

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



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Faculté de SCIENCE ET TECHNOLOGIE
Département d'Automatique, Télécommunication et d'Electronique

Mémoire de fin d'étude

En vue de l'obtention du diplôme *master* en électronique
Option : Instrumentation

THEME

Mise en œuvre du protocole de
communication I2C dans l'environnement
microcontrôleur PIC16F877A.

Préparé par :

- ROUBA Asma
- AIDEL Taous

Examiné par :

- Mr. Sadji
- Mme. Mezzah

Encadré par :

- Mr. Hanfoug Salah

Promotion : 2021-2022

Remerciement

*Nous rendons nos profondes gratitude à dieu qui nous a aidé à
réaliser ce modeste travail.*

*Nous exprimons nos profondes gratitude à nos parents pour
leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont
enduré.*

*Nous tenons à remercier notre encadrant Mr HANFOUG pour les
efforts qu'il a déployé, pour nous aider, conseiller, encourager et
corriger.*

*Nous tenons à remercier les membres de jury d'avoir accepter
d'examiner notre travail.*

*Nous souhaitons adresser nos remerciements à toute les personnes qui nous
ont accordé au moins une seconde pour contribuer à la réalisation de ce
mémoire.*

Dédicaces

*Avec l'aide d'ALLAH le tout puissant, ce travail est achevé. Je le
dédie;*

*A mes très chers parents qui ont donné sens à mon existence et qui
m'ont soutenu jour et nuit durant tout mon parcours*

*A mes chers frères MAHDI, RABIA et mes deux sœurs MERIEM et
KHADIDJA*

sans oublier mon neveu ISLAM et ma nièce AMIRA

A ma tante AICHA et son mari YAMINE.

A ma chère binôme ASMA et mes amies YASMINE et SARAH

*A tous ceux qui me connaissent et qui ont contribué de près ou de loin
à
la réalisation de ce mémoire.*

Taous.

Dédicaces

C'est avec une grande émotion que je dédie ce modeste travail

de fin

d'étude à mes êtres les plus chers ;

A mes très chers parents qui ont fait de moi ce que je suis

aujourd'hui.

A mes sœurs Hadil et Soundous et mon petit frère Adam.

A ma chère binôme Taous et ma chère amie Yasmine.

Asma.

Liste des figures

Chapitre 1 :

Figure I. 1 Contenu type d'un microcontrôleur.....	5
Figure I. 2 Description de la configuration du pic 16F877A.....	6
Figure I. 3 Les broches du PIC 16F877A.....	6

Chapitre 2 :

Figure II. 1 Les conditions Start et Stop.....	14
Figure II. 2 Exemple de condition de départ et d'arrêt.....	15
Figure II. 3 Exemple de transmission réussie.....	15
Figure II. 4 Exemple d'octet d'adresse.....	16
Figure II. 5 exemple lecture ou écriture d'une donnée	16
Figure II. 6 Exemple d'écriture d'une donnée.....	16
Figure II. 7 Ecriture d'une donnée (DS1307).....	17
Figure II. 8 Exemple de lecture d'une donnée	17
Figure II. 9 Lecture d'une donnée (DS1307)	18

Chapitre 3 :

Figure III. 1 Afficheur a cristaux liquide	23
Figure III. 2 Les entrées d'un afficheur	23
Figure III. 3 Schéma synoptique de l'afficheur	24
Figure III. 4 Branchement de L'afficheur LCD avec le PIC16F877A	25
Figure III. 5 PIC16F877A.....	25
Figure III. 6 Capteur de température TC74	26
Figure III. 7 EEPROM 24C02	27
Figure III. 8 Schéma d'application typique.....	28
Figure III. 9 Câblage DS1307	28
Figure III. 10 Le logiciel proteus	29
Figure III. 11 Présentation d'ISIS de PROTEUS	30
Figure III. 12 Création d'un nouveau projet.....	31
Figure III. 13 Les configurations de projet	32
Figure III. 14 L'emplacement du projet	32
Figure III. 15 Fenêtre de la saisie du programme	33

Chapitre 4 :

Figure IV. 1 Schéma synoptique du circuit	36
Figure IV. 2 Schéma électronique globale du circuit	37
Figure IV. 3 Chargement du programme dans le PIC sur ISIS	38
Figure IV. 4 Gravure du programme sur le Pic.....	40
Figure IV. 5 Le typon.....	41

Liste des tableaux

Chapitre 3 :

Tableau 1: Caractéristiques techniques	26
Tableau 2: registres du DS1307	29
Tableau 3 : Bibliothèque du MikroC.....	34

Sommaire

Introduction générale	1
Chapitre 1 : Mise en œuvre du microcontrôleur PIC16F877A	
1 Introduction :	4
2 Définition de microcontrôleur :	4
3 Définition d'un PIC :	5
3.1 Identification des PIC.....	5
4 Structure du PIC 16F877A :	6
4.1 Structure externe du PIC 16F877A.....	6
4.1.1 Les broches du PIC 16F877A.....	7
4.1.2 Les ports du PIC 16F877A	7
4.2 Structure interne du PIC 16F877A	7
4.2.1 Caractéristiques CPU.....	8
4.2.2 Les mémoires du PIC 16F877A.....	8
4.2.2.1 La mémoire programme :	8
4.2.3 Les timers du pic 16F877A.....	9
4.2.4 L'oscillateur.....	9
4.2.5 Les interruptions	10
4.2.5.1 Définition :	10
4.2.5.2 Catégories d'interruptions :	10
4.2.6 Le reset	10
5 Le bus I2C et le PIC 16F877A :	10
6 Conclusion :	11

Chapitre 2 : Le protocole I2C

1. Introduction :	13
2. Historique :	13
3. Caractéristiques du bus I2C :	13
4. Le protocole I2C :	14
4.1. L'échange entre le maitre et l'esclave :	14
4.2. La prise de contrôle du bus :	14

4.3.	La transmission d'un octet :.....	15
4.4.	La transmission d'une adresse :.....	15
4.5.	Écriture d'une donnée :.....	16
4.6.	Lecture d'une donnée :.....	17
7.	Les nouvelles caractéristiques :.....	19
a.	Les adresses réservées :.....	19
b.	Adresses d'appel général :.....	19
c.	Octet de START :.....	19
d.	Début d'adresse CBUS :.....	19
e.	Autres :.....	20
8.	Conclusion :.....	20

Chapitre 3 : Présentation et description des outils du projet

7	Introduction.....	22
8	Les composants utilisés :.....	22
9	Description matériel :.....	22
9.1	<i>Afficheur LCD:</i>	22
9.1.1	Définition de l'afficheur LCD :.....	22
9.1.2	Les entrées d'un afficheur LCD :.....	23
9.1.3	Brochage :.....	24
9.1.4	Branchements afficheur LCD- microcontrôleurs.....	24
9.2	<i>Le PIC 16F877 :</i>	25
9.3	<i>Le capteur de température TC74 :</i>	25
9.3.1	Caractéristiques techniques:.....	26
9.4	L'EEPROM 24C02 :.....	26
9.5	<i>Le module DS1307.....</i>	27
10	Description logiciels.....	29
10.1	<i>Le logiciel ISIS PROTEUS</i>	29
10.1.1	<i>Présentation générale</i>	29
10.1.2	Comment créer un nouveau projet sur ISIS PROTEUS ?.....	30
10.2	<i>Le logiciel MikroC</i>	30
10.2.1	Présentation du MikroC.....	30
10.2.2	Création d'un projet.....	31
10.2.3	Compilation.....	33

10.2.4	Les bibliothèques du MikroC :	33
11	Bus I2C et MikroC :	34
12	Conclusion.....	34

Chapitre 4 : Présentation et description des outils du projet

18	Introduction :	36
19	Description du système :	36
19.1	Schéma synoptique :	36
19.2	Présentation des blocs :	36
20	Simulation du circuit :	37
20.1	Schéma électronique :	37
20.2	Programmation du pic 16F877 :	37
20.3	Routage et création du circuit imprimé :	38
21	Réalisation pratique :	39
21.1	Listes des composants :	39
21.2	Gravure du programme sur le pic :	40
21.3	Réalisation du circuit imprimé et implantation des composants :	40
22	Conclusion :	41

	Conclusion générale	42
--	----------------------------------	----

Annexes

Références

Introduction générale :

Actuellement, l'électronique est au centre de toutes les réalisations et au cœur de plusieurs développements et perfectionnements. Les dispositifs électroniques sont aussi bien retrouvés dans le domaine médical, industriel mais aussi au niveau des systèmes de contrôle et de régulation automatique.

La majorité des processus industriels nécessitent de contrôler un certain nombre de paramètres : température, pression, niveau, débit, pH, concentration d'oxygène, etc. Ça appartient à la chaîne de régulation de maintenir à des niveaux prédéterminés les paramètres qui régissent le fonctionnement du processus.

Plusieurs applications industrielles exigent l'utilisation des méthodes de contrôle, permettant d'avoir une réponse rapide et des performances dynamiques élevées. Parmi ces méthodes, on cite la régulation de température.

Cela se fait par la mise en place d'un système à logique programmée basée sur le protocole de communication I2C et d'un microcontrôleur qui voudrait gérer toutes les opérations, les différents traitements et d'assurer par conséquent les liaisons entre les différentes parties électroniques du système. On a choisi ce microcontrôleur parce qu'il dispose d'une mémoire flash pour stocker le programme (donc il est possible de le reprogrammer en cas de bug ou d'évolution) et d'une mémoire RAM de taille suffisante.

Ce système réalise une solution de régulation numérique de température à base d'un PIC16F877A qui assure le contrôle du système de régulation constitué d'un ventilateur et une résistance de réchauffement par l'information logique et numérique en déclenchant le système de ventilation lorsque la température mesurée par le capteur TC74 est supérieure à la température pré-saisie, et le système de réchauffement quand la température mesurée est inférieure.

En addition, le système permet d'afficher l'heure et la date exactes sans la nécessité d'avoir accès à une connexion internet, qui est pratique dans un environnement industriel, en utilisant le module RTC-DS1307.

Le travail décrit dans ce présent document est structuré en une introduction, quatre chapitres, une conclusion ainsi que des références bibliographiques.

Le premier chapitre, intitulé « Mise en œuvre du microcontrôleur PIC16F877A » est consacré à la description de l'unité de commande et de traitement de notre système : Le microcontrôleur PIC16F877A, ainsi ses différentes ressources internes, et ses principales caractéristiques.

Le deuxième chapitre, intitulé « Le protocole I2C » qui un protocole de communication rapide et facile à mettre en œuvre, Il permet de créer des bus contenant plusieurs maîtres et esclaves. Le logiciel de programmation mikroC et utilisé pour l'implantation de ce protocole.

Le troisième chapitre, intitulé intitulé « Présentation et description des outils du projet »

nous avons présenté les éléments jugés importants pour la réalisation de notre travail. A cet effet nous avons présenté d'une manière générale les différents blocs nécessaires à la réalisation pratique de notre système électronique, ainsi que la description des outils de conception et de simulation (logiciel proteus, mikroC) avec lesquels nous avons travaillé.

Le quatrième chapitre, intitulé « Conception et réalisation pratique » concerne la conception et la réalisation pratique des différents blocs de notre système. On effectue nous avons décrit toutes les étapes de réalisation du prototype, et décrivant la partie informatique on programme le PIC16F877A avec le logiciel MikroC, et la simulation des différents montages par le simulateur ISIS Professionnel pour s'assurer de leur bon fonctionnement électronique et par conséquent du montage électronique du système, ce chapitre s'achève sur la réalisation de la partie électronique sur plaque d'essai.

Nous terminons par une conclusion générale et des perspectives.

Chapitre 1 :

Mise en œuvre du
microcontrôleur PIC16F877A

1 Introduction :

L'évolution des systèmes électroniques amène de plus en plus souvent les concepteurs à Remplacer l'électronique câblée à base de nombreux circuits intégrés par un circuit programmable qui remplit à lui seul toutes les fonctions, les microcontrôleurs appartiennent à cette famille de circuits.

Le développement des applications à base des microcontrôleurs PIC est assez courant, ceci est dû à plusieurs causes : réalisation d'un ensemble important de traitements d'informations, beaucoup de ressources internes, mémoires embarquées de plus en plus grande vitesse de calcul accrue...

Les microcontrôleurs sont très utilisés dans le monde de l'industrie, notamment dans les systèmes embarqués. On pourra donc les retrouver dans l'aéronautique, l'aérospatial, l'automobile, l'électronique grande publique. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques.

Au niveau de ce chapitre on va essayer de mieux connaître le PIC16F877A (PIC choisie pour ce projet), de savoir manipuler ces instructions internes.

2 Définition de microcontrôleur :

Le microcontrôleur (μC) est un dérivé du microprocesseur. Sa structure est celle des systèmes à base de microprocesseurs. La partie la plus « noble » du microcontrôleur est ce qu'on appelle la CPU (Central Processing Unit, unité centrale de traitement). C'est dans cette unité que se trouvent plusieurs éléments tels que l'ALU (Arithmetic and Logical Unit, unité arithmétique et logique) qui exécute une par une l'ensemble des instructions écrites dans le programme principal.

Le microcontrôleur est aussi composé d'une mémoire (mémoire vive RAM et mémoire morte ROM), une (ou plusieurs) interface de communication avec l'extérieur matérialisé par les ports d'entrée/sortie.

En plus de cette configuration minimale, les microcontrôleurs sont dotés d'autres circuits d'interface qui vont dépendre du microcontrôleur choisi à savoir les systèmes de comptage (TIMER), les convertisseurs analogique/numérique (CAN) intégré, gestion d'une liaison série ou parallèle, un Watchdog (surveillance du programme), une sortie PWM (modulation d'impulsion), ... [1] [2] comme le montre la figure suivante :

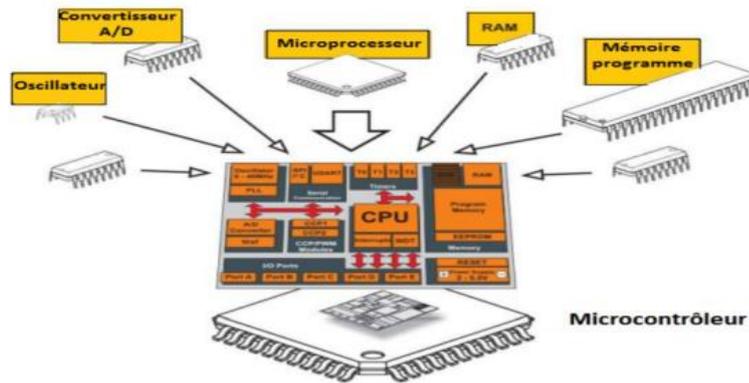


Figure I. 1 : Contenu type d'un microcontrôleur. [3]

3 Définition d'un PIC :

Le PIC (Programmable interface controller) est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques interne permettant de faciliter l'interfaçage avec le monde extérieur sans nécessité l'ajout de composants externes. Les PICs sont des composants dits RISC (Reduced Instruction set computer), ou encore composant à jeu d'instruction.[4].

3.1 Identification des PIC

Pour identifier un PIC, on utilise simplement son appellation du type : ww L xx yyy-zz

- WW : Représente la catégorie du composant (12, 14, 16, 17, 18).

- L : Tolérance plus importante de la plage de tension.

- XX : Type de mémoire de programme :

C: EPROM ou EEPROM.

CR: PROM.

F: FLASH.

- YYY : indique le modèle du PIC

-ZZ : la fréquence d'horloge maximale supportée par le PIC.

Concernant le PIC 16F877-A :

16 : indique la famille du PIC (Mid-Rang).

F : indique le type de la mémoire utilisée (Flash).

877 : identité du PIC.

A : fréquence maximale d'horloge qui est de 20MHZ.

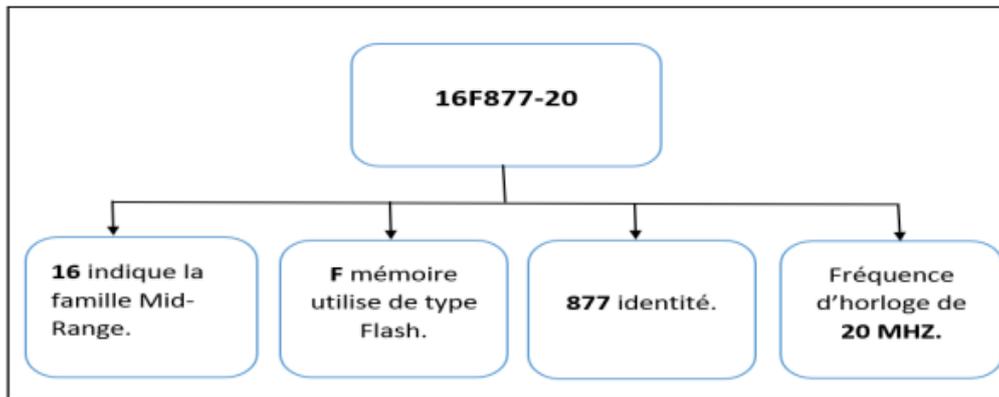


Figure I. 2 : Description de la configuration du pic 16F877A.

4 Structure du PIC 16F877A :

La diversité des domaines d'utilisation des Pics se présentent sous différentes formes et structures, pour quelles soit adaptées pour chaque domaine d'utilisation.

Nous allons présenter dans ce qui suit les structures susceptibles des PIC à savoir la structure externe et interne.

4.1 Structure externe du PIC 16F877A

Le schéma de la figure représente la structure externe du PIC 16F877 comportant [5]

- 33 pins d'entrées/sorties.
- 4 pins pour l'alimentation : deux connections VDD et deux connections VSS.
- 2 pins pour l'oscillateur.
- 1 pin pour le RESET : MCLR.

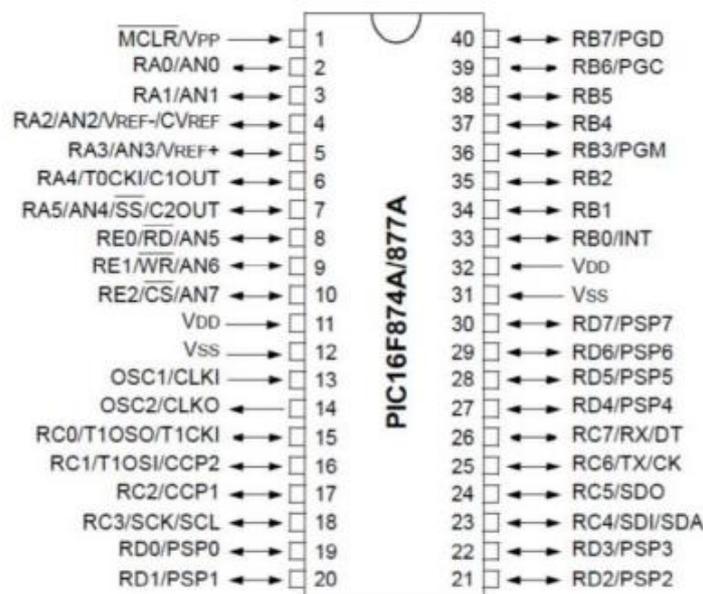


Figure I. 3 : Les broches du PIC 16F877A. [6]

4.1.1 Les broches du PIC 16F877A

1) Les broches de fonctionnement

Les broches de « fonctionnement » sont les broches qui permettent au microprocesseur de fonctionner. Ces broches doivent obligatoirement être connectées pour que le 16F877 fonctionne d'où leur nom[7] :

- Les broches d'alimentation.
- Les broches du Quartz.
- La broche de réinitialisation.

2) Les Ports du 16F877A

Toutes les autres broches du 16F877A sont des broches de port pour la communication avec l'extérieur.

4.1.2 Les ports du PIC 16F877A

Pour communiquer avec l'extérieur, le PIC 16F877A dispose de 5 ports qui sont bidirectionnels, ce qui signifie qu'ils peuvent être configurés et utilisés comme étant des sorties ou entrées : [8]

- Port A : 6 pins I/O numérotées de RA0 à RA5.
- Port B : 8 pins I/O numérotées de RB0 à RB7.
- Port C : 8 pins I/O numérotées de RC0 à RC7.
- Port D : 8 pins I/O numérotées de RD0 à RD7.
- Port E : 3 pins I/O numérotées de RE0 à RE2.

En résumé :

- Le sens E/S des broches est configuré dans un registre appelé TRIS. Ce dernier permet de définir si les différentes pattes du port fonctionnent en entrée ou en sortie.

- Un « 1 » dans un bit du registre TRISA met la sortie correspondante en haute impédance, elle peut ainsi servir d'entrée.
- Un « 0 » dans un bit de ce registre transfère le contenu de la sortie de la bascule D sur la sortie correspondante.

- Un registre de données est attaché à chaque port.

- Certaines pattes ont plusieurs fonctions : On dit que les fonctions sont multiplexées[9].

4.2 Structure interne du PIC 16F877A

On trouve à l'intérieur d'un microcontrôleur des éléments d'une structure à base d'un microprocesseur qui sont réunis dans un seul circuit intégré et ces éléments sont : [10]

- Une mémoire de donnée (RAM et EEPROM).
- Une mémoire de programme (FLASH, ROM, OTPROM, UVPROM ou EEPROM).
- Un chien de garde.
- 13 sources d'interruptions.
- Générateur d'horloge.
- Interface parallèle pour la connexion des entrées/sorties
- Interfaces séries (synchrone ou asynchrone).
- D'autres périphériques (compteur/timer, convertisseurs analogiques/numériques (10 bits) à 8 canaux, etc.).

Chapitre 1 : Mise en œuvre du microcontrôleur PIC16F877A

Le schéma bloc de la figure ci-dessous, représente la structure interne du PIC 16F877A.

4.2.1 Caractéristiques CPU

Un microcontrôleur, c'est avant tout un microprocesseur, c'est l'endroit où toutes les opérations arithmétiques et logiques contenant dans un microprogramme stocké en mémoire programme sont effectuées séquentiellement. L'unité centrale de traitement ou processeur contrôle les opérations internes du microprocesseur et envoie des signaux de contrôle à d'autres parties du microprocesseur pour exécuter les instructions requises [18].

❖ ALU (Unité arithmétique et logique) :

C'est le cerveau du microcontrôleur, permettant d'effectuer des opérations entre l'accumulateur et une opérande, son fonctionnement dépend de la fréquence d'horloge

4.2.2 Les mémoires du PIC 16F877A

La mémoire est une partie très importante du système de microcontrôleur, qui peut être classifié en deux différents types : mémoire de données et mémoire programme [10].

4.2.2.1 La mémoire programme :

C'est dans cette zone que nous allons écrire notre programme. Le PIC exécute une à une les instructions logées dans la mémoire de programme. Ce type de mémoires uniquement accessible en lecture. On retrouve :[11]

a) EEPROM programme (flash) :

Cette mémoire de 8 x 1024 mots de 14 bits sert à stocker le programme, mais elle est accessible par programme et peut donc être utilisée comme une extension de la mémoire EEPROM de données. Elle est non volatile (c-à-d que les données sont conservées en cas de coupure de l'alimentation ; flash) et reprogrammable à souhait.

C'est une mémoire qui stocke le programme du microcontrôleur. Après compilation du fichier source, le compilateur génère un fichier « .hex » qui est transféré ensuite dans la mémoire programme du PIC à l'aide d'une interface appropriée.

b) La mémoire de donnée :

C'est une mémoire similaire à la mémoire programme. On s'en sert surtout pour stocker des constantes. Elle permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats d'opérations, états des capteurs...). Ces mémoires sont accessibles en écriture et en lecture.

On en trouve 2 types :[11]

i. De la mémoire morte (EEPROM) :

Non-volatile, ayant un temps d'écriture assez élevé (quelques ms) par rapport au temps de lecture qui est assez faible (quelques ns). Elle est constituée de 256 octets, ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres Semi-permanents.

ii. De la mémoire vive (RAM) :

La mémoire RAM est celle qui est souvent utilisée. Elle est Volatile (données perdues en cas de coupure de l'alimentation) ayant un temps de lecture et écriture assez court (quelques ns), elle comprend tous les registres spéciaux permettant de contrôler le PIC ainsi que ses périphériques. La mémoire RAM disponible du 16F877 est de 368 octets.

4.2.3 Les timers du pic 16F877A

Les Timers/compteurs sont des périphériques de gestion de temps. réaliser les fonctions
Les timers sont des compteurs formés généralement d'un pré-diviseur suivi d'un registre compteur de 8 ou 16 bits. L'entrée d'horloge peut être interne (mode timer) ou externe (mode compteur d'événements). Lorsque le registre compteur atteint sa valeur maximale et repasse à 0, un bit indicateur (flag) sera positionné et une interruption pourra être générée, informant ainsi la CPU du débordement du timer. Il faut bien noter que le programmeur devra remettre à zéro cet indicateur après chaque débordement.

Le microcontrôleur PIC16F877A dispose de trois timers appelés Timer0, Timer1 et Timer2. [11]

1. Timer0 :

Le timer 0 est un compteur 8 bits qui peut compter (de 0 à 255) :

- soit les impulsions de l'horloge via un pré diviseur
- soit des impulsions externes, via la broche PA4 Le débordement (over flow) du compteur, qui a lieu lorsque le compteur passe de 255 à 0, provoque une interruption. [12]

2. Timer1 :

Le timer 1 est un compteur 16 bits qui peut compter (de 0 à 65535) :

- soit les impulsions de l'horloge
 - soit les impulsions externes, et en particulier les impulsions d'un quartz externe.
- Là aussi, le débordement provoque une interruption. [12]

3. Timer2 :

Le timer 2 est un timer couplé au module dit CCP. Il peut compter (de 0 à 65535) comme le timer1. C'est ce timer que nous utilisons pour compter les durées entre l'émission et la réception des signaux venant de l'émetteur [12].

4.2.4 L'oscillateur

Tous les microcontrôleurs nécessitent une horloge (ou un oscillateur) pour fixer la vitesse d'exécution des instructions.

On utilise pour ce faire un quartz dont le rôle est de créer une impulsion de fréquences élevées. Dans le cas du 16F877A, le quartz utilisé est typiquement un quartz de 8Mhz.

Le temps qui s'écoule entre 2 impulsions s'appelle un cycle d'horloge .Un cycle d'horloge dure donc 125 nanosecondes [12].

L'horloge peut être soit interne soit externe. L'horloge interne est constituée d'un oscillateur à quartz ou d'un oscillateur RC. [1]

RC : une résistance et un condensateur pour une utilisation normale. Elle peut varier légèrement d'un circuit à l'autre, et dont n'est pas trop précise.

4.2.5 Les interruptions

4.2.5.1 Définition :

Une interruption est un signal déclenché par un événement interne à la machine ou externe, qui provoque l'arrêt immédiat d'un programme en cours d'exécution, et cela juste à la fin de l'opération courante, et cela au profit d'un programme plus prioritaire appelé programme d'interruption.

Ensuite, le programme interrompu reprend son exécution à l'endroit où il avait été interrompu[1].

4.2.5.2 Catégories d'interruptions :

Les interruptions peuvent être des origines diverses, mais on peut les classer en deux grandes classes (ou catégories) : [13]

- Les interruptions externes ou matérielles.
- Les interruptions internes ou logicielles.

❖ **Interruptions externes** : (indépendantes du processus) : Ce sont les interruptions causées par des organes externes au processeur central, comme les horloges de temps, les périphériques d'E/S, ...etc [14].

❖ **Interruptions internes** : Ce sont des interruptions causées par des anomalies dans l'exécution du programme en cours protection du système et des processus, appelées par une instruction à l'intérieur d'un programme (erreur d'adressage, code opération inexistant, problème de parité...) [14].

4.2.6 Le reset

Il existe différentes situations pouvant entraîner un reset du circuit : [15]

- **Power on reset** : la tension d'alimentation descend en dessous du seuil de 1,2 à 1,7v.
- **Brown on reset** : la tension d'alimentation reste vers un niveau de 4 v pendant un temps assez long .
- **Watcdog reset** : action du chien de garde .
- **MCLR/ activé** : action sur l'entrée de MCLR.

Ceci provoque l'arrêt du programme qui va recommencer à la première instruction.

Pour que le microprocesseur fonctionne, il faut donc que cette broche soit connectée au +5V en permanence.

5 Le bus I2C et le PIC 16F877A :

Le PIC16F877A possède des E/S spécialement prévues pour la communication I2C (ce dont nous expliquerons en détail dans le chapitre suivant). Ce sont les broches nommées RC3 (SCL) et RC4 (SDA). Elles sont directement reliées à un contrôleur I2 C. Ce contrôleur prévoit la gestion de tout le protocole I2C (Prise de parole, collision, ligne occupée ...). Il permet une gestion facile de la liaison, étudier en détail au chapitre deux.

6 Conclusion :

Nous avons traité tout au long de ce chapitre l'élément que nous avons jugés importants pour notre travail à savoir ses caractéristiques, son architecture interne et externe, ...etc.

Partant de cette étude, on peut déduire que le microcontrôleur 16F877 peut bien jouer le rôle d'une unité de contrôle pour notre montage

Chapitre 2 :

Le protocole I2C

1. Introduction :

La norme I2C (Inter-Integrated Circuit) a été créée pour fournir un moyen simple de transférer des informations numériques entre des capteurs et des microcontrôleurs. En utilisant la programmation, et dans notre projet on a choisi le programme MikroC.

2. Historique :

Le bus I2C (Inter Integrated Circuit Bus) est le bus historique, développé par Philips pour les applications de domotique et d'électronique domestique au début des années 80, notamment pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur moderne.

Ainsi, dans notre téléviseur, tous les ensembles sont sur un bus I2C. Il existe d'innombrables périphériques exploitant ce bus, il est même implantable par logiciel dans n'importe lequel des microcontrôleurs[16].

3. Caractéristiques du bus I2C :

Le bus I2C a été inventé pour simplifier la communication entre des composants divers dont le signal signifie (Inter Integrated Circuit) ce qui donne IIC et par contraction I2C . C'est un bus de communication de type série synchrone. Ce bus est parfois nommé sous le nom de TWI (Two Wire Interface). Il permet de communiquer sur 3 fils :

- un signal de données (SDA).
- un signal d'horloge (SCL).
- un signal de référence électrique (Masse).

Ceci permet de réaliser des équipements ayant des fonctionnalités très puissantes en apportant toute la puissance des systèmes microprogrammés et conservant un circuit imprimé très simple, par rapport un schéma classique.

Les données sont transmises en série à 100Kbits/s en mode standard et jusqu'à 400Kbits/s en mode rapide. Ce qui ouvre la porte de cette technologie à toutes les applications où la vitesse n'est pas primordiale. De nombreux fabricants ayant adopté le système, la variété des circuits disponibles disposant d'un port I2C est énorme :

Ports d'E/ S bidirectionnels, Convertisseurs A/N et N/A, mémoires (RAM, EPROM, EEPROM, etc.), Circuits audio (Égaliseur, Contrôle de volume, ...) et autres drivers (LED, LCD, etc.) Le nombre de composants qu'il est ainsi possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL : 400 pF [17].

4. Le protocole I2C :

Le protocole I2C définit la succession des états logiques possibles sur SDA et SCL, et la façon dont doivent réagir les circuits en cas de conflits.

Il y a une condition appelée « Start » qui est toujours la condition du début des échanges de données, cela se traduit par le passage de l'état logique 1 (niveau haut) à l'état 0 (niveau bas) pour SDA avec le niveau haut pour SCL. La condition « Stop » détermine la fin des échanges avec le passage de SDA de l'état 0 (niveau bas) à l'état logique 1 (niveau haut) avec la ligne SCL à 1 [18].

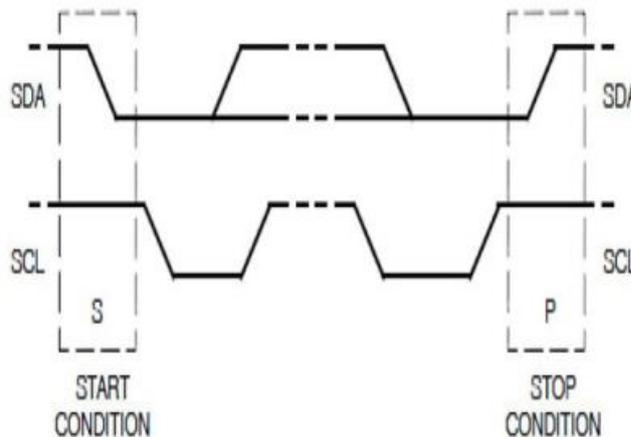


Figure II. 1 : Les conditions Start et Stop. [19]

4.1. L'échange entre le maître et l'esclave :

Un exemple type d'échange entre le maître et l'esclave :

Cela commence, comme on l'a vu précédemment par un bit de Start, puis l'adresse du composant esclave, suivi du bit de lecture/écriture mis à zéro (R/W=0). L'esclave répond au maître par un bit d'acquittement (ACK) puis peut demander une pause. Une fois la pause terminée, le maître envoie les données puis reçoit un bit d'acquittement ou de non-acquittement envoyé par l'esclave, il peut demander alors une pause. Le maître envoie un bit de restart, puis désigne le même esclave avec cette fois le bit d'écriture/lecture à 1, le maître devient alors récepteur et l'esclave émetteur. L'émetteur envoie alors une série de données, toujours suivi d'un bit d'acquittement envoyé par le maître. Une fois le dialogue terminé, le maître envoie un bit de non-acquittement pour terminer l'échange ainsi qu'un bit de stop à tous les esclaves.

4.2. La prise de contrôle du bus :

Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (SDA et SCL à '1'). Pour transmettre des données sur le bus, il faut donc surveiller deux conditions particulières :

- La condition de départ (SDA passe à '0' alors que SCL reste à '1')
- La condition d'arrêt. (SDA passe à '1' alors que SCL reste à '1')

Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le maître. C'est lui qui génère le signal d'horloge.

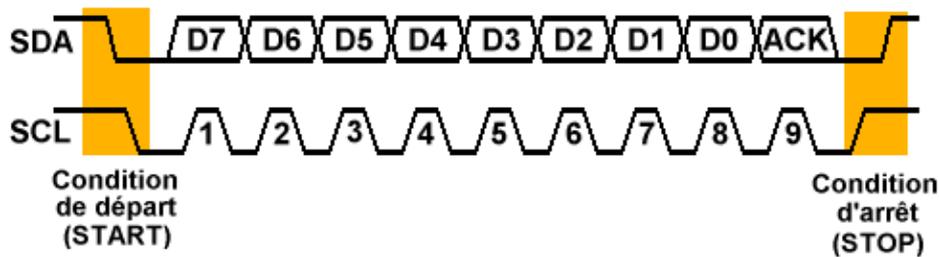


Figure II. 2 : Exemple de condition de départ et d'arrêt [20]

4.3. La transmission d'un octet :

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de poids fort D7. Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL. Lorsque SCL revient à « 0' », il recommence l'opération jusqu'à ce que l'octet complet soit transmis. Il envoie alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement. Les sorties de chacun étant à collecteurs ouverts, le maître voit le '0' et peut alors passer à la suite [18].

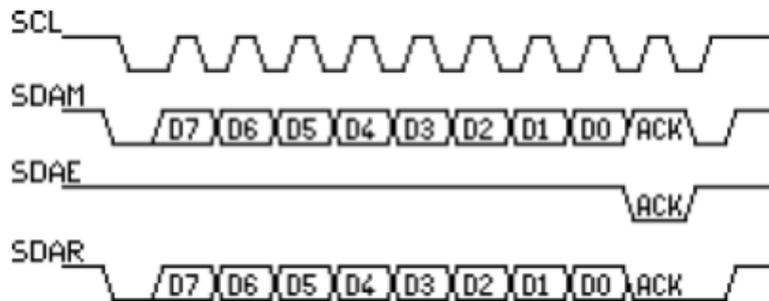


Figure II. 3 : Exemple de transmission réussie [21]

Dans cet exemple :

- SCL : Horloge imposée par le maître.
- SDAM : Niveaux de SDA imposés par le maître.
- SDAE : Niveaux de SDA imposés par l'esclave.
- SDAR : Niveaux de SDA réels résultants.

4.4. La transmission d'une adresse :

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il est nécessaire de définir pour chacun une adresse unique. L'adresse d'un circuit, codée sur sept bits, est définie d'une part par son type et d'autre part par l'état appliqué à

un certain nombre de ces broches. Cette adresse est transmise sous la forme d'un octet au format particulier [18].

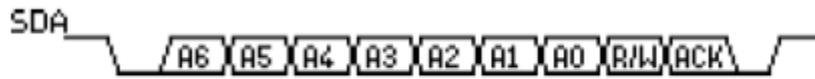


Figure II. 4 : Exemple d'octet d'adresse [22]

On remarque ici que les bits D7 à D1 représentent les adresses A6 à A0, et que le bit D0 est remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée. Le bit d'acquittement ACK fonctionne comme pour une donnée.

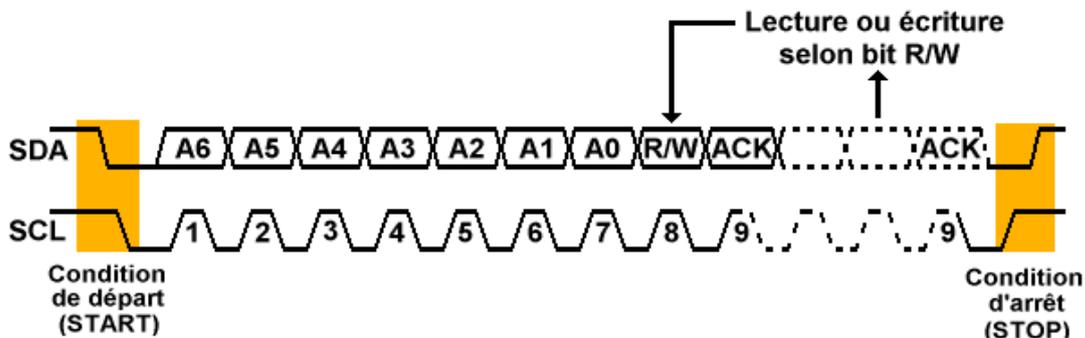


Figure II. 5 exemple lecture ou écriture d'une donnée [20]

4.5. Écriture d'une donnée :

L'écriture d'une donnée par le maître ne pose pas de problème particulier [22] :

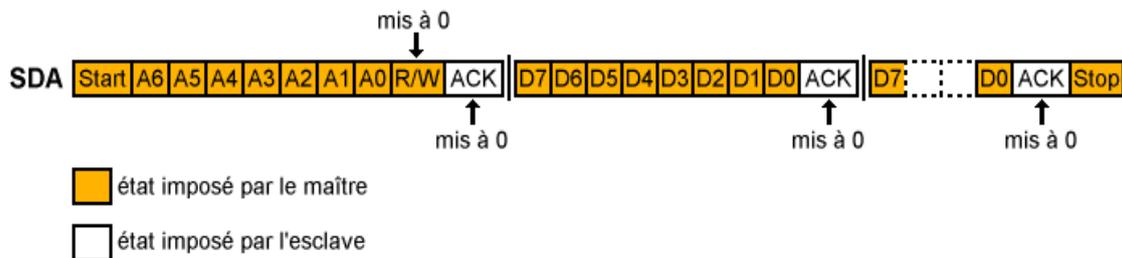


Figure II. 6 : Exemple d'écriture d'une donnée. [20]

Remarque : dans le cas particulier d'utilisation d'ACK, l'écriture d'un octet dans certains composants (Mémoires, microcontrôleur, ...) peut prendre un certain temps. Il est donc possible que le maître soit obligé d'attendre l'acquittement ACK avant de passer à la suite.

✚ Exemple d'écriture du DS1307 [18]

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

Le registre d'adresse 0x04 du DS1307 contient la date.

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 :

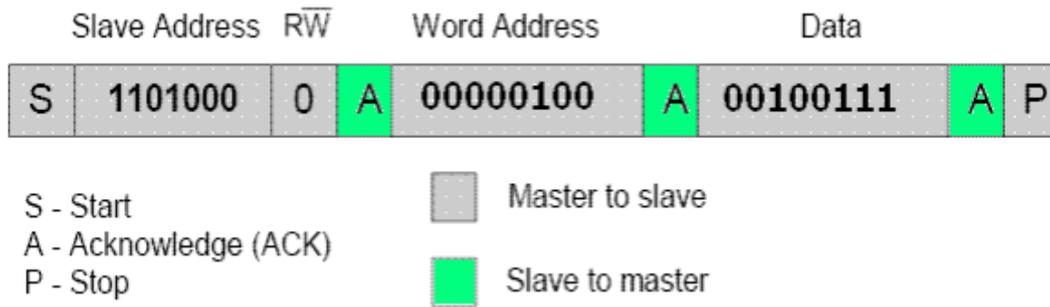


Figure II. 7 : Ecriture d'une donnée (DS1307) [23]

- 1) Pour initier le dialogue, le maître crée une condition Start.
- 2) Le maître envoie l'adresse de l'esclave (11001000) 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).

4.6. Lecture d'une donnée :

La lecture d'une donnée par le maître se caractérise par l'utilisation spéciale qui faite du bit ACK. Après la lecture d'un octet, le maître positionne ACK à '0' s'il veut lire la donnée suivante (cas d'une mémoire par exemple) ou à '1' la cas échéant. Il envoie alors la condition d'arrêt [22].

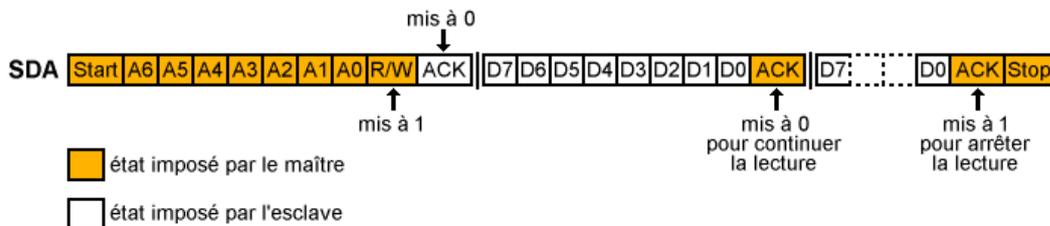


Figure II. 8 : Exemple de lecture d'une donnée [20]

✚ Exemple de lecture du DS1307 [18]

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.

Voici comment lire, le contenu des registres d'adresses 0x00 à 0x06 du DS1307

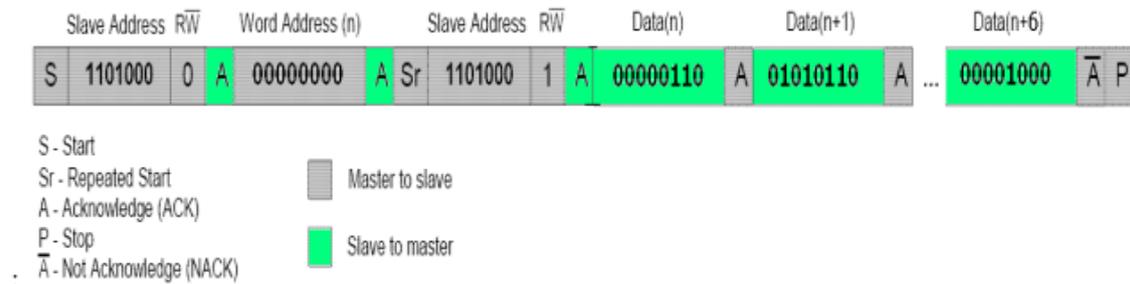


Figure II. 9 : Lecture d'une donnée (DS1307) [23]

- 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 0x07 (c'est-à-dire 7 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 0x0D (c'est-à-dire année 2013).
- (Mardi 07 mai 2013, 9 heures 56 minutes et 6 secondes).

7. Les nouvelles caractéristiques :

Afin de compenser quelques lacunes des premières spécifications du bus I2C (qui datent de 1982), quelques nouvelles améliorations ont été apportées à partir de 1993 : [6] [17]

- **Le mode rapide** : Le bus a désormais la capacité de transmettre des données jusqu'à une vitesse de 400 Kbit/s.
- **Des entrées à triggers de Schmitt** : Afin de limiter la sensibilité au bruit.
- **La mise en haute impédance d'un circuit non alimenté** : Ceci évite de bloquer le bus si un périphérique n'est pas alimenté.
- **Extension à 10 bits de l'adressage des circuits** répartis sur 2 octets de la manière suivante :

a. Les adresses réservées :

Les adresses 0000 0xxx ne sont pas utilisées pour l'adressage de composants. Elles ont été réservées par Philips pour effectuer certaines fonctions spéciales.

b. Adresses d'appel général :

Adresse : 0000 0000.

Après l'émission d'un appel général, les circuits ayant la capacité de traiter ce genre d'appel émettent un acquittement.

Le deuxième octet permet de définir le contenu de l'appel :

- 0000 0110 : RESET. Remet tout les registres de circuits connectés dans leur état initial (Mise sous tension). Les circuits qui le permettent rechargent leur adresse d'esclave.
- 0000 0010 : Les circuits qui le permettent rechargent leur adresse d'esclave.
- 0000 0100 : Les circuits définissant leur adresse de façon matériel réinitialisent leur adresse d'esclave.
- 0000 0000 : Interdit
- xxxx xxx1 : Cette commande joue le rôle d'interruption. xxxx xxx peut être l'adresse du circuit qui a généré l'interruption.

c. Octet de START :

Adresse : 0000 0001.

Cet octet est utilisé pour synchroniser les périphériques lents avec les périphériques rapides.

d. Début d'adresse CBUS :

Adresse : 0000 001x.

L'émission de cet octet permet de rendre sourd tout les circuits I2C présents sur le bus.

A partir de ce moment, on peut transmettre ce que l'on désire sur le bus, en utilisant par exemple un autre protocole. Le bus repasse en mode normal lors de la réception d'une condition d'arrêt.

e. **Autres :**

Adresses : 0000 0110 à 0000 1111.

Ces octets ne sont pas définis et sont ignorés par les circuits I2C. Ils peuvent être utilisés pour débayer un réseau multi master.

8. Conclusion :

Dans ce chapitre nous avons étudié le protocole de bus I2C, c'est un protocole de communication rapide et facile à mettre en œuvre. Nous avons présenté le principe de lecture et d'écriture d'une donnée, il permet de créer des bus contenant plusieurs maîtres et esclaves. Le logiciel de programmation mikroC a été utilisé dans l'implantation de ce protocole.

Chapitre 3 :

Présentation et description des outils du projet

1 Introduction

Avec l'évènement de ce que l'on appelle les "nouvelles technologies", l'objectif premier est de réaliser des traitements de plus en plus complexes le plus rapidement possible, en particulier dans le domaine d'électronique, cependant il n'est plus rare de nos jours de voir les appareils usuels remplacés par des appareils électroniques, nous pouvons penser entre autres à la régulation de température et à l'horloge temps réel (RTC).

Ce projet nous permettra donc de réaliser cet appareil. La conception et la réalisation de ce système dépend de microcontrôleur PIC16F877A en utilisant le protocole de communication I2C. Ici, le microcontrôleur va gérer tous les processus dans notre circuit. La simulation est réalisée à l'aide d'un programme élaboré sous ISIS et la programmation sous logiciel micro C.

2 Les composants utilisés :

- ✓ PIC 16F877A
- ✓ LCD 4X20
- ✓ TC74
- ✓ EEPROM 24C02
- ✓ Module RTC-DS1307- (Real Time Clock)
- ✓ Quartz 8MHz
- ✓ LEDs
- ✓ Potentiometer pot-Hg 1K
- ✓ Resistances

3 Description matériel :

3.1 Afficheur LCD :

3.1.1 Définition de l'afficheur LCD :

LCD est l'abréviation du terme anglais (Liquid Crystal Display) qui signifie en français (Ecran à cristaux liquides).

L'afficheur LCD est en particulier une interface visuelle entre un système (projet) et l'homme (utilisateur). Son rôle est de transmettre des informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploiter par l'utilisateur d'un système.

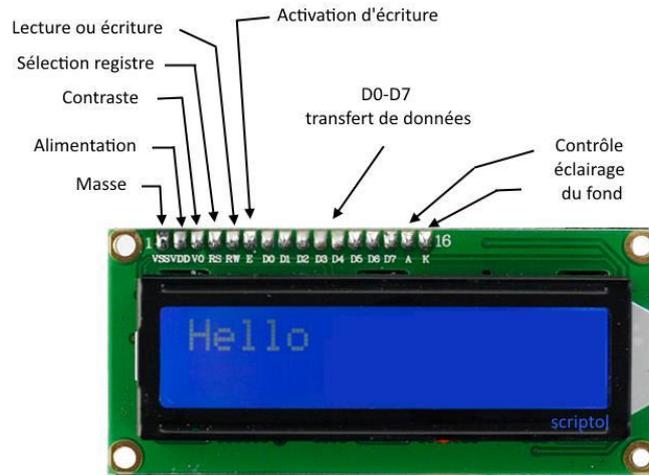


Figure III. 1 : Afficheur à cristaux liquide [24]

3.1.2 Les entrées d'un afficheur LCD :

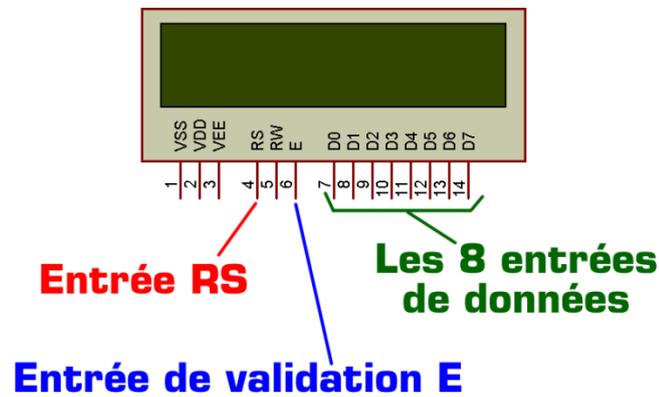


Figure III. 2 : Les entrées d'un afficheur. [25]

- ✓ RS permet de préciser si la commande présente sur les entrées D0 à D7 est une instruction ou une donnée, et E permet de valider cette commande.
- ✓ Une commande est une valeur numérique présente sur les entrées D0 à D7 et validée par une impulsion sur E.
- ✓ Le protocole d'envoi des commandes précise la liste des instructions à envoyer pour configurer l'afficheur (RS=0) suivie des données à envoyer (RS=1).
- ✓ L'entrée RW sera mise à zéro (connectée à la masse) et sera inutilisée ici.
- ✓ Les 3 entrées d'alimentation VSS, VDD et VEE n'ayant pas besoin d'être obligatoirement alimentées dans ISIS Proteus, elles resteront non connectées dans les exemples ci-dessous.

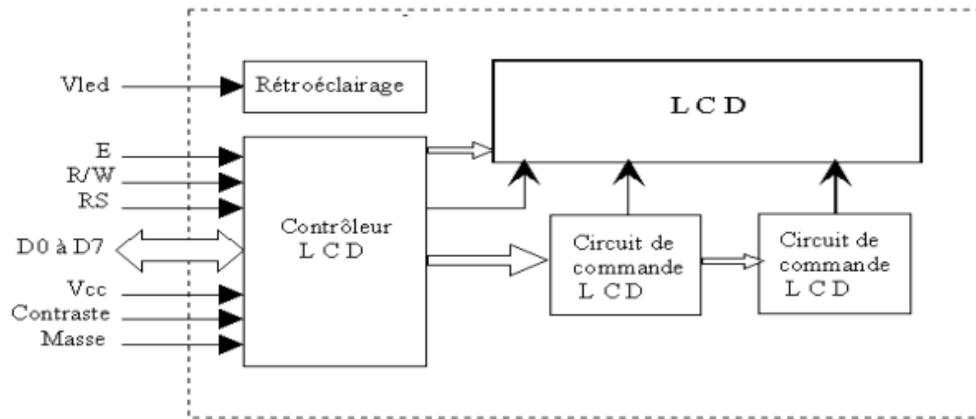


Figure III. 3 : Schéma synoptique de l'afficheur. [26]

3.1.3 Brochage :

La description des différentes broches est donnée comme suit :

- **VCC** : alimentation de l'afficheur LCD (5V).
- **GND** : la masse.
- **Contraste** : entrée permettant de régler le contraste de l'afficheur LCD. Il faut appliquer une tension continue réglable (entre 0V et 5V) à l'aide d'un potentiomètre.
- **Vled** : différence de potentiel permettant de commander le rétro éclairage afin d'assurer la visibilité dans l'obscurité.
- **E** : entrée de validation (ENABLE), elle permet de valider les données sur un front descendant. Lorsque E=0 alors le bus de données est à l'état haute impédance.
- **RS** : (Registre Select) cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (RS=0) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (RS=1).
- **R/W** : entrée de lecture (R/W=1) et d'écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.
- **D7...D0** : bus de données bidirectionnel, il permet de transférer les instructions ou les données à l'afficheur LCD. [26]

3.1.4 Branchements afficheur LCD - microcontrôleurs

Les fils des données (D0, D1, D2, D3) et R/W (Read Write) sont branchées à la masse tandis que (D4, D5, D6, D7) et RS (Registre Select) et E (Enable) sont branchées aux sorties du microcontrôleur [27].

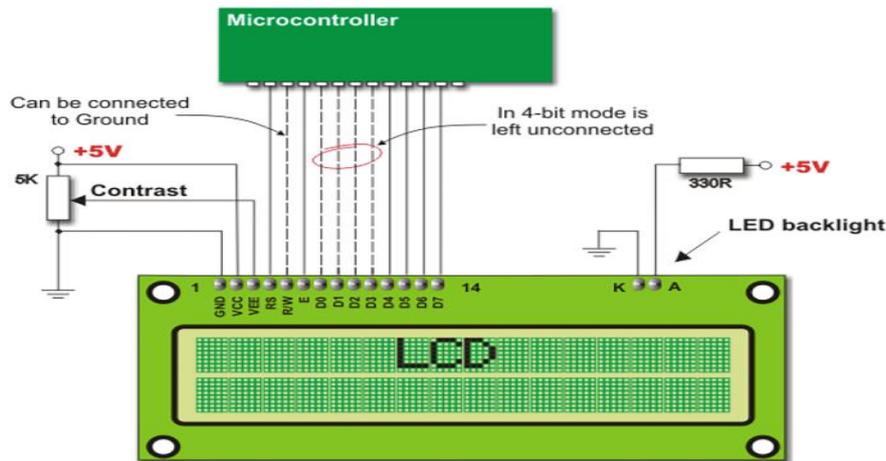


Figure III. 4 : Branchement de L'afficheur LCD avec le PIC16F877A.[28]

3.2 Le PIC 16F877A :

PIC16F877A est le microcontrôleur que nous avons utilisé dans notre projets. Il permet le contrôle et l'affichage des éléments sur l'afficheur LCD. Ce PIC est un circuit intégré contenu dans un boîtier nommé « DIL 40 », il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1ere broche est placé dans le coin situé à gauche de l'encoche de repérage. Certaines fonctionnalités de ce microcontrôleur ainsi que ça sa structure interne et externe sont décrites et détaillées dans le chapitre suivant [29].

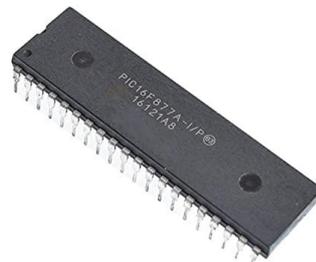


Figure III. 5 : PIC16F877A [29]

3.3 Le capteur de température TC74 :

Le capteur de température TC74, produit par « Microchip Technology », est un capteur numérique (digital) sur 8 bits. Le TC74 acquiert et convertit la température avec une résolution de $\pm 1^{\circ}\text{C}$. Il stocke les données dans un registre interne. Le TC74 comporte un mode 'attente' (standby) de faible consommation quand l'acquisition de la température est suspendue.[30]

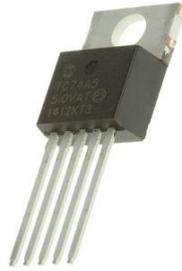


Figure III. 6 : Capteur de température TC74.

3.3.1 Caractéristiques techniques :

Attribut	Valeur
Fonction du capteur	Température
Type de sortie	Numérique
Type d'interface	Série-I2C, SMBus
Précision	$\pm 3^{\circ}\text{C}$
Type de montage	Traversant
Nombre de broches	5
Température de fonctionnement minimum	-40°C
Plage de température de fonctionnement	-40 à $+125^{\circ}\text{C}$.
Température d'utilisation maximum	$+125^{\circ}\text{C}$
Tension d'alimentation de fonctionnement minimum	2,7 V
Tension d'alimentation fonctionnement maximum	5,5 V
Type de boîtier	A-220
Longueur	10.54mm
Hauteur	15.57mm
Série	TC74
Largeur	4.7mm
Dimensions	10.54 x 4.7 x 15.57mm
Résolution	$\pm 1^{\circ}\text{C}$

Tableau 1: Caractéristiques techniques.[31]

3.4 L'EEPROM 24C02 :

L'AT24C02 est une puce de mémoire morte programmable (EEPROM) effaçable électriquement. Il est organisé en interne avec 32 pages de 8 octets chacune, il a 2 Kbits de mémoire. Il s'agit de l'EEPROM la plus couramment utilisée, il est livré avec un DIP à 8 broches,[32] illustré à la figure:

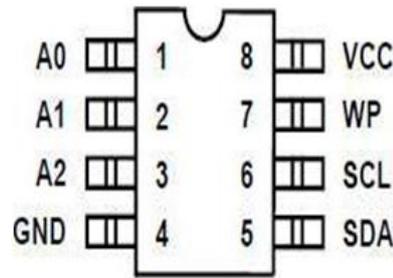


Figure III. 7 : EEPROM 24C02. [32]

Broche 1-3 : A0, A1, A2 sont les entrées d'adresse de la puce, dans ce A1 et A2 sont pour l'adressage et A0 est une broche NA (pas de connexion). Les huit appareils 2K peuvent être adressés sur un seul système de bus.

Broche 4 : Terre (GND).

Broche 5 : Il s'agit d'une broche de données série, elle est bidirectionnelle pour le transfert de données série.

Broche 6 : C'est une entrée d'horloge série, fournit les signaux d'horloge positifs.

Broche 7 : C'est la broche de protection en écriture, fournit la protection des données matérielles. Il permet une opération de lecture / écriture lorsqu'il est connecté à la broche de terre.

Broche 8 : Source de courant.

3.5 *Le module DS1307*

Le module DS1307 de Maxim Integrated est une horloge temps réel (aussi appelé "RTC", AKA "Real Time Clock"). C'est une horloge numérique autonome qui donne l'heure quand on la lui demande. Ce genre d'horloge est très utile dans des projets de mesure de grandeurs physiques.

Ce module RTC est capable de gérer l'heure (heures, minutes, secondes) et la date (jours, mois, année) tout en s'occupant des mois de 30 ou 31 jours, des années bissextiles, etc. Le calendrier intégré dans le module DS1307 est valable de l'an 2000 à l'an 2100, ce qui devrait être suffisant pour la plupart des projets.

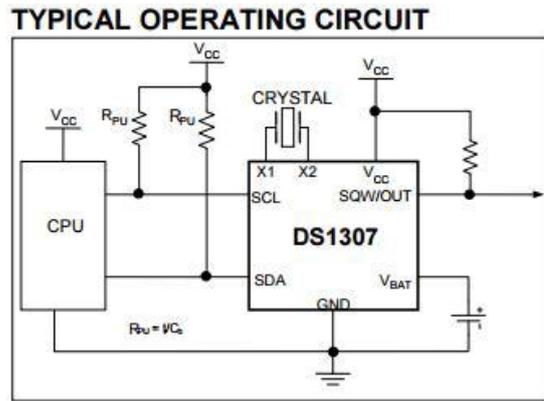


Figure III. 8 : Schéma d'application typique .

La communication avec le microcontrôleur maître se fait via un bus I2C. Le module dispose de tout le nécessaire pour garder en mémoire l'heure en cas de coupure d'alimentation grâce à une batterie externe. Une simple pile bouton permet de garder l'heure et la date à jour durant plusieurs années sans alimentation.

Le module DS1307 ne dispose pas de fonctionnalité "alarme" contrairement à d'autres modules RTC plus haut de gamme. Le module DS1307 dispose cependant d'une sortie "base de temps" permettant d'avoir un signal logique à une fréquence fixe (1 Hertz par exemple) pour faire fonctionner un circuit ou un compteur externe. Cela peut être utile dans certaines applications.

[33].

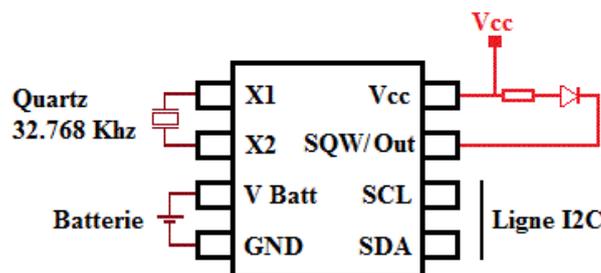


Figure III. 9 Câblage DS1307

La programmation ou la lecture du DS1307 s'effectue via des registres 8 bits en BCD. Ces registres sont contenus dans les adresses :

- 00h pour les secondes
- 01h pour les minutes
- 02h pour les heures
- 03h pour les jours
- 04h pour la date
- 05h Pour le mois
- 06h pour l'année
- Un autre registre est disponible à l'adresse 07h est permet soit d'avoir une sortie logique

Soit de générer un signal carré de fréquence variable

- Une mémoire de 56 bits est disponible aux adresses comprises entre 08h et 3Fh (8 à 63 en décimale) [34]

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
00h	CH	(10) Secondes			Secondes				Secondes	00 à 59
01h	0	(10) Minutes			Minutes				Minutes	00 à 59
02h	0	12	PM / AM	(10) Heure	HEURES				Heure	1 à 12 +AM/PM 00 à 23
		24	(10) Heure							
03h	0	0	0	0	0	JOURS			Jour	01 à 07
04h	0	0	(10) Date		Date				Date	01 à 31
05h	0	0	0	(10) Mois	MOIS				Mois	01 à 12
06h	(10) Année				Année				Année	00 à 99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Contrôle	-----
08h à 3Fh									RAM 56 * 8	00h à FFh

Tableau 2: registres du DS1307

4 Description logiciels

4.1 Le logiciel ISIS PROTEUS

4.1.1 Présentation générale

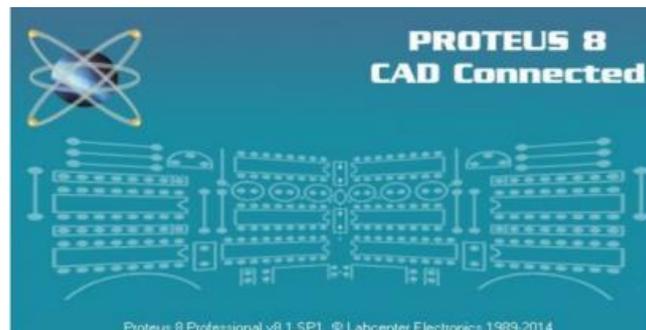


Figure III. 10 Le logiciel proteus

Le logiciel PROTEUS est un logiciel composé de nombreux modules. En général, on utilise souvent les deux principaux logiciels qui permettent de faire une Conception Assistée par Ordinateur ou Electronic Design Automation (EDA). Il est édité par la société LabcenterElectronics. Ses deux logiciels principaux sont :

- L'éditeur de schéma ISIS.
- L'outil de conception de circuit imprimé ARES.

Dans l'environnement électronique, PROTEUS est très apprécié. Grâce à sa capacité de faire combiner plusieurs circuits. En plus de la popularité de ses outils, PROTEUS possède aussi d'autres avantages très intéressants. Il est un outil de création de prototype virtuel permettant de réduire les coûts matériels et logiciels lors de la conception d'un projet. De plus, il a des supports techniques très performants et il contient aussi des packs

logiciels qui sont faciles et rapides à manipuler [35].

Le logiciel ISIS (Intelligent Schematic Input System) est principalement connue pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui nous permet de déceler certaines erreurs dès l'étape de conception.

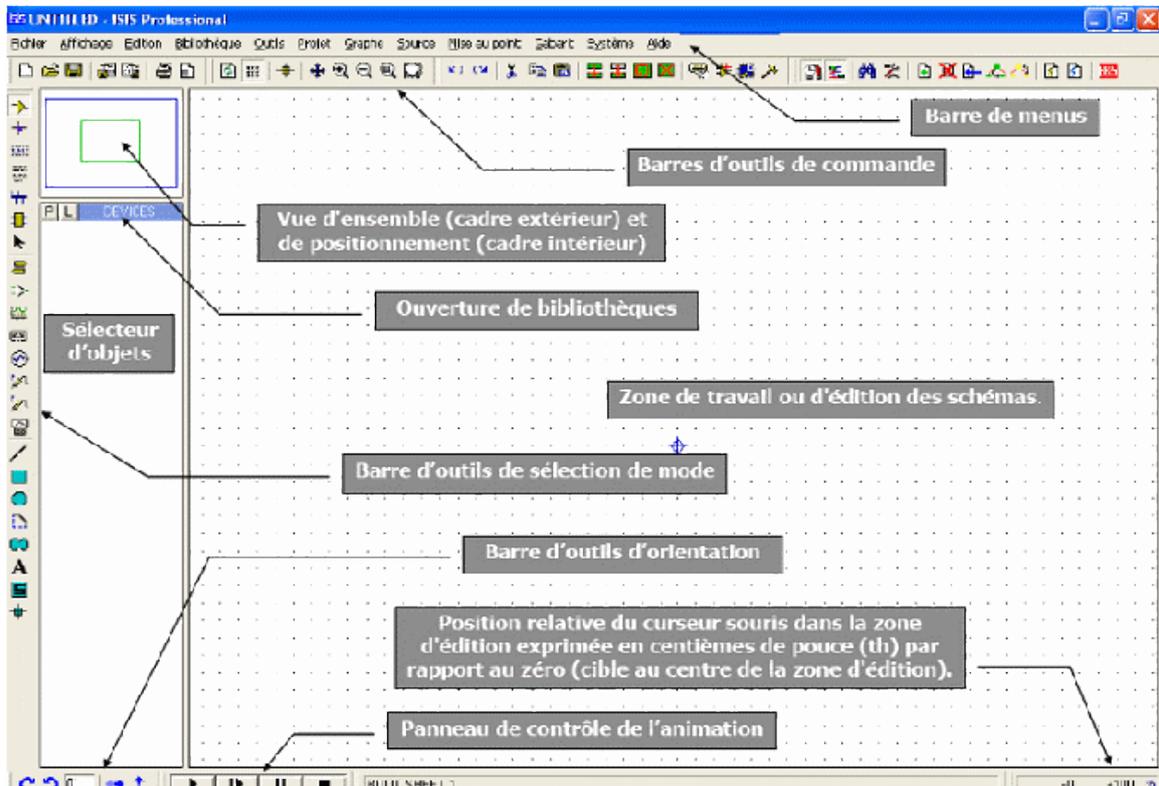


Figure III. 11 Présentation d'ISIS de PROTEUS

4.1.2 Comment créer un nouveau projet sur ISIS PROTEUS ?

Pour créer un nouveau projet, on clique sur «file-new design» puis on clique sur «Save design» et on donne le nom et l'emplacement de fichier sur PC. Une fois le nouveau projet est créé, on peut commencer la conception de schéma électrique de projet en ajoutant les composants directement de la bibliothèque de composants. Afin de tester le montage que nous avons fait, il suffit juste de démarrer la simulation en cliquant sur «Run the simulation »,

Proteus dispose aussi d'une boîte à outils pour la mesure (Ampèremètre, Voltmètre, Oscilloscope, ...). Pour passer de schéma électrique au design PCB on clique sur le bouton « ARES » dans la barre d'options, une autre fenêtre apparaît contenant les composants que nous avons utilisés dans le schéma avec toutes les connexions que nous avons fait. Il suffit juste de commencer le routage et la structuration de la carte PCB de la façon que nous souhaitons [35].

4.2 Le logiciel MikroC

4.2.1 Présentation du MikroC

Le « MikroC » est un compilateur pour PIC conçu par la société « Mikroelektronika », il bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, gestionnaire pour afficheurs 7 segments, correcteur d'erreur, explorateur de code...); Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, SPI, RS485, afficheurs LCD alphanumériques et graphiques...) de ce fait il est un des outils de développement incontournable et puissant. Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète[36].

4.2.2 Création d'un projet

Pour la création d'un nouveau projet, le saisi et la compilation du programme nous avons suivi les étapes suivantes :

- Double clique sur l'icône du logiciel Mikro C PRO FOR PIC pour le lancer.
- Ensuite on clique sur « new project » pour créer un nouveau projet.

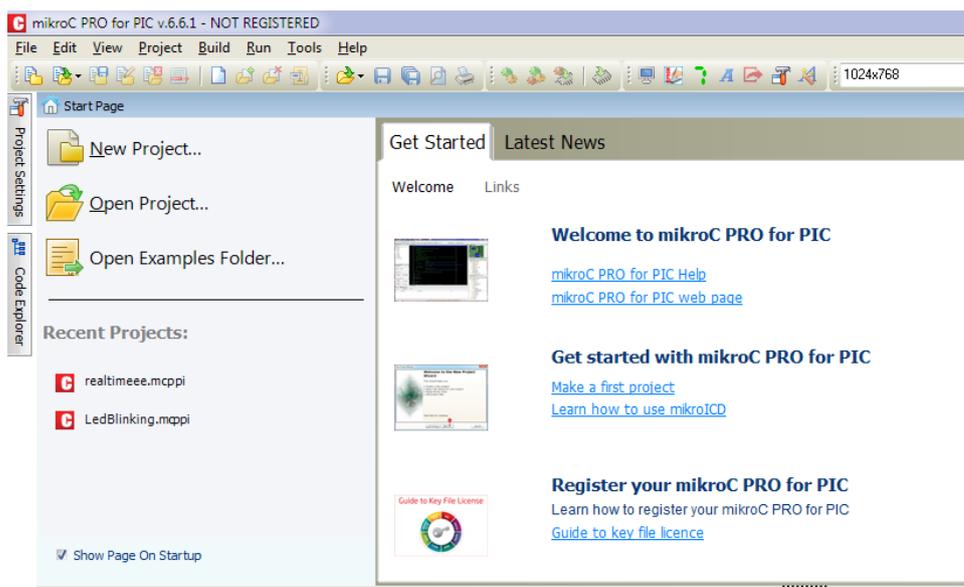


Figure III. 12 Création d'un nouveau projet

- Après l'action précédente une fenêtre apparaîtra, celle-ci va nous permettre de :
 - Nommer notre projet.
 - Choisir son emplacement.
 - Sélectionner la référence du pic avec lequel on souhaite travailler donc on sélectionne le P16F877A.
 - Définir la fréquence d'oscillation avec laquelle travaille notre pic, dans ce cas on sélectionne 8.000000 MHz.

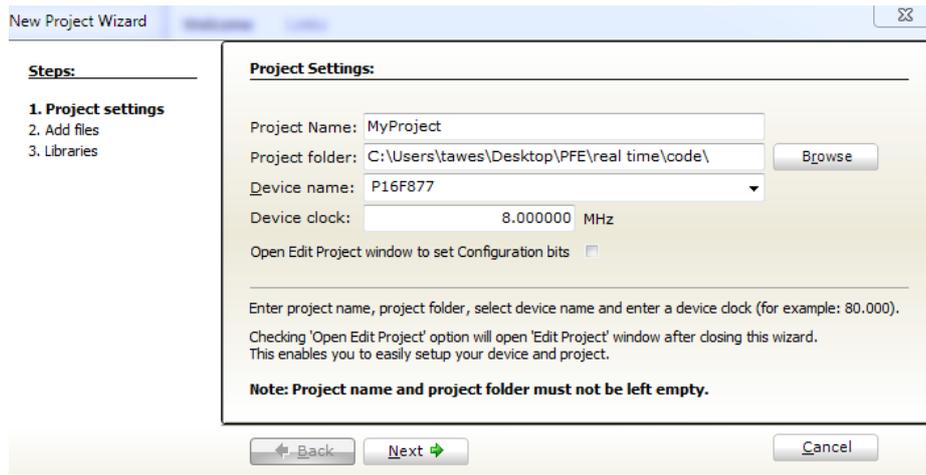


Figure III. 13 Les configurations de projet

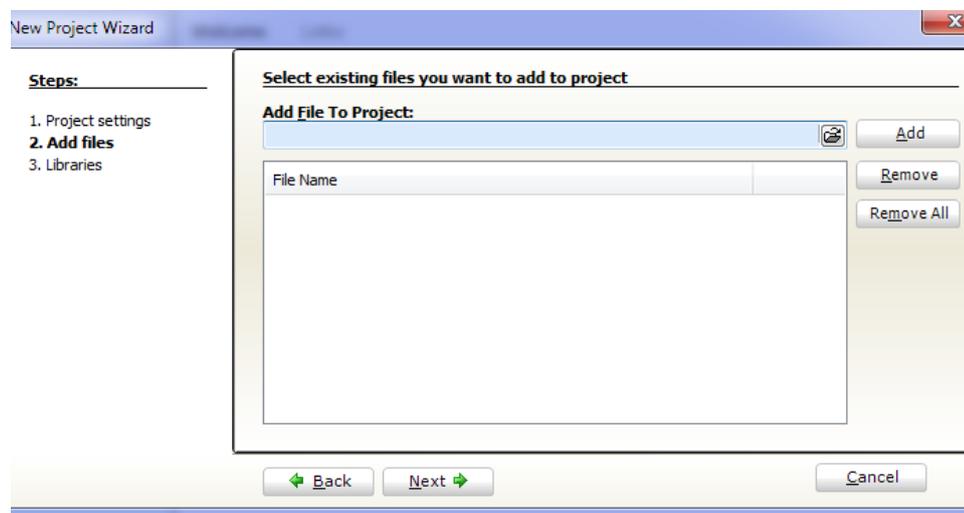


Figure III. 14 L'emplacement du projet

- On appui sur « Next » et enfin notre projet et crée et la configuration est terminé, maintenant cette fenêtre s'affiche pour qu'on puisse écrire notre programme dedans.

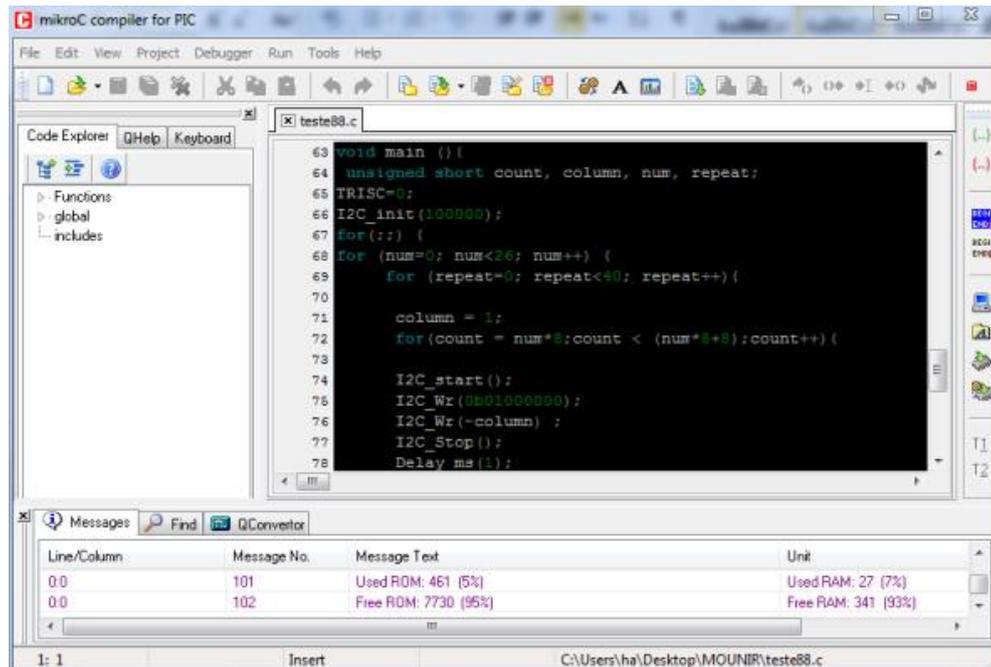


Figure III. 15 Fenêtre de la saisie du programme

4.2.3 Compilation

Lorsque vous avez créé le projet et écrit le code source, il est temps de le compiler. Sélectionnez ProjectBuild à partir du menu déroulant ou cliquez sur l'icône Build dans la barre d'outils du projet. Si plus d'un projet est ouvert, vous pouvez compiler tous ouverts projets en sélectionnant Project > Build All dans le menu déroulant, ou cliquez sur l'icône  de la barre d'outils du projet [36].

4.2.4 Les bibliothèques du MikroC :

MikroC fournit quelques bibliothèques et nous avons utilisé uniquement les bibliothèques suivantes :

Bibliothèque	Les fonctions	La description
I2C	I2C_Init	Initialise I2C avec horloge souhaitée Exemple : I2C_Init(100000);
	I2C_Start	Détermine si I2C bus est libre et lancer un signal START. Exemple : I2C_Start ();
	I2C_Repeated_Start	Répéter un signal START. Exemple : I2C_Repeated_Start ();
	I2C_Rd	Lit un octet de l'esclave, et envoie pas signal d'acquittement ACK si le paramètre est 0, sinon il envoie reconnaître. Exemple : take = I2C_Rd(0);
	I2C_Wr	Envoie octet de données (données de paramétrage) par l'intermédiaire du bus I2C. Exemple : I2C_Write(0xA3);
	I2C_Stop	Lancer un signal STOP. Exemple : I2C_Stop();

Tableau 3 : Bibliothèque du MikroC

5 Bus I2C et MikroC :

MikroC fournit une bibliothèque prête du bus I2C (Tableaux 3), nous utilisons cette bibliothèque pour communiquer entre le maître (PIC16F877A) et l'esclave (DS1307).

6 Conclusion

Dans ce chapitre nous avons présenté les outils utilisés dans notre projet, qui sont très importants pour accomplir notre travail en général la partie matérielle et spécifiquement les logiciels d'assimilation et de programmation. Nous avons vu aussi la bibliothèque du bus I2C. Le chapitre suivant sera consacré à la réalisation de notre projet.

Chapitre 4 :

Conception et réalisation pratique

1 Introduction :

Dans les chapitres précédents nous avons fait une étude générale de tous les éléments constituant notre projet et c'est ce qui nous a facilité la réalisation pratique.

Le système électronique proposé dans ce projet consiste à réguler une température spécifique, l'heure et la date détaillée dans un environnement industriel, commandés par un microcontrôleur PIC16F877A en utilisant le protocole de communication I2C.

Ce chapitre sera consacré à la description des différentes étapes de conception et réalisation physique de notre projet.

2 Description du système :

2.1 Schéma synoptique :

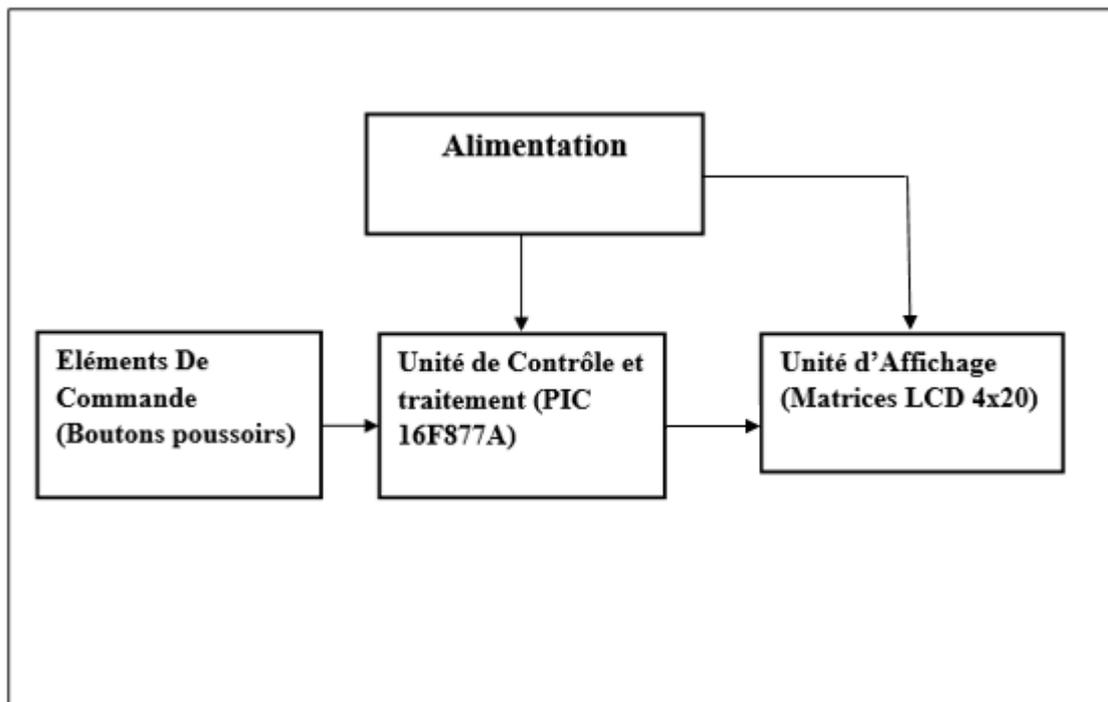


Figure IV. 1 Schéma synoptique du circuit

2.2 Présentation des blocs :

Le schéma de la figure ci-dessus donne une vue générale sur le principe de fonctionnement de notre montage, il est articulé par quatre principales parties fonctionnelles :

- Alimentation : une batterie de 9 volts et une tension de 5 volts pour alimenter le PIC et les autres circuits.
- Eléments de commande : boutons poussoirs.
- Unité de contrôle de traitement : PIC16F877A
- Système de régulation : résistance de réchauffement + ventilateur

- Unité d'affichage : Matrices LED 4x20.

3 Simulation du circuit :

Avant de commencer à implémenter réellement notre système, nous nous sommes tournés vers des simulations de différentes parties du système. Pour cela, nous utilisons le logiciel ISIS Proteus, qui est un très bon logiciel de simulation électronique.

3.1 Schéma électronique :

Afin de créer un schéma électronique de notre carte, nous avons utilisé le logiciel ISIS Proteus. Pour le saisir, nous avons créé un nouveau projet à l'écran, puis placé les composants que nous avons sélectionné dans la bibliothèque de composants dans l'espace de travail.

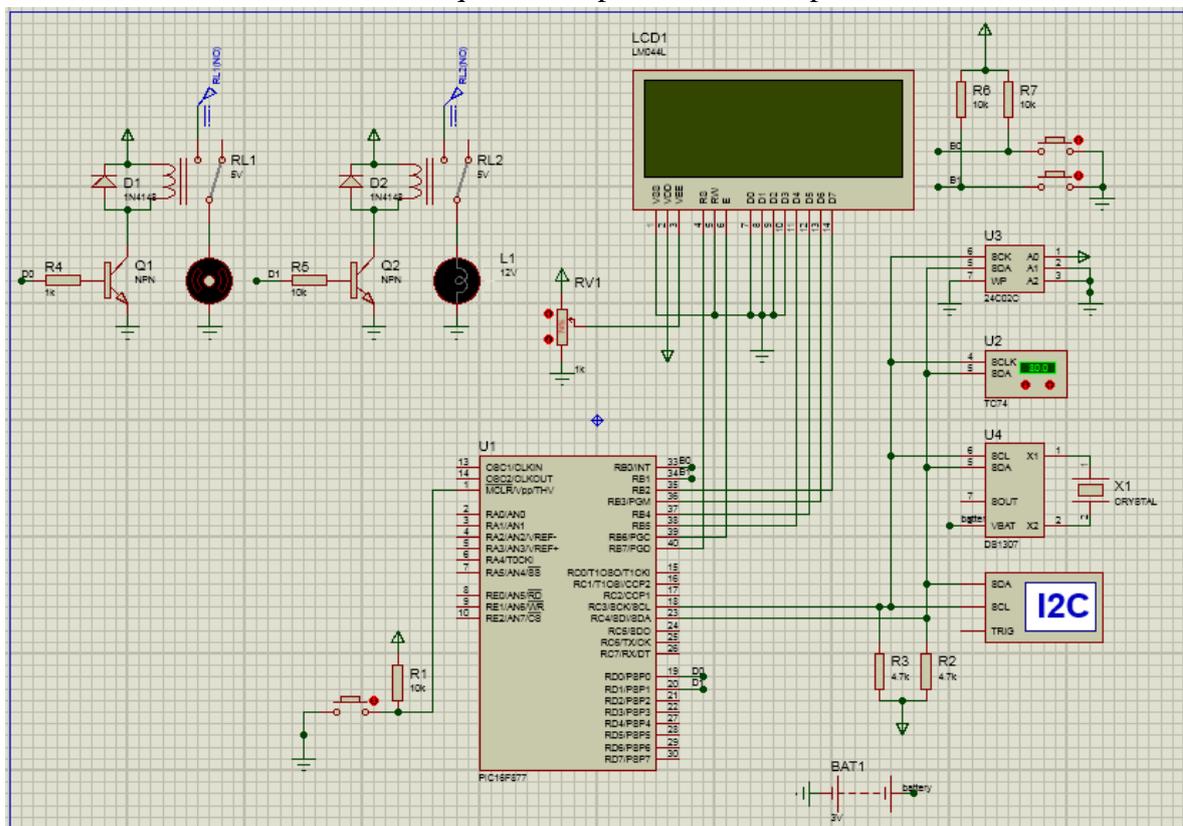


Figure IV. 2 Schéma électronique globale du circuit

3.2 Programmation du pic 16F877 :

Pour la programmation de notre microcontrôleur nous avons choisi la façon la plus simple et efficace et cela en utilisant le compilateur MikroC pro for PIC.

Pour la création d'un nouveau projet, la saisie et la compilation du programme, les étapes ont été déjà expliquées dans le troisième chapitre.

Après la création du programme, on doit l'associer en passant par les étapes suivantes :

- On revient à la zone de travail sur l'écran ISIS.
- Double clic sur le pic 16F877A

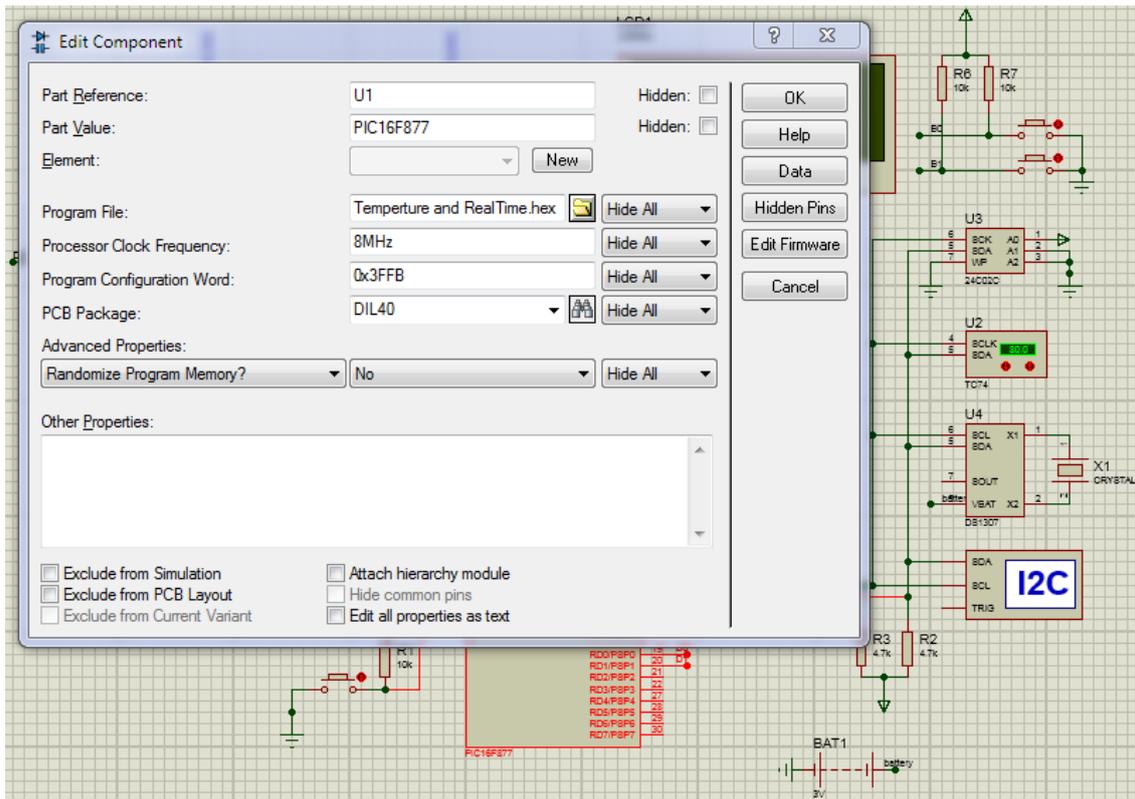


Figure IV. 3 Chargement du programme dans le PIC sur ISIS

On entre sur (program file) et on apporte le fichier (file.HEX) puis on appui sur ok. -Le programme est chargé dans le PIC16F84. -On clique sur le bouton (play) pour commencer la simulation.

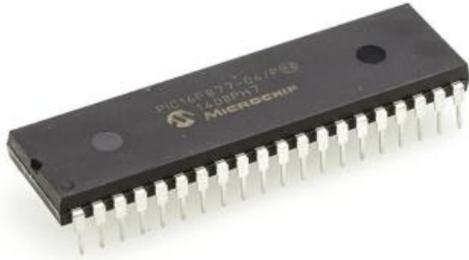
3.3 Routage et création du circuit imprimé :

Avant de passer à la réalisation matérielle de notre carte électronique nous allons d'abord réaliser le schéma du circuit imprimé et cela en utilisant le logiciel ARES pour simuler les connexions de tous les composants implantés sur notre carte.

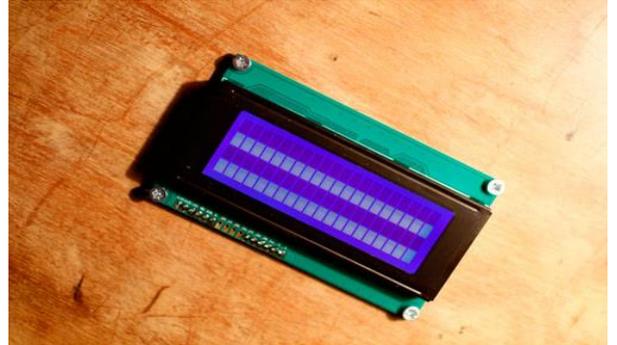
4 Réalisation pratique :

4.1 Listes des composants :

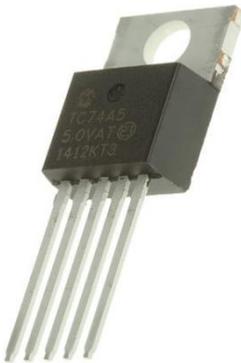
- Un microcontrôleur PIC16F877A :



- LCD 4X20



- TC74 :



- EEPROM 24C02



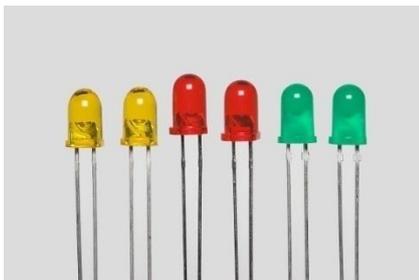
- Module RTC-DS1307



- Quartz 8MHz



- LEDs



- Potentiometer pot-Hg 1K



- Resistances



4.2 Gravure du programme sur le pic :

Après la simulation du programme avec le logiciel mikroC for pic pro, il est prêt à être chargé dans le pic et pour le faire nous avons utilisé le logiciel PICkit2 qui nous a permis de transférer le programme en hexadécimale vers le pic. Ensuite nous avons vérifié en cliquant sur « verify » pour s'assurer du transfert complet du programme.



Figure IV. 4 Gravure du programme sur le Pic

4.3 Réalisation du circuit imprimé et implantation des composants :

Après la conception logiciel du circuit et le routage nous avons obtenu le schéma du circuit imprimé, qui nous a ensuite permis d'imprimer le typon sur un transparent spécial et grâce à une imprimante laser.

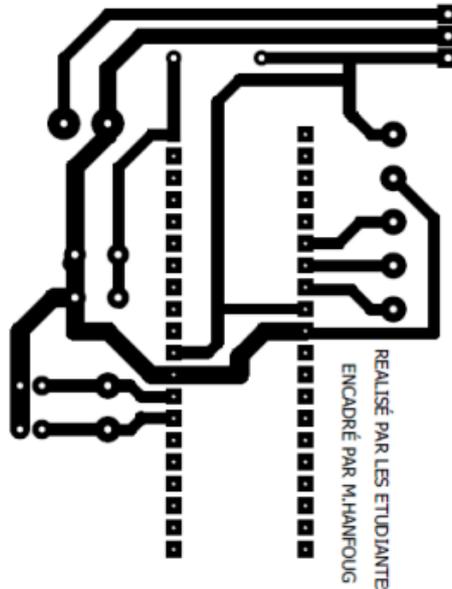


Figure IV. 5 Le typon

On va donc déposer le typon sur la face cuivrée de la carte époxy et démarrer le transfert du circuit en posant un fer à repasser bien chauffée sur le typon. Après cette opération il faut la refroidir, élever le typon et sécher la carte. On passe ensuite à la gravure, qui consiste en l'élimination du cuivre inutile afin de n'avoir que les pistes de connexion qui nous seront utiles. La gravure pourra se faire avec de l'acide ou au moyen d'une insoleuse UV. Puis on doit percer les trous du montage pour pouvoir placer les composants. Maintenant que notre circuit imprimé est terminé, on dépose les composants sur la plaque tout en vérifiant le positionnement et le bon sens de chacun d'entre eux, on procède ensuite à la soudure des composants, et enfin notre carte électronique est prête.

5 Conclusion :

Au niveau de ce dernier chapitre nous avons présenté les différentes étapes de réalisation pratique de notre projet, cela nous fait une expérience très enrichissante, à savoir la programmation du PIC16F877A avec le logiciel mikroC, la conception grâce à PROTEUS qui permet de réaliser facilement des schémas fonctionnels et électriques puis de gérer le routage du circuit. Ensuite la fabrication de la carte électronique en passant par plusieurs étapes.

Conclusion générale :

Notre objectif dans ce présent travail consiste en la conception et la réalisation d'un système à base d'un PIC 16F877A, en vue d'améliorer le contrôle et le maintien de la température et l'affichage précis de la date et heure dans un milieu industriel.

L'étude du thème mise en œuvre du protocole de communication I2C dans l'environnement microcontrôleur PIC16F877A déroulée en quatre parties (mise en œuvre du microcontrôleur PIC16F877A, le protocole I2C, présentation et description des outils du projet, , et la conception et réalisation pratique) nous a montré que tout étude de système technique suit une démarche méthodique constituée de différentes parties aussi importantes les unes que les autres.

Ce projet nous a permis d'approfondir nos connaissances en programmation des systèmes à microcontrôleurs pic en langage C ainsi la maîtrise de la gestion des périphériques d'entrées et sorties, il nous a aussi permet la maîtrise d'utilisation du logiciel PROTEUS, et à affronter les difficultés liées à la conception et la réalisation d'un appareillage électronique.

La chaine a été simulée sans le logiciels « porteuse ISIS » le schéma a été validé conformément aux résultats optime. Le programme est développé en langage C, compilé par mikroC est également validé.

Malheureusement, nous n'avons pas pu accomplir notre réalisation électronique par motif de l'absence de quelques composants indispensables pour le projet sur le marché algérien et le manque de temps pour en commander de l'étranger et finir la conception sur les délais.

Ce travail reste, comme toute œuvre humaine, incomplète et perfectible, nous recommandons d'en améliorer la conception et pour cela nous proposons ci-dessous des améliorations pour les futurs développeurs :

- Utilisation d'une carte Arduino au lieu du microcontrôleur 16F877A.
- Contrôler d'autres paramètres comme l'humidité.
- Insérer un système de sécurité en cas de défaillance du système.

Annexes

Annexe 1 : PIC16F877A Data Sh



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

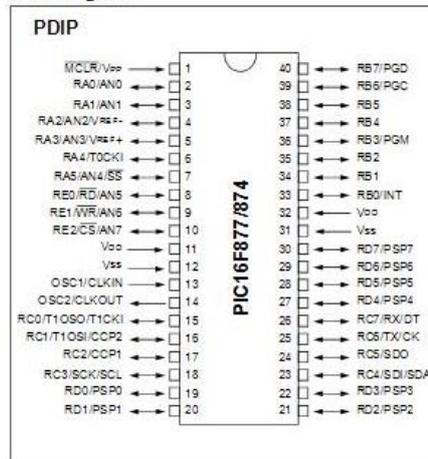
Devices Included in this Data Sheet:

PIC16F873	PIC16F876
PIC16F874	PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
- Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external \overline{RD} , \overline{WR} and \overline{CS} controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Annexe 2 : TC74 Data Sheet



TC74

Tiny Serial Digital Thermal Sensor

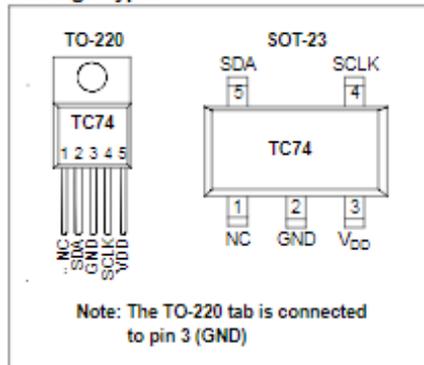
Features

- Digital Temperature Sensing in SOT-23-5 or TO-220 Packages
- Outputs Temperature as an 8-Bit Digital Word
- Simple SMBus/I²C™ Serial Port Interface
- Solid-State Temperature Sensing:
 - $\pm 2^{\circ}\text{C}$ (max.) Accuracy from $+25^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
 - $\pm 3^{\circ}\text{C}$ (max.) Accuracy from 0°C to $+125^{\circ}\text{C}$
- Supply Voltage of 2.7V to 5.5V
- Low Power:
 - 200 μA (typ.) Operating Current
 - 5 μA (typ.) Standby Mode Current

Applications

- Thermal Protection for Hard Disk Drives and other PC Peripherals
- PC Card Devices for Notebook Computers
- Low Cost Thermostat Controls
- Power Supplies
- Thermistor Replacement

Package Types



General Description

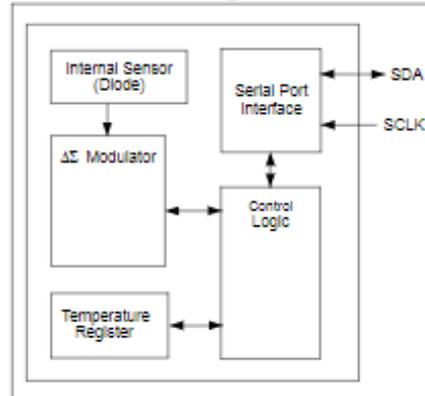
The TC74 is a serially accessible, digital temperature sensor particularly suited for low cost and small form-factor applications. Temperature data is converted from the onboard thermal sensing element and made available as an 8-bit digital word.

Communication with the TC74 is accomplished via a 2-wire SMBus/I²C compatible serial port. This bus also can be used to implement multi-drop/multi-zone monitoring. The SHDN bit in the CONFIG register can be used to activate the low power Standby mode.

Temperature resolution is 1°C . Conversion rate is a nominal 8 samples/sec. During normal operation, the quiescent current is 200 μA (typ). During standby operation, the quiescent current is 5 μA (typ).

Small size, low installed cost and ease of use make the TC74 an ideal choice for implementing thermal management in a variety of systems.

Functional Block Diagram



Annexe 3: DC1307 Data Sheet

DS1307 64 x 8, Serial, I²C Real-Time Clock

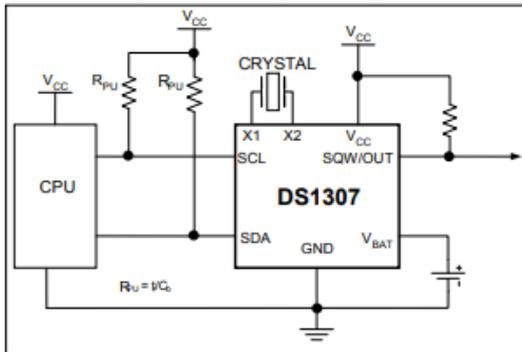
GENERAL DESCRIPTION

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.

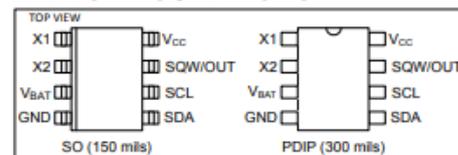
FEATURES

- 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

Annexe 4 : EEPROM 24C04 Data Sheet

SERIAL 2K (256 x 8) EEPROM

NOT FOR NEW DESIGN

- 1 MILLION ERASE/WRITE CYCLES with 40 YEARS DATA RETENTION
- SINGLE SUPPLY VOLTAGE:
 - 3V to 5.5V for ST24x02 versions
 - 2.5V to 5.5V for ST25x02 versions
 - 1.8V to 5.5V for ST24C02R version only
- HARDWARE WRITE CONTROL VERSIONS: ST24W02 and ST25W02
- TWO WIRE SERIAL INTERFACE, FULLY I²C BUS COMPATIBLE
- BYTE and MULTIBYTE WRITE (up to 4 BYTES)
- PAGE WRITE (up to 8 BYTES)
- BYTE, RANDOM and SEQUENTIAL READ MODES
- SELF TIMED PROGRAMMING CYCLE
- AUTOMATIC ADDRESS INCREMENTING
- ENHANCED ESD/LATCH-UP PERFORMANCES
- **ST24C/W02 are replaced by the M24C02**
- **ST25C/W02 are replaced by the M24C02-W**
- **ST24C02R is replaced by the M24C02-R**

DESCRIPTION

This specification covers a range of 2K bits I²C bus EEPROM products, the ST24/25C02, the ST24C02R and ST24/25W02. In the text, products are referred to as ST24/25x02, where "x" is: "C" for Standard version and "W" for hardware Write Control version.

Table 1. Signal Names

E0-E2	Chip Enable Inputs
SDA	Serial Data Address Input/Output
SCL	Serial Clock
MODE	Multibyte/Page Write Mode (C version)
WC	Write Control (W version)
V _{CC}	Supply Voltage
V _{SS}	Ground

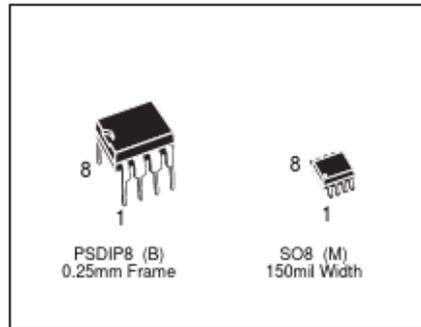
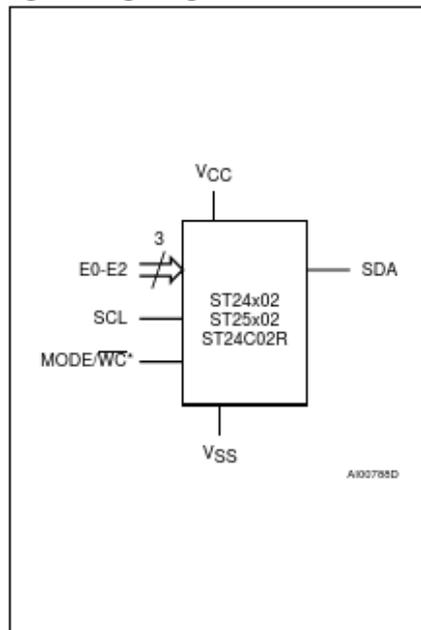


Figure 1. Logic Diagram



Note: WC signal is only available for ST24/25W02 products.

Figure 2A. DIP Pin Connections

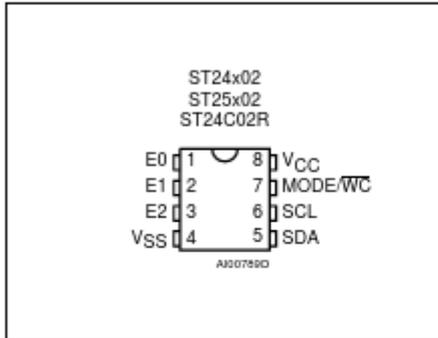


Figure 2B. SO Pin Connections

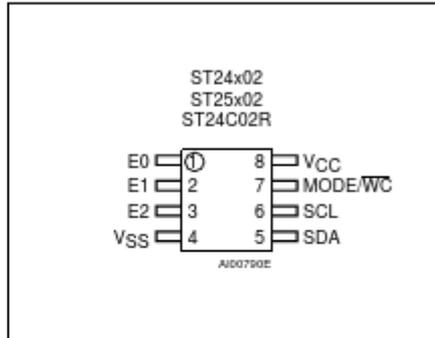


Table 2. Absolute Maximum Ratings ⁽¹⁾

Symbol	Parameter	Value	Unit
T _A	Ambient Operating Temperature	-40 to 125	°C
T _{STG}	Storage Temperature	-65 to 150	°C
T _{LEAD}	Lead Temperature, Soldering (SO8 package) 40 sec (PSDIP8 package) 10 sec	215 260	°C
V _{IO}	Input or Output Voltages	-0.6 to 6.5	V
V _{CC}	Supply Voltage	-0.3 to 6.5	V
V _{ESD}	Electrostatic Discharge Voltage (Human Body model) ⁽²⁾	4000	V
	Electrostatic Discharge Voltage (Machine model) ⁽³⁾	500	V

Références

- [1] « Cours Microcontrôleurs ». <https://www.technologuepro.com/cours-genie-electrique/cours-32-microcontroleurs/> (consulté le 24 juin 2022).
- [2] R. Mallard, *Les microcontrôleurs PIC pour les débutants qui veulent programmer sans patauger: Avec mikroPascal : intermédiaire idéal entre BASIC*. Roissy: PUBLITR ELEKTOR, 2014.
- [3] « chapitre-2-microcontroleur-pic-16f877.pdf ». Consulté le: 16 août 2022. [En ligne]. Disponible sur: <https://www.technologuepro.com/microcontroleur-2/chapitre-2-microcontroleur-pic-16f877.pdf>
- [4] M. H. Sarhene, « Cours Microcontrôleur PIC16F877 », *Technologue Pro*. <http://www.technologuepro.com/cours-genie-electrique/cours-56-microcontroleur-pic16f877/> (consulté le 24 juin 2022).
- [5] « MICROCONTROLEURS LE PIC 16F876/877. Famille Mid-Range de Microchip. A. Oumnad - PDF Free Download ». <https://docplayer.fr/18867066-Microcontroleurs-le-pic-16f876-877-famille-mid-range-de-microchip-a-oumnad.html> (consulté le 24 juin 2022).
- [6] « Smart | Connected | Secure | Microchip Technology ». <https://www.microchip.com/> (consulté le 24 juin 2022).
- [7] S. Selma, « Alimentation électrique », Consulté le: 24 juin 2022. [En ligne]. Disponible sur: https://www.academia.edu/38462539/Alimentation_electrique
- [8] « Academia.edu - Share research ». <https://www.academia.edu/> (consulté le 24 juin 2022).
- [9] « V. Chollet - cours-pic-13b-trous - 09/12/2012 Page 1 sur 48 MICROCONTROLEURS PIC PROGRAMMATION EN C - PDF Free Download ». <https://docplayer.fr/31477266-V-chollet-cours-pic-13b-trous-09-12-2012-page-1-sur-48-microcontroleurs-pic-programmation-en-c.html> (consulté le 24 juin 2022).
- [10] « Master: Réalisation D'un Prototype Permettant Le Contrôle Des Deux Modes Starting/Generation À Bord de l'ATR 72-500 | PDF | Aviation | Avion », *Scribd*. <https://fr.scribd.com/document/479478164/014-2017> (consulté le 24 juin 2022).
- [11] A. Hmidene, « Agrégé en Génie Electrique », p. 14.
- [12] « razafiariseraRalayT_PC_M2_07.pdf ». Consulté le: 24 juin 2022. [En ligne]. Disponible sur: http://biblio.univ-antananarivo.mg/pdfs/razafiariseraRalayT_PC_M2_07.pdf
- [13] R. R. Tiana, « RESISTIVIMETRE A BASE DE PIC 16F877 Présenté par », p. 81.
- [14] R. I. Finaritra, « Leçon N°2 INTRODUCTION SUR LES MICROCONTROLEURS », Consulté le: 24 juin 2022. [En ligne]. Disponible sur: https://www.academia.edu/35926844/Le%C3%A7on_N_2_INTRODUCTION_SUR_LES_MICROCONTROLEURS
- [15] « Cours-Microcontrôleur-microprocesseur-83.pdf ». Consulté le: 24 juin 2022. [En ligne]. Disponible sur: <https://www.electronique-mixte.fr/wp-content/uploads/2018/08/Cours-Microcontr%C3%B4leur-microprocesseur-83.pdf>
- [16] PARET, *LE BUS I2C De la theorie a la pratique*. Dunod, 1995.
- [17] C. Diou et U. de Metz, « Présentation du bus I2C Le protocole I2C Les conflits Spécificités et évolutions », p. 33.
- [18] « 8-bus_I2C.pdf ».
- [19] « Protocole I2C | Courbon Michel ». https://perso.univ-st-etienne.fr/courbon/docs/Teaching/TP_BusI2C/busi2c/2-ProtocoleI2C/ (consulté le 22 juin 2022).
- [20] « Le bus I²C ». <https://www.aurel32.net/elec/i2c.php> (consulté le 22 juin 2022).
- [21] « bus protocole specifications i2c ». <http://electro8051.free.fr/I2C/busi2c.htm> (consulté le 16 août

- 2022).
- [22] « Cours Systèmes Embarqués: BUS I2C ». <https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-Bus-I2C.htm> (consulté le 22 juin 2022).
- [23] « Mise en œuvre du bus I2C avec le PIC 16F876A ». http://fabrice.sincere.pagesperso-orange.fr/cm_electronique/projet_pic/aidememoire/16F876A_bus_I2C/bus_I2C_16F876A.htm (consulté le 16 août 2022).
- [24] « Utiliser un LCD, montage électronique ». <https://www.scriptol.fr/montages/salut-le-monde.php> (consulté le 24 juin 2022).
- [25] « Les afficheurs LCD dans ISIS Proteus ». <http://fractale.gecif.net/si/logiciels/proteus/lcd/> (consulté le 24 juin 2022).
- [26] « 809.pdf ».
- [27] I. Douchi, L. Belkacemi, et M. (promoteur) Hireche, « Permettant le Contrôle des Deux Modes Starting/Generation À Bord de l'ATR 72-500 », Thesis, 2017. Consulté le: 24 juin 2022. [En ligne]. Disponible sur: <https://di.univ-blida.dz/jspui/handle/123456789/1067>
- [28] « Description of Microcontroller Circuit and MikroC Program | SpringerLink ». https://link.springer.com/chapter/10.1007/978-3-030-92628-1_5 (consulté le 13 septembre 2022).
- [29] M. H. Sarhene, « Cours Microcontrôleur PIC16F877 », *Technologue Pro*. <http://www.technologuepro.com/cours-genie-electrique/cours-56-microcontrolleur-pic16f877/> (consulté le 24 juin 2022).
- [30] A. Zouinkhi, E. Bajic, et M. B. Gayed, « Concept de Produit Actif par Réseaux de Capteurs pour Contrôler la Sécurité des Produits Chimiques », p. 13, 2008.
- [31] « TC74A2-3.3VAT | Capteur de température Microchip, -40 → +125 °C, A-220 5-pin, TC74 | RS ». <https://fr.rs-online.com/web/p/capteurs-de-temperature-et-d-humidite/1779720> (consulté le 16 août 2022).
- [32] « <https://fr.jf-parede.pt/eprom-features-applicaitons-circuit-diagram> ». <https://fr.jf-parede.pt/eprom-features-applicaitons-circuit-diagram> (consulté le 16 août 2022).
- [33] F. Batteix, « Utiliser un module horloge temps réel DS1307 avec une carte Arduino / Genuino », *!-APP.SHORT_TITLE-!*, 7 décembre 2016. <https://www.carnetdumaker.net/articles/utiliser-un-module-horloge-temps-reel-ds1307-avec-une-carte-arduino-genuino/> (consulté le 24 juin 2022).
- [34] « Module Horloge DS1307 ». <https://lesiteferroviaire.pagesperso-orange.fr/Module%20Horloge%20DS1307.htm#Ancre5> (consulté le 24 juin 2022).
- [35] « Cours de Proteus professi ». <https://www.yumpu.com/fr/document/view/62153439/pcb/7> (consulté le 24 juin 2022).
- [36] « V.TOURTCHINE. Programmation en mikroC. Application pour les microcontrôleurs de la famille PIC », p. 123.