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

UNIVERSITÉ ABOU BEKR BELKAID DE TLEMCEN
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE

MÉMOIRE DE MASTER EN ELECTRONIQUE

OPTION : INSTRUMENTATION ELECTRONIQUE

Réalisation d'une horloge en temps réel

Soutenu le 17 juin 2017 devant le jury:

Président:	Mr Abdallah Hachim	UABB Tlemcen
Examinateur:	Mr ZOUGAGH	UABB Tlemcen
Encadreur:	Mr Nouredine MASSOUM	UABB Tlemcen

Présenté par: **Mohammed Amin RECHIDI**
Youssouf MOUSLIM

Année académique: 2016-2017

Dédicace

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie.

Je dédie ce modeste travail à :

A mes très chers parents, que dieu les garde et les protège pour leurs soutien moral et financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.

A mon frère: AHMED RAMI

A ma petite sœur : MARWA

A toutes les familles : RECHIDI, HEBRI, BENAMER.

A mon binôme YOUSSOUF.

A tous mes amis.

En fin à moi-même et toute la promotion master d'instrumentation électronique 2017.

RECHIDI MOHAMMED AMIN.

Dédicace

Au nom d'ALLAH, le tout puissant, le miséricordieux.

D'abord je remercie le bon Dieu qui m'a donné le courage pour arriver à la fin d'études.

Je dédie ce modeste travail à :

***A** mon père*

***A** ma plus belle étoile qui puisse exister dans l'univers ma chère mère*

***A** Mes frères : OMAR, ALI.*

***A** Ma sœur : KARIMA.*

***A** toutes les familles : MOUSLIM, BEKKOUCHE.*

***A** mon binôme MOHAMMED AMIN.*

***A** tous mes amis.*

En fin à moi-même et toute la promotion master d'instrumentation électronique 2017.

MOUSLIN YOUSSOUF .

Nous rendons nos profondes gratitudes à dieu qui nous a aidés à réaliser ce modeste travail.

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

Nous remercions. Notre encadreur
MONSIEUR MASSOUM NOUREDINE
pour les efforts qu'il a déployé, pour nous aider, conseiller, encourager et corriger.

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

Nous remercions aussi tout le corps enseignant et administratif qui a contribué à notre formation universitaire.
Sans oublier tous nos amis.

Liste des figures

Liste des tableaux

Résumé :

Ces pages sont une présentation simplifiée du composant. Elles permettront de lire plus facilement la documentation officielle.
Ce circuit propose de créer :

- soit une horloge associée à un calendrier
- soit un compteur d'évènements.

Nous n'étudions ici que la fonction horloge-calendrier.
Dans cette configuration, le circuit se comporte comme une horloge. Les anglo-saxons lui donnent le surnom de RTC pour Real Time Clock. Pour donner l'heure en permanence, le circuit doit posséder une alimentation autonome de sauvegarde.
Dans sa fonction horloge il donne :

- les centièmes de seconde
- les secondes
- les minutes
- les heures.

Elle se fait par une liaison I²C, en lecture ou en écriture. Chaque information est contenue dans un octet.
La conception et la réalisation de ce système dépend de microcontrôleur PIC16F876A afin que ce dernier va gérer tous les processus dans le circuit de notre horloge et calendrier.
La simulation est réalisée à l'aide d'un programme élaboré sous ISIS.

Mots clés : Real Time Clock, microcontrôleur PIC16F876A, liaison I²C, Simulation ISIS

Abstract :

These pages are a simplified layout of the component. They will make it easier to read the official documentation.
This circuit proposes to create:
• Either a clock associated with a calendar
• Or an event counter.

We are studying here only the clock-calendar function.
In this configuration, the circuit behaves like a clock. The Anglo-Saxons
Give the nickname of RTC for Real Time Clock. To give the time constantly, the circuit must have an autonomous backup power supply.
In its clock function it gives:
• hundredths of a second
• seconds
•minutes
•hours.

It is done by an I²C link, read or write. Each information is contained in one byte.
The design and realization of this system depends on PIC16F876A microcontroller so that it will handle all processes in the circuit of our clock and calendar.
The simulation is carried out using a program developed under ISIS.

Keyword: Real Time Clock, PIC16F876A microcontroller, I²C link, ISIS simulation.

Résumé :

ملخص:

هذه الصفحات هي عرض مبسط للعنصر. وسوف تسمح لتسهيل القراءة من الوثائق الرسمية
وتقترح هذه الدائرة لخلق:

- يتم على مدار الساعة مرتبطة مع تقويم
- أن يكون العداد الحدث.

نحن هنا دراسة وظيفة على مدار الساعة التقويم.
في هذا التكوين، على حلبة يتصرف مثل الساعة. الأنجلو ساكسونيا
منح لقب RTC في الوقت الحقيقي على مدار الساعة. لإعطاء مدار الساعة بشكل دائم، على حلبة
يجب أن يكون بها احتياطية امدادات الطاقة.
في وظيفة الساعة تعطي:

- المئات من الثانية
- ثواني
- دقيقة
- الساعات.

يتم ذلك عن طريق وصلة I^2C ، القراءة والكتابة. ويرد كل المعلومات في بايت واحد.
تصميم وتنفيذ هذا النظام يعتمد على متحكم PIC16F876A بحيث أنه سيتم إدارة جميع العمليات في نظام على مدار الساعة لدينا والتقويم.
يتم تنفيذ المحاكاة باستخدام برنامج التي وضعتها ISIS.

كلمات البحث: الوقت الحقيقي على مدار الساعة، PIC16F876A متحكم، وصلةI^2C ،المحاكاة ISIS.

Sommaire :

Sommaire :

Introduction générale :

Avec l'avènement 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. Le terme de **temps réel** n'est certes pas nouveau, il correspond toujours à une catégorie bien précise de traitements critiques dans le temps.

La fonction d'horloge, qui permet de savoir quel jour on est et quelle heure il est, peut être basée sur l'un des configurations qui est :

- L'horloge externe RTC (*) de type DS1307

(*) RTC = Real Time Clock en anglais, horloge temps réel en français

Le dialogue avec le DS1307 s'opère via deux fils de liaison, qui sont ici les lignes SCL (signaux d'horloge) et SDA (signaux de données). On a affaire à un bus de type I2C. Il aurait tout à fait été possible de se passer de ce composant externe et d'utiliser l'horloge interne du PIC (avec quartz externe) pour produire les données horaires, ce ne sont pas les ressources du PIC qui manquent. En réalité, utiliser une horloge dédiée externe permet de disposer d'une plus grande autonomie de fonctionnement secouru en cas de panne secteur. Mais l'usage de cette seule horloge externe ne suffit pas pour disposer d'une source de temps permanente. Il faut que cette horloge soit alimentée de façon autonome, avec une petite pile, une batterie. La sauvegarde en énergie est pour cette raison bien plus aisée à implémenter. En cas de coupure secteur, seul l'horloge externe (DS1307) reste alimentée grâce à la source de tension indépendante Bat1. Le reste du montage est complètement hors tension et la consommation globale n'est plus que de quelques dizaines de microampères. Quand la source d'énergie principale revient, le PIC se remet en marche, et questionne l'horloge externe pour se remettre aux bonnes dates et heure, en une seconde environ. Le proto a été réalisé avec un DS1307.

Bibliographie :

[1] P.F.E : Conception et réalisation d'un enregistreur de données Réalisé par: Massoum Nouredine.

[2] http://www.platea.123.fr/datasheet/cs.pdf.

[3] http://fr.wikipedia.org/wiki/Afficheur 7segments.

[4] https://www.carnetdumaker.net/articles/utiliser-un-module-horloge-temps-reel-ds1307-avec-une-carte-arduino-genuino/

Annexes :

Liste des composants :

- R1 ,R2,R3,R4,R5,R6,R7,R8,R9 ,R10,R11,R12,R13 ,R14 : 220 ohms
- R15,R16,R17,R18 ,R19,R20,R21,R22 : 2 .2 K
- R23,R24,R25 ,R26 ,R27 :1k
- D1,D2,D3 : 1N4148
- U1 :PIC16F876A
- U2 ,U3 :SN74HC595
- U4 : DS1307
- LA PILE CR2032.
- Quartz :12MHZ
- Quartz :3268 HZ
- C1,C2,C3,C4 :12pf
- C5 :100n
- C6 :10µf
- 14 chiffres d'afficheurs 7SEGMENTS

- Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9 :BC327
- 3 boutons poussoirs
- 5 Leds

CHAPITRE :I

I. 1/ Les microcontrôleurs :

I.1.1/ Introduction :

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 grand publique. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques. Ils sont certes peu puissant comparés à des processeurs dédiés, mais ils les compensent par leur prix mais surtout leur taille : un microcontrôleur peut être comparé à une carte mère d'ordinateur.

Dans ce chapitre nous présentons une description détaillée sur microcontrôleur (PICs de Microchip) [1].

I.1.2/ Définition d'un microcontrôleur :

Un microcontrôleur, est un composant électronique qui rassemble tous les éléments d'un "mini-ordinateur" et qui se présente sous la forme d'un circuit intégré. Un microcontrôleur permet de réaliser des systèmes et montages électroniques programmés. Cela veut dire que l'on pourra, avec le même montage, réaliser des fonctions très différentes qui dépendront du programme qui aura été programmé dans le microprocesseur[1].

I.1.3/ Les avantages du microcontrôleur :

L'utilisation des microcontrôleurs pour les circuits programmables a plusieurs points forts. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années[1].

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

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuit imprimé.

- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de donnée d'un composant à un autre.

- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimer diminue.

- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux[1] :

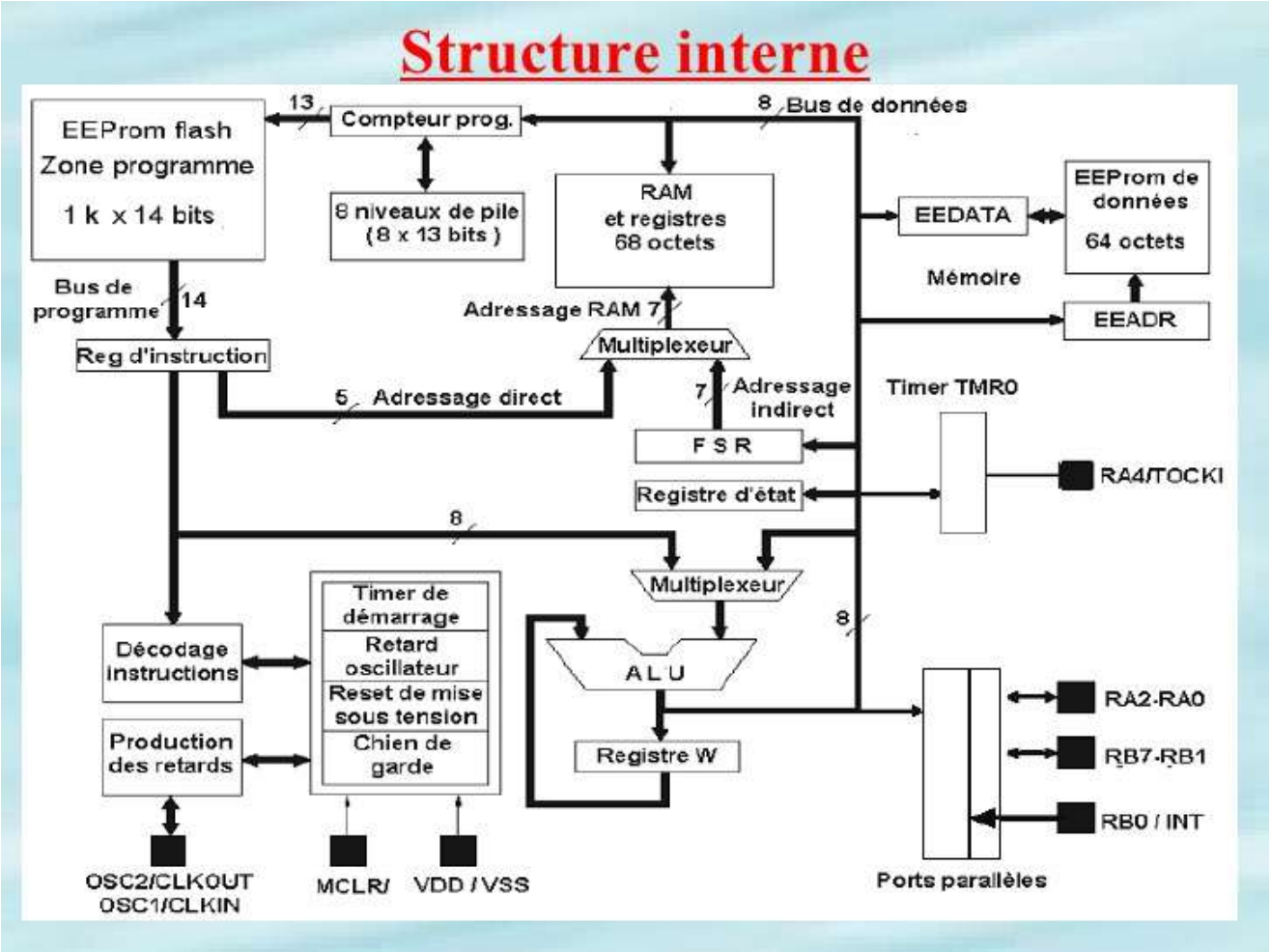
- Moins cher que les autres composants qu'il remplace.

- Diminuer les coûts de main d'œuvre.

•Réalisation des applications non réalisables avec d'autres composants.

I.1.4/ Contenu d'un microcontrôleur :

Un circuit microcontrôleur doit contenir dans un seul boîtier tous Les éléments de bases qu'on verra par la suite. En effet, pour l'analyse des divers systèmes réalisés avant l'avènement des microcontrôleurs, les fabricants des circuits intégrés ont affiné un peu la définition de ce qu'il fallait intégrer pour arriver à un schéma type analogue à la figure(1) suivante[1] :



Fig(1) :Structure interne d'un microcontrôleur [1]

I.2 / Les Pics :

I.2.1/Définition d'un PIC :

Un PIC (Programmable Interface Contrôler) est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes. Les PICs sont des composant dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Le microcontrôleur se trouve, dans plusieurs appareils telle que : les téléphones portables, machines à laver, télévisions vidéos ... etc.

I.2 .2/Les différentes familles de PIC :

Les Pics sont subdivisés en 3 grandes familles[1] [2] :

- 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 les 16F84 et16F876)
- la famille des 18Fxxx qui utilise des mots de 16 bits.

Structure d'un PIC :

Les PIC, au même titre que les microprocesseurs, sont composés essentiellement de registres ayant chacun une fonction bien définie. Les PIC possèdent également des périphériques intégrés, tels qu'une mémoire EEPROM, un timer, des ports d'entrées/ sorties ou bien encore un convertisseur analogique/numérique[1].

Selon le type de PIC utilisé, on retrouvera en interne un certain nombre de registres et périphériques possédant des caractéristiques différentes. Les différences de caractéristiques selon le PIC utilisé sont [1] :

- La taille de la RAM interne ;
- La mémoire EEPROM intégrée ;
- Le type de mémoire programme : FLASH, EPROM ou OTP et la taille de celle-ci.
- Le timer intégré ;
- Les convertisseurs analogique/numérique intégrés.

Structure minimale d'un PIC :

La structure minimale d'un PIC est constituée des éléments ci-dessous[2] :

- Une mémoire de programme contient le code binaire correspondant aux instructions que doit exécuter le microcontrôleur. La capacité de cette mémoire est variable selon les PIC
- Une mémoire RAM sauvegarde temporairement des données. Sa capacité est aussi variable selon les PIC.

- Une Unité Arithmétique et Logique (UAL ou ALU en anglais) est chargée d'effectuer. Toutes les opérations arithmétiques de base (addition, soustraction, etc.) ainsi que les opérations logiques de base (ET, OU logique, etc.…).

- Des ports d'entrées/sorties permettent de dialoguer avec l'extérieur du microcontrôleur, par exemple pour prendre en compte l'état d'un interrupteur (entrée logique), ou encore pour commander un relais (sortie logique).

- Un registre compteur de programme (CP ou PC en anglais), est chargé de pointer l'adresse mémoire courante contenant l'instruction à réaliser par le microcontrôleur. Le contenu du registre PC évolue selon le pas de programme.

- Un registre pointeur de pile (PP ou SP en anglais) est essentiellement utilisé lorsque l'on réalise un sous-programme. Le pointeur de pile est chargé de mémoriser l'adresse courante que contient le compteur de programme avant le saut à l'adresse du sous-programme.

Lorsque le sous-programme est terminé, le pointeur restitue l'adresse sauvegardée vers le compteur de programme.

- Un registre d'instruction contient tous les codes binaires correspondant aux instructions à réaliser par le microcontrôleur. Le PIC 18F458 comporte 35 instructions.

- Un registre d'état est en relation avec l'UAL et permet de tester le résultat de la dernière opération effectuée par le microcontrôleur. Selon la dernière opération effectuée, des bits sont positionnés dans le registre d'état et ceux-ci peuvent être testés à l'aide d'une instruction de branchement pour effectuer des sauts conditionnels.

- Une horloge système permet de cadencer tous les échanges internes ou externes au 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 la famille High-End, qui utilise des mots de 16 bits.

I.2 .3/identification de PIC16F876A :

Fig(2):Photo réel du PIC16F876A[2]

La dénomination PIC est sous copyright de Micro Chip, les autres fabricants sont dans l'impossibilité d'utiliser ce terme.

Les deux premiers chiffres indiquent la catégorie du PIC : ici 16 indique un PIC de la famille Mid Range (milieu de gamme) qui utilise des mots de 14 bits pour coder une instruction.

Ensuite on peut trouvez la lettre « L» qui indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante.

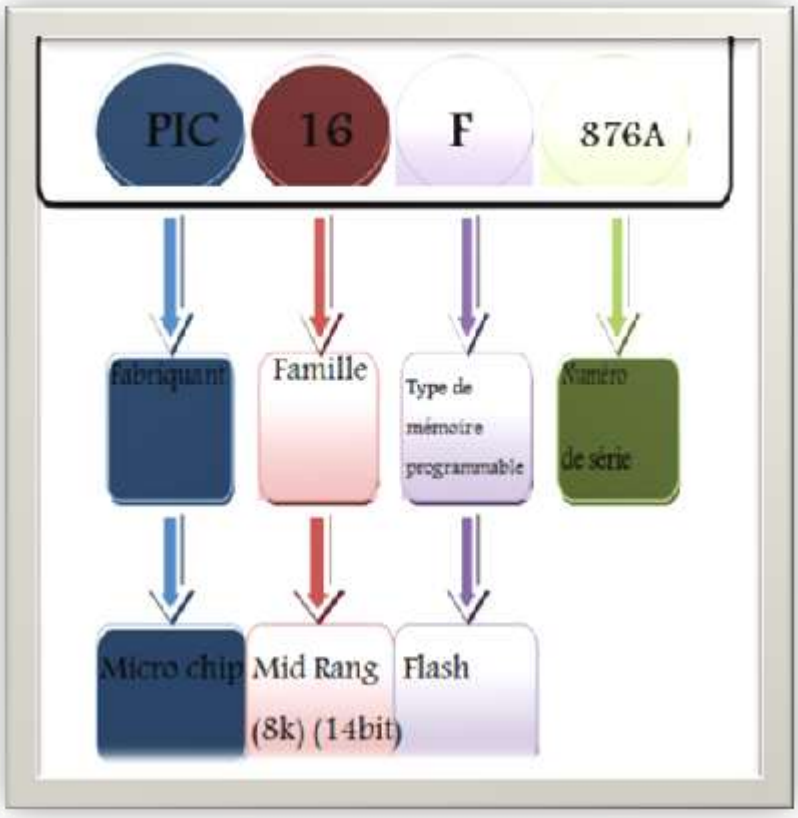
Ensuite vous trouverez les lettres suivantes [2] :

« C» : la mémoire programme est une EPROM ou plus rarement une EEPROM,

« CR» : la mémoire programme est de type ROM,

«F» : la mémoire programme est de type FLASH.

876 : représente le numéro du circuit en question.



Fig(3) : identification de PIC16F876A [2]

Le PIC 16F876A est un circuit intégré de type CMOS. Il est de la famille MID-RANGE(16) et la mémoire programme est de type FLASH (F). Il est capable de fonctionner à des fréquences d'horloge allant de 0 à 20MHz[2].

I.2 .4/Brochage :

Le 16F876A est un circuit intégré de 28 broches, que l'on peut trouver dans un boîtier DIL (Dual In Line) de 2x14pattes comme indiqué à la figure(4) ci-dessous. A chacune de ses broches, il est associé une ou plusieurs fonctions. Chaque broche peut donc jouer plusieurs rôles selon les configurations effectuées lors de la programmation du PIC [2].

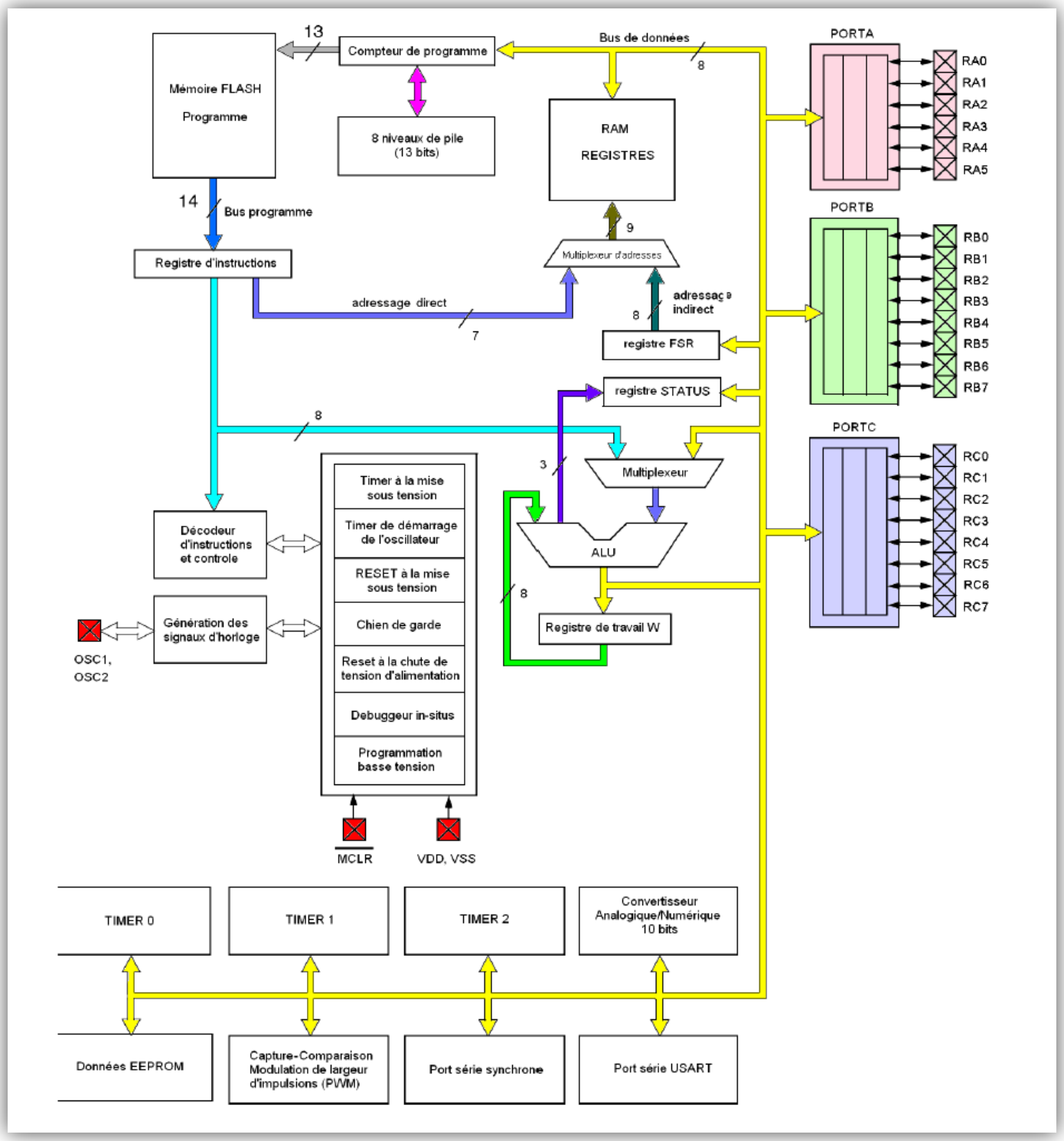
I

Fig(4):Brochage du PIC16F876A [2]

I.2 .5/les caractéristiques du PIC16F876A :

- Une mémoire morte de type FLASH de 8K mots (1mot = 14 bits), elle est réinscriptible à volonté.
- Une Fréquence de fonctionnement qui va de 0 à 20 MHz.
- 3 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ et TIMER2 (8 bits avec pré diviseur et post diviseur).
- 13 sources d'interruption.
- 3 ports de communication.

I.2 .6/ Architecture interne du PIC 16F876A:



Fig(5):Architecture interne du PIC16F876A [2]

I.3 / Etude des ports d'entrée sortie:

Le 16F876A possède jusqu'à 22 entrées/sorties[2] :

- 6 sur le port A (RA0 à RA5).
- 8 sur le port B (RB0 à RB7).
- 8 sur le port C (RC0 à RC7).

Notez qu'il y a deux broches de masse (broches 8 et 19).

Certaines broches sont multiplexées à d'autres fonctions comme indiqué ci-dessous[2] :

I.3 .1/Le PORT A :

Les abréviations et configurations relatives au port A sont données ci-dessous [2]:

- ***RA0/AN0 :***

- RA0 : Entrée Sortie numérique.
- AN0 : Entrée analogique.

- ***RA1/AN1 :***

- RA1 : Entrée Sortie numérique.
- AN1 : Entrée analogique.

- ***RA2/AN2 :***

- RA2 : Entrée Sortie numérique.
- AN2 : Entrée analogique.

- ***RA3/AN3/VREF :***

- RA3 : Entrée Sortie numérique.
- AN3 : Entrée analogique.
- VREF : Tension de référence.

- ***RA4/T0CKI*** :

- RA4 : Entrée Sortie numérique.
- T0CKI : Entrée d'horloge du TMR0.

- ***RA5/$\overline{SS}$/AN4*** :

- RA5 : Entrée Sortie numérique.
- $\overline{SS}$: Entrée de sélection esclave pour le port série synchrone.
- AN4 : Entrée analogique.

I.3 .2/Le PORT B:

Les abréviations et configurations relatives au port B sont données ci-dessous [2]:

- ***RB0/INT :***

- RB0 : Entrée Sortie numérique.
- INT : Broche d'interruption externe.

- ***RB1 :***

- RB1 : Entrée Sortie numérique.

- ***RB2 :***

- RB2 : Entrée Sortie numérique.

- ***RB3/PGM*** :

- RB3 : Entrée Sortie numérique.
- PGM : Entrée de la tension de programmation basse tension.

- ***RB4*** :

- RB4 : Entrée Sortie numérique.

- ***RB5*** :

- RB5 : Entrée Sortie numérique.

- ***RB6/PGC*** :

- RB6 : Entrée Sortie numérique.
- PGC : Entrée d'horloge en mode programmation.

- ***RB7/PGD*** :

- RB7 : Entrée Sortie numérique.
- PGD : Entrée de donnée en mode programmation.

I.3 .3/Le PORT C :

Les abréviations et configurations relatives au port C sont données ci-dessous [2]:

- ***RC0/T1OS0/T1CKI :***

- RC0 : Entrée Sortie numérique.
- T1OSO : Sortie d'oscillateur du TMR1.
- T1CKI : Entrée d'horloge du TMR1.

- ***RC1/T1OSI/CCP2 :***

- RC0 : Entrée Sortie numérique.
- T1OSI : Entré de l'oscillateur du TMR1.
- CCP2 : Entrée/Sortie du module CCP2.

- ***RC2/CCP1 :***

- RC2 : Entrée Sortie numérique.
- CCP1 : Entrée/Sortie du module CCP1.

- ***RC3/SCK/SCL*** :

- RC3 : Entrée Sortie numérique.
- SCK : Entrée d'horloge en mode SPI.
- SCL : Entrée d'horloge en mode I2C.

- ***RC4/SDI/SDA*** :

- RC4 : Entrée Sortie numérique.
- SDI : Entrée de données en mode SPI.
- SDA : Entrée/Sortie de données en mode I2C.

- ***RC5/SDO*** :

- RC5 : Entrée Sortie numérique.
- SDO : Sortie de données en mode SSP.

- ***RC6/TX/CK*** :

- RC6 : Entrée Sortie numérique.
- TX : Broche de transmission en mode USART Asynchrone.
- CK : Entrée d'horloge en mode USART synchrone.

- ***RC7/RX/DT*** :

- RC7 : Entrée Sortie numérique.
- RX : Broche de réception en mode USART Asynchrone.
- DT : Entrée/Sortie en mode USART synchrone.

MCLR : master clear. Broche de réinitialisation.

VPP : Tension de programmation≈13V.

I.4 / Les registres spéciaux:

I.4 .1 Les registres TRISA, TRISB, TRISC:

Ces registres permettent de configurer les broches soit en entrée soit en sortie. Exemple : ***TRISC.2 = 1*** signifie que RC2 est en entrée. Lorsqu'un bit est à 1, la broche correspondante est en entrée, si c'est 0 la broche est en sortie[2].

I.4 .2 Le registre OPTION_REG.:

Il permet la configuration les paramètres du TMR0, du chien de garde et des résistances de tirage comme indiqué ci-dessous [2]:

- ✓ ***OPTION_REG.7 = RBU*** : Permet d'activer les résistances de tirage (0).
- ✓ ***OPTION_REG.6 = INTEDG*** : Permet de sélectionner le front, montant (1) ou descendant, du signal d'interruption de la broche RB0.
- ✓ ***OPTION-REG.5. = T0CS*** : Permet de sélectionner la source d'horloge, interne (0) ou externe (1), du TMR0.
- ✓ ***OPTION-REG.4. = T0SE*** : Permet de sélectionner le front d'horloge, montant (0) ou descendant (1), du TMR0.
- ✓ ***OPTION-REG.3. = PSA*** :Permet d'appliquer le pré-diviseur soit sur le TMR0 (0),

soit sur le chien de garde (1).

- ✓ ***OPTION-REG.2 : OPTION-REG.0. = PS2 :PS0*** : permettent de fixer le rapport de la pré-division selon Le tableau(1) ci-dessous[2] :

Valeur binaire	Rapport de la pré-division	
	TMR0	Chien de garde
000	1/2	1/1
001	1/4	1/2
010	1/8	1/4
011	1/16	1/8
100	1/32	1/16
101	1/64	1/32
110	1/128	1/64
111	1/256	1/128

Tableau(1) : les rapports de la pré-division [2]

I. 5/ Les Bus I2C :

Le bus I2C, dont le sigle signifie Inter Integrated Circuit ce qui donne I2C et par contraction I2C.Le protocole est initialement proposé par Philips mais adopté de nos jours par de très nombreux fabricants. C'est un bus de communication de type série.

I. 5.1/Présentation :

Le bus I2C qui n'utilise que deux lignes de signal permet à un certain nombre d'appareils d'échanger des informations sous forme série avec un débit pouvant atteindre 100 Kbps ou 400 Kbps pour les versions les plus récentes[2].

Ceci étant précisé, voici quels sont les points forts du bus I2C [2] :

C'est un bus série bifilaire utilisant une ligne de données appelée SDA (Serial Data) et une ligne d'horloge appelée SCL (Serial Clock) .

Les données peuvent être échangées dans les deux sens sans restriction.

Le bus est multi-maître.

Chaque abonné dispose d'une adresse codée sur 7 bits. On peut donc connecter simultanément 128 abonnés d'adresses différentes sur le même bus, sous réserve de ne pas le surcharger électriquement .

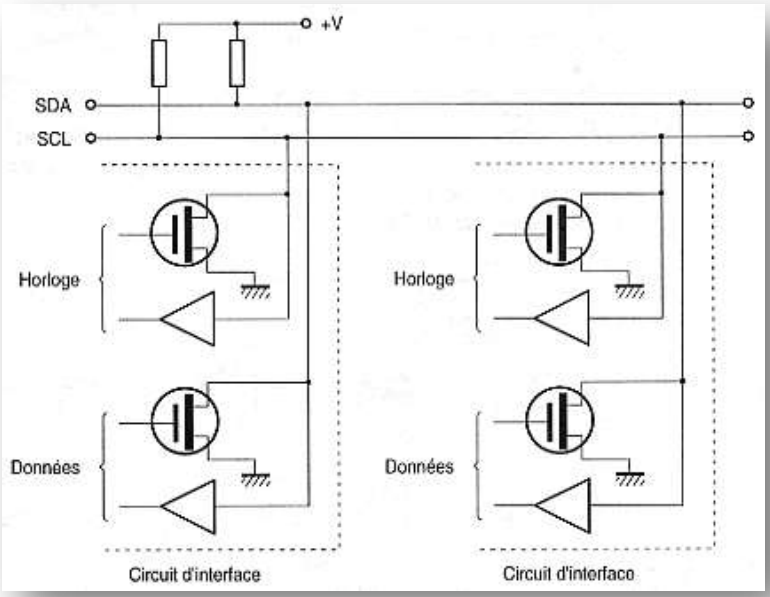
· Un acquittement est généré pour chaque octet de donnée transféré .

· Le bus peut travailler à une vitesse maximum de 100 Kbps (ou 400 Kbps) le protocole permet de ralentir automatiquement l'équipement le plus rapide pour s'adapter à la vitesse de l'élément le plus lent, lors d'un transfert .

· Le nombre maximum d'abonnés n'est limité que par la charge capacitive maximale du bus qui peut-être de 400 pF.

· Les niveaux électriques permettent l'utilisation de circuits en technologies CMOS, NMOS ou TTL.

I. 5.2/Principe d'un échange de données :

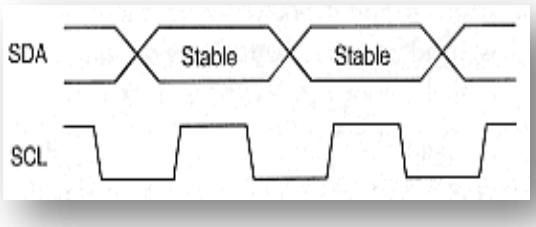


Fig(6):Schéma interne interface I2C [2]

Cette figure (6) montre le principe adopté au niveau des étages d'entrée/sortie des circuits d'interface au bus I2C [2].

Aucune charge n'étant prévue dans ces derniers, une résistance de rappel à une tension positive doit être mise en place. Le niveau électrique n'est pas précisé pour l'instant car il dépend de cette tension. Nous parlerons donc de niveaux logiques hauts ou « 1 » ou bien encore de niveaux logiques bas ou « 0 » étant entendu que l'on travaille en logique positive c'est-à-dire qu'un niveau haut correspond à une tension plus élevée qu'un niveau bas[2].

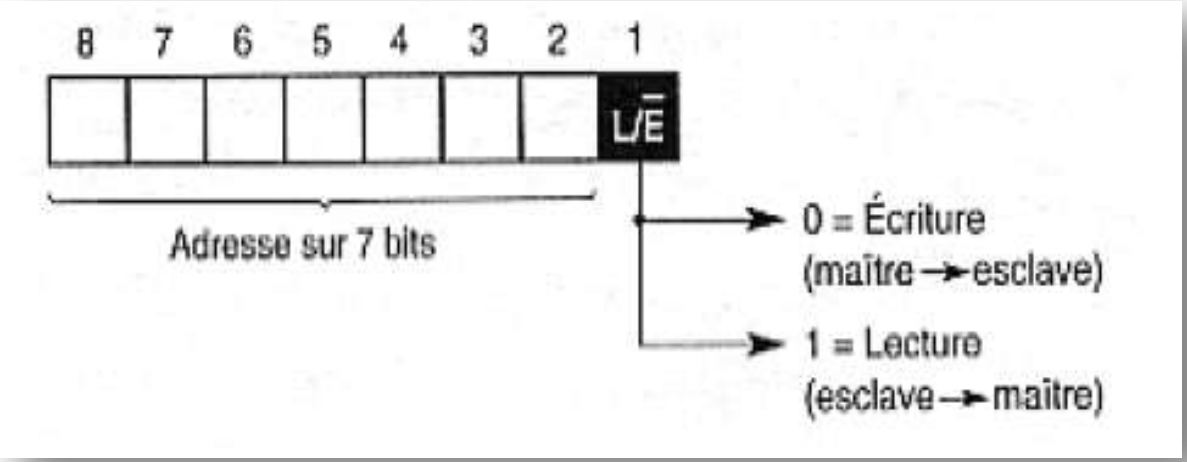
Compte tenu de ce mode de connexion en ET câblé, lorsqu'aucun abonné n'émet sur le bus, les lignes SDA et SCL sont au niveau haut qui est leur état de repos[2].



Fig(7): Principe fondamental d'un transfert Bus I2C [2]

I. 5.3/Formats de transmission :

Nous savons maintenant comment se déroulent les échanges ; il nous reste à examiner le format des données transmises afin de comprendre comment fonctionne l'adressage, mais aussi la définition du sens de transferts des données.



Fig(8):Le contenu du premier octet d'un Bus I2C [2]

Cette figure (8) montre le contenu du premier octet qui est toujours présent en début d'échange. Ses sept bits de poids forts contiennent l'adresse du destinataire du message ce qui autorise 128 combinaisons différentes [2].

Le bit de poids faible indique si le maître va réaliser une lecture ou une écriture. Si ce bit est à zéro le maître va écrire dans l'esclave ou lui envoyer des données. S'il est à un, le maître va lire dans l'esclave c'est-à-dire que le maître va recevoir des données de l'esclave[2].

Lorsqu'un maître désire effectuer plusieurs échanges à destination d'esclaves d'adresses différentes, il n'est pas obligé de terminer le premier échange par une condition d'arrêt mais peut les enchaîner en générant une condition de départ dès la fin d'un échange[2].

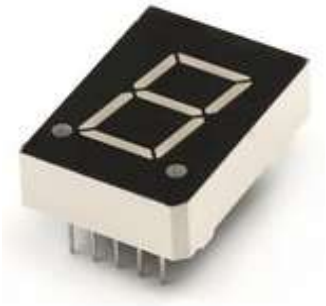
Enfin, il existe une procédure dite d'appel général où l'adresse envoyée par le maître, c'est-à-dire rappelons-le, les sept bits de poids forts du premier octet, est nulle. Tous les circuits connectés sur le bus capables de répondre à un tel appel général doivent alors le faire et prendre en compte les données qui suivent. Leur attitude dépend du bit de lecture/écriture de ce premier octet.

CHAPITRE :II

II.1/Afficheur 7 SEGMENTS:

II.1.1/ Définition :

Comme son nom l'indique, l'afficheur 7 segments possède 7 segments. Mais un segment c'est quoi au juste? Et bien c'est une portion de l'afficheur, qui est allumée ou éteinte pour réaliser l'affichage. Cette portion n'est en fait rien d'autre qu'une LED qui au lieu d'être ronde comme d'habitude est plate et encastré dans un boiter. On dénombre donc 8 portions en comptant le point de l'afficheur (mais il ne compte pas en tant que segment à part entière car il n'est pas toujours présent) [3]. Regardez à quoi ça ressemble



Fig(9): ***Afficheur 7 segment*** [3]

En général, un afficheur à 7 segments se programme sur 4 bits grâce à 4 entrées conformément à la table de vérités suivante[3] :

- ***Programmation :***

affichage	Entrée1	Entrée2	Entrée3	Entrée4
0	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**1**	**1**
4	**0**	**1**	**0**	**0**
5	**0**	**1**	**0**	**1**
6	**0**	**1**	**1**	**0**
7	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**
9	**1**	**0**	**0**	**1**
A	**1**	**0**	**1**	**0**
B	**1**	**0**	**1**	**1**
C	**1**	**1**	**0**	**0**
D	**1**	**1**	**0**	**1**
E	**1**	**1**	**1**	**0**
F	**1**	**1**	**1**	**1**

Tableau(2):Table de vérité d' un afficheur à 7 segments se programme sur 4 bits[3]

En notant les entrées 1, 2, 3, 4 du tableau ci-dessus respectivement i1, i2, i3 et i4, les équations des segments (pour afficher les nombres de 0 à F) sont [3] :

- a = (not (i1) and i3) or (i1 and not (i4)) or (i2 and i3) or not(i2 or i4) or (i1 and not(i2) and not(i3)) or (not(i1) and i2 and i4)
- b = not (i1 or i2) or not (i2 or i3) or not(i2 or i4) or (not(i1) and not(i3 xor i4)) or (i1 and not(i3) and i4)
- c = (i1 xor i2) or (not (i3) and i4) or (not (i3 xor i4) and not(i2))
- d = (i1 and not (i3)) or not(i1 or i2 or i4) or (i2 and (i3 xor i4)) or (not(i2) and i3 and i4)
- e = not (i2 or i4) or (i3 and not(i4)) or (i1 and i2) or (i1 and i3)
- f = (i1 and not (i2)) or not (i3 or i4) or (not (i3) and (i1 xor i2)) or (i1 and i3) or (i2 and not(i4))
- g = (i3 and (i1 or not (i2) or not (i4))) or (i1 and i4) or (not(i3) and (i1 xor i2))

On peut retrouver ces équations en établissant la table de karnaugh de chaque segment ; il existe d'autres possibilités de formules.

Dans le cas d'un afficheur 7 segments commandé par 8 bits, la table de vérité donne (segment G correspondant à bit 7 et A à bit 1) [3] :

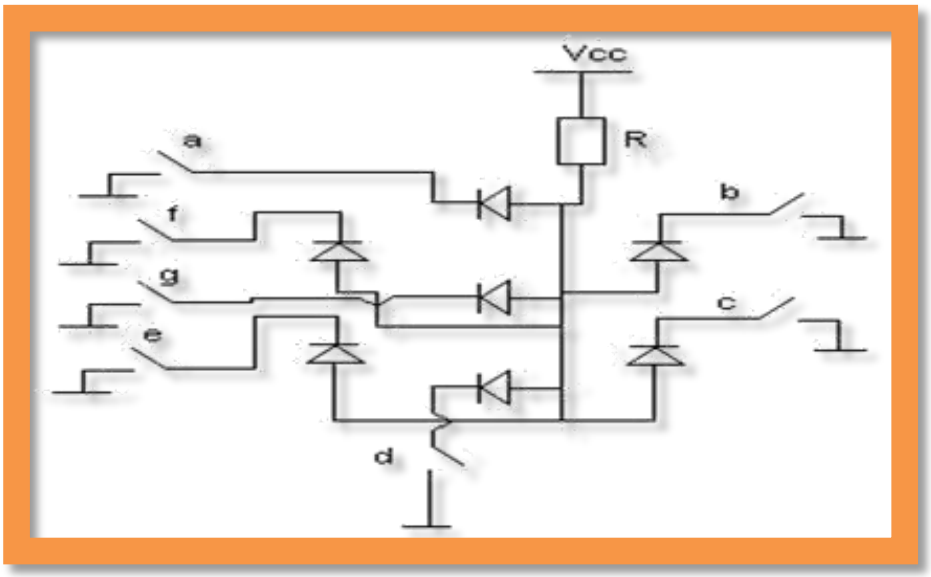
Programmation :

Affichage	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Hexa
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F
A	0	1	1	1	0	1	1	1	0x77
B	0	1	1	1	1	1	0	0	0x7C
C	0	0	1	1	1	0	0	1	0x39
D	0	1	0	1	1	1	1	0	0x5E
E	0	1	1	1	1	0	0	1	0x79
F	0	1	1	1	0	0	0	1	0x71

Tableau(3):Table de vérité d'un afficheur à 7 segments se programme sur 8bits [3]

II.2/ les LED :

Les LED, il y en a Entre 7 et 8 selon les modèles (c'est ce que je viens d'expliquer), voir beaucoup plus, mais on ne s'y attardera pas dessus. Voici un schéma vous présentant un modèle d'afficheur sans le point [3]:



Fig(10):Un modèle d'afficheur sans le point [3]

Les interrupteurs a, b, c, d, e, f, g représentent les signaux pilotant chaque segments.

Comme vous le voyez sur ce schéma, toutes les LED possèdent une broche commune, reliée entre elle. Selon que cette broche est la cathode ou l'anode on parlera d'afficheur à cathode commune ou… anode commune. Dans l'absolu, ils fonctionnent de la même façon, seule la manière de les brancher diffère (actif sur état bas ou sur état haut) [3].

II.3/ Cathode commune ou Anode commune :

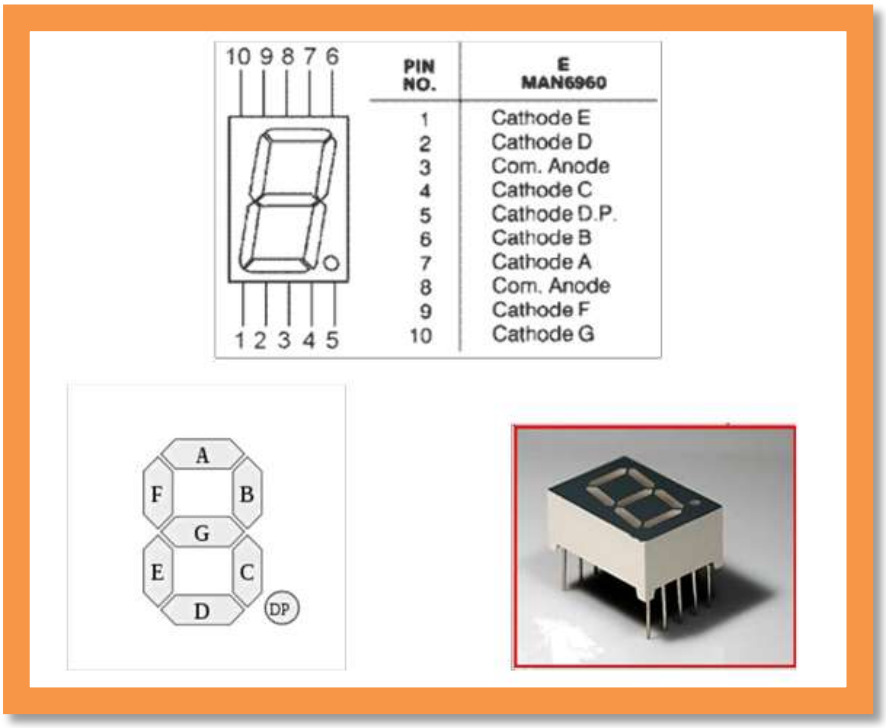
Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal[3].

II.4/ Présentation du boîtier :

Les afficheurs 7 segments se présentent sur un boîtier de type DIP 10. Le format DIP régie l'espacement entre les différentes broches du circuit intégré ainsi que d'autres contraintes (présence d'échangeur thermique etc…). Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boitier). Voici une représentation de ce dernier[3].

Voici la signification des différentes broches[3] :

1. LED de la cathode E.
2. LED de la cathode D.
3. Anode commune des LED.
4. LED de la cathode C.
5. (facultatif) le point décimal.
6. LED de la cathode B.
7. LED de la cathode A.
8. Anode commune des LED.
9. LED de la cathode F.
10. LED de la cathode G.

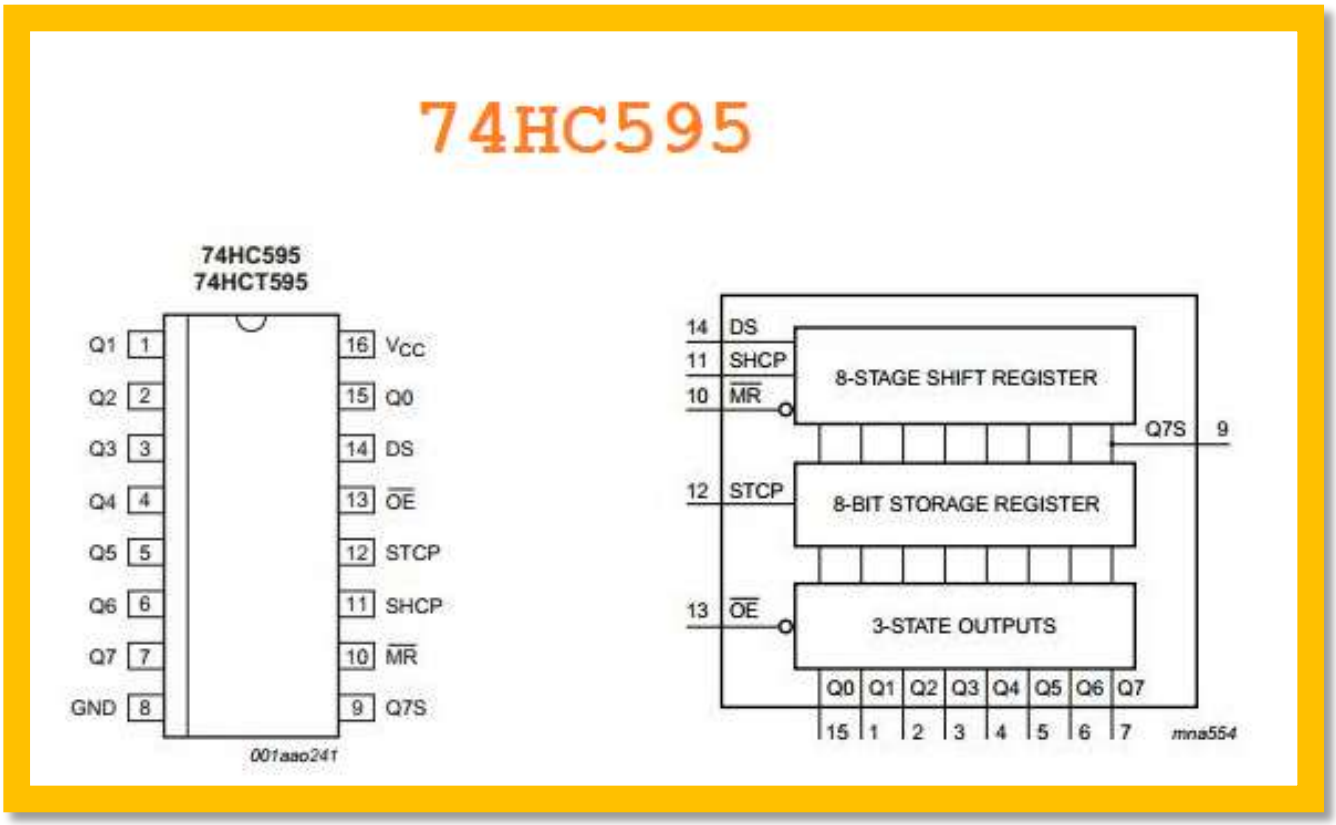


Fig(11):Un boîtier de type DIP 10 d'un afficheur 7segment

Pour allumer un segment c'est très simple, il suffit de le relier à la masse.

II.5/ Registre à décalage 8 bits 74HC595 :

Un registre à décalage est un circuit électronique formé de bascules (flip- flops) en cascade, avec un signal dit d'horloge (clock).C'est un circuit intégré permettant le multiplexage d'entrées et de sorties avec Arduino, contrôle de réseaux de diodes, afficheurs 7 segments, etc[3].



Fig(12) : Un registre à décalage

Caractéristiques74HC595 :

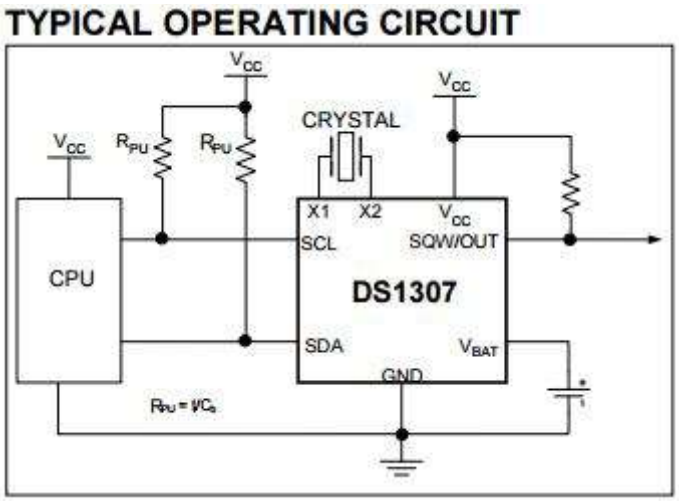
- Circuit intégré 74HC595
- Entrée série de 8 bits sur 1 broche
- Sortie série ou parallèle des 8 bits sur 8 broches
- Registre à décalage avec tampon de stockage en sortie, 3états
- Compatible TTL, LSTTL
- Câblage 2*8 broches
- Décalages jusqu'à 100 MHz
- Température de -40 à +85C°

II.6/ Horloge temps réel DS1307 :

II.6 .1/Définition:

Le module DS1307 est une horloge temps réel (aussi appelé "RTC", "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 avec horodatage par exemple.

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.PS Le module dérive de quelques secondes par jours en moyenne. Cela dépend de la température ambiante et de la qualité du quartz d'horloge[4].



Fig(13):Circuit du DS1307 [4]

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.

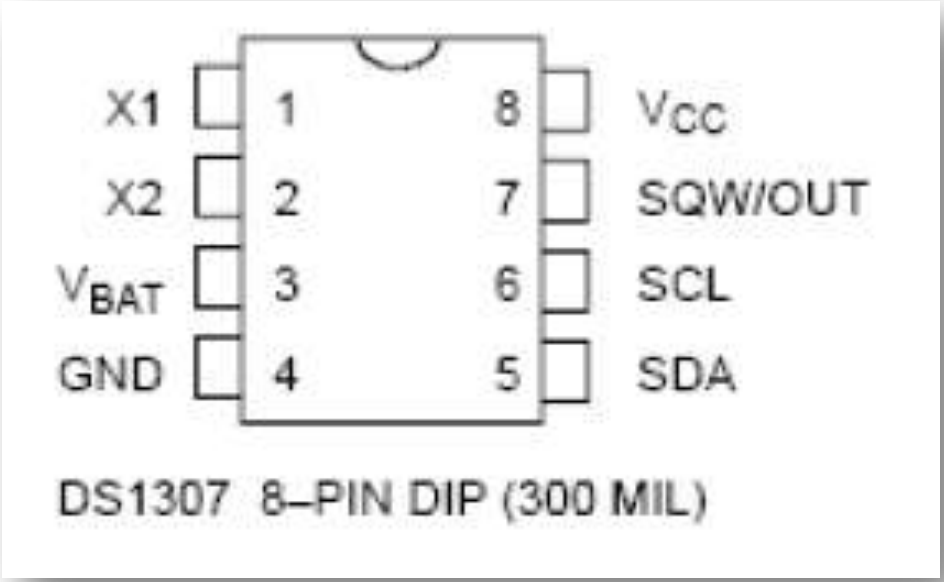
Le module DS1307 ne dispose pas de fonctionnalité "alarme" contrairement à d'autres modules RTC plus haut de gamme. Le module DS1307 dispose cependant d'une sortie "base de temps" permettant d'avoir un signal logique à une fréquence fixe (1 Hertz par exemple) pour faire fonctionner un circuit ou un compteur externe. Cela peut être utile dans certaines applications [4].

II.6 .2/ Principe d'utilisation du module :

Le module DS1307 se comporte comme une petite mémoire externe I²C contenant l'heure et la date à des emplacements fixes. La plage mémoire du module est d'une taille impressionnante de 64 octets. Ce n'est pas énorme, mais il ne faut pas beaucoup de mémoire pour stocker une date et une heure. Lire la mémoire du module revient à lire l'heure et la date courante. Écrire la mémoire du module revient à mettre à jour l'heure et la date[4].

II.6 .3/ Brochage du DS1307 :

L'horloge temps réel DS1307 est commercialisé dans un boitier de 8 broches classiques, comme illustré dans le schéma suivant[4] :

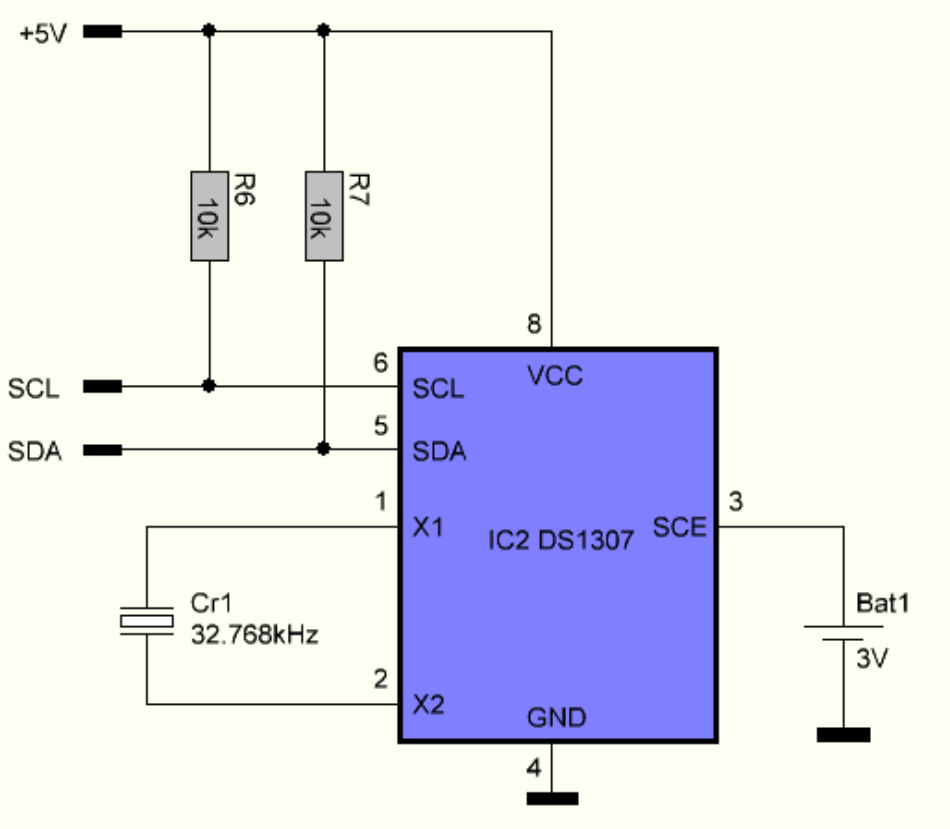


Fig(14):Un boitier de 8 broches du DS 1307 [4]

Pin	*Nom*	*Rôle*
1	X1	cristal
2	X2	cristal
3	Vbat	tension batterie
4	GND	masse
5	SDA	ligne des données (data) I²C
6	SCL	ligne d'horloge (clock) I²C
7	SQW/out	sortie signal carré
8	Vcc	tension logique 5V

Tableau(4) : Le rôle des PIN du DS1307 [4]

II.6 .4/ Schéma du module RTC :



Fig(15):Circuit module RTC [4]

CHAPITRE :III

❖ Partie Logicielle :

III. 1/ *Introduction* :

La programmation des PIC se fait par le langage assembleur qui est un langage de bas niveau qui représente le langage machine sous une forme lisible par un humain.
Le programme assembleur convertit ces mnémoniques en langage machine en vue de créer par exemple un fichier exécutable.
Le développement des environnements de programmation , nous a permis de voir naitre de nouveaux compilateurs qui permettent de programmer avec les langages haut niveau telsque le C , PASCAL,BASIC etc…
Ces environnements comportent aussi des bibliothèques qui permettent de faciliter le développement. Il existe plusieurs outils de développement, les uns sont gratuits, les autres sont payants .
Dans notre recherche de l'outil que nous allons utiliser pour programmer notre PIC , nous avons optés pour le langage C. Ce choix est à la fois un choix personnel et un choix technologique.
D'une part le langage C est utilisé dans différents systèmes et domaines de développement, ce qui nous permettra une évolution future , d'autre part le langage C'est l'un des langages les plus puissants.

- ***Avantages du C :***

Pour de la programmation de base, le C est intéressant. Il permet rapidement, sans gros effort, de développer des programmes fonctionnels. Il permet aussi de s'affranchir de connaissances complexes sur l'architecture des PIC. Il a l'avantage de gérer facilement les boucles, les choix, ainsi que l'affichage[5].

- ***Inconvénients du C :***

Le C n'est pas le langage naturel du microcontrôleur. Il permet de programmer plus intuitivement. Les logiciels de programmation en C transforment alors les lignes en C en lignes assembleurs directement compréhensibles par le microcontrôleur. Pour programmer efficacement, il est souvent nécessaire d'aller voir le code assembleur, il est donc conseillé d'avoir des bases solides en assembleur[5].

Nous avons fait le choix d'utiliser l'environnement de développement MikroC de Mikroelektronika.

III. 2/ *Présentation du MikroC :*

Le « MikroC » est un compilateur pour PIC Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs PIC bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) ; Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...); de ce fait il est un des outils de développement incontournable et puissant[5].

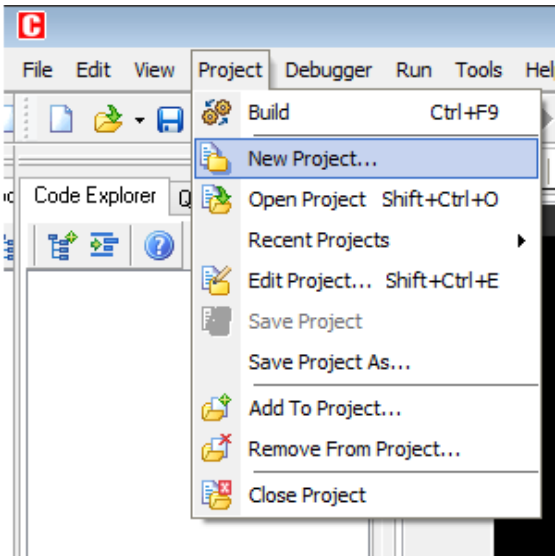
Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète[5].

Fig(16):Interface du logiciel MikroC [5]

Le compilateur MikroC nous permet de développer rapidement des applications complexes.

III. 2.1/ *Création d'un projet*:

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez New Project (Nouveau Projet) de puis le menu Project (Projet), comme indiqué sur figure(17) [5].

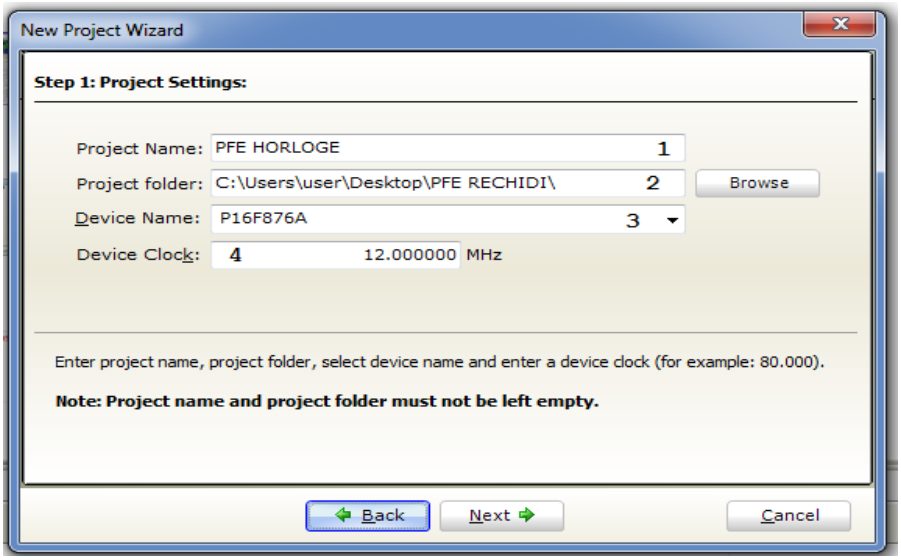


Fig(17):Création d'un projet [5]

III. 2.2/ *Les configurations de projet*:

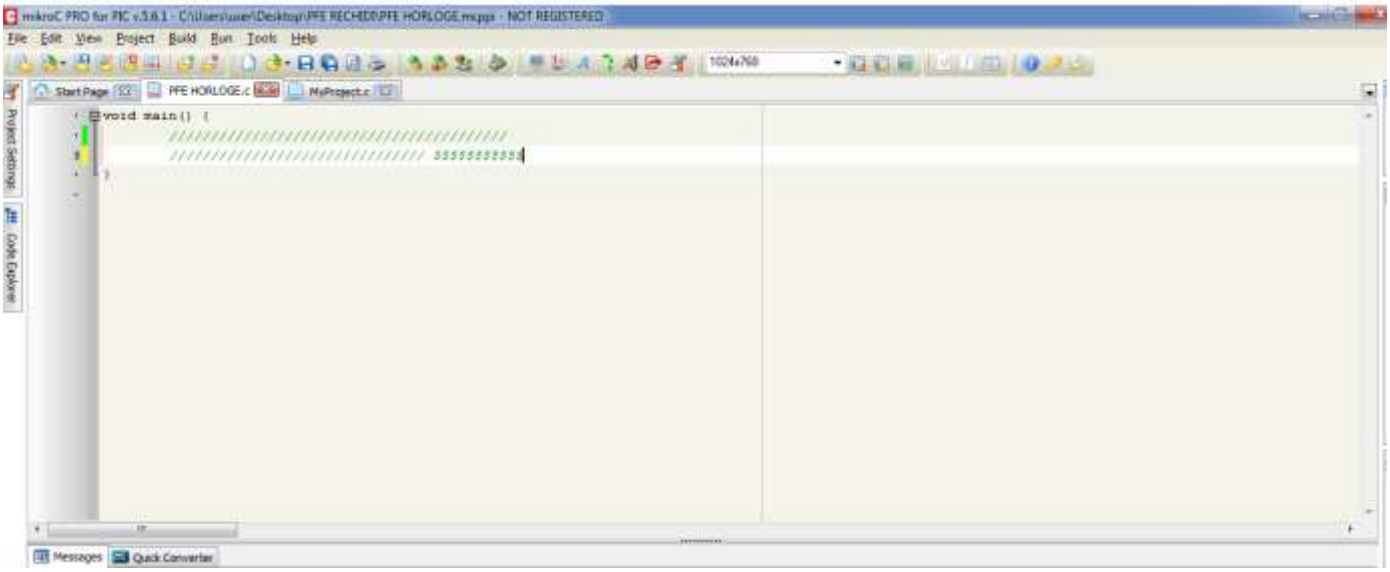
Une nouvelle fenêtre apparaîtra. Comme indiqué sur Figure(18), il y a plusieurs champs à renseigner comme le nom du projet, l'emplacement du projet, sa description, l'horloge et les options du composant. Le tableau device flags (options composant) est utilisé pour la configuration des paramètres du microcontrôleur[5].

1. Nom du projet
2. Nom de l'emplacement du projet
3. Choix du PIC
4. Choix de de la fréquence de l'horloge interne du PIC



Fig(18):Les configurations de projet [5]

Après ça, une nouvelle fenêtre (Fig19) vide s'affiche pour écrire notre programme



Fig(19):Fenêtre de saisie de programme [5]

❖ Partie Pratique :

III. 3/Elaboration du schéma synoptique :

Commençons d'abord par donner le schéma synoptique de notre enregistreur de données :

Fig(20):schéma synoptique

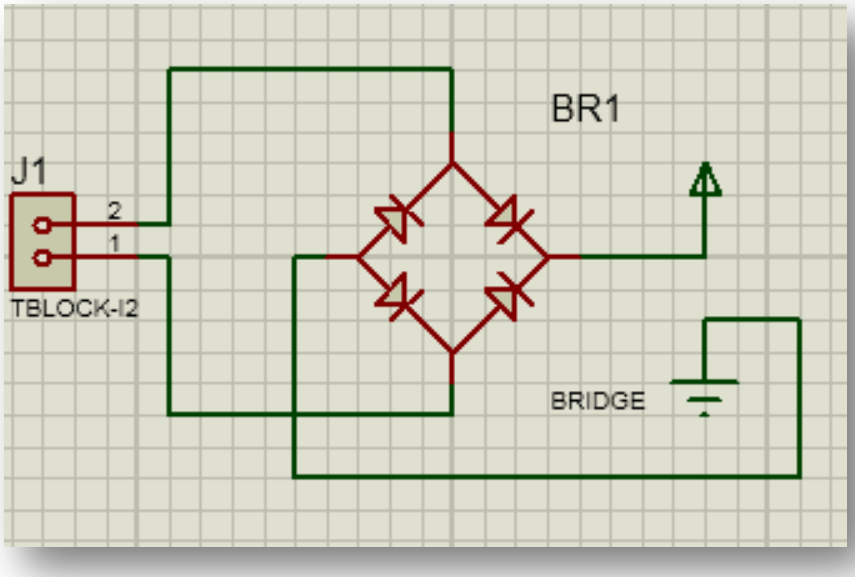
En effet, on trouve :

- l'unité de traitement et de contrôle des données qui est le microcontrôleur (PIC 16F876A).
- Un afficheur de 7SEGMENT pour afficher les données.
- 3 Bouton poussoir touches de commande pour le défilement des instructions contenues dans l'afficheur.
- Une horloge temps réel qui joue le rôle d'un vrai calendrier.

III. 4/circuit d'alimentation :

Tout montage électronique nécessite une alimentation pour fonctionner. Notre montage nécessite une alimentation 5V pour alimenter le PIC et ces périphériques.
Autre contrainte présente dans ce circuit est qu'il peut être, dans certains cas, alimenté par une batterie dans des zones qui ne sont pas couverts par le réseau électrique .

.



Fig(21):circuit d'alimentation

Pour ces raisons nous avons optés pour une alimentation externe, qui sera régulé et stabilisé vers les tensions nécessaires

III. 5/ *Simulation de circuit par ISIS*:

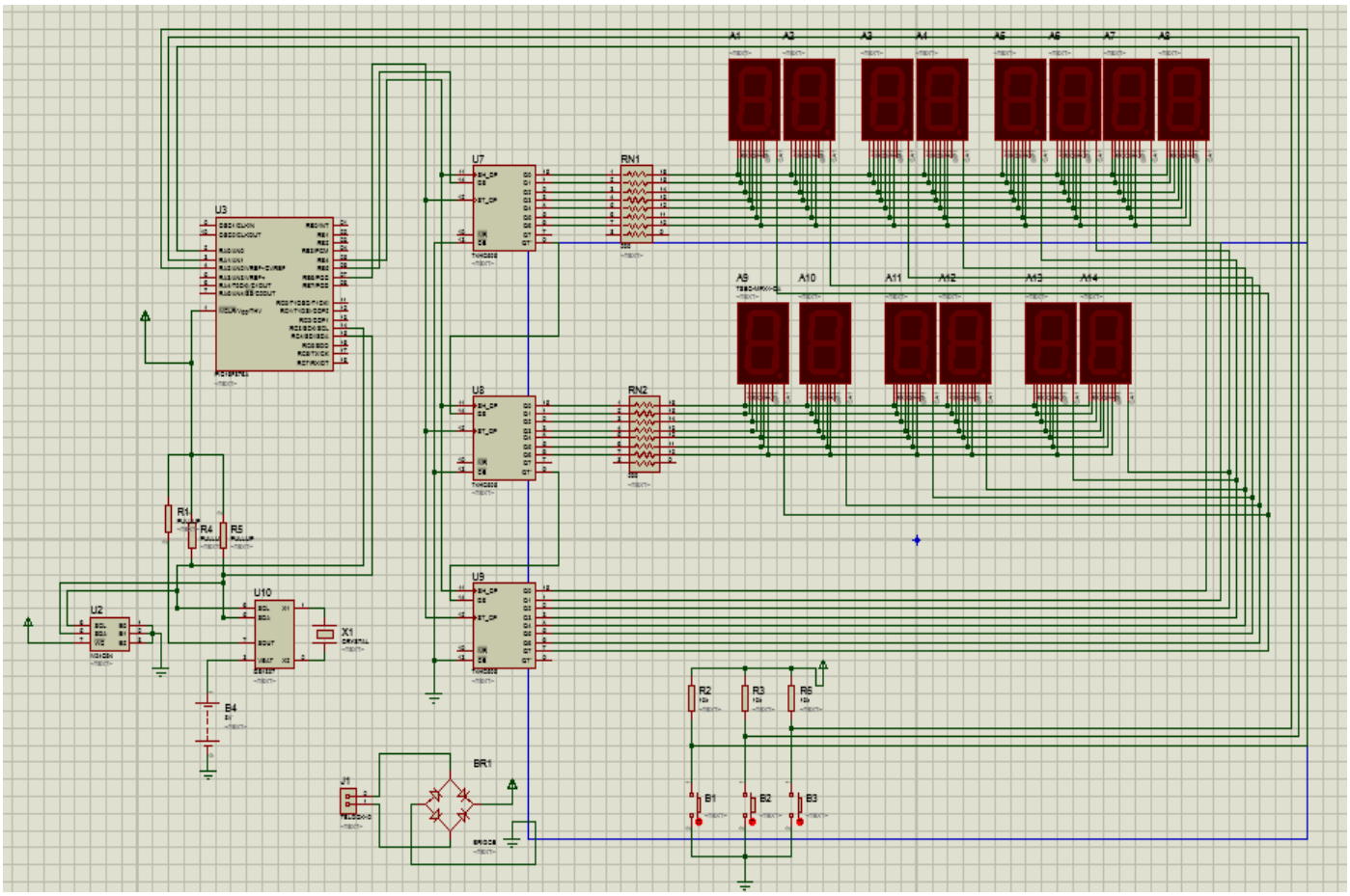
Avant de passer à la réalisation pratique de notre système nous avons eu recours à la simulation des différentes parties du système.

Pour cela on utilise le logiciel ISIS qui est un très bon logiciel de simulation en électronique. Isis est un éditeur de schémas qui intègre un simulateur analogique, logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût.

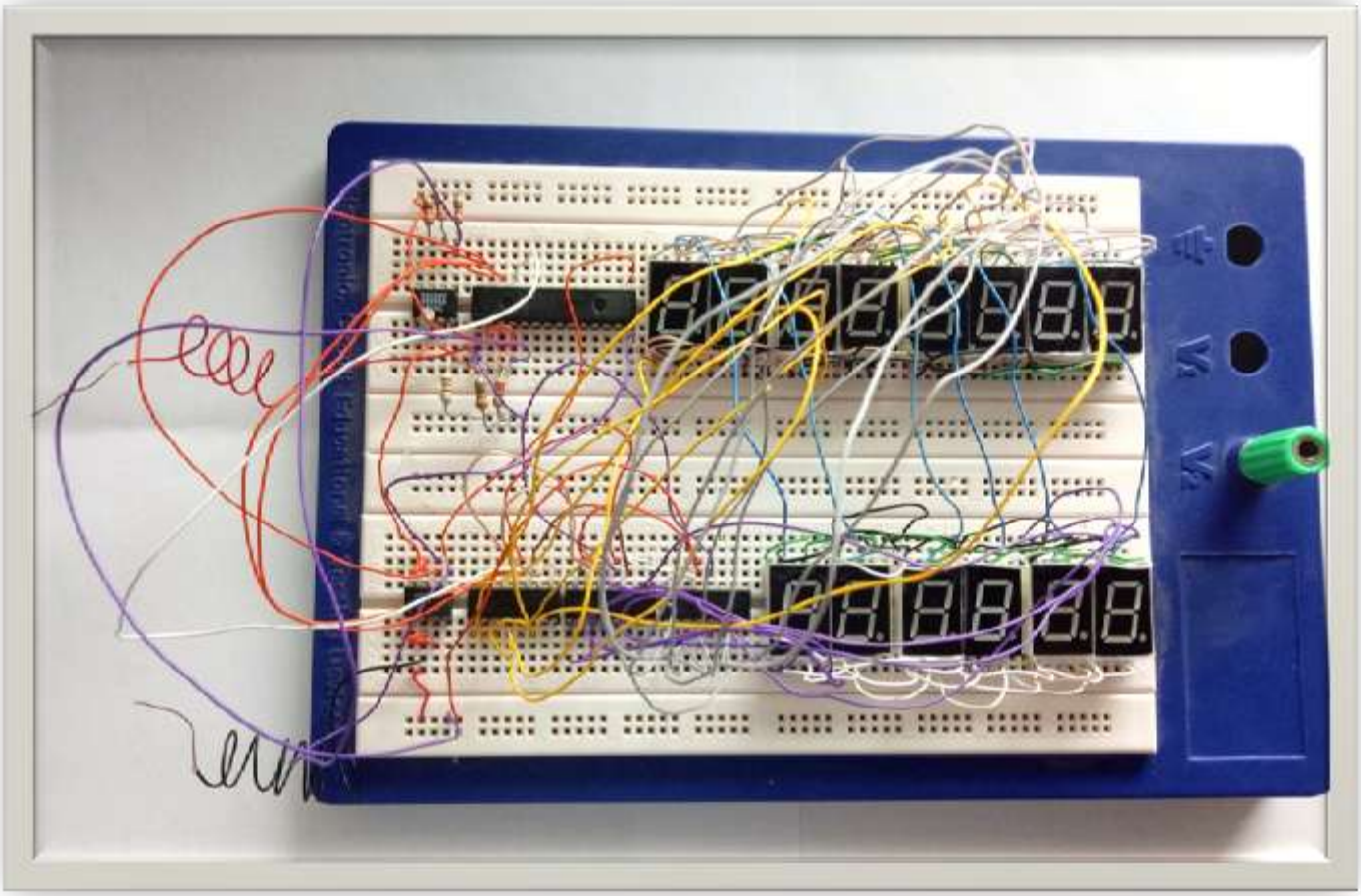
Il faut toujours prendre en considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel.

Voilà la simulation en ISIS figure (Fig22)



Fig(22):la simulation en ISIS

III. 6/ *Le circuit réel dans la plaque d'essai*:



Fig(23):Le circuit réel dans la plaque d'essai

Conclusion générale :

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation des microcontrôleurs (programmation embarquée). Nous avons aussi appris des nouvelles connaissances au niveau de la gestion du temps et des équipes.

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

- ✓ Créer un logiciel sous PC pour lire les données et les transférer vers des formats Standards.
- ✓ Améliorer la protection des entrées analogiques.
- ✓ Permettre à l'enregistreur de communiquer en temps réel avec un PC via interface Série (RS232) ou USB.
- ✓ Créer des cartes d'entrées spécifiques à chaque application.

Les améliorations qu'on peut apporter à notre travail sont énormes et peuvent varier Selon le type d'application qu'on souhaite.

Bibliographie :

[1] http://blewando.dlinkddns.com/cours/sin/006/fichier.pdf

[2] http://fabrice.sincere.pagesperso-orange.fr/cm_electronique/projet_pic/aidememoire/16F876A_bus_I2C/bus_I2C_16F876A.htm#3

[3] http://assets.nexperia.com/documents/data-sheet/74HC_HCT595.pdf

[4] https://www.carnetdumaker.net/articles/utiliser-un-module-horloge-temps-reel-ds1307-avec-une-carte-arduino-genuino/

[5] http://download.mikroe.com/documents/compilers/mikroc/pic/mikroc-pic-manual-v101.pdf

Annexes :

Liste des composants :

- R1,R2,R3,R4,R5,R6 10K
- U7,U8,U9 Registre à décalage 74HC595
- U3 PIC16F876A
- U2 M24C64
- U11 AFFICHEUR 7 SEGMENTS 1 digit Anode commune
- U10 DS 1307
- X1 Crystal(Quartz du DS1307) 20MHz
- B1,B2,B3 buttons poussoir
- B4 pile CR2032H