

## Introduction :

Notre robot mobile est un ensemble de mécanique associée à l'électronique. La mécanique consiste à construire une base roulante qui pourra accueillir l'électronique qui permettra la commande de la base.

Ce chapitre consiste à présenter la partie matérielle du robot, et expliquer les différentes étapes de la conception et la réalisation de notre robot mobile.

## III.1 .La structure mécanique :

### III.1.1.Le châssis :

Le châssis est une base sur laquelle sont disposés les éléments de notre robot mobile, il doit être solide, léger et il doit avoir une forme adaptée pour notre application.

La forme que nous avons choisie est circulaire pour faciliter la programmation et le déplacement de notre robot car si notre base est en face d'un obstacle cette forme lui permet de pivoter sur lui-même.

### III.1.2. La structure :

Pour notre application nous avons utilisé un châssis en plexiglas ayant les dimensions suivantes :

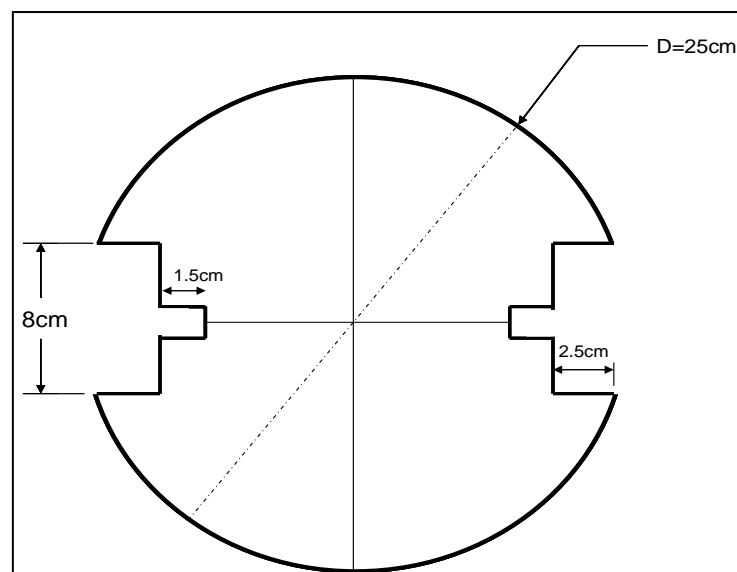


Figure. III.1: Dimension et forme du châssis

### III.1.3.Les roues :

Notre base est équipée de deux roues motrices et une roue folle en caoutchouc. Les roues motrices possèdent les caractéristiques suivantes :

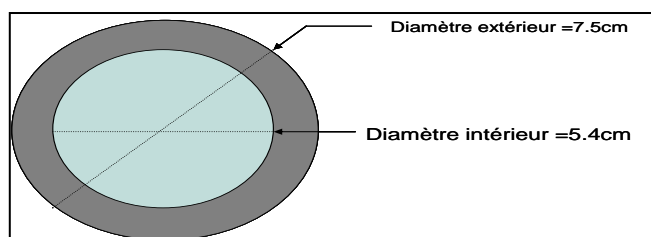


Figure. III.2 : Roue motrice

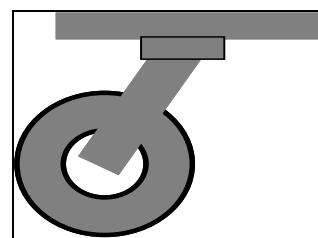
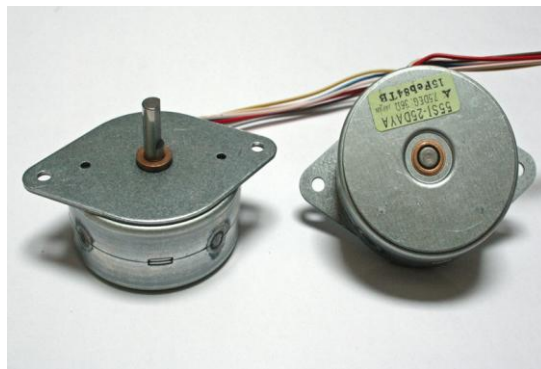


Figure. II .3: Roue folle

**III.1.4.Les moteurs :**

Pour notre application on a choisi des moteurs pas à pas avec les caractéristiques suivantes:

- Les deux moteurs ont un diamètre de 4 cm et une profondeur de 2.5 cm.
- Résistance interne de 115  $\Omega$ .
- Une tension d'alimentation de 12 volts à 36 volts.
- Un courant nominal de 0.25 A.
- 48 pas par tour.
- 6 fils : deux pour l'alimentation (rouge), le jaune et l'orange pour la première bobine, le noir et le marron pour la deuxième bobine.
- Le moteur est du type pas à pas unipolaire
- La gamme de fréquence pour laquelle le moteur tourne de façon continue 100 Hz à 15 Hz.



**Figure. III.4: Nos moteurs pas à pas utilisés**

**III.1.4.1.Calcul la vitesse de notre robot :**

La vitesse du robot est proportionnelle au diamètre des roues utilisées, nous avons un pas qui correspond à 10 ms, donc 48 pas (un tour) correspondent à 480 ms.

La vitesse du robot est alors déterminée de la façon suivante :

$$1 \text{ tour} = \pi * d = 3.14 * 7.5 = 23.55 \text{ cm}$$

Avec d : le diamètre de la roue.

Nous avons un pas qui correspond à une distance de 0.4906 cm dans une période de 1/f.

$$\text{Donc la vitesse} = 0.4906 * f \text{ cm/s.}$$

**III.1.4.2.Le poids de notre robot :**

Le poids est un facteur important pour notre application car il joue le rôle d'un couple résistant pour les deux moteurs, plus le poids est grand plus la vitesse est lente.

**III.2.La structure électronique :**

Pour réaliser n'importe quelle carte électronique, il faut d'abord étudier ses composants et son environnement logiciel qu'on veut employer pour gérer cette carte.

Dans cette partie on décrit les trois cartes réalisé :

- Carte mère.
- Carte de puissance.
- Carte d'alimentation.

Le schéma suivant présente la synoptique de notre robot mobile :

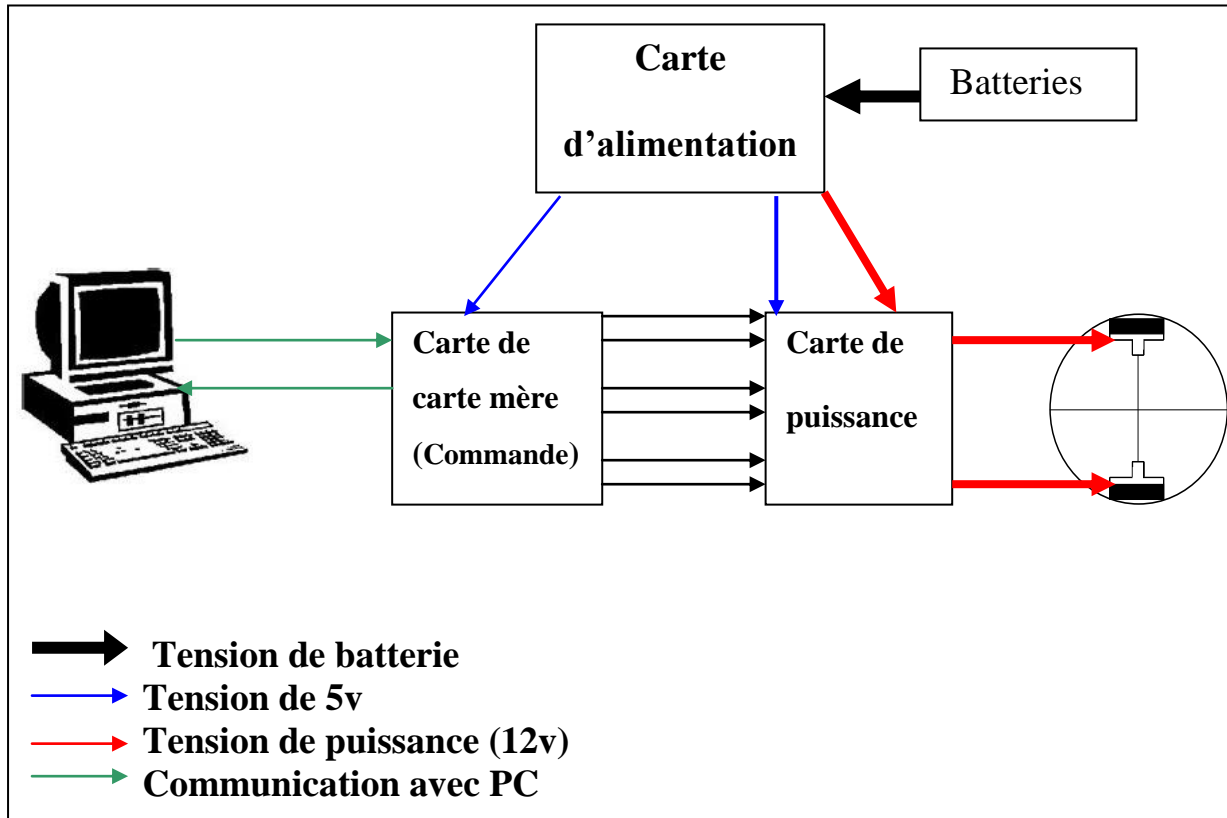


Figure. III.5: Synoptique de notre réalisation

### III.2.1. La carte mère (de commande) :

Notre carte de commande est basée sur un microcontrôleur **PIC16F877** qui est le cœur de notre carte et un circuit intégré **MAX232** qui permet à la communication série.

Cette carte permet de commander nos deux moteurs, et aussi la communication série entre le PC et le robot.

#### III.2.1.1. Présentation des PIC :

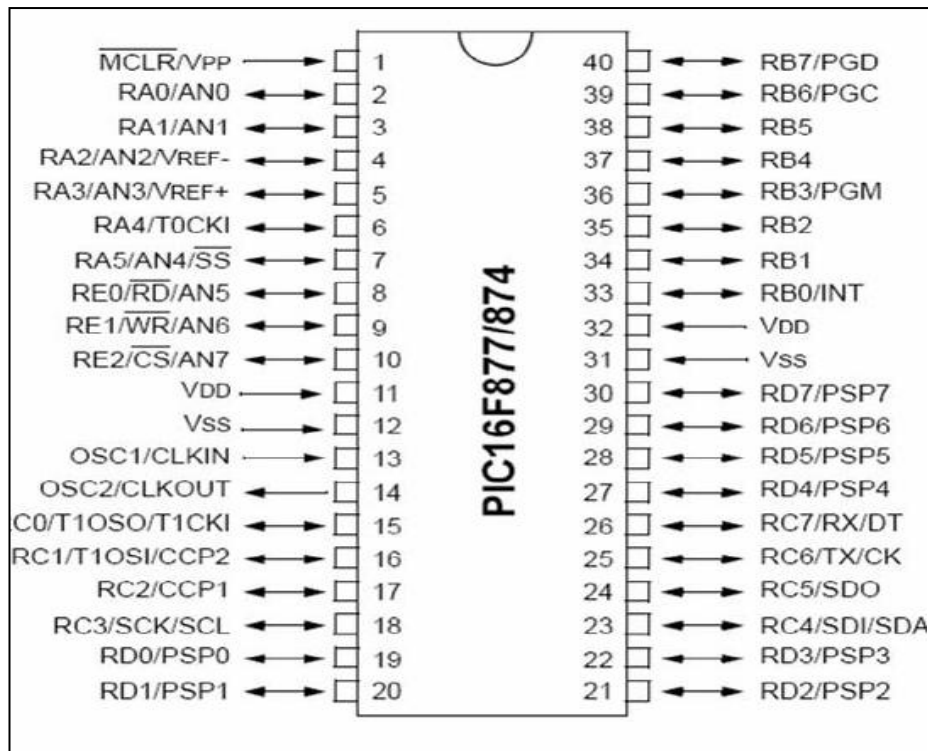
Les PIC 16F877 est un microcontrôleur de type *Mid-range*, ils possèdent un jeu d'instructions réduit qui caractérise les circuits RISC (Reduced Instruction Set component). Les circuits RISC sont caractérisés par leur rapidité d'exécution.

Le 16F877-04 veut dire :

- 16 : c'est un PIC de la famille Mid-Range.
- F : pour indiquer que la mémoire est de type flash.
- 877 : identification du PIC.
- 4 : fréquence d'horloge que le PIC peut supporter.

**III.2-1.2. Brochage du PIC 16F877 :**

Comme pour tout circuit intégré, chacune de ses broches a une ou plusieurs fonctions qui sont résumées par un sigle mnémotechniques. La figure.III.2.1.1 présente le brochage du PIC 16F877 qui dispose de 40 broches montées sur un boîtier DIL.



**Figure. III.6: Brochage du PIC 16F877**

**III.2.1.3. Description du PIC 16F877-04 :**

Le PIC qu'on a utilisé, est le 16F877-04 qui fait partie de la famille MID\_RANG (mots à 14bits), possède les caractéristiques suivantes :

- Une unité arithmétique et logique (ALU).
- Une horloge de 4Mhz.
- Une mémoire programme de type EEPROM flash.
- Une mémoire RAM de données de 368 octets.
- Une mémoire EEPROM de 256 octets.
- 5 Ports d'entrées/sorties :
  - ✓ Port A : 6 pins.
  - ✓ Port B : 8 pins.
  - ✓ Port C : 8 pins.
  - ✓ Port D : 8 pins.
  - ✓ Port E : 3 pins.
- 3 Timers avec leur préscale TMR0, TMR1, TMR2.
- Deux modules de comparaisons et de capture CCP1, CCP2.
- LE WATCHDOG.
- 1 Convertisseur analogique/numérique 10bits de 8 canaux.
- Récepteur/Transformateur série synchrone et asynchrone USART.
- SSP, Port série synchrone.

III.2.1.4.L'Horloge :

Pour fonctionner le PIC 16F877 doit être soit connecter à un circuit d'horloge.

Les PIC disposent d'un oscillateur interne nécessitant une adaptation à un circuit externe. Les pins OSC1 et OSC2 sont les entrées de l'oscillateur. Pour cela on y trouve différents types d'horloges à Quartz ou à cellule RC, qu'on peut classer suivant leurs gammes de fréquences :

- **LP :** à Quartz, la plage de fréquence est comprise entre 32KHz et 200KHz.
- **XT :** à Quartz, la fréquence est comprise entre 100KHz et 4 MHz
- **RC :** à oscillateur RC fonctionnant jusqu'à 4MHz, on note que la stabilité de fréquence est plus fiable que les versions à quartz.
- **HS :** à Quartz fonctionnant de 4MHz jusqu'à 20MHz.

Il existe trois schémas différents de connexion des éléments externes à l'oscillateur.

Figure.III.2.1.3 représente ces trois schémas, sachant que celui de (a) est le plus utilisé. Le Datasheet de chaque PIC fournit les valeurs des composants correspondants.

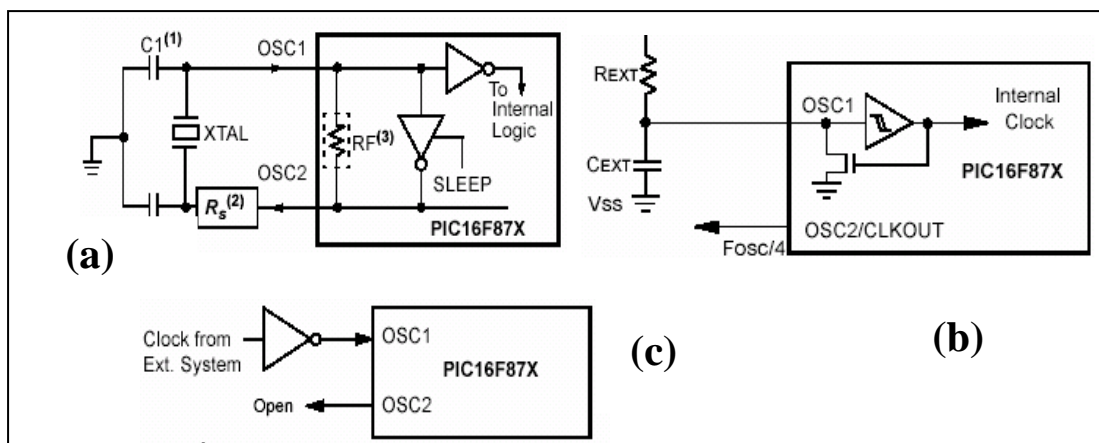


Figure. III.7: Différents connexions à l'oscillateur

III.2.1.5.Structure interne du PIC 16F877 :

L'architecture interne présente les principaux organes qui composent le microcontrôleur (Figure. III.2.1.6). Le but ici n'est pas d'expliquer chaque organe mais de permettre de visualiser l'ensemble pour mieux comprendre le fonctionnement du microcontrôleur.

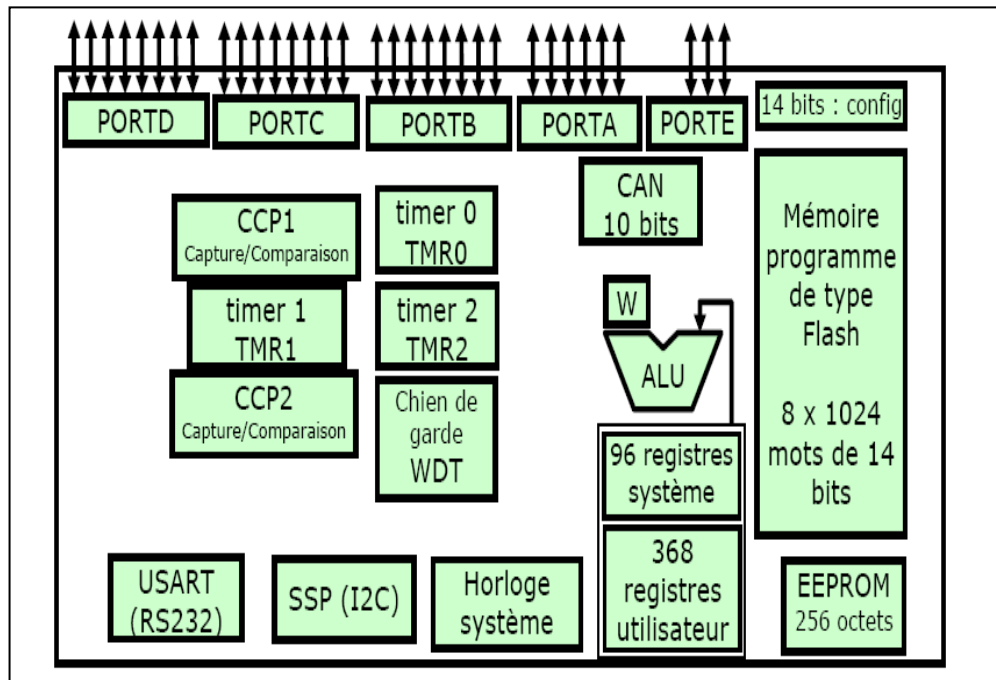


Figure. III.8 : Les éléments constituant 16F877

III.2.1.6.L'organisation de la mémoire :

- Le PIC16F877 a une mémoire programmée (flash) de 8k octets, elle contient le code binaire sur 14 bits des instructions que doit exécuter le microcontrôleur.
- Le PIC16F877 contient une RAM de taille 368 octets divisée en 4 banques de 128 registres. Ces registres servent à contrôler et configurer les différents modules (Comme le la figure

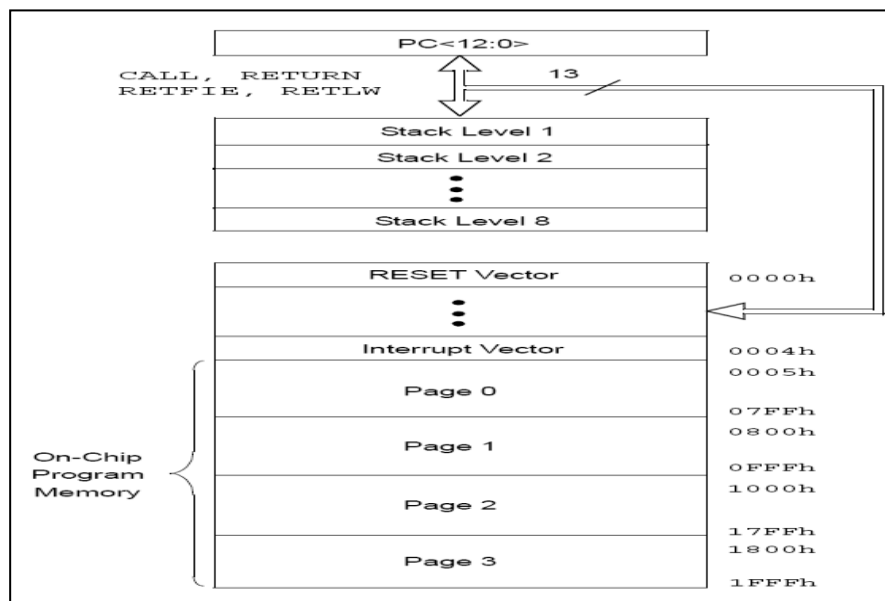


Figure. III.9 : Organisation de la mémoire

**III.2-1.7. Présentation des modules utilisés :**

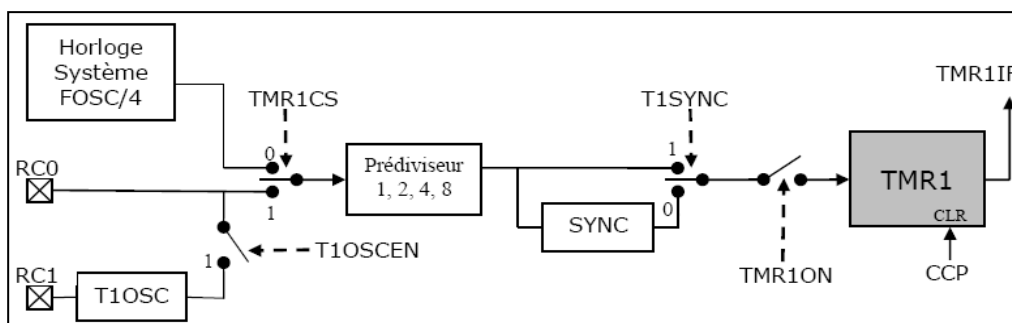
Les modules utilisés pour notre application sont :

- Le Timer1 : La fréquence du signal de commande est lente donc on a besoin d'un Timer de 16 bits, c'est pourquoi nous avons utilisé ce module dans notre application au lieu du Timer0 et du Timer2 qui est utilisé pour la génération de la commande PWM.
- L'USART : Pour gérer la communication série entre le PC et le PIC.

**III.2.1.7.1. Timer 1 :**

Timer1 est un Timer/compteur 16bits accessible en lecture/écriture par intermédiaire des registres 8bits TMR1H et TMR1L qui constituent sa partie haute et sa partie basse.

Le Timer 1 permet de générer une interruption une fois le débordement effectué, son prédiviseur ne permet qu'une division maximale de 8.



**Figure. III.10 : Schéma interne de Timer1**

**Le registre de configuration TICON :**

La gestion du Timer1 est gérée par le registre TCON qu'il active ou désactive.

<b>T1CON :</b>	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
----------------	---	---	---------	---------	---------	--------	--------	--------

**T1CKPS1, T1CKPS0 :** Control du préscaler

- 00 : Division par 1
- 01 : Division par 2
- 10 : Division par 4
- 11 : Division par 8

**T1OSCEN :** Validation de l'oscillateur associé à TMR1

- 1 : Oscillateur arrêté
- 0 : Oscillateur activé

**T1SYNC :** Synchronisation de l'horloge externe

- 0 : Synchronisation
- 1 : pas de synchronisation

**TMR1CS :** Choix de l'horloge du Timer

- 0 : Horloge système (Fosc/4) : mode Timer
- 1 : Horloge externe : mode compteur

**TMR1ON :** Démarrer /Arrête le timer

- 0 : Timer stoppé
- 1 : Timer en fonctionnement

**III.2.1.7.2.L'USART :**

L'USART (Uiversal Synchrone Asynchrone Receiver Transmitter) est un module de communication série. Il peut être configuré comme module de communication **asynchrone full duplex** ou **synchrone half duplex**.

La communication se fait sur les deux broches RC6/TX et RC7/RX.

**L'USART contient deux ports :**

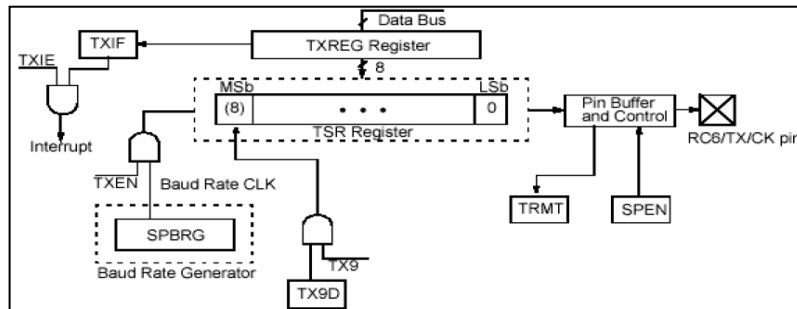


Figure. III.11: Port de transmission

**1. Le port de la transmission est contrôlé par le registre TXSTA :**

CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
------	-----	------	------	---	------	------	------

**CSRC** : Non utilisé.

**TX9** et **TX9D** : Pour utiliser 9 bits.

**TXEN** : Pour valider ou interdire la transmission.

**SYNC** : 0 → mode asynchrone, 1 → mode synchrone.

**BRGH** : Sélectionne le mode haut débit

**TRMT** : Indicateur de l'activité de registre à décalage  
1 → Pas de décalage, 0 → En activité.

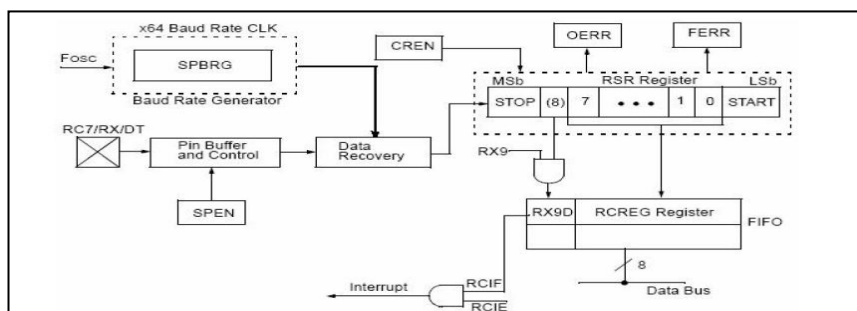


Figure. III.12: Port de réception.

**2. Le port de la réception est contrôlé par le registre RCSTA :**

SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
------	-----	------	------	-------	------	------	------

**SPEN** : Validation du port série (1 → Validé, 0 → Inhibé).

**RX9** : Validation du mode 9 bits (1 → mode 9 bits, 0 → mode 8 bits).



**SREN** : Validation de la réception d'un seul octet.

**CREN** : Validation du mode de réception continue.

**ADDEN** : Validation du mode de réception d'adresse en mode 9bits.

**FERR** : Erreur de synchronisation.

**OERR** : Erreur de débordement du buffer.

**RXD9** : En mode 9 bits, le 9<sup>ème</sup> bits reçue.

Pour que la communication série s'effectue correctement l'utilisation de l'USART doit se dérouler suivant les étapes suivantes ;

**Pour la transmission :**

- 1-Assurer que l'interruption TXI n'est pas validée.
- 2-Configurer la broche TX/RC6 en entrée.
- 3-Configurer le registre TXSTA.
- 4-Initialiser le registre SPBRG pour définir la vitesse de transmission
- 5-Placer la donnée à transmettre dans TXREG.

**Pour la réception :**

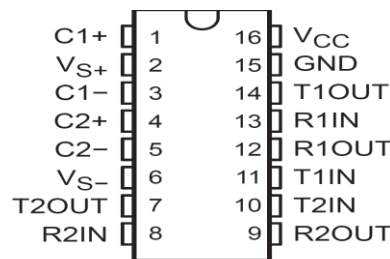
- 1-Assurer que l'interruption RCI n'est pas validée.
- 2-Configurer la broche RX/RC7 en entrée.
- 3-Initialiser le registre SPBRG pour définir la vitesse de la réception.
- 4-Configurer le registre RCSTA.
- 5-Attendre le drapeau RCIF ce qui donne la fin de la réception. [1], [2], [6], [7]

**III.2.1.8.Présentation de la communication série :**

Pour garantir la communication entre la base mobile et le PC, on a utilisé la liaison série RS232. Sur cette liaison les bits d'informations circulent l'une après l'autre sur le même fil. Pour cette liaison les niveaux sont -12V pour 1 logique et +12V pour 0 logique. Ils sont donc totalement incompatibles avec ceux du PIC +5V pour « 1 » et 0V pour « 0 ».

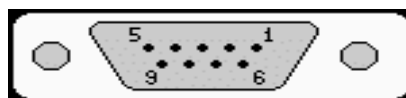
L'utilisation du circuit MAX232 permet d'adapter les niveaux à l'aide d'un convertisseur de tension intégré qui délivre une tension de -12V et +12V en partant d'une alimentation de 5V.

Le MAX 232 va convertir les niveaux bas (0V) en 12V et les niveaux hauts (5V) en -12V. [13]



**Figure. III.13: Brochage du MAX232**

Pour le raccordement entre la sortie du MAX 232 et le PC, on utilise une fiche D9F femelle.



**Figure. III.14 : Connecteur D9F femelle**

La communication série nécessite trois fils au minimum, une masse pour référencier les signaux, un fil émetteur et un fil récepteur. Notre liaison série est en effet full\_duplex, c'est à dire que l'on peut émettre et recevoir en même temps.

La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en BAUDS (1 baud =1 bit/seconde) .Il existe différentes vitesses normalisées : 9600, 4800, 2400,1200...bauds.

Les lignes utilisées dans notre application sont représentées ci-dessous :

PC	PIC
RXD	RC7
TXT	RC6
GND	GND

Tableau 01 : Brochage entre le PIC et le MAX232

III.2.1.9.La carte mère: la figure.III.2.1.9 représente la carte mère :

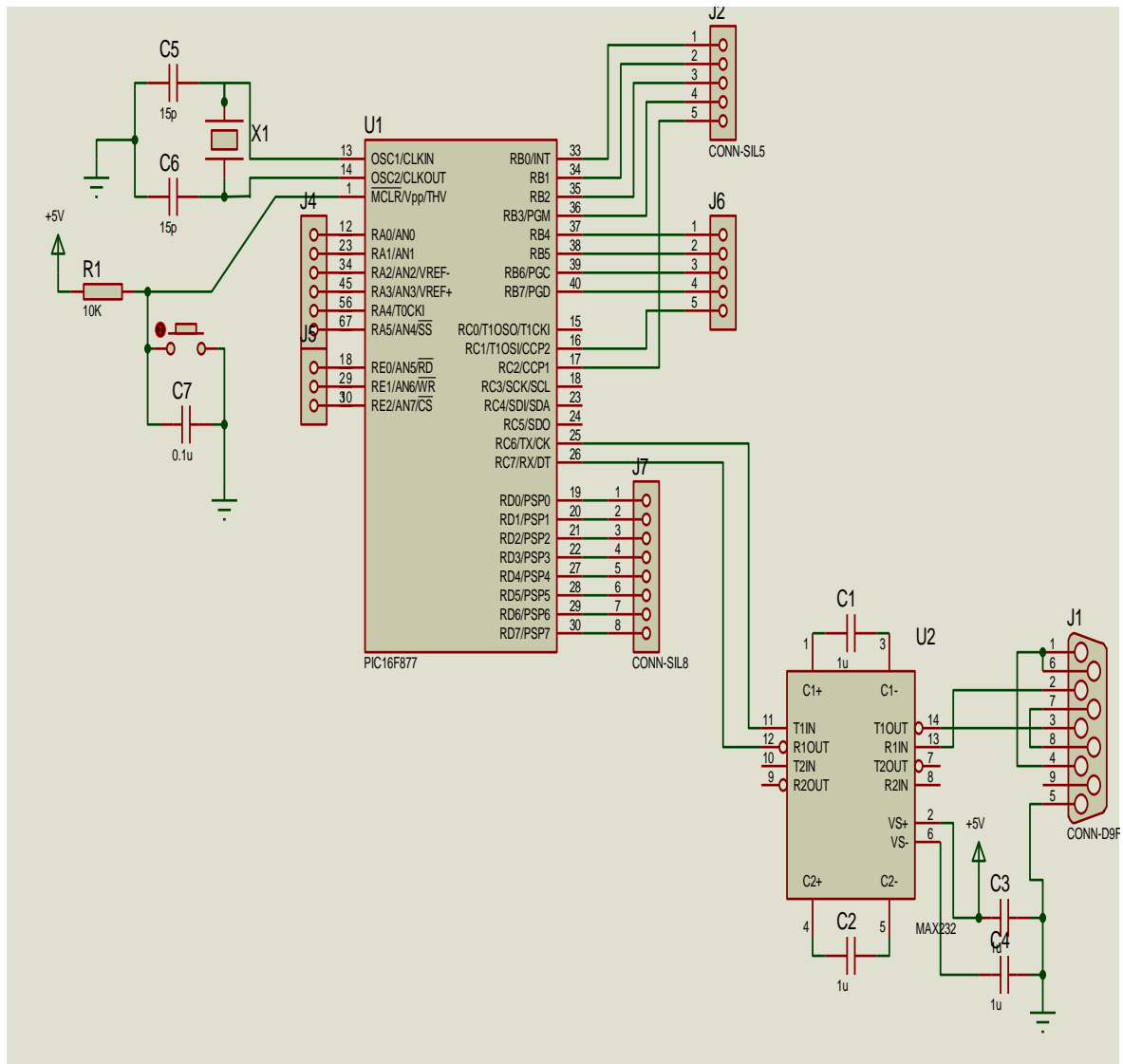
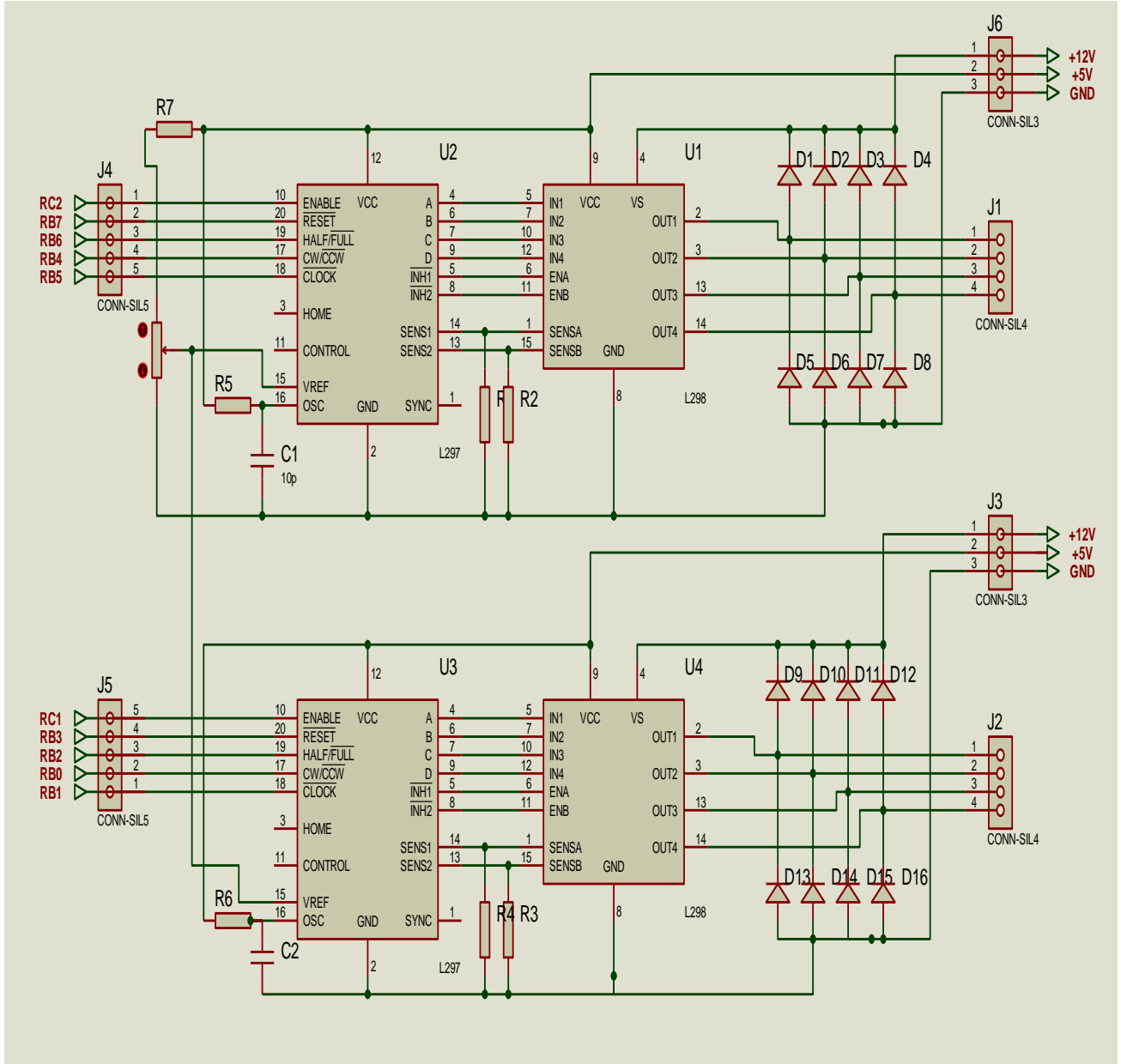


Figure. III.15: La carte mère

**III.2.2.Circuit de puissance :**

Pour notre projet on a utilisé les circuits intégrés L298 et L297 qui sont des composants les plus utilisés pour la commande des moteurs pas à pas. Ils permettent en effet de commander n'importe quel type de moteur qu'il soit unipolaire ou bipolaire.

La figure III.2.2.1 représente la carte de puissance réalisée :



Les valeurs des composants :

$R4=R3=R1=R2=1\Omega$ .

$R5=R6=240K\Omega$ .

$R7= 47K\Omega$ .

$R_v =10K\Omega$  (Résistance variable).

$C1=C2=10nF$ .

J1, J2 connecteurs vers les moteurs.

J3, J6 connecteurs d'alimentation.

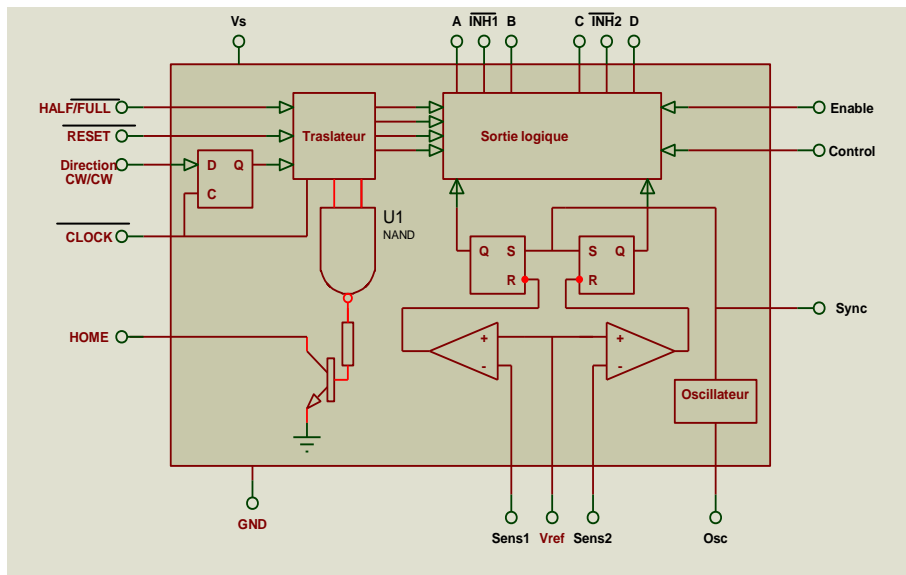
J4, J5 Les entrées du L297.

2 circuits L298.

2 circuits L297.

**III.2.2.1.Circuit L297 :**

Le circuit L297 est un contrôleur de moteur pas à pas, il fonctionne avec un circuit de puissance à double pont. Il suffit de lui fournir les signaux d'horloge (CLOCK : pour avance des pas), de direction et de mode afin de piloter le moteur pas-à-pas. Le L297 génère alors la séquence de commande de l'étage de puissance. [10]



**Figure. III.17: Structure interne du circuit L297**

Le circuit L297 possède deux étages principaux :

- Un translateur (séquenceur) qui génère les différentes séquences de commande.
- Un double hacheur PWM (Pulse With Modulation) qui régule le courant traversant les bobinages du moteur.

Le translateur génère trois séquences différentes. Ces différentes séquences sont déterminées par le niveau logique appliqué sur l'entrée HALF/FULL :

- La commande en mode monophasé : une seule phase alimentée.
- La commande en mode biphasé : Deux phases alimentées.
- La commande en demi-pas : Alternativement une phase puis deux phases alimentées.

Le L297 génère deux signaux d'inhibition (INH1 et INH2) dans les modes monophasés et demi pas. Ces signaux qui sont appliqués directement aux entrées de validation du L298, permettent d'accélérer la décroissance du courant dans les bobinages du moteur lorsque ceux-ci ne sont plus alimentés. Lorsque le L297 est utilisées pour la commande d'un moteur unipolaire, les hacheurs agissent sur ces lignes.

Une entrée «CONTROL » détermine le moment où le hacheur devra agir sur les sorties A, B, C, D ou les entrées INH1 et INH2.

Le L297 contient un oscillateur interne qui commande les hacheurs. Lorsque le courant traversant l'un des bobinages du moteur atteint la tension qui est fixée par les résistances connectées aux entrées SENS1 SENS2, le comparateur correspondant interrompt l'alimentation du moteur jusqu'à la prochaine impulsion de l'oscillateur.

### III.2.2.2.Circuit L298 :

Le circuit intégré L298 est le circuit complémentaire du L297. Il permet la simplification extrême de la construction d'une carte de commande pour moteur pas-à-pas.

Le circuit L298 est un double pont de commande de puissance, il permet l'utilisation d'une tension d'alimentation élevée. Il peut être utilisé aussi pour la commande de deux moteurs à courant continu.

Le schéma interne du L298 contient deux ponts de 4 transistors de puissances commandées par les portes logiques, ainsi que la connexion des résistances de mesure du courant consommée par la charge reliée aux émetteurs de chaque paire de transistors. [10]

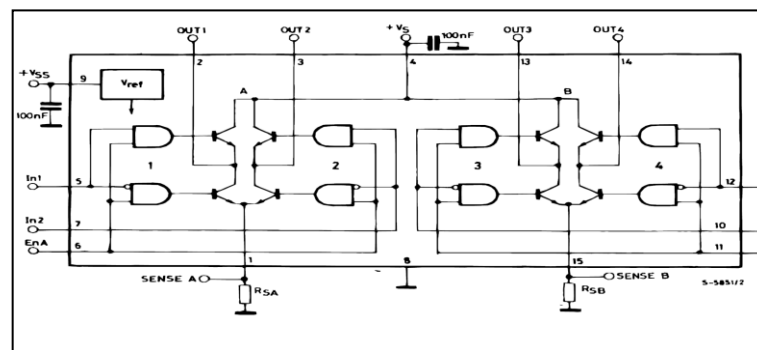
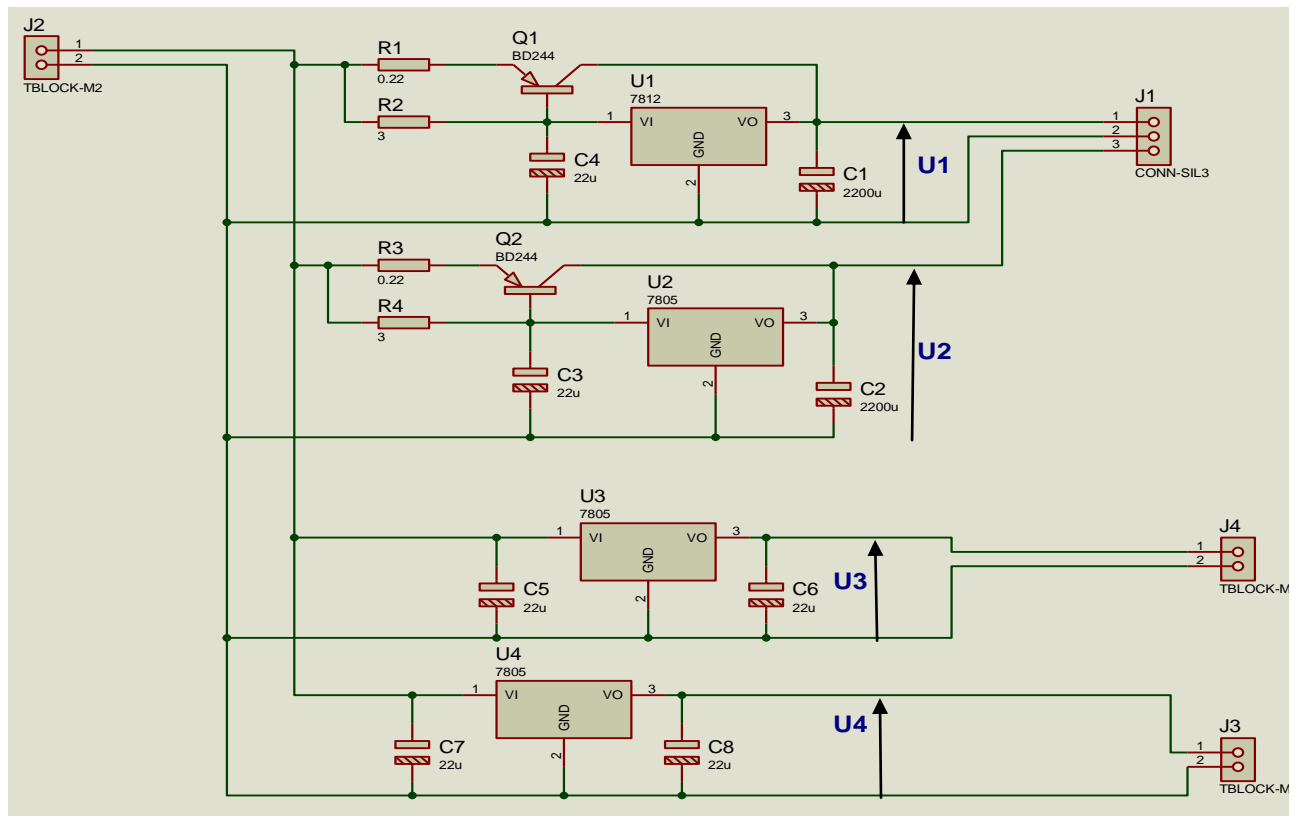


Figure. III.18: Structure interne du circuit L298

III.2.3. La carte d'alimentation :

Cette carte permet d'alimenter toutes les interfaces à partir de la tension de la batterie :  
Le schéma du circuit que nous avons réalisé :



**Figure. III.19: La carte d'alimentation**

Cette alimentation contient :

- $R1=R2=0.22 \Omega$ .
- $R3=R4=3 \Omega$ .
- $C1=C2=2200\mu F$ .
- $C3=C4=C5=C6=C7=C8=22\mu F$ .
- Q1 et Q2 sont deux transistors de type BD244.
- 3 régulateurs de type 7805.
- 1 régulateurs de type 7812.

Notre alimentation peut fournir :

- 2 tensions de 5V courant faible U3 et U4 ne dépassent pas les 500 mA.
- 1 tension de 5V courant élevé U2 jusqu'à 4A.
- 1 tension de 12V courant élevé U1 jusqu'à 4A.

Pour générer les tensions U1 et U2 avec un courant élevé nous avons utilisé les régulateurs 7812 et 7805 équipés de transistors BD244. Les deux transistors BD244 sont montés d'une manière permettent d'amplifier le courant. Ces derniers nous permettent d'atteindre un courant de 4A.

Si le courant consommé à travers R2 et R4 dépasse 0.3A la tension de la base de Q1 ou Q2 atteint 0.6V ce qui entraîne la commutation du ballaste (il devient passant).

La sortie U3 de 5v avec un courant faible est utilisée pour l'alimentation de la carte de commande, le PIC et le MAX232

La sortie U4 de 5 v avec un courant élevé est utilisée pour l'alimentation de la carte de puissance, le L298 et le L297.

**III.2.4.Le brochage entre les différents composants:**

Les deux tableaux ci-dessous présentent les connexions utilisées pour notre application :

PIC 16F877	L297
RB0	CW/CCW
RB1	CLOCK
RB2	HALF/FULL
RB3	RESET
RC1	ENABLE
RB4	CW/CCW
RB5	CLOCK
RB6	HALF/FULL
RB7	RESET
RC2	ENABLE

**Tableau 02 :** Brochage entre Le PIC et le L297

L297	L298	Moteur
A	Input1	Moteur 1
B	Input 2	
C	Input3	
D	Input4	
INH1	Enable 1	
INH2	Enable2	
Sens1	Current sensing A	
Sens2	Current sensing B	
A	Input1	Moteur 2
B	Input 2	
C	Input3	
D	Input4	
INH1	Enable 1	
INH2	Enable2	
Sens1	Current sensing A	
Sens2	Current sensing B	

**Tableau 03 :** Brochage entre le L297 et le L298

**III.2.5.Conclusion :**

Dans ce chapitre nous avons présenté la partie matérielle qui réalise notre robot.Cette partie matérielle est constitué en deux étapes :

La première consiste à construire une structure mécanique qui est composé d’une plateforme, les roues, et les moteurs pas à pas.

Pour la deuxième partie nous avons réalisé la carte de commande pour commander notre robot.