذه الحياة (دلال, اشواق, ناريمان , نور  
المدى).

وامتن لكل من كان له فضل في مسيرتي, وساعدني ولو باليسير.

الى من احببتهم واحبوني (اسدقائي).

ولا ينبغي ان انسى اساتذتي ممن كان لهم الدور الاكبر في مسانذتي ومدتي  
بالمعلومات القيمة ...

اهدي لكم بحثي تخرجي

دعيا للمولى عز وجل ان يطيل في اعماركم ويرزقكم بالخيرات.

شهباز

## اهداء

\* الى كل من علمني حرفا في هذه الدنيا الفانية.

\* الى سبب وجودي في الحياة والذي ينبض قلبي باسمه

\* وردة قلبي ريحانة حياتي امي وفالاخير لا انسى نصفي الثاني اخوتي

الى من احبتهم واحبوني **(احدائي)**.

تهاني

## *Remerciements*

*Tout d'abord nous remercions le bon dieu puissant pour la bonne santé, la volonté et la patience qu'il nous a donnée tout au long de notre étude, et nous voudrions en premier lieu remercier nos parents.*

*Nous exprimons nos sincères gratitude et remerciements à notre encadreur Dr **LEMITA Abdallah** pour ses conseils précieux, son aide dans le cheminement de cette étude et pour la peine qu' elle s'est donnée tout au long de ce travail afin de faire de ce document ce qu' il représente.*

*Nous tenons également à exprimer une reconnaissance aux membres de jury Dr **DJARI A bdelhamid** et Dr **GOUDER Soraya***

*Un très grand merci pour nous frères, nos sœurs , nos collègues d'études.*

## Liste des symboles

DC/DC : convertisseur continue / continue

K1 : Interrupteur

D : Diode de rou libre

L : Inductance

C : Capacité

R : Résistance

Us : Tension de sortie

Ue : Tension de la source continue

D : Rapport cyclique

Is : Courant de sortie

f : Fréquence de commutation

$\eta$  : Rendement

P : Puissance

Ve : Tension d'entrée

NE555: Circuit Intégrée

MOSFET (IGBT-IRF540N): Metal Oxide Semiconductor Field Effect Transistor

Pic16f877: Micro contrôle

PID: Régulateur Automatique

## Liste des tableaux

Tableau 1. 1 Rendement des convertisseurs DC-DC.....	15
Tableau 2. 1 Bits de sélection des banques.....	23
Tableau 2. 2 Les différentes configurations.....	25
Tableau 4. 1 Valeurs des composants utilisés.....	45
Tableau 4. 2 Comparaison entre les valeurs théoriques et les valeurs de simulation. ...	45
Tableau 4. 3 Comparaison entre les valeurs théoriques et celles de simulation. ....	48
Tableau 4. 4 Effet de chaque paramètre d'un régulateur PID. ....	50

## Liste des figures

Figure 1. 1 convertisseur hacheur. ....	4
Figure 1. 2 (a) et (b) Caractéristique d'une diode et d'une diode + IGBT.....	5
Figure 1. 3 Schéma d'un hacheur série. ....	6
Figure 1. 4 Schéma d'un hacheur série: $0 < t < \alpha T$ . ....	7
Figure 1. 5 Schéma d'un hacheur série: $\alpha T < t < T$ . ....	8
Figure 1. 6 Chronogramme du convertisseur abaisseur (Buck).....	9
Figure 1. 7 Schéma d'un hacheur boost .....	10
Figure 1. 8 Schéma d'un hacheur parallèle: $0 < t < \alpha T$ . ....	10
Figure 1. 9 Schéma d'un hacheur parallèle: $0 < t < \alpha T$ . ....	11
Figure 1. 10 Chronogrammes du convertisseur boost. ....	12
Figure 1. 11 Schéma d'un hacheur Buck-boost.....	13
Figure 1. 12 Schéma d'un hacheur série-parallèle: $0 < t < \alpha T$ . ....	13
Figure 1. 13 Schéma d'un hacheur série-parallèle: $\alpha T < t < T$ .....	14
Figure 2. 1 Logiciel Isis Proteus. ....	18
Figure 2. 2 Architecture interne du PIC 16F877.....	19
Figure 2. 3 L'organisation de la mémoire RAM. ....	21
Figure 3. 1 Logiciel Isis Proteus. ....	29
Figure 3. 2 Présentation de la page d'Isis. ....	30
Figure 4. 1 Schéma d'un convertisseur Boost. ....	41
Figure 4. 2 Schéma équivalent d'un convertisseur Boost quand $0 < t < \alpha T_{sw}$ .....	42
Figure 4. 3 Schéma équivalent d'un convertisseur Boost quand $\alpha T_{sw} < t < T_{sw}$ .....	43
Figure 4. 4 hacheur Boost en boucle ouverte.....	44
Figure 4. 5 Brochage du circuit intégré NE 555. ....	46
Figure 4. 6 Signal de sortie du circuit NE 555.....	47
Figure 4. 7 Schémas d'un convertisseur Boost avec NE555.....	47
Figure 4. 8 Schéma d'un convertisseur Boost avec un microcontrôleur ( $\alpha=75\%$ ).....	48
Figure 4. 9 Schéma équivalent d'un régulateur PID. ....	50
Figure 4. 10 schéma d'un convertisseur DC/DC par un régulateur PID avec le pic 16f877a.....	51

## ملخص

محول boost هو محول كهربائي (مستمر/مستمر) , فهو يحول توتر كهربائي غير منظم الى توتر كهربائي منظم او متغير يطلق عليه محول boost لان توتر المخرج اكبر من توتر المدخل .انه يعمل بشكل اساسي عن طريق فتح واغلاق مفتاح الكتروني بشكل دوري . بعد ذلك , تحتاج دائرة الكهربائية الى دائرة الكترونية تنتج او تولد اشارة PWM لمفتاح الطاقة يمكن برمجة الميكرو كونترولر لإنتاج اشارة PWM .

يتم التحكم في مخرج المحول BOOST باستخدام وحدة التحكم PID.تمت برمجة PID لقياس الخطأ بين توتر المخرج للمحول BOOST التوتر المطلوب لاجراء تصحيح لاننتاج اشارة PWM مناسبة.

### الكلمات المفتاحية :

محول DC\_DC محول دفعة , وحدة التحكم PID , اشارة PWM ,مفتاح الكتروني.

## **Résumé:**

Le convertisseur Boost est un convertisseur DC-DC qui convertit une tension continue qui n'est pas réglable en tension continue réglable à une valeur désirée. Ce convertisseur est nommé " Boost " parce que la tension de sortie du convertisseur est supérieure à celle d'entrée. Il est principalement basé sur l'ouverture et la fermeture d'une manière périodique d'un interrupteur électronique. Alors, le convertisseur Boost a besoin d'un circuit électronique capable de produire ou générer un signal PWM (Pulse Width Modulation) ou MLI (modulation de largeur d'impulsions) pour commander l'interrupteur électronique. Le microcontrôleur peut être utilisé pour générer un signal PWM.

La sortie du convertisseur Boost est contrôlée par un régulateur PID. Le régulateur PID est programmé pour mesurer l'erreur entre la tension de sortie du convertisseur Boost et la tension souhaitée (tension de référence) pour faire une correction afin de produire un signal PWM adéquat.

## **Keywords:**

Convertisseur DC-DC, Convertisseur Boost, régulateur PID, signal PWM, interrupteur électronique.

**Abstract:**

Boost converter is a DC-DC converter; it converts an unregulated input DC voltage to a regulated or variable DC output voltage. It is called a Boost converter because the output voltage is larger than the input voltage. It basically operates by opening and closing an electronic switch periodically. Then, the boost circuit needs a electronic circuit that produces or generates a PWM signal (Pulse Width Modulation) for the power switch. The microcontroller can be programmed to produce the PWM signal.

The output of the Boost converter is controlled by using the PID controller. The PID is programmed to measure the error between the output voltage of the Boost converter and the desired voltage to make a correction to produce suitable PWM signal.

**Keywords:**

DC-DC converter, Boost converter, PID controller, PWM signal, electronic switch.

## Table des matières

Remerciements	
Listes des symboles.....	<b>i</b>
Listes des tableaux .....	<b>ii</b>
Listes des figures.....	<b>iii</b>
ملخص.....	<b>iv</b>
Résumé:.....	<b>vi</b>
Abstract: .....	<b>vii</b>
Introduction générale .....	<b>1</b>
<b>CHAPITRE I : CONVERTISSEUR DC-DC .....</b>	<b>3</b>
1. Introduction.....	<b>4</b>
2. Le convertisseur dc-dc (hacheur).....	<b>4</b>
2.1. Définition d'un hacheur : .....	<b>4</b>
3. Les différents éléments d'un hacheur .....	<b>4</b>
4. Les semi-conducteurs disponibles comme fonction interrupteur .....	<b>5</b>
5. Les différents types des hacheurs .....	<b>6</b>
5.1. Hacheur série (Buck).....	<b>6</b>
5.1.2. Applications d'un hacheur Buck .....	<b>9</b>
5.2. Hacheur parallèle (Boost).....	<b>9</b>
5.2. Applications de l'hacheur boost .....	<b>11</b>
5.3. Hacheur série-parallèle (Buck-Boost).....	<b>12</b>
6. Rendement du convertisseur statique.....	<b>15</b>
7. Conclusion: .....	<b>15</b>
<b>CHAPITRE II : LE MICROCONTROLEUR PIC 16F877 .....</b>	<b>16</b>
1. Introduction.....	<b>17</b>
2. Architecture externe du microcontrôleur PIC 16F877.....	<b>17</b>
3. Architecture interne du microcontrôleur PIC 16F877 .....	<b>19</b>
4. Mémoires du microcontrôleur PIC 16F877 .....	<b>20</b>
4.1. Mémoire de données RAM .....	<b>20</b>
4.2. Mémoire de programme EEPROM.....	<b>22</b>
4.3. Mémoire de donnée EEPROM.....	<b>22</b>
5. Explication de certains registres spéciaux .....	<b>22</b>
5.1. Registre STATUS .....	<b>22</b>
5.2. Registre TRISA .....	<b>23</b>
5.3. Registre PORTA .....	<b>23</b>
5.4. Présentation des registres du module CAN.....	<b>23</b>

5.4.1. Register ADCON0 (A/D control register 0).....	24
5.4.2. Register ADCON1 (A/D control registers 1) .....	24
5.4.2. Registres ADRESH et ADRESL.....	25
5.5. Présentation du module PWM.....	25
5.5.1. Registre CCP1CON.....	26
6. Conclusion .....	<b>26</b>
<b>CHAPITRE III : ISIS _PROTEUS .....</b>	<b>27</b>
1.Introduction.....	<b>28</b>
2. Définition de logiciel ISIS .....	<b>28</b>
2.1. Isis intelligent schématique input system.....	29
2.2. Prise en main du logiciel Isis.....	29
2.3. L'environnement de travail de Proteus .....	32
2.3.1. La barre d'outils principale .....	32
2.3.2. Zone de travail .....	34
2.3.3. Sélection d'objet.....	34
2.3.4 Gestion d'un projet.....	34
2.4. Simulation .....	35
2.4.1. Placement d'un graphe de simulation .....	35
2.4.2. Position et dimension d'un graphe .....	35
2.4.3. Export des graphes.....	35
3. Configuration de l'application D'ISIS PROTEUS V7 .....	<b>36</b>
4. Description de fonctionnement de logiciel ISIS .....	<b>37</b>
5. Présentation de l'interface ARES .....	<b>37</b>
6. Conclusion .....	<b>39</b>
<b>CHAPITRE IV : COMMANDE D'UN CONVERTISSEUR DC-DC BOOST.....</b>	<b>40</b>
1. Introduction :.....	<b>41</b>
2. Convertisseurs dc/dc boost .....	<b>41</b>
2.1. Principe.....	41
2.3. Analyse du fonctionnement.....	42
2.4 Relation entre la tension d'entrée et la tension de sortie .....	43
3. Commande d'un convertisseur Boost en boucle ouverte .....	<b>44</b>
4. Commande d'un convertisseur Boost par le circuit NE555 .....	<b>45</b>
4.1. Présentations du circuit NE555 .....	46
5. Simulation d'un convertisseur Boost avec un microcontrôleur PIC 16F877A .....	<b>48</b>
6. Commande d'un convertisseur Boost par un régulateur PID.....	<b>49</b>
6.1. Le régulateur PID :.....	49

6.2. Commande d'un convertisseur DC /DC par un régulateur PID avec le pic 16f877a.....	51
7. Conclusion .....	54
<b>Conclusion général.....</b>	<b>55</b>
<b>BIBLIOGRAPHIE.....</b>	<b>56</b>

## Introduction générale

Les convertisseurs de puissance sont des dispositifs électroniques capables de modifier la tension et/ou la fréquence de l'onde électrique. Puisque on a deux types de l'électricité (continue et alternative), on peut distinguer quatre types des convertisseurs de puissance:

- \* Le redresseur: est un convertisseur alternatif-continu (AC-DC), il peut être utilisé pour charger les batteries, alimenter les moteurs à courant continu, etc.
- \* Le hacheur: est un convertisseur continu-continu (DC-DC), il peut être utilisé pour commander les moteurs à courant continu (vitesse variable), il est aussi utilisé dans l'alimentation à découpage, etc [1].
- \* Le gradateur: est un convertisseur alternatif-alternatif (AC-AC), il est utilisé dans la production des vitesses variables en alternatif, etc.
- \* L'onduleur: est un convertisseur continu-alternatif (DC-AC), il peut être utilisé dans l'alimentation des appareils électriques autonomes, la protection contre les coupures de réseau, la commande des machines à courant alternatif, etc.

Ce travail s'intéresse aux convertisseurs DC-DC. Ils permettent de modifier la valeur de la tension d'une source de tension continue non réglable en tension continue d'une valeur moyenne réglable. Le principe consiste à hacher ou bien découper la source de la tension continue à l'entrée du convertisseur DC-DC avec une fréquence élevée via des interrupteurs électroniques commandés (transistor MOSFET, thyristors, etc.)[2].

Il existe trois types de convertisseurs DC-DC:

- \* Un hacheur série ou dévolteur (convertisseur Buck).
- \* Un hacheur parallèle ou survolteur (convertisseur Boost).
- \* Un hacheur série-parallèle ou dévolteur-survolteur (convertisseur Buck-Boost).

Les convertisseurs DC-DC sont commandés (commande de l'ouverture et la fermeture de l'interrupteur) via un signal PWM (Pulse Width Modulation, ou MLI en français (modulation de largeur d'impulsions). Parmi les circuits électroniques permettant de générer ce signal PWM, il y a les microcontrôleurs.

Notre travail consiste à commander un convertisseur DC-DC (Boost) par un régulateur PID via un microcontrôleur.

Ce mémoire sera organisé de la manière suivante:

\* Dans le premier chapitre, on va présenter les différents types de convertisseurs DC-DC (convertisseur Buck, Boost et Buck-Boost) et les caractéristiques électriques de chaque type.

\* Le deuxième chapitre sera consacré à l'étude du microcontrôleur PIC 16F877A.

\* Le troisième chapitre présentera: un logiciel de simulation très utilisé en électronique nommé ISIS-PROTEUS et un compilateur nommé mikroC. Ce compilateur sert à programmer le microcontrôleur PIC 16F877 en langage C++ [3].

\* Dans le dernier chapitre, on réalisera un montage électronique d'un convertisseur Boost à l'aide de logiciel de simulation ISIS PROTEUS. Le travail sera divisé en trois étapes: premièrement, on utilise un circuit intégré qui se nomme NE555 afin de générer un signal PWM pour **commander** le convertisseur Boost (commande en boucle ouverte). Deuxièmement, on va utiliser un microcontrôleur PIC 16F877A pour générer un signal ce commande PWM (toujours en boucle ouverte). Troisièmement, la commande du convertisseur Boost sera en boucle fermée en utilisant le régulateur PID. Le régulateur PID mesure l'erreur entre la tension de sortie du convertisseur Boost et la tension souhaitée (tension de référence) pour produire un signal PWM adéquat.

# **CHAPITRE I : CONVERTISSEUR DC-DC**

## 1. Introduction

Ce chapitre est consacré à l'étude des différents types des convertisseurs DC-DC. Les convertisseurs dc-dc en général réalisent deux fonctions : modifier le niveau de tension (élever ou abaisser) et réguler la tension.

On peut distinguer deux types de convertisseurs dc-dc:

- Les convertisseurs appelés hacheurs qu'ils soient abaisseurs (Buck), élévateur (Boost), ou bien abaisseur-élévateur (Buck-Boost).
- Les alimentations à découpages qui correspondent aux applications petites puissances de la conversion DC-DC.

## 2. Le convertisseur dc-dc (hacheur)

### 2.1. Définition d'un hacheur :

Un hacheur est un convertisseur de puissance dc-dc qui permet de modifier la valeur de la tension d'une source de tension continue en tension continue d'une valeur moyenne réglable. Le principe consiste à découper la source de la tension continue avec une fréquence élevée via des interrupteurs commandés. Dans le cas idéal, les interrupteurs commandés ne consomment aucune puissance active, c'est la raison pour laquelle les hacheurs ont un bon rendement [4].

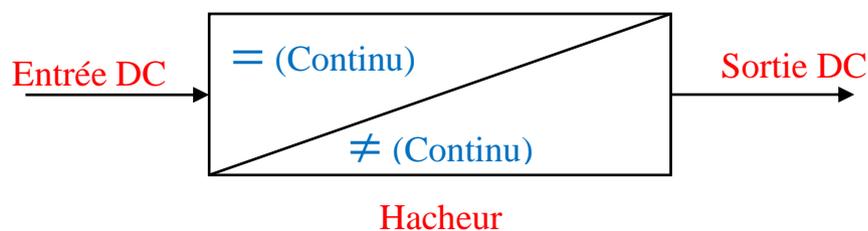


Figure 1.1 convertisseur hacheur.

## 3. Les différents éléments d'un hacheur

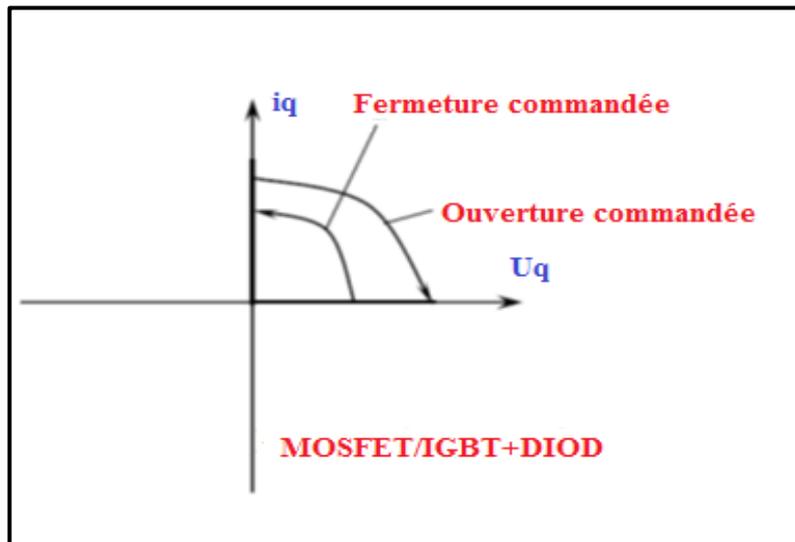
Un hacheur est généralement constitué des composants suivants:

- Des diodes.
- Des transistors.
- Des thyristors.
- Source de tension continue telle que: des batteries, des générateurs de tensions continues etc.
- Inductances.

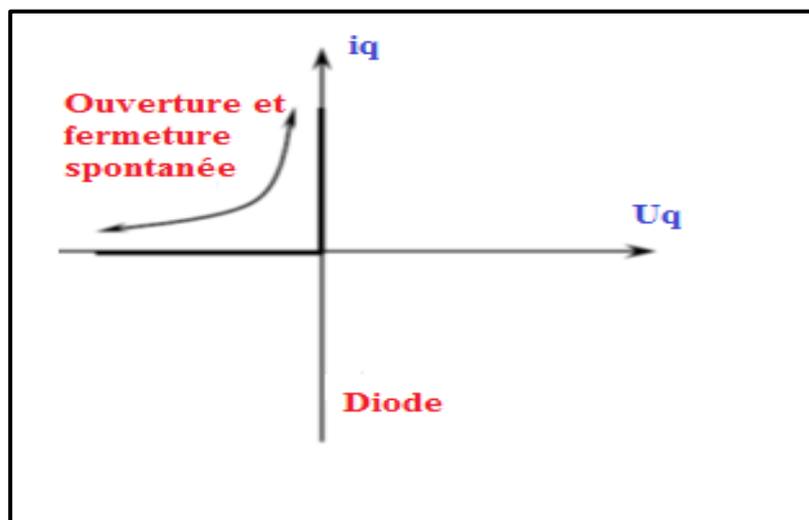
- Capacités.
- Résistances

#### 4. Les semi-conducteurs disponibles comme fonction interrupteur

Les deux types de semi-conducteurs les plus utilisés dans les hacheurs sont la diode et le transistor MOSFET/IGBT associé à une diode de conduction dont les caractéristiques sont représentées:



(a)



(b)

Figure 1.2 (a) et (b) Caractéristique d'une diode et d'une diode + IGBT.

#### 5. Les différents types des hacheurs

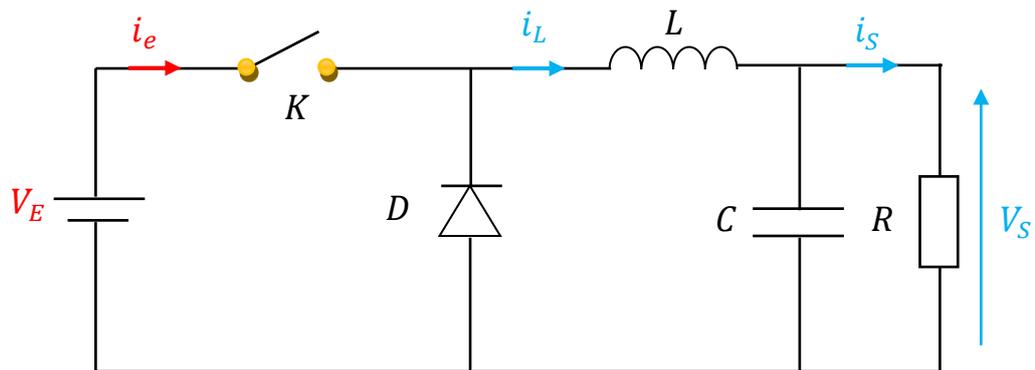
Il existe trois types de convertisseurs dc-dc:

- Un hacheur série ou dévolteur (Buck).

- Un hacheur parallèle ou survolteur (Boost).
- Un hacheur série-parallèle ou dévolteur-survolteur (Buck-Boost).

### 5.1. Hacheur série (Buck)

Un convertisseur Buck est un convertisseur dc-dc qui convertit une tension continue en une autre tension continue d'une valeur plus faible que celle d'entrée. Le schéma générale d'un hacheur série est donné dans la figure ci-dessous.



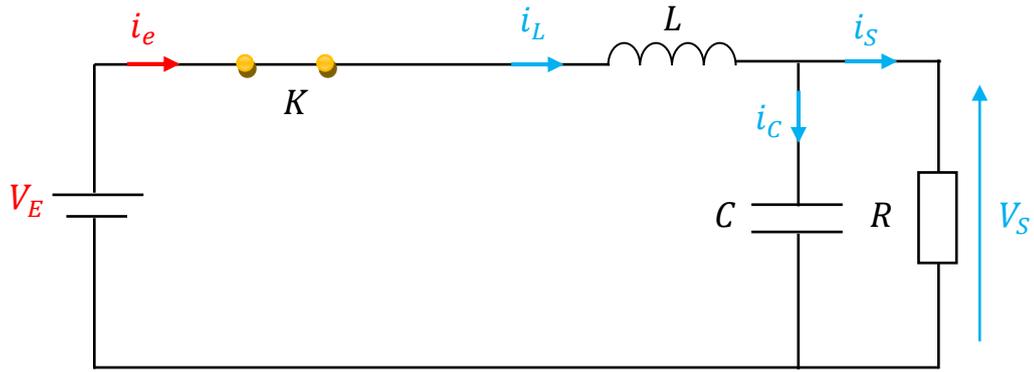
**Figure 1.3** Schéma d'un hacheur série.

L'interrupteur  $K$  peut être remplacé par un transistor mosfet, thyristor etc, puisque le courant est toujours positif et que les commutations doivent être commandées (au blocage et à l'amorçage). C'est un convertisseur qui donne en sortie une tension plus petite que celle de l'entrée [5].

La charge est constituée par la résistance  $R$ . Les éléments  $L$  et  $C$  forment un filtre dont le but est de limiter l'ondulation résultante du découpage sur la tension et le courant de sortie.

Le fonctionnement du circuit hacheur Buck peut se diviser en deux configurations selon l'état de l'interrupteur  $K$ : l'état passant quand l'interrupteur est fermé et l'état bloqué quand l'interrupteur est ouvert.

\* Pour  $0 < t < \alpha T$ , l'interrupteur  $K$  est fermé, un courant circule dans le circuit, mais aucun courant ne passe pas dans la diode  $D$  puisqu'elle est polarisée en inverse. Le courant  $i_L$  augmente linéairement avec un taux d'accroissement imposé par l'inductance  $L$  [6,7]



**Figure 1.4** Schéma d'un hacheur série:  $0 < t < \alpha T$ .

En appliquant la loi de Kirchhoff au circuit, on obtient les équations suivantes:

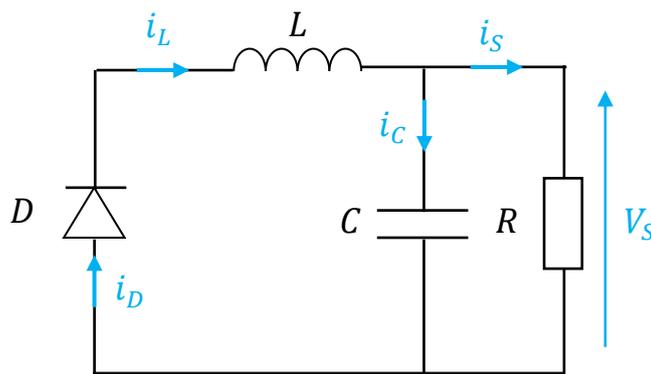
$$V_E = V_L + V_S = L \frac{di_L}{dt} + V_S \tag{1.1}$$

$$L \frac{di}{dt} = V_E - V_S \tag{1.2}$$

$$i_L = i_C + i_S = C \frac{dV_C}{dt} + \frac{V_S}{R} \tag{1.3}$$

$$C \frac{dV_C}{dt} = i_L - \frac{V_S}{R} \tag{1.4}$$

\* Pour  $\alpha T < t < T$ , l'interrupteur  $K$  est ouvert. Durant cet intervalle le courant de l'inductance  $i_L$  diminue (l'énergie accumulée dans l'inductance  $L$  est transférée à la capacité et la charge).



**Figure 1.5** Schéma d'un hacheur série:  $\alpha T < t < T$ .

En appliquant la loi de Kirchhoff, on obtient les équations suivantes:

$$L \frac{di_L}{dt} = -V_S \tag{1.5}$$

$$C \frac{dV_C}{dt} = i_L - \frac{V_S}{R} \quad (1.6)$$

On peut obtenir un seul modèle rassemblant les deux cas (l'interrupteur  $K$  est fermé et ouvert):

$$L \frac{di_L}{dt} = KV_E - V_S \quad (1.7)$$

$$C \frac{dV_C}{dt} = i_L - \frac{V_S}{R} \quad (1.8)$$

Le modèle mathématique du convertisseur Buck est décrit par les équations suivantes:

$$\begin{cases} \frac{di_L}{dt} = K \frac{1}{L} V_E - \frac{1}{L} V_S \\ \frac{dV_C}{dt} = \frac{1}{C} i_L - \frac{V_S}{RC} \end{cases} \quad (1.9)$$

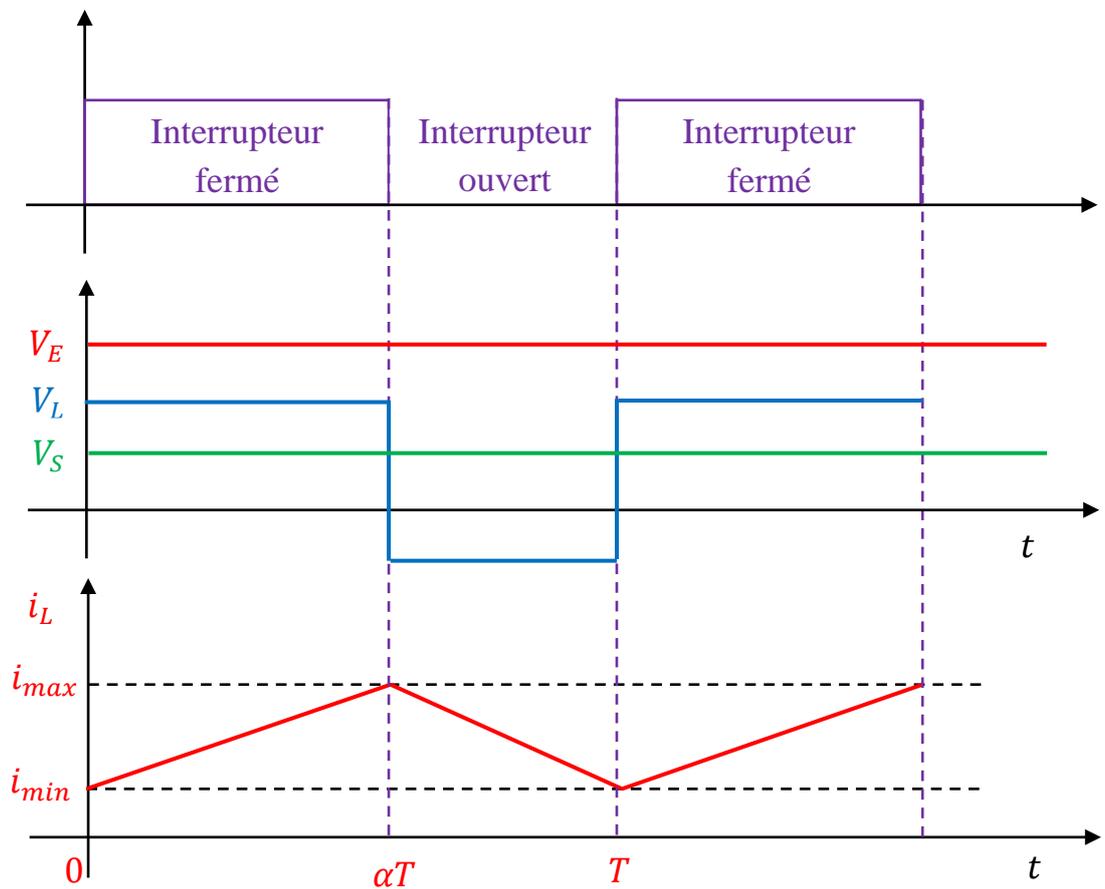


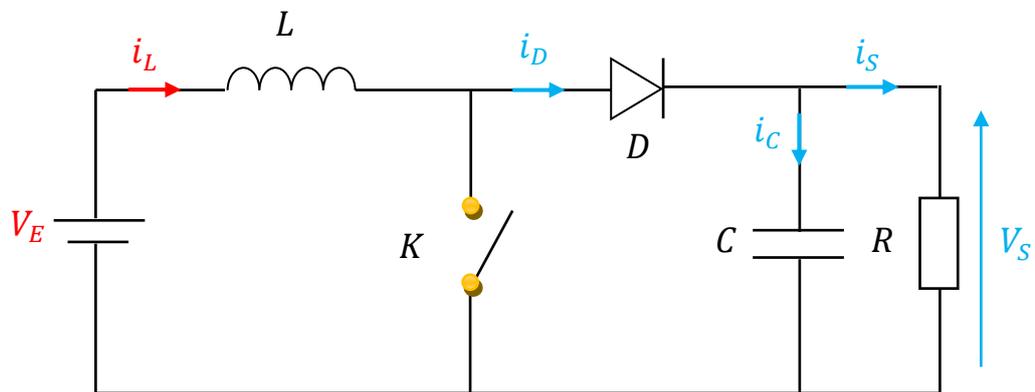
Figure 1.6 Chronogramme du convertisseur abaisseur (Buck) [8].

### 5.1.2. Applications d'un hacheur Buck

La façon la plus simple de réduire une tension est d'utiliser un diviseur de tension, mais les diviseurs de tension possèdent plusieurs inconvénients: dissipation sous forme de chaleur, un faible rendement ce qui est rédhibitoire pour les applications d'électronique de puissance. un convertisseur Buck possède un fort rendement (jusqu'à 95%), et offre la possibilité de réguler la tension de sortie plus adaptée pour les applications d'électronique de puissance comme la réduction des 12-24 v fournis par une batterie d'ordinateur portable vers les quelques volts nécessaires au processeur.

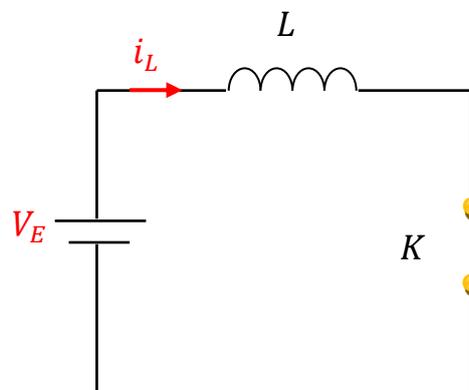
## 5.2. Hacheur parallèle (Boost)

Un convertisseur boost ou hacheur parallèle est une alimentation à découpage qui convertit une tension continue en une autre tension continue d'une valeur plus forte que celle de l'entrée [5]. Le schéma général d'un hacheur parallèle est donné dans la figure ci-dessous.



**Figure 1.7** Schéma d'un hacheur boost.

\* Pour  $0 < t < \alpha T$ , l'interrupteur est fermé ( $k = 1$ ) où le courant  $i_L$  est à l'instant initial. pendant cet intervalle le courant traversant l'inductance augmente.



**Figure 1.8** Schéma d'un hacheur parallèle:  $0 < t < \alpha T$ .

En appliquant la loi de Kirchhoff, on obtient les équations suivantes:

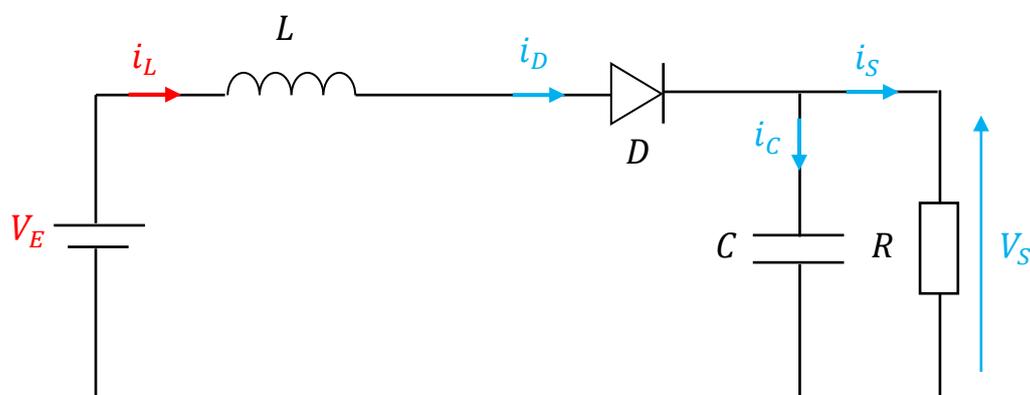
$$L \frac{di_L}{dt} = V_E \tag{1.10}$$

$$C \frac{dV_C}{dt} = \frac{V_S}{R} \tag{1.11}$$

\* Pour  $\alpha T < t < \alpha$ , l'interrupteur est ouvert ( $K = 0$ ), et le courant de l'inductance diminue car l'énergie accumulée dans l'inductance  $L$  sera transférée à la capacité et la charge. En appliquant la loi de Kirchhoff, on obtient les équations suivantes:

$$L \frac{di_L}{dt} = V_E - V_S \tag{1.12}$$

$$C \frac{dV_C}{dt} = i_L - \frac{V_S}{R} \tag{1.13}$$



**Figure 1.9** Schéma d'un hacheur parallèle:  $\alpha T < t < T$ .

On peut rassembler les deux cas en un seul modèle:

$$L \frac{di_L}{dt} = V_E - (1 - K)V_S \tag{1.14}$$

$$C \frac{dV_C}{dt} = (1 - K)i_L - \frac{V_S}{R} \tag{1.15}$$

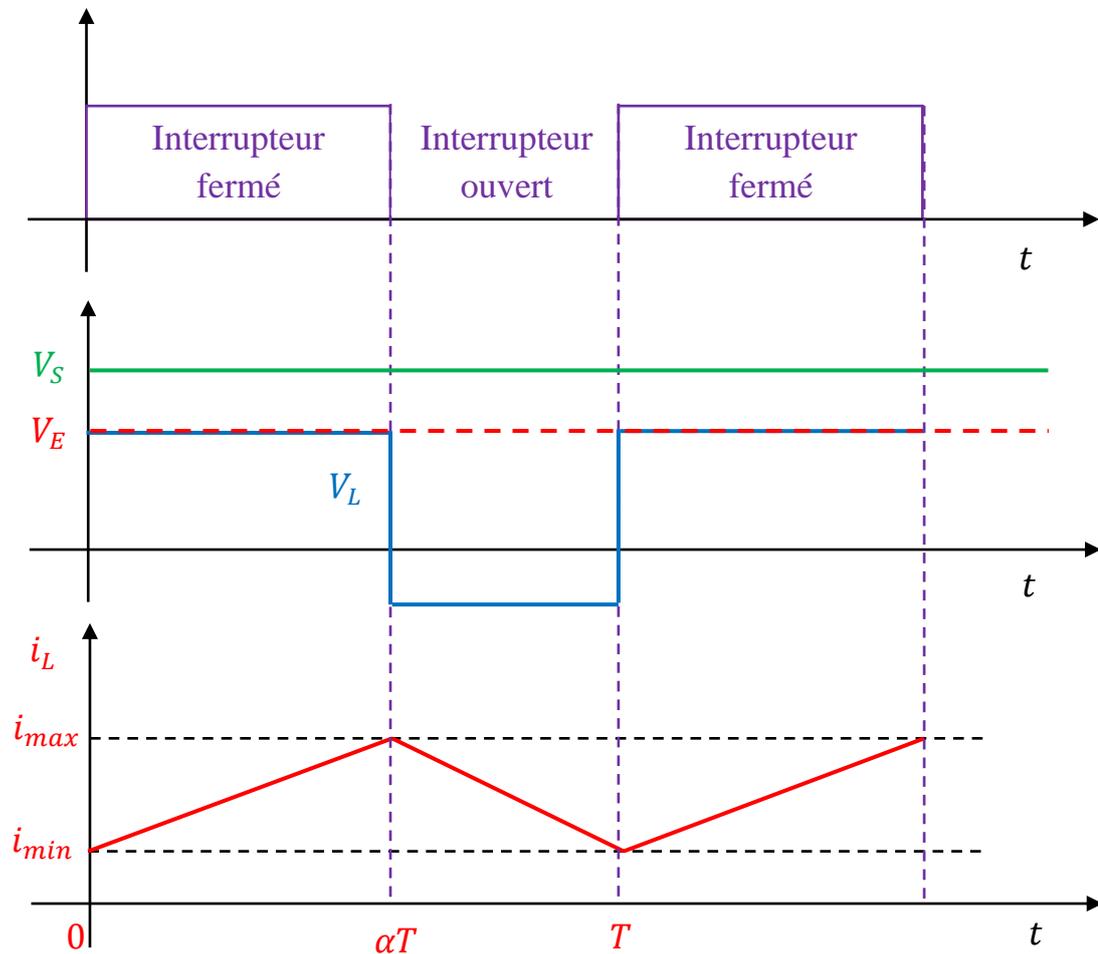
Le modèle mathématique du convertisseur boost est décrit par les équations suivantes:

$$\begin{cases} \frac{di_L}{dt} = \frac{1}{L}V_E - \frac{1}{L}(1 - K)V_S \\ \frac{dV_C}{dt} = \frac{1}{C}(1 - K)i_L - \frac{V_S}{RC} \end{cases} \tag{1.16}$$

## 5.2. Applications de l'hacheur boost

Le convertisseur boost est utilisé lorsqu'on veut augmenter la tension disponible d'une source continue. Les systèmes alimentés par les batteries d'accumulateurs utilisent souvent plusieurs accumulateurs en série afin de disposer d'un niveau de tension suffisamment élevé. La place disponible étant souvent limitée, il n'est pas

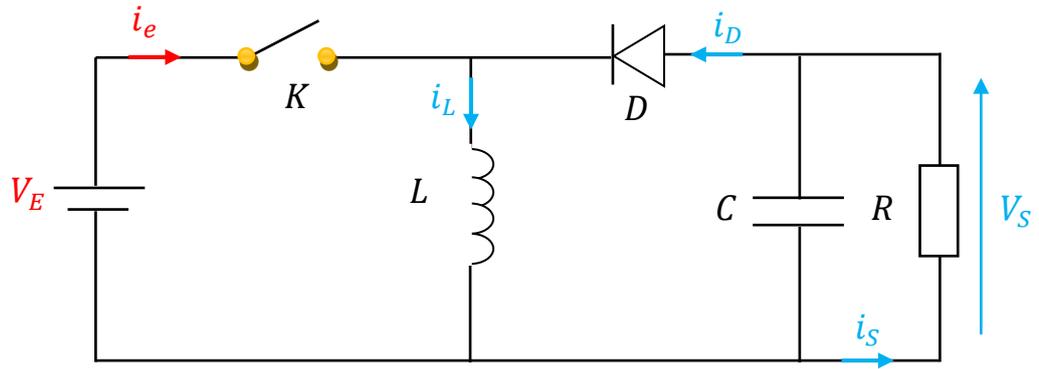
toujours possible de disposer d'un nombre suffisant d'éléments. Un convertisseur boost permet d'augmenter la tension fournie par les batteries et ainsi diminuer le nombre d'éléments nécessaires pour atteindre le niveau de tension désiré.



**Figure 1.10** Chronogrammes du convertisseur boost.

### 5.3. Hacheur série-parallèle (Buck-Boost)

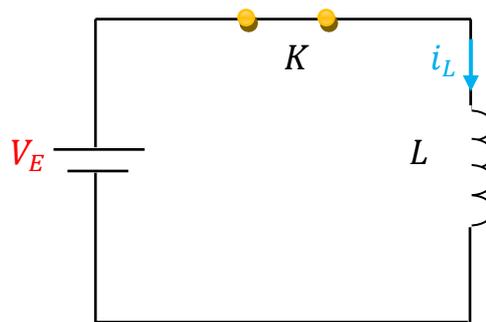
Un convertisseur Buck-boost est un convertisseur de puissance qui fournit une tension plus faible, ou plus élevée que celle appliquée en son entrée. Un inconvénient de ce convertisseur est que son interrupteur ne possède pas de borne reliée au zéro, compliquant ainsi sa commande. Le schéma général d'un hacheur série-parallèle est donné dans la figure ci-dessous.



**Figure 1.11** Schéma d'un hacheur Buck-boost.

Le fonctionnement d'un convertisseur Buck-Boost peut être divisé en deux configurations suivant l'état de l'interrupteur  $K$ :

\* Pour  $0 < t < \alpha T$ , l'interrupteur est fermé ( $k = 1$ ) et durant cette période, le courant  $i_L$  qui traverse l'inductance  $L$  augmente.



**Figure 1.12** Schéma d'un hacheur série-parallèle:  $0 < t < \alpha T$ .

En appliquant la loi de Kirchhoff, on obtient les équations suivantes:

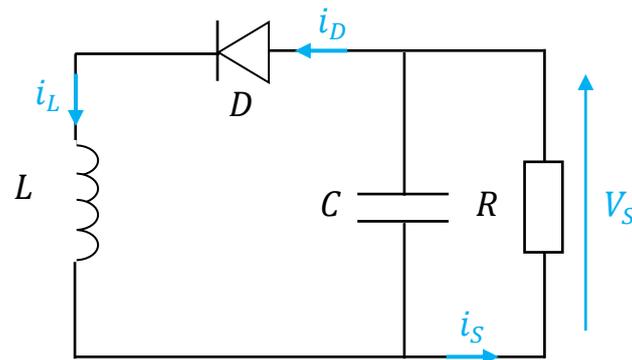
$$L \frac{di_L}{dt} = V_E \tag{1.17}$$

$$C \frac{dV_C}{dt} = -\frac{V_S}{R} \tag{1.18}$$

\* Pour  $\alpha T < t < T$ , l'interrupteur est ouvert ( $K = 0$ ), l'inductance  $L$  est reliée à la charge et à la capacité. Il en résulte un transfert de l'énergie accumulée dans l'inductance  $L$  vers la capacité  $C$  et la charge  $R$ .

En comparant le convertisseur Buck-Boost aux convertisseurs Buck et boost, les principales différences sont:

- La tension de sortie est de polarité inverse par rapport à celle d'entrée.



**Figure 1.13** Schéma d'un hacheur série-parallèle:  $\alpha T < t < T$

En appliquant la loi de Kirchhoff, on obtient les équations suivantes:

$$C \frac{dV_C}{dt} = -i_L - \frac{V_S}{R} \quad (1.19)$$

$$C \frac{dV_C}{dt} = -i_L - \frac{V_S}{R} \quad (1.20)$$

Le modèle rassemblant les deux cas:

$$L \frac{di_L}{dt} = KV_E + (1 - K)V_S \quad (1.21)$$

$$C \frac{dV_C}{dt} = -(1 - K)i_L - \frac{V_S}{R} \quad (1.22)$$

Le modèle mathématique du convertisseur Buck-Boost est décrit par les équations suivantes:

$$\begin{cases} \frac{di_L}{dt} = K \frac{1}{L} V_E + \frac{1}{L} (1 - K) V_S \\ \frac{dV_C}{dt} = -\frac{1}{C} (1 - K) i_L - \frac{V_S}{RC} \end{cases} \quad (1.23)$$

Pour les convertisseurs Buck et Boost, le convertisseur Buck-Boost est caractérisé par trois régimes de fonctionnement distincts: conduction continue dans lequel le courant  $i_L$  ne s'annule pas, et une conduction critique où le courant s'annule ponctuellement lorsque, et une conduction discontinue dans lequel le courant  $i_L$  prend une valeur nulle, pendant une durée non nulle.

## 6. Rendement du convertisseur statique

Les trois convertisseurs cités précédemment présentent un meilleur rendement. En effet, il existe d'autres configurations de convertisseur, mais dont le rendement est plus faible.

Le tableau ci-dessous donne le rendement de quelques convertisseurs connus.

L'expression du rendement du convertisseur est donné par:  $\eta = \frac{P_s}{P_E}$ .

Convertisseur DC-DC	Rendement de conversion
Buck	93 %
Boost	92 %
buck-boost	92 %

**Tableau 1.1:** Rendement des convertisseurs DC-DC.

## 7. Conclusion:

Dans ce chapitre on a présenté les trois types de convertisseurs de puissance dc-dc qui sont: l'hacheur série (appelé aussi convertisseur Buck) qui est un abaisseur de tension, l'hacheur parallèle (appelé aussi convertisseur Boost) qui est un élévateur de tension et le dernier est l'hacheur série-parallèle (convertisseur Buck-Boost) qui est un abaisseur-élévateur de tension. Le principe de fonctionnement ainsi que les caractéristiques électriques de chaque type de convertisseur a été aussi décrite.

**CHAPITRE II : LE MICROCONTROLEUR**  
**PIC 16F877**

## 1. Introduction

Un microcontrôleur est un circuit intégré rassemblant dans un même boîtier tous les éléments d'une structure à base de microprocesseur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques (ports, timers, convertisseur) et interfaces d'entrées-sorties. Un microcontrôleur peut fonctionner de façon autonome après avoir été programmé.

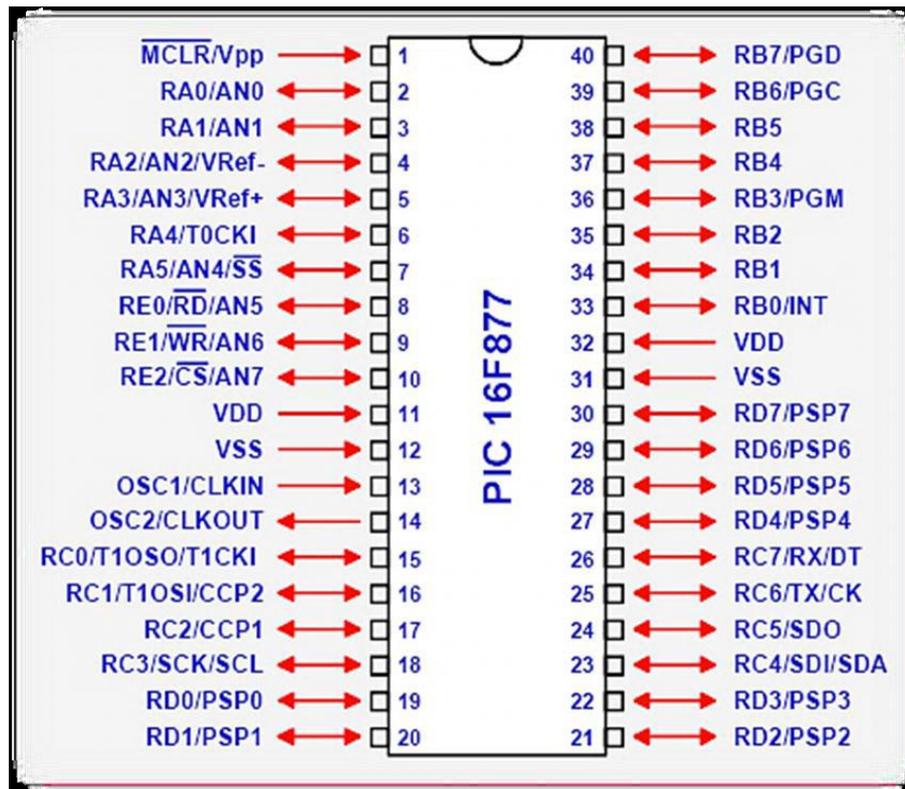
Les microcontrôleurs PIC (Périphérie Interface Controller) peuvent être divisés en trois grandes familles:

- La famille Base-line qui utilise des mots d'instructions de 12 bits.
- La famille Mid-Range qui utilise des mots de 14 bits.
- La famille High-End qui utilise des mots de 16 bits.

Dans ce chapitre, on va étudier le microcontrôleur PIC16F877 en présentant: son architecture externe et interne, l'organisation de sa mémoire (mémoire de données RAM, mémoire de programme EEROPM et mémoire de données EEPROM), et finalement, on donnera certains registres spéciaux se trouvant dans la mémoire RAM [9].

## 2. Architecture externe du microcontrôleur PIC 16F877

Le microcontrôleur PIC 16F877 est un circuit intégré qui fait partie de la famille Mid-Range qui utilise des mots d'instructions de 14 bits [10,11]. Il comprend 40 broches comme il est montré dans la figure ci-dessous:



**Figure 2.1** Brochage du PIC16F877A.

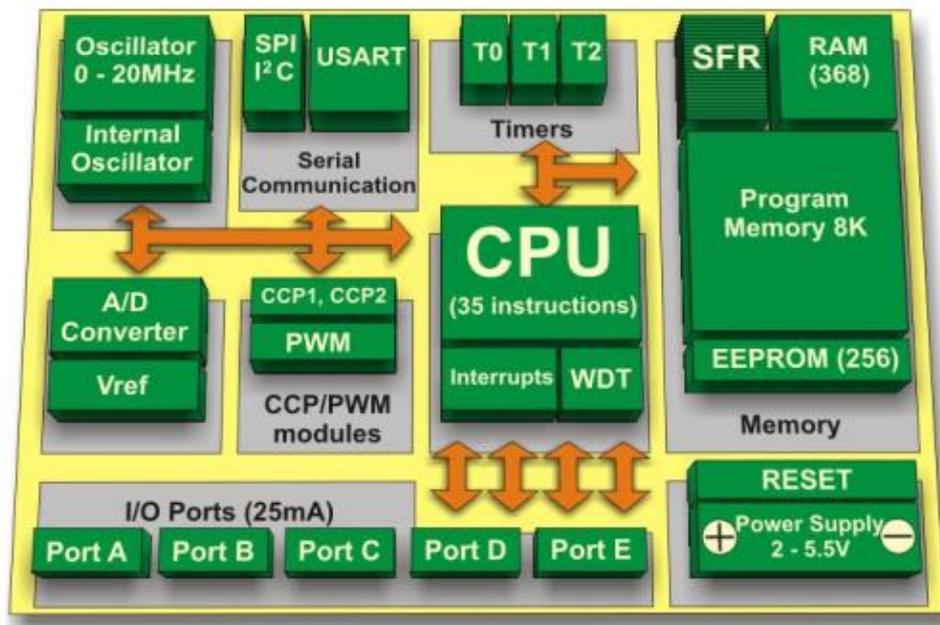
**Description des différentes broches:**

- Les broches VDD et VSS servent à alimenter le microcontrôleur sous une tension d'alimentation entre 2 et 5.5 V [12].
- La broche  $\overline{MCLR}$  sert à réinitialiser le microcontrôleur.
- Les broches OSC1 et OSC2 permettent de faire fonctionner l'oscillateur interne du microcontrôleur qui peut être un quartz (jusqu' à 20 MHz) ou un oscillateur externe RC.
- Cinq ports d'entrée sortie bidirectionnels:
  - PORTA: le port A possède 6 d'entrées/sorties de RA0 jusqu'à RA5.
  - PORTB: le port B possède 8 d'entrées/sorties de RB0 jusqu'à RB7.
  - PORTC: le port C possède 8 d'entrées/sorties de RC0 jusqu'à RC7.
  - PORTD: le port D possède 8 d'entrées/sorties de RD0 jusqu'à RD7.
  - PORTE: le port E ne possède que 3 d'entrées/sorties de RE0 jusqu'à RE2.
- Les broches CCP1 et CCP2 présentent les deux modules de comparaison et Capture avec PWM (modulation de largeur d'impulsions).
- Les broches AN0 jusqu'à AN7 sont les entrées des convertisseurs analogiques/numériques (CAN à 10 bits).

- Les broches RX et TX présentent l'interface de communication série asynchrone et synchrone USART.
- Les broches SDO, SDI et SS présentent l'interface de communication série synchrone SPI.
- Les broches SCL et SDA l'interface de communication série synchrone I2C.

### 3. Architecture interne du microcontrôleur PIC 16F877

L'architecture interne du PIC 16F877 est donnée dans la figure ci-dessous [13].



**Figure 2.2** Architecture interne du PIC 16F877.

D'après ce schéma architectural, on peut citer les éléments constitutifs du microcontrôleur PIC 16F877:

- Mémoire de données RAM de 368 Octets.
- Mémoire de programme EEPROM (type Flash) de 8 K mots de 14 bits.
- Mémoire de données EEPROM de 256 Octets.
- Quatre sources interruptions.
- Générateur d'horloge de type quartz ou RC (fréquence de l'oscillateur maximale 20 MHz).
- Cinq ports d'entrée/sortie pouvant produire 25 mA par sortie (PORTA: 6 bits, PORTB, PORTC et PORTD: 8bits et PORTE: 3 bits).
- Chien de garde (Watch dog).
- TIMER0: Timer/Compteur 8 bits avec un pré-diviseur 8 bits.
- TIMER1: Timer/Compteur 16 bits avec un pré-division de 1, 2, 4, ou 8.

- TIMER2: Timer 8 bits avec deux diviseurs.
- Deux modules de: comparaison, capture et PWM:
- Module capture 16 bits avec une résolution de 12,5 ns.
- Module Compare 16 bits avec une résolution de 200 ns.
- Module PWM avec une résolution de 10 bits.
- Convertisseur analogique 10 bits à 8 entrées.
- Interface de communication série:
- USART: Universel Synchronous Asynchronous Receiver Transmitter.
- SPI: Serial Peripheral Interface.
- I2C: Inter-Integrated Circuit.

#### **4. Mémoires du microcontrôleur PIC 16F877**

Le microcontrôleur PIC 16F877a contient trois types de mémoires:

##### **4.1. Mémoire de données RAM**

C'est une mémoire vive (RAM= Random Access Memory ou mémoire à accès aléatoire), c'est-à-dire son contenu est perdu lorsque elle n'est plus alimentée. Elle est principalement utilisée pour stocker les données et les variables temporaires [14].

Cette mémoire comprend les registres de configuration spéciaux SFR (Special Function Registers) et généraux GPR (General Propose Registers) permettant de configurer le microcontrôleur ainsi que ses périphériques.

L'espace mémoire RAM adressable est de 512 positions de 1 octet chacune:

- \* 96 positions sont réservées au SFR (Special Function Registers).
- \* Les 416 positions restantes constituent les registres GPR (General Propose Registers) ou RAM utilisateur. Trois blocs de 16 octets ne sont pas implantés physiquement, donc la capacité de la mémoire RAM disponible du 16F877 est de 368 octets.

La Figure ci-dessous montre l'organisation de la RAM avec les zones allouées au SFR et aux GPR.

Indirect addr	00h	Indirect addr	80h	Indirect addr	100h	Indirect addr	180h
		OPTION_REG	81h				
TMR0	01h	PCL	82h	TMR0	101h	TMR0	181h
PCL	02h	STATUS	83h	PCL	102h	PCL	182h
STATUS	03h	FSR	84h	STATUS	103h	STATUS	183h
FSR	04h	TRISA	85h	FSR	104h	FSR	184h
PORTA	05h	TRISB	86h		105h		185h
PORTB	06h	TRISC	87h	PORTB	106h	PORTB	186h
PORTC	07h	TRISD	88h		107h		187h
PORTD	08h	TRISE	89h		108h		188h
PORTE	09h	PCLATH	8Ah		109h		189h
PCLATH	0Ah	INTCON	8Bh	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	PIE1	8Ch	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE2	8Dh	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PCON	8Eh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh		8Fh	EEDATH	10Eh		18Eh
TMR1H	0Fh		90h	EEADRH	10Fh		18Fh
T1CON	10h	SSPCON2	91h		110h		190h
TMR2	11h	PR2	92h		111h		191h
T2CON	12h	SSPADDD	93h		112h		192h
SSPBUF	13h	SSPSTAT	94h		113h		193h
SSPCON	14h		95h		114h		194h
CCPR1L	15h		96h		115h		195h
CCPR1H	16h		97h		116h		196h
CCP1CON	17h	TXSTA	98h		117h		197h
RCSTA	18h	SPBRO	99h		118h		198h
TXREG	19h		9Ah		119h		199h
RCREG	1Ah		9Bh		11Ah		19Ah
CCPR2L	1Bh	CMCON	9Ch		11Bh		19Bh
CCPR2H	1Ch	CVRCON	9Dh		11Ch		19Ch
CCP2CON	1Dh	ADRESL	9Eh		11Dh		19Dh
ADRESH	1Eh	ADCON1	9Fh		11Eh		19Eh
ADCON0	1Fh		A0h		11Fh		19Fh
	20h				120h		1A0h
			FFh				
	7Fh	<b>Banque 1</b>			17Fh		1FFh

Figure 2.3 L'organisation de la mémoire RAM.

La RAM apparaît alors organisée en 4 banques de 128 octets chacun. Les deux bits RP0 et RP1 du registre STATUS permettent de choisir une banque.

Les zones grisées ne sont pas implantées physiquement.

#### 4.2. Mémoire de programme EEPROM

C'est une mémoire morte de type EEPROM (Electrically Erasable Programmable Read Only Memory) c'est-à-dire qu'elle ne perd pas son contenu même si elle n'est plus alimentée. L'avantage principal de cette mémoire est qu'elle est reprogrammable et effaçable par un courant électrique. Elle est utilisée pour stocker le programme du microcontrôleur, mais pas les variables [15].

Cette mémoire de programme est d'une taille 8 kilo octets. Le microcontrôleur PIC 16F877 est caractérisé par la possibilité d'écrire des données, le compilateur génère un fichier " hex ", ce fichier sera ensuite transféré dans la mémoire programme du microcontrôleur à l'aide du programmeur.

#### 4.3. Mémoire de donnée EEPROM

Cette mémoire est utilisée pour stocker les données à long terme, alors que la RAM est utilisée pour les données et les variables du programme. La lecture et l'écriture dans cette mémoire de donnée EEPROM sont beaucoup plus lentes que dans la mémoire de données RAM. Son utilisation implique une procédure spéciale (par exemple mot de passe). La taille de cette mémoire EEPROM est de 256 octets [16,17]

### 5. Explication de certains registres spéciaux

La mémoire de données RAM contient des registres spécifiques qui permettent de configurer le microcontrôleur. Ces registres sont utilisés par le microprocesseur et les modules périphériques pour contrôler les opérations du microcontrôleur. Pour le PIC 16F877A, ces registres sont nombreux, mais on ne va citer que les registres qui seront utilisés dans notre travail.

#### 5.1. Registre STATUS

Chaque bit de ce registre a une fonction spécifique.

IRP	RP1	RP0	TO	PD	Z	DC	C
-----	-----	-----	----	----	---	----	---

\* IRP : ce bit permet la sélection des pages en adressage indirect.

\* RP0 et RP1 : ces deux bits sont utilisés pour la sélection de l'une des banques 0 à 3.

RP1	RP0	Page ou banque sélectionnée
0	0	Page 0

0	1	Page 1
1	0	Page 2
1	1	Page 3

**Tableau 2.1** Bits de sélection des banques.

- \* TO: Time Out Bit: ce bit indique le débordement du TIMER watchdog (WDT).
- \* PD: Power Down Bit indique quel événement a entraîné le dernier arrêt du PIC.
- \* Z: Zéro Bit: mis à 1 si le résultat de la dernière opération est zéro.
- \* DC: Digital Carry bit: retenue sur le quatrième bit des poids faibles.
- \* C: Carry Bit: retenue sur le bit MSB sur un octet.

## 5.2. Registre TRISA

Les ports du PIC sont contrôlés par les registres de directions TRISx (TRISA, TRISB, TRISC, TRISD et TRISE). Le registre TRISA permet de déterminer la direction du port A (ce port est bidirectionnel et constitué de 6 bits) [18].

-	-	RA5	RA4	RA3	RA2	RA1	RA0
---	---	-----	-----	-----	-----	-----	-----

RA0 à RA5: quand on écrit un "1" dans TRISA, le bit correspondant du PORTA est configuré en entrée. Si on écrit un "0", bit correspondant du PORTA devient une sortie.

Les bits 6 et 7 ne sont pas implémentés.

## 5.3. Registre PORTA

Ce registre permet de déterminer l'état logique de la sortie des broches RA0 jusqu'à RA5.

-	-	RA5	RA4	RA3	RA2	RA1	RA0
---	---	-----	-----	-----	-----	-----	-----

Quand on écrit un "1" dans registre PORTA, le bit correspondant sera mise à 5 V. Si on écrit un "0", bit correspondant devient 0 V.

## 5.4 .Présentation des registres du module CAN

Le module de convertisseur analogique/numérique du microcontrôleur PIC 16F877 possède quatre registres permettant de configurer ces CAN. On utilise ces registres pour configurer des fonctionnalités telles que: la sélection des canaux, la tension de référence pour les CAN, la fréquence d'échantillonnage, l'horloge analogique/numérique et le format du résultat (stockage du résultat).

### 5.4.1. Register ADCON0 (A/D control register 0)

Ce registre permet d'activer et de sélectionner les CAN du PIC16F877A.

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
-------	-------	------	------	------	---------	---	------

\* ADON: ce bit permet d'activer les CAN.

\* Le deuxième bit n'est pas utilisé.

\* GO-DONE: ce bit indique l'état de la conversion, s'il est à " 1 ", la conversion est en cours, s'il est à " 0 " la conversion est terminée.

\* CHS0-2: Analog Channel Select bits.

000 = Channel 0 (AN0).

001 = Channel 1 (AN1).

010 = Channel 2 (AN2).

011 = Channel 3 (AN3).

100 = Channel 4 (AN4).

101 = Channel 5 (AN5).

110 = Channel 6 (AN6).

111 = Channel 7 (AN7).

Ces trois bits permettent de sélectionner les convertisseurs analogiques/numériques (AN0-7).

\* ADSC1 et ADSC0=Clock select bits.

Ces bits permettent de choisir la vitesse de conversion.

00 =  $F_{osc}/2$ .

01 =  $F_{osc}/8$ .

10 =  $F_{osc}/32$ .

11 = Oscillateur RC interne.

#### 5.4.2. Register ADCON1 (A/D control registers 1)

Le registre ADCON1 a deux fonctions: la première c'est de sélectionner la fonctionnalité des broches en tant que broche numérique ou broche analogique, mais, elles ne peuvent être utilisées qu'une seule fonction à la fois. La deuxième est de déterminer la tension de référence pour les convertisseurs analogiques/numériques.

ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
------	---	---	---	-------	-------	-------	-------

PCFG0-3: ces bits permettent de configurer les broches en tant que broches numériques ou analogiques. Le tableau ci-dessous montre les différentes configurations

PCFG0-3	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A
0011	A	A	A	A	A	A	A	A
0100	A	A	A	A	A	A	A	A
0101	A	A	A	A	A	A	A	A
0110	D	A	A	A	A	A	A	A
0111	D	D	A	A	A	A	A	A

**Tableau 2.2** Les différentes configurations.

\* Les cinquième, le sixième et le septième bit ne sont pas implémentés.

\* ADFM: ce bit définit le format du résultat ADC.

#### 5.4.2. Registres ADRESH et ADRESL

Le résultat de la conversion analogique/numérique est donné sur 10 bits, mais étant donné qu'un registre ne peut contenir que 8 bits, donc, le résultat de la conversion doit être sauvegardé dans deux registres séparés qui sont les registres ADRESL et ADRESH.

Le registre ADRESL contient les huit bits du résultat de conversion analogique/numérique les moins significatifs ou les bits les plus significatifs d'un nombre numérique à 10 bits [19]. Le registre ADRESH contient les deux bits du résultat de conversion analogique/numérique supérieurs / inférieurs d'un nombre numérique à 10 bits.

#### 5.5. Présentation du module PWM

Un signal PWM (Pulse Width Modulation) est un signal rectangulaire d'une fréquence fixe et d'un rapport cyclique variable. Il y a deux modules identiques CCP1 et CCP2 composés chacun d'un registre 16 bits. Ces deux registres peuvent opérer soit comme un registre 16 bits en mode capture soit en mode comparaison et enfin comme un registre 8 bits pour générer signal PWM.

Les broches permettant de produire un signal PWM sont les bits 1 et 2 du PORTC (RC1/CCP2 et RC2/CCP1) [20].

### 5.5.1. Registre CCP1CON

-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
---	---	-------	-------	--------	--------	--------	--------

\* CCP1M0-3: ces bits permettent de sélectionner le mode de fonctionnement des broches.

0000 = le module CCP est arrêté (le mode de capture, le mode de comparaison et le mode PWM ne sont pas valides).

0100 = mode capture à chaque front descendant.

0101 = mode capture à chaque front montant.

0110 = mode capture tous les 4 fronts montants.

0111 = mode capture tous les 16 fronts montants.

1000 = mode comparaison, mise à " 1 " de la sortie lorsque le bit CCPxIF passe à " 1 ".

1001 = mode comparaison, mise à " 0 " de la sortie lorsque le bit CCPxIF passe à " 1 ".

1010 = mode comparaison, production d'une interruption lorsque le bit CCPxIF passe à " 1 ".

1011 = mode comparaison, lorsque le bit CCPxIF passe à un, mise à " 0 " de TMR1 et début d'une conversion analogique numérique (si elle est active).

11xx = mode production du signal PWM.

Mode production du signal PWM : permet de produire un signal rectangulaire de fréquence fixe et de rapport cyclique variable [21].

\* CCPxX et CCPxY : Ce sont les deux bits de poids faible du temps du rapport cyclique en mode PWM.

## 6. Conclusion

Ce chapitre était consacré à l'étude du microcontrôleur PIC 16F877. Ce microcontrôleur possède 40 broches réparties sur cinq ports d'entrées/sorties: PORTA, PORTB, PORTC, PORTD et PORTE. Il comprend tous les éléments dont on a besoin dans notre travail tels que: le convertisseur analogique/numérique (CAN sur 10 bits) et deux modules PWM travaillant également sur 10 bits [22].

## **CHAPITRE III : ISIS \_PROTEUS**

## 1. Introduction

Isis-Proteus est un logiciel de simulation, qui est largement utilisé dans ces dernières années comme un outil pédagogique, à cause de sa simplicité due à son environnement graphique et interactif. Le document est rédigé d'une façon relativement simple, avec un niveau de difficulté Progressif, afin qu'il soit utile [23].

Ce logiciel de simulation permet en premier lieu une révision et une assimilation de quelques notions de base d'électricité et d'électronique. Cet outil facilite par la suite la tâche à l'utilisateur durant la réalisation de sont mini-projet.

Les différents outils mis à la disposition de l'utilisateur dans Isis-Proteus permettent aussi des analyses dans des domaines différents (temporel, fréquentiel), ainsi que des régimes différents (statique, dynamique).

Les différents outils du logiciel sont décrits au fur et à mesure, via une série d'exemples et d'applications, couvrant différents domaines de l'électronique, allant des circuits analogiques simples (RLC, circuits à base de diodes et transistors, AOP, etc.) aux circuits numériques y compris les microcontrôleurs [24].

Certain applications sont validées par des tests expérimentaux effectuées au niveau du laboratoire, afin de permettre à l'utilisateur d'analyser et de comparer les résultats obtenus, théoriquement, expérimentalement, et par simulation informatique.

Ce chapitre est consacré à l'étude d'un logiciel de simulation des circuits et systèmes électroniques celui de l'Isis Proteus.

## 2. Définition de logiciel ISIS

ISIS "intelligent schematic input system" est un logicielle destinée aux électroniciens. Il a été développé par la société. Le logiciel Isis de Proteus est principalement connu pour éditer des schémas électriques.

Ce logiciel permet également de simuler des montages. Les résultats de la simulation sont très proches de la réalité, il permet la simulation d'éléments analogique, numérique, logique de microcontrôleur, d'afficheur LDC [25].

Proteus possède d'autres avantages :

- \* Pack contenant des logiciels facile et rapide à comprendre et à utiliser.
- \* Le support technique est performant.
- \* L'outil

L de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet [26].

### 2.1. Isis intelligent schématique input system

Le logiciel Isis Proteus est principalement connu pour éditer des schémas électriques puis procéder à une animation en temps réel du montage. par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations, car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits [27].

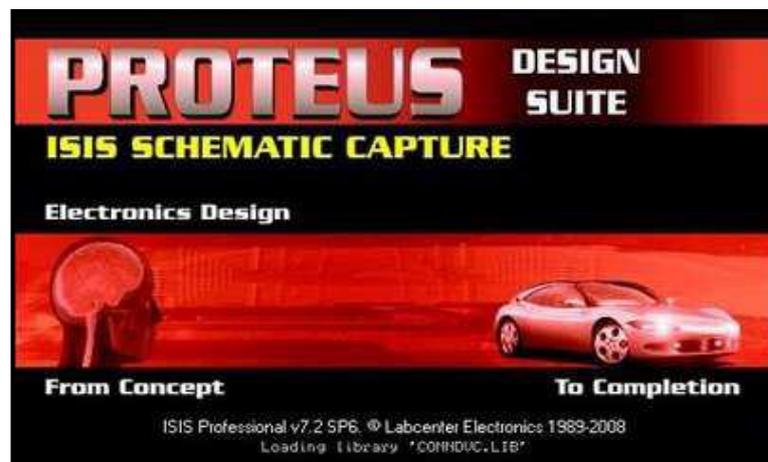


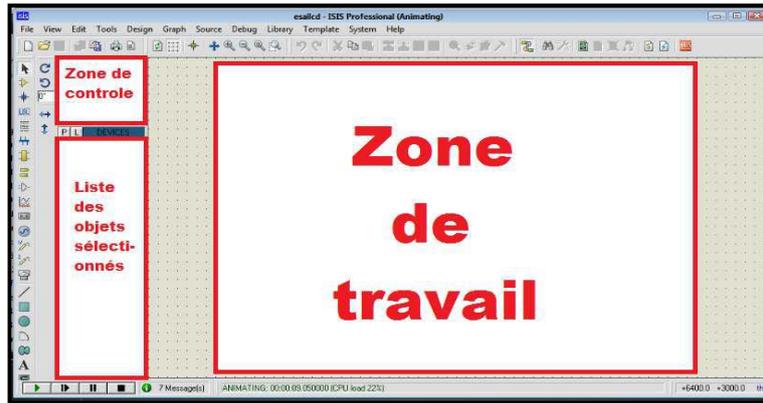
Figure 3.1 Logiciel Isis Proteus.

### 2.2. Prise en main du logiciel Isis

#### a. La saisie de schéma:

L'écran se compose de plusieurs parties:

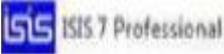
- \* La zone de travail, dans laquelle le schéma sera disposé.
- \* La zone de contrôle qui permet de se déplacer d'un endroit à un autre de la zone de travail.
- \* Le clavier.
- \* La liste des objets sélectionnés pour réaliser le schéma.



**Figure 3.2** Présentation de la page d'Isis.

\* Démarrer le logiciel:

On lance depuis le raccourci du bureau ou du menu démarrer de l'ordinateur le logiciel

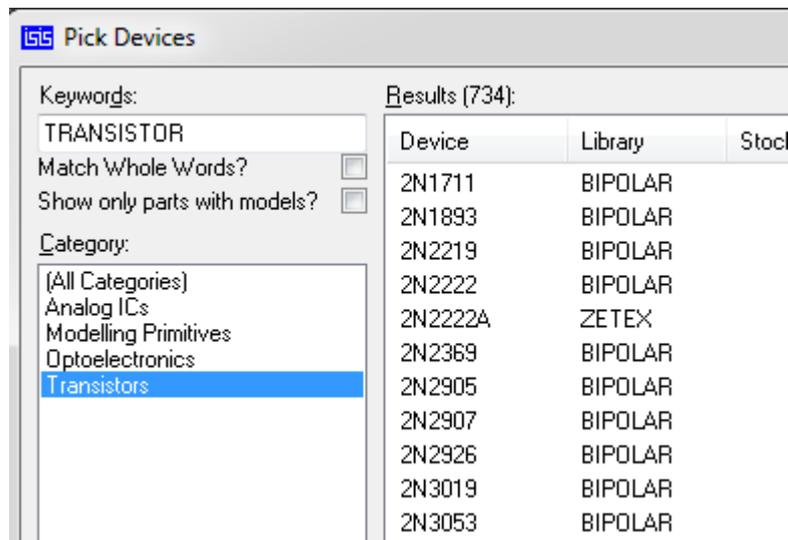
Proteus – Isis : 

\* Création de la liste des composants :

- On sélectionne l'icône composant : 

- Dans la fenêtre du sélectionneur d'objets, on clique sur p : 

- Une nouvelle fenêtre (pick devices) de choix de composants s'ouvre : on trouve les composants désirés à partir d'un mot clé, ou dans la catégorie proposée, puis dans la Sous-catégorie.



- Pour faire apparaître le composant on clique une fois, et on clique deux fois pour que le composant s'inscrive sur la liste de travail devices.

- Pour placer tous les composants dès le début du travail qu'on aura besoin par la suite sur la liste du travail, puis on ferme la fenêtre en cliquant sur ok.

- Placement des composants sur le schéma:

- On sélectionne le composant à placer dans la liste faite précédemment et le placer dans la zone de travail de la fenêtre d'édition.

-On peut faire tourner le composant ou lui faire une symétrie à l'aide des outils d'orientation du menu de la boîte à outils :



- Placement des différentes connexions externes :

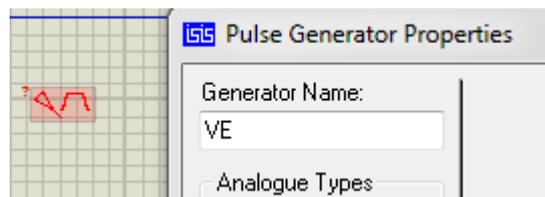
Pour les éléments : grounds ; output ; input, etc.

Après avoir cliqué sur l'icône , on sélectionne la connexion et après l'avoir mise dans le « panier », la placer sur le schéma.

- Placement des alimentations et des générateurs :

Il faut maintenant placer les différentes alimentations ou générateurs utilisés.

- On clique sur l'icône  et on sélectionne un générateur (dc, sine, pulse...).



Ensuite on clique sur le schéma pour le placer (ici pulse = rectangle).

-Les alimentations ou générateurs placés, il faut leur donner un nom et régler les différents paramètres qui sont fonctions du type de générateurs choisi.

-Lorsqu'on clique gauche sur le générateur, le symbole devient rouge, lorsqu'on reclique gauche, une fenêtre pour éditer les propriétés s'ouvre. grâce à cette fenêtre, on donne un nom et une amplitude des tensions, et on peut aussi régler la forme de la fréquence du signal [28].

- Réalisations des connexions :

Il reste à relier les connexions entre les différents symboles.

-Sélectionner dans le menu .

### **b. Placement des instruments de mesure :**

Il faut choisir le ou les appareils de mesure que l'on souhaite utiliser en cliquant sur

License  :



### c. Animation et simulation du fonctionnement :

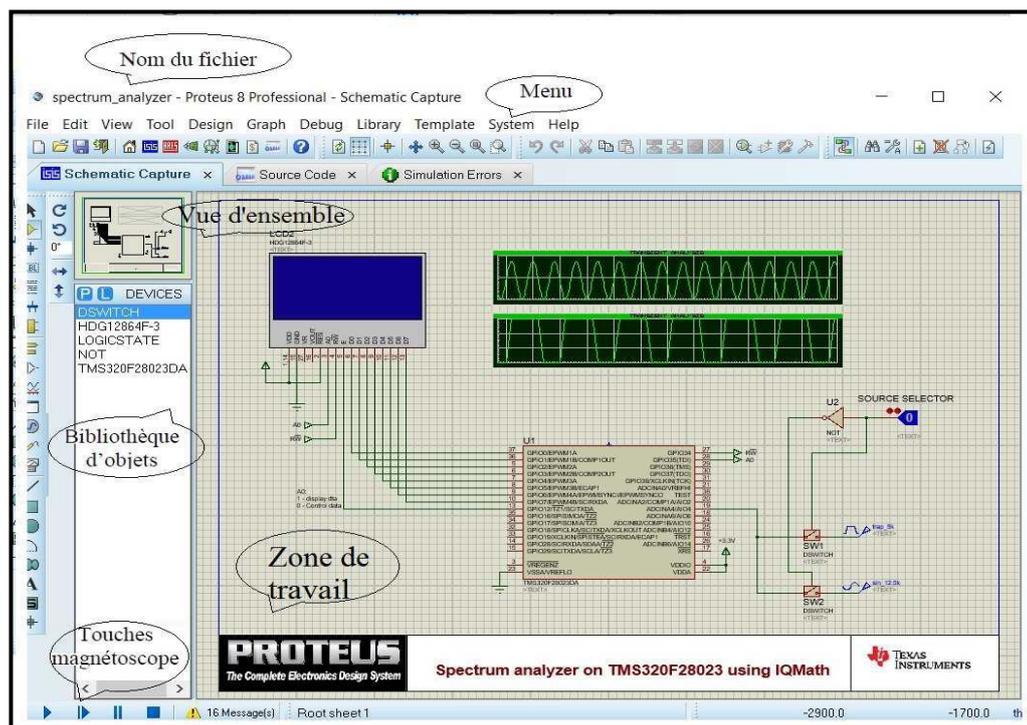
Pour lancer l'animation, on clique sur la flèche dans l'image ci-dessous



### 2.3. L'environnement de travail de Proteus

Le lancement de l'Isis Proteus donne un environnement classique de type Windows, constitué d'une fenêtre principale, et d'un ensemble de barres d'outils.

En outre le menu classique contient: la gestion des fichiers, l'affichage, et les options des projets. La fenêtre principale comprend une zone de travail [29].



#### 2.3.1. La barre d'outils principale

Sans cliquer, une info-bulle affiche le nom du bouton, dont les plus importants sont:

- ✓ **Component mode (Mode Composant) :**

Un clic sur ce bouton Component from libraries, permet l'ajout de différents composants.

✓ **Générateur Mode (Mode Générateur):**

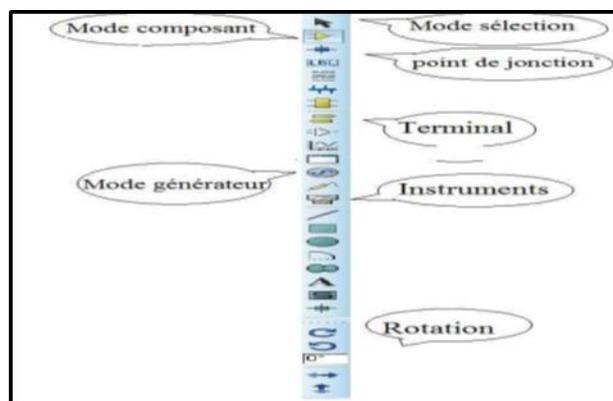
Permettant l'accès aux différents types de générateurs.

✓ **Instruments :**

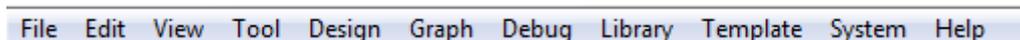
Un raccourci permettant l'ajout des appareils de mesure tels que le voltmètre, l'ampèremètre, et l'oscilloscope.

✓ **Terminal :**

Permettant d'ajouter des points particuliers dans un schéma tels que les entrées/sorties.



a) **Barre de menus**



Elle permet de gérer les travaux (ouverture, sauvegarde, etc.) sur les fichiers.

b) **barre des outils de commande**



Elle reprend ce qui est accessible par les menus.

Commandes sur les fichiers (nouveau, ouvrir, etc.)



Commandes d'affichage (grille, zoom, etc.)



Commandes Edition / Objets / Bibliothèque...



Commandes Outlie / Projects, etc.



c) Barre d'outils de sélection des Modes :

- **Mode Principal:**
  - Sélection d'objet
  - Composants
  - nœuds
  - Label
  - Bus, etc.
- **Mode Gadgets:**
  - Terminal
  - Graphes
  - Générateurs
  - Sondes
  - Instruments virtuels, etc.
- **Mode Graphique:**
  - Traits
  - Forme
  - Textes
  - Arcs, etc. [10]

### 2.3.2. Zone de travail

Zone rectangulaire où on dépose les composants pour dessiner le schéma structurel du modèle à simulé ou de la carte à router [30].

### 2.3.3. Sélection d'objet

La zone de sélection se présente par une Zone rectangulaire où on trouve tous les composants présents dans le dessin [31].

### 2.3.4 Gestion d'un projet

-Création d'un projet: menu "fichier" puis "nouveau projet".

-Ouverture d'un projet: menu "fichier" puis "ouvrir".

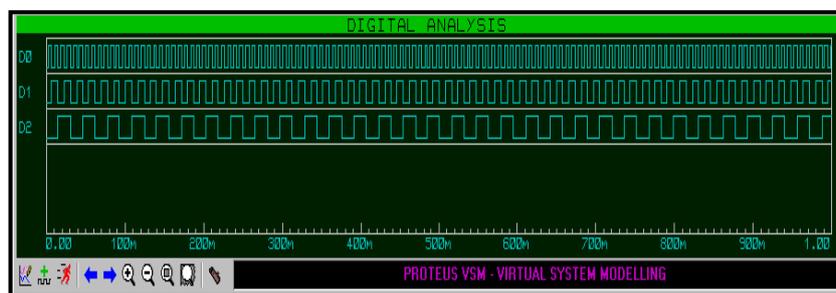
Enregistrement d'un projet: menu "fichier" puis "enregistrer sous, etc [32].

## 2.4. Simulation

Après avoir saisi le schéma, il faut définir les types de générateurs qui vont alimenter le circuit (pas d'alimentation des composants logiques : elles existent en interne, cela évite de charger le dessin). Pour pouvoir visualiser les résultats il faudra placer des sondes et des graphes.

### 2.4.1. Placement d'un graphe de simulation

Dans la grille de dessin pour faire apparaître un graphe de simulation : on clic droit maintenu pour dessiner une fenêtre de graphe (relâchement à la taille voulue) puis on re-clic droit. Par clic gauche sélectionner sur la structure une sonde (tension ou courant) et la déposer sur le graphe (drag & drop)

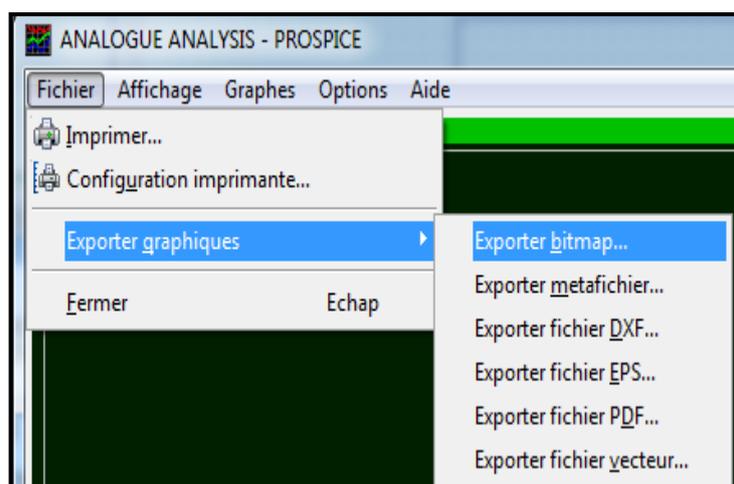


### 2.4.2. Position et dimension d'un graphe

Clic gauche sur le graphe puis étendre le graphe au besoins, ou double clic pour ouvrir le graphe et avoir un plein écran.

### 2.4.3. Export des graphes

Dans le menu "Fichier" puis "Exporter graphiques" puis " Exporter bitmap" puis OK votre graphe est dans le presse-papier. On peut aussi le nommer pour le retrouver [33].



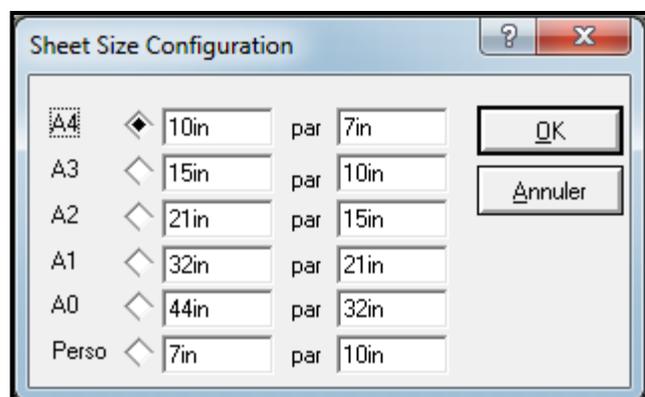


### 3. Configuration de l'application D'ISIS PROTEUS V7

On peut configurer Proteus pour avoir par exemple une sauvegarde automatique: dans "système" puis "définir environnement" afin d'obtenir la fenêtre ci-après et la modifier selon vos souhaits:

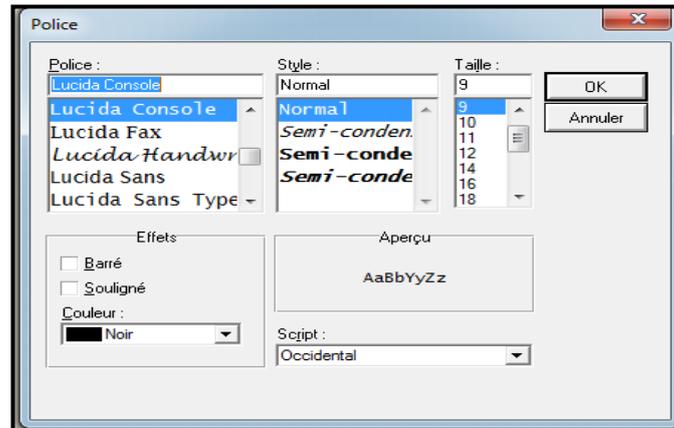


On peut configurer Proteus pour avoir par exemple une taille de feuille définie : dans "système" puis "définir taille des feuilles" afin d'obtenir la fenêtre ci-dessous et la modifier selon vos souhaits:



On peut configurer Proteus pour avoir par exemple la police des caractères utilisés:

Dans "système" puis "définir éditeur de texte" afin d'obtenir la fenêtre ci-dessous et la modifier selon les besoins:



#### 4. Description de fonctionnement de logiciel ISIS

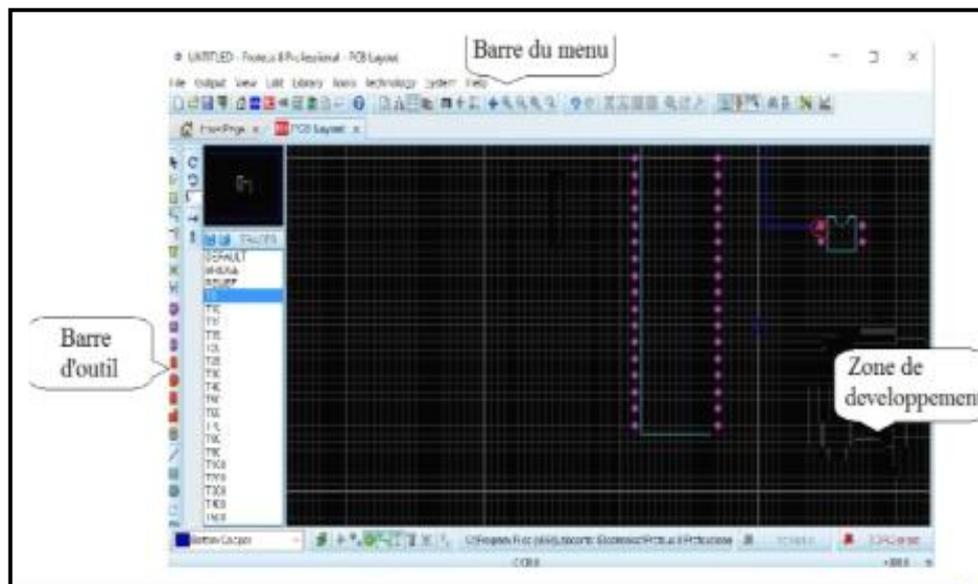
Après l'installation de notre sociale, on lancer l'étude ISIS PROTEUSE .Ce logiciel nous permet de dessiner des schémas de haute qualité, on peut ainsi produire des schémas attrayant tels que ceux publiés dans les magazines.

\* L'interface utilisateur de l'éditeur de schémas ISIS et facile et pratique, les actions courantes sont simples et rapides à exécuter.

\* Ce simulateur contient des aides au concepteur [34].

#### 5. Présentation de l'interface ARES

Le lancement d'ARES donne PROTEUS.



La barre d'outils offre quatre modes, dont les noms sont donnés dans la figure ici:

Les principaux raccourcis seront déroutage automatique.

- **Remarques**

- La visibilité et la couleur des différentes couches est contrôlable via "View>Edit layer colour and Visibility".

- L'affichage de la grille, et le changement de l'unit de mesure des distances se font "View>Toggle Grid" et "View>ToggleMetric/Imperial".

- ✓ Les règles de conception du circuit imprimé sont fixées via "Technologie>Design Rule Manager".

Elles définissent le type et taille des pistes du circuit.

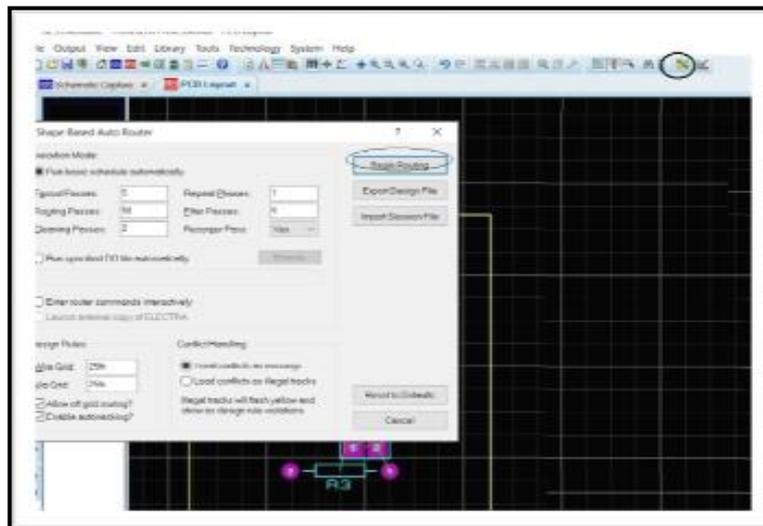
Ces règles de routage (leur stratégies sont définies pratiquement dès la saisie du schéma, et sont classées en trois différents type:

- ✓ POWER : pour les signaux VCC et GND.
- ✓ BUS : pour les liens nommés.
- ✓ SIGNAL : pour le reste de liens.

Un exemple est donné dans la figure ci:

- ✓ Le mode "piste" (Trace) permet de définir la taille des pistes à concevoir durant le processus de routage.

Le lancement du routage automatique est effectué via l'icône "auto-router (figure cici-



## **6. Conclusion**

Ce chapitre était consacré à la représentation du logiciel de simulation celui d'ISIS PROTEUS. Ce logiciel présente un outil très efficace aux électroniciens et aux amateurs en électronique. Il leur permet d'implémenter leur schémas électrique avant de passer à la réalisation pratique [35]. Il leur permet aussi de tester connaissances et d'approfondir leur compréhensions.

# **CHAPITRE IV : COMMANDE D'UN CONVERTISSEUR DC-DC BOOST**

## 1. Introduction :

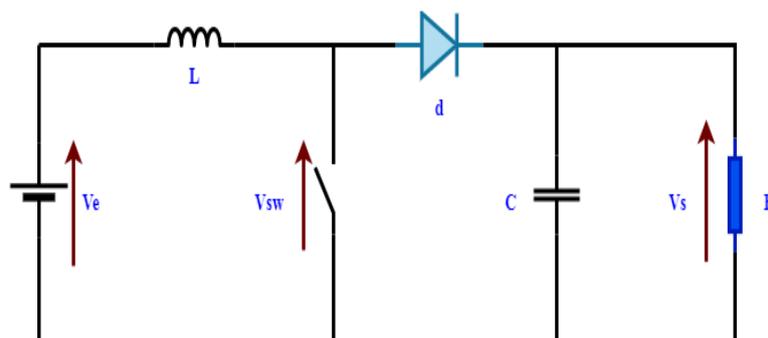
Dans ce chapitre, un convertisseur Boost sera étudié à l'aide d'un logiciel de simulation très utilisé en électronique celui de l'Isis-Proteus et un compilateur mikroC PRO for PIC qui utilise le langage C++ comme langage de programmation. Dans un premier de temps, on va commander le convertisseur Boost (en boucle ouverte) en faisant varier le rapport cyclique manuellement à l'aide d'un générateur d'impulsions disponible dans le logiciel Isis-Proteus, après, un circuit intégré nommé NE555 sera utilisé afin de générer un signal PWM pour commander le convertisseur Boost. Ensuite, on utilisera un microcontrôleur PIC 16F877A (à la place du circuit NE555) pour générer un signal PWM [37].

Enfin, on va commander le convertisseur Boost en boucle fermée en utilisant un régulateur PID. La tension de sortie du convertisseur Boost sera manipulée (avoir des valeurs désirées) automatiquement via le régulateur PID. On terminera ce chapitre par une conclusion.

## 2. Convertisseurs dc/dc boost

### 2.1. Principe

Un convertisseur Boost ou hacheur parallèle est un convertisseur qui convertit une tension continue en tension continue d'une valeur plus forte que celle de l'entrée en faisant varier le rapport cyclique. Le schéma générale d'un convertisseur Boost est donné dans la figure ci-dessous.

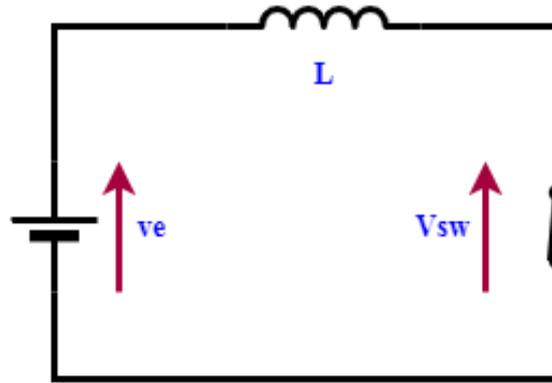


**Figure 4.1** Schéma d'un convertisseur Boost.

### 2.3. Analyse du fonctionnement

On peut décomposer l'analyse du circuit du convertisseur Boost en deux cas:

\* Le premier cas quand:  $0 < t < \alpha T_{sw}$  (l'interrupteur est fermé et la diode  $D$  est ouverte).



**Figure 4.2** Schéma équivalent d'un convertisseur Boost quand  $0 < t < \alpha T_{sw}$

En appliquant la loi de mailles, on obtient l'équation suivante:

$$V_e = V_L + V_{sw} \tag{4.1}$$

$$V_L = V_e - V_{sw} \tag{4.2}$$

$$L \frac{di}{dt} = V_e - V_{sw} \tag{4.3}$$

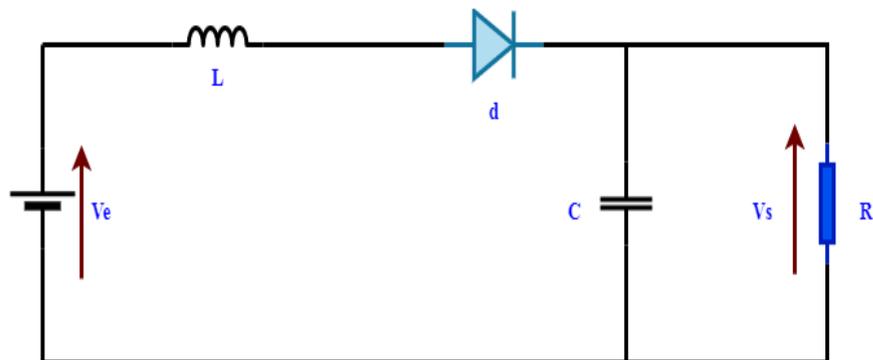
$$di = \frac{1}{L} (V_e - V_{sw}) dt \tag{4.4}$$

En intégrant les deux parties de l'équation (4.4), on aura:

$$\int_0^{\alpha T_{sw}} di = \frac{1}{L} \int_0^{\alpha T_{sw}} (V_e - V_{sw}) dt \tag{4.5}$$

$$i_{ch} = \frac{1}{L} (V_e - V_{sw}) (\alpha T_{sw}) \tag{4.6}$$

\* Le deuxième cas quand:  $\alpha T_{sw} < t < T_{sw}$  (l'interrupteur est ouvert et la diode  $D$  est fermée).



**Figure 4.3** Schéma équivalent d'un convertisseur Boost quand  $\alpha T_{sw} < t < T_{sw}$

En appliquant la loi de mailles, on obtient l'équations suivante:

$$V_e = -V_L + V_d + V_s \tag{4.7}$$

$$V_L = -V_e + V_d + V_s \quad (4.8)$$

$$L \frac{di}{dt} = -V_e + V_d + V_s \quad (4.9)$$

$$\frac{di}{dt} = \frac{1}{L} (-V_e + V_d + V_s) \quad (4.10)$$

Par intégration des deux parties de l'équation (4.10), on obtient l'équation suivante:

$$\int_{\alpha T_{sw}}^{T_{sw}} di = \frac{1}{L} \int_{\alpha T_{sw}}^{T_{sw}} (-V_e + V_d + V_s) dt \quad (4.11)$$

$$i_{dch} = \frac{1}{L} (-V_e + V_d + V_s) (T_{sw} - \alpha T_{sw}) \quad (4.12)$$

## 2.4 Relation entre la tension d'entrée et la tension de sortie

Puisque la tension moyenne aux bornes de l'inductance est nulle, on peut dire que:

$i_{ch} = i_{dch}$ . En égalisant l'équation (4.6) avec l'équation (4.12), on obtient:

$$\frac{1}{L} (V_e - V_{sw}) (\alpha T_{sw}) = \frac{1}{L} (-V_e + V_d + V_s) (1 - \alpha) T_{sw} \quad (4.13)$$

$$(V_e - V_{sw}) \alpha = (-V_e + V_d + V_s) (1 - \alpha) \quad (4.14)$$

$$\frac{(V_e - V_{sw}) \alpha}{1 - \alpha} = (-V_e + V_d + V_s) \quad (4.15)$$

$$\frac{(V_e - V_{sw}) \alpha + (V_e - V_d) (1 - \alpha)}{(1 - \alpha)} = V_s \quad (4.16)$$

Donc, la relation entre la tension d'entrée et la tension de sortie d'un convertisseur Boost peut être décrite par l'équation suivante:

$$V_s = \frac{(V_e - V_{sw}) \alpha + (V_e - V_d) (1 - \alpha)}{(1 - \alpha)} \quad (4.17)$$

Si on néglige la tension directe de la diode  $V_d$  et la tension du transistor MOSFET  $V_{DS}$ , c'est-à-dire:  $V_d = 0$  et  $V_{sw} = 0$ .

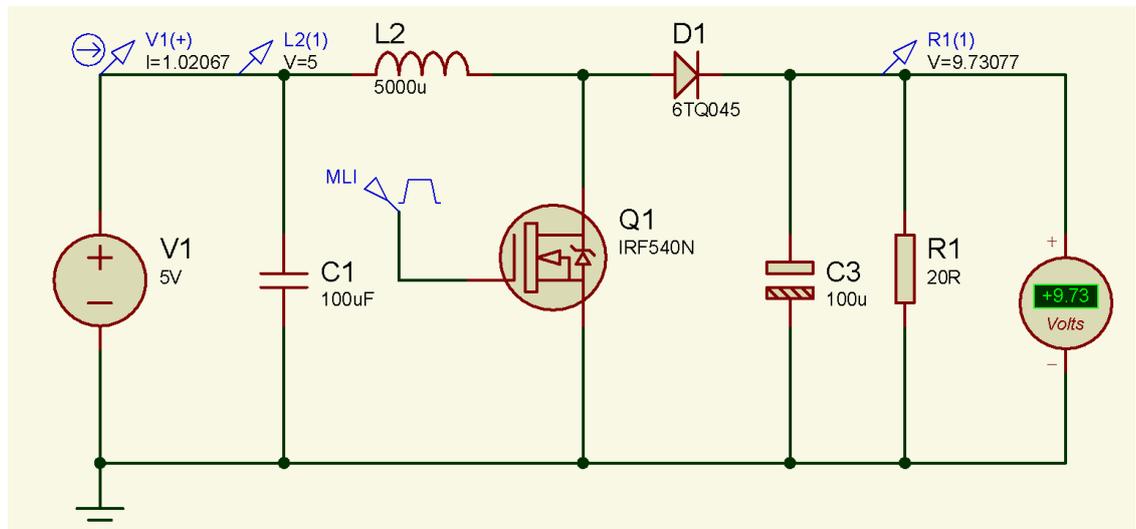
Donc, l'expression de la tension de sortie du convertisseur Boost en fonction de la tension d'entrée et le rapport cyclique devient:

$$V_s = \frac{V_e}{(1 - \alpha)} \quad (4.18)$$

## 3. Commande d'un convertisseur Boost en boucle ouverte

D'après l'équation (4.18) qui exprime la tension de sortie d'un convertisseur Boost en fonction de sa tension d'entrée, on peut contrôler ou commander la tension de sortie d'un convertisseur Boost en faisant varier le rapport cyclique  $\alpha$ .

Dans un premier temps, on commande le convertisseur Boost en boucle ouverte, c'est-à-dire: on fait varier le rapport cyclique manuellement.



**Figure 4.4** hacheur Boost en boucle ouverte.

Les valeurs des paramètres du convertisseur Boost sont récapitulés dans le tableau suivant:

Les paramètres	Les Valeurs
La tension d'entrée $V_e$	5 V
La bobine $L$	5000 $\mu H$
Le condensateur $C_1$	100 $\mu F$
Le condensateur $C_2$	100 $\mu F$
La résistance $R$	20 $\Omega$

**Tableaux 4.1** Valeurs des composants utilisés.

Les caractéristiques électriques des éléments du convertisseur Boost sont:

\* L'interrupteur est un transistor MOSFET de type IRF540N avec les caractéristiques électriques suivantes:

- MOSFET N\_channel.
- $i_d = 33$  Ampères (le courant maximal supporté par ce transistor).
- $V_{gs} = 10$  V.
- $V_{dss} = 100$  V (la tension maximale supportée par ce transistor).
- $P_d = 130$  W.

\* Diode 10A01

- $V_r = 50$  V tension inverse de cette diode.

- Courant direct: 10A.
- $V_f = 1$  V Tension directe de la diode.

Le tableau ci-dessous montre les valeurs calculées théoriquement en utilisant l'expression  $V_s = \frac{V_e}{(1-\alpha)}$  (le cas idéal) et les valeurs trouvées en simulation.

$\alpha$	$V_s$ théorique	$V_s$ simulation
10	5.5v	5.25v
25	6.66v	6.39v
30	7.14v	6.88v
50	10v	9.82v
75	20v	20.05v
85	33.33v	31.20 v

**Tableau 4.2** Comparaison entre les valeurs théoriques et les valeurs de simulation.

#### 4. Commande d'un convertisseur Boost par le circuit NE555

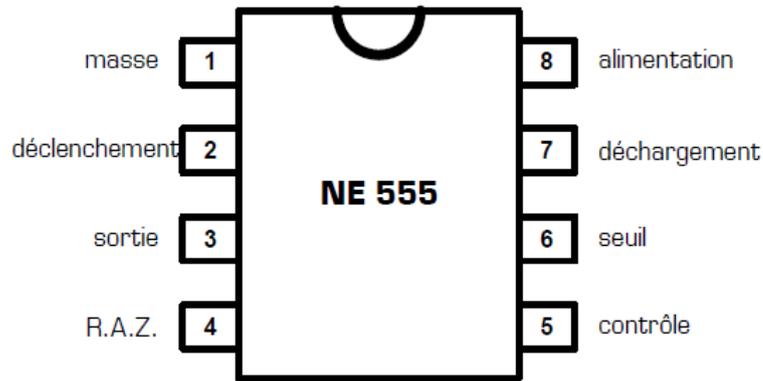
Afin de contrôler le convertisseur Boost (toujours en boucle ouverte), on doit commander l'ouverture et la fermeture de l'interrupteur (transistor MOSFET IRF540N), pour ce faire, on a besoin d'un circuit permettant de générer un signal PWM (Pulse Width Modulation ) ou MLI (Modulation de Largeur d'Impulsion) afin de commander le transistor. Parmi les circuits qui permettent de générer un signal PWM, il y a le circuit intégré NE555 (fonctionne en mode astable).

##### 4.1. Présentations du circuit NE555

Le NE555 est un circuit qui se présente sous la forme d'un circuit intégré TTL à 8 bornes. Il permet de réaliser les fonctions: astable, bistable et monostable. Il représente aujourd'hui une des solutions les plus utilisées pour la génération de signaux rectangulaire comme le montre le brochage sur la figure 4.5.

Les 8 bornes du circuit intégré NE 555 sont:

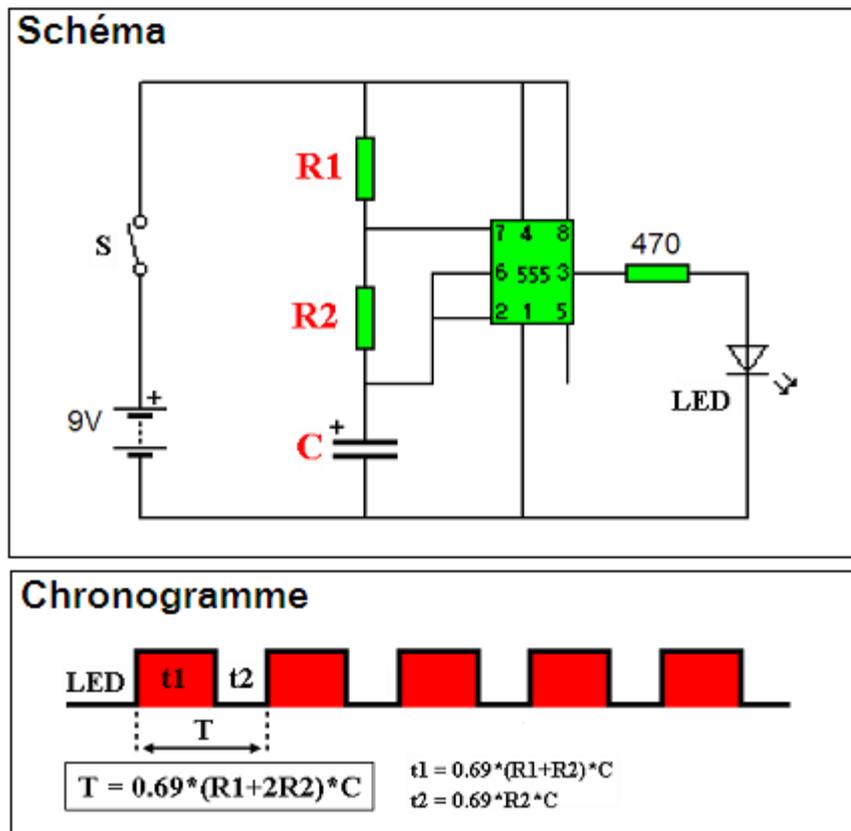
- Broche 1: La masse (alimentation 0 V).
- Broche 2: L'entrée de déclenchement.
- Broche 3: La sortie principale.
- Broche 4: L'entrée raz (remise a zéro).
- Broche 5: La sortie de contrôle.
- Broche 6: L'entrée de seuil.
- Broche 7: La sortie de déchargement.
- Broche 8: L'alimentation (tension Vcc).



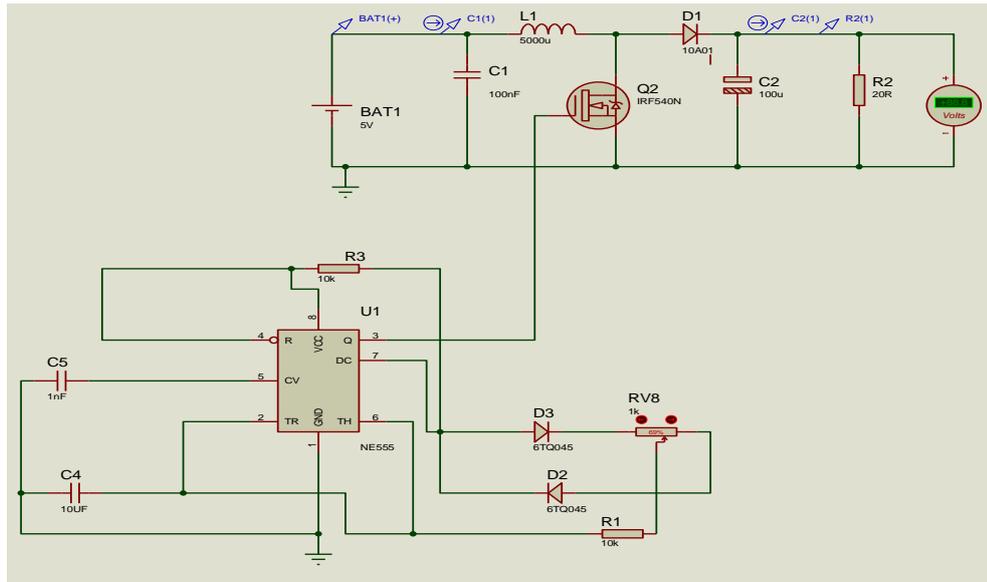
**Figure 4.5** Brochage du circuit intégré NE 555.

Le circuit NE555 peut générer un signal PWM sur sa sortie (broche 3), et la période de ce signal rectangulaire est déterminée par l'expression suivante:

$$T = T_H + T_L = \ln(2) (R_1 + R_2)C + \ln(2) R_2C = \ln(2) (R_1 + 2R_2)C \quad (4.19)$$



**Figure 4.6** Signal de sortie du circuit NE 555.



**Figure 4.7** Schémas d'un convertisseur Boost avec NE555.

Le tableau ci-dessous montre les valeurs calculées théoriquement et celles trouvées en simulation.

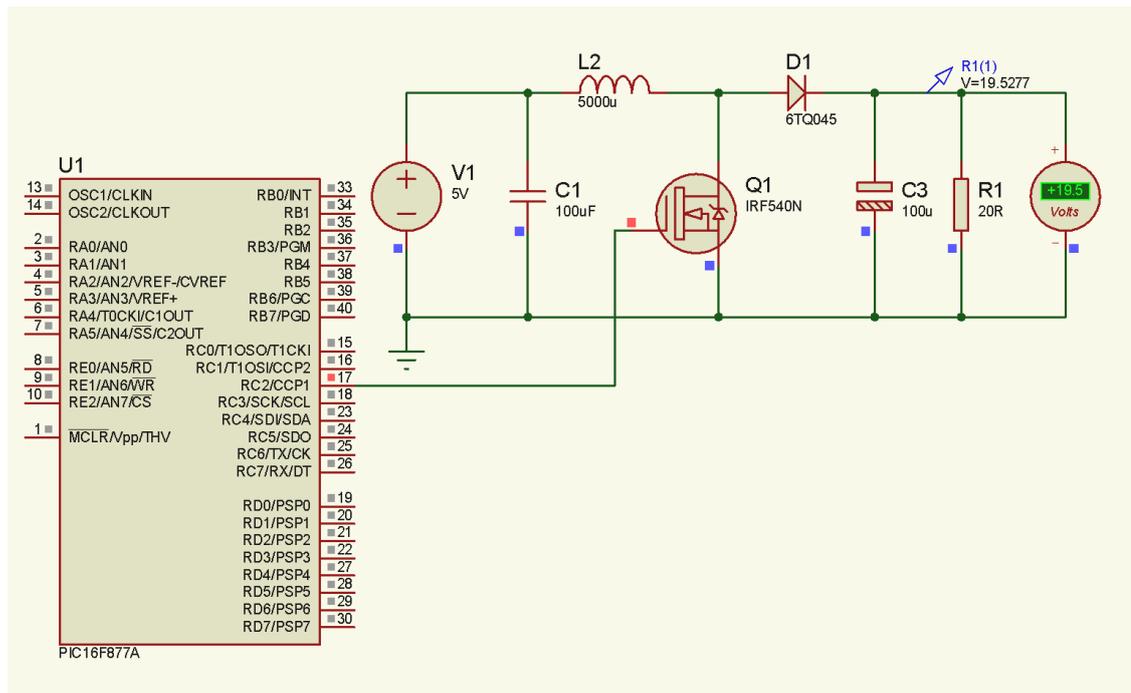
$\alpha$	$V_S$ théorique	$V_{S\text{simulation}}$	$R_v$
10	5.5v	5.7v	1k
25	6.66v	7.1v	2k
30	7.14v	7.78v	2.5k
50	10v	11.5v	3k
75	20v	18.8v	3.5k

**Tableau 4.3** Comparaison entre les valeurs théoriques et celles de simulation.

### 5. Simulation d'un convertisseur Boost avec un microcontrôleur PIC 16F877A

L'utilisation du circuit intégré NE555 pour générer un signal PWM pour commander le convertisseur Boost est une solution facile et pas coûteuse. Cependant, la détermination d'une période souhaitée (un rapport cyclique souhaité) en utilisant le circuit NE555 nécessite d'utiliser des résistances et des condensateurs avec des valeurs bien spécifiques, qui n'est pas facilement faisable.

On pourrait utiliser un microcontrôleur pour générer un signal PWM avec un rapport cyclique voulu sans besoin d'utiliser des résistances ou des condensateurs. La figure ci-dessous montre un convertisseur Boost commandé par un microcontrôleur PIC 16F877A.



**Figure 4.8** Schéma d'un convertisseur Boost avec un microcontrôleur ( $\alpha = 75\%$ ).

La programmation du microcontrôleur PIC 16F877A a été faite avec le compilateur mikroC PRO for PIC qui est basé sur le langage de programmation C++.

```
Void main ()
```

```
{
```

```
TRISC.F2 = 0; // RC2 est configuré comme sortie.
```

```
PWM1_Init(25000); // initialisé le module pwm1 à une fréquence 25 kHz.
```

```
PWM1_Start ();
```

```
For (;;) // Boucle infinie. on aurait pu aussi écrire while(1).
```

```
{
```

```
PWM1_Set_Duty(127); // La valeur du rapport cyclique est déterminée par la relation suivante:(percent*255)/100.
```

```
}
```

```
}
```

## 6. Commande d'un convertisseur Boost par un régulateur PID

### 6.1. Le régulateur PID :

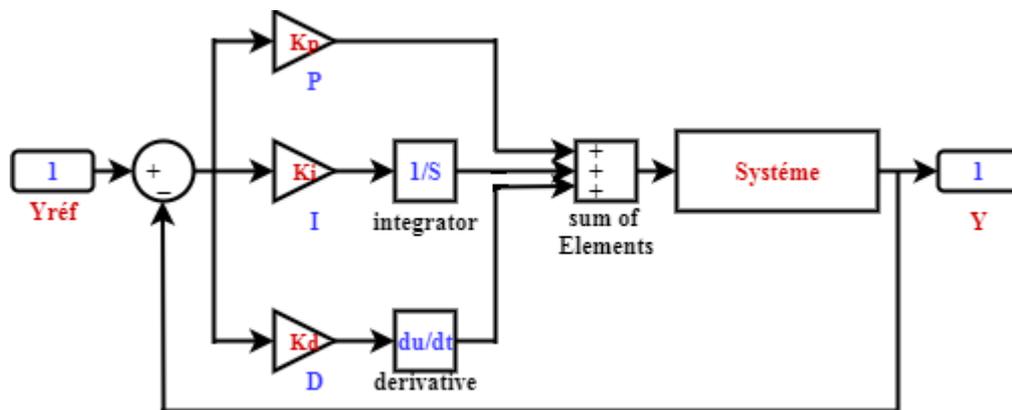
Le régulateur ou correcteur PID (proportionnel, intégral, dérivé) c'est le contrôleur le plus utilisé dans plusieurs domaines, surtout dans le domaine industriel grâce à ses performances et son faible coût [38,39].

Le régulateur PID est un contrôleur qui délivre un signal de commande à partir de la différence entre la consigne (sortie désirée) et la mesure.

Le correcteur PID agit de trois manières:

- \* Action proportionnelle : l'erreur est multipliée par un gain  $k_p$ .
- \* Action intégrale : l'erreur est intégrée et divisée par un gain  $k_i$ ;
- \* Action dérivée : l'erreur est dérivée et multipliée par un gain  $k_d$ .

Il existe plusieurs architectures possibles pour combiner les trois effets (série, parallèle ou mixte). La figure ci-dessous montre la structure PID parallèle qui agit sur l'Erreur.



**Figure 4.9** Schéma équivalent d'un régulateur PID.

L'expression mathématique continue de la loi de commande d'un régulateur PID s'écrit comme suit:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \tag{4.20}$$

L'expression discrète de la loi de commande d'un régulateur PID :

$$u(k) = u(k - 1) + K_p(e(k) - e(k - 1)) + K_i e(k) + K_d(e(k) - 2e(k - 1) - e(k - 2)) \tag{4.21}$$

Le tableau ci-dessous montre l'effet de chaque paramètre d'un régulateur PID sur le temps de monté, le temps de réponse et le dépassement ainsi que la stabilité.

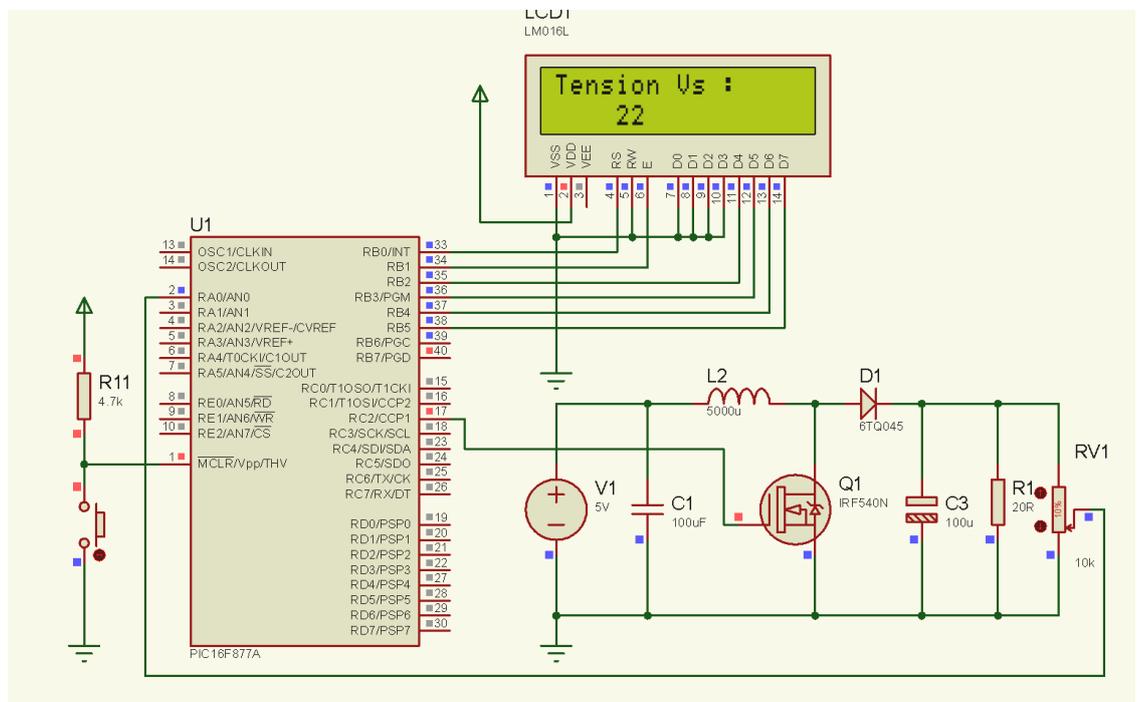
Paramètres	Temps de monté	Temps de réponse	Dépassement	Erreur en régime permanent	Stabilité
$k_p$	Diminue	Diminue	Augmente	Diminue	Dégrade

$k_i$	Diminue	Augmente	Augmente	Elimine	Dégrade
$k_d$	Petit changement	Diminue	Diminue	Aucun effet en théorie	Améliore la stabilité

**Tableau.4.4** Effet de chaque paramètre d'un régulateur PID.

**6.2. Commande d'un convertisseur DC /DC par un régulateur PID avec le pic 16f877a**

Le convertisseur Boost est commandé par un régulateur PID classique afin d'avoir une tension de 24 V à la sortie du convertisseur. Le régulateur PID mesure l'erreur entre la sortie du convertisseur et la tension désirée 24 V pour générer un signal PWM approprié. Ce signal PWM généré par le régulateur PID va commander l'interrupteur (transistor MOSFET IRF540N) du convertisseur Boost. Le schéma ci-dessous montre la commande d'un convertisseur Boost en utilisant le microcontrôleur PIC 16F877.



**Figure 4.10** schéma d'un convertisseur DC/DC par un régulateur PID avec le pic 16f877a.

Le code de programmation du microcontrôleur PIC 16F877:

```
// Configuration des PINs de l'afficheur LCD.

sbit LCD_RS at RB0_bit;

sbit LCD_EN at RB1_bit;

sbit LCD_D7 at RB5_bit;

sbit LCD_D6 at RB4_bit;

sbit LCD_D5 at RB3_bit;

sbit LCD_D4 at RB2_bit;

sbit LCD_RS_Direction at TRISB0_bit;

sbit LCD_EN_Direction at TRISB1_bit;

sbit LCD_D7_Direction at TRISB5_bit;

sbit LCD_D6_Direction at TRISB4_bit;

sbit LCD_D5_Direction at TRISB3_bit;

sbit LCD_D4_Direction at TRISB2_bit;

void main ()

{

float Vs, Vref = 24; // Vs est la tension de sortie du convertisseur Boost

// Vref est la tension de référence.

// Déclaration des paramètres du régulateur PID.

float uk ,ek ,uk_1=0, kp=5, ki=0.1, kd = 0.1,ek_1=0.1, ek_2 = 0;

char text[16];

TRISB = 0x00; // PORTB est configuré comme sortie.

TRISC.f2 = 0; // RC2 est configuré comme sortie.

TRISA.f0 = 1; // RA0 est configuré comme entrée.
```

```
LCD_init(); // Initialisé l'afficheur LCD

LCD_cmd(_LCD_CURSOR_OFF);

pwm1_init(25000); // initialisé le module pwm1 à une fréquence 25khz.

pwm1_start ();

ADC_Init ();

while (1) // Boucle infinie.
{
    Vs = ADC_Read(0); // lire et convertir la tension venant du convertisseur Boost en
    valeur numérique.

    //  $V_x = (R_x/RT)*V_s \Rightarrow V_s = V_x*RT/R_x = V_x*RT/0.1*RT = 10*V_x$ 

    Vs = Vs*5/1024; // convertir valeur numérique de Vs en valeur analogique en la
    multipliant par le quantum q = 5/1024.

    Vs = Vs*10; // Multiplié la tension Vs par 10 (Vs a été déjà divisée par 10, puisque
    le PIC ne supporte pas une tension supérieure à 5 V).

    ek = (Vref - Vs); // Mesure l'erreur entre la tension de référence et la tension venant
    du convertisseur Boost.

    // La loi de commande du régulateur PID.

    uk = uk_1 + kp *(ek - ek_1) + ki*ek + kd*(ek - 2*ek_1 + ek_2);

    // La mise à jours des paramètres.

    uk_1 = uk ;

    ek_1 = ek ;

    ek_2 = ek_1 ;

    uk = uk*255/100; // Convertir le niveau de tension en rapport cyclique 0 --> 255

    if (uk < 0)
    {
        uk = 0;
    }

    else if (uk > 255)
```

```
{uk = 255;  
}  
PWM1_set_duty(uk); // La valeur du rapport cyclique.  
delay_ms(25);  
inttostr(Vs,text);  
LCD_out(1,1,"Température :");  
LCD_out(2,1,text);  
}  
}
```

## 7. Conclusion

Ce chapitre était consacré à la commande d'un convertisseur boost par un microcontrôleur PIC 16F877. Pour ce faire, on a fait appel à un logiciel de simulation Isis Proteus qui permet d'implémenter des circuits électroniques. La programmation du microcontrôleur a été faite à l'aide du compilateur mikroC (il existait aussi des autres compilateur qui tels que: mikroPascal et mikroBasic) . Les résultats de simulation nous montrent qu'il est facile et efficace de commander un convertisseur DC-DC par un microcontrôleur que d'utiliser des autres circuits tel que le circuit NE555 (le circuit NE555 nécessite d'utiliser des résistances et des condensateurs avec des valeurs bien déterminées pour avoir un rapport cyclique souhaité).

Les résultats de simulation montrent que le régulateur PID peut être utilisé pour commander le convertisseur Boost, cependant, ce régulateur ne peut avoir des meilleurs résultats qu'avec des bons valeurs des paramètres du régulateur  $k_p$ ,  $k_i$  et  $k_d$ . Pour remédier à ce problème, on peut utiliser des techniques intelligentes permettant de bien déterminer les paramètres du régulateur PID.

## **Conclusion général**

Le travail présenté dans ce mémoire porte sur la commande d'un convertisseur DC-DC (boost) par un microcontrôleur pic 16F877. Dans un premier temps, on a présenté les différents types des convertisseurs DC-DC que sont : le hacheur dévolteur (convertisseur Buck), le hacheur survolteur (convertisseur boost) et le hacheur-survolteur (convertisseur Buck- Boost).

Dans notre travail, on a choisi d'utiliser le convertisseur Boost afin d'avoir une tension continue de 24V à partir d'une source de tension continue de 5V. Pour ce faire on a fait appel au logiciel de simulation ISIS PROTEUS et au compilateur mikroC. Ces deux logiciels nous ont permis d'implémenter notre circuit. Les résultats de simulation nous montrent qu'on peut commander les convertisseurs DC-DC par microcontrôleur via un régulateur PID. Cependant les performances de ce régulateur reposent sur le bon choix des valeurs de ses paramètres. Pour remédier à ce problème, on pourrait utiliser des techniques intelligentes telles que : la technique de la logique floue , les réseaux de neurones artificiels etc.

## Bibliographie

- [1] A. Belkhir, « commande floue d'un convertisseur statique DC/DC », ingénieur d'état en électrotechnique, Béjaia, 2005.
- [2] A. Saifi, « alimentation d'un moteur à courant continu à l'aide d'un hacheur commandé par un microcontrôleur arduino », mémoire de master, université de Batna 2, 2016.
- [3] A. Souaci « commande robuste d'une machine à induction par mode glissant d'ordre supérieur », mémoire de magister, université de batna.2013.
- [4] S. PETIBON « nouvelles architectures distribuées de gestion et de conversion de l'énergie pour les applications photovoltaïques », université de Toulouse, 2009.
- [5] H .Sahraoui « modélisation et commande des convertisseurs DC-DC utilisés dans les systèmes photovoltaïques (théorie et expérimentation) », thèse de doctorat, université de Batna 2, 2016.
- [6] A .Abdelhak « propriétés électriques des composants électroniques minéraux et organiques, conception et modélisation d'une chaîne photovoltaïque pour une meilleure exploitation de l'énergie solaire ». Thèse de doctorat, université Toulouse, 2006.
- [7] a. Rufer, p. Barrade « cours d'électronique de puissance conversion DC/DC », octobre 2001, EPFL, 90 pages. École polytechnique fédérale de LAUSANNE.
- [8] S.Ghandour « conception de convertisseurs dc/dc à base de mems », thèse de doctorat, université de Grenoble, 2011.
- [9] <http://denis.rabaste.free.fr/>
- [10] présentation de la gamme <<Mid-range des microcontrôleur PIC16F877A>>.
- [11] C.Tavernier : «Microcontrôleurs PIC 10,12,16 'description et mise en œuvre' » DUNOD, Collection TE,2007.
- [12] [www.memoireonline.com](http://www.memoireonline.com).
- [13] [www.memoireonline.com](http://www.memoireonline.com).
- [14] <https://informatique-et-electronique.fr>.
- [15] H.ABIDI:«Carte de développement pour micro contrôleur de la famille PIC » 2006 .
- [16] B.Rjeb, R.Waz : « Projet de réalisation d'une maquette didactique a base de pic 16F877 »2007.

- [17] C. Tavernier : « Les microcontrôleurs PIC ‘recueil d’application’ » 3eme edition, DUNOD, Collection TE ,2005.
- [18] <https://www.researchgate.net>.
- [19] <https://denis.rabaste.free.fr>.
- [20] [www .electronique-mixte.fr](http://www.electronique-mixte.fr).
- [21] B. Béghyn « Les microcontrôleurs pic » Lavoisier 2003.
- [22] [https:// www.scribd .com](https://www.scribd.com).
- [23] le site web: [http:// www.univ-usto.dz/images/coursenligne/Introduction\\_Isis\\_Proteus.pdf](http://www.univ-usto.dz/images/coursenligne/Introduction_Isis_Proteus.pdf)
- [24] le site web: <http://dspac.univ-osto.dz>
- [25] le site web: <http://dSPACE.univ-tlemcen.dz>
- [26] le site web: <http://www.Isis.org/dea/M6optionD/Exp-GL41-SADT.Pdf>
- [27] le site web: <http://fr.farnell.com/microchip/pic18f8520-i-pt/microcontroleur-8-bits-flash-cms/dp/9762>
- [28] le site web: HP: [projet .eu.org](http://projet.eu.org).
- [29] le site web: <http://dspac.univ-osto.dz>
- [30] le site web: <http://dSPACE.univ-tlemcen.dz>
- [31] le site web: <http://fractale.gectif.net>
- [32] le site web: <http://www.irsn.fr>.
- [33] le site web: <http://www.univ-bejaia.dz>
- [34] le site web: <http://biblio.univ-antananariv.mg>
- [35] le site web: [Contact.multipower.fr](http://Contact.multipower.fr) (Proteus design suite 7-labcentre Electronique – (2007) Traduction multipower 1999-2007.
- [36] M.Abdou, N.Amazi. (2016). Conception et réalisation d'un procédé commandé par PC avec application au traitement d'image. Mémoire de master, UMBB, FHC.
- [37] Prouvost Patrick, Automatique contrôle et régulation :cours et exercices corrigés ,dunode 2010 (ISBN 9782100547777)
- [38] Henri Bourlés, Systèmes linéaires : De la modélisation à la commande, paris, Hermes Science publishing ,2006,510p.(ISBN 2729875352).
- [39] F.de Carfort, C Foulard (préf.R.Perret), Asservissements linéaires continus –Maitrise d'E.E.A
- c3-Automatique , paris , DUNOD UNIVERSIT2 ,1971,164p.

