

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUEES
DÉPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Electrique

Spécialité : Informatique Industrielle

THÈME

**Automatisation d'une station de moulage
de sable**

Préparé par :

- ***HARIZ BELGACEM Mohammed***
- ***DERRAR Nacer***

Devant le Jury :

Noms et prénoms

Mr. M. SAHLI

Mr. A. BOUMEDIENNE

Mr. A. TIFFOUR

Qualité

Président

Examineur

Encadreur

PROMOTION 2016

Dédicace

DEDICACES

Je dédie ce travail à ...

A ma très chère mère Messaouda

*Affable, honorable, aimable : Tu représentes pour moi le
symbole de la bonté par excellence, la source de tendresse et
l'exemple du dévouement qui n'a pas cessé de m'encourager et
de prier pour moi.*

*Ta prière et ta bénédiction m'ont été d'un grand secours
pour mener à bien mes études.*

*Aucune dédicace ne saurait être assez éloquente pour
exprimer ce que tu mérites pour tous les sacrifices que tu n'as
cessé de me donner depuis ma naissance.*

A mon très cher Père Ammar

*Aucune dédicace ne saurait exprimer l'amour,
l'estime, le dévouement et le respect que j'ai toujours eu pour vous.*

*Rien au monde ne vaut les efforts fournis jour et
nuit pour mon éducation et mon bien être.*

*Ce travail est le fruit de tes sacrifices que tu as
consentis pour mon éducation et ma formation.*

« Que "ALLAH" vous garde »

A mes très cher sœurs et frère : Baraa, Ayoub, Ramzi.

*A mes très chères amies : Nacer, Okba, Souleymane, Youssef, Yasser,
Oussama, Boubaker.*

Et toute la famille : HARIZ BELGACEM et HAMIDI.

HARIZ BELGACEM Mohammed

Dédicace

DEDICACES

Je dédie ce travail à ...

A ma très chère mère Zohra

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance.

A mon très cher Père Ali

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

« Que "ALLAH" vous garde »

A mes très cher sœurs et frère : Massi ,Abdlhake, Mohammed.

A mes très chères amies : Fouzi, Bilal, Youssef, Slïman, hosin.rachid.

Et toute la famille : Derrar Ahmed et Rachid.

DERRAR NACER

Remerciement

REMERCIEMENT

Nous tenons tout d'abord à remercier le bon " ALLAH " de nous avoir guidés et donné la force et la volonté pour atteindre notre objectif.

Nous remercions nos très chers parents pour leurs soutiens et leurs patiences.

Nous remercions chaleureusement notre encadreur Mr. TIFFOUR Abdelkader pour sa disponibilité, ses précieux conseils et motivations qui nous ont gardés sur le droit chemin afin de réaliser ce modeste travail.

Nous remercions les membres de jury qui nous font l'honneur de présider et d'examiner ce modeste travail.

Et également nos remerciements sont exprimés :

A tous les enseignants de département Génie électrique et Informatique Industriel qui nous ont enrichis de connaissances et de savoir, ainsi Mr B.okba, , ingénieur dans la société sonatrach El oued , nous le remercions pour avoir toujours été présent par ses conseils et ses encouragements.

A tous ceux qui nous ont aidés de près ou de loin dans la réalisation de ce projet de fin d'étude.

Merci



Sommaire

INTRODUCTION GENERALE.....	1
CHAPITRE I : GENERALITE SUR L'ENTREPRISE ALFET	
I.1. Introduction	3
I.1.1. Localisation	3
I.1.2. Schéma architectural	3
I.1.3. Historique	4
I.1.4. Moyens	4
I.1.5. Direction commerciale	5
I.1.6. Direction technique	5
I.1.7. Laboratoire	5
I.1.8. Atelier de modelage.....	5
I.1.9. Atelier de production.....	6
I.2. Qu'est-ce que le moulage	9
I.2.1. Où utilise-t-on le moulage.....	10
I.2.2. L'art de moulage	10
I.2.3. Différents types de moulage.....	10
I.2.4. Le moulage en sable	12
I.2.5. Le moule.....	14
I.2.5.1. Les moules permanents	15
I.2.5.2. Les moules non-permanents	16
I.2.6. Moulage à la machine (mécanique)	16
I.3. Conclusion.....	17
CHAPITRE II : ETUDE DU MICROCONTROLEUR PIC 16F877A	
II.1. Introduction	19
II .2. Définition d'un PIC	19
II .2.1. Classification des Pics de Micro chip	19
II .2.2. Identification d'un PIC	20
II .2.3. Le choix du microcontrôleur PIC 16F877A : Pourquoi le PIC16F877A?	20
II .2.4. Les caractéristiques du PIC 16F877A.....	21
II .2.4.1. Description générale du PIC 16F877A	21
II .2.4.2. Architecture externe du PIC 16F877A.....	22
II .2.4.3. Architecture interne du PIC 16F877A	23
II .2.5. Les memoires	24
II .2.6. Les registres du PIC 16F877A	26
II .2.7. Les interruptions.....	27
II .2.8. Les circuits d'horloges.....	27
II .2.9. Présentation des modules utilisés dans PIC 16F877A	29
II .2.10. Le convertisseur analogique numérique.....	31
II .2.11.L'USART	31
II .2.12. Les modules CCP1 et CCP2.....	32
II .2.13. Domaines d'application des microcontrôleurs.....	32
II .2.14. Les avantages des microcontrôleurs.....	33
II .2.15. Les défauts des microcontrôleurs.....	33
II .3. Langage c pour microcontrôleurs.....	34
II .3.1. Généralité	34
II .3.2. Le langage c du compilateur CCS.....	34
II .3.3. Les règles de bases	34
II .3.4. Les variables et les constantes.....	34

Sommaire

II .3.5. Les opérateurs du langage C	36
II .3.6. Les structures répétitives.....	39
II .3.7. Les fonctions adaptées aux microcontrôleurs PICs.....	40
II .11.Conclusion.....	46

CHAPITRE III : REALISATION D'UNE MACHINE DE MOULAGE DU SABLE

III.1. Introduction sur la machine de moulage de sable	48
III.1.1. Description de la machine architecture.....	49
III.1.2. Description de la machine organisation.....	49
III.1.3. Organigramme machine de moulage de sable	50
III.2. Vue d'ensemble du projet.....	50
III. 2.1. Les différents blocs	50
III.2.2. L'organigramme sur lequel est basé ce programme	51
III.2.3. Grafset machine de moulage de sable	53
III.3. Simulation schéma électronique du système sur proteus.....	54
III.3.1. Schéma sur ISIS pic 16F887A.....	54
III.3.2. Schéma sur ISIS Arduino	55
III.3.3. Explication des schémas	56
III.3.4. Afficheur LCD.....	59
III.3.5. Génération de typon de la carte avec Ares.....	60
III.4. Réalisation schéma électronique du system.....	61
III.5. Conclusion	63

CONCLUSION GENERALE.....	64
---------------------------------	-----------

BIBLIOGRAPHIE.....	65
---------------------------	-----------

ANNEXE.....	66
--------------------	-----------

Liste des figures

Chapitre I : Généralité sur l'entreprise ALFET

Figure I.1 : Organisation de la filiale.....	3
Figure I.2 : Organisation du service de Direction commerciale vente.....	5
Figure I.3 : Organisation des ateliers de production.....	6
Figure I.4 : Architecture de l'unité de fonderie de fonte	7
Figure I.5 : Architecture de l'unité de fonderie de l'acier.....	9
Figure I.6 : Modèle de moulage	9
Figure I.7 : Moulage en sable	12
Figure I.8 : Schéma des opérations de moulage en sable	12
Figure I.9 : Le moule en sable.....	14
Figure I.10 : Moulage en fonderie	15
Figure I.11 : Pièce moulé par injection	15
Figure I.12 : Moulage en Cire perdue	16

Chapitre II : Etude du microcontrôleur pic16F887A

Figure II.1 : Microcontrôleur PIC 16F877A	21
Figure II.2 : Boitier du PIC 16F877A	22
Figure II.3 : L'architecture interne du PIC 16F877A.....	24
Figure II.4 : Les différents ports du PIC16F877A.....	25
Figure II.5 : Oscillateur à quartz	28
Figure II.6 : Connexion d'une horloge externe	28
Figure II.7 : Oscillateur RC	29
Figure II.8 : Schéma du Timer0	29
Figure II.9 : Schéma descriptif du Timer1	30
Figure II.10 : Schéma descriptif du Timer2.....	31
Figure II.11 : Schéma du module de conversion analogique numérique.....	31

Chapitre III: Réalisation d'une machine de moulage du sable

Figure. III.1 : Schéma Machine de moulage de sable	49
Figure. III.2 : Organigramme machine de moulage de sable	50
Figure. III.3 : Schéma synoptique.....	51
Figure. III.4 : Organigramme de programme	52
Figure. III.5 : grafcet machine de moulage de sable.....	53
Figure. III.6 : schéma sur Isis pic16F877A.....	54
Figure. III.7 : schéma sur Isis Arduino.....	55
Figure. III.8 : schéma de puissance sur Isis	56
Figure. III.9 : schéma réel de relais.....	57
Figure. III.10 : schéma interne de relais.....	57
Figure. III.11 : schéma résistance variable	58
Figure. III.12 : Les valeurs des différentes résistances du montage.....	58
Figure. III.13 : Schéma réel de Diode	59
Figure. III.14 : Schéma interne de Diode.....	59
Figure. III.15 : Afficheur LCD avec 2 lignes de 16 caractères.....	59

Figure. III.16 : Schéma du typon..... 61

Liste des Tableaux

Tableau II.1 : Différents circuit de la famille 16F87X..... 20
Tableau II.2 : Broches du PIC 16F877A..... 23
Tableau II.3 : Les interruptions du PIC 16F877A 27
Tableau II.4 : Les types des variables 35
Tableau II.5 : Les opérateurs arithmétiques 38
Tableau II.6 : Les opérateurs de relationnels..... 38
Tableau II.7 : Les opérateurs logiques 38
Tableau II.8 : Les opérateurs binaires 39
Tableau III.1: Composant utilisés dans réalisation..... 61

Introduction générale

Dans les années précédentes .la fabrication de moulage se fait manuellement et le matériel utilisé est très simple. Mais avec le développement des moyens technologiques et informatiques (matériel et logiciel), ont permis de trouver des solutions pour améliorer cette industrie.

En premier lieu on a étudié et conçu un système d'automatisation d'une machine de moulage qui sert à réduire au maximum le temps de moulage du châssis, surtout pour les modèles qui ont beaucoup de difficultés d'exécution.

En agissant sur les boutons poussoir de commande ou sur la valve de régulation on peut commander des opérations comme :

1. Commande du tapis roulant pour le déversement du sable dans le moule à partir de la trémie
2. Vibration du moule pour l'entassement du sable
2. Commande du bras presseur du sable pendant un temps prédéterminé
3. Décochage du moule
4. Programmation du temps de pression

Ensuite nous avons consacré le deuxième chapitre pour étudier les moyens utilisés pour la réalisation de cette automatisation et les caractéristiques de chaque composant.

Après, on a vue les domaines d'application des microcontrôleurs qui sont extrêmement variés et touchent tous les secteurs grand public, industriel, informatique, etc..

Les applications à réaliser par les PIC sont de plus en plus complexes, la programmation en assembleur devient vite très fastidieuse. L'utilisateur est ainsi amené à utiliser les langages "évolués" pouvant simplifier la tâche de programmation.

Enfin le dernier chapitre a été consacré à la réalisation de cette étude qui sert à commander automatiquement la machine de moulage grâce à la carte ARDIUNO qui à remplacé le PIC16F877A pour de nombreux avantages tel que : la simplicité, la fiabilité l'efficacité et la précision des résultats.



Chapitre I

Généralité sur l'entreprise ALFET

I.1. Introduction

<< ALFET dès sa création à partir de 1999 s'est inscrite dans la logique du progrès, en adaptant son organisation, en vue de prendre en charge les préoccupations de son client, à savoir :

- Assurer la qualité de ses prestations.
- Prendre en charge les attentes du client en faisant de lui un partenaire performant.

En plus des actions de rationalisations des coûts, ALFET a mis en place le système de management de la qualité pour s'assurer de la prise en charge des exigences des clients et de la gestion de ses attentes.

Nom de l'entreprise

- * ALFET Algérienne des fonderies de Tiaret

Symbole :



I.1.1. Localisation

Située à Tiaret, Au sud-ouest d'Alger, à 280 km de l'aéroport Abdelhamid BOUSSOUF «Ex Ain BOUCHEKIF» de Tiaret.

I.1.2. Schéma architectural

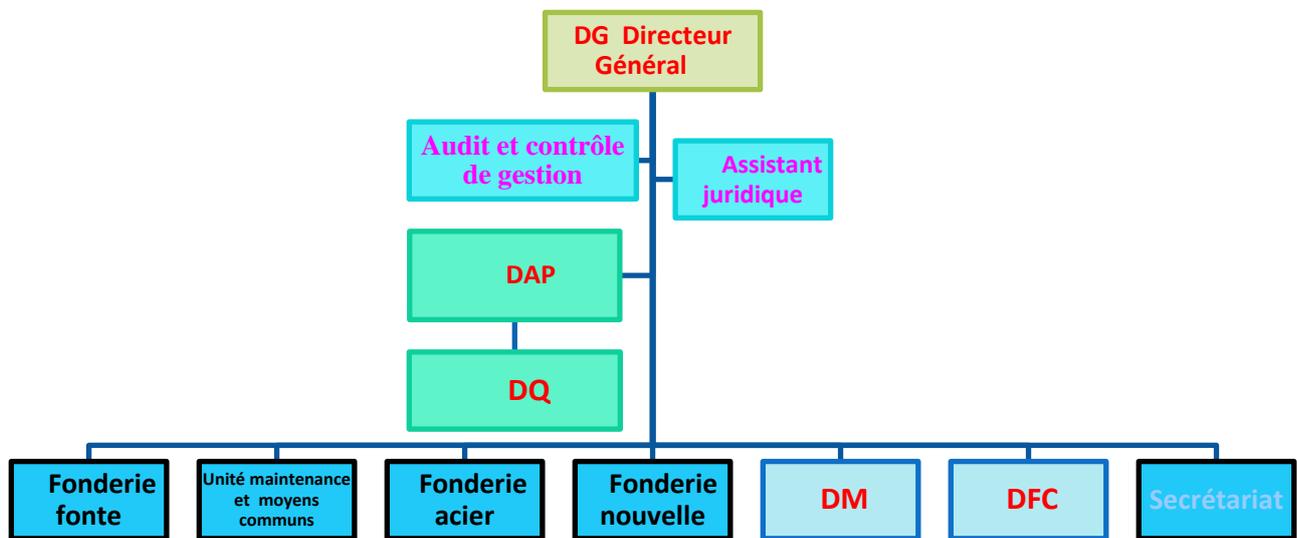


Figure.I.1 : Organisation de la filiale

Légende

- DQ : Direction Qualité
- DM : Direction Marketing
- DFC : Direction finances et comptabilité
- DAP : Direction Administration et personnel

I.1.3. Historique

L'a été créée en 1983 suite à la restructuration organique de la SN- METAL et de SONACOME. Elle fut érigée en société par action (SPA) autonome et dotée d'un conseil d'administration en septembre 1995, suite au processus de restructuration industrielle décidée par le holding public mécanique, elle a été organisée en groupe industriel en décembre 1998, après l'opération de filialisation de ses 3 sites de production.

I.1.4. Moyens

ALFET dispose d'un ensemble d'infrastructures (superficie totale des terrains de 430.000 m² bâtie). Elle est dotée d'installations et de procédés adaptés pour proposer des nuances de métal très variées et fournir des pièces moulées de grande qualité conformes aux exigences et attentes du client.

a) moyens humains

Compte un effectif de 300 agents, reparti par groupe socioprofessionnel :

	Groupe	Nombre	%
Groupe 1	Exécution	200	66.6%
Groupe 2	Maîtrises	56	18.7%
Groupe 3	Cadres	44	14.7%

b) moyens matériels

- * Ateliers de fusion assurée par 4 fours à induction de 10 T, deux 1T et 2 fours à arc de 5T.
- * Ateliers de moulage assurés par 2 chantiers automatiques à sable vert, 2 mécaniques à sable vert et silicate de soude et 2 chantiers manuels au furanique.
- * Installation de traitement et de régénération de sable.
- * Ateliers de noyautage (proceed No-back, Cold box, Co2, Crooning, huile).
- * Laboratoire (spectrométrie, métallographie, chimie, essais physique, sable).
- * Installations de parachèvement (Grenailage, ébarbage traitement thermique, soudage, finition).
- * Autres installations (Equipements énergétique, Manutention, casse fonte, Transport, machines-outils,...).

c) Capacité

La fonderie fonte de Tiaret, 8.000 T/ An.
 La fonderie Acier de Tiaret, 4.000 T/ An.
 La fonderie Nouvelle de Tiaret, 350T/ An fonte.

I.1.5. Direction commerciale

La direction commerciale assure la gestion des produits en vantant les pièces finis, elle accueille les clients au service de commande pour le lancement du produit, puis, ils vont se diriger vers le service facturation afin de passer par le recouplement et créance pour l'obtention des pièces au magazine de l'expédition.

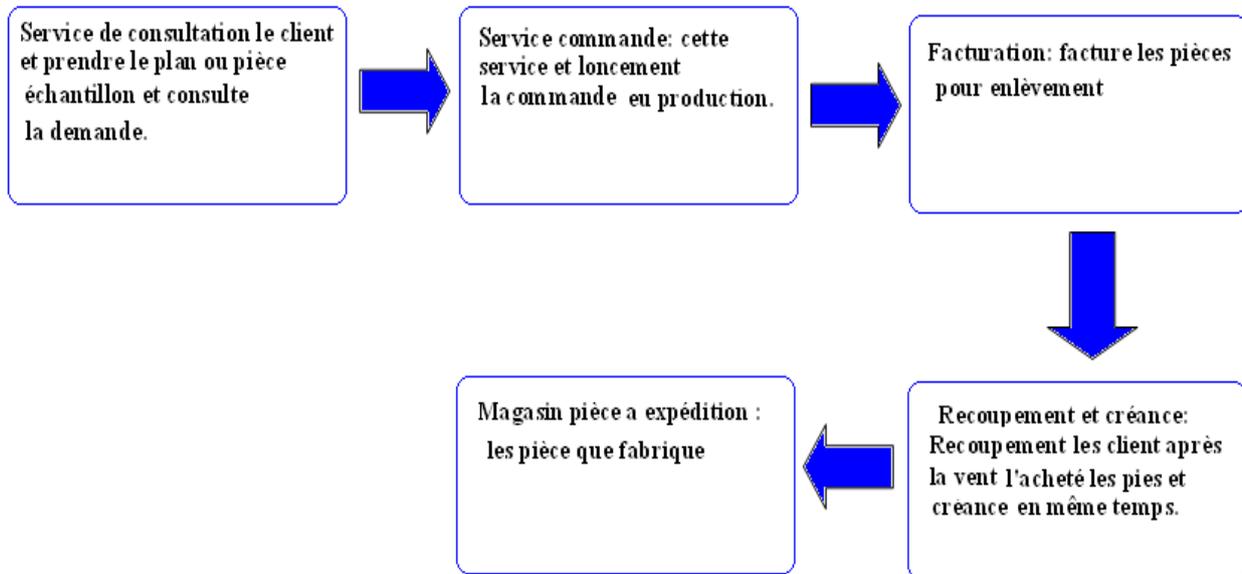


Figure I.2 : organisation du service de Direction commerciale vente.

I.1.6. Direction technique

La direction technique réalise le modèle technique de la pièce commandée, elle contrôle les dimensions et lance la production des pièces.

I.1.7. Laboratoire

ALFET possède un laboratoire équipé en moyen de mesure étalonnée permet, des essais pour produit finis (essais destructif et non destructif s, qualité physico-chimique et métallique) en cours de production et avant expédition. Il comprend :

- Laboratoire pour analyse de sable
- Laboratoire chimique et spectral
- Laboratoire métallographique
- Laboratoire pour essais non destructifs
- Laboratoire pour les essais mécanique

I.1.8. Atelier de modelage

Cet atelier est lié directement au département des méthodes (direction technique), assure la fabrication de modèle et procède au contrôle des outillages (modèles) sous traités ou fournis par le client.

I.1.9. Atelier de production

Elle reçoit le lancement de fabrication par la direction technique et commence directement la production.

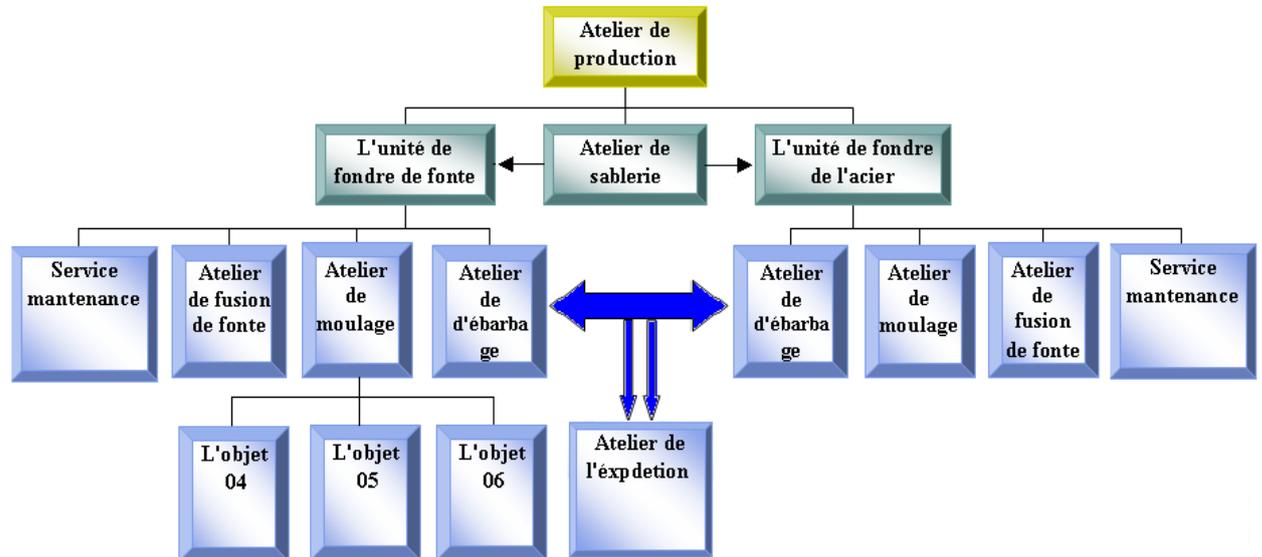


Figure I.3 : Organisation des ateliers de production

a) Atelier de sablerie

Dans les grandes fonderies le sable est la matière la plus importante au moulage des pièces géantes et même parfois des petites pièces, ce pendant; après avoir le contrôler à la réception dans un laboratoire physico-chimique pour l'agréer, le sable est donc séché et refroidit en le classant par silo de stockage, pour l'envoyer aux unités de fonderie.

b) l'unité de la fonderie de fonte

1. Atelier de fusion de la fonte

Contient 4 fours à induction électromagnétique de capacité de 10 tonnes chacun pour la fusion de fonte, on a besoin de fer brut, le pourcentage de carbone C indique le genre de fonte, et si (silicium), Mn (manganèse), p (phosphore), s (soufre) pour des Performances différentes.

2. Atelier de sablerie

Pour faire fondre la fonte on utilise le sable à vert, on le mélange avec l'argile (bentonite) et l'eau, cet atelier est doté de deux mélangeurs de capacité 1250 kg.

3. Atelier de moulage

3.1. Atelier de moulage mécanique

On moule dans cet atelier une grande quantité de pièces de moyennes dimensions, elle comprend deux chantiers de moulage avec deux lignes de moulage semi-automatique $500 \times 620 \times 200 \times 150$ et quatre machines à mouler par secousse et pression et deux machines à démouler le châssis pour le déplacer vers l'atelier d'ébarbage.

3.2. Atelier de moulage à main

On moule dans cet atelier la grande pièce et la petite pièce et se compose par un chantier de moulage manuel avec malaxeur à vis rapide (25M3/H) et un pont roulant de 20/5 tonnes pour déplacer les pièces à l'atelier d'ébarbage.

4. Atelier d'ébarbage de fonte

On élimine les bavures des pièces moulées à partir des instruments manuels pour les grandes pièces et machine d'ébarbage pour les petites pièces. Il contient : des sableuses à projection de grenaille, des meuleuses et des tambours de dessablage.

4.1. Atelier d'ébarbage de petites et moyennes pièces

Ici les pièces sont réparties en deux. L'une valable à l'expédition et l'autre non valable pour le recyclage.

4.2. Atelier d'ébarbage de grandes pièces

Les grandes pièces sont ébarbées manuellement avant l'expédition.

5. Atelier de traitement thermique

L'atelier de traitement thermique destiné pour le traitement des boulets de broyage chauffé jusqu' à 900 °C et reste dans le four thermique 12 heures jusqu'à 16 heures pour être solidement utilisée.

6. Atelier d'expédition

Au service commercial on relève les statiques de production (le poids, le nombre des pièces).

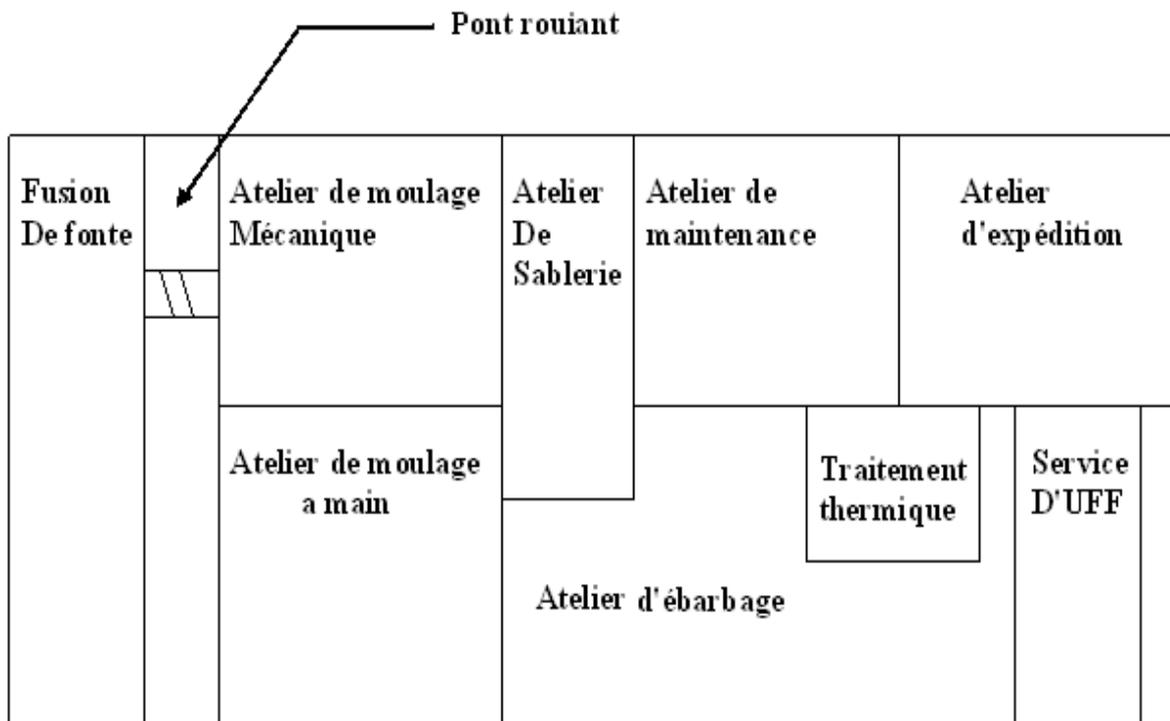


Figure I.4 : Architecture de l'unité de fonderie de fonte

c) l'unité de la fonderie l'acier**1. Atelier de fusion l'acier**

Deux fours à arc électrique de 6 tonnes et deux fours MF de 1000kg, l'acier se réchauffe jusqu'à 1600 °C pour le faire liquéfier, on distingue deux types de l'acier :

- Acier ordinaire faiblement allier < 5% de (Cr, Mn, Ni)
- Acier fortement allier > 5% de (Cr, Mn, Ni)

On les déplace vers les ateliers de moulage par le pont roulant de capacité 25 tonnes.

2. Atelier de sablerie

Le sable est mélangé en deux mélangeurs ont une capacité de chacune, avec le silicate de soude pour nous donne le sable à vert.

3. Ateliers de moulage**3.1. Atelier du moulage manuel**

Le moulage manuel est destiné aux grandes pièces ainsi qu'aux petites pièces de quantités réduites.

3.2. Atelier du moulage mécanique

Cet atelier est destiné au moulage des petites pièces en grandes quantités .Il dispose de huit machines à mouler et quatre chaînes.

4. Atelier de manutention

Ponts roulants de 20t+chariots élévateurs.

5. Traitement thermique:

Il traite la résistivité thermique avec :

- 4 fours de haute température
- 2 fours de basse température
- 2 bacs de tempe (huile-eau)

6. Atelier d'ébarbage

Dans cet atelier, l'ébarbage des grandes pièces est manuel, pour les petites pièces on utilise la machine d'ébarbage.

7. Atelier d'expédition de l'acier

Après l'ébarbage ont réunis les pièces dans l'atelier d'expédition pour les envoyer vers le client.

8. Atelier de maintenance :

Un atelier de maintenance est équipé d'installation de dessablage et de moulage, elle dispose le matériel de l'entretien de toutes pannes mécaniques ou électriques.

9. Atelier d'usinage

Assure l'usinage des pièces moulées afin d'obtenir les dimensions désirées.



Figure .I.5 : Architecture de l'unité de fonderie de l'acier

d). Atelier Energies

- ✓ Installations de distribution d'électricité
- ✓ Installations de distribution de Gaz naturel
- ✓ Installations de production d'Air comprimé
- ✓ Installations de production d'Oxygène
- ✓ Installations de traitement et de refroidissement d'eau.

e) clinique (dispensaire)

Il y a un dispensaire disponible pour le secourisme en cas de nécessité (accident de travail.....)>> [1]

I.2. Qu'est-ce que le moulage

<< Le moulage est l'action de prendre une empreinte qui servira ensuite de moule dans lequel sera versé un matériau et qui permettra le tirage ou la production en plusieurs exemplaires d'un modèle.

Le moulage consiste donc à injecter un matériau liquéfié par la chaleur dans un moule dont il prendra la forme.



Figure .I.6: Modèle de moulage

I.2.1. Où utilise-t-on le moulage

Un très grand nombre de métiers font appel au moulage, que cela soit dans le domaine industriel, artisanal médical, artistique, scientifique, etc.

En voici une brève énumération:

- Les métiers de la céramique
- Dans l'artisanat la bijouterie, les métiers d'art, fondeurs, mouleurs, statuaires, les restaurateurs d'objets, les antiquaires
- Les inventeurs, les designers
- Les pâtisseries, les traiteurs, les chocolatiers
- Le domaine de l'art regroupe les sculpteurs, les décorateurs
- La prothèse dentaire et orthopédique : pour les empreintes sur le patient et la réalisation des appareillages
- Les sciences. >> [2]

I.2.2. L'art de moulage

Un moulage réussi demande beaucoup de réflexion et de goût .avant toute opération il faut étudier minutieusement les formes, en sentir les moindres détails. D'expérience, cette observation attentive du modèle allié à un peu de pratique permet de déterminer le choix de la technique, et des produits à utiliser.

C'est principalement le travail du modelleur, qui d'après la pièce à réaliser les empreintes dans les moules, les noyaux; le tout en tenant compte des impératifs de démoulage, de retrait de la matière après coulée .Le modelleur est celui qui dessine les outillages au bureau d'études, mais aussi celui qui réalise les modèles à l'atelier d'outillage.

I.2.3. Différents types de moulage

I.2.3.1 Moulage en sable

<< Technique très courante en fonderie, démoulage par destruction du moule .L'intérêt de cette technique est qu'elle permet de travailler avec un matériau réfractaire(le sable), et que le moule ne coûte pas cher à fabriquer.

I.2.3.2 Moulage en sable humide

Le sable n'est pas mélangé à une résine quelconque, mais tient par son état d'humidité. Le moulage s'effectue soit, en horizontal, dans des châssis (cas le plus ancien et le plus fréquent) où soit sous forme de mottes avec plan de joint vertical; c'est le moulage en motte.

I.2.3.3 Moulage en sable à durcissement thermique

Le durcissement se fait soit dans l'empreinte ou hors empreinte.

I.2.3.4 Moulage en sable à durcissement chimique

Le durcissement se fait par réaction chimique, soit entre les composants (résines), soit par l'action d'un gaz.

I.2.3.5 Moulage en sable à durcissement physique

Le durcissement se fait par action directe et physique sur le sable comme le moulage sous vide ou procédé V- procès.

I.2.3.6 Moulage à modèle perdu

Les techniques les plus diffuses sont pour la plus ancienne celle du moulage à cire perdue et pour la plus moderne, le moulage à modèle polystyrène connu sous le nom de procédé PMP.

I.2.3.7 Moulage métallique

Le moule métallique est réutilisable plusieurs fois .Le démoulage est rendu possible par la différence de matière entre le moule et métal coulé. Le moule subit toujours un potage avant l'introduction du métal en fusion.

I.2.3.8 Moulage par gravité

De pièce en aluminium (piston d'automobile).

I.2.3.9 Moulage à base pression

Procédé semi-automatique pour la fabrication de pièces moulées de haute qualité, précises et homogènes, en petite et grande série. Le métal, aluminium en général,est introduit dans le moule par le bas sous une faible pression (0.2 à 2 bras).

I.2.3.10 Moulage sous pression

De pièce en aluminium (industrie automobile, aéronautique) et en zamak, ou en matière plastique (moulage par injection).

La machine à mouler sert à fabriquer des demi-moules avec châssis, de préférence d'après la procédé à sable vert, les moules sont fabriqués à l'aide des opérations suivantes:

- Secousse pression
- Pression et secousse et vibration
- Soulèvement par pression et secousses.

On atteint de haute degrés de dureté des moules a un court temps de serrage la programmation pour de différentes suites d'opérations rend un travail a de différentes conditions technologiques (arrêt pour remplissage de sable, tirage d'air etc.) possible. >> [3]

I.2.4. Le moulage en sable

<< Ils offrent une grande liberté de formes et de dimensions et ils permettent d'obtenir des pièces de bonne qualité.

Le sable devra être :

- ✓ Réfractaire
- ✓ Plastique
- ✓ Poreux >> [2]

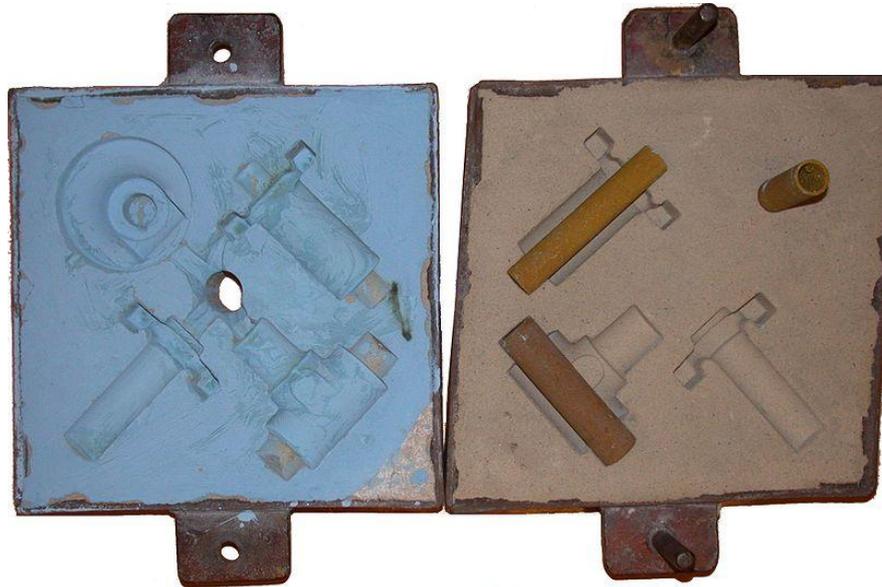


Figure I.7: Moulage en sable

➤ L'ensemble des opérations de moulage en sable est donné par le schéma ci-dessous :

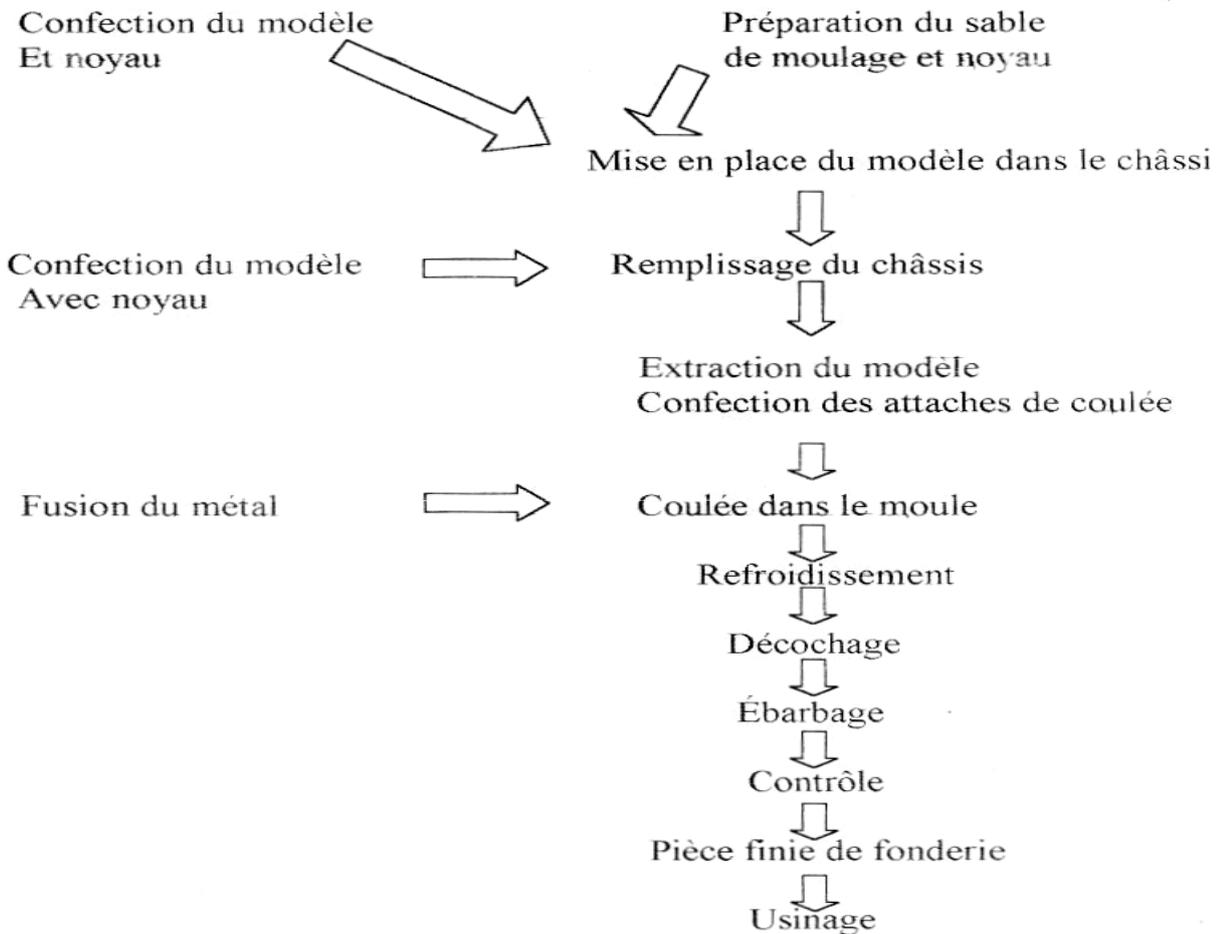


Figure I.8: Schéma des opérations de moulage en sable

I.2.4.a. Caractéristiques de sable en fonderie

Les sables utilisés pour réaliser les moules doivent réunir un certain nombre de propriétés

- Reproduire les formes et les détails les plus fins du modèle.
- Résister à la température du métal fondu.
- Se désagréger facilement après le refroidissement de la pièce coulée pour permettre le décochage.

Pour répondre à ces impératifs, le sable de moulage doit satisfaire à deux exigences fondamentales qui sont:

- La mise en forme au contact d'un modèle en épousant tous les détails; cela signifie que son état initial doit être pulvérulent, soit liquide ou liquide plastique, soit plastique; suivant cet état, le compactage est effectué soit par simple gravité, soit par un effort mécanique de serrage (pression, secousse, vibration, projection mécanique ou pneumatique).

- La conservation de cette forme jusqu'à la solidification complète du métal

Les propriétés recherchées pour le sable de moulage sont alors:

I.2.4.a-1-Réfractaire

Le sable doit présenter une température de ramollissement supérieure à la température de fusion de l'alliage considéré

I.2.4.a-2-Perméable

Le sable doit permettre l'évacuation des gaz contenus dans l'empreinte du moule lors du remplissage ou générés à la coulée par la réaction moule-alliage (indépendamment des événements et des tirages d'air réalisés dans le moule). La perméabilité dépend des dimensions des grains de sable, de la qualité du liant, du serrage...

I.2.4.a-3-Plastique

Sous l'action de pressions exercées par l'outillage sur le sable, son état subsiste après suppression de l'action de l'outillage (conservation de l'empreinte après démoulage). La plasticité dépend de la finesse des grains de sable, du liant utilisé, de l'enrobage des grains.

I.2.4.a-4-Récupérable

Une économie de matériaux est assurée par recyclage des sables, après régénération et contrôle des caractéristiques.

I.2.4.b. Principe

<< Le matériau est coulé dans un moule en sable. Le moule sera détruit après la fabrication de chaque pièce, il faut donc refaire un moule pour chaque pièce. Le moule est fabriqué en au moins deux parties.

Chaque partie ayant une empreinte donnant la moitié des formes de la pièce. Pour chaque partie du moule on utilise donc un modèle (en métal, bois ou autre matériau) qui enfoncé dans le sable va laisser son empreinte dans la partie du moule. Le modèle est ensuite extrait du moule avant que les deux parties du moule soient assemblées.

Lorsque la pièce à des formes intérieures qui empêche toute extraction du modèle, on insère entre les deux parties du moule un noyau en sable qui devra être maintenu dans le moule, puis détruit après démoulage de la pièce.

Ce noyau est donc fabriqué de la même manière que chacune des parties du moule en tassant du sable dans un modèle en deux parties ayant les formes de la pièce.>>[4]

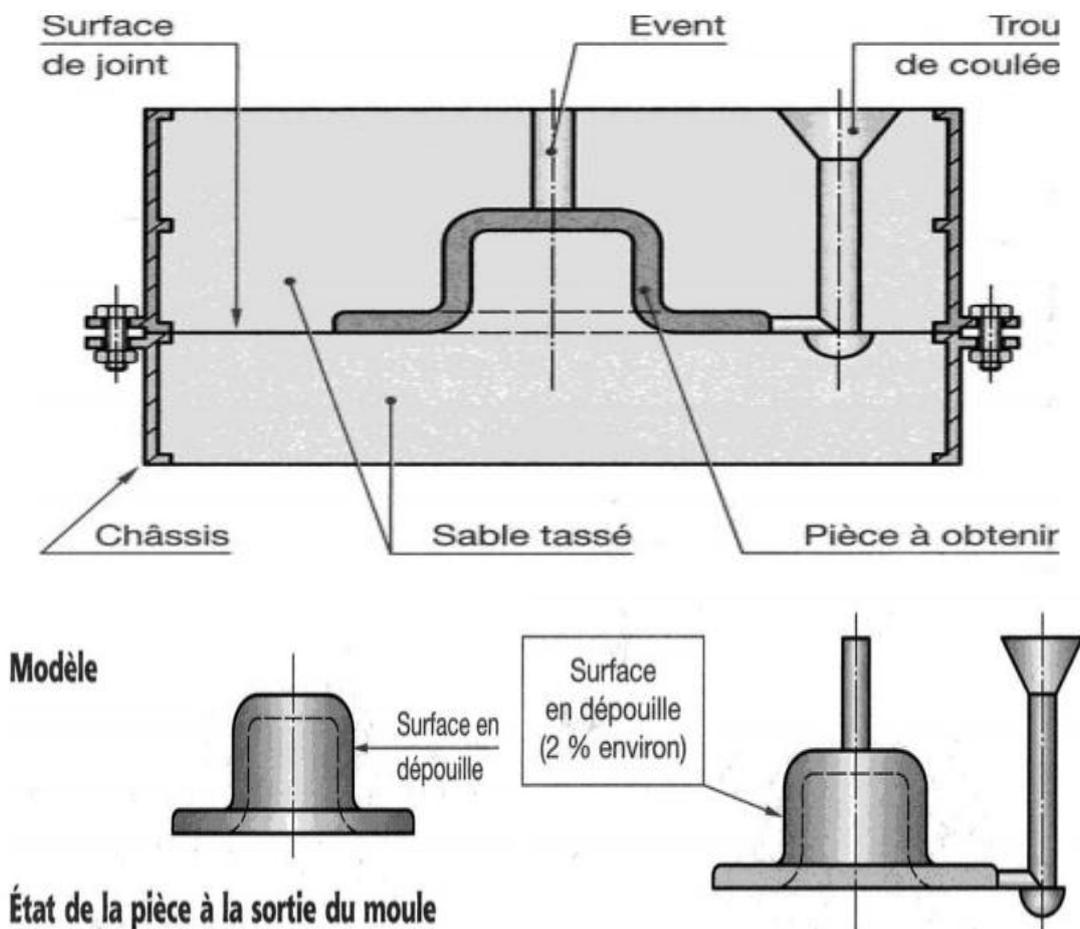


Figure I.9: Le moule en sable.

I.2.5. Le moule

Ensemble des éléments en matériaux appropriés délimitant l'empreinte et recevant le métal liquide qui après solidification, donnera la pièce.

Un moule comprend au minimum deux parties qui peuvent être séparées l'une de l'autre (châssis, chapes, coquilles). La surface commune aux deux demi moules porte le nom de plan de joint même si cette surface n'est pas effectivement un plan. On distingue deux types de moules :

- * Moule non permanent
- * Moule permanent

I.2.5.1. Les moules permanents

I.2.5.1.a. Moulage en fonderie

Le moulage ou fonderie est un ensemble de procédés qui permet de réaliser des pièces métalliques brutes. Le moulage proprement dit, consiste à réaliser des pièces brutes par coulée du métal en fusion dans un moule en sable ou en métal (représentant l'empreinte de la pièce à obtenir). Le métal en se solidifiant, reproduit les contours et dimensions de l'empreinte du moule (fig. 10).

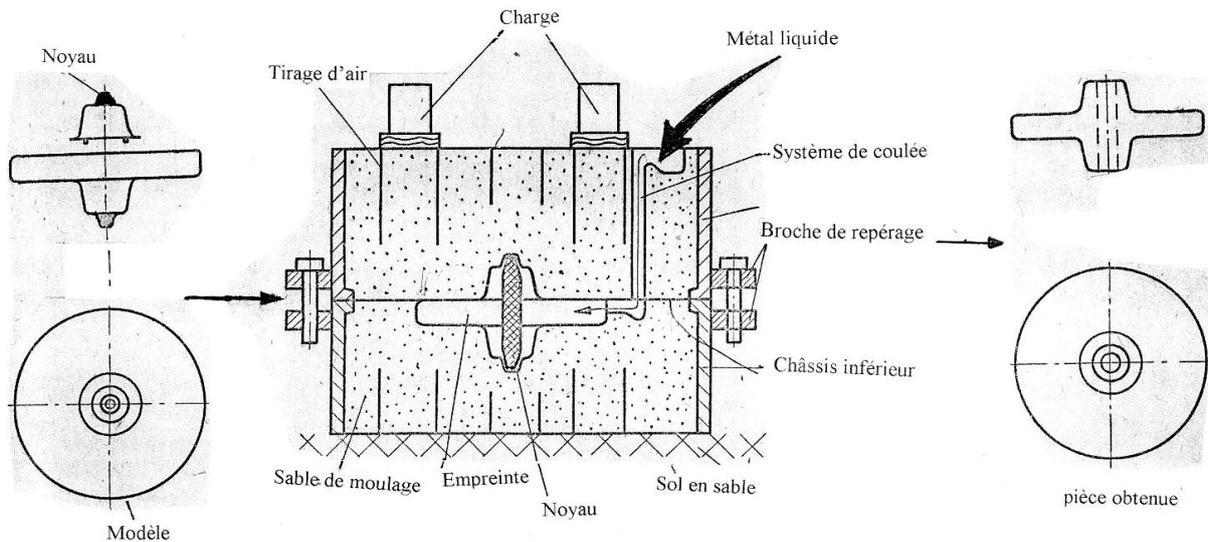


Figure I.10 : Moulage en fonderie

I.2.5.1. b. Le moulage par injection

<< C'est une technique de fabrication de pièces en grande ou très grande série essentiellement pour les matières plastiques les pièces vont de quelques millimètres à plusieurs mètres (fig. 11). >> [2]

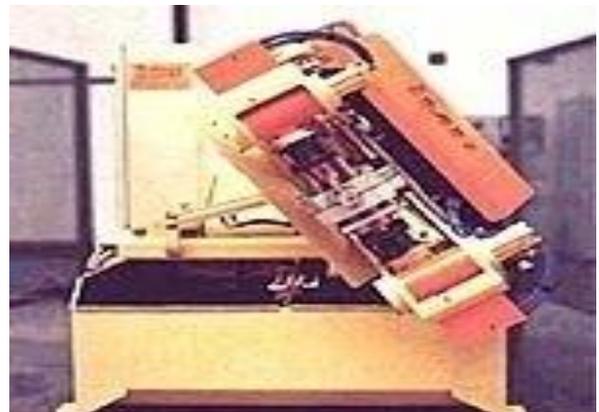
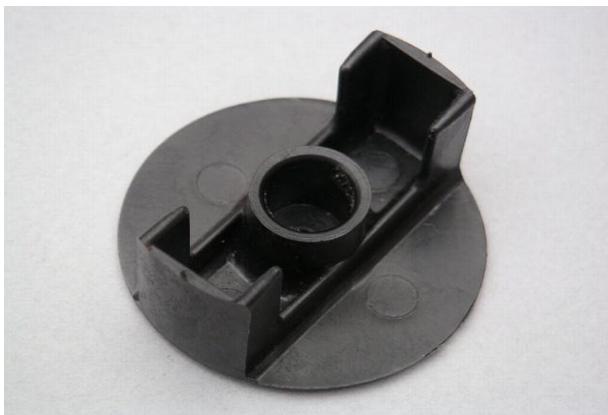


Figure .I.11: Pièce moulé par injection

I.2.5.2. Les moules non-permanents

<< Un moule non-permanent est un moule qui ne sert qu'une seule fois, pour réaliser une pièce. Pour chaque pièce brute de coulée produite, le moule est détruit.

-La fonderie à partir de moules non-permanents se divise en deux catégories :

-Le moulage traditionnel pour lequel chaque moule est fabriqué à partir d'un modèle permanent (moulage en sable).

-le moulage par modèle perdu pour lequel on réalise autant de modèles que de moules (moulage en cire perdue). >> [2]



Figure .I.12: Moulage en Cire perdue

I.2.6. Moulage à la machine (mécanique)

Le moulage à la main ne se pratique que pour un nombre de pièces réduits ou pour des pièces volumineuses, en plus le serrage du sable à la main est long, pénible et souvent peu régulier. De plus le démoulage est délicats car s'il n'est pas fait verticalement, les arrachements du sable imposent des retouches et peuvent rendre le moule inutilisable. Tous ces inconvénients peuvent être supprimés par le moulage mécanique. Donc dans le moulage mécanique, le sable est serré mécaniquement soit :

- Par pression.
- Par secousses.
- Le plus souvent par combinaison des deux procédés (pression et secousses).
- Par projection.

L'empreinte de la pièce n'est plus donnée par un modèle ordinaire, mais par un modèle fixe sur un support dont l'ensemble est appelé plaque modèle, d'où les pièces obtenues par moulage mécanique sont limitées en dimension par la grandeur des châssis.

I.2.6.1. Les inconvénients du moulage

<< Moules parfois très onéreux.

- ✓ La conception d'un moule pour des pièces uniques et détaillées demande beaucoup de temps.
- ✓ La santé des pièces moulées est parfois assez faible >> [2]

I.2.6.2 Avantages du procédé

- << - Possibilité de mouler des matériaux dont le point de fusion est élevé (acier, fonte).
- Possibilité de mouler des pièces de (très) grande taille (Moulage de carters de machines)
- Procédé rentable pour les petites et moyennes séries (quelques dizaines à quelques milliers de pièces).

I.2.6.3 Inconvénients du procédé

- Surfaces obtenues peu précises ($\pm 0,5$ mm) Nécessite un usinage des surfaces fonctionnelles
- Procédé peu rentable pour les grandes séries. >> [4]

I.3. Conclusion

Dans ce chapitre, nous avons présenté un aperçu de l'entreprise sur ALFET et son fonctionnement à travers plusieurs ateliers. Chaque atelier a sa propre fonction, Ensuite nous avons présenté un aperçu des différents types de moulages, ainsi que les caractéristiques du sable utilisé en fonderie, On distingue deux types de moules :

- Moule non permanent
- Moule permanent

En fin on décrit ces différents composants la sablerie et la fonderie et puis en a donné le rôle les deux composants qui fabrique des pièces industriels.

Chapitre II

Etude du Microcontrôleur PIC 16F877A

II.1. Introduction

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires « mémoire morte pour le programme, mémoire vive pour les données », unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique « quelques milliwatts en fonctionnement, quelques nano watts en veille », une vitesse de fonctionnement plus faible « quelques mégahertz » et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans le domaine d'industrie pour la commande et contrôle des systèmes.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Des nombreux fabricants se sont implantés sur le marché des microcontrôleurs, la société américaine Micro-chip a mis au point un microcontrôleur CMOS « Complementary Metal Oxide Semi-conductor » appelé PIC ; cette dénomination est sous copyright de Micro-chip, donc les autres fabricants ont été dans l'impossibilité d'utiliser ce terme pour leur propre microcontrôleur. Actuellement ce microcontrôleur est très utilisé.

II.2. Définition d'un PIC

<< Un PIC (Programmable Interface Controller) est un microcontrôleur de Micro-chip Technologie.

Un PIC est un composant dit RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Ces microcontrôleurs sont conçus sur une architecture dite HAVARD, elle est basée sur deux bus, un pour les données (bus de données) et l'autre pour les instructions (bus de programme). >> [5]

II.2.1. Classification des Pics de Micro chip

<< Actuellement les modèles micro chip, sont classés en trois grandes familles, comportant chacune plusieurs références. Ces familles sont:

- Baseline: c'est la famille où l'unité centrale travailleur 12-bits, exemple : 12CXXX...
- Mid-range : c'est la famille où l'unité centrale travaille sur 14-bits, exemple : 16F877A.
- High-end : c'est la famille où l'unité centrale travaille sur 16-bits, exemple : 18FXXX, 18CXXX ...>> [6]

<< Les PIC s sont des composants STATIQUES, Ils peuvent fonctionner avec des fréquences d'horloge allant du continu jusqu'à une fréquence max spécifique à chaque circuit.

Un PIC16F876-04 peut fonctionner avec une horloge allant du continu jusqu'à 4 MHz. Nous nous limiterons dans ce chapitre à la famille Mid-Range et particulièrement au PIC 16F876/877. Le tableau 01 indique les différents circuits de la famille 16F87X, >> [7]

PIC	FLASH	RAM	EEPROM	I/O	A/D	Port //	Port Série
16F870	2K	128	64	22	5	NON	USART
16F871	2K	128	64	33	8	PSP	USART
16F872	2K	128	64	22	5	NON	MSSP
16F873	4K	192	128	22	5	NON	USART/MSSP
16F874	4K	192	128	33	8	PSP	USART/MSSP
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Tableau II.1: Différents circuit de la famille 16F87X.

Dans notre application, nous avons choisis d'utiliser le PIC 16F877A ; qui contient un espace mémoire plus large que les autres Pics ; et disponible dans le marché et très utilisé.

II.2.2. Identification d'un PIC

<< Un PIC est généralement identifié par une référence de la forme **NN LLL XXX - ZZ** où :

NN : désigne la famille à laquelle appartient le circuit « 12, 14, 16,17 et 18 ».

LLL: est un ensemble d'une, deux ou trois lettres qui indique le type de la mémoire de programme contenue dans le circuit et si la plage de tension d'alimentation est normale ou étendue.

- ✓ **C**: EPROM ou EEPROM.
- ✓ **CR**: PROM.
- ✓ **F**: FLASH.

XXX : est un ensemble de deux ou trois chiffres constituant la référence du circuit.

ZZ : vitesse maximale du quartz de pilotage. >> [8]

Exemple de PIC 16F877-20 :

16 : indique la famille Mid-Range.

F : indique le type de mémoire programme utilisée : F pour Flash.

877 : identité.

20 : indique la fréquence d'horloge.

II.2.3. Le choix du microcontrôleur PIC16F877A : Pourquoi le PIC16F877A ?

Le choix de microcontrôleur PIC16F877A est primordial, il est employé dans ce projet en raison de sa grande partie de ses performances, sa taille, sa facilité d'utilisation et le prix du montage.

Le PIC16F877A possède 5 ports d'entrées sorties, une capacité mémoire programme flash de 8K qui est nécessaire pour mettre l'application arithmétique à un point mobile, une mémoire RAM répartie sur 368 octet, ce qui permet l'utilisation de plus de variables dans un programme, un convertisseur analogique-numérique 10 bits, une horloge jusqu'à 20MHZ, une tension de fonctionnement de 2V jusqu'à 5.1V, un courant d'entrée-sortie des pins allant jusqu'à 25mA, un module de communication USART (Universal Synchronous Asynchronous Receiver Transmitter) qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone, soit asynchrone,etc.

Le PIC 16F877A possède en plus des instructions très puissantes, donc un programme à développer réduit, surtout lorsqu'on utilise le logiciel de programmation comme le « CCS PICC » qui est un langage de programmation à haut niveau, simple, facile et optimisé par rapport à l'assembleur, il possède un nombre important de procédures et fonctions prédéfinies dédié au PIC 16F877A. Le CCS PICC a pour rôle principal de contrôler la cohérence du code au moment de la compilation, c'est-à-dire qu'il offre la possibilité de vérification d'erreurs pour les corriger au moment de la programmation.

En fait la cause principale du choix de microcontrôleur PIC est qu'il dispose de l'option du convertisseur A/D pour satisfaire le côté d'acquisition, aussi la possibilité de l'adaptation au protocole de la liaison RS232.

II.2.4. Les caractéristiques du PIC 16F877A

<< Le PIC 16F877A est un microcontrôleur de Micro-chip, fait partie intégrante de la famille Mid-rang (16) dont la mémoire programme et de type flash (F), ce µc est un circuit intégré à 40 broches dont les caractéristiques générales peuvent être résumé comme suit :



Figure II.1: Microcontrôleur PIC 16F877A

- Une fréquence de fonctionnement allant jusqu'à 20MHz.
- Jeu d'instructions de 35 instructions
- Une mémoire de programme de type FLASH sur 8K mots (1 mot=14 bits).
- Une mémoire de données de type RAM sur 368 octets.
- Une mémoire de données de type EEPROM sur 256 octets.
- Une pile (Stack) à 8 niveaux.
- 14 sources d'interruptions.
- Un chien de garde WDT (Watch Dog Timer).
- Timer0 (compteur 8-bits avec pré-diviseur).
- Timer1 (compteur 16-bits avec pré-diviseur et possibilité d'utiliser une horloge externe).
- Timer2 (compteur 8-bits avec pré-diviseur et post-diviseur).
- Un convertisseur Analogique-Numérique 10-bits à 8 entrées multiplexées.
- Deux modules de capture/comparaison/PWM.
- Ports Entrées/Sorties : A, B, C, D et E.
- Une interface de communication série (MSSP, USART).
- Une interface de communication parallèle (PSP).
- Tension de fonctionnement est entre 2.0V et 5.5V. >> [5]
-

II.2.4.1. DESCRIPTION GENERALE DU PIC 16F877A

<< Le pic 16F877A est un circuit intègre contenu dans un boîtier nommer « DIL 40 », il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1^{ère} broche est placée dans le coin situé à gauche de l'encoche de repérage. >> [6]

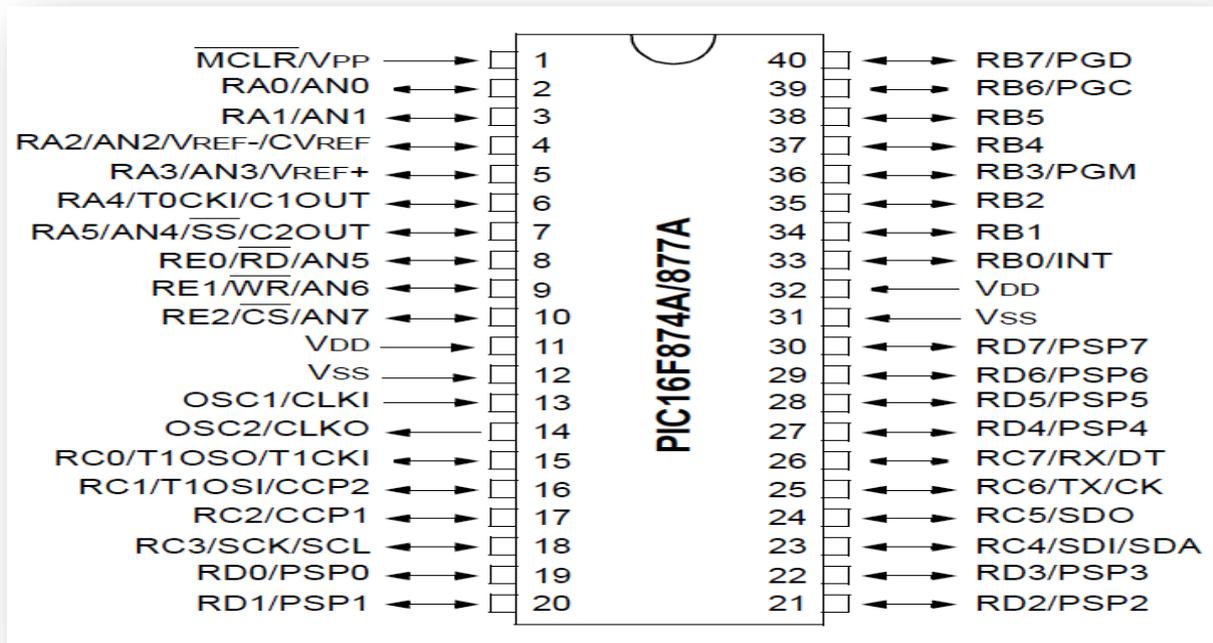


Figure II.2: Boitier du PIC

II.2.4.2. Architecture externe du PIC 16F877A

Le boitier ci-contre décrit l'architecture externe du 16F877A qui comprend 40 pins dont :

- 33 pins d'entrées/sorties multiplexées avec d'autres fonctions.
- 4 pins pour l'alimentation : VDD et VSS.
- 2 pins pour l'oscillateur : OSC0 et OSC1
- 1 pin pour le RESET : MCLR

PIN	UTILISATION
RA0 à RA5	Lignes d'entrées /sorties du port A
RB0 à RB7	Lignes d'entrées /sorties du port B
RC0 à RC7	Lignes d'entrées /sorties du port C
RD0 à RD7	Lignes d'entrées /sorties du port D
RE0 à RE3	Lignes d'entrées /sorties du port E
VDD	Broche d'alimentation +5V
VSS	Broche du 0V
MCLR	Reset du circuit
OSC1 et OSC2	Broche recevant quartz externe.

Tableau II.2: Broches du PIC 16F877A

II.2.4.3. Architecture interne du PIC 16F877A

<< Comme pour tous les circuits intègre, chacun de ses broches à une ou plusieurs fonctions qui sont résumées par un sigle mnémotechnique. Ce microcontrôleur présente une architecture Harward, les données sont placées dans une mémoire de type RAM de 368 bytes. La mémoire de programme est constituée de mot de 14 bytes, est type FLASH (non volatile). Ces ressources sont donc précieuses, en comparaison de celles d'autres composantes. Le 16F877A possède encore 5 ports (A à E) et 3 temporisateurs (timers), ce diagramme bloc présente les composantes du 16F877A :>> [5]

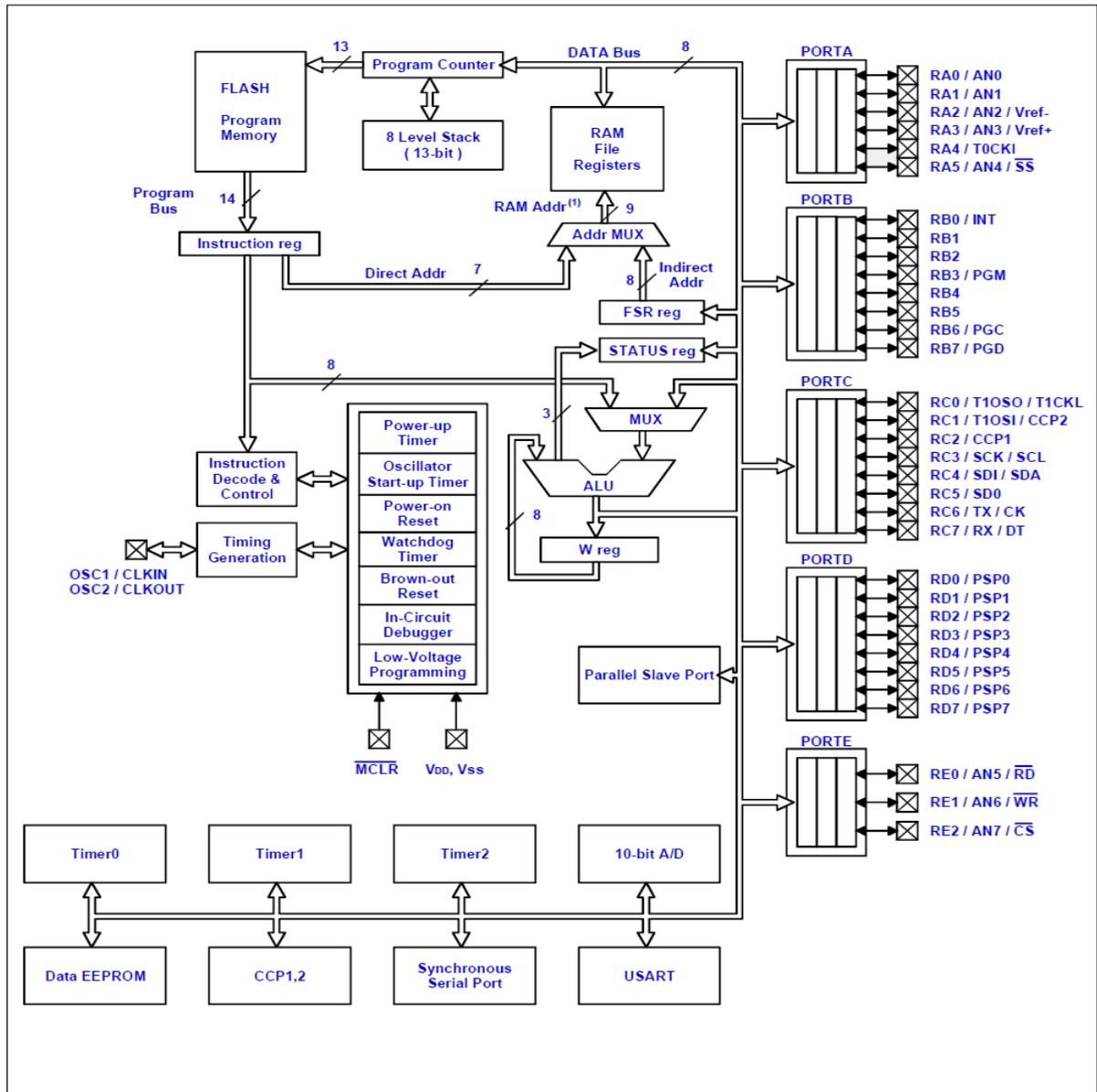


Figure II.3: L'architecture interne du PIC 16F877A

Le schéma ci-dessus représente les principaux éléments du PIC 16F877A. Ça a l'air un peu complexe au premier abord, une visite guidée nous permettra d'y voir plus clair :

II.2.5. Les mémoires

<< Le PIC16F877A possède trois types de mémoires :

II.2.5.1. La mémoire programme

La Mémoire Programme, de type FLASH sur le 16F877A. Capacité : 8K mots de 14 bits. C'est dans celle-ci est stocké le programme du PIC. Après compilation de votre code, le compilateur génère un fichier « *.hex », une suite de codes hexadécimaux. Celui-ci est transféré ensuite dans la mémoire programme du PIC à l'aide du programmeur.

Cette mémoire n'est pas reliée au bus de données (DATA Bus), sa vocation est de stocker le programme du PIC, mais pas les variables de votre programme. Le gros avantage de la mémoire FLASH c'est que vous pouvez la réécrire, donc implanter un nouveau programme dans le PIC.

II.2.5.2. La mémoire EEPROM

La mémoire EEPROM « Electrical Erasable Programmable Read Only Memory », L'EEPROM est plutôt une mémoire de stockage de données à long terme, alors que la RAM est utilisée pour les variables du programme. Sur le PIC 16F877A, on a 256 octets d'EEPROM disponible. Les mémoires de type EEPROM sont limitées en nombre de cycles d'effacement / écriture. Ce nombre de cycle est tout de même de l'ordre du million pour le PIC.

II.2.5.3. La mémoire RAM

La Mémoire RAM, qui fait partie de la zone d'adressage des données. Elle comprend tous les registres spéciaux permettant de contrôler le cœur du PIC (le microprocesseur) ainsi que ses périphériques. Elle contient également des cases mémoires à usage générique dans lesquelles pourront être stockées les variables de nos futurs programmes. La mémoire RAM disponible du 16F877A est de 368 octets.

II.2.5.4. Les ports d'entrées/sorties

Le PIC 16F877A dispose de 5 ports entrées/sorties (au total 33 pins) : port A, B, C, D et E (figure II.4). De plus il y a le PSP (Parallèle Slave Port) : le Port Parallèle Esclave.

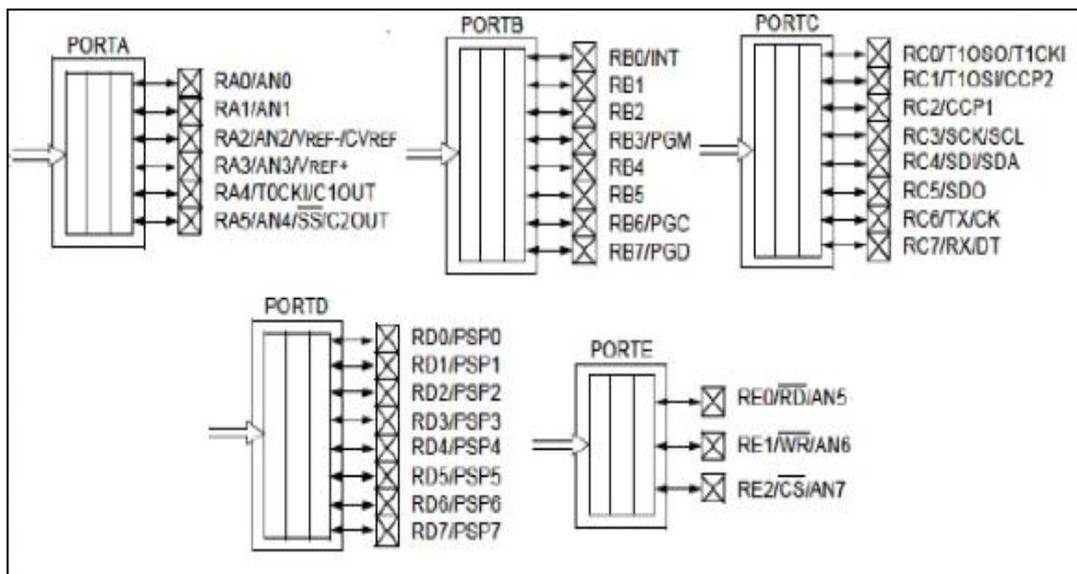


Figure II.4: Les différents ports du PIC16F877A

Un examen plus attentif du schéma ci-dessus nous montre cependant que les lignes d'entrées/sorties (les pattes du composant) correspondantes sont également utilisées par d'autres modules du PIC. Ainsi, les pattes du PORTA servent également au convertisseur Analogique/Numérique, les pattes du PORTD au Port Parallèle Esclave, etc. On voit également que les ports B, C et D ont 8 lignes d'entrée/sortie, alors que le port A n'en a que 6 et le port E que 3. Ils sont tous connectés au bus de donnée (DATA BUS), on pourra donc librement les adresser pour y

lire ou écrire des données, et donc allumer des LED, commander des moteurs, des afficheurs LCD, lire les données envoyées par un clavier ou un bouton poussoir...

On peut configurer les entrées/sorties de chaque port en entrée ou en sortie, grâce à un registre spécial dédié à chaque port (TRISX).

Bit k de TRISX = 0 \implies broche k de PORTX = sortie

Bit k de TRISX = 1 \implies broche k de PORTX = entrée

De plus, un des ports (le port B) possède des résistances de « pull-up » internes qui peuvent être validées ou non par logiciel.

II.2.6. Les registre du PIC 16F877A

Dans ce paragraphe, nous allons essayer de décrire seulement les registres utilisés dans notre projet.

II.2.6.1. Le registre W

C'est le registre de travail (workregister) possède 14 bits. Il est utilisé par les Pics pour réaliser toutes sortes de calculs. La destination du résultat peut en général être un emplacement dans la RAM ou le registre de travail (W). C'est donc un registre fondamental.

II.2.6.2. Le registre STATUS ou registre d'états

Le registre d'état appelé STATUS est un registre 8 bits, son rôle est de donner diverses informations à l'utilisateur sur l'état de fonctionnement ou sur le résultat d'une opération.

II.2.6.3. Le registre INTCON ou registre de contrôle d'interruptions

Ce registre permet de configurer les différentes sources d'interruption.

II.2.6.4. Le registre OPTION-REG

Ce registre peut être lu et écrit, il contient les bits de contrôle qui permettent de configurer le pré-diviseur du Timer0, les interruptions externes et le choix du pull-up sur le port B.

II.2.6.5. Le registre T1CON

Ce registre permet de contrôler le Timer1.

II.2.6.6. Le registre T2CON

C'est un registre 8bits qui permet de paramétrer le pré diviseur et le post diviseur ainsi que d'autoriser ou non le fonctionnement du Timer2.

II.2.6.7. Les registre CCP1CON et CCP2CON

Ces registres permettent de contrôler deux modules identiques CCP1 et CCP2.

II.2.6.8. Les registre ADCON0 et ADCON1

Sont des registres de configuration du convertisseur analogique numérique. >> [5]

II.2.7. Les interruptions

Le PIC 16F877A possède 13 sources d'interruptions, les plus importantes à connaître, car ce sont les plus utilisées dans notre projet, sont :

Interruption	Source d'interruption
TMR0I	Dépassement du TIMER0 « débordement le passage de 255 à 0 »
TMR1I	Dépassement du TIMER1 « débordement le passage de 65535 à 0 »
TMR2I	Débordement de TIMER2
INT	Une interruption externe, action sur la broche INT/RB0
PSPI	Lecture/Ecriture sur le port parallèle(PORTD)
ADI	Fin de conversion analogique numérique
CCP1I	Capture/comparaison/PWM avec module CCP1
CCP2I	Capture/comparaison/PWM avec module CCP2
RCI	Réception sur le port RS232
TXI	Emission sur le port RS232
SSPI	Caractère émis/reçu sur SSP (port série synchrone)
E EI	Ecriture dans EEPROM
BCLI	Collision sur bus SSP en mode I ² C

Tableau II.3: Les interruptions du PIC 16F877A

II.2.8. Les circuits d'horloges

<< Le PIC 16F877A peut fonctionner en 4 modes d'oscillateur :

- **LP** : «Low Power» : version à quartz également, mais prévue spécialement pour les applications à très faible consommation. C'est donc une version dont la fréquence maximale de fonctionnement est limitée à 20KHz.

- **XT** : version à oscillateur à quartz «standard» fonctionnant jusqu'à une fréquence maximale de 4MHz.
- **HS** : « High Speed » : version à oscillateur à quartz capable de fonctionner jusqu'à une fréquence maximale de 20 MHz.

RC : « circuit RC » : capable de fonctionner jusqu'à 4MHz mais avec une stabilité de fréquence plus faible que les versions à quartz.

II.2.8.1. Oscillateur à quartz

En mode LP, XT ou HS, un quartz est connecté aux pins OSC1/CLKIN et OSC2/CLKOUT équivalent à les pattes 13 et 14 sur le boîtier du PIC 16F877A, pour établir l'oscillation. La figure II.5 montre comment ce quartz est connecté au PIC :

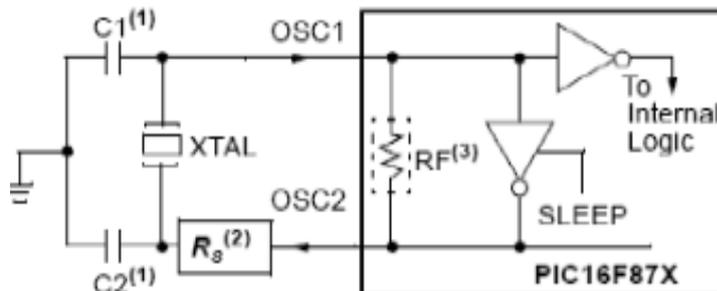


Figure II.5: Oscillateur à quartz

Toujours dans l'un de ces modes de fonctionnement, le microcontrôleur peut avoir une horloge externe connectée à la broche OSC1/CLKIN comme le montre la figure suivante :

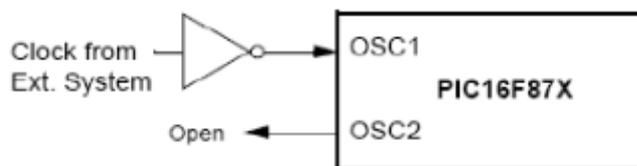


Figure II.6: Connexion d'une horloge externe

II.2.8.2. Oscillateur RC

La fréquence de l'oscillation dépend du voltage, des valeurs de R et C et de la température de fonctionnement. La figure (II.7) montre comment connecter un circuit RC au microcontrôleur :

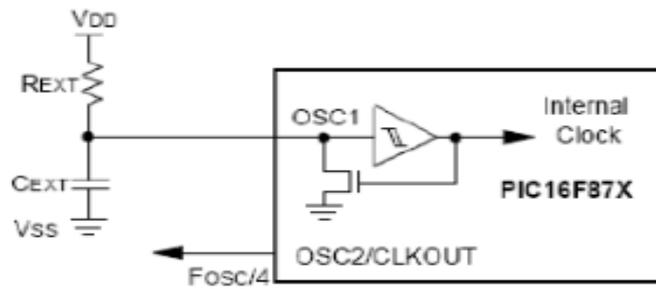


Figure II.7: Oscillateur RC

Pour générer un « cycle d’instruction », il faut 4 cycles d’horloge. Autrement dit, l’horloge système dite aussi horloge instruction est obtenue en divisant la fréquence par 4. Avec un quartz de 4 MHz, les instructions du programme s’exécutent à une cadence de 1 MHz.

II.2.9. Présentation des modules utilisés dans PIC 16F877A

Les modules utilisés PIC 16F877A dispose de 3 timers :

II.2.9.1. Le Timer0

C’est un compteur 8 bits (0 à 255), Il peut être incrémenté par des impulsions extérieures via la broche « RA4/T0CKI » du port A (mode compteur) ou par l’horloge interne «Fosc/4 » (mode timer).

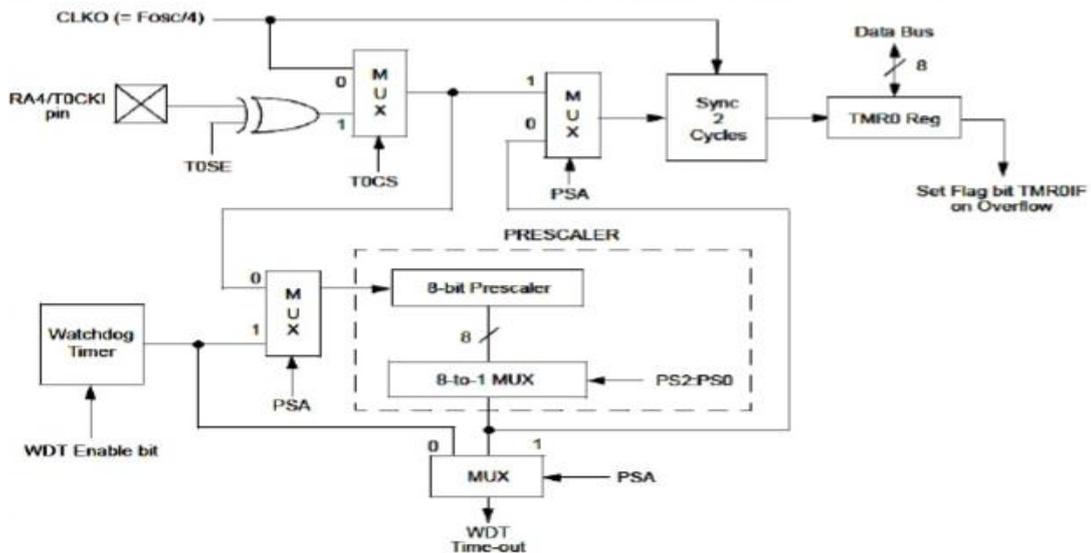


Figure II.8: Schéma du Timer0

II.2.9.2. Le Timer1

Le Timer1 fonctionne sur le même principe que le Timer0, mais avec un registre de comptage plus gros : 16 bits au lieu de 8, ce qui étend notablement ces capacités de comptage. De plus, il possède un mode de fonctionnement particulier : on peut l'utiliser en association avec un des modules CCP (modules de capture et de comparaison).

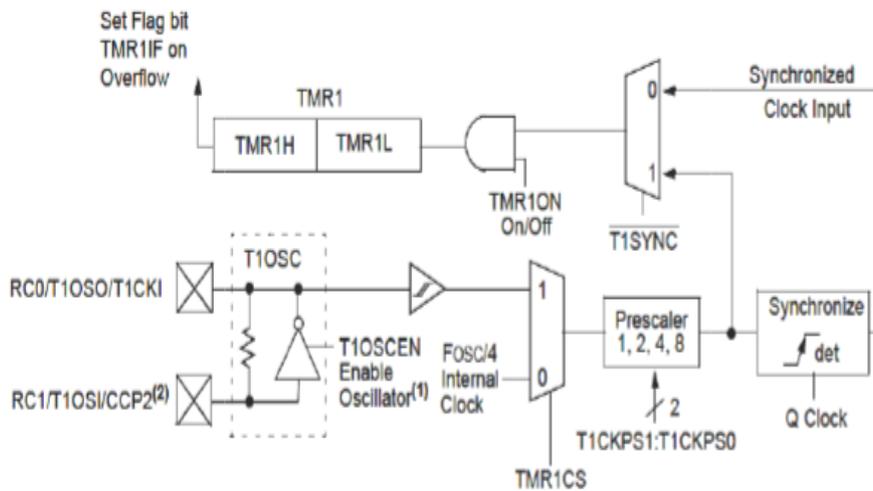


Figure II.9: Schéma descriptif du Timer1

II.2.9.3. Le Timer2

Le Timer2 a un fonctionnement différent des Timer0 et Timer1. C'est un compteur 8 bits avec pré-diviseur et post-diviseur. On s'en sert pour générer des signaux carrés, ou, en association avec le module CCP, des signaux PWM.

➤ **Fréquence du timer2 :**

Le Timer2 est un compteur 8 bits avec pré diviseur et post diviseur. C'est un registre en lecture et écriture. L'entrée du compteur est l'horloge cycle interne : $F_{osc}/4$ qui passe à travers un pré diviseur programmable par 1, 4, 16.

La sortie du compteur passe dans un post diviseur programmable sur 4 bits entre 1, 16.

La fréquence est calculée comme suit :

$$T_{max} = T_{quartz} \times \text{Valeur .du .pre .diviseur .fixe} \times \text{Valeur .du .pre .compteur .reglage} \times (\text{Valeur .definie} + 1)$$

$$f = \frac{1}{T_{max}}$$

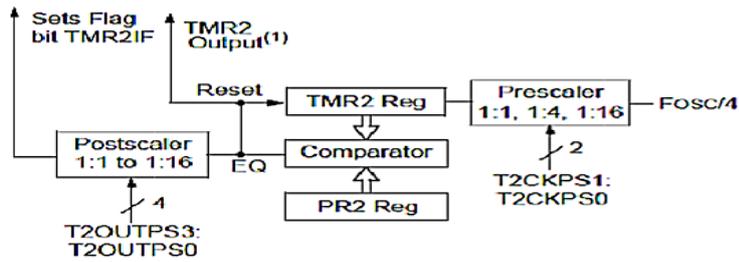


Figure II.10 : Schéma descriptif du Timer2

II.2.10. Le convertisseur analogique numérique

Notre 16F877A travaille avec un convertisseur analogique/numérique qui permet un échantillonnage sur 10 bits. Il est composé de :

- Un multiplexeur analogique 5 voies.
- Un échantillonneur bloqueur.
- Un convertisseur analogique numérique de 10 bits. >> [5]

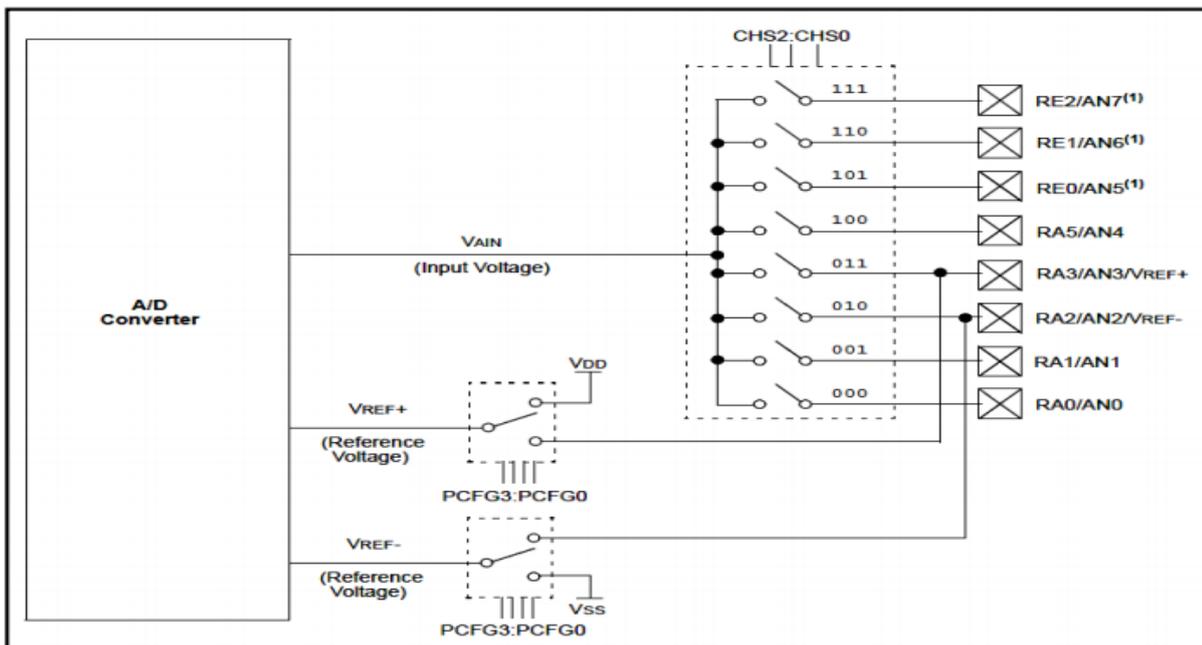


Figure II.11 : Schéma du module de conversion analogique numérique

II.2.11. L'USART

<< L'USART "Universal Synchronous Asynchronous Receiver Transmitter" ou "SCI" en anglais (Serial Communication Interface) est l'un des deux modules de communication série du PIC. Comme son nom l'indique, il peut établir une liaison synchrone ou asynchrone, recevoir et transmettre des données, selon la manière dont il est configuré.

L'USART permet de communiquer avec le reste du monde : un ordinateur ou tout autre matériel équipé d'une interface série RS232, des circuits intégrés convertisseurs Numérique/Analogique ou Analogique/Numérique, des EEPROMs série...

La communication se fait sur les deux broches RC6/TX et RC7/RX qui doivent être configurés toutes les deux en entrée par TRISC.

L'USART peut être configurée selon 3 modes :

- ✓ Asynchrone (Full Duplex).
- ✓ Synchrone Maître (Half Duplex).
- ✓ Synchrone Esclave (Half Duplex). >> [9]

II.2.12. Les modules CCP1 et CCP2

<< Le PIC 16F877A possède deux modules CCP qui fonctionnent de la même façon. Chaque module contient un registre sur 16-bits qui peut fonctionner dans l'un des trois modes ci-dessous :

- Mode capture « CAPTURE »: ce mode permet en outre d'effectuer des mesures de temps.
- Mode comparaison « COMPARE »: ce mode permet en outre de générer des événements périodiques.
- Mode PWM signifie «Pulse Width Modulation», ce qu'on pourrait traduire par modulation de largeur d'impulsion, il s'agit d'un signal binaire de fréquence fixe dont le rapport cyclique peut être modulé par logiciel. Le rapport cyclique d'un signal binaire à fréquence fixe peut être défini comme étant le rapport entre le temps où il se trouve à l'état «1» par rapport au temps total d'un cycle.

Ces modules sont associés aux broches RC2/CCP1 et RC1/T1OSI/CCP2 suivant le mode choisi, les timer1 ou 2 vont être utilisés. Les modes capture et comparaison utilisent le timer1, tandis que le mode PWM utilise le timer2.

II.2.13. Domaines d'application des microcontrôleurs

Les domaines d'application sont extrêmement variés et touchent tous les secteurs : grand public, industriel, informatique, les télécommunications avec les téléphones portables, l'automobile pour toutes les fonctions autour du tableau de bord, de l'ouverture des portes, contrôle des sièges, feux, et autres alarmes.

II.2.14. Les avantages des microcontrôleurs

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts et bien réels. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.

Nous allons voir que le nombre d'entre eux découle du simple sens.

- ✓ Diminution de l'encombrement du matériel et du circuit imprimé
- ✓ Simplification du tracé du circuit imprimé
- ✓ Augmentation de la fiabilité du système
 - nombre de composants diminués
 - connexion composants /supports et composant circuit imprimé diminués
- ✓ Intégration en technologie MOS, CMOS, ou HCMOS
- ✓ diminution de la consommation
- ✓ Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux:
 - moins cher que les composants qu'il remplace
 - Diminution des coûts de main d'œuvre (conception et montage)
- ✓ Environnement de programmation et de simulation évolués

II.2.15. Les défauts des microcontrôleurs

- Le microcontrôleur est souvent surdimensionné devant les besoins de l'application
- Investissement dans les outils de développement.
- Écrire les programmes, les tester et tester leur mise en place sur le matériel qui entoure le microcontrôleur.
- Incompatibilité possible des outils de développement pour des microcontrôleurs de même marque.
- Les microcontrôleurs les plus intégrés et les moins coûteux sont ceux disposant de ROM programmables par masque.
- Le microcontrôleur présente l'avantage des microprocesseurs mais limités aux applications ne nécessitant pas trop de puissance de calcul (architecture courante 8bits).
- Il existe plusieurs architectures de microcontrôleurs de 4 à 128 bits pour les applications demandant une certaine puissance de calcul (injecteurs automobile). >> [5]

II.3. Langage c pour microcontrôleurs

II.3.1. Généralité

Les applications à réaliser par les PIC sont de plus en plus complexes, la programmation en assembleur devient vite très fastidieuse et les réaliser demandent de plus en plus de mémoires.

L'utilisateur est ainsi amené à utiliser les langages "évolués" pouvant simplifier la tâche de programmation. Une fois qu'on a choisi le langage de haut niveau pour la programmation des PIC, il faut se demander s'il existe le compilateur adéquat ?

Exemple:

En Basic : Mikrobasic; Pic Basic.

En Pascal : MikroPascal.

La base du langage est conservée, mais les fonctions et les bibliothèques d'entrée/sortie changent d'un compilateur à un autre.

II.3.2. Le langage c du compilateur CCS

<< Le compilateur C de la société CCS (Custom Computer Services : www.ccsinfo.com) est compilateur C adapté aux microcontrôleurs Pics. Il ne respecte pas complètement la norme ANSI, mais il apporte des fonctionnalités très intéressantes. >> [11]

II.3.3. Les règles de bases

<<

- Toutes instructions ou actions se terminent par un point-virgule ;
- Une ligne de commentaires doit commencer par /* et se terminer par */ ou commence par // norme C++.
- Un bloc d'instructions commence par { et se termine par }. >> [10]

II.3.4. Les variables et les constantes

II.3.4.1. Les constantes

<< Les constantes n'existent pas, c'est-à-dire qu'il n'y a pas d'allocation mémoire, mais on peut affecter à un identificateur (Nom : Il ne doit pas dépasser 32 caractères sans accent)

Une valeur constante par l'instruction **#define**.

Syntaxe : < #define> <identificateur> <valeur> ;

➤ Déclarations spécifiques au compilateur CCS

- #bit id =x, y
- Id : identifiant (Nom d'un bit)
- X : Nom du variable ou d'une constante

- Y : position du bit

Exemple:

```
#bit RW=PORTA, 2
#bit BUZZER =PORTD, 7
```

- #byte id = X
 - Id: identifiant
 - X: valeur 8 bits

Exemple: #byte PORTA = 5 // adresse du port A

```
#byte PORTB = 6 // adresse du port B
#byte PORTC = 7 // adresse du port C
#byte PORTD = 8 // adresse du port D
#byte PORTE = 9 // adresse du port E
```

II.3.4.2. Les variables

Les variables sont définies par signé ou non signé,

Syntaxe : <Signed><type><identificateur1>,....., <identificateur n>

- **L'identificateur:** C'est le nom (il ne doit pas dépasser 32 caractères sans accent) affecté à la variable.
- **Le type :** il détermine la taille de la variable et les opérations pouvant être effectuées. On peut rajouter le mot **signed** devant le type de la variable, alors les variables deviennent signées. >> [11]

➤ **Les types du compilateur CCS :**

Type	Taille	Valeurs
Int1	1 bit	0 OU 1
Int8	8 bits	De 0x00 à 0xFF ou 0 à 255
Int16	16 bits	De 0x0000 à 0xFFFF ou 0 à 65535
Int32	32 bits	De 0x0000000 à 0xFFFFFFFF
Char	8 bits	De 0 à 255 ou une lettre A..... Z
Float	32 bits	Format flottant
Short		Même type que int1
Int		Même type que int8
Long		Même type que int16

Tableau II.4 : Les types des variables [10]

Exemples :

Int A, B, C, D ;

Char MESSAGE [10] ;

- **Les types signés :** Par défaut, tous ces types de données sont signés, ils peuvent être signés en rajoutant le mot clé signed devant le type.

Exemple :

Signed int A ; // Entier de type signé, de -128 à +127

Signed long NB;//Entier de type signé, de -32768 à +32767

- **Les bases du compilateur CCS :**

- ✓ Le décimal: A=10;
- ✓ L'octale : A=012 ;
- ✓ L'hexadécimal: A=0x0A ;
- ✓ Le binaire: A=0b00001010 ;

- **Le caractère :** Exemple la lettre A code ASCII 65(Décimal) ou \$41(Hexadécimal), peut s'écrire :

LETTRE = 65 Ou LETTRE = 0x41 Ou LETTRE = 'A'.

II.3.5. Les opérateurs du langage C

<< Lors de son exécution, un programme est amené à effectuer des opérations qui peuvent être purement arithmétiques, de comparaison, d'affectation, etc.... >> [10]

II.3.5.a. L'opérateur d'affectation

Ces opérateurs permettent d'affecter une valeur à une variable avec possibilité d'effectuer une opération sur cette variable.

Type	Symbole	Exemple
Opérateur d'affectation	=	X=10 ; Y = a + b

- **'=' :** Opérateur d'affectation, la valeur ou l'expression à droite de cet opérateur est affectée à la variable qui est à gauche de cet opérateur.

x=5 ; //quel que soit la valeur de x elle sera remplacée par 5

- **'+=' :** Opérateur addition et affectation, la variable à gauche de cet opérateur sera additionnée avec celle d'à droite, puis le résultat sera affecté à la variable d'à gauche.

x+=2 ; //si x avait la valeur 14, après cette opération x vaudra 16

- **'-=' :** opérateur soustraction et affectation, la variable à droite de cet opérateur sera soustraie de celle d'à droite, puis le résultat sera affecté à la variable d'à gauche.

`x-=2 ;// si x avait la valeur 14,après cette opération x vaudra 12`

- **'*='** : opérateur multiplication et affectation, la variable à gauche de cet opérateur sera multipliée par celle d'à droite, puis le résultat sera affecté à la variable d'à gauche.

`x*=2 ; //si x avait la valeur 14, après cette opération x vaudra 28`

- **'/='** : opérateur division et affectation, la variable à gauche de cet opérateur sera divisée par celle d'à droite, puis le résultat sera affecté à la variable d'à gauche.

`x/=2 ; //si x avait la valeur 14, après cette opération x vaudra 7`

- **'%='** : opérateur reste de la division et affectation, le reste de la division de la variable qui est à gauche de cet opérateur par celle qui est à droite sera affecté à la variable qui est à gauche.

`x%=2 ;//si x avait la valeur 15, après cette opération x vaudra 1`

- **'&='** : opérateur ET bit à bit et affectation. Cet opérateur effectue le ET logique (bit à bit) entre la variable à sa gauche et celle à sa droite, ensuite il affecte le résultat à la variable à gauche.

`x &= 0b11110001 ;//si x avait la valeur 0b10001111,après cette opération x //vaudra 0b10000001`

- **'|='** : opérateur OU bit à bit et affectation. Cet opérateur effectue le OU logique (bit à bit) entre la variable à sa gauche et celle à sa droite, ensuite il affecte le résultat à la variable à gauche.

`x |= 0b11110000 ;//si x avait la valeur 0b10001110,après cette opération x //vaudra 0b11111110`

- **'^='** : opérateur XOR et affectation. Cet opérateur effectue le XOR logique (bit à bit) entre la variable à sa gauche et celle à sa droite, ensuite il affecte le résultat à la variable à gauche.

`x ^= 0b11111000 ;//si x avait la valeur 0b00011110,après cette opération x //vaudra 0b11100110`

- **'<<='** : opérateur de décalage à gauche et affectation. Les bits de la variable à gauche vont être décalés à gauche, le nombre de cases de décalage est indiqué par la variable à droite.

`x <<= 0b00000010; // si x avait 0b00011110, après cette opération x vaudra//0b01111000.`

'>>=': opérateur de décalage à droite et affectation. Les bits de la variable à gauche vont être décalés à droite, le nombre de cases de décalage est indiqué par la variable à droite.

`x >>= 0b00000010; // si x avait 0b00011110, après cette opération x vaudra // 0b00000111.`

II.3.5.b. Les opérateurs arithmétiques :

Type	Symbole	Exemple
Addition	+	a = a + b ; x = 5+ a
Soustraction	-	a = a - b ; y = c - 5
Moins unaire	-	a = - b
Multiplication	*	a = a * a ; b = y * 8
Division	/	c = 9 / b ; d = a / b
Reste de la division entière	%	r = a % b

Tableau II.5: Les opérateurs arithmétiques [11]

II.3.5.c. Les opérateurs de comparaison ou relationnels

Type	Symbole	Exemple
Egalité	==	a == b ; c == 9
Différent	!=	c != a
Supérieur	>	a > b ; 8 > a
Supérieur ou égal	>=	a >= b ; 8 >= a
Inférieur	<	a < b ; 8 < a
Inférieur ou égal	<=	a <= b ; 8 <= a

Tableau II.6: Les opérateurs de relationnels [11]

II.3.5.d. Les opérateurs logiques

Type	Symbole
Et logique	&&
Ou logique	
Non logique	!

Tableau II.7: Les opérateurs logiques [11]

II.3.5.e. Les opérateurs binaires bit à bit

Type	Symbole	Exemple
Et binaire	&	x = y & z
Ou binaire		x = y z
Ou exclusive	^	x = y ^ z
Complément à 1	~	x = b ~ z
Décalage de n bits à droite	>>	x = b >> n
Décalage de n bits à gauche	<<	x = b << n

TableauII.8: Les opérateurs binaires

II.3.6. Les structures répétitives

Le langage “C” possède des instructions permettant de répéter plusieurs fois une même séquence en fonction de certaines conditions.

II.3.6.1. Structure “while” : tant que ... faire ...

Avec ce type d’instruction le nombre de répétitions n’est pas défini et dépend du résultat du test effectué sur la condition. Si cette dernière n’est jamais vérifiée, la séquence n’est pas exécutée.

```
while (int x!=0)
{ ... .. }
```

La structure précédente répète la suite d’instruction comprise entre crochets tant que la variable entière “x” est différent de 0.

II.3.6.2. Structure “do ... while” : faire ... tant que...

Cette structure ressemble fortement à la précédente à la seule différence que la séquence à répéter est au moins exécuter une fois même si la condition n’est jamais vérifiée.

```
do { ... }
while (int x!=0);
```

II.3.6. 3. Structure “for” : Pour <variable> allant de <valeur initiale> à <valeur finale> faire...

Cette instruction permet de répéter, un nombre de fois déterminé, une même séquence.

```
for (i=0;i<5;i++)
{ ... }
```

La structure précédente répète 5 fois la suite d’instruction comprise entre crochets. La variable “i” prendra les valeurs successives de : 0, 1, 2, 3 et 4.

II.3.6.4. Les structures alternatives.

Ces structures permettent d'exécuter des séquences différentes en fonction de certaines conditions.

II.3.6.5. Structure "if ... Else" : Si <condition> faire ... sinon faire ...

Avec cette structure on peut réaliser deux séquences différentes en fonction du résultat du test sur une condition.

```
if (a<b) c=b-a;  
else c=a-b;
```

La structure précédente affecte la valeur "b-a" à "c" si "a" est inférieur à "b" sinon "c" est affecté par la valeur "a-b".

II.3.6.6. Structure "switch ... case"

Cette structure remplace une suite de "if ... else if ...else" et permet une de réaliser différentes séquences appropriées à la valeur de la variable testée.

```
switch (a)  
{  
case '1' : b=16;  
case '2' : b=8;  
case '3' : b=4;  
case '4' : b=2;  
}
```

Dans la structure précédente "b=16" si "a=1", "b=8" si "a=4" etc.

II.3.7. Les fonctions adaptées aux microcontrôleurs PICs :**II.3.7.a. La gestion des entrées et des sorties :**

<< Les fonctions suivantes permettent d'agir sur les ports d'entrées et de sorties :

- Output_low ();
- Output_high ();

- Output_bit (pin_xx, 0 ou 1);
- Input (pin_xx);
- Output_x (valeur); // X : nom de port A...E
- Input_x ();
- Set_tris_x (valeur) ; // Valeur : définit la configuration de port (1 entrée et 0 sortie).

Exemples 1 :**1) Test la broche RB4.**

```
RES = input(PIN_B4) ;// RES = 0 si la broche est à 0 et 1 si elle est à 1
```

Autre façon d'écrire,

```
#define SENS PIN_B4
```

```
RES = input(SENS) ;// RES = 0 si la broche appelée SENS est à 0 et 1 si elle est à 1
```

2) Mise à 1 de RA5.

```
output_high(PIN_A5) ;
```

Autre façon d'écrire,

```
output(PIN_A4,1) ;
```

Autre façon d'écrire,

```
#define RELAIS PIN_A4
```

```
output_high(RELAIS) ;
```

3) Configuration du PORTC, 4 entrées et 4 sorties.

```
#byte PORTC = 7 // Adresse mémoire du PORTC
```

```
set_tris_c(0b11110000) ; // Configuration du PORTC
```

II.3.7.b. La gestion des temporisations

Le compilateur intègre des fonctions très pratiques pour gérer les délais :

- delay_cycles (valeur); // temporisation en NB de cycles
- delay_us (valeur); // temporisation en µs
- delay_ms (valeur); // temporisation en ms

Pour pouvoir utiliser ces fonctions, il faut indiquer la fréquence du Quartz de votre application, cette ligne doit être définie au début du programme.

#use delay (clock = fréquence_de_quartz) ;

Exemples :

```
#use delay (clock=4000000); // quartz de 4 MHz
#use delay (clock = 20000000); //quartz en 20 MHz
```

II.3.7.c. La gestion de la liaison série :

Toutes les fonctions d'entrée et de sortie peuvent être redirigées sur le port série du microcontrôleur, il suffit d'ajouter la ligne ci-dessous pour configurer le port série :
#use rs232 (BAUD=9600, xmit=PIN_C6, rcv=PIN_C7) //Cette ligne configure la liaison série de PIC avec une vitesse de 9600 bauds.

Les fonctions suivantes utiliseront le port série comme moyenne de communication :

- **Printf() ;**
- **Putc() ;**
- **getc() ;**
- **Kbhit() ; >> [11]**

II.3.7.d. Afficheur LCD et le PIC

<< Pour que les affichages de données soient plus attrayants on peut utiliser un afficheur LCD. Le compilateur CCS est fourni avec un pilote d'afficheur LCD, on va donc utiliser cette bibliothèque de fonction :

- **lcd_init()** : fonction d'initialisation de l'afficheur LCD. A appeler en premier
- **lcd_putc(c)** : fonction qui permet d'envoyer un caractère vers l'afficheur LCD.
- **lcd_putc('\f')** : Les caractères spéciaux « \f » (**effacement écran**), « \n » (**passer à la deuxième ligne**) peuvent être utilisés dans la chaîne de caractères envoyée au LCD.
- **lcd_gotoxy(x,y)** : fonction qui permet de déplacer le curseur de l'afficheur.

Il y a d'autre fonction, pour cela il faut lire le fichier "**lcd.c**".

Exemple:

```
#include <lcd.c> // fichier de la bibliothèque pilote du LCD
```

II.3.7. e. Les interruptions en C pour le PIC

L'exécution d'une fonction d'interruption répond a un évènement qui peut être interne (périphérique : CAN, TIMER, EEPROM, USART, I2C) ou externe (RB0, PORTB) du

microcontrôleur. L'appel d'une fonction d'interruption ne dépend pas de programme principal, mais elle l'interrompt pendant son exécution.

Une fonction d'interruption n'a pas de paramètre d'entrée et de sortie. Le compilateur CCS utilise une directive spéciale `INIT_XXXX` (`XXXX` nom de l'interruption) pour les différencier avec les autres fonctions logicielles.

Syntaxe :

//Nom de la fonction :

//Description du rôle de la fonction :

`#INIT_XXXX` //Nom de l'interruption

Afin de comprendre la configuration et la programmation d'une interruption, on présente. >> [10]

A. Boucle For

1. La ligne 1 précise quel type de µC dans notre cas PIC16F877
2. La ligne 2 précise les « fusibles » à programmer dans le PIC.
3. La ligne 3 permet de préciser à quel fréquence le quartz de votre carte oscille. Dans la plupart des cas cette valeur sera 4000000 pour un quartz à 4MHz ou 20000000 pour un quartz à 20MHz.
4. La ligne 4 déclaration de variable
5. La ligne 5 indique le début de notre programme principal (fonction « main »), vous ouvrez donc une accolade.
6. La ligne 7 permet de faire boucler avec boucle for de l'état initiale 0 jusqu'à l'état finale 255.
7. La ligne 9 permet d'envoyer (output) vers le port D (`_D`)
8. La ligne 10 permet d'attendre le nombre de ms indiqué entre parenthèse (ici 1000ms donc 1s).

B. Boucle while

9. la ligne 6 déclaration de variable x
10. La ligne 7 : permet de faire boucler indéfiniment notre programme (« while (1) » signifie « tant que vrai »). Les instructions entre les accolades seront donc exécutées sans arrêt.
11. la ligne 10 incrémentation de la variable.

Remarque : on dit while (true) ou bien while (1) c'est la même fonction.

➤ **Quelques directives et fonctions du pic C compiler** [10]

1- Directives

#use delay	
Syntaxe	: #use delay (<i>clock=fréquence</i>)
Rôle	: Renseigne le compilateur sur la fréquence du quartz utilisé.
Exemple	: #use delay (clock=4000000)

#fuses	
Syntaxe	: #fuses <i>options</i>
Rôle	: Permet de définir le mot de configuration. Les options sont : <ul style="list-style-type: none"> - LP, XT, HS, RC - WDT, NOWDT - PUT, NOPUT - PROTECT, NOPROTECT.
Exemple	: #fuses XT, NOWDT, NOPUT, NOPROTECT

#int_xxxx	
Syntaxe	: #int_ext : interruption externe.
	#int_RB : changement d'état de RB4 à RB7.
	#int_TIMER0 : débordement du timer0.
	#int_EEPROM : fin d'écriture dans l'EEPROM.
Rôle	: Spécifie la source de l'interruption.

2- Fonctions

BIT_CLEAR()	
Syntaxe	: BIT_CLEAR (<i>var, bit</i>)
Rôle	: Mettre à 0 le bit « bit » de la variable « var ».
Exemple	: a=0x1F BIT_CLEAR(a, 3) a devient 17 hexa.

BIT_SET()	
Syntaxe	: BIT_SET (<i>var, bit</i>)
Rôle	: Mettre à 1 le bit « bit » de la variable « var ».
Exemple	: a=0x1F BIT_SET(a, 6) a devient 3F hexa.

BIT_TEST()	
Syntaxe	: BIT_TEST (<i>var, bit</i>)
Rôle	: Teste l'état du bit « bit » de la variable « var ».
Exemple	: a=0x1F

BIT_TEST(a, 2)

Le résultat est 1 car le bit 2 de la variable « a » est à 1.

DELAY_MS()

Syntaxe : DELAY_MS(x)

Rôle : Temporisation se « x » ms.

Exemple : DELAY_MS(2000)

Temporisation de 2 secondes.

INPUT()

Syntaxe : état=INPUT (pin)

Rôle : Permet de lire l'état d'une broche d'un port préalablement configurée en entrée.

Exemple : a=input(PIN_B0)

la variable « a » l'état de la broche 0 du port B.

INPUT_x()

Syntaxe : etat=INPUT_x ()

Rôle : Permet de lire l'état d'un port (x sur 8 bits) préalablement configuré en entrée.

Exemple : c=input_A ()

la variable « c » l'état du port A (1 octet).

OUTPUT_x()

Syntaxe : OUTPUT_x (valeur)

Rôle : Permet de sortir l'octet «valeur» sur le port x, préalablement configuré en sortie.

Exemple : OUTPUT_B (0x1F)

La valeur 1FH est envoyée sur le port B.

OUTPUT_BIT ()

Syntaxe : OUTPUT_BIT (pin, état)

Rôle : Permet de mettre le pin (pin) à l'état logique (etat).

Exemple : OUTPUT_BIT (PIN_A3,1)

Mettre à 1 la broche 3 du port A.

OUTPUT_HIGH()

Syntaxe : OUTPUT_HIGH(pin)

Rôle : Permet de mettre à 1 la pin (pin).

Exemple : OUTPUT_HIGH(PIN_A3)

Mettre à 1 la broche 3 du port A.

OUTPUT_LOW()

Syntaxe : OUTPUT_LOW(pin)

Rôle : Permet de mettre à 0 la pin (*pin*).
 Exemple : OUTPUT_LOW(PIN_A3)

Mettre à 0 la broche 3 du port A.

SET_TRIS_x()

Syntaxe : SET_TRIS_x(*valeur*)
 Rôle : Configure la direction du port « x ».
 Exemple : SET_TRIS_B(0x0F)

Les pins B7, B6, B5, B4 sont configurées en sortie (0).

Les pins B3, B2, B1, B0 sont configurées en entrée (1).

➤ Avantages du CCS C

- IDE est bien pensé et facile à utiliser.
- Assistant Code pour générer du code C adapté à tout micro PIC vous pouvez travailler avec chaque outil et fonction du compilateur est intégrée proprement dans l'IDE.
- Qualité du code compilé est excellente. L'optimiseur semble faire un bon travail de la compression du code.
- Le support technique est rapide, précis et réactif.
- Beaucoup de fourni un exemple de code C. Le code fourni échantillon C démontre une vaste gamme de techniques et de fonctionnalités pour une bonne gamme de micros PIC.

II.4. Conclusion

Dans ce chapitre on a fait une présentation générale des microcontrôleurs et nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de commande pour notre système. Pour faire fonctionner cette unité de contrôle, il faut la programmer et l'adapter à un compilateur de programmation et nous avons appris à utiliser le programme de CCS et comment exécuter pour chaque instruction.

Le CCS PICC a pour rôle principal de contrôler la cohérence du code au moment de la compilation, c'est-à-dire qu'il offre la possibilité de vérification d'erreurs pour les corriger au moment de la programmation.

Cette étude théorique nous a permis de traiter et de comprendre le fonctionnement utilise Langage c pour microcontrôleurs CCS et des différentes ressources interne du PIC 16F877A.

Chapitre III

Réalisation et simulation la machine de moulage de sable

III.1 Introduction sur la machine de moulage de sable

<< Le terme fonderie définit aussi bien les installations métallurgiques dans lesquelles on fond les métaux et où on les coule dans des moules pour leur donner la forme des objets à fabriquer, que l'ensemble des opérations nécessaires à leur production.

C'est tout à la fois un art et un métier dont les origines remontent à plusieurs millénaires avant notre ère utilisant et mettant en œuvre des techniques aussi diverses que variées, comme le moulage, le noyautage, la métallurgie, la fusion et la coulée des métaux et alliages métalliques, les traitements thermiques, la finition, le parachèvement des pièces, etc.

La fonderie, c'est aussi une industrie produisant des pièces de toutes natures finies ou semi finies, en toutes quantités, de la façon la plus économique à partir de matières premières brutes (métaux, ferrailles, déchets de tôles, bocages), contrairement à certaines techniques concurrentes (forgeage, mécano soudage...) utilisant des produits semi-finis comme les tôles neuves, les barres et billettes en acier de toutes nuances, élaborés par la sidérurgie.

On s'adresse à elle soit pour la fabrication de pièces de formes compliquées qu'il serait difficile ou impossible de réaliser partout autre procédé, soit pour la production à des prix de revient plus intéressants de pièces plus simples, soit encore pour profiter des propriétés physiques ou d'utilisation (dilatation, corrosion, frottement, usure, basses et hautes températures) et des propriétés mécaniques des divers métaux et alliages coulés, soit enfin pour couler des alliages non corrodables ou difficilement usinages .

De plus, la fonderie est une industrie économe en énergie, grâce à l'utilisation immédiate du métal liquide dont l'élaboration peut nécessiter, par exemple 650kWh/t pour de la fonte grise au gros four à induction basse fréquence.

A cela s'ajoute les progrès fait en fonderie, notamment dans le domaine de la précision dimensionnelle, qui entraîne souvent une réduction des dépenses définition (diminution ou suppression de l'usinage par exemple).

Après une présentation exhaustive des procédés de moulage, nous étudierons plus précisément la technique du moulage par basse pression. >> [12]

III.1.1. Description de la machine architecture

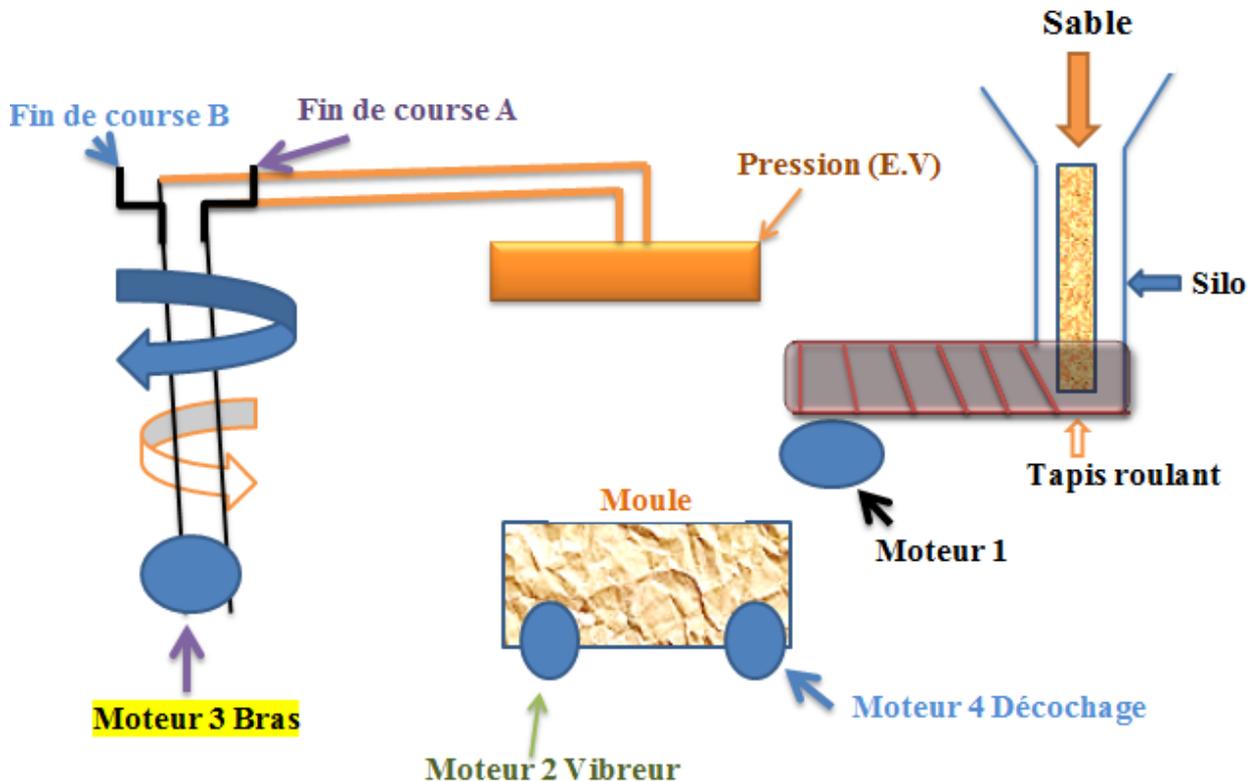


Figure. III.1 : Schéma Machine de moulage de sable

III.1.2. Description de la machine organisation

Dans la fig1 on à réaliser un schéma et un programme Pour commande machine de moulage de sable, Dans ce résumé on à expliquer les étapes suivantes :

- **Etape 01:** Lorsque on appui sur le bouton 'marche' le tapis roulant démarre, il porte avec lui du sable puis remplit le sable dans le moule. Quand on remarque que le moule est rempli on fait un appui sur le bouton « Arrêt » du moteur, le tapis s'arrête.
- **Etape 2:** Lorsque on appui sur le bouton 'marche' du moteur vibreur, le sable dans le moule est réparti sur la surface du moule puis on appui sur le bouton « Arrêt » du moteur de vibreur pour l'arrêter.
- **Etape 03:** Lorsque on appui sur bouton « auto marche » du système séquentiel automatisée, le moteur de bras étant initialement en position de fin de course B avance vert le fin cours A avec un angle de 90 degrés puis démarrage une électrovanne (LED vert) pour presser le sable dans le moule pendant 30s. Puis le bras de robot fait le chemin inverse pour rejoindre le fin cours B.
- **Etape 04:** En appuyant du moteur découchage le moule décoche le moule à son état initial.

Dans cette opération on a fait l'étude par deux méthodes différentes qui sont :

- ✓ Le programme C.
- ✓ Le programme ARDUINO.

III.1.3. Organigramme machine de moulage du sable

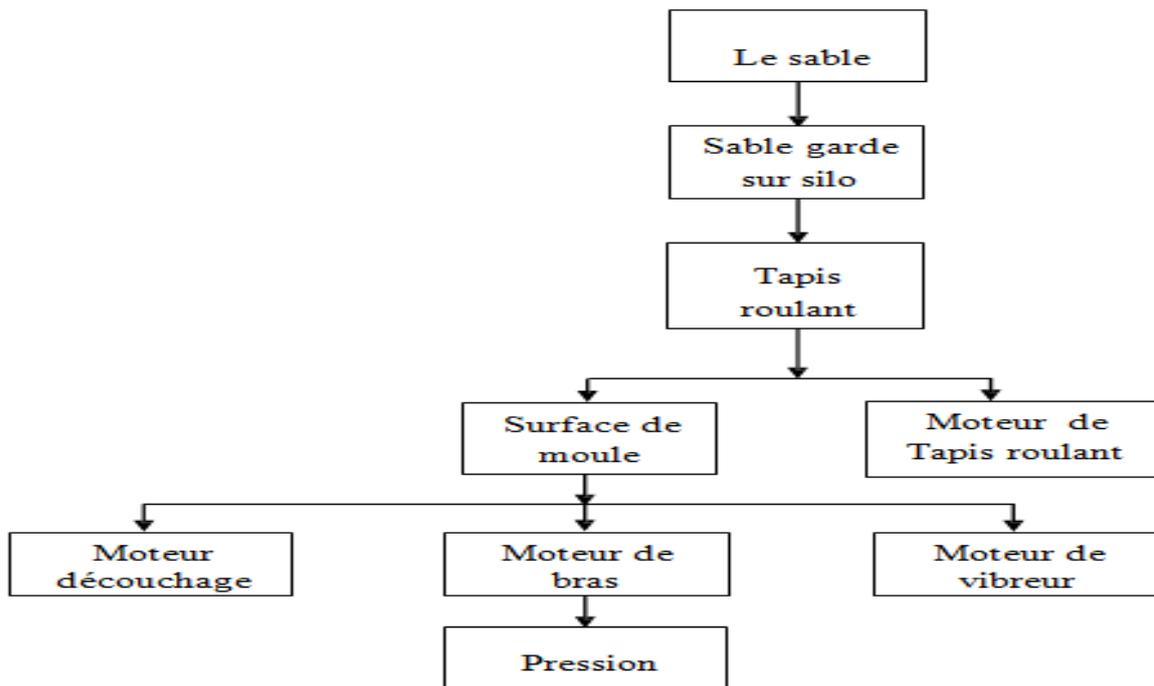


Figure. III.2 : Organigramme machine de moulage de sable

III. 2. Vue d'ensemble du projet

III.2.1. Les différents blocs

Notre carte comme l'indique la figure ci-dessous est menue de plusieurs unités qui assurent le bon fonctionnement de la carte de commande de Machine de Moulage :

- ✚ Une carte d'alimentation « alimentation stabilisé ». Ce bloc a pour but l'alimentation des différents blocs de la carte électronique.
- ✚ Un microcontrôleur PIC 16F877A ou ARDUINO UNO qui gère tous les traitements et les liaisons entre les différents blocs de la carte.
- ✚ Un afficheur LCD (2 lignes, 16 caractères) qui permet d'afficher les résultats, le traitement faisant intervenir le microcontrôleur PIC ou ARDUINO UNO.

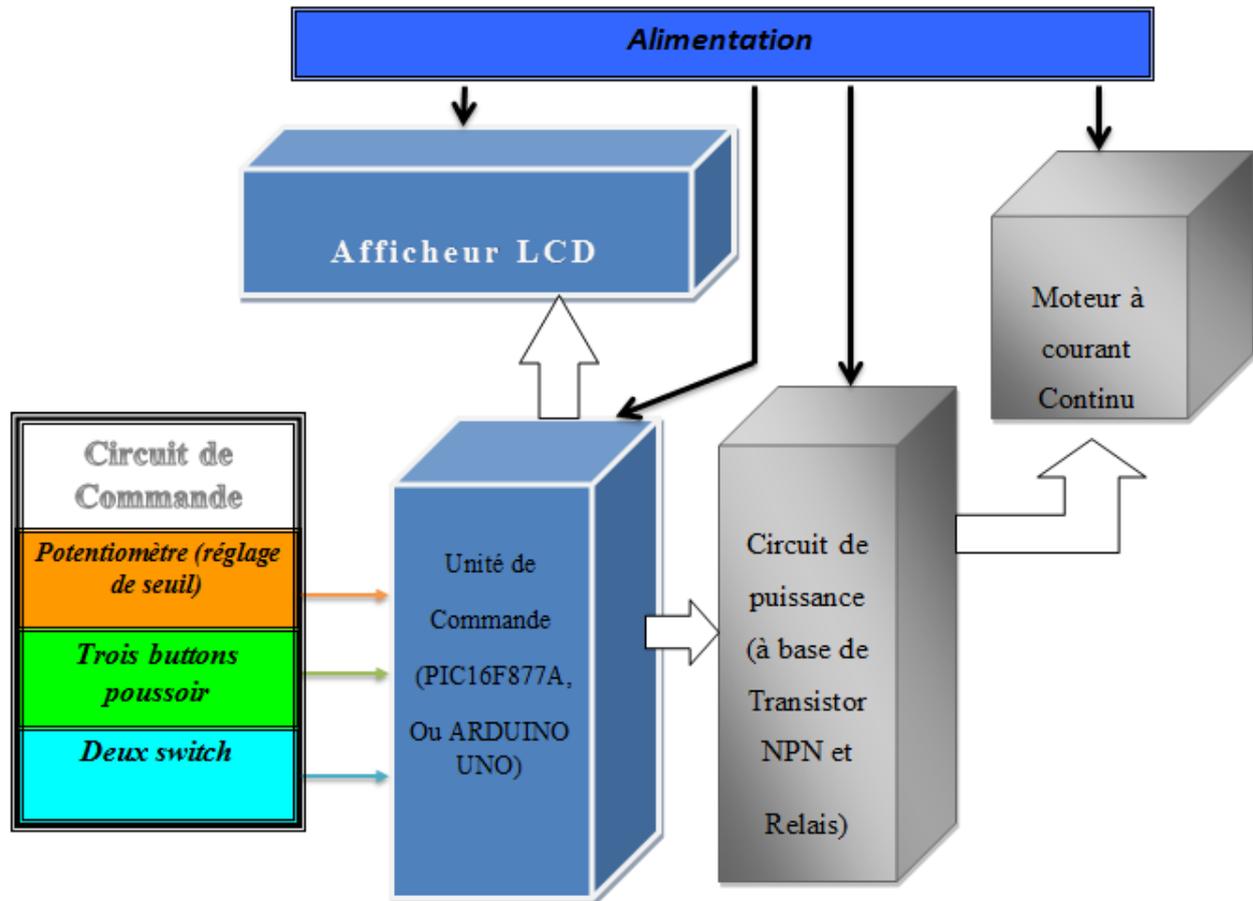


Figure. III.3 : Schéma synoptique

III.2.2. L'organigramme sur lequel est basé ce programme

- **B1 / B2 / B3** : Boutons poussoirs.
- **M1 / M2 / M3 / M4** : Moteurs.
- **S_A / S_B** : Switch.
- **Sup** : Supprime.
- Marche **D** : Marche moteur sens droit.
- Marche **G** : Marche moteur sens gauche.

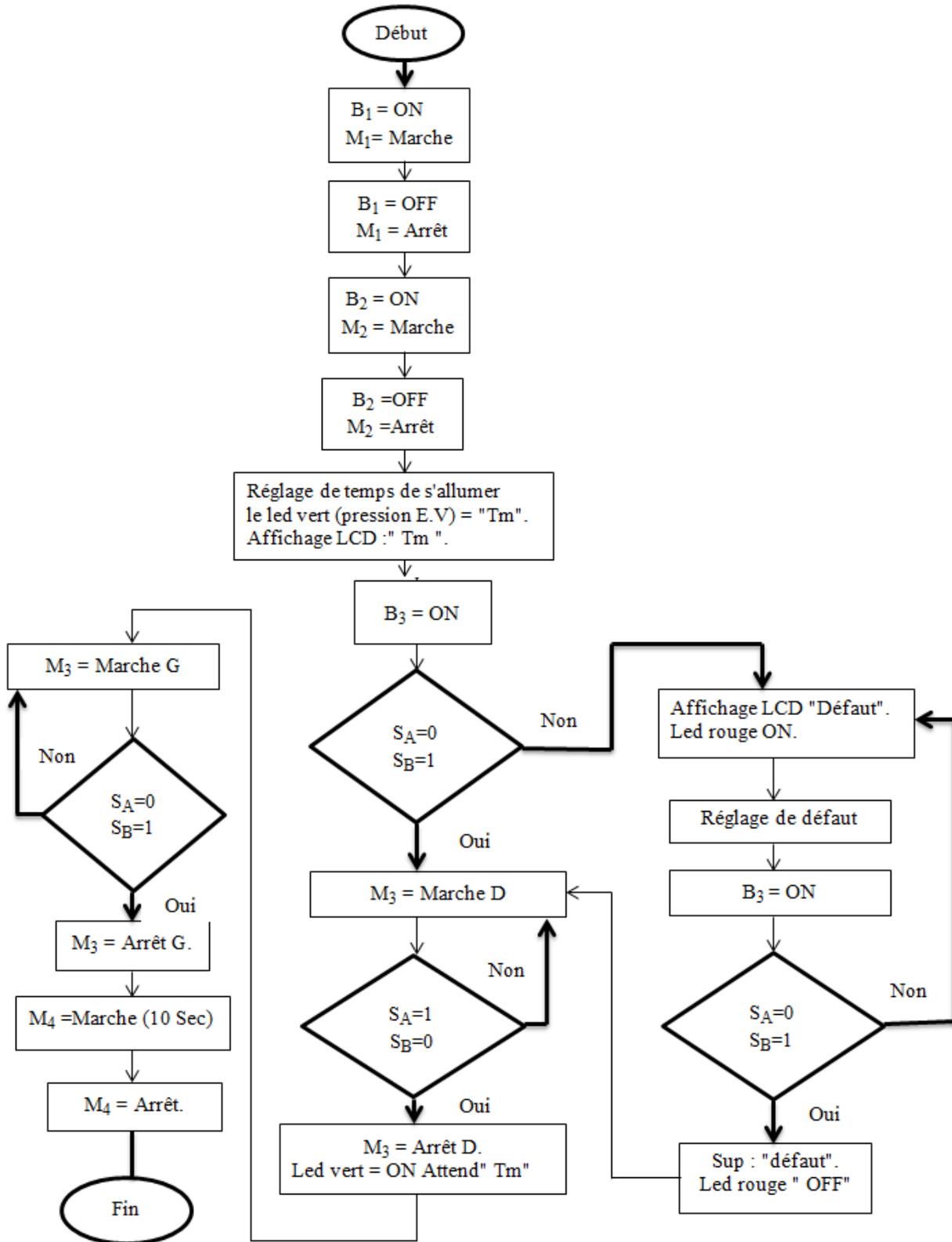


Figure. III.4 : Organigramme de programme

III.2.3. Grafcet machine de moulage de sable

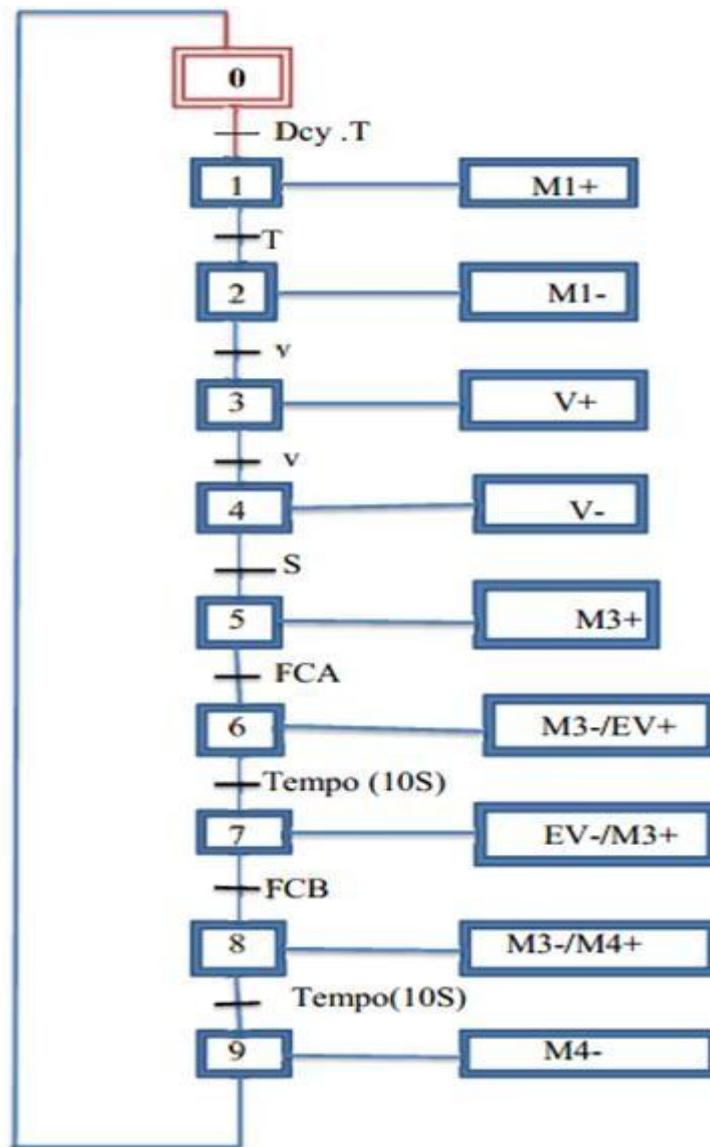


Figure. III.5 : grafcet machine de moulage de sable

T : commande de tapis roulant

V : commande de vibreur

S : séquence automatisée

Tempo : temporisateur

FCA : fin de course A avancée

FCB : fin de course B arrive

III.3. Simulation schéma électronique du système sur PROTEUS

III.3.1. Schéma sur Isis PIC16F877A

❖ La programmation de PIC

L'emploi d'un microcontrôleur de type 16F877A simplifie à l'extrême le schéma de principe du montage. Il est relié aux autres blocs, le programme implémenté dans sa mémoire EEPROM assure le bon fonctionnement de l'ensemble.

Après avoir réalisé la carte de notre système, il ne reste plus qu'à programmer le microcontrôleur avec le logiciel PCM compiler de la compagnie CCS (Costum Computer Service), est distribué gratuitement sur le site web : www.ccsinfo.com/download.

Cela consiste à créer un programme avec langage C et entrer la référence du PIC, la fréquence de l'horloge et les différents modes.

Après l'édition du programme, sa compilation permet de voir s'il n'y a pas d'erreurs. Si tout est correct, le fichier format hexadécimal (.hex) est créé. En effet cette étape est indispensable pour que le programme soit reconnu et exécuté par le microcontrôleur.

Ainsi, une fois le fichier hexadécimal créé, on le transfère vers la mémoire programme du PIC.

La mémoire programme du PIC utilisée : 58% (mémoire ROM)

La mémoire de donnée utilisée : 26% (Mémoire RAM)

La simulation du programme peut se faire facilement avec le logiciel ISIS de PROTEUS.

❖ Simulation dans ISIS

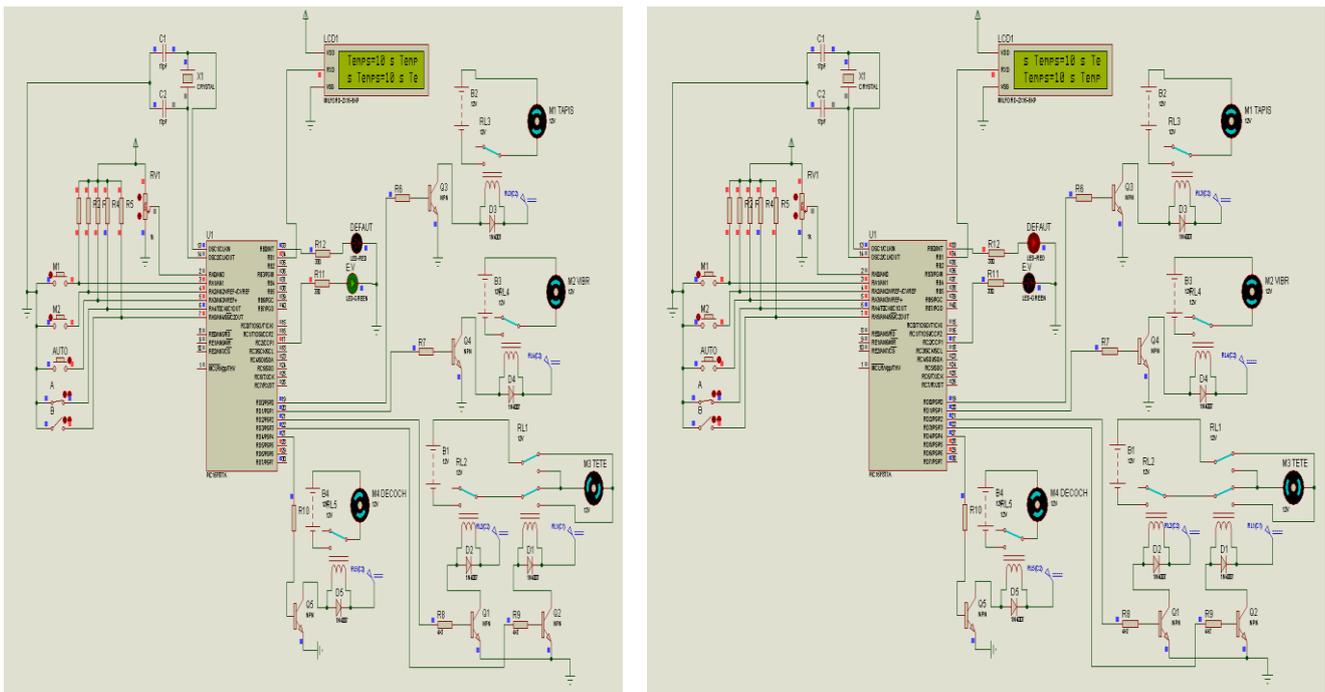


Figure. III.6 : schéma sur Isis pic 16F877A

Switch A	Switch B	Led rouge (defaut)
0	0	Allume
1	0	Allume
0	1	Ne pas allume
1	1	Allume

III.3.2. Schéma sur Isis ARDUINO

❖ Ecriture du programme

La programmation de l'ARDUINO UNO se fait avec le logiciel ARDUINO. Après l'édition du programme, sa compilation permet de voir s'il n'y a pas d'erreurs sur les syntaxes. Si tout est correct, un fichier en format hexadécimal (.hex) sera créé. Cette étape est indispensable pour que le programme soit reconnu et exécuté par l'ARDUINO UNO. Ainsi, une fois le fichier hexadécimal créé, le transfert vers la mémoire programme du ARDUINO pourra ont se faire.

❖ Simulation dans ISIS

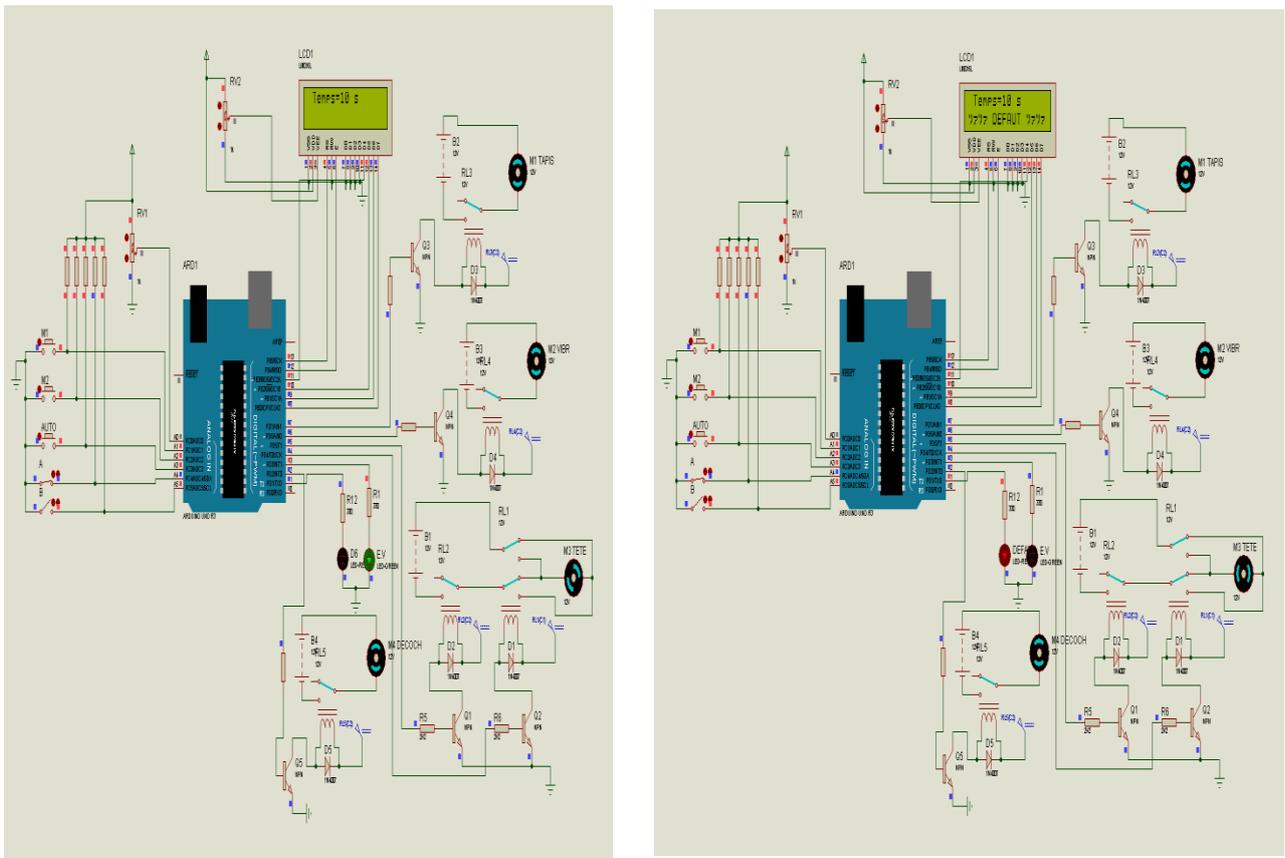


Figure. III.7 : schéma sur Isis Arduino

❖ Le même résultat dans simulation que pic 16F877A.

III.3.3. Explication des schémas

Les figures 6 et 7 montre deux schémas réalisé dans le logiciel ‘ ‘Isis porteuse‘ ’ et avec deux composants différents qui sont respectivement le ‘ ‘PIC16F877A’ ’ et la ‘ ‘carte ARDUINO’ ’. Dans les années précédentes le ‘ ‘PIC16F877A’ ’ est utilisé pour commander les composant de la station de moulage, mais cette méthode est complexe et difficile alors grâce au développement des technologies et de l’informatique ces difficultés sont dépassées en utilisant une "carte ARDIUNO" qui remplace le ‘ ‘PIC16F877A’ ’ pour les avantages qu'elles représentent qui sont cités ci-dessus:

- Simplicité de montage.
- le facteur de temps.
- le développement de ces composants.
- la simplicité dans création de programmes.

❖ Interface commande puissance

Dans notre application nous avons utilisé un relais piloté via un transistor de commande, une diode de roue libre est montée aux bornes du relais pour la protection du relais. Une résistance lié à la base du transistor pour le protéger contre le sur intensité comme il est indiqué sur la figure ci-dessous :

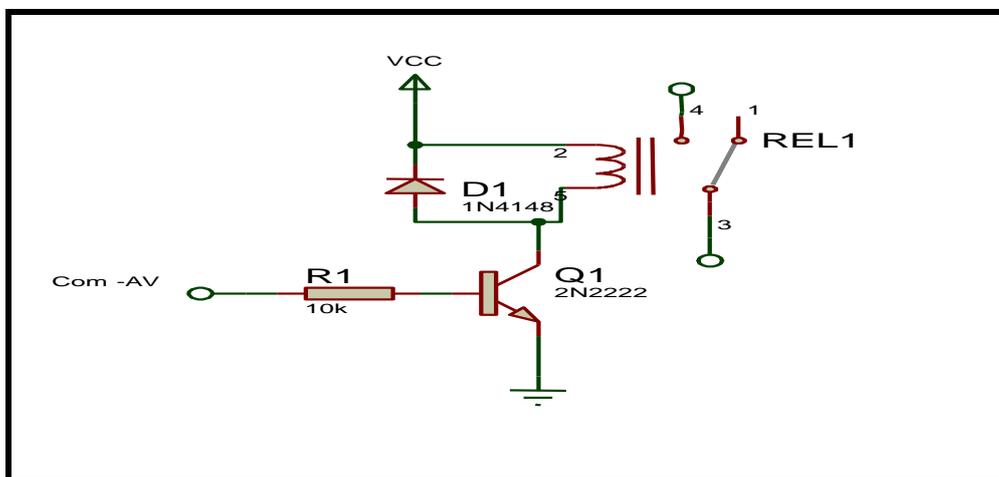


Figure. III.8 : schéma de puissance sur Isis

❖ Les Relais

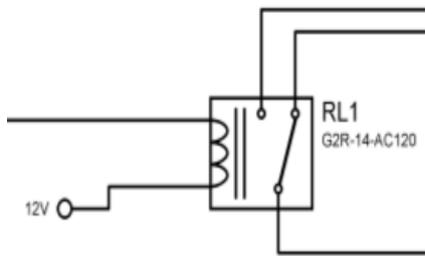


Figure. III.10 : schéma interne de relais

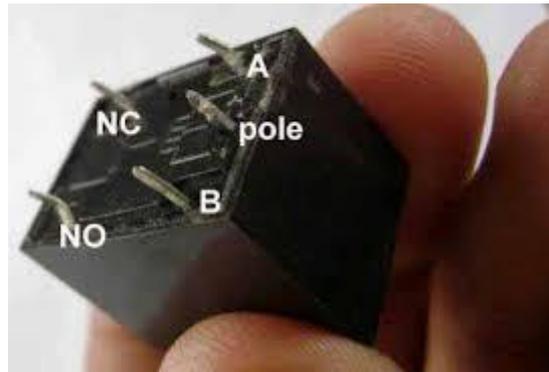


Figure. III.9 : schéma réel de relais

C'est un appareil composé d'une bobine (électroaimant) qui lorsqu'elle est parcourue par un courant électrique, elle agit sur un ou plusieurs contact. Un phénomène électrique (courant ou tension) contrôle la commutation On / Off d'un élément électrique (relais statique).

Comme la commande peut être réalisée sous faible puissance (faible tension, faible courant), et que la partie coupure peut commuter des puissances importantes, on peut dire que ce composant est un amplificateur de courant.

Dans le cas des relais électromécaniques, la bobine peut générer de fortes surtensions au moment où le courant cesse de la traverser. Cette surtension (qui peut atteindre plusieurs dizaines de volts ou plus de 100 volts, même avec une alimentation de 12V) peut détruire le transistor ou la porte logique qui la commande. Pour éviter tout risque de destruction de la commande électronique qui précède le relais, il est d'usage de placer une diode dite de roue libre, en parallèle sur la bobine du relais. Cette diode doit être câblée en inverse, c'est à dire cathode vers le pôle le plus positif de l'alimentation (si vous la branchez à l'envers, elle grillera instantanément dès la première activation de la commande).

- Avantages du relais électromécanique :

Capacité de commuter aussi bien des signaux continus qu'alternatifs sur une large gamme de fréquences.

Fonctionnement avec une dynamique considérable du signal commuté.

- Aucun ajout de bruit ou de distorsion.
- Résistance de contact fermé très faible.
- Résistance de contact ouvert très élevée.

❖ Le potentiomètre de réglage

➤ Définition

Un potentiomètre est un type de résistance variable à trois bornes, dont une est reliée à un curseur se déplaçant sur une piste résistante terminée par les deux autres bornes.



Figure. III.11 : schéma résistance variable

➤ **Fonctionnement**

Les potentiomètres sont couramment employés dans les circuits électroniques. Ils servent par exemple à contrôler le volume d'une radio. Ils convertissent une position en une tension.

❖ **Les Résistances**

➤ **Descriptions**

La résistance (Résistor) est l'élément le plus simple, très utilisé en électronique. C'est un composant dit passif, il conduit l'électricité avec un effet résistif. Il est bidirectionnel, il n'y a pas de sens obligatoire du passage du courant.

➤ **Symboles**

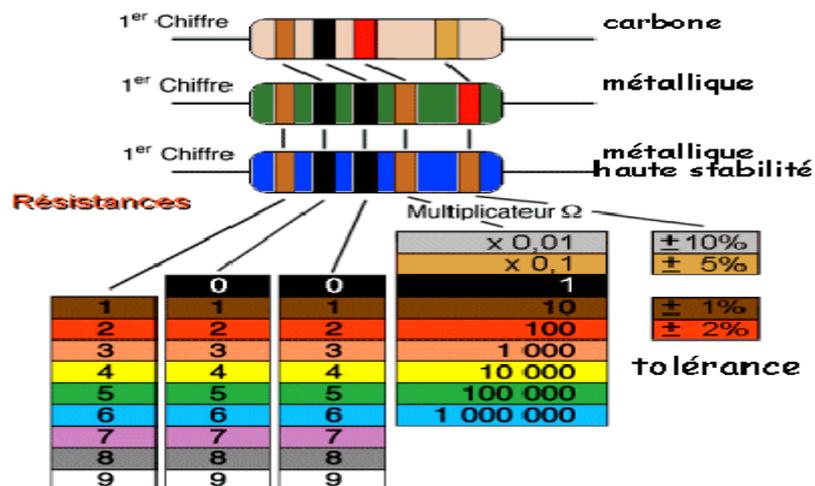


Figure. III.12 : Les valeurs des différentes résistances du montage

❖ **Les diodes**

➤ **Descriptions**

Les D.E.L (Diode Electro Luminescente) ou en Anglais : L.E.D (Light Emitting Diode) éclairent lorsqu'elles sont parcourues par un courant de l'anode vers la cathode.

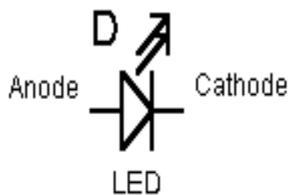


Figure. III.14 : Schéma interne de Diode

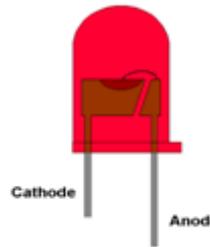


Figure. III.13 : Schéma réel de Diode

➤ Utilisations

L' avantage d' utiliser des leds est qu' elles ne s' usent pas , elles sont moins chères que des voyants, elles consomment moins d' énergie .Mais l' inconvénient est qu'elles ne peuvent fonctionner qu'avec une faible tension , et qu' elles n' éclairent pas beaucoup par rapport aux ampoules classiques.

III.3.4. Afficheur LCD

▪ Présentation

Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitant peu de composants externes pour un bon fonctionnement. Ils sont relativement bon marché et s' utilisent avec beaucoup de facilité.

Un microprocesseur en technologie CMOS diminue considérablement leur consommation (<0.1mW) ils sont pratiquement seuls à être utilisés sur les appareils à alimentation par piles. Plusieurs afficheurs sont disponibles sur le marché et ne diffèrent les uns des autres, non par leurs dimensions (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristique technique et leurs tensions de service. Certains sont dotés d' un rétro- éclairage de l' affichage. Cette fonction fait appel à des LEDs montées derrière l' écran du module, cependant cet éclairage est gourmand en intensité (250 mA max).



Figure. III.15 : Afficheur LCD avec 2 lignes de 16 caractères

▪ Principe de fonctionnement

Un afficheur LCD est capable d' afficher tous les caractères alphanumériques usuels et quelques symboles supplémentaires. Pour certains afficheurs, il est même possible de créer ses propres caractères.

Chaque caractère est identifié par son code ASCII qu'il faut envoyer sur les lignes D0 à D7, broches 7 à 14. Ces lignes sont aussi utilisées pour la gestion de l'affichage avec l'envoi d'instructions telles que l'effacement de l'écran, l'écriture en ligne 1 ou en ligne 2, le sens de défilement du curseur.

- **Description**
- **Connexion de l'afficheur**

Pour utiliser le LCD il faut bien sur l'alimenter par 5V et la masse. Ensuite il va falloir relier les ports RS, R/W et E au PIC (pin RA0, RA1 et RA2) puis les ports D0 à D7.

- **Utilisation des ports pour envoyer une instruction**

Pour envoyer une instruction à l'afficheur il faut régler le bit READ/WRITE (généralement à WRITE) spécifié le bit RS (si c'est une commande ou une donnée), mettre les bits du poids fort de la commande ou la donnée en question sur les bits D7 à D0 et enfin valider pour que l'afficheur en tienne compte.

La validation se fait en mettant le bit E à 1 puis en le remettant à 0. Il y a tout de même des contraintes de temps à respecter (il faut un certain temps aux LCD pour comprendre et exécuter une commande).

- **La commande**

Après avoir alimenté le LCD il faudra envoyer une série de commande pour l'initialisation. Si la phase de l'initialisation est terminée, on peut envoyer directement des codes ASCII pour que le LCD les affiche.

Le programme n'est ni gourmand en taille mémoire (moins de 270 instructions, en comptant les timers), ni en RAM.

L'exécution est préalablement plus longue mais le coût de l'affichage se fait à temps court.

III.3.5. Génération de typon de la carte avec Ares

Pour préparer le typon on a recours au même logiciel de simulation qui est ISIS produit par la société LABCENTER ELECTRONICS.

L'ISIS est un éditeur de création de schémas électroniques ; il fait de même toutes les manipulations de simulation analogique et logique à fin d'atteindre un résultat bien précis.

Un typon est une image du circuit imprimé. Les pistes sont en noir et le reste doit être transparent à la lumière. L'idéal est d'imprimer avec une imprimante jet d'encre sur du papier transparent spécial jet d'encre. Il faut régler l'impression sur "papier photo épais" pour avoir un maximum d'encre sur le papier et obtenir un noir plus opaque. La photocopieuse et l'imprimante laser convient aussi.

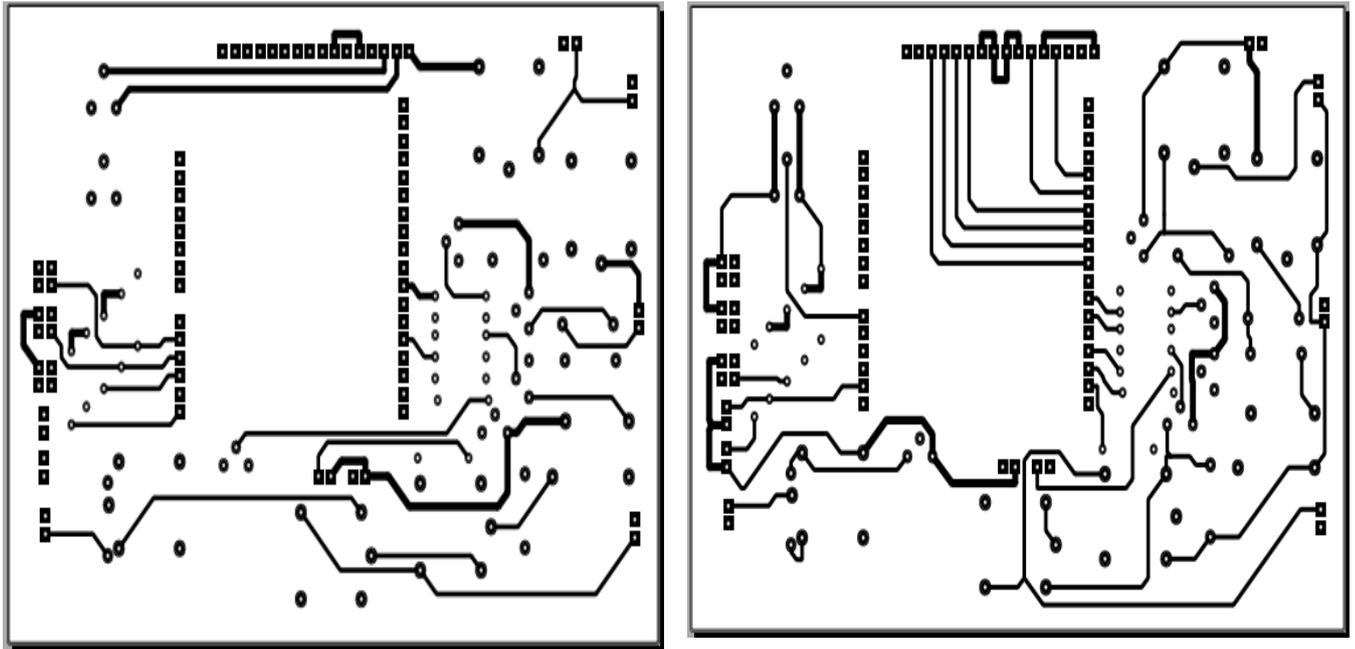


Figure. III.16 : Schéma du typon

III.4. Réalisation schéma électronique du système

❖ Composant utilisés dans la carte électronique

Composant	Définition	Quantité
Arduino	UNO	01
Afficheur LCD	16x2	01
Moteurs	DC 12V	04
Bouton poussoir		03
Résistance	4,7K Ω	05
	360 Ω	02
	1K Ω	04
	1,2K Ω	01
LED	Rouge	01
	Vert	01
Potentiomètre	10K Ω	02
Transistor NPN	2N1711	05
Diode	1N4007	05

Relais	DC 12V	06
Switch		02

Tableau III.1: Composant utilisés dans réalisation

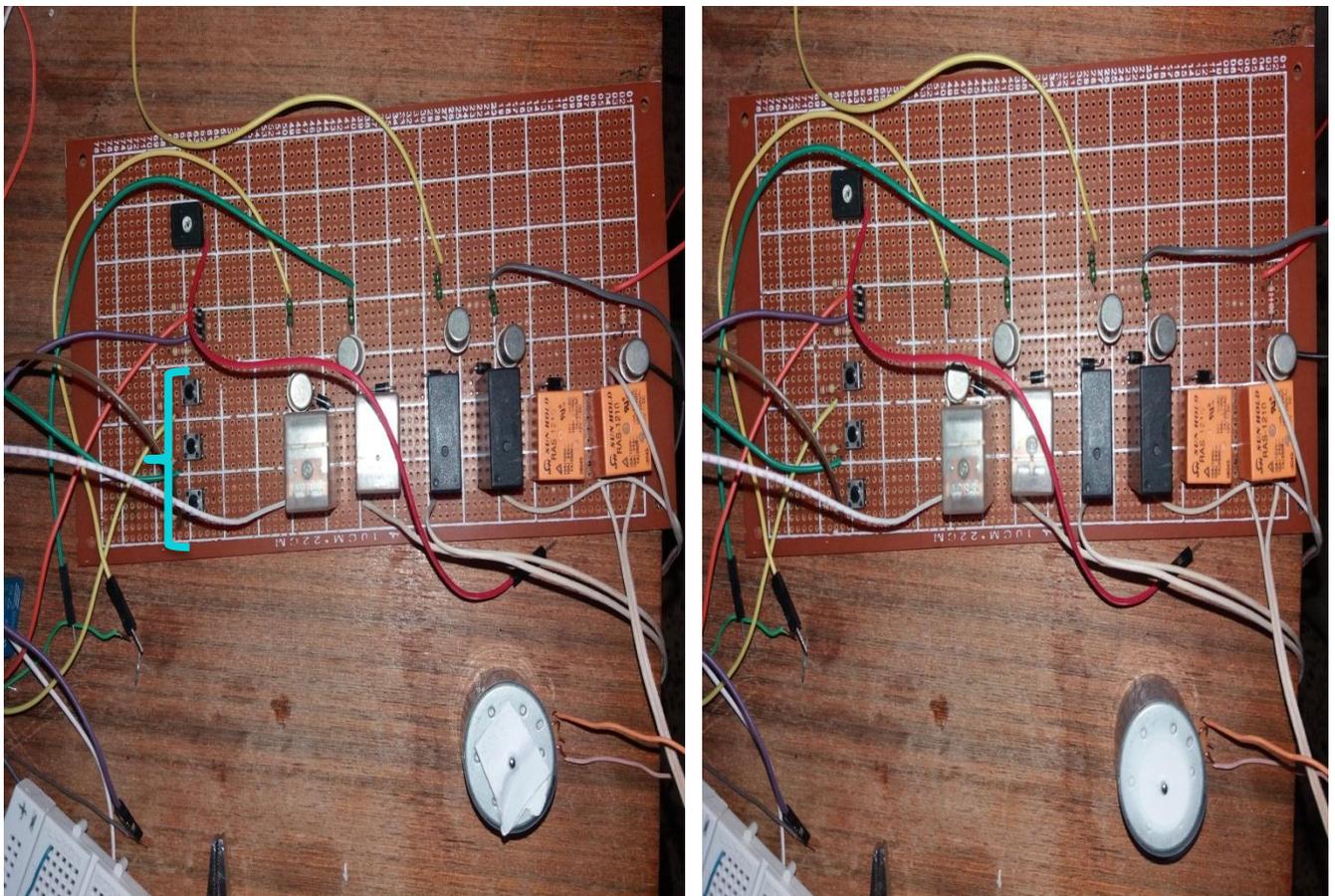
❖ Test final

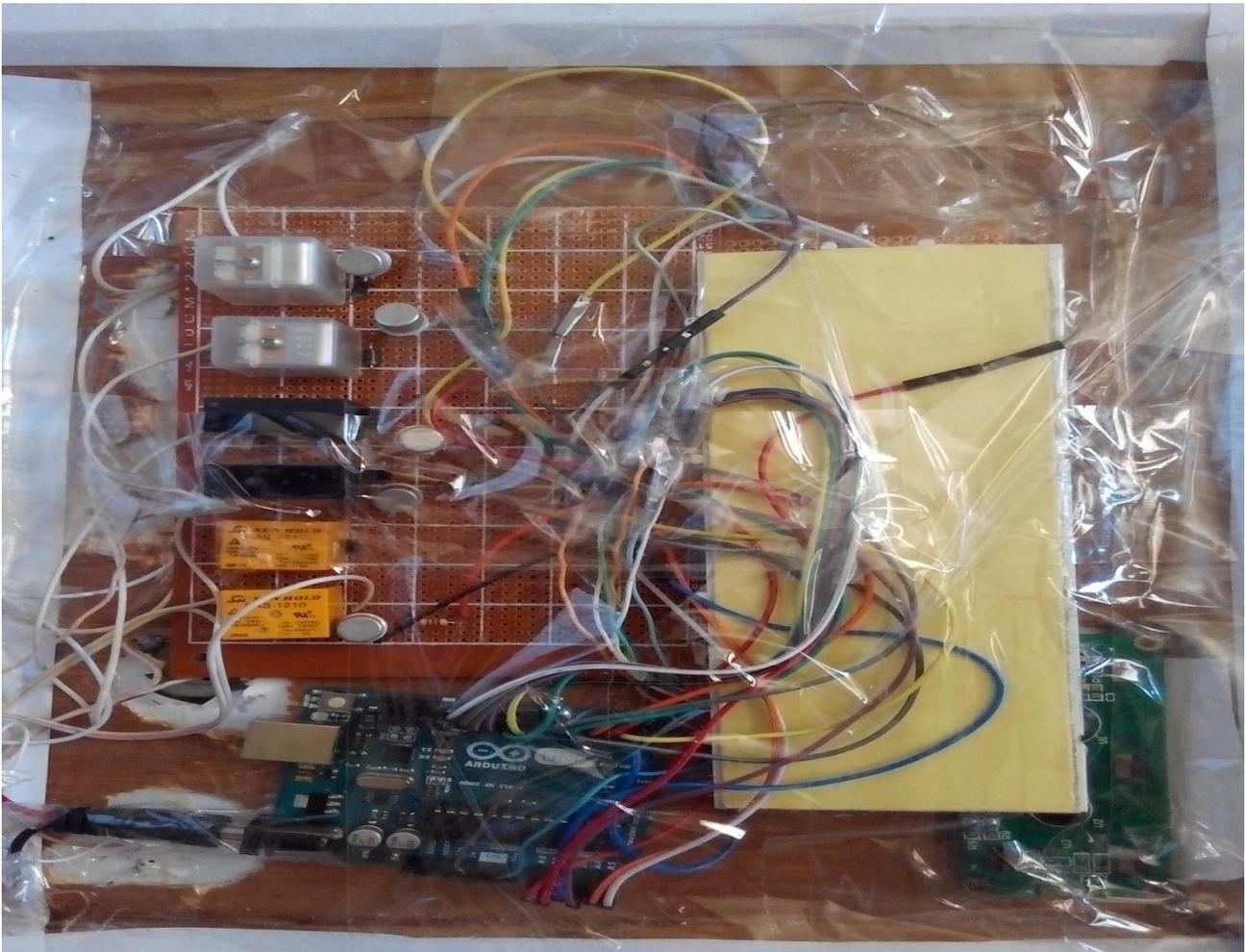
Après la réalisation de la carte électronique, il est important de ne pas avoir un court-circuit entre les pistes du circuit. On contrôlant la continuité de toutes les pistes les tests finals peuvent être faits.

Le premier test est de contrôler chaque partie du système indépendamment, c'est-à-dire, de tester la partie communication en exécutant un programme pour l'affichage d'un message, par exemple, sur l'écran LCD, les relais sera testé avec les moteurs, qui déterminerons si le transistor NPN fonctionne bien et si le seuil est respecté par le contrôleur, Le deuxième test sera d'additionner différentes parties du système, puis on ajoutera le circuit référence pour voir le fonctionnement des boutons poussoir destiné à varier la temps de référence.

Après s'être assuré que toutes les parties ont fonctionnelles et que le circuit ne présente aucun défaut, l'assemblage de toutes les parties du système pourra ont être réalisé.

- Teste le circuite de puissance avec moteur DC 12V.





III.5. Conclusion

Dans ce chapitre, on a fait une présentation globale sur la machine de moulage du sable et ses architectures et ses organisations et nous avons aussi fait un graficet à partir d'un organigramme bien détaillé puis des descriptions sur les étapes de fonctionnement de cette machine.

Ensuite, grâce au logiciel (ISIS PROTEUS), nous avons fait une réalisation du schéma de machine avec deux méthodes qui sont basées sur (PIC16F877A) et (carte ARDIUNO). Aussi les différents circuits utilisés ont été présentés.

Nous avons montré comment se fait la programmation des composants de commande qui gère plusieurs Actionneurs (Moteurs, LED, LCD etc.) et à l'aide de ARES on a créé des typons. Enfin nous avons montré la photo de la réalisation de la carte électronique du système.

Conclusion générale

Grâce à ce projet, nous devons achever le développement d'une machine de moulage en sable qui fonctionnent automatiquement, et ARDUINO UNO est élément à base commande dans la machine. Afin de faciliter et d'accélérer l'avancement des travaux dans l'entreprise ALFET afin qu'elle se compose d'accomplir des pièces industrielles plus vite, mieux et permet l'utilisation d'une seule personne.

Ainsi ce projet est centré sur l'automatisation et simulation d'une carte de commande d'une machine de moulage en sable à base de microcontrôleur PIC16F877A et ARDUINO UNO, l'implantation d'un programme d'Algorithme écrit en C qui permet de contrôler le PIC et ARDUINO.

Ce projet nous a effectivement permis de proposer des schémas de commande électrique ainsi que d'avoir développé une simulation sur PROTEUS.

Les travaux présentés dans ce mémoire ont porté sur l'étude et la réalisation d'une commande d'un moteur à courant continu et d'élaborer un programme en langage C permettant d'atteindre un fonctionnement performant qui sera implémenté dans un ARDUINO.

Les résultats obtenus lors de nos tests nous montrent que l'ARDUINO UNO ou PIC 16F877A s'adapte bien à la commande du moteur à courant continu et Nous concluons également qu'il peut être contrôlé dans plusieurs moteurs monophasés ou triphasés.

En effet, ce projet nous a été une expérience très fructueuse bénéfique et enrichissante. Il nous a permis d'exploiter les différentes notions théoriques vues durant notre cycle et l'exploitation de différents logiciels comme PROTEUS, ARDUINO et PIC C compiler. Il nous a permis également de mettre le premier pas dans le monde pratique de l'industrie.

Ce travail nous a permis, d'une part, d'utiliser les connaissances acquises et aussi de se familiariser tant qu'avec les outils de simulation qu'avec l'aspect pratique. D'autre part, il nous a permis d'approfondir dans le domaine de la programmation.

- Le troisième chapitre est consacré aux difficultés rencontrées et expose quelques résultats expérimentaux obtenus validant le bon fonctionnement.

Enfin, nous souhaitons que notre travail constituera un document pédagogique pour les futures promotions du domaine de l'électronique et constituera par la suite une référence bibliographique très utile pour les futurs travaux.

Nous espérons de tout cœur que notre travail va être amélioré dans le futur par d'autres étudiants.

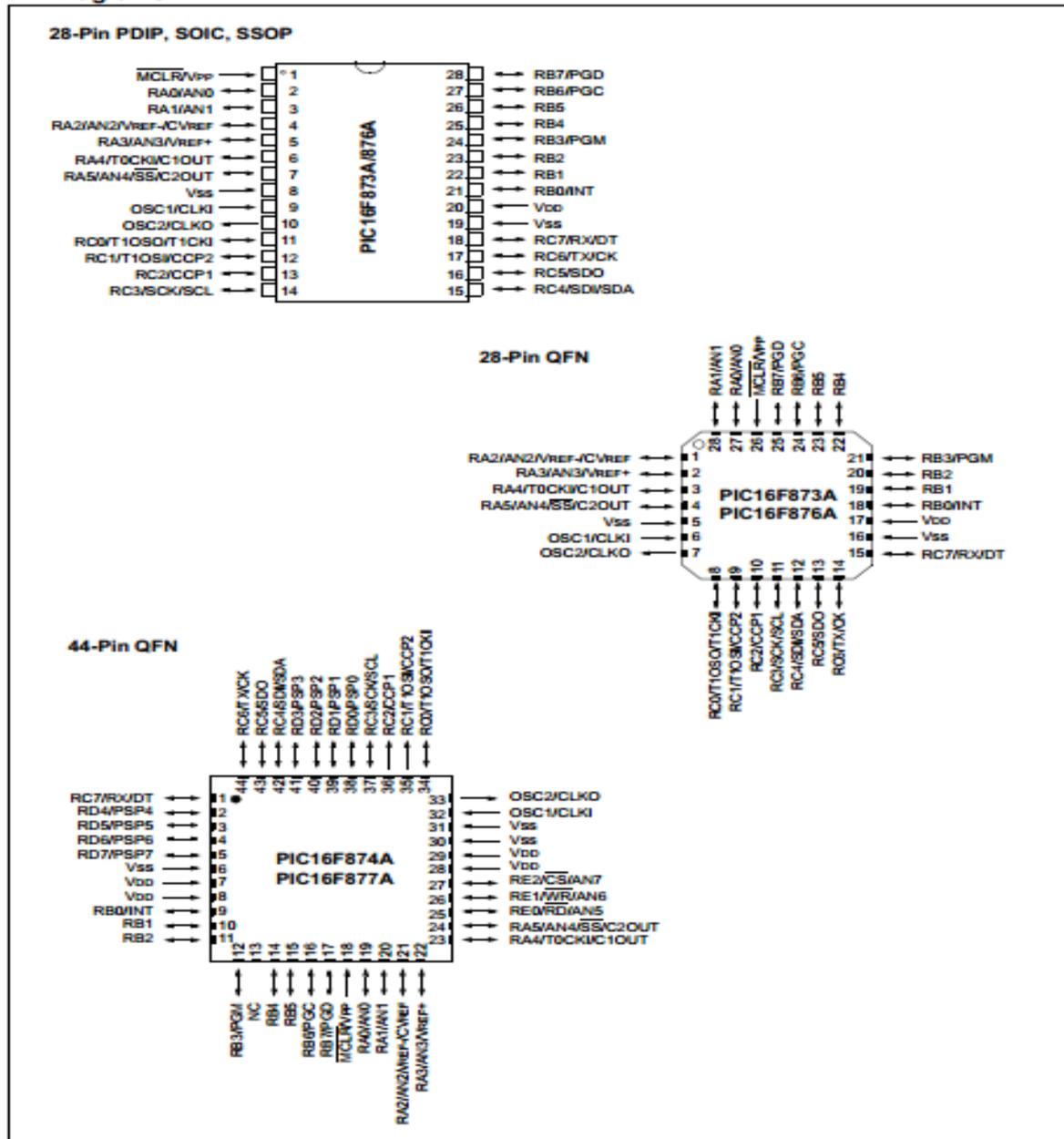
Références bibliographiques

- [1] Projet de fin d'étude Thème "Elaboration d'un système de contrôle et de commande d'une unité de recyclage de sable de moulage".
Présenté par : BENDJAMAA NABIL Et DAHMANI ABDELKADER.
Promotion 2013/2014.
*Cahier de charge de sablerie au niveau de la société ALFET « <http://alfet-dz.com/>».
- [2] Exposé construction : Le moulage
Présenté par : RIBLET Nicolas & AUGELMANN David .PPT(power point)
- [3] Rapport de stage (machine de moulage) Université ibn Khaldoun Tiaret 2009.
- [4] L.P.T.I. Saint Joseph La Joliverie (Le moulage des métaux et polymères) PDF.
- [5] BIGONOFF, La programmation des PICS, La gamme MID-RANGE par l'étude des 16F87X (16F876-16F877).
- [6] T.BEN HADDADA, M. HAMAM, R. MAHJOUB. «Un robot suiveur de ligne», Rapport de projet de fin d'étude, Université Tunis-El Manar, 2009.
- [7] A. AMRAOUI, Y. CHALA. « Réalisation D'un Robot mobile commandé à distance » Mémoire d'ingénieur d'état en électronique Biskra, Juin 2010.
- [8] Les Microcontrôleurs PIC: Description et mise en œuvre, Christian TAVERNIER, Nouvelle présentation de la 2ème édition, DUNOD, 2002.
- [9] Programmation En C Sur Carte "EASYPIC 2" du Microcontrôleur PIC16F877 , Karim SAAD, 2007/2008.
- [10] http://www.technologuepro.com/microcontroleur/Programmation_PIC_C_compiler.htm
- [11] LANGAGE C (CCS Info pour les PICs) Philippe LETENNEUR -GRANVILLE -2003 –PDF.
- [12] Web: xa.yimg.com/kq/groups/31819573/535819207/Name/moulage (PDF).

Annexe

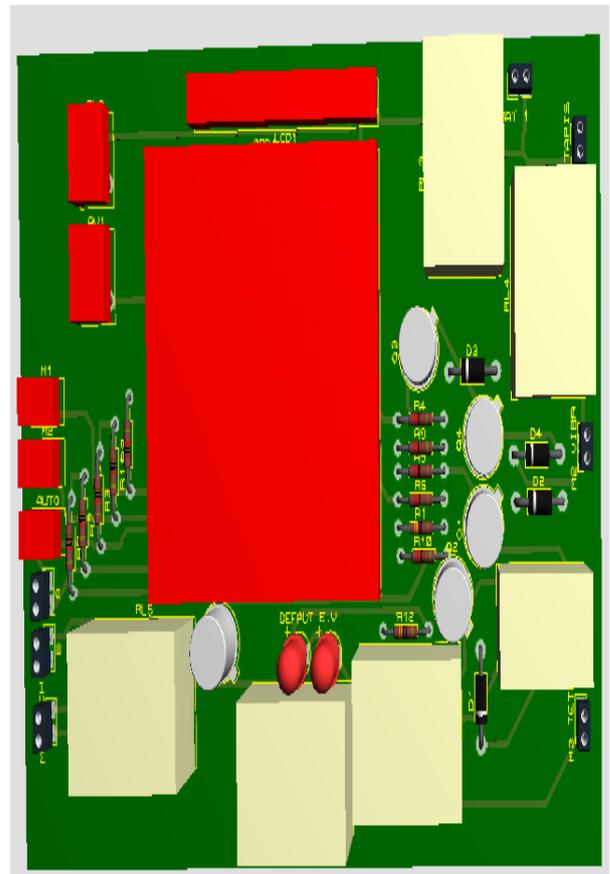
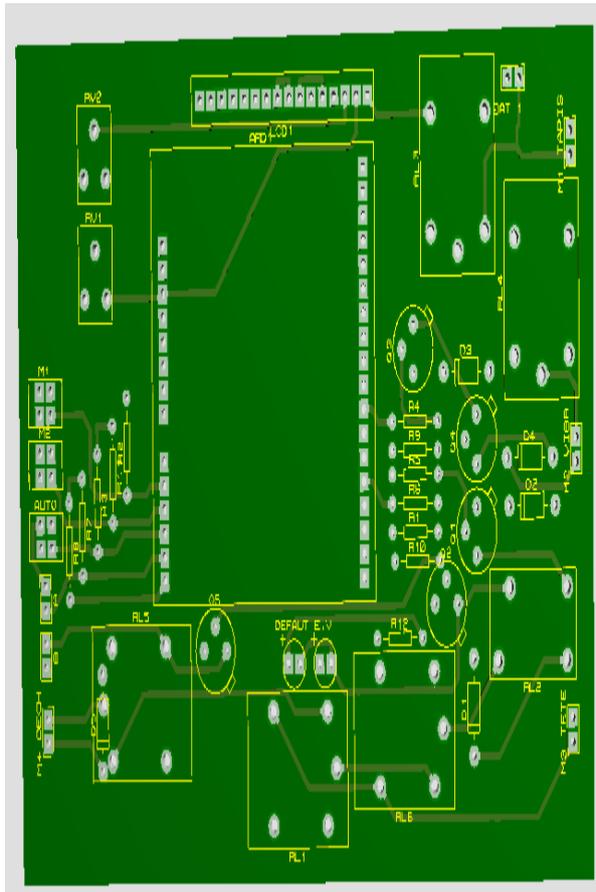
➤ Data Sheet PIC 16F877A :

Pin Diagrams

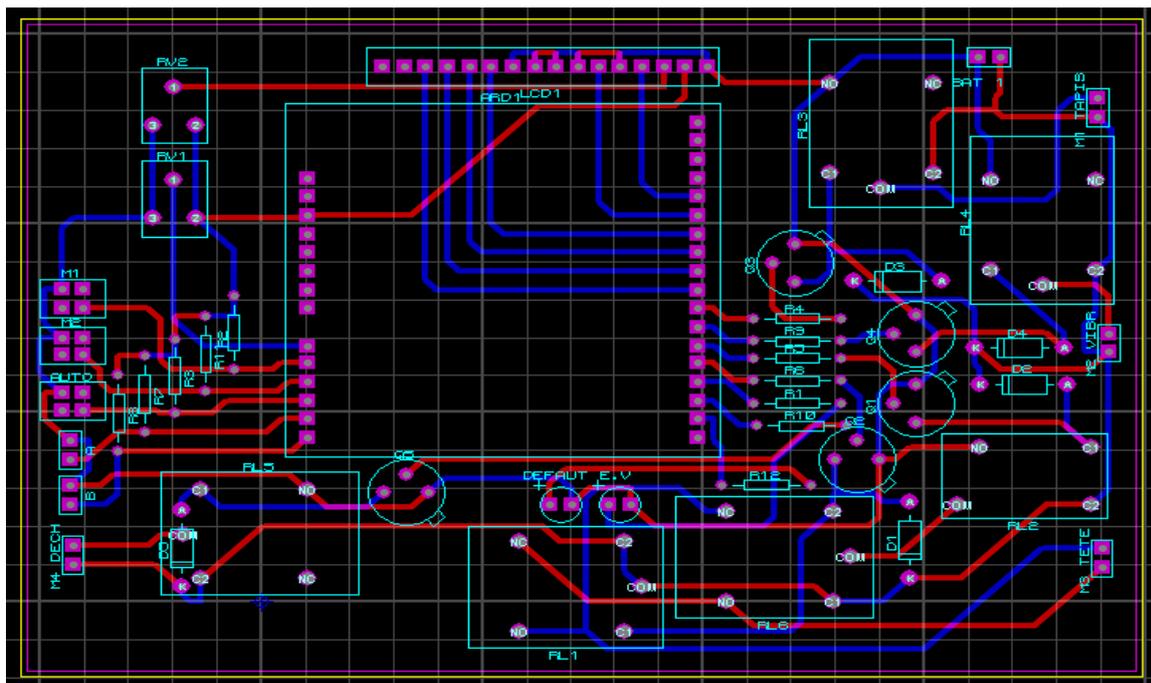


▪ Pour Schématisation de la carte sous ARES :

➤ La carte 3D dans Ares:



➤ Routage de la carte dans Ares:



✚ Fiche technique de transistor NPN 2N1711


2N1711

EPITAXIAL PLANAR NPN

DESCRIPTION

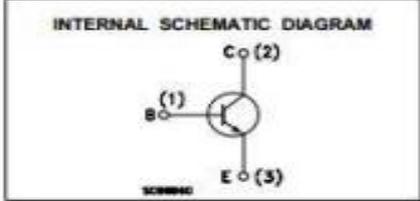
The 2N1711 is a silicon Planar Epitaxial NPN transistor in JEDEC TO-39 metal case. It is intended for use in high performance amplifier, oscillator and switching circuits.

The 2N1711 is also used to advantage in amplifiers where low noise is an important factor.



TO-39

INTERNAL SCHEMATIC DIAGRAM



SCHEMATIC

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CB0}	Collector-Base Voltage ($I_C = 0$)	75	V
V_{CE0}	Collector-Emitter Voltage ($R_{\theta JC} \leq 1002$)	50	V
V_{BE0}	Emitter-Base Voltage ($I_C = 0$)	7	V
I_C	Collector Current	500	mA
P_{tot}	Total Dissipation at $T_{amb} \leq 25^\circ\text{C}$	0.8	W
	at $T_C \leq 25^\circ\text{C}$	3	W
	at $T_C \leq 100^\circ\text{C}$	1.7	W
T_{stg}	Storage Temperature	-65 to 175	$^\circ\text{C}$
T_J	Max. Operating Junction Temperature	175	$^\circ\text{C}$

September 2002
1/4

Fiche technique de (transistor) 2N1711

Résumé

ملخص:

العمل المقدم في هذه المذكرة هو عبارة عن إنجاز ودراسة نظام السيطرة على آلة الرمال الصب من خلال أربعة محركات (حلقة مفرغة وذراع هزاز الملف اللولبي، يهز) و مدة ضغط صمام الملف اللولبي. حيث أن هذه الأجهزة وجدت في شركة ALFET تيارت استخداما في تصنيع قطع مختلفة.

يتكون هذا النظام بالأساس على عضو التحكم المتمثل في الأردوينو . بالإضافة إلى أن النظام مزود بتقنية الكريستال السائل LCD لرؤية المعلومات الأساسية لتعديل وتشغيل النظام (تحديد مدة ضغط صمام الملف اللولبي أو أخطاء تحدث أثناء العملية).

في إطار المساعدة المتبادلة بين الصناعة والجامعة، وجدنا أنه من المفيد لتطوير هذا المشروع من أجل إثبات أن الجامعة الجزائرية قادرة على مواجهة التحدي المتمثل في المساهمة في تنمية وتطوير البحث العلمي لجلب لإيجاد حلول حقيقية وملموسة في صناعة الإنتاج.

مساهمتنا في توفير نظام التحكم وحدة التحكم من خلال اعتماد التكنولوجيا حيث البطاقات التي تستند إلى المنطق القديمة سوف تفسح المجال لجيل جديد أردوينو. لقد قمنا بتطوير نظام الطريقة التلقائية وفقا لأردوينو إينو وتعمل برنامجه ونتائج التجارب التي أجريت في المحاكاة والممارسة.

Abstract:

The work presented in this thesis is dedicated to the implementation of a standard system and study for of the sand molding machine through four motors (treadmill and vibrator, solenoid arm, shake) and the pressure of time the solenoid valve. As these machines have found in the ALFET -Tiaret use in the manufacture of various parts.

The system in question is based on a ARDUINO (UNO) representing the control member, in addition it is equipped with a LCD display (Light Control Display) for display settings operation (Indication of the pressure valve duration or defects occurring during operation).

In the context of mutual assistance between industry and university, we found it useful to develop this project to demonstrate that the Algerian university is able to meet the challenge of contributing to the development and evolution of the scientific research to bring it to provide real and tangible solutions in the field of industrial production.

Our contribution is to propose a system of command and control unit by adopting a technology where the old maps based logic circuits will be replaced with the new generation of Arduino. We developed the system to automated basis Arduino Uno and operated its program and the results of tests conducted under proteus simulation and realization.

Résumé:

Le travail présenté dans ce mémoire est dédié à la réalisation et l'étude d'un système de commande de la machine de moulage de sable à travers quatre moteurs (tapis roulant et vibreur, bras électrovanne, décochage) et la durée de pression de l'électrovanne. On trouve ces machines au sein de l'entreprise ALFET-Tiaret pour être utilisées dans la fabrication des pièces mécaniques. Le système d'automatisation des ces machines est à base d'une carte ARDUINO UNO représentant l'organe de commande. En plus, il est équipé d'un afficheur LCD pour la visualisation de la consigne temporelle représentant la pression exercée sur le sable de moulage par le bras de robot, ainsi que les défauts survenant lors de son fonctionnement).

Dans le cadre de l'entraide entre l'industrie et l'université, nous avons jugé utile de développer ce projet afin de démontrer que l'université Algérienne est capable de relever le défi de la contribution au développement et à l'évolution de la recherche scientifique pour l'amener à apporter des solutions réelles et palpables dans le domaine de l'industrie de production.

Notre contribution consiste à proposer un système de commande et de contrôle de l'unité en adoptant une technologie où les anciennes cartes à base de circuits logiques feront place aux cartes micro-programmées de la nouvelle génération.

Nous avons développé le système de manière automatisée à base d'une carte Arduino Uno et exploité son programme ainsi que les résultats des tests réalisés en simulation et en pratique.