

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique

Option : Electronique Communication et Signaux

Réf:.....

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

MASTER

Thème

**Réalisation d'une horloge numérique pour
l'indication automatique des temps de la prière
islamique à base de PIC16F877**

Présenté par :

ZEMALI Mounir

Soutenu le : 10 Juin 2012

Devant le jury composé de :

Mr.BELGACEM Hocine

M.C.B

Président

Mr.ALLAOUI Abdelhamid

M.A.B

Examineur

Mr.HEZABRA Adel

M.A.B

Encadreur

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique

Option : Electronique Communication et Signaux

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

MASTER

Thème

**Réalisation d'une horloge numérique pour
l'indication automatique des temps de la prière
islamique à base de PIC16F877**

Présenté par :

Avis favorable de l'encadreur :

-ZEMALI Mounir

Signature

Avis favorable du Président du Jury

Nom Prénom

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique

Option : Electronique Communication et Signaux

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

MASTER

Thème

**Réalisation d'une horloge numérique pour
l'indication automatique des temps de la prière
islamique à base de PIC16F877**

Proposé par : Mr. HEZABRA Adel

Dirigé par : Mr. HEZABRA Adel

Remerciement

Tout d'abord, Nous tiens à remercier DIEU le miséricordieux de nos avoir donné la possibilité de réaliser notre projet, d'arriver à notre souhaits et d'atteindre notre objectifs.

Nous aimerons dans ces quelques lignes remercier toutes les personnes qui d'une manière ou d'une autre, ont contribué au bon déroulement de notre travail, tout au niveau humain qu'au niveau scientifique.

Nous tenons tout d'abord à remercier notre encadreur MONSIEUR, Hezabra ADEL on a pu bénéficier à la fois de ses compétences scientifiques, et de sa grande disponibilité, tant pour résoudre les difficultés rencontrées lors de notre réalisation, de répondre à nos questions.

Nous ajoutons en particulière sa patience et ses encouragements, nous a permis de travailler dans bonnes conditions.

Nous exprimons notre gratitude à l'ensemble des professeurs du département d'électronique qui ont contribué à notre formation, on les prie de bien vouloir croire à notre gratitude en espérant que cet humble travail de trois années fera crédibilité de leurs efforts.

Grand remerciement à tous les ingénieurs des laboratoires de génie électrique : Mr. HAMZA, Mr. ABDELATIF, Mr AHMED.

Nos remerciements s'adressent également à tous membres de Jury, qui ont accepté de nous honorer de leur présence et de juger notre travail Merci. Et à toute personne ayant contribué de près ou de loin à notre soutien moral.

Dedicace

Je dédie ce travail à mes très chers parents qui ont tant prié à mon réussite,

De soutien moral. Je les remercie de m'avoir encouragé

Et aidés à devenir ce que Je suis.

Et A, mes frères et mes sœurs,

et à toute la famille « zemali ».

et à toute la promotion

Master De l'année 2012

A mes très chers amis.

Liste des figures

CHAPITRE I :

Fig. I.1. : Mouvement de la terre autour du soleil.	2
Fig. I.2 : Coordonnées terrestres.	3

CHAPITRE II :

Fig. II-1 : Les éléments constitutifs du PIC 16F877	15
Fig. II-2 : Brochage de Pic 16F877.	16
Fig. II-3 : le registre de configuration STATUS	17
Fig. II -4: organisation de la RAM du 16F876/877.....	18
Fig. II-5 : organisation de la RAM du 16F876/877.....	19
Fig. II-6 : résistance de pull-up.....	23
Fig. II-7 : LED sur RA4.....	24
Fig. II-8 : Oscillateur à quartz du PIC 16F87x.....	30
Fig. II -9 : Circuit de Reset	31

CHAPITRE III :

Fig. III.1 : Fenêtre principale de mikroC PRO for PIC.....	38
Fig.III.2.Création d'un nouveau projet.....	39
Fig.III.3.Choix de PIC.....	39
Fig.III.4.Choix de fréquence.	40
Fig.III.5.Nommer le projet fréquence	40
Fig.III.6.Fenêtre d'ajoute des fichiers supplémentaires.....	40
Fig.III.7.La sélectionne de la bibliothèque	41

Fig. III.8.La fin de la création d'un projet.....	41
Fig. III.9.fenêtre de compilation.....	41
Fig. III.10. Fenêtre principale d' Universal Programmer.....	42
Fig. III.11. La fenêtre de coix du PIC.....	42
Fig. III.12. La fenêtre de chargement de fichier hexa.....	43
Fig. III.13. La fenêtre du transfert de fichier hexa.....	43
Fig. III.13. L'Organigramme de programme global	44

CHAPITRE IV :

Fig. IV. 1. la fenêtre d'interface utilisateur.....	49
Fig. IV. 2. l'organisation de la boîte d'outils(1).....	51
Fig. IV. 3. l'organisation de la boîte d'outils(2).....	51
Fig. IV. 4. l'organisation de la boîte d'outils(3).....	52
Fig. IV. 5. Simulation par ISIS Professionnel.....	52
Fig. (IV.8) Affichage à sept segments de type LED.....	53
Fig. IV-10 :Brochage de DS1307.....	54
Fig. IV-11 : Le branchage de PIC 16F877 avec DS1307.	55
Fig. IV-12 : Liaison I2C -1-.....	55
Fig. IV-13 : Liaison I2C -2-.....	56
Fig. IV-14 : Liaisons I2C a plusieurs circuits.....	56
Fig. IV-15 : Le protocole I2C	56
Fig. IV-16 : le chronogramme des lignes SDA et SCL.....	57
Fig. IV-17 : Les registres d'adresses 0x00 à 0x06 du DS1307.....	59
Fig. IV-18 : Chronogramme du Bus I2C.....	60
Fig. IV-18.l'LCD 2x16.....	61
Fig. IV-18.Fenêtre principale de ARES.....	63

Liste des tableaux

CHAPITRE I :

Tableau(I.1) Temps du Choix et Temps de la nécessité de prière.....	6
---	---

CHAPITRE II :

Tableau II-1 : caractéristiques de la famille 16F876 – 16F877.....	14
--	----

Tableau II-2 : OPTION REGISTER	22
--------------------------------------	----

Tableau II-3 : les instructions orienté octets.....	26
---	----

Tableau II-4 : Les instructions orienté bit.....	28
--	----

Tableau II-5 : Les instructions de contrôle.....	28
--	----

Tableau II-6 :Les interruptions primaires et les interruptions périphériques	32
--	----

Tableau II-7 :Les Sources d'interruption	32
--	----

Tableau II-8 : Select Mode.....	34
---------------------------------	----

CHAPITRE IV

Tableau IV.1: Parallèle LCD protocole.....	62
--	----

Tableau IV.2: broche Fonction de bit de commande et de la valeur.....	62
---	----

Sommaire

Introduction générale.....	01
Chapitre I : Les temps de la prière	
I.1.INTRODUCTION	02
I.2. Le mouvement de la terre autour du soleil	02
I.3. Direction du rayonnement solaire	03
I.3. 1. Coordonnées terrestres	03
I.3. 1. a. La longitude	03
I.3. 1. b. La latitude	03
I.3. 1. c. L'altitude	03
I.4. Mesure du temps	04
I.4.1. Les heures et les jours	04
I.4.2. Les temps de la prière	05
I.4.3. a. Le temps de la nécessité	05
I.4.3.b. Le temps du choix	06
I.5. Les Equations de la prière	07
I.6. Exemple d'application	11
I.7.Conclusion.....	13
Chapitre II: Présentation du PIC16F877	
II-1. Introduction	14
II.2. BROCHAGE 16F877	16
II.3. l'organisation mémoire du PIC 16F877	17
II.3. 2 La mémoire EEPROM	17
II.3. 3 La mémoire RAM:	17
II.4. Les registres internes	20
II.4.a. Le registre de STATUS	20
II.4.b. OPTION REGISTER	21
II.4.c. INTCON REGISTER	22
II.5. Les ports d'Entrée /Sortie	23
II.5. a. Le port d' E/S PORTA	23
II.5. b. Les autres broches de PORTA	24
II.5. c. Le port d' E/S PORTB	24

II.5. d. Le port d' E/S PORTC	25
Le port d' E/S PORTE	25
II.6. Le jeu d'instructions	26
II.6. a. les instructions orienté octets	26
II.6. b. Les instructions orienté bit	28
II.6. c. Les instructions de contrôle	28
II.7. Configuration des PORTx (les registres PORTx et TRISx).....	29
II.8. Horloge du système	30
II.9. Circuit de Reset	30
II.10. Les interruptions	31
II.10.a. Les sources d'interruption	32
II.11. Module MSSP pour I2C	32
II.11.1. SSPSTAT REGISTER	33
II.11.2. SSPCON REGISTER.....	34
II.11.3. SSPCON2.....	34
II.11.4. MODE MAITRE	35
II.11.4.a. Lancement d'un START	36
II.11.4.b. Transmission.....	36
II.11.4.c. Acquiescement.....	36
II.11.4.d. Lancement d'un STOP.....	36
II.11.5. Réponse du maître à l'esclave.....	36
II.11.5.a. Réception d'un octet par le maître.....	37
II.12. CONCLUSION.....	37

Chapitre III: Réalisation software

III.1. Introduction.....	38
III.2.1. Présentation de logiciel (Mickro C pro for pic).....	38
III.3. Les étapes suivies pour crée un projet en mikroC PRO for PIC.....	39
III.4. L'Universal Programmer(ALL11P3).....	42
III.5. L'Organigramme de programme global.....	44
III.6. Explication de l'organigramme globale.....	45
III.7. Organigramme de Real time clock	46
III.8. conclusion	48

Chapitre IV: Réalisation hardware

IV.1 Introduction	49
IV. 2. Réalisation avec logiciel proteus	49
IV. 2.1.Présentation du logiciel	49
IV. 2.2. Fenêtre d'ensemble	50
IV. 2.3. Fenêtre d'édition	50
IV. 2.4. La boite a outils	50
IV. 2.5. Cordonnées du curseur	50
IV. 2.6. Raccourcis clavier DES FONCTIONS D'EDITION COURANTES	50
IV. 2.7. L'organisation de la boite d'outils	51
IV. 3. Schéma électronique de circuit	52
IV. 3.1. Simulation par "proteus 7"	52
IV. 3.2. Réalisation par matériel	53
IV. 3.2.a. Une carte d'alimentation	53
IV. 3.2.b. Carte de commande	53
IV. 3.2.c. Carte de circuit RTC	53
IV. 3.2.d. Circuit d'affichage LCD	53
IV.4. définition de DS1307	53
IV.5. Le branchage de PIC 16F877 avec DS1307	54
IV.6. Introduction au bus I2C	54
IV.6.1. Exemple de communication entre un maître (PIC 16F877) et un esclave	56
IV.6.2. Exemple d'écriture du DS1307	56
IV.6.3 . Exemple de lecture du DS1307	59
IV. 7.Définition de LCD	61
IV. 8.Présentation de ARES	63
IV. 8.1.Création d'une carte	63
IV. 8.2.Création d'un boîtier	64
IV. 8.3.Créer un composant	64
IV. 8.4.Le placement des composants	64
IV. 8.4.a. Placement automatique	64
IV. 8.4.a.Placement Manuel	64
IV. 8.5.Le routage	64
IV. 8.5.a.Routage automatique	64

IV. 8.5.b. Routage Manuel	65
IV. 9.Conclusion	65
Conclusion générale	66
Bibliographié.....	67
Annexe.....	68

En islam tous les musulmans doivent réaliser cinq prières par jour, nommées comme suit :Fadj, Zuhr, Asr, Maghrib et Isha, tous a des horaires différents.et , « **Allah** » l'a recommandée en maints versets du Coran.

" وأقيموا الصلاة إن الصلاة كانت على المؤمنين كتابا موقوتا "

En effet, l'horaire de ces cinq prières obligatoires diffère d'un endroit a un autre, et diffère d'un jour à un autre. Ainsi, une prière peut se décaler de plusieurs heures, sur plusieurs mois.

Et d'après Abou Houreira (qu'Allah l'agrée), le Prophète (bénédition et salut soient sur lui) a dit: « Voyez-vous si l'un de vous avait devant sa porte un fleuve et qu'il s'y baigne 5 fois par jour, resterait-il une quelconque saleté? ». Ils ont dit: Il ne va rester aucune saleté. Le Prophète (bénédition et salut soient sur lui) a dit alors: « Ceci est l'exemple des 5 prières par lesquelles Allah efface les péchés ». (Rapporté par Boukhari n°528 et Mouslim n°667).

Chaque prière doit être faite en son temps prescrit, celui qui laissera passer le temps d'une prière (sans l'avoir fait) et sans aucune raison ou excuse, aura commis un grand péché. Il aura toujours obligation bien sûr de rattraper les prières manquées dans tous les cas (et sans attendre).Sauf le cas de la femme qui a ses menstrues(Hayd). Alors on s'est proposé la réalisation d'une horloge multifonctions qui outre l'indication de l'heure et la date du jour, elle permettra d'indiquer le moment précis de chaque prière, cette horloge est basée sur le microcontrôleur **PIC 16F877**.

L'analyse théorique fait dans le premier chapitre explique la bonne méthode de calculer les temps des prières islamique.

La description de l'architecteur de microcontrôleur utilisé est développée dans le deuxième chapitre, aussi que le protocole « I2C » qui permet de communiquer entre le PIC 16F877 et le DS1307.

Et nous avons réservée le 3eme chapitre pour la réalisation software pour programmer le microcontrôleur, et le dernier chapitre concernant de la partie pratique ou la réalisation pratique du circuit électronique.

Notre projet se termine par la conclusion générale la quelle on a citer les problèmes liés à Cette réalisation software et hardware.

Chapitre 1

Les temps

De la prière

I.1.INTRODUCTION :

Allah le Très Haut a prescrit à Ses serviteurs cinq prières au cours du jour et de la nuit. De par Sa sagesse, Il a fixé à ces prières des heures de sorte à permettre au fidèle d'être en contact permanent avec Allah le Très Haut à travers les prières effectuées durant ces heures. Celles-ci sont pour le cœur ce que l'eau est pour la plante ; elles l'irriguent au goutte à goutte et ne le font pas d'un seul coup pour s'arrêter ensuite. La répartition de ces prières à ces heures repose sur une sagesse qui vise à éviter au fidèle l'ennui et la lourdeur qui résulteraient de leur accomplissement en même temps. Allah le Très Haut est la source de la bénédiction et le plus sage des sages . [1]

I.2. Le mouvement de la terre autour du soleil : [2]

La terre décrit autour du soleil une orbite elliptique quasi circulaire avec une période 365, 25 jours. Le plan de cette orbite est appelé plan de l'écliptique. C'est au solstice d'hiver (21décembre) que la terre est la plus proche du soleil 147 millions de Km et au solstice d'été (22juin) qu'elle en est la plus éloignée 152 millions de Km la moyenne étant de 149, 5 millions de Km. La terre coupe le plan perpendiculaire au grand axe de son orbite et contenant le soleil aux équinoxes de printemps (21mars) et d'automne (21septembre). Elle tourne sur elle-même avec une période de 24h. Son axe de rotation (l'axe des pôles) a une orientation fixe dans l'espace.

Il fait un angle $\delta = 23^{\circ}27'$ (ou $23,45^{\circ}$ en fraction décimale du degré) avec la normale au plan de l'écliptique.

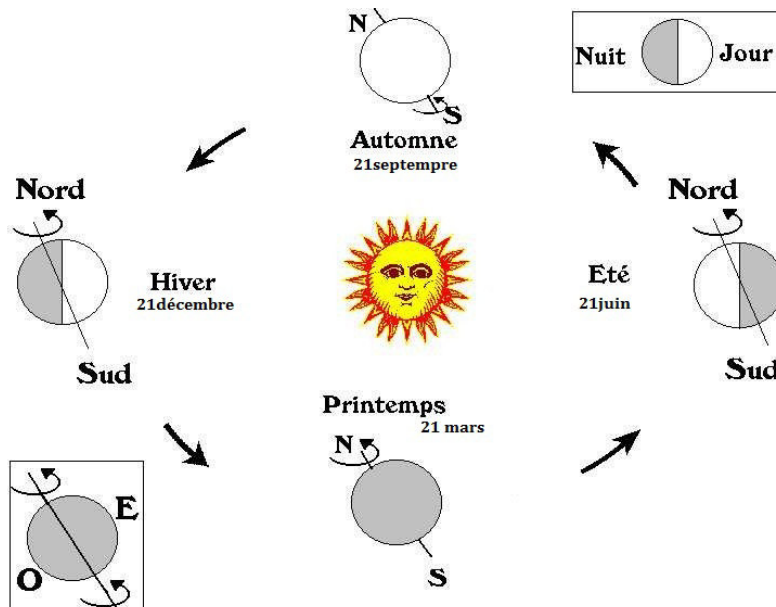


Fig. I.1. : Mouvement de la terre autour du soleil. [2]

I.3. Direction du rayonnement solaire : [2]

I.3. 1. Coordonnées terrestres :

Tout point de la surface terrestre est repéré par ses coordonnées géographiques; longitude et latitude, ainsi que par son altitude. Fig. (I.1)

I.3. 1. a. La longitude :

La longitude d'un lieu correspond à l'angle que fait le plan méridien passant par ce lieu avec un plan méridien retenu comme origine. On a choisi pour méridien origine 0° le plan passant par l'observatoire de Greenwich. Par convention, on affecte le signe + les méridiens situés à l'Est de ce méridien, et de signe - les méridiens situés à l'Ouest. La longitude sera désignée ici par la lettre λ elle peut être comprise entre -180° et $+180^\circ$.

I.3. 1. b. La latitude :

La latitude d'un lieu correspond à l'angle, avec le plan équatorial, que fait le rayon joignant le centre de la terre à ce lieu.

L'équateur terrestre est donc caractérisé par une latitude égale à 0° , le pôle nord par la latitude $+90^\circ$ et le pôle sud par la latitude -90° , cette convention de signe + à tous les lieux de l'hémisphère nord et le signe - à tous les lieux de l'hémisphère sud. La latitude sera désignée ici par la lettre φ . Elle peut ainsi être comprise entre -90° et $+90^\circ$.

I.3. 1. c. L'altitude :

L'altitude d'un point correspond à la distance verticale entre ce point et une surface de référence théorique, géoïde, figurant le niveau moyen de la mer. On l'exprime généralement en mètre.

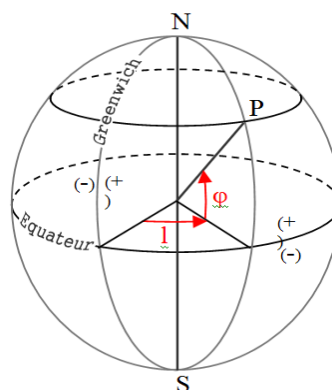


Fig. I.2 : Coordonnées terrestres. [3]

I.4. Mesure du temps : [2]

I.4.1. Les heures et les jours :

Les mouvements de la terre et sa forme influencent directement notre perception du temps. La succession des jours et des nuits du fait de la rotation de la terre, permet de définir le temps solaire vrai :

$$T_{s,vrai} = \omega / 15 + 12h . \text{ avec } \omega [\text{degré}]. \quad (\text{eq. I-1})$$

Où ω est l'angle horaire du Soleil. Les 12 h de différence entre les deux temps reflètent le fait qu'il est midi lorsque le Soleil est au méridien ($\omega = 0$). Du fait de l'orbite elliptique de la terre autour du soleil et à cause de l'inclinaison de l'écliptique le temps solaire vrai change au long de l'année. On définit donc un temps solaire moyen basé sur un soleil fictif qui se déplacerait uniformément le long de l'équateur durant le même temps requis par le vrai soleil pour se mouvoir le long de l'écliptique. C'est l'angle horaire de ce soleil fictif qui définit le temps solaire moyen de telle sorte que :

$$T_{s, vrai} - T_{s, moy} = ET. \quad (\text{eq. I-2})$$

La valeur de ET au cours du temps est appelée l'équation du temps. Les valeurs extrêmes de ΔT sont -14,3 minutes le 12 février, +3,7 minutes le 14 mai, -6,4 minutes le 26 juillet et +16,4 minutes le 4 novembre. Le temps solaire moyen dépend du méridien où l'on se trouve.

Afin de simplifier les transports et communications dans notre société, 24 zones de temps ont été constituées. Certaines zones sont prises pour références. Par exemple le méridien de Greenwich donne le Greenwich Mean Time (GMT). L'Europe centrale est dans le système du Central Mean Time (CET).

Pour des raisons scientifiques, il est parfois indispensable de coordonner les observations obtenues dans des endroits différents en utilisant le même temps.

On définit donc le temps universel, comme le temps solaire moyen au méridien de Greenwich. On le note TU. Il vaut 0 heures à minuit. Par exemple 12.00 h TU correspond à 12.00 h GMT et à 13.00 h CET.

Pour un lieu de longitude λ , on a :

$$T_{s, moy} = TU + \lambda / 15 \quad (\lambda \text{ en degré, } + \text{ vers l'est et } - \text{ vers l'ouest}) \quad (\text{eq. I-3})$$

Au cours de l'année le Soleil moyen se déplace par rapport au point vernal à raison de 360° (ou encore 24 h) tous les 365 jours. Le jour solaire moyen est donc 24 h/360 j plus long que le jour sidéral, soit 3 minutes 56 secondes, comme déjà vu plus haut.

I.4.2. Les temps de la prière: [4]

1. Tout d'abord, le **Fajr** débute du crépuscule du matin, ou plutôt à l'aube, le moment de l'apparition du soleil et ses rayons lumineux. El **Fajr** se détermine, cependant, juste avant le moment qui coïncide avec le lever du soleil.

2. Ensuite, vient el **Zuhr** (ou El Douhr) qui débute au commencement de la seconde partie de la journée, l'après-midi lorsque la branche mobile du soleil a dépassé le méridien de quelques centimètres. Pour plus de commodités et de précisions, dans de nombreux horaires de prière, cinq minutes sont ajoutées à la mi-journée (zawal) pour obtenir le début de **Zuhr**. **Zuhr** se termine au début d'**Asr**. Ainsi, il vaut mieux faire cette prière avant l'annonce de la nouvelle qui suit quelques heures après.

3. Vient ensuite la prière d'**Asr**, qui selon la grande école de Hanafi, commence ou débute lorsque la longueur de l'ombre est égale à 2 fois la longueur de l'objet plus l'ombre à midi. Ainsi, sa détermination relève du domaine du difficile. De plus, selon un communiqué officiel de l'école de Hanafi, l'horaire de la prière d'**Asr** est déterminé par la longueur de l'ombre qui devient égale à la longueur de l'objet plus l'ombre à midi. La prière d'**Asr** s'achève avant le coucher du soleil, tout juste avant la prière qui suit.

4. El **Maghrib**, commence en effet, au moment du coucher du soleil et s'achève au début de l'isha, la prière qui suit tout juste après.

5. El **Isha** commence après la tombée complète de la nuit lorsque le crépuscule du soir disparaît laissant lieu à un ciel noir ou bleu, et avec nuls rayons de soleil.

Les temps pour l'accomplissement de la prière sont divisés en deux catégories :

I.4.3. a. Le temps de la nécessité :

C'est l'intervalle de temps qui se trouve généralement à la fin du temps d'une prière particulière avant que n'arrive la prochaine prière. On l'appelle 'temps de l'urgence' ou 'de la nécessité', puisque la personne qui prie pendant ce temps le fait en raison d'une nécessité ou d'une excuse valable. Ceci ne signifie pas que l'on soit exempté de l'obligation d'accomplir sa prière si on la retarde jusqu'au temps de la nécessité. Plutôt, cela signifie que si l'on n'a pas d'excuse valable pour la retarder, alors on est considéré comme coupable d'un péché à cause

de ce retard. Les excuses valables sont les choses telles que dormir, s'évanouir, oublier la prière, atteindre la puberté pendant le temps de la prière, être frappé de folie, accepter l'islam ou apostasier et se repentir avant que le temps de la prière ne se soit écoulé, ou que l'écoulement du sang des menstrues ou des lochies cesse avant que le temps de la prière ne se soit écoulé quand il reste suffisamment de temps à la femme pour accomplir le bain rituel (ghusl) et prier au moins une rakat avant que le temps ne s'écoule. Le tableau ci-dessous indique les cinq prières et leurs temps respectifs de choix et de nécessité.

I.4.3.b. Le temps du choix :

c'est l'intervalle de temps pendant lequel il nous est permis d'accomplir la prière sans encourir le moindre péché si l'on retardait la prière jusqu'à la fin de cet intervalle particulier. C'est comme le temps imparti pour prier la prière du midi (Dhohr) qui s'étend jusqu'au temps de la prière de l'après-midi ('Asr). Tout le temps entre ces deux prières est considéré comme étant le temps du choix, puisque l'on a le choix d'accomplir la prière du début du premier temps de la prière jusqu'au début du temps suivant de la prière sans qu'il y ait le moindre péché pour l'avoir repoussée.

Tableau(I.1) Temps du Choix et Temps de la nécessité de prière.

Nom de la Prière	Temps du Choix	Temps de la nécessité
1. Subh	Commence: au moment de l'aube véritable quand le fil blanc du jour se distingue de la noirceur de la nuit. Finit: immédiatement avant que le soleil perce l'horizon.	Commence: immédiatement avant que le soleil perce l'horizon (le temps nécessaire à pouvoir prier une rakat de la prière). Finit: quand le soleil perce l'horizon.
2. Dhohr	Commence: dès que le soleil commence à décliner de son zenith quand l'ombre d'une chose commence à s'agrandir. Finit: quand l'ombre d'une chose devient égale à la taille de cette chose.	Commence: quand l'ombre d'une chose devient égale à la taille de cette chose. Finit: avant le coucher du soleil, le temps nécessaire à pouvoir prier 5 rakats.

3. Asr	<p>Commence: quand l'ombre d'une chose est égale à la taille de cette chose.</p> <p>Finit: une fois que l'ombre d'une chose devient deux fois plus grande que cette chose ou quand le soleil devient jaune.</p>	<p>Commence: une fois que l'ombre d'une chose devient deux fois plus grande que cette chose ou quand le soleil devient jaune.</p> <p>Finit: avant le coucher du soleil, le temps nécessaire à pouvoir prier 1 rakat. [Note: le temps de nécessité partagé par Zuhr et 'Asr finit avant le coucher du soleil de manière à pouvoir les 4 rakats spécifiques à 'Asr].</p>
4. Maghrib	<p>Commence: immédiatement après le coucher du soleil.</p> <p>Finit: immédiatement après que la prière soit accomplie selon Malik et ash-Shafi'i. Selon l'opinion de Abu Hanifa - la plus retenue - son temps s'étend jusqu'au temps de 'isha. Donc, son temps de nécessité commencerait immédiatement avant la disparition de la rougeur du crépuscule et finirait une fois que le crépuscule se serait totalement dissipé.</p>	<p>Commence: après l'accomplissement de la prière sans délai.</p> <p>Finit: avant le lever du jour (Subh) de façon à ce qu'il soit possible de prier 5 rakats.</p> <p>[Note: le temps de nécessité partagé par Maghrib et 'Isha finit avant l'aube véritable de manière à pouvoir prier les 4 rakats spécifiques à 'Isha].</p>
5. 'Isha	<p>Commence: une fois que le crépuscule s'est dissipé.</p> <p>Finit: une fois que le second tiers de la nuit a commencé.</p>	<p>Commence: une fois que le second tiers de la nuit a commencé.</p> <p>Finit: une fois que l'aube véritable apparaît au moment de la prière du matin.</p>

I.5. Les Equations de la prière:[5]

Les musulmans astronomes ont développé des équations pour déterminer les temps de prière.

Equation 1:

Le jour Julien est donnée par:

$$Dj = ((367 * Y) - (INT ((7/4)*(Y+INT ((M+9)/12)))) + INT (275 * (M/9)) + d - 730531.5. \text{ (eq. I-4)}$$

(d : le jour M : Le mois Y : l'année).

Dj : le jour Julien, INT : intégrer ; exemple : INT(11.565)=11.

Equation 2:

La longueur du soleil moyenne est donnée par :

$$L=280.461+0.9856474*Dj. \quad (\text{eq. I-5})$$

Si $L>360 \rightarrow L$ prend le reste de la division $L/360$.

Equation 3:

La part du soleil moyenne est donnée par :

$$M=357.528+0.9856003*Dj. \quad (\text{eq. I-6})$$

Si $M>360 \rightarrow M$ prend le reste de la division $M/360$.

Equation 4:

La longueur du soleil Albroujy (Λ) est donnée par :

$$\Lambda=L+1.915*\sin(M)+0.02*\sin(2*M). \quad (\text{eq. I-7})$$

Equation 5:

L'obliquité est donnée par :

$$\text{Obliquité}=23.439-0.0000004*Dj. \quad (\text{eq. I-8})$$

Equation 6:

L'Apparition solaire est donnée par :

$$\text{Alpha}=\text{atan}(\cos(\text{obliquité})*\tan(\Lambda)). \quad (\text{eq. I-9})$$

Si $90 < \Lambda < 180 \rightarrow \text{Alpha} = \text{Alpha} + 180$.

Si $180 < \Lambda < 360 \rightarrow \text{Alpha} = \text{Alpha} + 360$

Equation 7:

Temps interstellaire est donnée par :

$$ST=100.46+0.985647352*Dj. \quad (\text{eq. I-10})$$

Equation 8:

La pente angulaire de soleil est donnée par :

$$\text{Dec} = \sin(\text{obliquité}) * \sin(\text{Lambda}). \quad (\text{eq. I-11})$$

Equation 9:

La disparition du soleil moyenne est donnée par :

$$\text{Noon} = \text{Alpha} - \text{ST}. \quad (\text{eq. I-12})$$

si ($\text{noon} < 0$) $\rightarrow \text{noon} = \text{noon} + 360$.

Equation 10:

Le méridien haut est donnée par :

$$\text{Utnoon} = \text{noon} - \text{langitude}. \quad (\text{eq. I-13})$$

Le méridien local est donnée par :

$$\text{Local Noon} = (\text{Utnoon}/15) + \text{zone}. \quad (\text{eq. I-14})$$

Zone : le fuseau horaire

Le temps de la prière El Dohr = Local Noon

L'équation de prière El Asr :

$$\text{Asr Alt} = \text{atan}(1 + \tan(\text{lat} - \text{dec})). \quad (\text{eq. I-15})$$

$$\text{Asr Arc} = \text{acos}((\sin(90 - \text{Asr Alt}) - \sin(\text{Dec}) * \sin(\text{Lat})) / (\cos(\text{Dec}) * \cos(\text{Lat}))). \quad (\text{eq. I-16})$$

Le résultat Asr Arc on degree ,on devise cette résultat sur 15 pour déterminer temps

$$\text{Asr Arc} = \text{Asr Arc} / 15. \quad (\text{eq. I-16})$$

$$\text{Asr time} = \text{Local Noon} + \text{Asr Arc}. \quad (\text{eq. I-17})$$

Equation 11:

$$\text{Durinal Arc} = \arccos \left(\frac{\sin(-0.8333) - \sin(\text{Dec}) \cdot \sin(\text{Lat})}{\cos(\text{Dec}) \cdot \cos(\text{Lat})} \right). \quad (\text{eq. I-18})$$

$$\text{ELshourog} = \text{Sun rise} = \text{Local Noon} - (\text{Durinal Arc}/15). \quad (\text{eq. I-19})$$

$$\text{ELghourobe} = \text{Sun Set} = \text{Local Noon} + (\text{Durinal Arc}/15). \quad (\text{eq. I-20})$$

L'équation de prière El Icha :

$$\text{Esha Arc} = \arccos \left(\frac{\sin(-18) - \sin(\text{Dec}) \cdot \sin(\text{Lat})}{\cos(\text{Dec}) \cdot \cos(\text{Lat})} \right). \quad (\text{eq. I-21})$$

$$\text{Esha time} = \text{Local Noon} + (\text{Esha Arc}/15). \quad (\text{eq. I-22})$$

L'équation de prière El Fajr :

$$\text{Fajr Arc} = \arccos \left(\frac{\sin(-18) - \sin(\text{Dec}) \cdot \sin(\text{Lat})}{\cos(\text{Dec}) \cdot \cos(\text{Lat})} \right). \quad (\text{eq. I-23})$$

$$\text{Fajr Time} = \text{Local Noon} - (\text{Fajr Arc}/15). \quad (\text{eq. I-24})$$

I.6. Exemple d'application :

Les données à introduire dans ces équations pour calculer l'horaire des cinq prières pour la ville de Biskra sont les suivantes :

Date : 17 Mai 2012 ; 17/05/2012

$d=17$

$M=05$

$Y=2012$

Position : latitude : 34 degrés et 85 mn.

Longitude : 5 degrés et 733mn

Fuseau horaire : GMT +01 :00.

Alors ;

$Dj=4519.5$.

$L=55.0944$.

$M=131.94828$.

$\text{Lambda}=56.4987914$.

$\text{Obliquity}=23.4371922$.

$\text{Alpha}=54.19$.

$\text{ST}=235.093$.

$\text{Dec}=19.370$.

$\text{Noon}=179.0987$.

$\text{Utnoon}=174.0987$.

$\text{Local Noon} = 12.60658$.

Alors le temps de prière El Dohr est :

Pour calculer l'heure on prend l'entier de $12.60658 = 12$, et pour les minutes on prend l'entier du produit $(0.60658 * 60) = 36,3948$ est égale à 36 alors le temps el Dohr est : **12 h 36 min.**

Asr Alt=51.586.

Asr Arc=66.141.

Asr Arc=51.586/15=3.742.

Asr Time=12.60658+3.742=16.34858.

Pour calculer l'heure on prend l'entier de $16.34858 = 16$, et pour les minutes on prend l'entier du produit $(0.34858 * 60) = 20.9148$ est égale à 20 alors le temps est : **16 h 20 min.**

Durinal Arc=104.8169.

Elshouroge=5.618787227.

Pour calculer l'heure on prend l'entier de $5.618787227 = 5$, et pour les minutes on prend l'entier du produit $(0.618787227 * 60) = 37.1272$ est égale à 37 alors le temps **Elshouroge** est : **05 h 37 min.**

Elghouroube=19.59438.

Pour calculer l'heure on prend l'entier de $19.59438 = 19$, et pour les minutes on prend l'entier du produit $(0.59438 * 60) = 35.6628$ est égale à 35 .

alors le temps **Elghouroube** est : **19 h 35min .**

Esha Arc=129.2157.

Esha time=21.22096561.

Pour calculer l'heure on prend l'entier de $21.22096561 = 21$, et pour les minutes on prend l'entier du produit $(0.22096561 * 60) = 13.2579$ est égale à 13 alors le temps **Esha time** est : **21h 13min .**

Fajr Arc=129.2157.

Fajr time=3.9922.

Pour calculer l'heure on prend l'entier de $3.9922 = 3$, et pour les minutes on prend l'entier du produit $(0.9922 * 60) = 59.532$ est égale à 59 alors le temps **Fajr time** est : **03 h 59 min .**

I.7. CONCLUSION

La position de la terre joue un rôle essentiel pour la variation des temps de la prière islamique, et c'est pour ça les astronomes arabe ont développer des équations pour calculer c'est temps de la prière en fonction de longitude et latitude et la date, dans notre projet nous allons utiliser c'est équations dans le programme principale de Pic .

Chapitre 2

Présentation du PIC16F877

II.1. Introduction :

Les PICs sont des microcontrôleurs à architecture RISC (Reduce Instructions Construction Set), ou encore composant à jeu d'instructions réduit.

L'avantage est que plus on réduit le nombre d'instructions, plus leur décodage sera rapide ce qui augmente la vitesse de fonctionnement du microcontrôleur. La famille des PICs est subdivisée en 3 grandes familles :

La famille **Base-Line**, qui utilise des mots d'instructions de 12 bits, la famille **Mid-Range**, qui utilise des mots de 14 bits (et dont font partie le 16F86 et 16F877), et la famille **High-End**, qui utilise des mots de 16 bits. Les PICs 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 document à la famille Mid-Range et particulièrement au PIC 16F876/877, sachant que si on a tout assimilé, on pourra facilement passer à une autre famille, et même à un autre microcontrôleur.

Tableau II-1 : caractéristiques de la famille 16F876 – 16F877.

PIC	FLASH	RAM	EEPROM	I/O	A/D	Port //	Port Série
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Les éléments essentiels du PIC 16F876 et 16F877 sont : [6]

- Une mémoire programme de type EEPROM flash de 8K mots de 14 bits .
- Une RAM donnée de 368 octets .
- Une mémoire EEPROM de 256 octets .
- Trois ports d'entrée sortie, A (6 bits), B (8 bits), C (8 bits) .
- Convertisseur Analogiques numériques 10 bits à 5 canaux pour le 16F876, (8 canaux pour le 16F877) .
- USART, Port série universel, mode asynchrone (RS232) et mode synchrone .
- SSP, Port série synchrone supportant I2C .
- Trois TIMERS avec leurs Prescalers, TMR0, TMR1, TMR2 .
- Deux modules de comparaison et Capture CCP1 et CCP2 .
- Un chien de garde .
- 13 sources d'interruption .

Chapitre 2 : Présentation du PIC.

- Générateur d'horloge, à quartz (jusqu' à 20 MHz) .
- Fonctionnement en mode sleep pour réduction de la consommation.
- Programmation par mode ICSP (In Circuit Serial Programming) 12V ou 5V.
- Possibilité aux applications utilisateur d'accéder à la mémoire programme.
- Tension de fonctionnement de 2 à 5V.
- Jeux de 35 instructions.

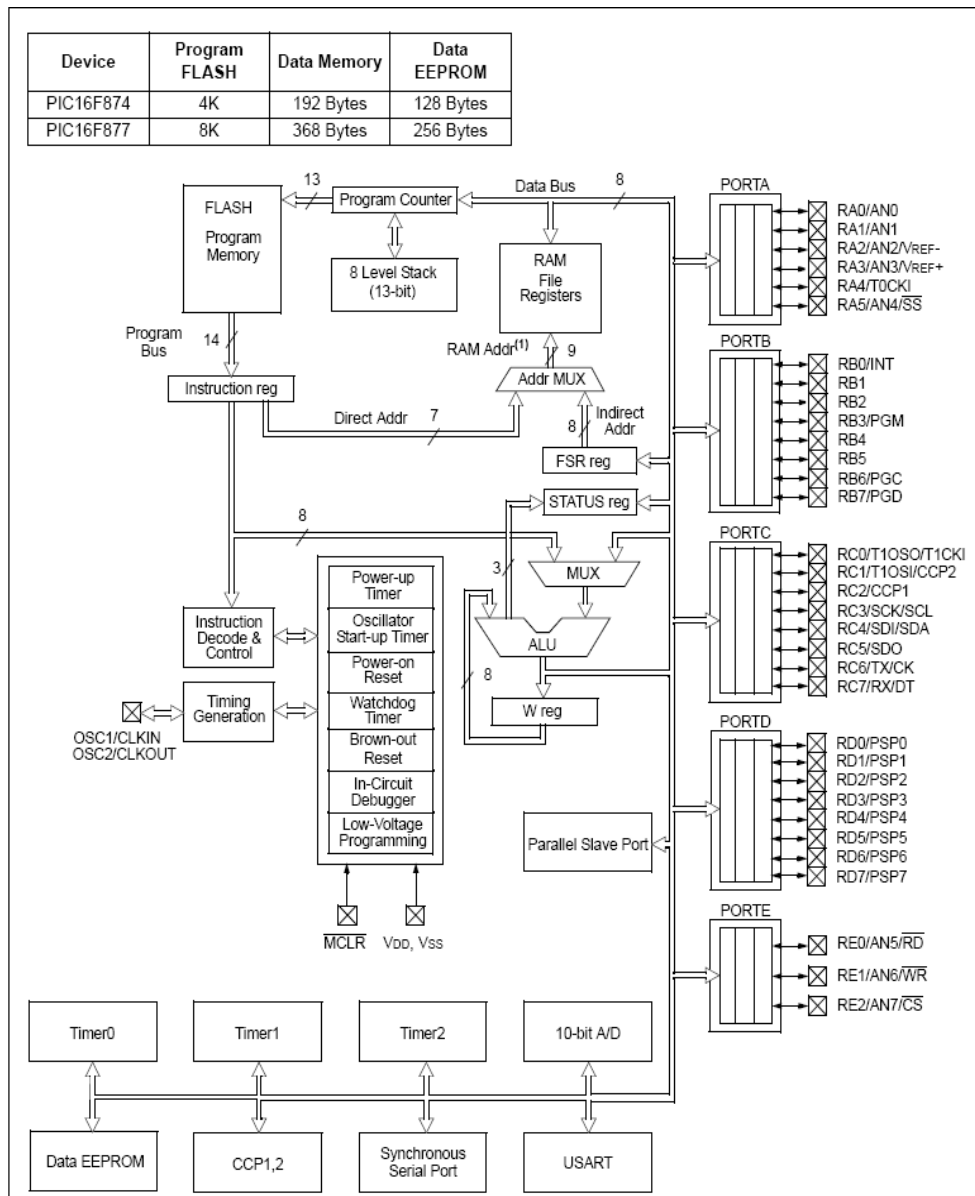


Fig. II-1 : Les éléments constitutifs du PIC 16F877 . [6]

Le port D (8 bits) et le port E (3 bits) ne sont pas disponibles sur 16F876.

II.2. BROCHAGE 16F877 :(40 broches) [6]

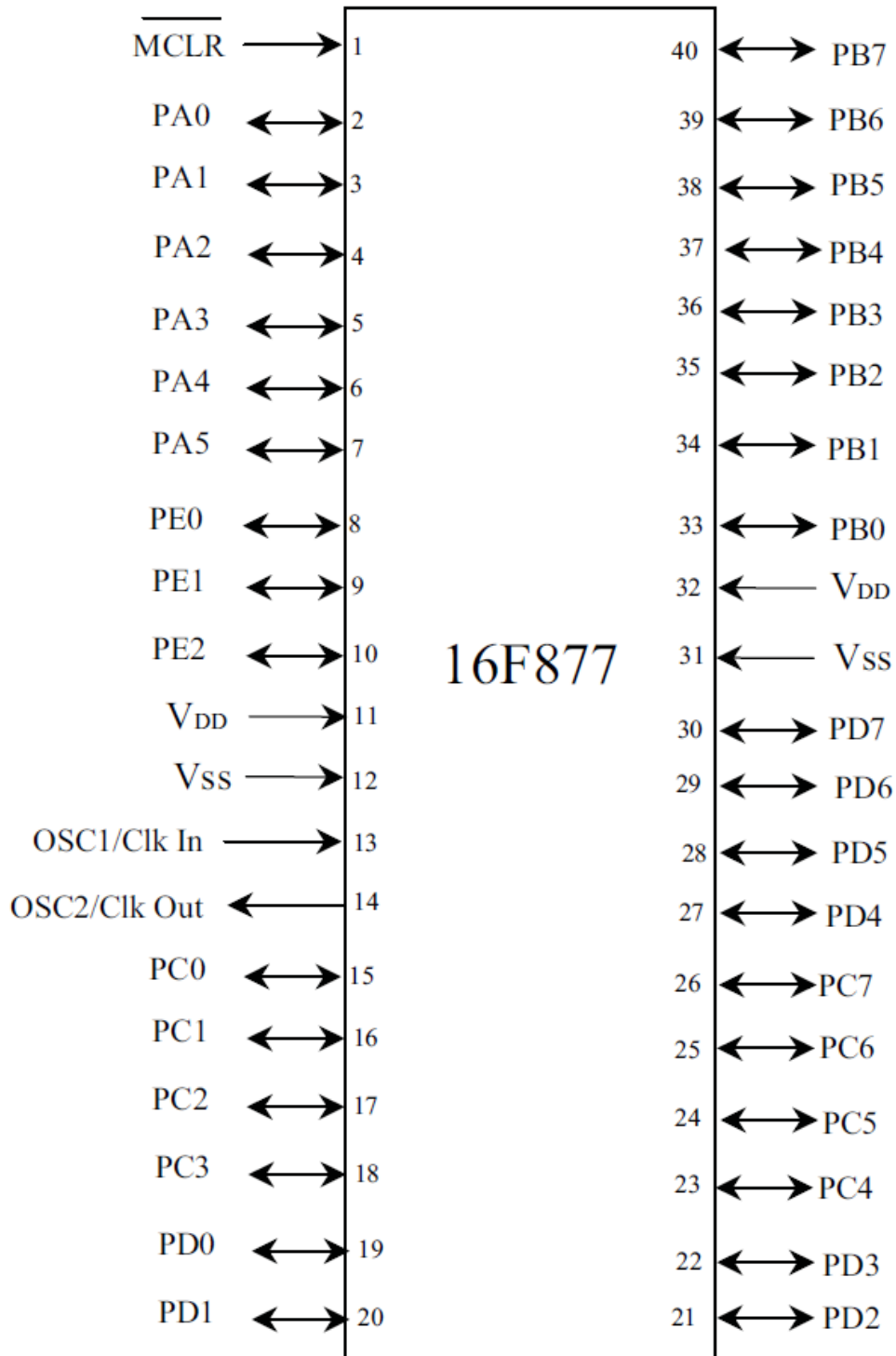


Fig. II-2 : Brochage de Pic 16F877. [6]

II.3. l'organisation mémoire du PIC 16F877: [6]

II.3.1 La mémoire programme :

La mémoire programme est constituée de 8K mots de 14 bits. C'est dans cette zone qu'on va écrire notre programme. Elle est accessible par le programme et peut donc être utilisée comme Une Extension de la mémoire EEPROM de données. Elle est non volatile (flash) et reprogrammable à souhait. Chaque position de 14 bits contient une instruction.

II.3. 2 La mémoire EEPROM :

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets qu'on peut lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant (non Volatile) et sont très utiles pour conserver des paramètres semi permanents. Leur utilisation implique une procédure spéciale que nous verrons par la suite, car ce n'est pas de la RAM, mais bien une ROM de type spécial. Il est donc plus rapide de la lire que d'y écrire.

II.3. 3 La mémoire RAM:

La mémoire RAM est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant donc c'est une mémoire volatile. La mémoire RAM disponible du 16F877 est de 368 octets. Elle est répartie et organisée en quatre banques de la manière suivante :

- 1) 80 octets en banque 0, adresses 0x20 à 0x6F
- 2) 80 octets en banque 1, adresses 0xA0 à 0XEF
- 3) 96 octets en banque 2, adresses 0x110 à 0x16F
- 4) 96 octets en banque 3, adresses 0x190 à 0x1EF

Dans chacune des banques nous allons trouver des « Cases mémoires spéciales » appelées REGISTRES SPÉCIAUX et des cases mémoires « libres » dont nous pouvons nous servir à notre guise. Chaque registre provoque un fonctionnement spécial du PIC ou la mise en service d'une fonction particulière. La Figure II.4 montre l'organisation de la RAM verrons par la suite, car ce n'est pas de la RAM, mais bien une ROM de type spécial. Il est donc plus rapide de la lire que d'y écrire.

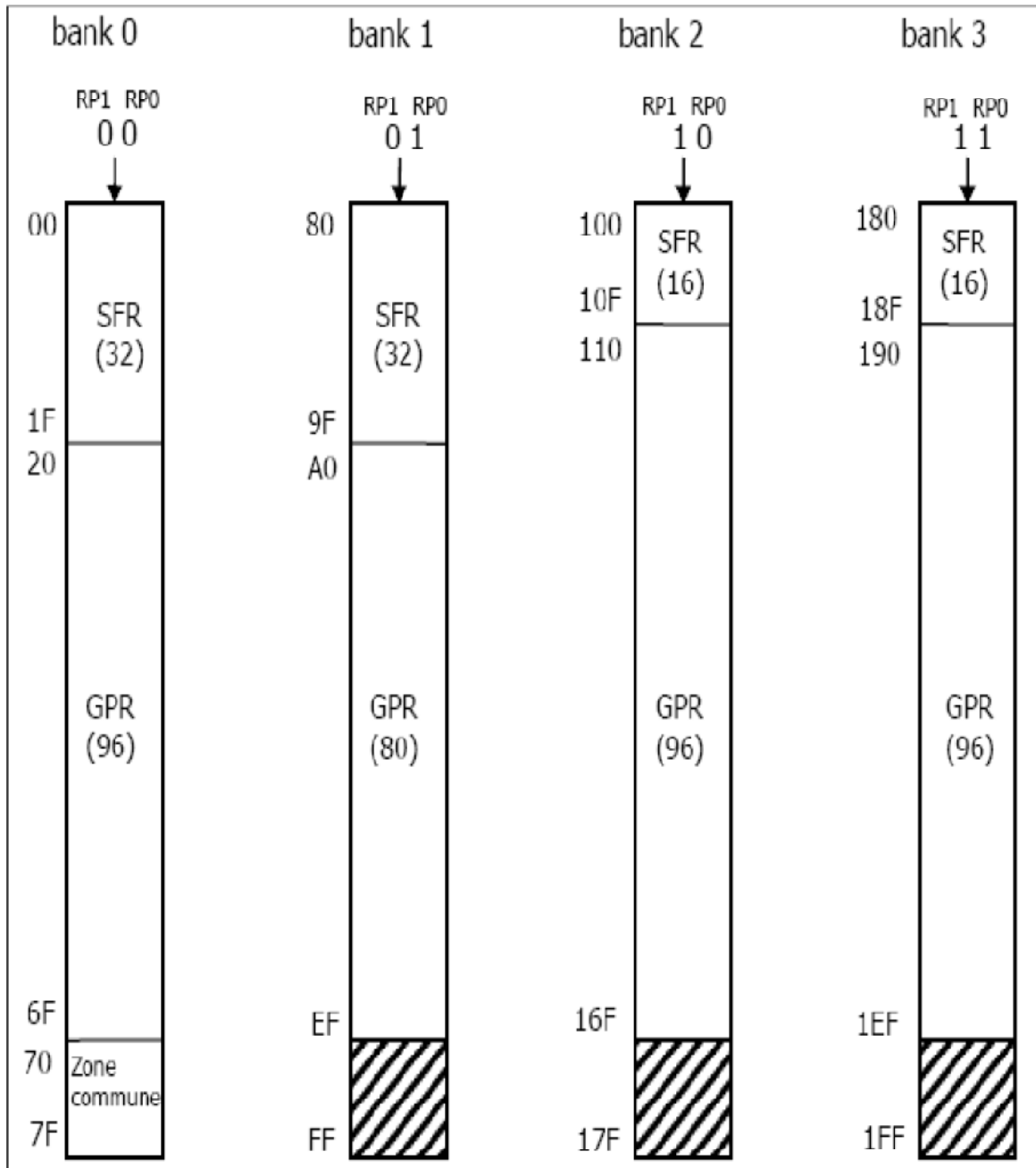


Fig. II -4: organisation de la RAM du 16F876/877

Chapitre 2 : Présentation du PIC.

Bank 0 (00)		Bank 1 (01)		Bank 2 (10)		Bank 3 (11)	
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION_REG	101h	TMR0	181h	OPTION_REG
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h	PORTC	87h	TRISC	107h		187h	
08h	PORTD (*)	88h	TRISD (*)	108h		188h	
09h	PORTE (*)	89h	TRISE (*)	109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	
0Fh	TMR1H	8Fh		10Fh	EEADRH	18Fh	
10h	T1CON	90h		110h		190h	
11h	TMR2	91h	SSPCON2				
12h	T2CON	92h	PR2				
13h	SSPBUF	93h	SSPADD				
14h	SSPCON	94h	SSPSTAT				
15h	CCPR1L	95h					
16h	CCPR1H	96h					
17h	CCP1CON	97h					
18h	RCSTA	98h	TXSTA				
19h	TXREG	99h	SPBRG				
1Ah	RCREG	9Ah					
1Bh	CCPR2L	9Bh					
1Ch	CCPR2H	9Ch					
1Dh	CCP2CON	9Dh					
1Eh	ADRESH	9Eh	ADRESL				
1Fh	ADCON0	9Fh	ADCON1				
20h		A0h					
6Fh		EFh		16Fh		1EFh	
70h	Zone commune	F0h		170h		1F0h	
7Fh		FFh		17Fh		1FFh	

Fig. II-5 : organisation de la RAM du 16F876/877

II.4. Les registres internes : [7]

II.4.a. Le registre de STATUS: (h'03' ou h'83' ou h'103 ou h'183').

Bit 7

Bit 0

IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
-----	-----	-----	------------------------	------------------------	---	----	---

Au reset : STATUS = 00011XXX

IRP = permet la sélection des pages en adressage indirect.

RP1 et RP0 = permettent la sélection des pages en adressage direct.

RP1	RP0	Page sélectionnée
0	0	PAGE 0 de 00 à 7F
0	1	PAGE 1 de 80 à FF
1	0	PAGE 2 de 100 à 17F
1	1	PAGE 3 de 180 à 1FF

Bit 4 : **TO** = Time Out bit.

Bit en lecture seulement.

1 = Après une mise sous tension, après une RAZ du watchdog (CLRWDT) ou bien après l'instruction SLEEP.

0 = Signifie qu'un Time Out du timer de watchdog est survenu.

Bit 3 : **PD** = Power Down bit.

1 = Après une mise sous tension ou bien après une RAZ du Watchdog.

0 = Après l'instruction SLEEP.

Bit 2 : **Z** = Zero bit.

1 = Le résultat d'une opération arithmétique ou logique est zéro.

0 = Le résultat d'une opération arithmétique ou logique est différent de zéro.

Bit 1 : **DC** = Digit Carry bit.

1 = Une retenue sur le 4eme bit des poids faible est survenue après les instructions : ADDWF et ADDLW.

0 = Pas de retenue sur le 4eme bit des poids faible.

Bit 0 : **C** = Carry bit.

1 = Une retenue sur le bit MSB est survenue après les instructions ADDWF et ADDLW.

Chapitre 2 : Présentation du PIC.

0 = Pas de retenue sur le bit MSB.

II.4.b. OPTION REGISTER : (h'81' ou h'181').

Ce registre en lecture écriture permet de configurer les prédiviseurs du Timer et du Watchdog , la source du Timer, le front des interruptions et le choix du Pull up sur le Port B.

Bit 7

Bit 0

$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
--------------------------	--------	------	------	-----	-----	-----	-----

Au reset : OPTION = 1111111

Bit 7 : RBPU = Pull up Enable bit on Port B.

1 = Pull up désactivé sur le Port B.

0 = Pull up active.

Bit 6 : INTEDG = Interrupt Edge select bit.

1 = Interruption si front montant sur la broche PB0/IRQ (pin 6).

0 = Interruption si front descendant sur PB0/IRQ.

Bit 5 : TOCS = Timer TMR0 Clock Source select bit.

1 = L'horloge du Timer est l'entrée PA4/Clk (pin 3).

0 = Le Timer utilise l'horloge interne du PIC.

Bit 4 : TOSE = Timer TMR0 Source Edge select bit.

1 = Le Timer s'incrémente à chaque front montant de la broche PA4/Clk.

0 = Le Timer s'incrémente à chaque front descendant de la broche PA4/Clk.

Bit 3 : PSA = Prescaler Assignment bit.

1 = Le prédiviseur est affecté au watchdog..

0 = Le prédiviseur est affecté au Timer TMR0.

Bits 2 à 0 : PS2 PS1 PS0 = Prescaler Rate Select bits.

Tableau II-2 : OPTION REGISTER.

PS2	PS1	PS0	Prédiv Timer	Prédiv Watchdog
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

Quand le pré diviseur est affecté au Watchdog (PSA=1), TMR0 est pré divisé par 1.

II.4.c. INTCON REGISTER : (h'0B' ou h'8B' ou h'10B' ou h'18B').

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

Bit 7

Bit 0

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
-----	------	------	------	------	------	------	------

Au reset : INTCON = 0000000X

Bit 7 : GIE = Global Interrup Enable bit

1 = Autorise toutes les interruptions non masquées.

0 = Désactive toutes les interruptions.

Bit 6 : PEIE = Peripheral Interrupt Enable bit.

1 = Autorise les interruptions causées par les périphériques.

0 = Désactive les interruptions causées par le périphériques.

Bit 5 : TOIE = Timer TMR0 Overflow Interrup Enable bit.

1 = Autorise les interruptions du Timer TMR0.

0 = Désactive les interruptions du Timer TMR0.

Bit 4 : INTE = RB0/Int Interrup Enable bit.

1 = Autorise les interruptions sur la broche : PB0/IRQ (pin6).

0 = Désactive les interruptions sur la broche : PB0/IRQ (pin6).

Bit 3 : RBIE = RB Port Change Interrup Enable bit.

1 = Autorise les interruptions par changement d'état du Port B (PB4 à PB7).

Chapitre 2 : Présentation du PIC.

0 = Désactive les interruptions par changement d'état du Port B (PB4 à PB7).

Bit 2 : TOIF = Timer TMR0 Overflow Interrupt Flag bit.

1 = Le Timer à débordé. Ce flag doit être remis à zéro par programme.

0 = Le Timer n'a pas débordé.

Bit 1 : INTF = RB0/Int Interrupt Flag bit.

1 = Une interruption sur la broche PB0/IRQ (pin 6) est survenue.

0 = Pas d' interruption sur la broche PB0/IRQ (pin 6).

Bit 0 : RBIF = RB Port Change Interrupt Flag bit. Ce flag doit être remis à zéro par programme.

1 = Quand au moins une entrée du port B (de PB4 à PB7) a changé d'état.

0 = Aucune entrée de PB4 à PB7 n'a changé d'état.

II.5. Les ports d'Entrée /Sortie :

II.5. a. Le port d' E/S PORTA :

Le port A désigné par PORTA est un port de 6 bits (RA0 à RA5). RA6 et RA7 ne sont pas accessibles.

La configuration de direction se fait à l'aide du registre TRISA :

Bit i de TRISA = 0 .bit i de PORTA configuré en **sortie**

Bit i de TRISA = 1 .bit i de PORTA configuré en **entrée**

En entrée, la broche RA4 peut être utilisée soit comme E/S numérique normale, soit comme entrée horloge pour le Timer TMR0.

En sortie, RA4 est une E/S à drain ouvert, pour l'utiliser comme sortie logique, il faut ajouter une résistance de pull-up externe. Le schéma (Fig. II-6) illustre le principe d'une sortie drain ouvert (ou collecteur ouvert) :

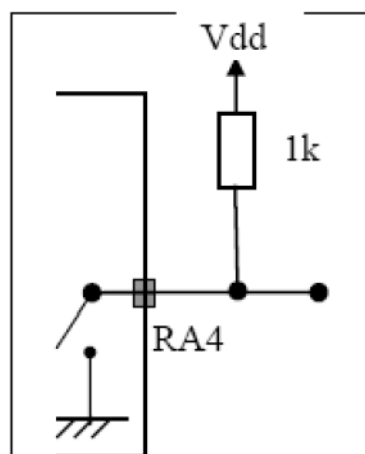


Fig. II-6 : résistance de pull-up

Chapitre 2 : Présentation du PIC.

Si RA4 est positionnée à 0, l'interrupteur est fermé, la sortie est reliée à la masse, c'est un niveau bas. Si RA4 est placée à 1, l'interrupteur est ouvert, la sortie est positionnée en niveau haut grâce à la résistance externe qui place la sortie au niveau haut.

Si on veut utiliser RA4 pour allumer une LED, on peut utiliser le schéma de Fig. II-7 faut juste remarquer que la logique est inversée, si on envoie 0 sur RA4, l'interrupteur se ferme et la LED s'allume. Si on envoie 1, l'interrupteur s'ouvre et la LED s'éteint.

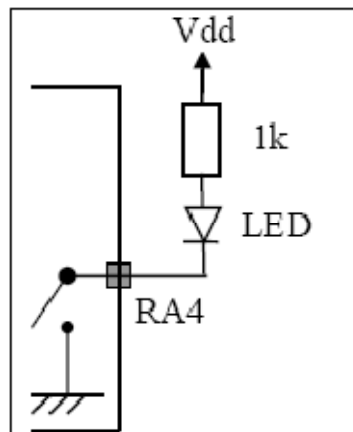


Fig. II-7 : LED sur RA4

II.5. b. Les autres broches de PORTA :

Les autres broches (RA0, RA1, RA2, RA3 et RA5) peuvent être utilisées soit comme E/S numériques soit comme entrées analogiques.

Au RESET, ces E/S sont configurées en entrées analogiques. Pour les utiliser en E/S numériques, il faut écrire '00000110' dans le registre ADCON1.

Pour utiliser PORTA en port Numérique (normal), il faut placer 06h dans le registre ADCON1 (bank1).

Quelque soit le mode (Analogique ou Numérique), il faut utiliser le registre TRISA pour configurer la direction des E/S :

II.5. c. Le port d' E/S PORTB :

Il comporte 8 bits. Le registre de direction correspondant est TRISB.

Si on écrit un "1" dans le registre TRISB, le driver de sortie correspondant passe en haute impédance. Si on écrit un "0", le contenu du Latch de sortie correspondant est recopié sur la broche de sortie.

Chaque broche du PORT B est munie d'un tirage au +VDD que l'on peut mettre ou non en service en mode entrée uniquement. On active cette fonction par la mise à "0" du bit 7 dans le registre OPTION en h'81'.

Chapitre 2 : Présentation du PIC.

Au reset, le tirage est désactivé. Il est inactif quand le port est configuré en sortie. Les 4 broches PB7 PB6 PB5 et PB4 provoquent une interruption sur un changement d'état si elles sont configurées en ENTREE.

On doit remettre à zéro le Flag de cette interruption (bit 0 du registre INTCON en h'0B') dans le programme d'interruption.

Cette possibilité d'interruption sur un changement d'état associé à la fonction de tirage configurable sur ces 4 broches, permet l'interfaçage facile avec un clavier. Cela rend possible le réveil du PIC en mode SLEEP par un appui sur une touche du clavier.

Le bit 0 du PORT B peut également être utilisé comme entrée d'interruption externe. Le choix du front de déclenchement se fait en configurant le bit 6 du registre OPTION.

II.5. d. Le port d' E/S PORTC :

Il s'agit d'un PORT 8 bits bidirectionnel.

- Il est partagé avec le module de transmission synchrone I2C et l'USART.
- La configuration de direction se fait à l'aide du registre TRISC, positionner un bit de TRISC à 1 configure la broche correspondante de PORTC en entrée et inversement.

Au départ toutes les broches sont configurées en entrée.

Le port d' E/S PORTD

- Le port D désigné par PORTD est un port bidirectionnel de 8 bits (RD0 à RD7).
- Toutes les broches sont compatibles TTL et ont la fonction trigger de Schmitt en entrée.
- Chaque broche est configurable en entrée ou en sortie à l'aide du registre TRISD. Pour configurer une broche en entrée, on positionne le bit correspondant dans TRISD à 1 et inversement.
 - PORTD n'est pas implémenté sur 16F876, il est disponible sur le 16F877.
 - PORTD peut être utilisé dans un mode particulier appelé parallel slave port, pour cela il faut placer le bit PSPMODE (bit 4) de TRISE à 1. Dans ce cas les 3 bits de PORTE deviennent les entrées de control de ce port (RE, WE et CS)

Pour utiliser PORTD en mode normal, il faut placer le bit PSPMODE de TRISE à 0.

Le port d' E/S PORTE :

- PORTE contient seulement 3 bits RE0, RE1 et RE2. Les 3 sont configurables en entrée ou en sortie à l'aide des bits 0, 1 ou 2 du registre TRISE.
- PORTE n'est pas implémenté sur 16F876, il est disponible sur le 16F877.

Chapitre 2 : Présentation du PIC.

- Les 3 bits de PORTE peuvent être utilisés soit comme E/S numérique soit comme entrées analogiques du CAN. La configuration se fait à l'aide du registre ADCON1.
- Si le bit PSPMODE de TRISE est placé à 1, Les trois bits de PORTE deviennent les entrées de control du PORTD qui (dans ce cas) fonctionne en mode parallel Slave Mode .
- A la mise sous tension (RESET), les 3 broches de PORTE sont configurées comme entrées analogiques.

Pour utiliser les broches de PORTE en E/S numériques normales :

- Placer 06h dans ADCON1
- Placer le bit PSPMODE de TRISE à 0.

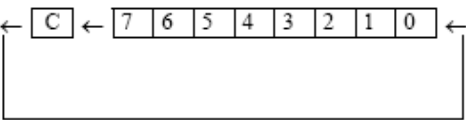
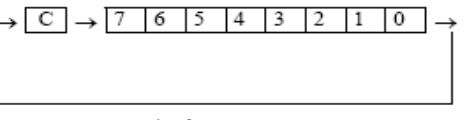
II.6. Le jeu d'instructions:

II.6. a. les instructions orienté octets :

Tableau II-3 : les instructions orienté octets.

Mnémonique	Instructions sur les registres (octets)	Bits status modifiés
ADDWF f,d	d:=W+f	C,DC,Z
ANDWF f,d	d:=W AND f	Z
CLRF f	f:=0	Z
CLRW	W:=0	Z
COMF f,d	d:=NOT(f)	Z
DECF f,d	d:=f-1	Z
DECFSZ f,d	d:=f-1 ; Skip if Zero	.
INCF f,d	d:=f+1	Z
INCFSZ f,d	d:=f+1 ; Skip if Zero	.
IORWF f,d	d:=W OR f	Z

Chapitre 2 : Présentation du PIC.

MOVF f,d	d:=f (permet de savoir si f=0 en faisant MOVF f, 1)	Z
MOVWF f	f:=W	.
NOP	n'effectue aucune opération	.
RLF f,d	 <p>d=f SHL 1</p>	C
RRF f,d	 <p>d=f SHR 1</p>	C
SUBWF f,d	d:= f-W (en complément à 2-->d:=f + not (W) +1)	C,DC,Z
SWAPF f,d	d:= f [4..7] <-- > f [0..3] (inverse les quartets)	.
XORWF f, d	d:= W XOR f	Z

II.6. b. Les instructions orienté bit :

Tableau II-4 : Les instructions orienté bit.

Mnémonique	Instructions sur les registres (bit par bit)	Bits status modifiés
BCF f, b	f[b]:=0 (mets à 0 le bit b de f).	.
BSF f, b	f[b]:=1 (mets à 1 le bit b de f).	.
BTFSC f,b	teste le bit b de f ; Skip if Clear (0)	.
BTFSS f,b	teste le bit b de f ; Skip if Set (1).	.

II.6. c. Les instructions de contrôle :

Tableau II-5 : Les instructions de contrôle.

Mnémonique	Instructions de contrôle	Bits status modifies
ADDLW k	W: =W+k	C,DC,Z
ANDLW k	W:=W AND k	Z
CALL k	appel un sous programme	.
CLRWDT	remet à 0 le timer du chien de garde	TO,PD
GOTO k	se branche à l'adresse k	.
IORLW k	W:=W OR k	Z
MOVLW k	W:=k	.
RETFIE	fin d'une interruption	.
RETLW k	w=k, et effectue un retour de sous-programme	.

RETURN	effectue un retour de sous programme	.
SLEEP	place le circuit en mode sommeil et stoppe l'oscillateur	TO,PD
SUBLW k	W: =W-k	C,DC,Z
XORLW k	W: =W XOR k	Z

Moyens mnémotechniques pour se rappeler des commandes :

Le L (comme littéral) indique une valeur immédiate .

f = registre d'adresse 7 bits

k = valeur immédiate 8 bits (0 à 127)

d = 0 : destination = W vers le Registre W de l'ALU du pic

d = 1 : destination = f vers la zone File (Fichier)

II.7. Configuration des PORTx (les registres PORTx et TRISx) :

Tous les ports sont pilotés par deux registres :

- Le registre de PORTx, si le PORT x ou certaines lignes de PORT X sont configurées en sortie, ce registre détermine l'état logique des sorties.
- Le registre TRISx, c'est le registre de direction. Il détermine si le PORTx ou certaines lignes de port sont en entrée ou en sortie.

L'écriture d'une 1 logique correspond à une entrée (1 comme Input) et l'écriture d'une 0 logique correspond à une sortie (0 comme Output).

Remarque :

- Les registres TRISx appartiennent à la BANQUE 1. Lors de l'initialisation du μ C il ne faut pas oublier de changer de page mémoire pour les configurer.
- Au RESET toutes les lignes de ports sont configurées en entrées.

Exemple

On souhaite obtenir la configuration suivante des PORTA et PORTB.

SENS	NC	NC	S	E	S	S	S	E
PORTA	NC	NC	RA5	RA4	RA3	RA2	RA1	RA0
SENS	E	E	S	S	E	S	S	E
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

Donc il faut avoir :

TRISA = B"11010001".

TRISB = B"11001001".

II.8. Horloge du système:

L'horloge peut être soit interne soit externe. L'horloge interne est constituée d'un oscillateur à quartz ou d'un oscillateur RC.

Avec l'oscillateur à Quartz, on peut avoir des fréquences allant jusqu'à 20 MHz selon le type de μ C. Le filtre passe bas (R_s , C_1 , C_2) limite les harmoniques dus à l'écrêtage et réduit l'amplitude de l'oscillation, il n'est pas obligatoire.

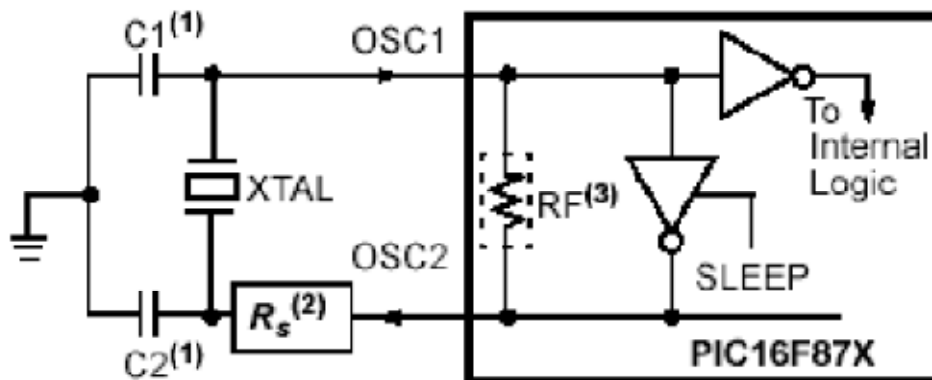


Fig. II-8 : Oscillateur à quartz du PIC 16F87x

II.9. Circuit de Reset:

Les microcontrôleurs PIC intègrent un circuit de RESET automatique qui se déclenche à la mise sous tension. En ajoutant un circuit externe, on réalise ainsi un RESET manuel.

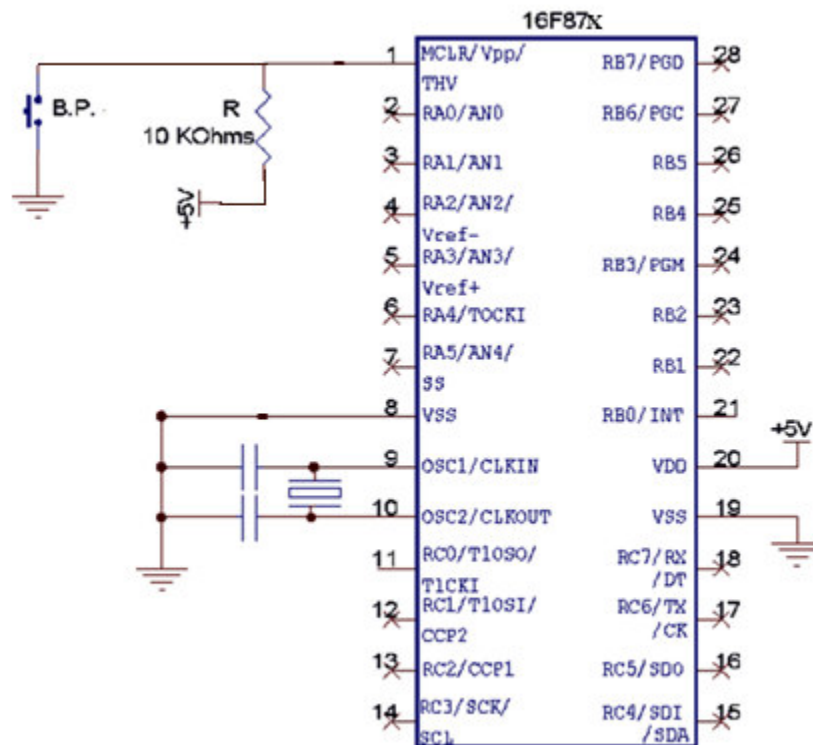


Fig. II -9 : Circuit de Reset

II.10. Les interruptions :

Une interruption provoque l'arrêt du programme principal pour aller exécuter une procédure d'interruption. A la fin de cette procédure, le microcontrôleur reprend le programme principal à l'endroit où il l'a laissé. A chaque interruption sont associés deux bits, un bit de validation et un drapeau. Le premier permet d'autoriser ou non l'interruption, le second permet au programmeur de savoir de quelle interruption il s'agit. Sur le 16F876/877, l'es interruptions sont classées en deux catégories, les interruptions primaires et les interruptions périphériques.

Elles sont gérées par les registres :

Tableau II-6 : Les interruptions primaires et les interruptions périphériques

INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
PIE1 (bk1)	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
PIR1 (bk0)	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE2 (bk0)	-	-	-	EEIE	BCLIE	-	-	CCP2IE
PIR2 (bk1)	-	-	-	EEIF	BCLIF	-	-	CCP2IF
OPTION_REG(bk1)		INTEDG						

- Toutes les interruptions peuvent être validées/interdites par le bit INTCON.GIE
- Toutes les interruptions périphériques peuvent être validées/interdites par le bit INTCON.PEIE .
- Chaque interruption peut être validée/interdite par son bit de validation individuel.

II.10.a. Les sources d'interruption :

Tableau II-7 : Les Sources d'interruption .

Interruption : Source d'interruption	Validation	Flag	PEIE
T0I : Débordement Timer 0	INTCON,T0IE	INTCON,T0IF	non
INT : Front sur RB0/INT	INTCON,INTE	INTCON,INTF	non
RBI : Front sur RB4-RB7	INTCON,RBIE	INTCON,RBIF	non
ADI : Fin de conversion A/N	PIE1,ADIE	PIR1,ADIF	oui
RCI : Un Octet est reçu sur l'USART	PIE1,RCIE	PIR1,RCIF	oui
TXI : Fin transmission d'un octet sur l'USART	PIE1,TXIE	PIR1,TXIF	oui
SSPI : Caractère émis/reçu sur port série synchrone	PIE1,SSPIE	PIR1,SSPIF	oui
TMR1I : Débordement de Timer 1	PIE1,TMR1IE	PIR1,TMR1IF	oui
TMR2I : Timer 2 a atteint la valeur programmée	PIE1,TMR2IE	PIR1,TMR2IF	oui
PSPI : Lecture/écriture terminée sur Port parallèle (16F877)	PIE1,PSPIE	PIR1,PSPIF	oui
CCP1I : Capture/comparaison de TMR1 avec module CCP1	PIE1,CCP1IE	PIR1,CCP1IF	oui
CCP2I : Capture/comparaison de TMR1 avec module CCP2	PIE2,CCP2IE	PIR2,CCP2IF	oui
EEI : Fin d'écriture en EEPROM	PIE2,EEIE	PIR2,EEIF	oui
BCLI : Collision sur bus SSP en mode I2C	PIE2,BCLIE	PIR2,BCLIF	oui

II.11. Module MSSP pour I2C :

Le module MSSP (Master Synchronous Serial Port) permet de faire des transmissions au format I2C (Inter Integrated Circuits) en tant que Maître ou Esclave.

Il utilise les pins 3 et 4 du PORT C: **PC3 = SCL et PC4 = SDA.**

Les 5 registres utilisés sont :

Registre Buffer **SSPBUF** en h'13' page 0.

Chapitre 2 : Présentation du PIC.

Registre d'adresse **SSPADD** en h'93' page 1.

Registre d'état **SSPSTAT** en h'94' page 1.

Registre de contrôle **SSPCON** en h'14' page 0.

Registre de contrôle n°2 **SSPCON2** en h'91' page 1.

II.11.1. SSPSTAT REGISTER : (h'94' : page 1).

Bit 7

Bit 0

SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
-----	-----	--------------	---	---	--------------	----	----

Au reset : SSPSTAT = 00000000

Bit 7 : **SMP** = bit d'échantillonnage.

Bit 6 : **CKE** = Clock Edge Select. Ce bit à "0" : conforme aux spécifications I2C.

Bit 5 : **D/A** = Data / Adresse bit.

1 = indique que le dernier octet reçu est une Data.

0 = indique que le dernier octet reçu est une adresse.

Bit 4 : **P** = STOP bit. Ce bit est remis à "0" quand le module SSP est désactivé.

1 = indique qu'un Stop a été détecté.

0 = indique qu'il n'a pas été détecté de Stop.

Bit 3 : **S** = START bit. Ce bit est remis à "0" quand le module SSP est désactivé.

1 = indique qu'un Start a été détecté.

0 = indique qu'il n'a pas été détecté de Start.

Bit 2 : **R/W** = Information sur le bit R/W. Donne la valeur de ce bit qui suit la dernière adresse. Ce bit est valable jusqu'à la réception du STOP.

- Mode ESCLAVE :

1 = R.

0 = W.

- Mode MAITRE :

1 = Transmission en cours.

0 = Pas de transmission en cours.

Bit 1 : **UA** = mise à jour adresse. Utilisé en mode 10 bits seulement.

Bit 0 : **BF** = Buffer plein.

- Mode Réception I2C:

Chapitre 2 : Présentation du PIC.

1 = Registre plein : réception terminée.

0 = Registre vide : donc réception non terminée.

- Mode Emission I2C:

1 = Registre plein : donc émission en cours.

0 = Registre vide : donc émission terminée.

II.11.2. SSPCON REGISTER : (h'14' : page 0).

Bit 7

Bit 0

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
------	-------	-------	-----	-------	-------	-------	-------

Au reset : SSPCON = 00000000

Bit 7 : **WCOL** = détection de collision à l'écriture.

1 = Collision à l'écriture.

0 = Pas de collision.

Bit 6 : **SSPOV** = Overflow indicator en réception. Ne sert pas en émission Doit être remis à "0" par soft

1 = Un octet vient d'être reçu dans le buffer qui n'était pas vidé : Overflow.

0 = Pas d'Overflow.

Bit 5 : **SSPEN** = Enable Module SSP.

1 = Module I2C activé et pin SCL et SDA sur PC3 et PC4 configurées.

0 = Module I2C désactivé. PC3 et PC4 libres pour usage I/O.

Bit 4 : **CKP** = Polarité du Clock. Inutilisé en mode maître.

En mode Esclave :

1 = Autorise horloge.

0 = Maintient CLK = 0.

Bit 3 à bit 0 : **SSPM3 / SSPM0** = Select Mode.

Tableau II-8 : Select Mode.

SSPM3	SSPM2	SSPM1	SSPM0	MODE
0	1	1	0	I2C esclave 7 bits.
0	1	1	1	I2C esclave 10 bits.
1	0	0	0	I2C maître CLK = $F_{osc}/4(SSPAD+1)$.

II.11.3. SSPCON2 REGISTER : (h'91' : page 1).

Bit 7

Bit 0

GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
------	---------	-------	-------	------	-----	------	-----

Au reset : SSPCON2 = 00000000

Bit 7 : **GCEN**= Appel général en mode I2C esclave seulement..

1 = Autorise IT quand une adresse appel général (h'0000) est reçue.

0 = Adresse appel général désactivée.

Bit 6 : **ACKSTAT** = Bit de ACK en mode I2C maître seulement.

1 = Pas de ACK reçu de l'esclave.

0 = ACK de l'esclave a été reçu.

Bit 5 : **ACKDT** = ACK Data bit. En mode maître et réception : valeur qui sera transmise quand on va lancer une séquence de ACK en fin de réception.

1 = NO ACK.

0 = ACK.

Bit 4 : **ACKEN** = ACK Sequence Enable. En mode maître et réception, lance une séquence de ACK ou de NOACK suivant la valeur dans ACKDT.

1 = Lance la séquence de ACK et transmet ACKDT. Remis à "0" par hard.

0 = Pas de séquence de ACK.

Bit 3 : **RCEN** = Receive Enable Bit. En mode I2C maître seulement.

1 = Autorise mode réception.

0 = Pas de réception autorisée.

Bit 2 : **PEN** = Stop Condition Enable. En mode I2C maître seulement.

1 = Lance la séquence de STOP sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de STOP.

Bit 1 : **RSEN** = Repeated Start Condition . En mode I2C maître seulement.

1 = Séquence de répétition de START sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de répétition de START.

Bit 0 : **SEN** = Start Condition Enable. En mode I2C maître seulement.

1 = Lance la séquence de START sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de START.

II.11.4. MODE MAITRE :

Les pins SDA et SCL sont manipulées par le Hard. Les événements qui

causent le passage à "1" du flag SSPIF (bit 3 de PIR1 en h'0C') et éventuellement une IT si elles sont autorisées, sont :

- Condition de START.
- Condition de STOP.
- Répétition d'un START.
- ACK après un transfert.
- Transfert d'octet en Emission ou réception.

Attention: Le flag SSPIF doit être remis à "0" par soft.

II.11.4.a. Lancement d'un START:

- L'utilisateur doit mettre le bit SEN (de SSPCON2) à "1".
- Les pins SDA et SCL étant toutes les deux à "1", le module fait passer la pin SDA de "1" à "0" ce qui génère une condition de START.
- Le bit S (de SSPSTAT) passe à "1" pour signaler le START.
- A la fin du START, le bit SEN est remis à "0" par le hard.
- Dès que le flag SSPIF (de PIR1) passe à "1" pour signaler la fin du START, on peut charger le registre de transmission SSBUF avec l'octet à transmettre. Ne pas oublier de remettre ce flag à "0" par le soft.

II.11.4.b. Transmission :

- Dès que le registre SSBUF est chargé, le bit BF (de SSPSTAT) passe à "1" pour signaler que la transmission est en cours.
- Au 8eme coup de CLK, la transmission est terminée et le bit BF repasse à "0".

II.11.4.c. Acquiescement :

- Quand l'esclave répond le ACK, le bit ACKSTAT passe à "0".
- La fin du ACK est signalé par le Flag SSPIF qui passe à "1". On doit alors remettre ce flag à "0" par soft.

II.11.4.d. Lancement d'un STOP :

- L'utilisateur doit mettre le bit PEN (de SSPCON2) à "1". Le module fait alors passer SDA à "0" puis force SCL à "1". Quand SCL est à "1" il fait passer SDA de "0" à "1", ce qui génère une condition de STOP.
- Le bit P (de SSPSTAT) passe à "1" pour signaler le STOP.
- A la fin du STOP, le bit PEN est remis à "0" par le hard.

Chapitre 2 : Présentation du PIC.

- Le flag SSPIF (de PIR1) passe à "1" pour signaler la fin du STOP.

II.11.5. Réponse du maître à l'esclave :

Il s'agit soit d'un ACK soit d'un NOACK :

- Positionner le bit ACKDT (de SSPCON2) en fonction de la réponse à faire.
- Mettre le bit ACKEN (de SSPCON2) à "1".
- La fin de transmission du ACK ou du non ACK est signalé par le flag SSPIF qui passe à "1". Ne pas oublier de le remettre à "0" par soft.

II.11.5.a. Réception d'un octet par le maître :

- Pour cela il faut mettre le module en réception en portant le bit RCEN à "1".
- La fin de réception est signalé par le passage à "1" du flag SSPIF, que l'on remettra à "0" par soft.
- On peut alors lire l'octet reçu dans SSBUF.

II.12. CONCLUSION

En conclusion, nous pouvons dire que le microcontrôleur 16F877 peut bien jouer le rôle d'une unité de commande pour notre système, il contient tous les modules nécessaires pour générer le signal d'horloge.

Chapitre 3

Réalisation

Software

III.1 Introduction :

Le processus de développement doit être exécuté en utilisant certains logiciels comme « mikroC PRO for PIC » pour programmer le Pic 16F877, et dans ce chapitre on va voir le fonctionnement de ce logiciel.

III.2. Présentation du logiciel « mikroC PRO for PIC » :

mikroC PRO pour PIC est un plein de fonctionnalités compilateur C ansi pour les appareils PIC de Microchip . Il est la meilleure solution pour développer du code pour les appareils PIC. Il dispose d'IDE intuitive, puissant compilateur avec des optimisations avancées, beaucoup de matériel et des bibliothèques logicielles et des outils supplémentaires qui nous aideront dans notre travail ,et il est fourni avec le fichier d'aide complet et beaucoup de prêt-à-utiliser des exemples destinés à nous aider à démarrer en un rien de temps , la (Fig. III.1) présente la fenêtre principale de mikroC PRO for PIC.

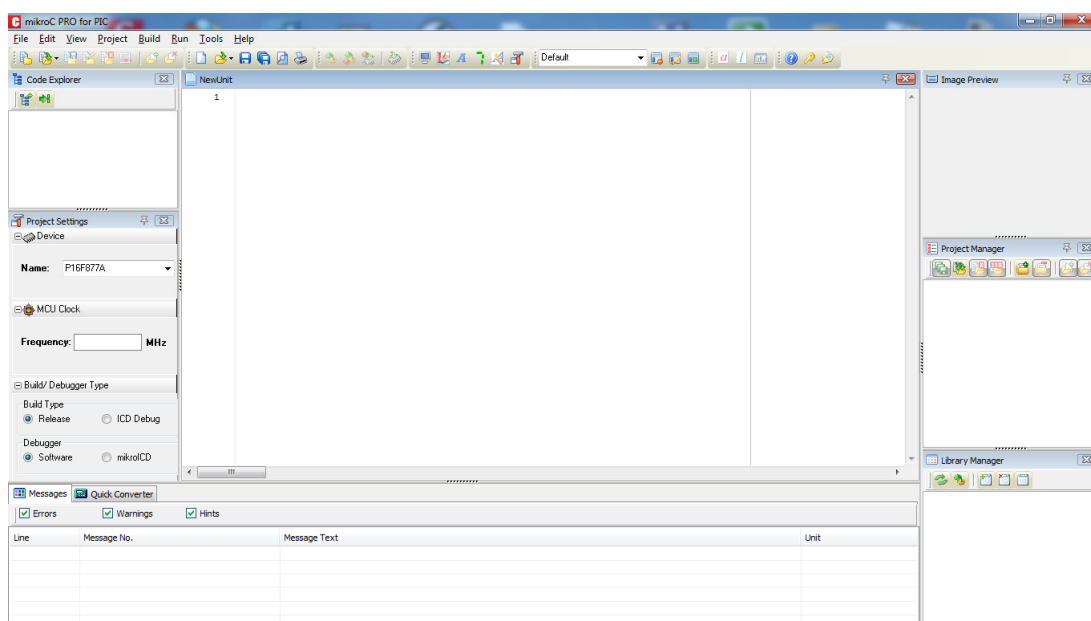


Fig. III.1 : Fenêtre principale de mikroC PRO for PIC.

III.3. Les étapes suivies pour crée un projet en mikroC PRO for PIC :

- Lancement du programme mikroC PRO for PIC : double cliques sûr l'icône de logiciel.
- Création d'un nouveau projet : aller a Project »New Project »Next. Voir figure (III.2).

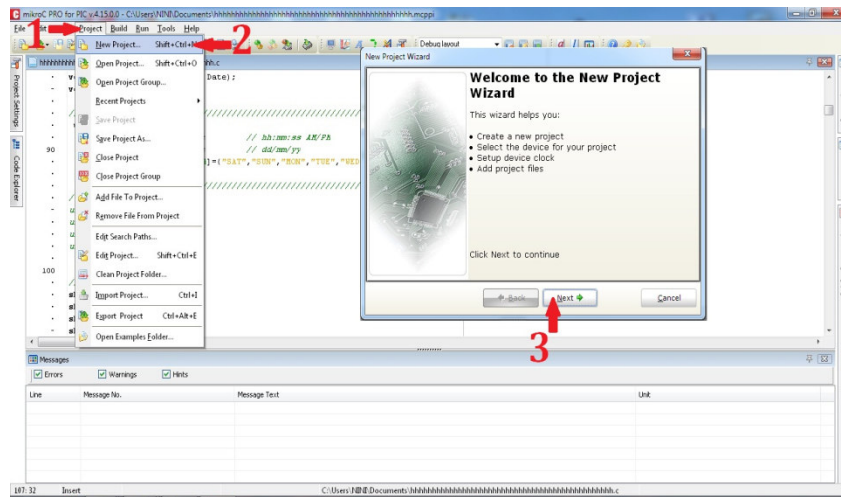


Fig.III.2.Création d'un nouveau projet.

- Choix de pic : Voir figure(III.3).

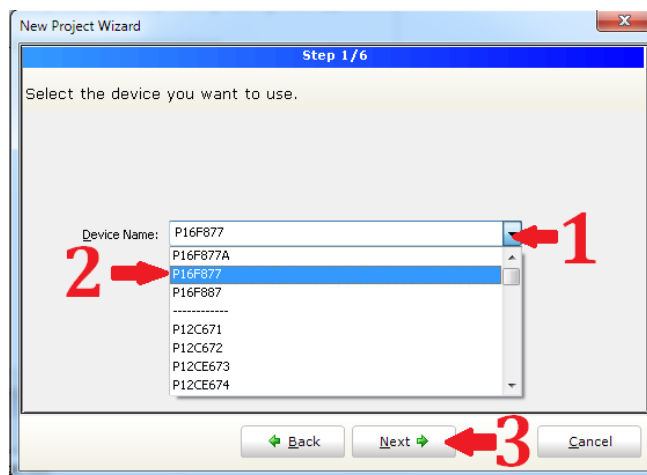


Fig.III.3.Choix de PIC.

- Choix de fréquence : Voir figure(III.4).

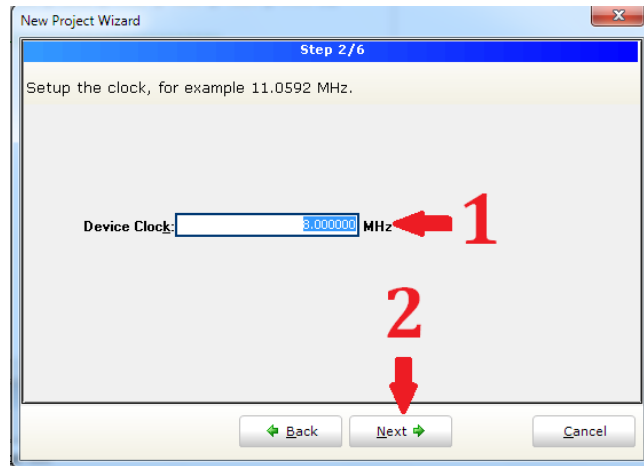


Fig.III.4.Choix de frequence.

- Nommer le projet : Voir figure(III.5).

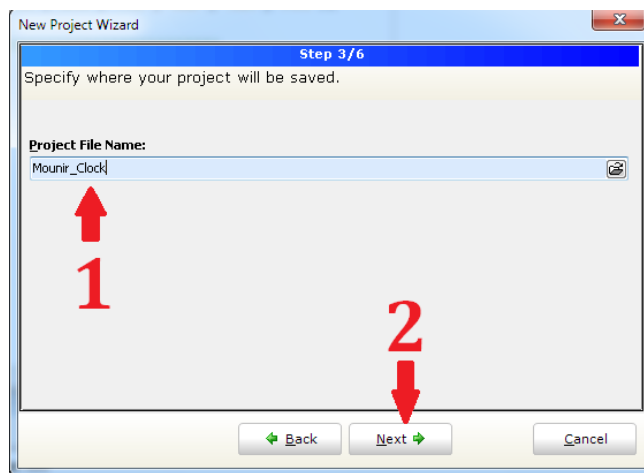


Fig.III.5.Nommer le projet.

- L'ajoute des fichiers supplémentaires : Voir figure(III.6).

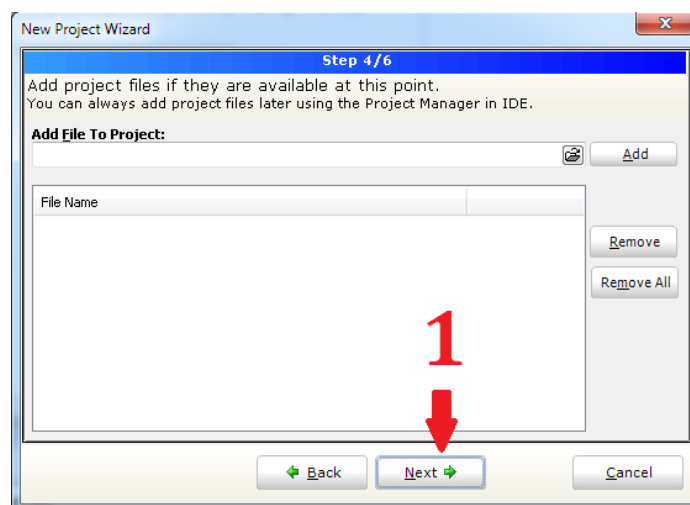


Fig.III.6.Fenêtre d'ajoute des fichiers supplémentaires.

- La sélectionne de la bibliothèque : Voir figure(III.7).

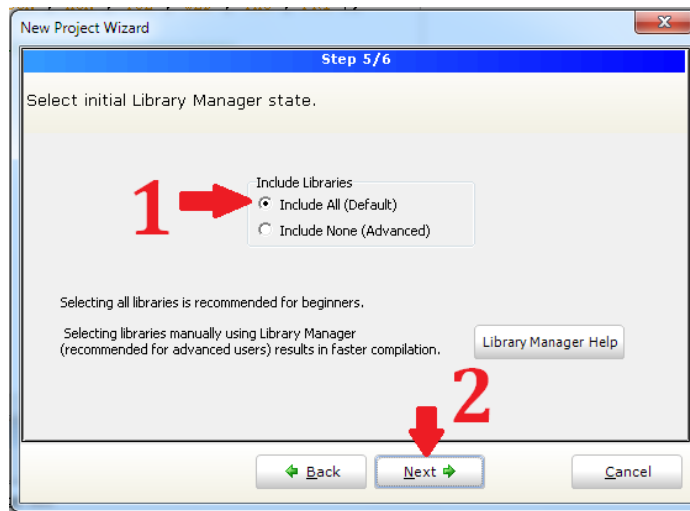


Fig.III.7.La sélectionne de la bibliothèque.

- La fin de la création d'un projet : Voir figure(III.8).

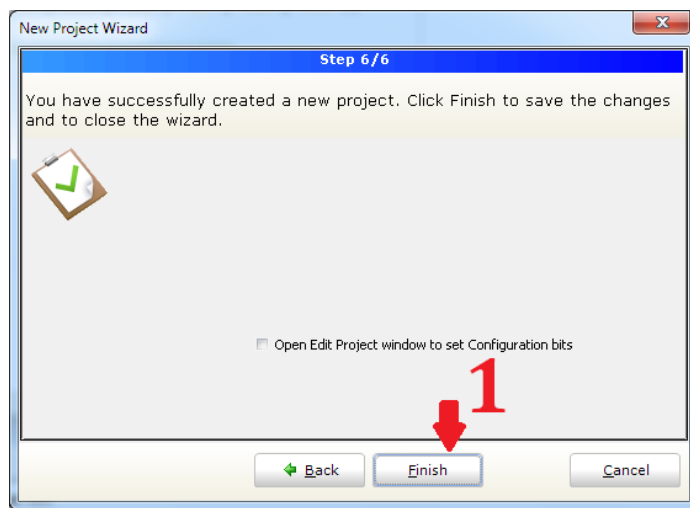


Fig. III.8.La fin de la création d'un projet.

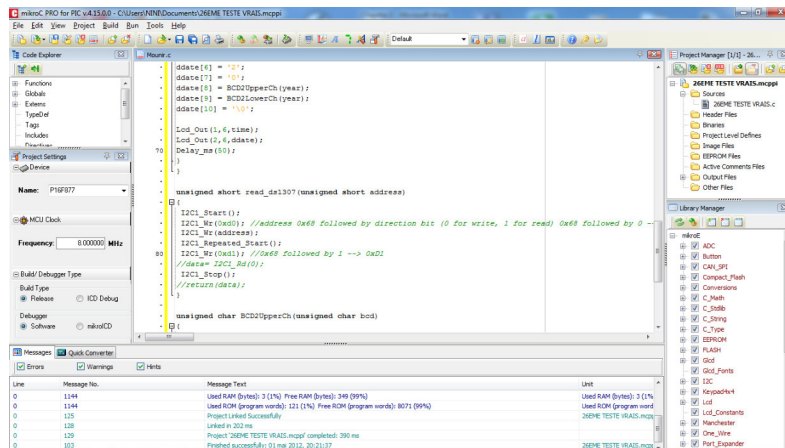


Fig. III.9.fenêtre de compilation .

III.4. L' Universal Programmer (ALL-11P3):

Universal Programmer est un programme permet de transférer un fichier compilé (hex) vers un Microcontrôleur (PIC 16F877) ou un mémoire de type EEPROM (DS1307).



Fig. III.10. Fenêtre principale d' Universal Programmer

- Le choix du Pic :on choisie MPU/MCU »PIC16F877 »Run. Voir figure(III.11).

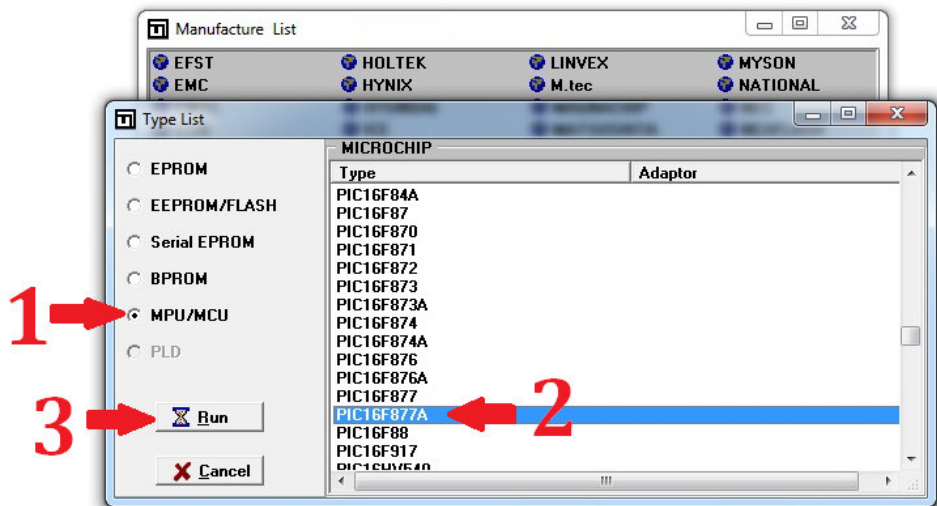


Fig. III.11. La fenêtre de choix du PIC.

- Le chargement de fichier hexa : On charge le fichier que nous désirons envoyer au pic . Voir figure(III.12).

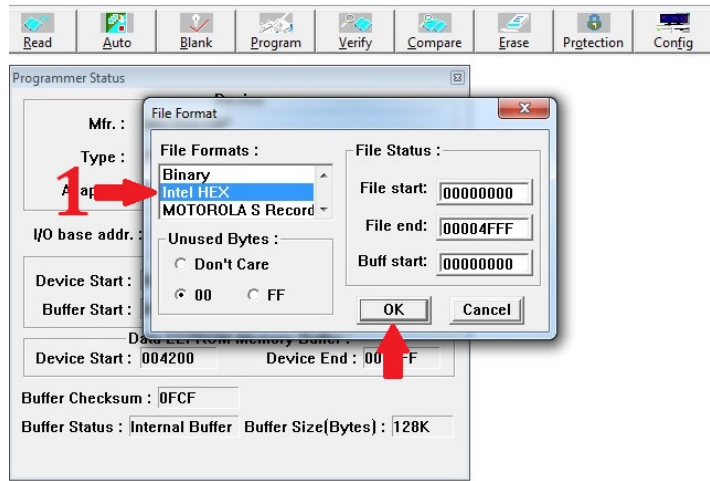


Fig. III.12. La fenêtre de chargement de fichier hexa.

- Le transfert de fichier hexa au Pic : Aller a Auto »cocher les opérations »Run. Voir figure(III.13).

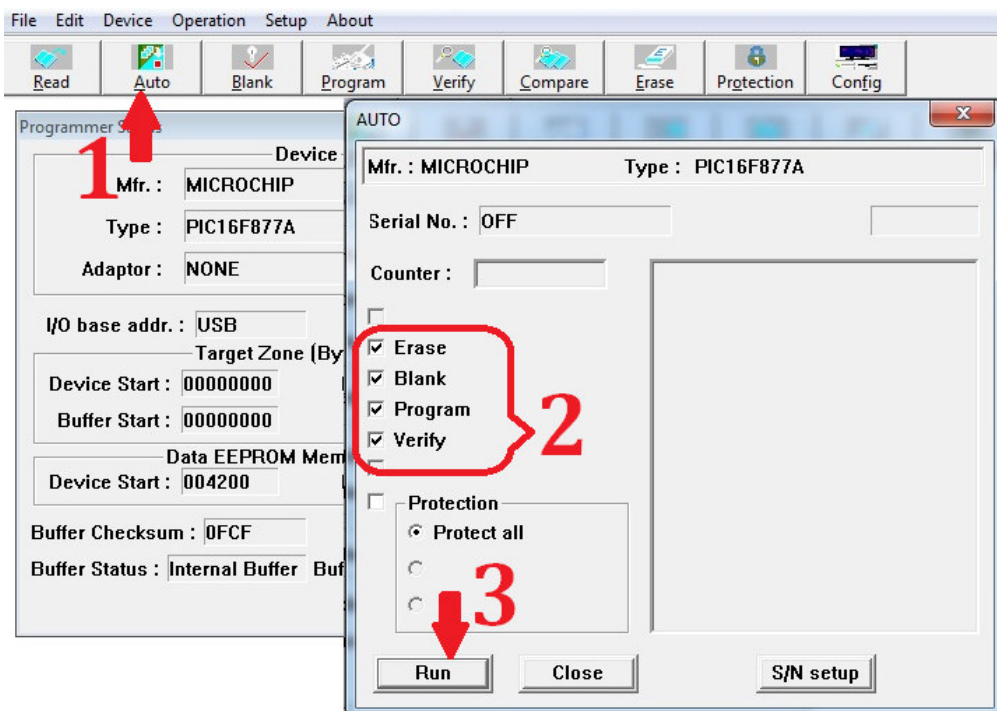


Fig. III.13. La fenêtre du transfert de fichier hexa.

III.5. L'Organigramme de programme global :

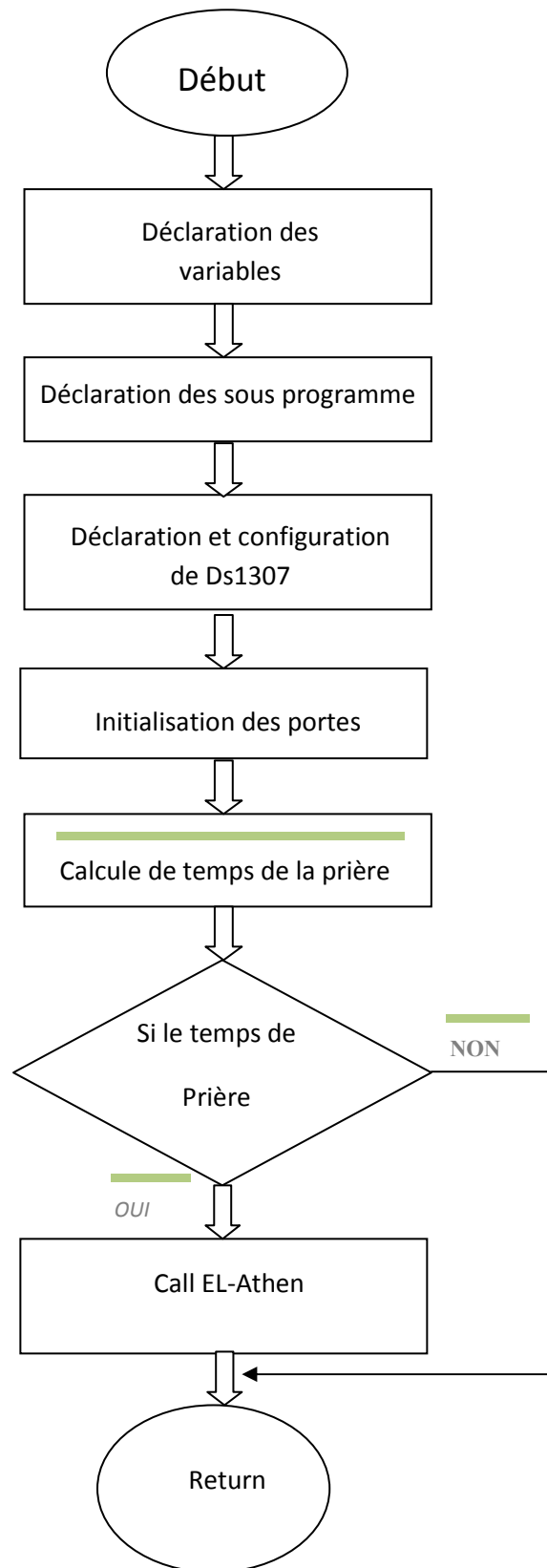
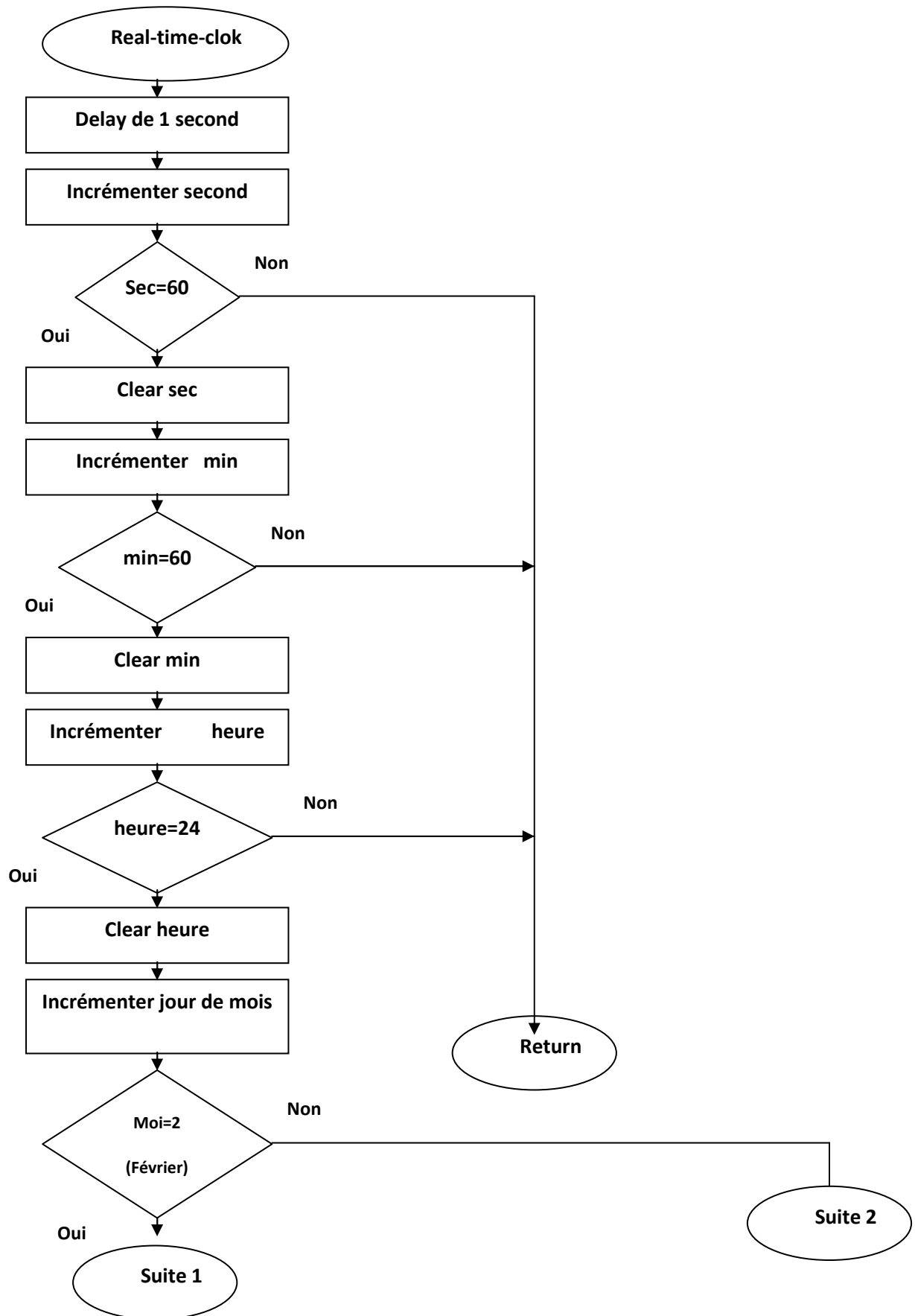


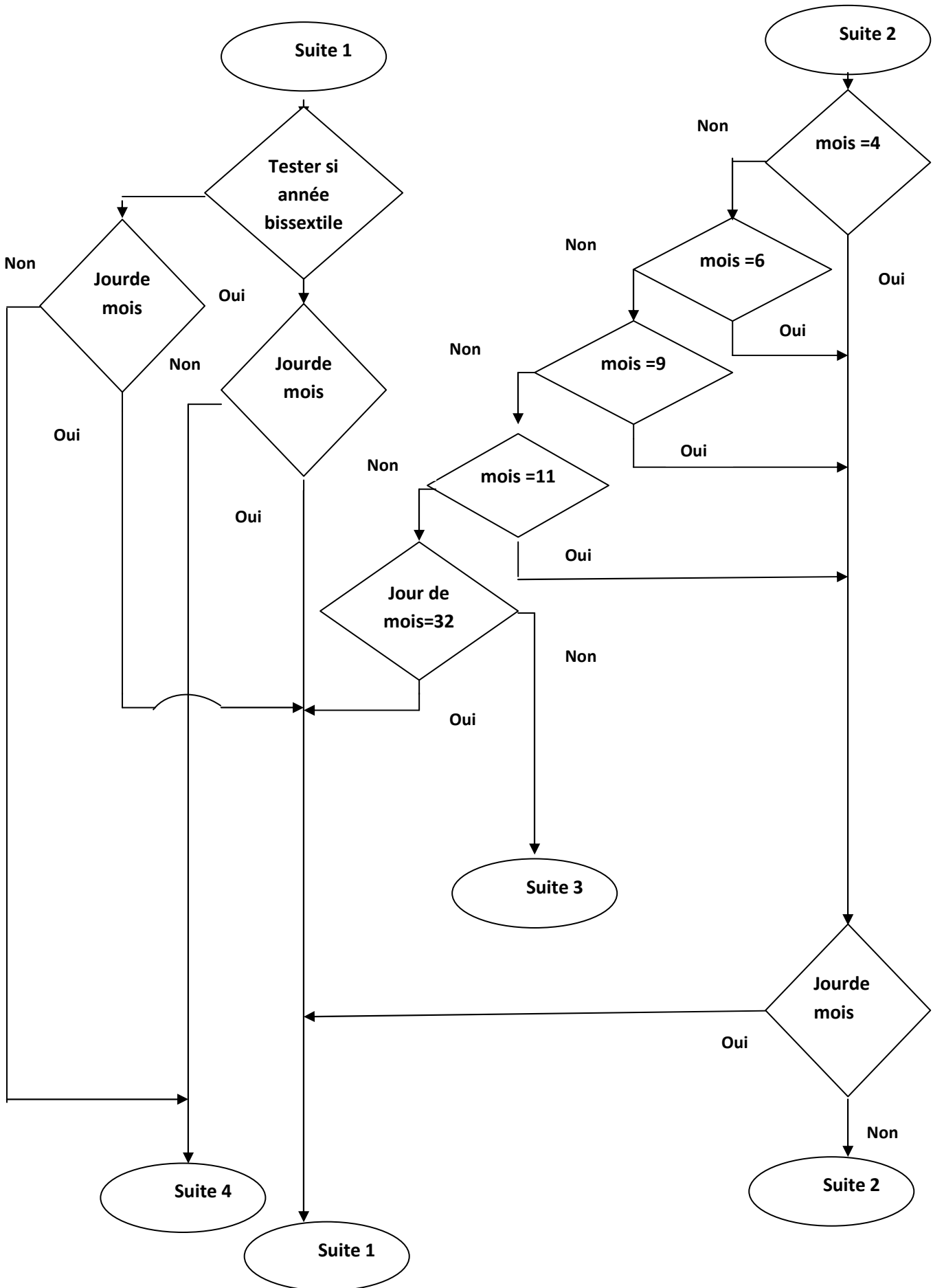
Fig. III.13. L'Organigramme de programme global .

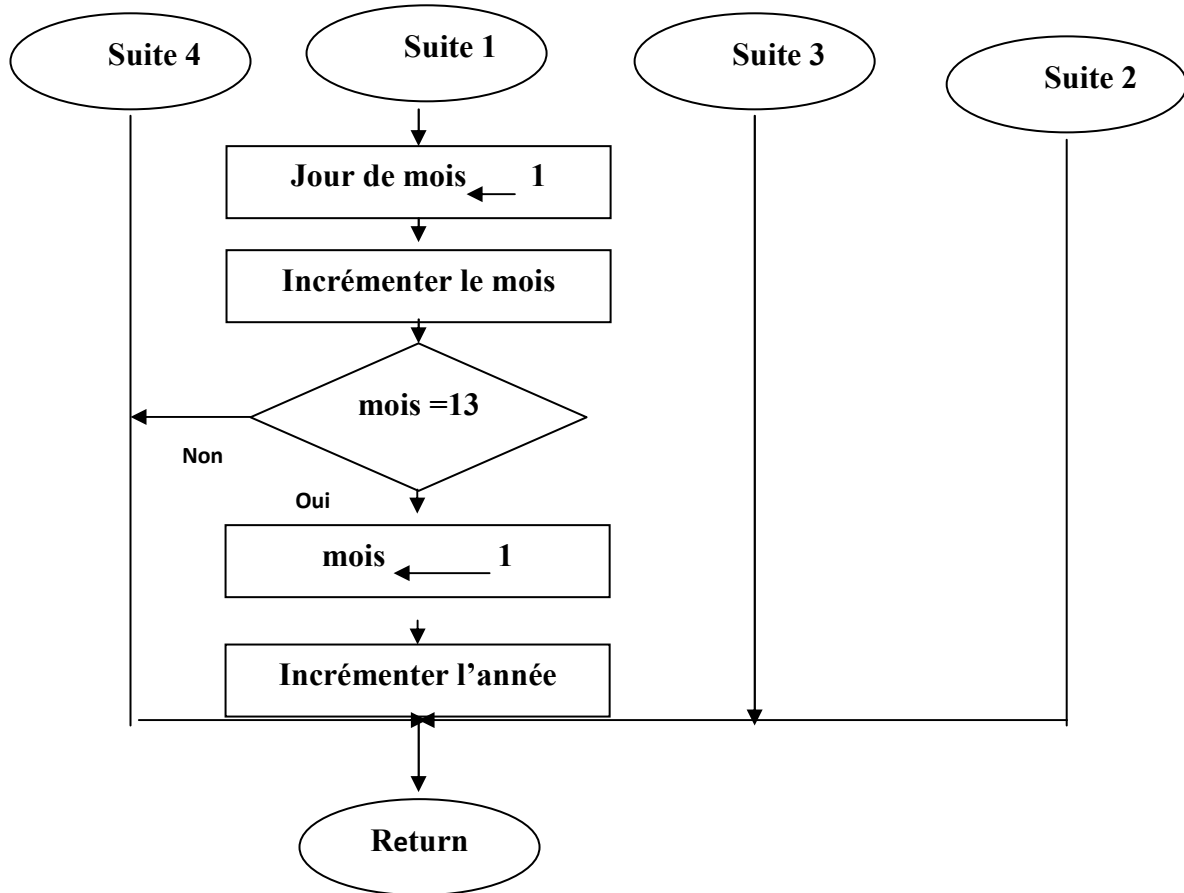
III.6. Explication de l'organigramme globale :

- Au début on déclare les variables :
Char sec ;
Char min ;
Char hour ;
Char day ;
Char date ;
Char month ;
Char year ;
Char data ;
Char time[8] ;
Char ddate[11] ;
- Ensuite on déclare le sous programmes :
void Read_ds1307 () ;
void Write_ds1307 () ;
void Display () ;
- Déclaration et configuration de Ds1307 ;
I2C_Init (100000) ;
I2C_Star () ;
I2C_Wr (0xD0) ;
I2C_Wr (0x00) ;
I2C_Wr (0x80) ;
I2C_Wr (0x17) ;
- Initialisation des portes :
TRISD=0x00 ; RD2, RD3, RD4, RD5, RD6, RD7 sont reliée au LCD.
TRISB=0x07 ; RB0 et RB2 sont reliée au deux Botton poussoir.
- Calcule les temps des prières :
Les équations de prière sont programmées en MikroC.
- Le teste :
Teste si le temps actuel est le temps de prière appeler el-adhan.

III.7. Organigramme de Real time clock:







III.8.Conclusion :

Dans ce chapitre on a constater le rôle de la réalisation software, et le rôle essentiel de logiciel « mikroC » pour programmer le microcontrôleur 16F877 , et alors en peut maintenant passer a la réalisation pratique.

Chapitre 4

Réalisation

Hardware

IV. 1. Introduction :

Le processus d'élaboration du matériel a été conçu en utilisant un microcontrôleur PIC16F877 en tant que contrôleur pour contrôler tout le système D'horloge numérique et un circuit RTC (Real time clock) ,Puis le circuit de l'écran LCD 2x16 caractères qui est utilisée pour afficher le temps et la date actuel, et dans ce chapitre on va simulées ces circuit dans un logiciel appelée ISIS Professionnel ,

IV. 2. Réalisation avec logiciel proteus :

IV. 2.1.Présentation du logiciel :

Isis fournit un espace pour entrer une conception de circuit sur un emplacement, et pour la simulation temps réel interactive et de la gestion des ressources et le code objet associé à chaque projet. En outre, certains des objets graphiques peuvent être placés sur le schéma de simuler l'oscillation de fréquence classique et la variable mis en œuvre. La figure (IV. 2) montre la fenêtre d'interface utilisateur.

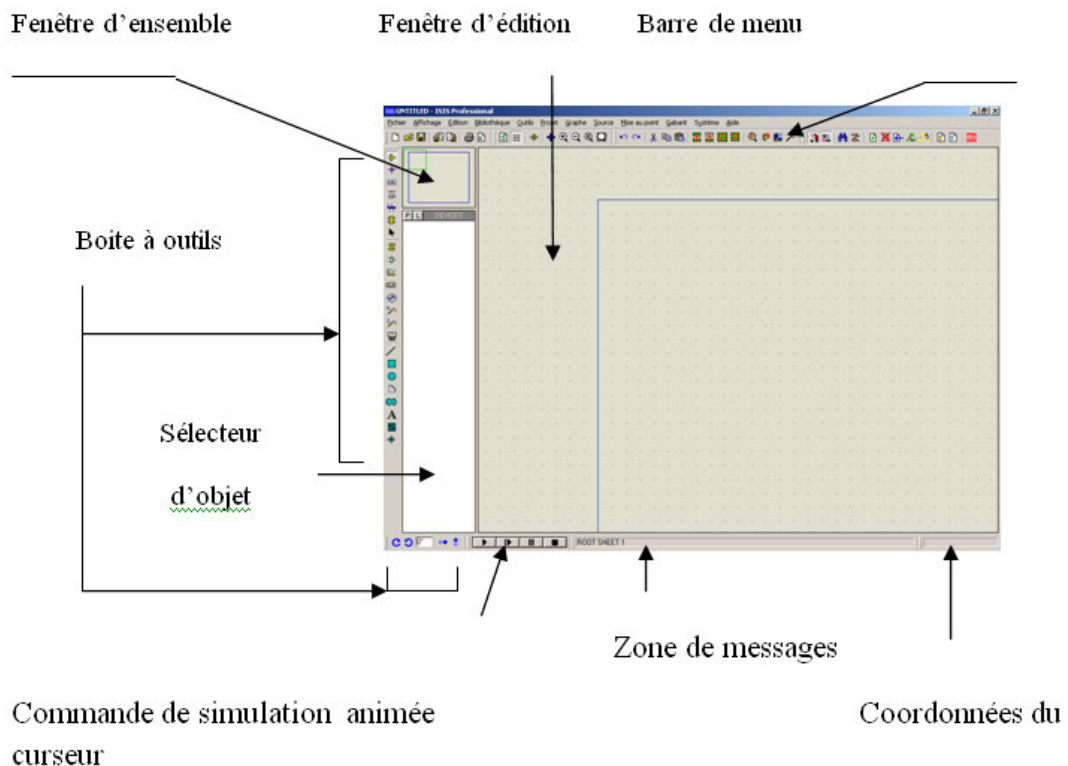


Fig. IV. 1. la fenêtre d'interface utilisateur.

IV. 2.2. Fenêtre d'ensemble :

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande '*Définir taille des feuilles*' du menu '*système*'.

Le cadre en vert délimite la zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

- Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Vous pouvez redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule '*shift*' du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche.

IV. 2.3. Fenêtre d'édition :

C'est dans cette fenêtre que vous éditez votre circuit. Elle ne représente que la partie du circuit qui est définie par le cadre vert dans la fenêtre d'ensemble.

- Vous pouvez redéfinir la zone travail à l'aide de la fenêtre d'ensemble ou en utilisant la commande '*Zoom*' du menu '*Affichage*'.

IV. 2.4. La boîte a outils :

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc...

IV. 2.5. Cordonnées du curseur :

Les coordonnées déterminent la position du curseur par rapport à l'origine qui par défaut se trouve au centre de la fenêtre d'édition.

Les coordonnées sont affichées en 1/1000 de pouce (th).

IV. 2.6. Raccourcis clavier DES FONCTIONS D'EDITION COURANTES :

Ajustement de la fenêtre de visualisation : placez le pointeur de la souris à l'endroit que vous voulez agrandir puis utilisez les touches F6 à F8.

Occultation de la grille touche G.

Ajustement du pas de grille : touches F2 à F4.

Définition d'une nouvelle origine : touche O.

Marquage d'un objet : cliquez sur l'objet à l'aide du bouton droit de la souris, il passe en sur-brillance.

Démarquage d'un objet : cliquez avec la bouton droit de la souris dans une zone vierge de la feuille.

Annuler la dernière commande : Ctrl Z.

Rétablir la dernière commande : Ctrl Y.

Effacement d'un objet : cliquez 2 fois avec le bouton droit de la souris.

IV. 2.7. L'organisation de la boîte d'outils :

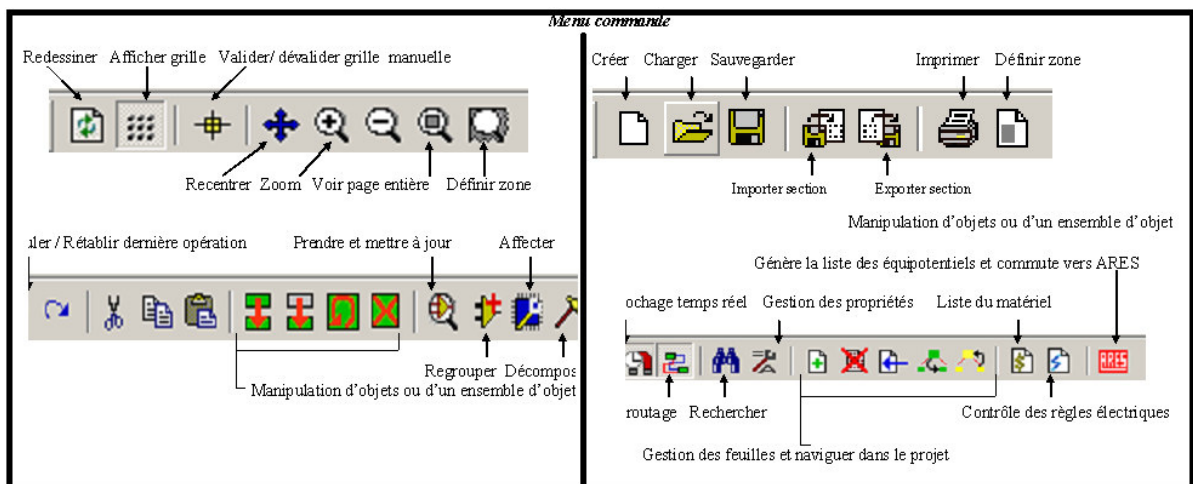


Fig. IV. 2. l'organisation de la boîte d'outils(1).

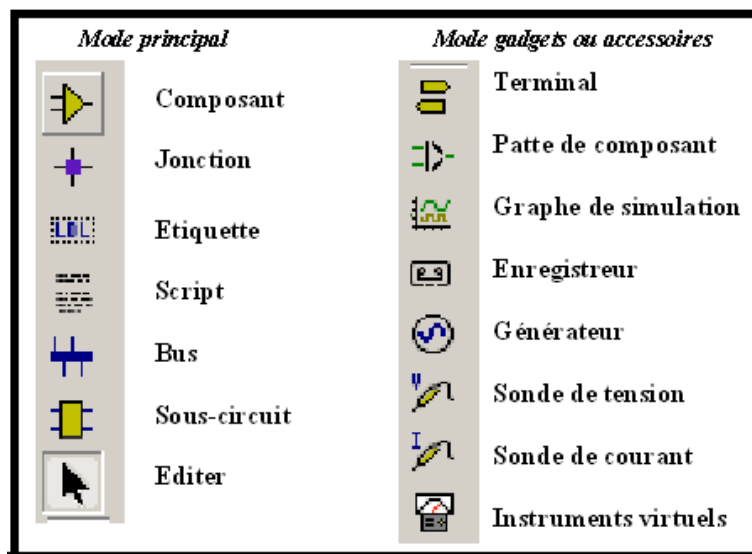


Fig. IV. 3. l'organisation de la boîte d'outils(2).

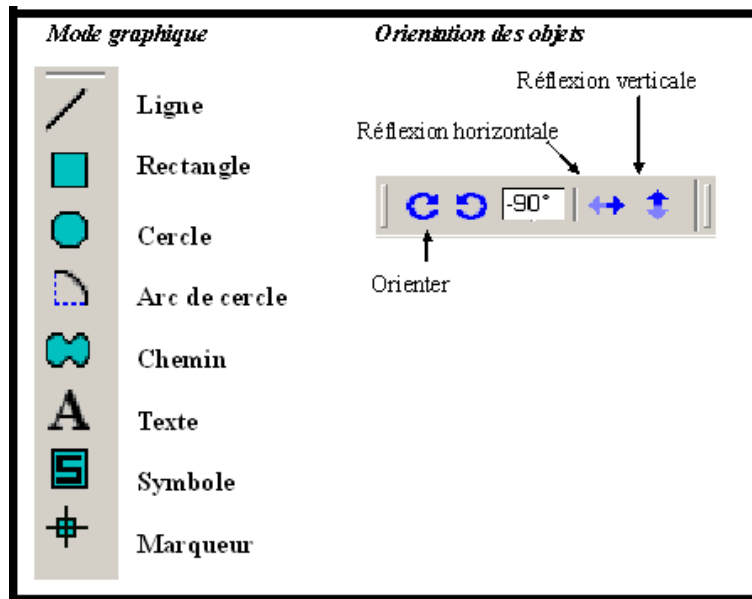


Fig. IV. 4. l'organisation de la boîte d'outils(3).

IV. 3. Schéma électronique de circuit :

IV. 3.1. Simulation par "proteus 7" :

On a appliqué ce programme au microcontrôleur pour utiliser un programme de simulation qui est "proteus 7". Voir la figure (IV. 5).

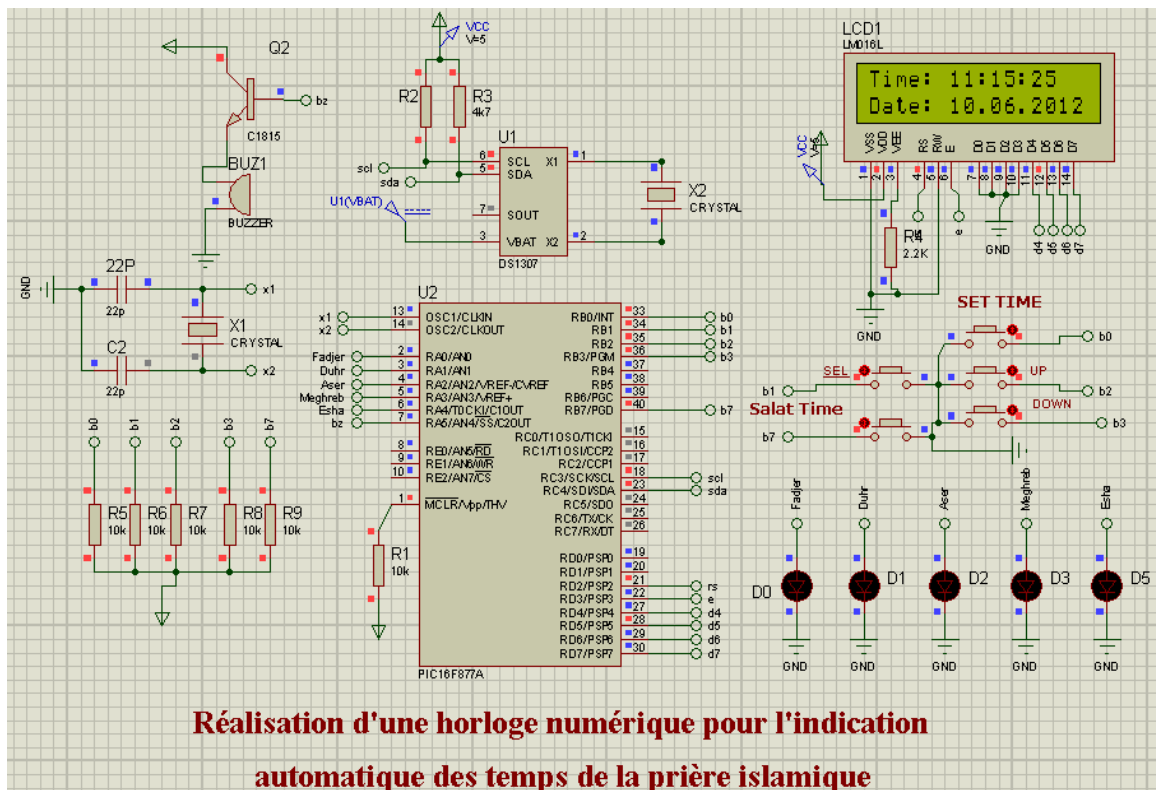


Fig. IV. 5. Simulation par ISIS Professionnel.

IV. 3.2. Réalisation par matériel :

IV. 3.2.a. Une carte d'alimentation :

Notre carte d'alimentation contient deux batteries couplée en série Pour obtenir une valeur de tension égale à 5V, on utilise régulateur 7805.

IV. 3.2.b. Carte de commande :

Cette carte est basée sur le microcontrôleur (PIC16F877) .

IV. 3.2.c. Carte de circuit RTC :

Cette carte est basée sur l'eprom DS1307.

IV. 3.2.d. Circuit d'affichage LCD :

C'est un afficheur LCD de deux ligne chaque ligne ce compose de 16 caractères.

IV.4. définition de DS1307 :

Le DS1307 série Real-Time Clock est une faible puissance, pleine décimal codé binaire (BCD) horloge / calendrier, ainsi que 56 octets de SRAM NV. L'adresse et les données sont transférées en série a travers I2C, avec bus bidirectionnel. L'horloge / calendrier fournit des secondes, minutes, heures, jour, date, mois, et de l'information par an.

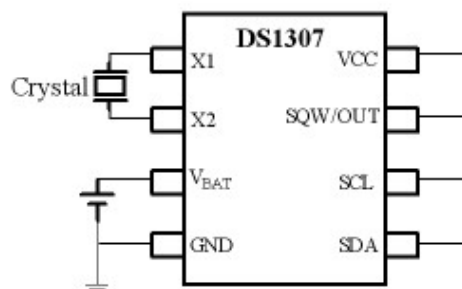


Fig. IV-10 : Brochage de DS1307.

IV.5. Le brochage de PIC 16F877 avec DS1307 :

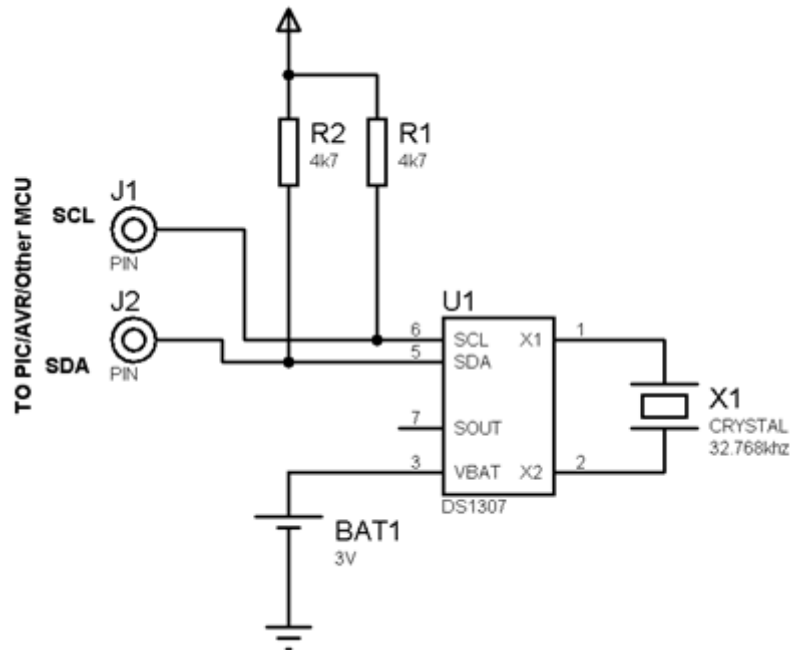


Fig. IV-11 : Le brochage de PIC 16F877 avec DS1307.

IV.6. Introduction au bus I2C :

Avant de parler du module MSSP en mode I2C du PIC, introduisons très brièvement Le standard I2C Le bus I²C permet de faire communiquer entre eux des composants Électroniques très divers grâce à seulement 3 fils : Un signal de donnée (SDA), un signal d'horloge (SCL), et un signal de référence électrique (Masse). Comme les lignes SDA et SCK sont utilisées dans les deux sens par les deux circuits qui communiquent, on peut avoir un circuit qui place la ligne à 1 (Vcc) et l'autre qui la place à 0 (masse) ce qui correspond à un court circuit qui peut détruire les deux composants. Pour éviter ce problème, les E/S SDA et SCK fonctionnent en mode collecteur ouvert (ou drain ouvert) de sorte qu'un circuit ne peut imposer que le niveau bas ou ouvrir la ligne, le niveau haut est obtenu par une résistance de tirage externe. Ainsi une ligne est à 0 quand un des deux circuits impose le 0. Elle passe à 1

quand les deux circuits imposent le 1 (circuit ouvert). Le protocole I2C jongle avec cette situation pour organiser l'échange des données entre les deux composants Pic 16F877 et DS1307.

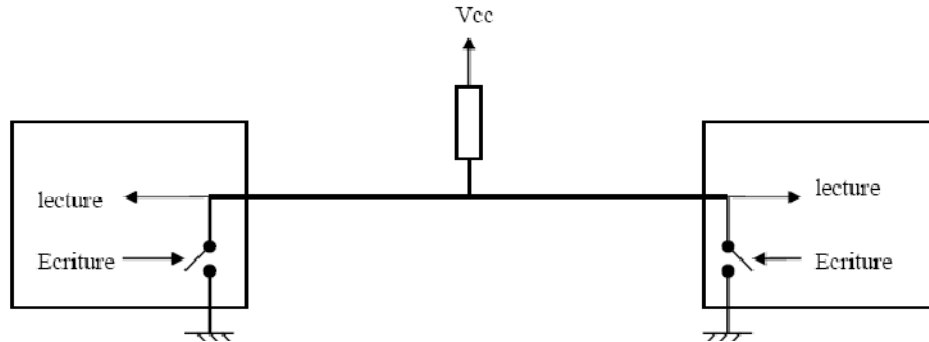


Fig. IV-12 : Liaison I2C -1-

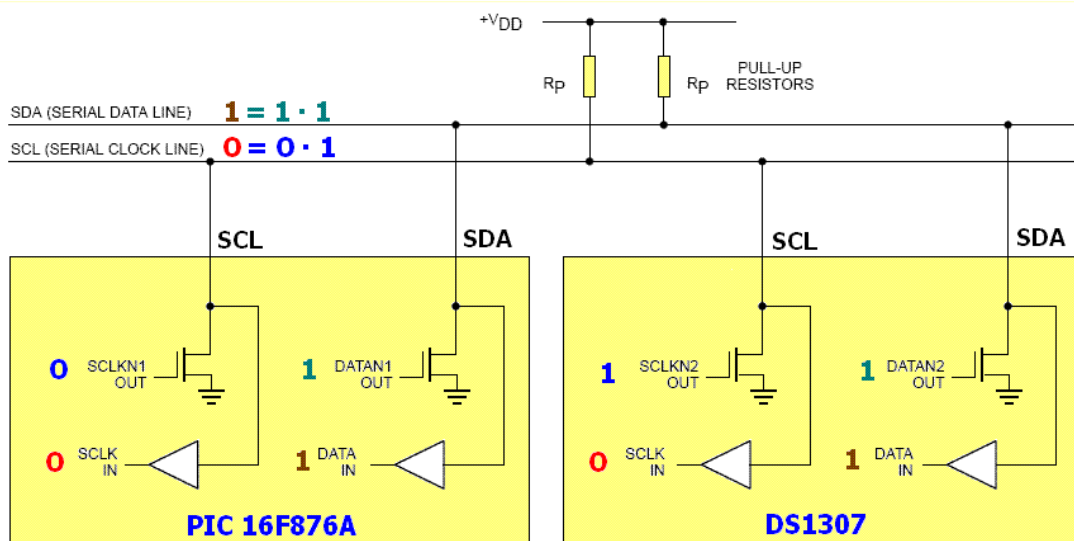


Fig. IV-13 : Liaison I2C -2-

Un bus I2C peut être relié à plusieurs circuits, mais pendant une communication, un des circuits est le maître, c'est lui génère l'horloge et initie les séquences de transmission, l'autre est l'esclave, il subit l'horloge du maître sur la ligne SCK mais il peut tout de même recevoir et émettre des données sur la ligne SDA. Chaque esclave a une adresse, au début d'une séquence de communication, le maître qui initie la séquence envoie l'adresse du slave avec lequel il désire communiquer, celui-ci reconnaît son adresse et répond, les autres slaves (s'il y en a) restent muets.

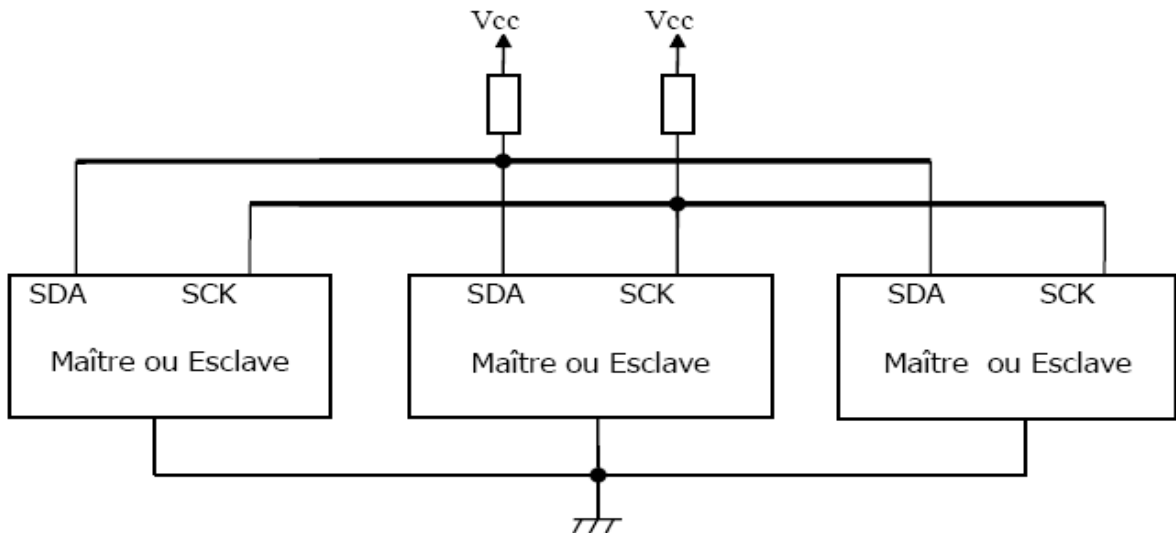


Fig. IV-14 : Liaisons I2C a plusieurs circuits.

IV.6.1. Exemple de communication entre un maître (PIC 16F877) et un esclave :

Le circuit Dallas DS1307 est une horloge temps réel (Real Time Clock), qui fournit secondes, minutes, heures, jours, dates, mois et années , les années bissextiles sont prises en compte (jusqu'en 2100).

Le DS1307 travaille dans le mode standard (fréquence d'horloge f SCL de 100 kHz).

L'adresse I2C (7 bits) du DS1307 est : 1 1 0 1 0 0 0 (adresse fixée par le constructeur => adresse non modifiable).

IV.6.2. Exemple d'écriture du DS1307 :

L'émetteur est le maître et le récepteur est l'esclave.

Le registre d'adresse 0x04 du DS1307 contient la date (cf. datasheet du DS1307). Pour régler le calendrier au 27 du mois, il faut écrire la donnée 0x27 (codage BCD) dans le registre d'adresse 0x04 du DS1307.

Le bus I2C utilise le protocole suivant :

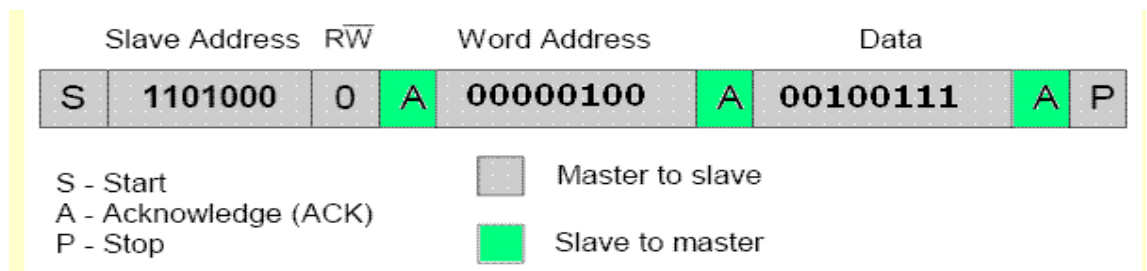


Fig. IV-15 : Le protocole I2C.

- 1) Pour initier le dialogue, le maître crée une condition **Start** .
- 2) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 0 (bit Write).
- 3) L'esclave répond (accusé de réception : bit Acknowledge) .
- 4) Le maître envoie l'adresse du registre (0x04) à écrire.
- 5) L'esclave répond (accusé de réception : bit Acknowledge) .
- 6) Le maître envoie la donnée (0x27) à écrire.
- 7) L'esclave écrit la donnée puis envoie un accusé de réception (bit Acknowledge) .
- 8) Le maître termine le dialogue avec une condition **Stop** .

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).

Un peu moins évident, le chronogramme correspondant des lignes SDA et SCL :

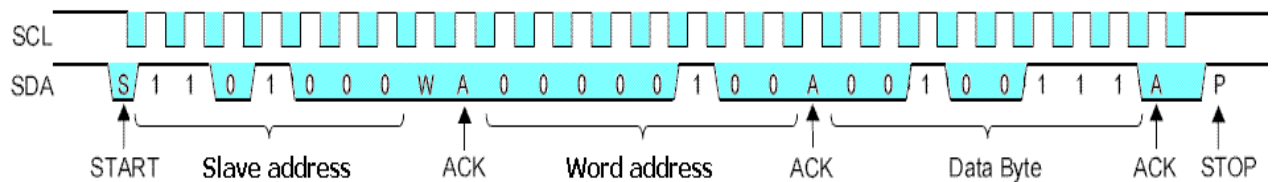


Fig. IV-16 : le chronogramme des lignes SDA et SCL

Plus précisément, la **condition Start** est définie par la transition de la ligne SDA du niveau haut au niveau bas, SCL étant au niveau haut.

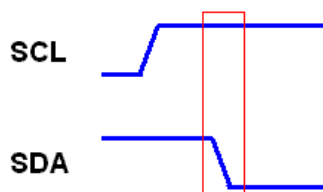
C'est bien sûr le maître qui force la ligne SDA au niveau bas :

Start condition



C'est exactement la même chose avec la condition **Repeated Start** :

Repeated Start



La condition Stop est définie par la transition de la ligne SDA du niveau bas au niveau haut, SCL étant au niveau haut.
C'est bien sûr le maître qui agit sur la ligne SDA :

Stop condition



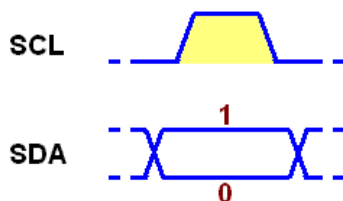
En dehors de ces trois conditions, la ligne SDA ne peut changer de niveau que lorsque SCL est au niveau bas :

-Transmission d'un bit de donnée .

La donnée présente sur la ligne SDA doit être stable sur l'intervalle de temps où SCL est au niveau haut.

C'est l'émetteur qui agit sur la ligne SDA, et c'est évidemment le maître qui génère le signal d'horloge SCL.

bit de donnée



- Les données sont transmises octet par octet (1 octet = 8 bits) .

- **Bit Acknowledge**

Le protocole est le même que pour la transmission d'un bit de donnée.

2 cas :

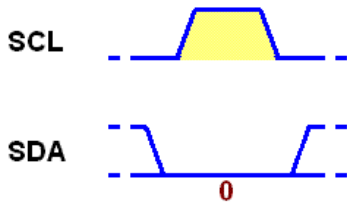
- bit Acknowledge (master to slave) .

Le bit Acknowledge est généré par le maître (le maître force la ligne SDA au niveau 0).

- bit Acknowledge (slave to master).

Le bit Acknowledge est généré par un esclave (l'esclave force la ligne SDA au niveau 0).

bit Acknowledge

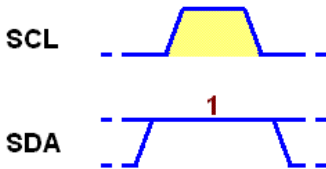


- Bit Not Acknowledge .

Le protocole est le même que pour la transmission d'un bit de donnée.

Le bit Not Acknowledge est toujours généré par le maître (le maître ramène la ligne SDA au niveau 1).

bit Not Acknowledge

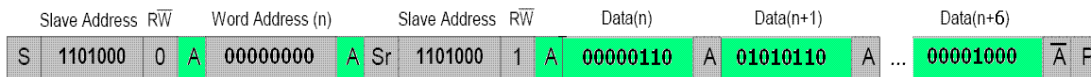


Vous remarquerez qu'il faut 9 bits pour transmettre un octet (8 bits de l'octet et le bit ACK).

IV.6.3 . Exemple de lecture du DS1307 :

L'émetteur est l'esclave et le récepteur est le maître.

Les registres d'adresses 0x00 à 0x06 du DS1307 contiennent respectivement les secondes, minutes, heures, jours, dates, mois et années (cf. datasheet du DS1307). Voici comment lire, d'une seule traite, le contenu des registres d'adresses 0x00 à 0x06 du DS1307 :



S - Start
 Sr - Repeated Start
 A - Acknowledge (ACK)
 P - Stop
 \overline{A} - Not Acknowledge (NACK)



 Master to slave
 Slave to master

Fig. IV-17 : Les registres d'adresses 0x00 à 0x06 du DS1307.

Chronogramme correspondant :

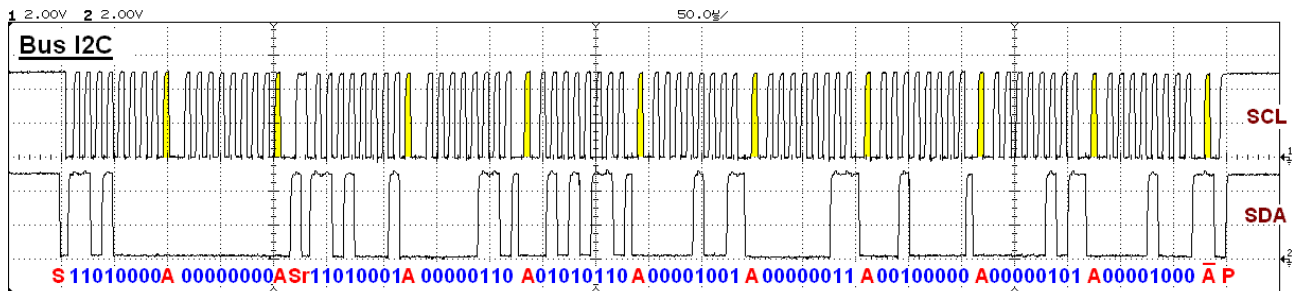


Fig. IV-18 : Chronogramme du Bus I2C

- 1) Pour initier le dialogue, le maître crée une condition **Start**
- 2) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 0 (bit Write)
- 3) L'esclave répond (accusé de réception : bit Acknowledge)
- 4) Le maître envoie l'adresse du registre (0x00) à lire
- 5) L'esclave répond (accusé de réception : bit Acknowledge)
- 6) Le maître émet une condition **Repeated Start**
- 7) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 1 (**bit Read**)
- 8) L'esclave répond (accusé de réception : bit Acknowledge)
- 9) L'esclave envoie le contenu du registre d'adresse 0x00 au maître
- 10) **Le maître répond** (accusé de réception : bit Acknowledge)
- 11) L'esclave envoie le contenu du registre d'adresse 0x01 (automatiquement incrémenté) au maître
- 12) Le maître répond (accusé de réception : bit Acknowledge)
- 13) L'esclave envoie le contenu du registre d'adresse 0x02 (automatiquement incrémenté) au maître
- 14) Le maître répond (accusé de réception : bit Acknowledge)
- ...
- 21) L'esclave envoie le contenu du registre d'adresse 0x06 (automatiquement incrémenté) au maître
- 22) Le maître répond (accusé de réception : bit **Not Acknowledge**)
- 23) Le maître termine le dialogue avec une condition Stop .

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).
 Le contenu du registre d'adresse 0x00 du DS1307 est 0x06 (codage BCD : 06 secondes).
 Le contenu du registre d'adresse 0x01 est 0x56 (c'est-à-dire 56 minutes).
 Le contenu du registre d'adresse 0x02 est 0x09 (c'est-à-dire 09 heures).
 Le contenu du registre d'adresse 0x03 est 0x03 (c'est-à-dire Mardi).
 Le contenu du registre d'adresse 0x04 est 0x20 (c'est-à-dire 20 ème jour du mois).
 Le contenu du registre d'adresse 0x05 est 0x05 (c'est-à-dire mois de mai).
 Le contenu du registre d'adresse 0x06 est 0x08 (c'est-à-dire année 2008).
 (Mardi 20 mai 2008, 9 heures 56 minutes et 6 secondes).

IV. 7. Définition de LCD :

2x16 caractères LCD est le dispositif de message le plus commun à afficher une quelconque alphabet et des caractères numériques. En utilisant le SK40C, l'utilisateur est en mesure de compléter le circuit pour ce projet facilement.

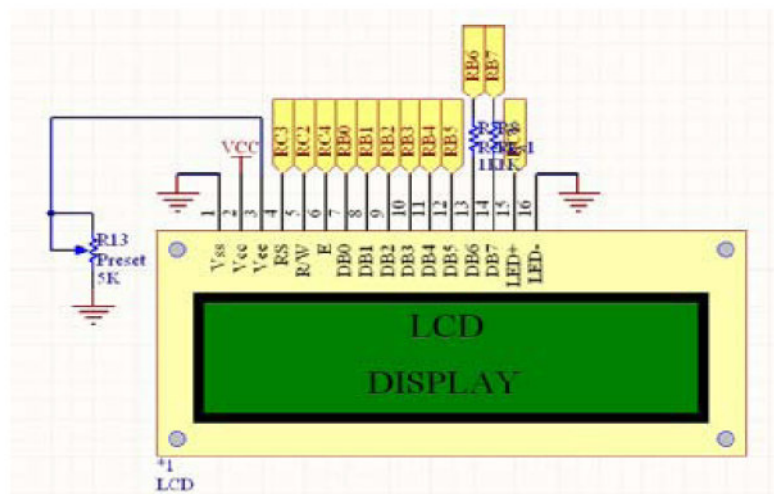


Fig. IV-18. l'LCD 2x16.

L'utilisateur peut contrôler l'écran LCD en utilisant soit à 8 broches (8 - Interface bits) ou 4 broches (4 -bits d'interface), en fonction des repères d'E / S utilisateurs. Pour 16F877 qui a 33 I / O broches, il devrait y avoir aucun problème en utilisant l'interface 8-bit qui est beaucoup plus facile. Les broches 15 et 16 est pour le rétro-éclairage LCD qui est facultatif pour vous si vous voulez que la lumière à être sous tension. Pour info, activer le rétro-éclairage de l'écran LCD fera de votre batterie à sécher rapidement. La broche 3eme (VEE)

doit être connecté à une résistance variable pour contrôler la Contraste de l'écran, tandis que la broche R / W (lecture ou écriture) est reliée à la masse. R / S et E sont connectés à RD2 et RD3 à votre PIC, respectivement. Pour broches de communication de données, D0-D7 (8 - bits) est relié directement à RB0-RB7 de ton PIC [10].

Avant d'afficher une l'alphabet numérique, certains initialise de l'écran LCD est nécessaire. Tableau LCD IV.1 est le protocole / fonctions qui est besoin de lancer avant de l'utiliser à cristaux liquides et le tableau IV.2 est Fonction LCD bit de commande et de la valeur. Les utilisateurs ont besoin des deux tables pour initier LCD afficheur.

Tableau IV.1: Parallèle LCD protocole.

R/S	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Functions
4	14	13	12	11	10	9	8	7	Pin count at LCD
0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	1	X	Return Cursor to Home Position
0	0	0	0	0	0	1	IL	S	Set Cursor Move Direction
0	0	0	0	0	1	D	C	E	Enable Display/Cursor
0	0	0	0	1	SC	RL	X	X	Move Cursor/Shift Display
0	0	0	1	DL	N	F	X	X	Set Interface Length
0	1	A	A	A	A	A	A	A	Move Cursor to DDRAM
1	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position

X = Don't care
A = Address

Tableau IV.2: broche Fonction de bit de commande et de la valeur.

Bit Name	Bit Function	Bit = Low (0)	Bit = High (1)
R/S	Command or Character	Command	Character
ID	Set cursor move direction	Decrement cursor position	Increment cursor position
S	Specifies to shift the display	No display shift	Display shift
D	Sets On/Off of all display	Display off	Display on
C	Cursor On/Off	Cursor off	Cursor on
B	Cursor blinks	Cursor blink off	Cursor blink on
SC	Sets cursor move or display shift	Move cursor	Shift Display
RL	Shift direction	Shift left	Shift right
DL	Sets interface data length	4-bit interface	8-bit interface
N	Number of display line	1 line	2 lines
F	Character font	5x7 dots	5x10 dots

IV. 8.Présentation de ARES :

L'aspect Général de ARES est très approchant de celui de ISIS.

1. Une barre de menu.
2. Une fenêtre principale dans laquelle vous allez créer votre routage mais aussi vos nouveaux composants
3. Une fenêtre d'aperçu en haut à droite
4. Une palette en dessous et un sélecteur de boîtiers..
5. Un sélecteur de surface active, en bas à gauche.

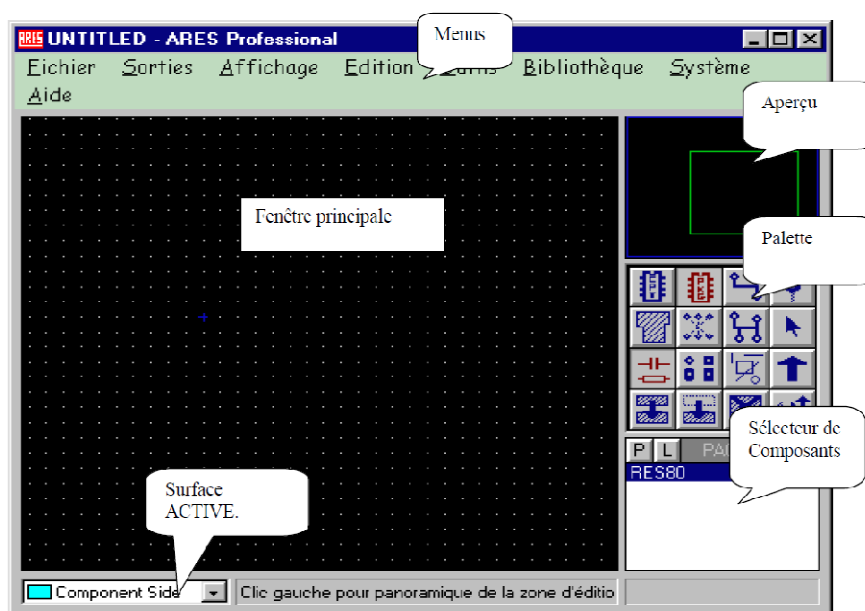


Fig. IV-18.Fenêtre principale de ARES.

IV. 8.1.Création d'une carte :

La première chose à faire avant de commencer un routage est de définir une carte aux dimensions de votre projet. Il est fortement conseillé de réaliser, si possible, votre projet sur des formats standards (Demi Europe 100x80, Europe 100x160).

Pour cela placez vous en mode graphique et sélectionnez "Board Edge" dans le sélecteur de surfaces. (Voir Présentation de ARES).

Sélectionnez l'icône rectangle et dessinez un rectangle correspondant au contour de votre carte. (en jaune si vous n'avez pas modifié les préférences).

IV. 8.2.Création d'un boîtier :

Tous les types de boîtiers sont dans les bibliothèques (sic), mais comme vous avez toujours des besoins très particuliers il est utile de pouvoir créer ses propres boîtiers. La procédure est la même que pour la création de symboles dans ISIS. Vous dessinez votre boîtier en mode graphique, vous lui attribuez des pastilles (Mode pastille) que vous numérotez (clic droit clic gauche sur la pastille). Lorsque tout est à votre convenance vous entourez votre création de manière à sélectionner tout les éléments.

IV. 8.3.Créer un composant :

Cette fonction n'est utile que lorsque vous réalisez un routage sans le schéma de ISIS. Vous devez vous placer dans le mode composant et cliquer sur le bouton de création "C". Nous avons parfois tendance à confondre le composant et son boîtier.

IV. 8.4.Le placement des composants :

IV. 8.4.a. Placement automatique :

Après avoir défini une carte vous pouvez placer les composants en mode automatique.

Outils > Placement automatique

La fenêtre ci-contre apparaît.

Vous pouvez choisir les composants à placer. Après validation les composants sont positionnés sur la carte.

IV. 8.4.a.Placement Manuel :

On peut rarement se contenter d'un placement automatique. En raison d'exigences dues au projet il est très souvent nécessaire de placer des composants à des endroits précis. Dans ce cas vous sélectionnez le composant dans la fenêtre *de sélecteur*. Vous orientez votre sélection au moyen des icônes appropriés. Vous cliquez dans la fenêtre de placement et sans relâcher le bouton de la souris vous déplacez votre fantôme de composant.

IV. 8.5.Le routage :

IV. 8.5.a. Routage automatique :

Le routage automatique se lance dans le menu *Outils > Routeur automatique*

Dans la fenêtre de configuration vous pouvez choisir de router tout le chevelu, ou une partie. Vous pouvez également déterminer les isolations à respecter.

IV. 8.5.b. Routage Manuel :

Comme pour le placement automatique il est très rare qu'un routage automatique convienne sans modifications. Il est donc très utile de pouvoir modifier ou créer des pistes manuellement.

Choisissez l'icône *pistes* puis sélectionnez un guide (en vert). Par des clics successifs dessinez votre piste en partant d'une extrémité du guide. Le tracé se termine lorsque vous atteignez l'autre extrémité.

IV. 9. Conclusion :

La partie pratique est très importante dans notre projet, est dans ce chapitre en résumé que l'horloge est basée sur le microcontrôleur comme un maître et le Ds1307 comme un esclave et la relation entre ces deux derniers est définie dans un programme écrit en langage « mikroC ».

Conclusion générale

Ce mémoire traite le développement d'une horloge numérique pour l'indication du temps de la prière en utilisant un microcontrôleur PIC 16F877 et un logiciel de programmation des PIC « mikroC ».

L'horloge numérique qui indique le temps de la prière a été développée. L'objectif de ce projet est créé un dispositif fiable, pour aider les gens à faire leur prière à l'heure.

Ce projet fournit un moyen simple et efficace pour trouver automatiquement les temps de la prière, dispose également d'une fonctionnalité non disponible dans La plupart des horloges similaires, il peut être utilisé partout.

L'étude faite en vue de la réalisation de ce travail ainsi que la réalisation pratique du montage nous a permis de mettre en application les notions électronique requises tout au long de la formation .Des notions ont été approfondies par l'apport de connaissances nouvelles aussi bien dans le domaine Hardware et Software.

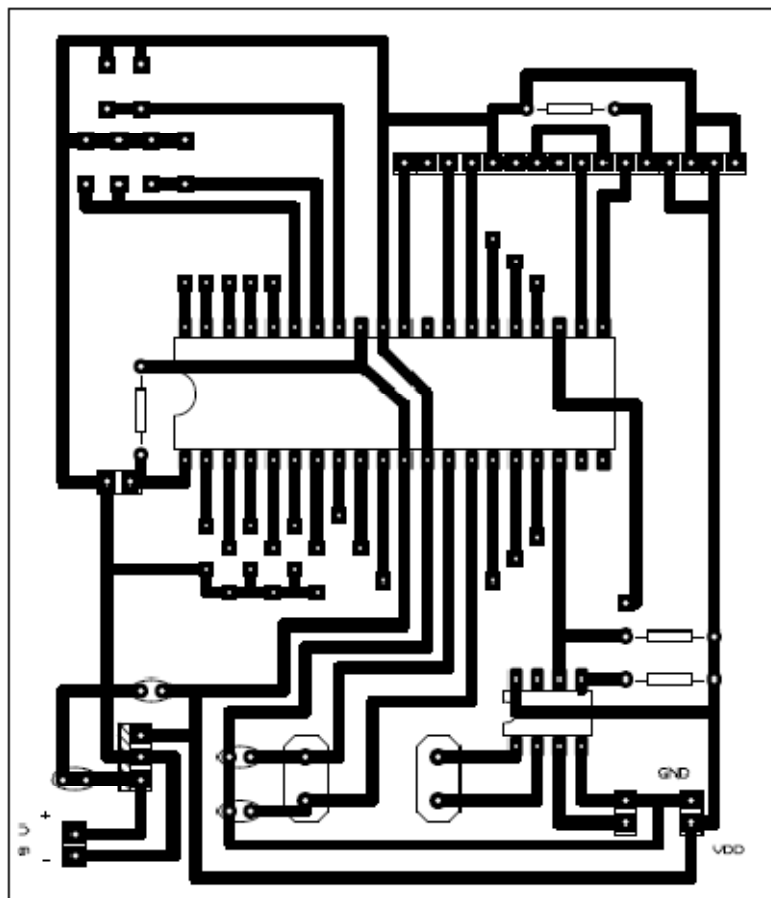
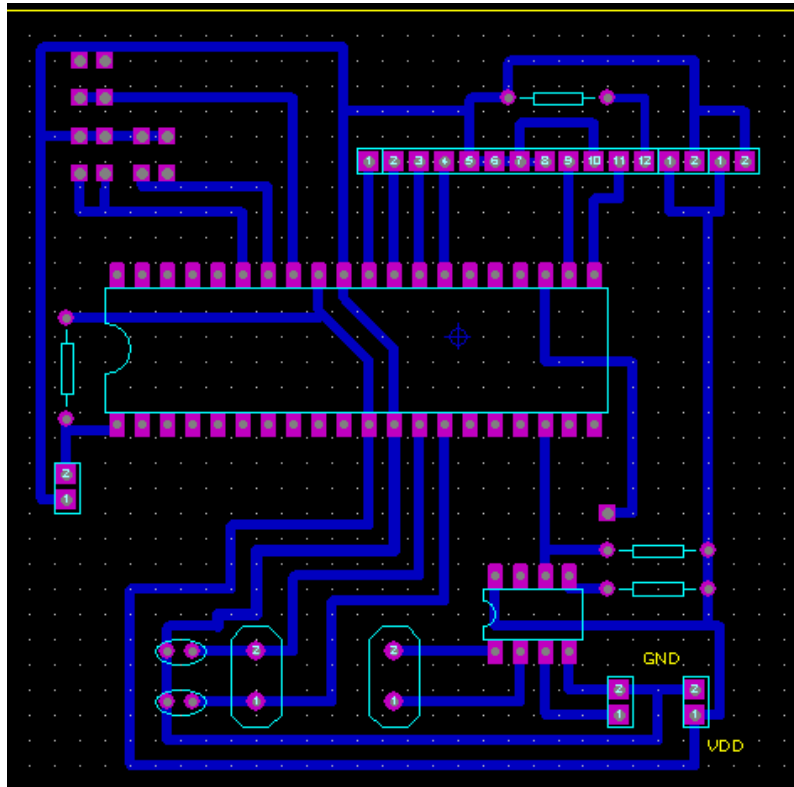
Nous estimons que les objectifs visés dans ce projet sont atteints, et nous espérant que ce travail fera l'objet d'autres améliorations par les étudiants des prochaines années.

Ce dispositif peut être amélioré on ajoutant par exemple un indicateur de la direction du Qibla, ou Global Positioning System (GPS) pour déterminer automatiquement la latitude et la longitude.

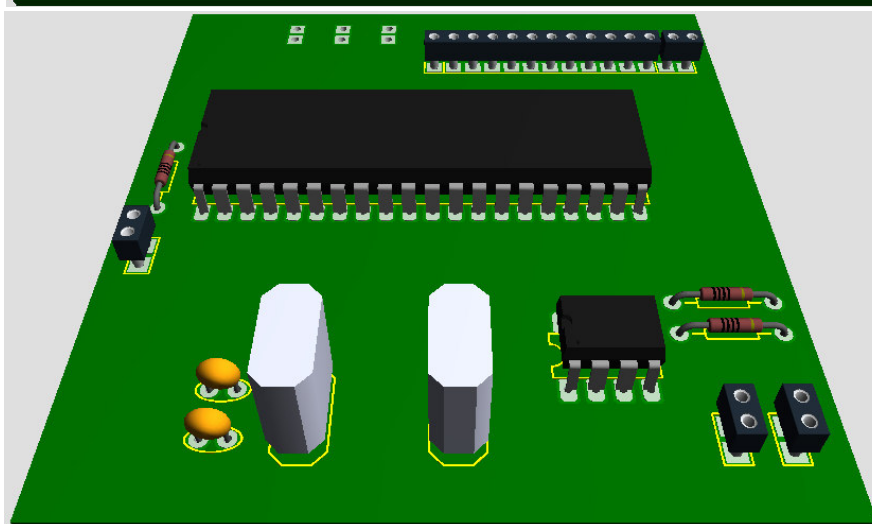
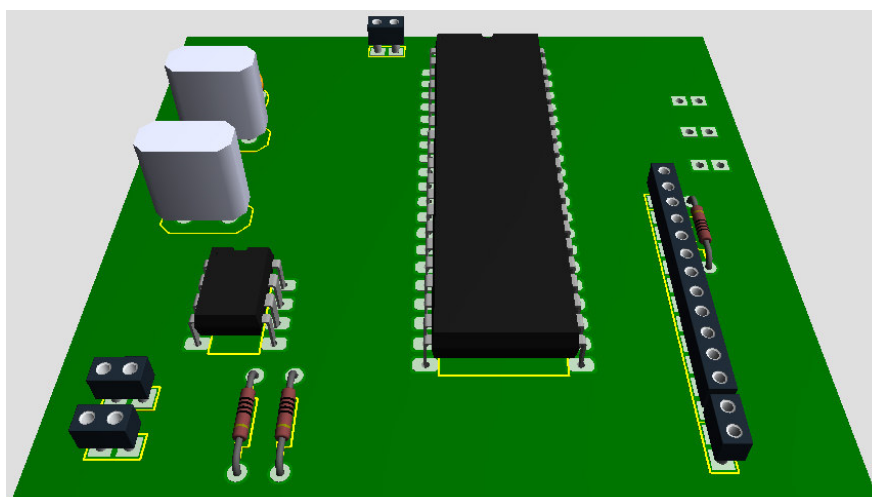
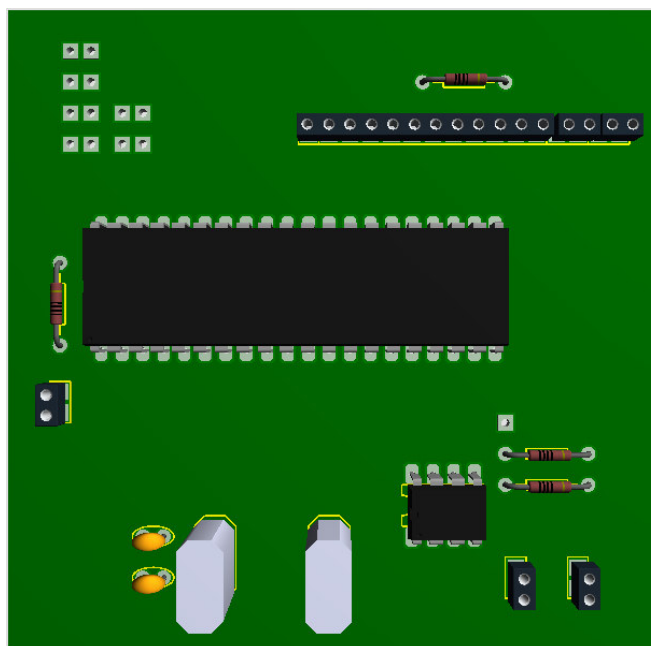


Annexe

1) Circuit imprimé de la carte de commande :



2) la carte de commande :



Bibliographie:

- [1] Droit musulman»Les actes cultuels»la prière»Les heures de prière,
www.islamqa.info/en.com
- [2] ABDELKARIM BOURAS. Etude et conception d'un convertisseur solaire pour la production d'eau chaude sanitaire. Mémoire fin d'étude. 2007.
- [3] Educorsica. Latitude Et Longitude D'un Lieu. <http://educorsica.fr/>
- [4] Les temps de la prière, www.Aslama.com.
- [5] Qatif Astronomy Society. Astronomical Applications in Islam ' la prière'
<http://qasweb.org/> .
- [6] DANIEL MENESPLIER. Microcontrôleurs PIC 16F876 et 16F877.
daniel.menesplier.free.fr/

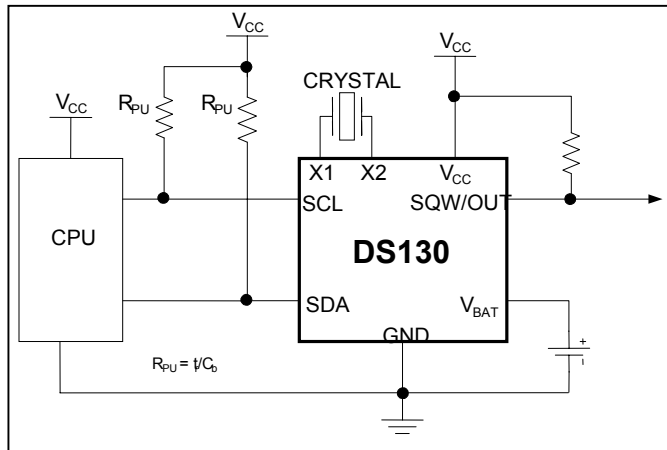
GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

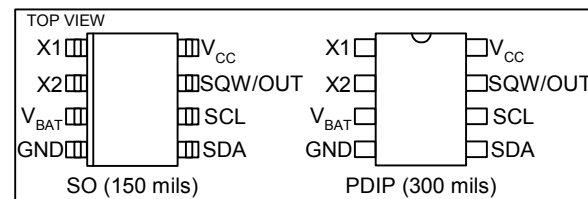
FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

TYPICAL OPERATING CIRCUIT



PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial.....	0°C to +70°C
Industrial	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (DIP, leads).....	+260°C for 10 seconds
Soldering Temperature (surface mount).....	Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

($T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		$V_{CC} + 0.3$	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 4.5\text{V}$ to 5.5V ; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	μA
Logic 0 Output ($I_{OL} = 5\text{mA}$)	V_{OL}				0.4	V
Active Supply Current ($f_{SCL} = 100\text{kHz}$)	I_{CCA}				1.5	mA
Standby Current	I_{CCS}	(Note 3)			200	μA
V_{BAT} Leakage Current	I_{BATLKG}			5	50	nA
Power-Fail Voltage ($V_{BAT} = 3.0\text{V}$)	V_{PF}		$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 0\text{V}$, $V_{BAT} = 3.0\text{V}$; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON); SQW/OUT OFF	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}			480	800	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t _{BUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HD:STA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SU:STA}		4.7			μs
Data Hold Time	t _{HD:DAT}		0			μs
Data Setup Time	t _{SU:DAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _R				1000	ns
Fall Time of Both SDA and SCL Signals	t _F				300	ns
Setup Time for STOP Condition	t _{SU:STO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{I/O}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{IH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{HD:DAT} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

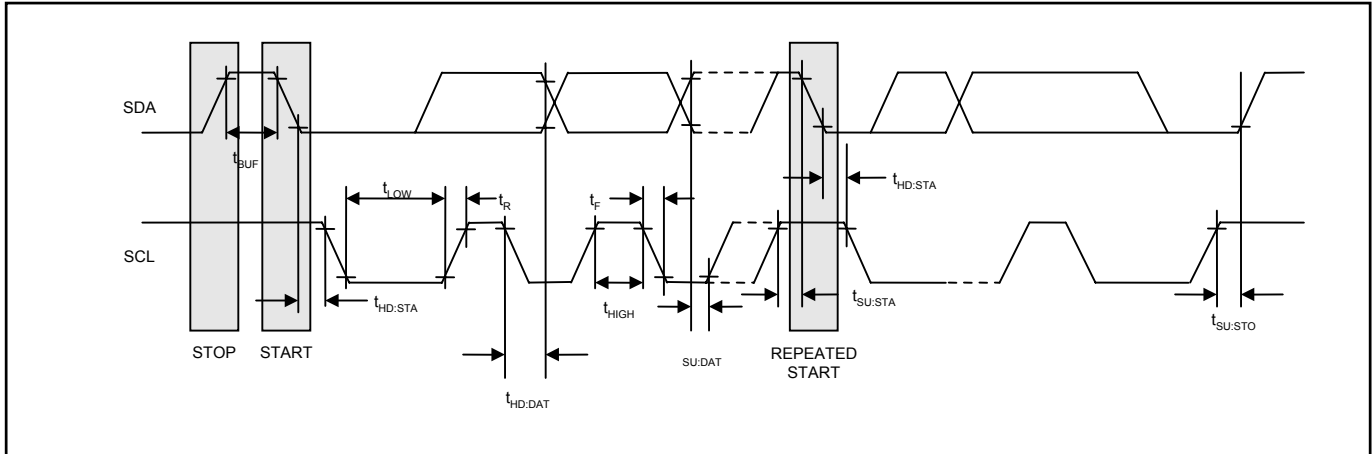
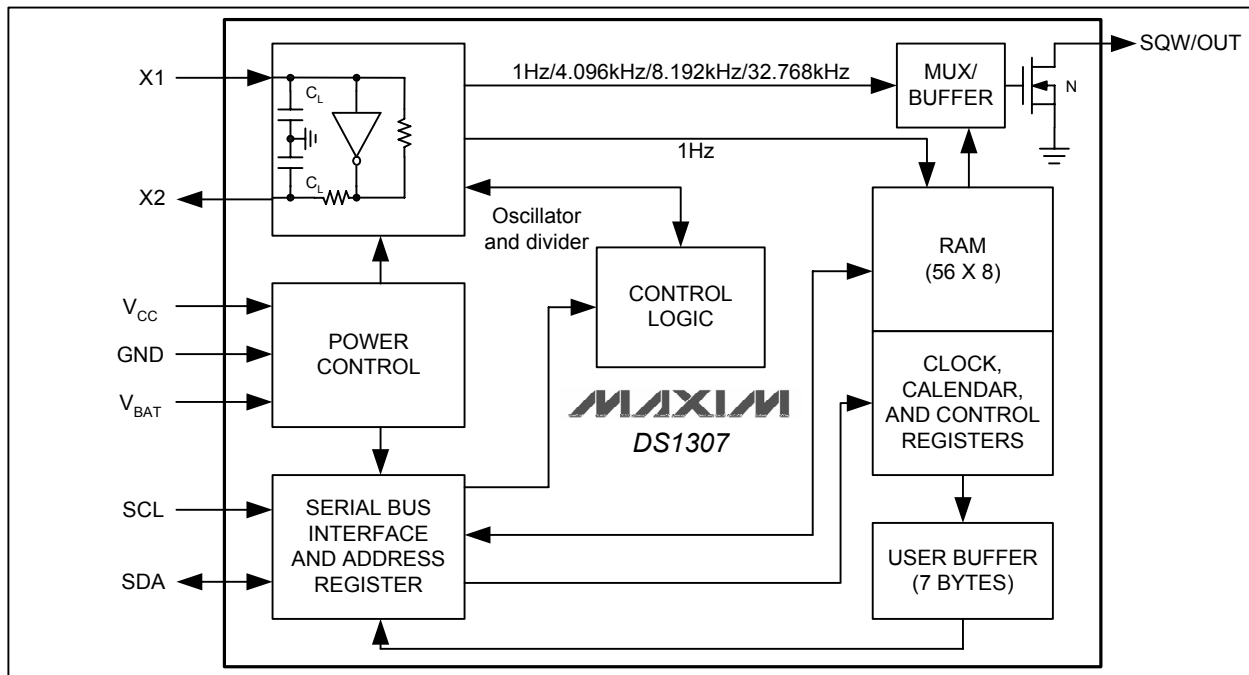
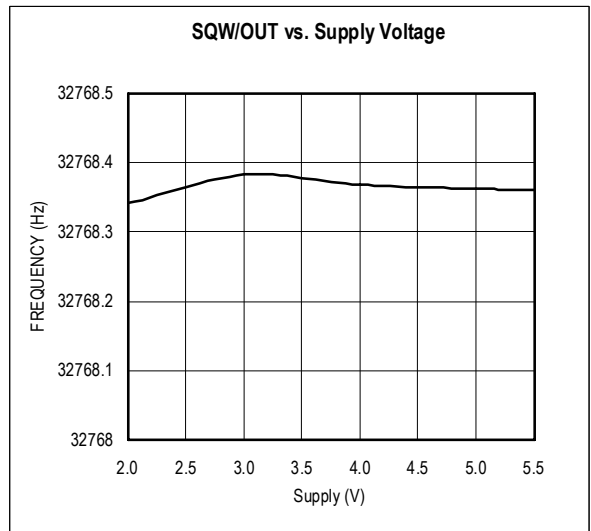
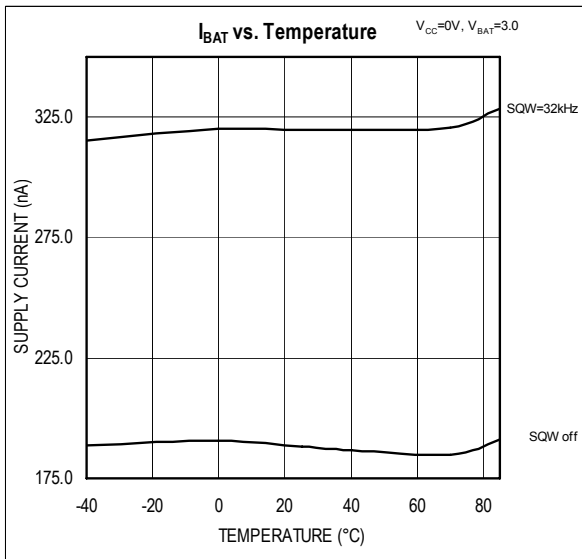
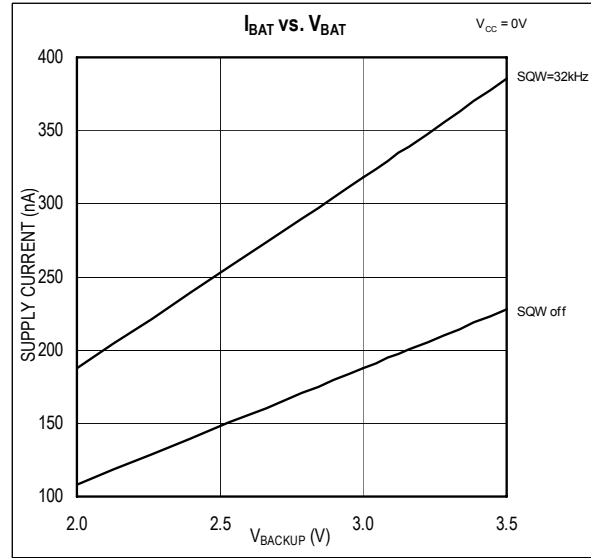
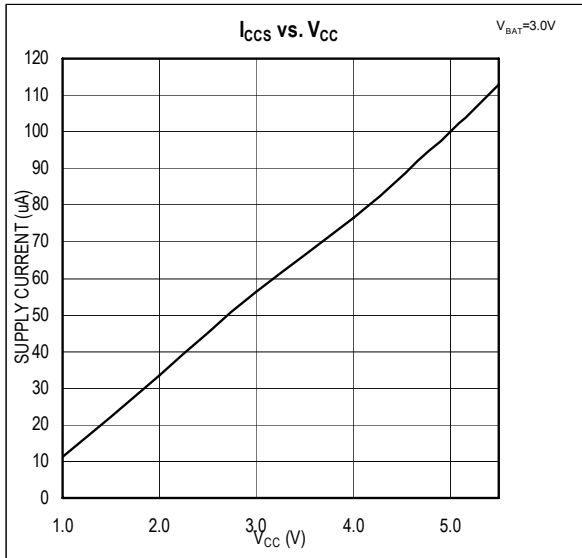


Figure 1. Block Diagram



TYPICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)



PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	Note: For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
3	V _{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V _{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/ga/info/ul/ .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
6	SCL	Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . If not used, this pin can be left floating.
8	V _{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{TP} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} + 0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

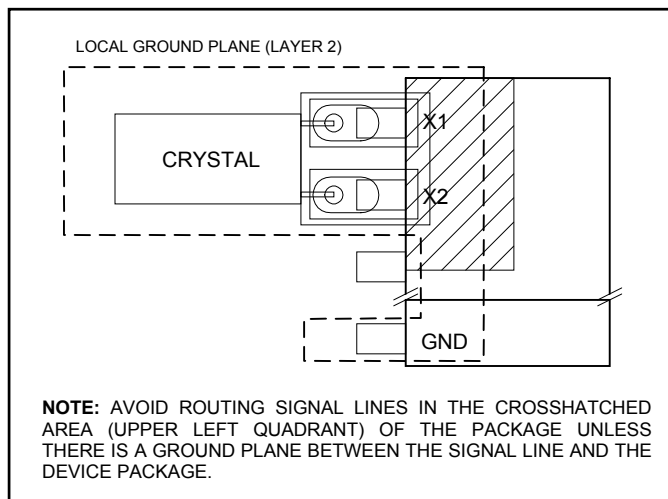
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_o		32.768		kHz
Series Resistance	ESR			45	k Ω
Load Capacitance	C_L		12.5		pF

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: *Crystal Considerations for Dallas Real-Time Clocks* for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{BATDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

I²C DATA BUS

The DS1307 supports the I²C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I²C bus.

Figures 3, 4, and 5 detail how data is transferred on the I²C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

START data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

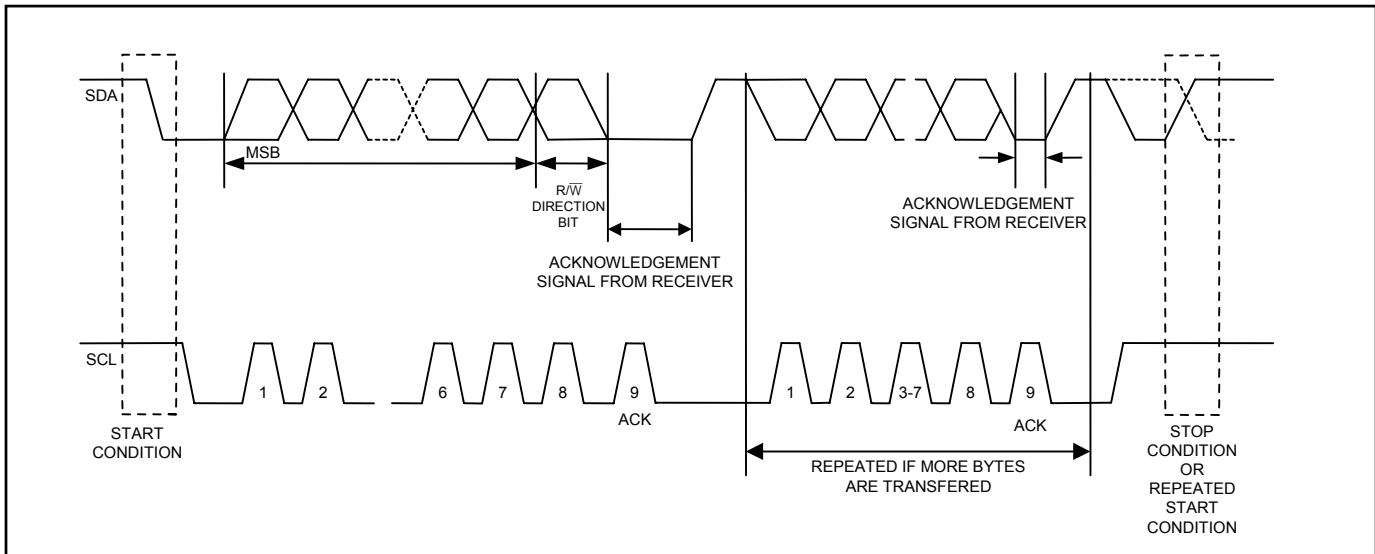
STOP data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I²C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I²C Serial Bus

Depending upon the state of the $\overline{R/W}$ bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 can operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ($\overline{R/\overline{W}}$), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ($\overline{R/\overline{W}}$), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

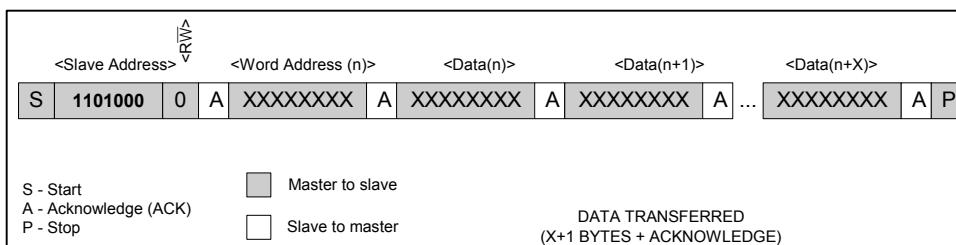


Figure 5. Data Read—Slave Transmitter Mode

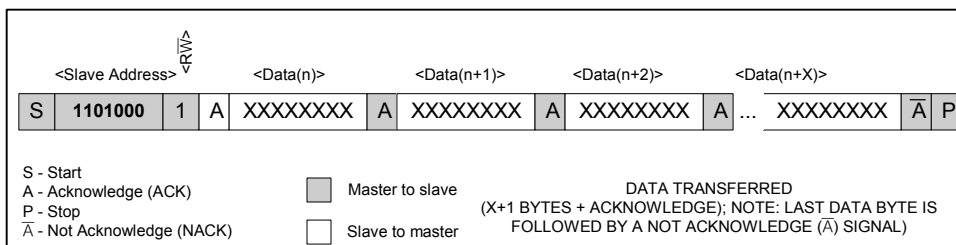
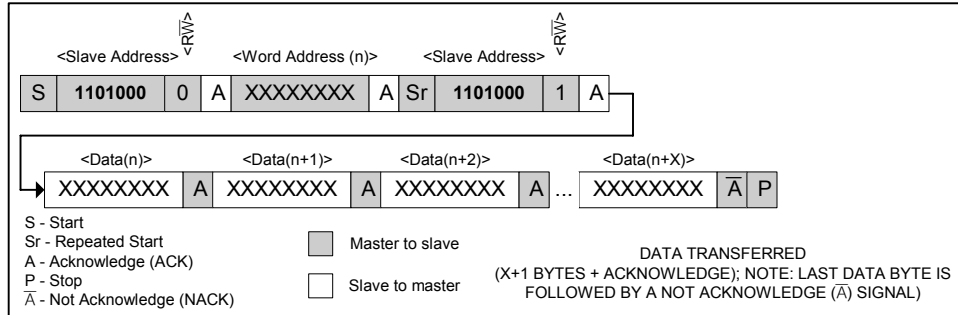


Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit

PACKAGE INFORMATION

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
8 PDIP	—	21-0043
8 SO	—	21-0041

REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
100208	Moved the <i>Typical Operating Circuit</i> and <i>Pin Configurations</i> to first page.	1
	Removed the leaded part numbers from the <i>Ordering Information</i> table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the <i>Pin Description</i> table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the <i>Clock and Calendar</i> section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the <i>Control Register</i> section.	9
	Updated the <i>Package Information</i> section to reflect new package outline drawing numbers.	13