

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

جامعة أبي بكر بلقايد- تلمسان -

Université Aboubakr Belkaïd- Tlemcen -
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : **Electronique**

Spécialité : **INSTRUMENTATION**

Présenté par : HOULALA ILYES & SAIDI OMAR

Intitulé du Sujet

**Etude, conception et simulation d'une horloge-calendrier en
temps réel avec affichage des horaires de prières**

Soutenu en juin 2022 ,devant le jury composé de :

- | | | |
|------------------------------|--------------------------------|-------------|
| • BOUAZZA née GUEN Ahlam | PROF à l'Université de Tlemcen | Président |
| • BRIXI NIGASSA MED EL AMINE | MCB à l'Université de Tlemcen | Examinateur |
| • NEMMICHE Ahmed | MCB à l'Université de Tlemcen | Encadreur |
| • LACHACHI Djamel | MCB à l'Université de Tlemcen | CoEncadreur |

Anné Universitaire 2021-2022

Remerciements

Nous remercions et glorifions Allah le tout puissant et miséricordieux d'avoir guidé nos pas pour accomplir ce modeste travail et de le mener à terme. Ce dernier, a été effectué au sein de l'université (Abou Bekr Belkaid) de Tlemcen et plus précisément au département de génie électrique et électronique de la faculté de technologie.

Nous tenons à remercier vivement notre encadreur, Mr NEMMICHE AHMED, d'avoir accepté de nous guider tous le long de ce travail. Et cela, avec sa grande disponibilité, sa rigueur scientifique et ses précieux conseils qui nous ont permis de travailler dans les meilleures conditions, et ce, depuis le début de la préparation de notre projet.

Nous tenons à témoigner toute nos reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce mémoire :

Monsieur Lachachi Djamel, pour avoir contribué à ce travail et de nous avoir accordé des entretiens et répondre à nos questions , ainsi que de son expérience personnelle. Il a été un grand soutien dans l'élaboration de ce mémoire.

Nos parents, pour leur soutien constant et leurs encouragements.

Nous tenons à remercier, aussi, Madame GUEN Ahlam de nous avoir fait l'honneur, de bien vouloir accepter de présider ce jury.

Les remerciements sont, également, adressés à monsieur BRIXI NIGASSA MED EL AMINE Nous sommes très sensibles à l'honneur que vous nous faites en acceptant d'examiner le présent travail .

Par ailleurs, trouveront ici, nos sincères reconnaissances à tous les ingénieurs de laboratoires du département de génie électrique et électronique et à toute personne ayant contribué de près ou de loin à la réalisation de ce projet de fin d'étude.

Dédicace

Tout d'abord, je rends grâce à Dieu le tout puissant de m'avoir donné la force de mener à terme ce modeste travail que je dédie particulièrement:

A mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, mon père que Dieu lui prête longue vie, santé et joie.

A la lumière de mes jours, la source de mes efforts, ma mère que j'adore.

A ma mère qui m'a conseillé, aidé et encouragé durant toute la période de préparation de ce mémoire.

A mes frères qui m'ont été d'un grand soutien aussi bien moral que matériel.

Aux familles HOULALA, du plus grand au plus petit pour leurs encouragements.

A mon encadreur Dr NEMMICHE AHMED pour la qualité de ses orientations, de ses conseils, pour sa patience et sa disponibilité durant toute la période de ce travail.

A mes amis notamment, , , pour leur soutien et leur encouragement. A tous qui m'ont inculqué le savoir du primaire jusqu'à l'université.

HOULALA ILYES ABDELKADER

Dédicace

Je tiens tout d'abord à remercier ALLAH tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

Ce projet d'étude dédié à mon cher père qui a contribué toutes les années de sa vie à me soutenir et à renforcer ma confiance en moi, qui restera dans mon cœur

Au joyau le plus précieux de ma vie, à ma chère mère qui a souffert sans me laisser souffrir, qui n'a jamais cessé de me soutenir et de m'épauler pour que je puisse atteindre mon objectif. Qui n'a jamais dit non à mes exigences mon adorable mère que le dieu la protège et la garde

A ma chère sœur pour me soutenir et pour me donner ses conseils précieux durant toutes mes études

C'est un moment de plaisir de dédier ce projet à mes frères, en signe d'amour et de gratitude pour les sacrifices qu'ils m'ont toujours faits

A toutes la famille SAIDI

A Mr. NEMMICHE AHMED qui est toujours à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu me consacrer et sans leur contribution ce mémoire n'aurait vu le jour

Et enfin à mes amis, , qui n'ont jamais cessé de me soutenir pour faire tout ce que je pouvais dans tous les domaines de ma vie et surtout le domaine de mes études

Ce projet marque la fin de mes études comme l'aboutissement du soutien et de l'encouragement qui m'a été donnés tout au long de mes études, je les remercie infiniment pour ce dévouement très modeste.

SAIDI OMAR

Liste des abréviations

BCD: Binary Coded Decimal

EPROM : Erasable Programmable Read Only Memory (ca veut dire mémoire morte)

I2C : Inter Integrated Circuit Bus

Pwm : Pulse Width Modulation

PIC : Peripheral Interface Controller

CMOS : Complementary metal-oxide-semiconductor

LCD : liquid crystal display (affichage à cristaux liquides)

RAM : Random Access Memory

CPU : Unité central de traitement

TABLE DES FIGURES

Chapitre 1 :

Figure 1 : interface du logiciel mikroC

Figure 2 : interface du simulateur ISIS Proteus

Chapitre 2 :

Figure 3 : image réelle du PIC 18F2550

Figure 4 : différents pins du PIC 18F2550

Figure 5 : les pins du 74HC595

Figure 6 : les registres du DS1307

Figure 7 : configuration afficheur 7 segments

Figure 8 : les chiffres de 0 à 9

Figure 9 : l'alphabet de A à F.

Figure 10 : la table de vérité de 8 bits.

Chapitre 3 :

Figure 11: Schéma électrique complet du projet

Figure 12: simulation du circuit affichage de la date miladi et les horaires de prières

Figure 13 : simulation du circuit affichage de la date hijri et les horaires de prières

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : LE COMPILATEUR MIKROC ET LE SIMULATEUR ISIS

I.1. LE COMPILATEUR MIKROC ET LES PIC	4
I.2. COMPILATEUR MIKROC PRO.....	5
I.3. LE SIMULATEUR ISIS PROTEUS	6
I.4. CREATION DU PREMIER PROGRAMME EN MIKROC PRO.....	6

Chapitre II : ÉTUDE DES DIFFÉRENTS ÉLÉMENTS UTILISES

II.1. Microcontrôleur	10
II.1.1. Introduction	10
II.1.2. Classification des PICs de Microchip.....	10
II.1.3. Identification des PICs.....	10
II.1.4. pic 18F2550	11
II.1.4.1. caractéristiques	12
II.2 Registre à décalage 74HC595.....	13
II.2.1. qu'est ce que c'est 74HC595 ?.....	13
II.2.2. pourquoi on utilise 74HC595 ?.....	13
II.2.2.1. caractéristiques	13
II.2.2.2. Utilisations.....	14
II.2.3. comment fonctionne t-il ?.....	14
II.3 Module Horloge DS 1307.....	17
II.3.1. Introduction.....	17
II.3.2. caractéristique du circuit DS 1307.....	18
II.3.3. Le décimal codé binaire.....	19
II.3.4. le DS 1307	19
II.3.5. les registres du DS 1307	20
II.3.6. la Programmation du DS 1307.....	24
II.3.6.1. principe de programmation.....	24
II.4. le M24C64	24
II.4.1. caractéristique du circuit M24C64.....	24
II.5. le 7SEG-MPX2-CA.....	25
II.5.1. Affichage à sept segments	25
II.5.2. Représentations des caractères.....	25

II.5.2.1. Caractères décimaux	26
II.5.1.2. Caractères hexadécimaux.....	26
II.5.3. Programmation	27
II.6. Les batteries	28
II.7 bouton poussoir.....	28
II.7 résistance.....	28

Chapitre III : conception & programmation & simulation

III.1. conception.....	30
III.2. programmation.....	31
III.3. simulation.....	41
Conclusion générale	
Références biographiques	

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

Introduction générale

Introduction générale

Introduction générale

Le temps est important dans notre vie quotidienne ,Le temps est une notion qui rend compte du changement dans le monde.

Par exemple, une mesure de temps ne peut pas être considérée comme une mesure de charge. Dans ce qui suit, il faut comprendre « mesure de durée » plutôt que mesure de temps. La mesure de la durée, c'est-à-dire du temps qui s'écoule entre deux événements, est basée sur des phénomènes périodiques (le ciel, l'oscillation d'un pendule, etc.) ou quantiques (comme les temps de transition des électrons dans les atomes). La généralisation de la mesure du temps a transformé la vie quotidienne, la religion, la philosophie et la pensée scientifique. Pour la science, le temps est une mesure de l'évolution des phénomènes. Selon la théorie de la relativité, le temps est relatif (selon l'observateur, avec certaines limites), et l'espace et le temps sont étroitement liés, parfois même interchangeables.

Dans ce mémoire de Master, nous allons proposer une architecture d'un système qui constitue les six horaires de prière, la date et le temps actuel à base de deux logiciels Mirkoc (logiciel de compilation) et IsisProteus(construction des composants),et à partir d'un montage électronique sur IsisProteus et un programme de compilation sur Mirkoc .On doit indiquer les six temps de prière , la date et le temps actuel.[1]

Chapitre I :

LE COMPILATEUR MIKROC ET LE SIMULATEUR ISIS

I.1.LE COMPILATEUR MIKROC ET LES PIC

Un microcontrôleur est un circuit électronique conditionné dans un circuit intégré avancé. Les microcontrôleurs sont vendus par différents fabricants tels que Motorola, Intel, Philips, Microchip, etc. Microchip est surtout un fabricant de circuits électroniques. Dans leur gamme, il existe des microcontrôleurs PIC, il existe différentes séries, certaines d'entre elles sont : 12F, 16F, 18F, 24F, 30F, 33F, etc. Les développeurs de projet choisissent la famille et la référence qui correspondent à leurs besoins, comme le PIC12F675 ; un microcontrôleur à 8 broches avec des modules intégrés de base tels que des minuteries et des ADC. Un microcontrôleur comme le 16F877 possède 40 broches et des modules tels que : Timer, ADC, USART, I2C, PWM, etc. La différence entre ces deux images peut être facilement vue.

La documentation technique, la commercialisation à grande échelle et une multitude d'informations publiées sur les microcontrôleurs PIC en font un outil idéal pour l'enseignement. Microchip gère un portail Web, www.microchip.com, où vous pouvez télécharger des informations et des applications logicielles qui facilitent le développement avec ses microcontrôleurs. Le développement d'applications basées sur un microcontrôleur PIC comprend l'identification des problèmes de développement, l'édition, le débogage des programmes de la machine et la programmation du microcontrôleur à l'aide du programmeur spécifique de PICmicro. Microchip propose des programmeurs professionnels de différentes tailles, le plus populaire étant peut-être PICSTART plus, mais il en existe d'autres tels que PICkit2, PICkit3. Bien qu'il existe des programmeurs commerciaux, Les développeurs peuvent créer ou acheter des programmeurs éducatifs à faible coût. L'architecture de base d'un microcontrôleur est similaire à celle d'un PC, avec un bloc OTP ou une mémoire flash pour stocker les instructions du programme, cette partie est similaire au disque dur d'un ordinateur, un PICmicro a une RAM, qui agit de la même manière que l'accès aléatoire mémoire d'un ordinateur personnel, et un microcontrôleur possède des ports d'entrée et de sortie, similaires aux périphériques d'entrée et de sortie d'un ordinateur, tels qu'une souris, une imprimante, un écran, un clavier, etc.

Ces caractéristiques des microcontrôleurs les rendent idéales pour créer de petites applications avec des interfaces utilisateur telles que des claviers, des boutons, des lecteurs de mémoire de masse et des capteurs de différentes grandeurs telles que la température, l'humidité, la pression,

la lumière, la proximité, etc. . De même, des environnements de visualisation avec des affichages numériques, alphanumériques et graphiques peuvent être créés. Les ports série tels que USART et USB peuvent créer une communication série et sans fil avec d'autres appareils. Bref, les possibilités sont infinies.[2]

I.2. LE COMPILATEUR MIKROC PRO

La manipulation des microcontrôleurs a besoin d'un code machine, il est défini comme code assembleur, le code assembleur est détaillé et dure à écrire. Le programme en code assembleur est très long et difficile à comprendre. La création et l'édition de programmes devient facile à cause de la création des compilateurs de haut niveau, bien entendu les microcontrôleurs ne font pas exception. Dans le commerce, il y a plusieurs compilateurs des différents fabricants et avec différents langages de haut niveau. On peut acheter les compilateurs PICC, CCS, PIC Basic, entre autres. Ce cours se base sur le compilateur MikroC PRO, qui est un compilateur de langage C pour les microcontrôleurs PICmicro famille 12F, 16F, 18F etc. MikroC PRO est un logiciel qui contient plusieurs helps, supports et d'outils, qui facilite la création des projets et des applications pour les microcontrôleurs PICmicro. L'étude de cet environnement de développement est possible, parce que l'étudiant peut télécharger une démo ou version étudiant, qui a les mêmes caractéristiques que la version complète, juste que la taille du code machine ne peut pas excéder 2 Koctets, Une capacité suffisante pour un premier apprentissage. Le compilateur de haut niveau en langage C marche à base des structures, facilitant la programmation, optimisant les opérations mathématiques, et les processus à cause de l'utilisation des fonctions prédéfinies et des fonctions que le développeur peut créer et l'utilisation d'un groupe de variables, caractères, nombres entiers, et réels. Le compilateur crée automatiquement le code assembleur et un code enregistré dans un fichier avec l'extension *. Hex, ce fichier est le résultat principal de la compilation, qui nous aide à programmer le microcontrôleur ou le simuler sur ordinateur.[3]

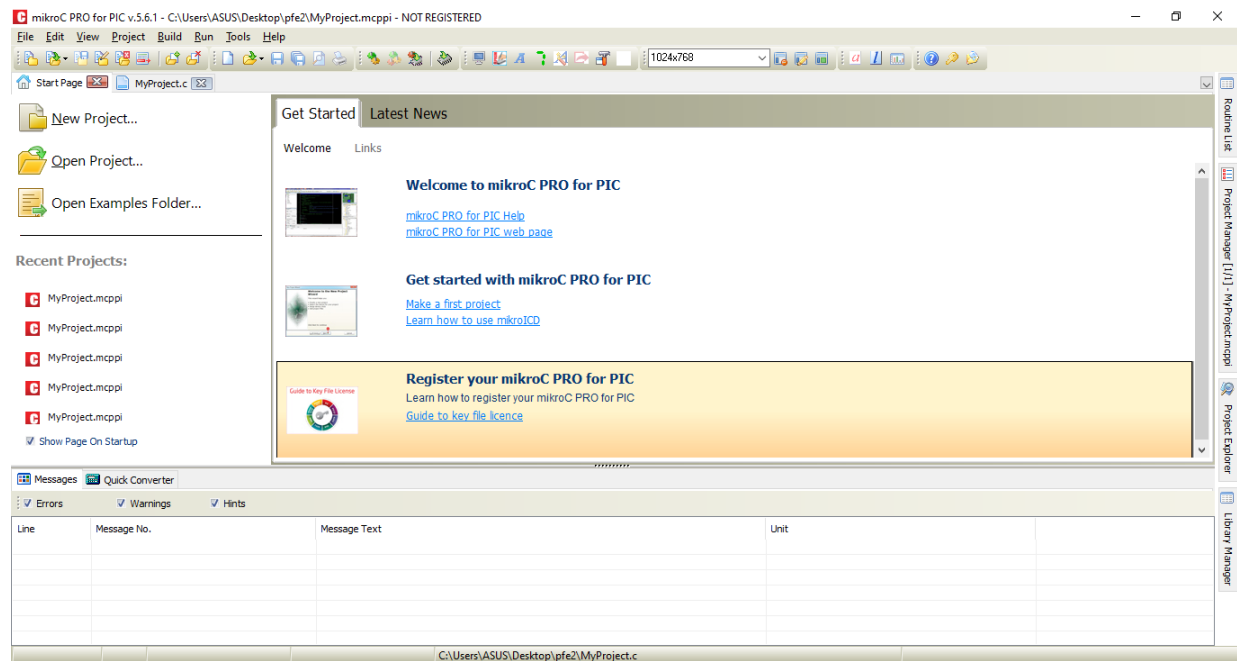


Figure 1 : interface du logiciel mikroC

I.3.LE SIMULATEUR ISIS PROTEUS

Le simulateur ISIS de Proteus est un logiciel utile, développé par la compagnie électronique Labcenter, il est créé depuis 10 ans c'est l'outil le plus efficace pour simuler les microcontrôleurs PIC. L'ISIS permet la simulation des familles des PIC les plus populaires. En plus des PIC, ISIS peut simuler différents dispositifs numériques et analogiques, comme les afficheurs sept segments, les LCD, les capteurs de température, d'humidité, la pression, etc. Il peut aussi simuler les actionneurs tels que des moteurs à courant continu, les servomoteurs, etc. Il simule des périphériques d'entrée et de sortie comme les claviers, ordinateur et les ports physiques tels que RS232 et USB. Ce simulateur a plusieurs instruments de mesure tels que voltmètres, ampèremètres, oscilloscopes et analyseurs de signaux. En conclusion, ces caractéristiques font d'ISIS de Proteus, un outil super pour la conception et l'étude des PIC. Vous pouvez télécharger à partir du site web : www.labcenter.com.

I.4.CREATION DU PREMIER PROGRAMME EN MIKROC PRO

Le processus suivant doit être appris par cœur, pour pouvoir faire marcher les PIC. En lançant MikroC PRO, on identifie dans le menu supérieur l'outil Project, et on choisit New Project... avec cette action, le programme affiche un assistant facile à utiliser pour créer un nouveau projet.

Chapitre I : le compilateur mikroc et le simulateur isis

La prochaine étape est de cliquer sur le bouton Next, à cette étape la case de la référence de PIC s'affiche, qu'on souhaite utiliser. Dans ce champ, on sélectionne le PIC avec lequel on veut travailler. Ensuite on définit la fréquence d'oscillation avec laquelle travaillera le PIC. Dans cet exemple on sélectionne 4.000000 MHz. La fenêtre suivante demande où on veut enregistrer le projet. Dans ce répertoire le programme enregistrera tous les fichiers nécessaires, le fichier hex est utilisé pour l'exécution. Enfin, la configuration est terminée et le projet est créé.

Après l'écriture du programme, on compile le programme. Les résultats de la compilation se trouvent en bas de la fenêtre du programme. Si le programme est juste, le message affiché est donc « Finished successfully ». Pour commencer l'édition d'un projet, on configure les ports du PIC, puis on insère le programme. Le PIC 16F84A dispose de deux ports A et B. On manipule le registre TRIS pour configurer les ports en sortie ou entrée. Les registres TRIS ont le même nombre de bits que les ports, par exemple le port B ou PORTB de ce PIC est de 8 bits, donc le TRISB comporte également 8 bits. Les bits des registres TRIS correspondent à ceux des ports, et définissent bit à bit l'état du port. Si un bit du TRIS est 0, le même bit du port est en sortie, et si un bit du TRIS est 1 le même bit du port est en entrée ou en haute impédance.[4]

Pour voir ce concept plus clairement, on observe et on analyse l'exemple suivant. Cet exemple utilise un bouton et deux LED pour afficher le comportement du programme. On observe et on analyse le programme ci-dessous :

```
void main ( void ) {
unsignedint COMPTEUR=0 ;
TRISB = 0b11110000 ;
PORTB=0b00000000 ;
while( 1 )
    { if( PORTB.F7==0 ) { if( PORTB.F0==1 ) PORTB.F0=0;
      else PORTB.F0=1;
      while( PORTB.F7==0); }
    COMPTEUR++;
    if(COMPTEUR&0x0100 )
    PORTB.F1=1 ;
    else PORTB.F1=0 ; } }
```

L'étape suivante est de faire la simulation sur ISIS, les résistances des LED doivent être changées à 330Ω, l'entrée Master Clear, MCLR doit être connectée à Vcc pour que le PIC ne redémarre pas, à la fin on devrait voir la forme suivante :

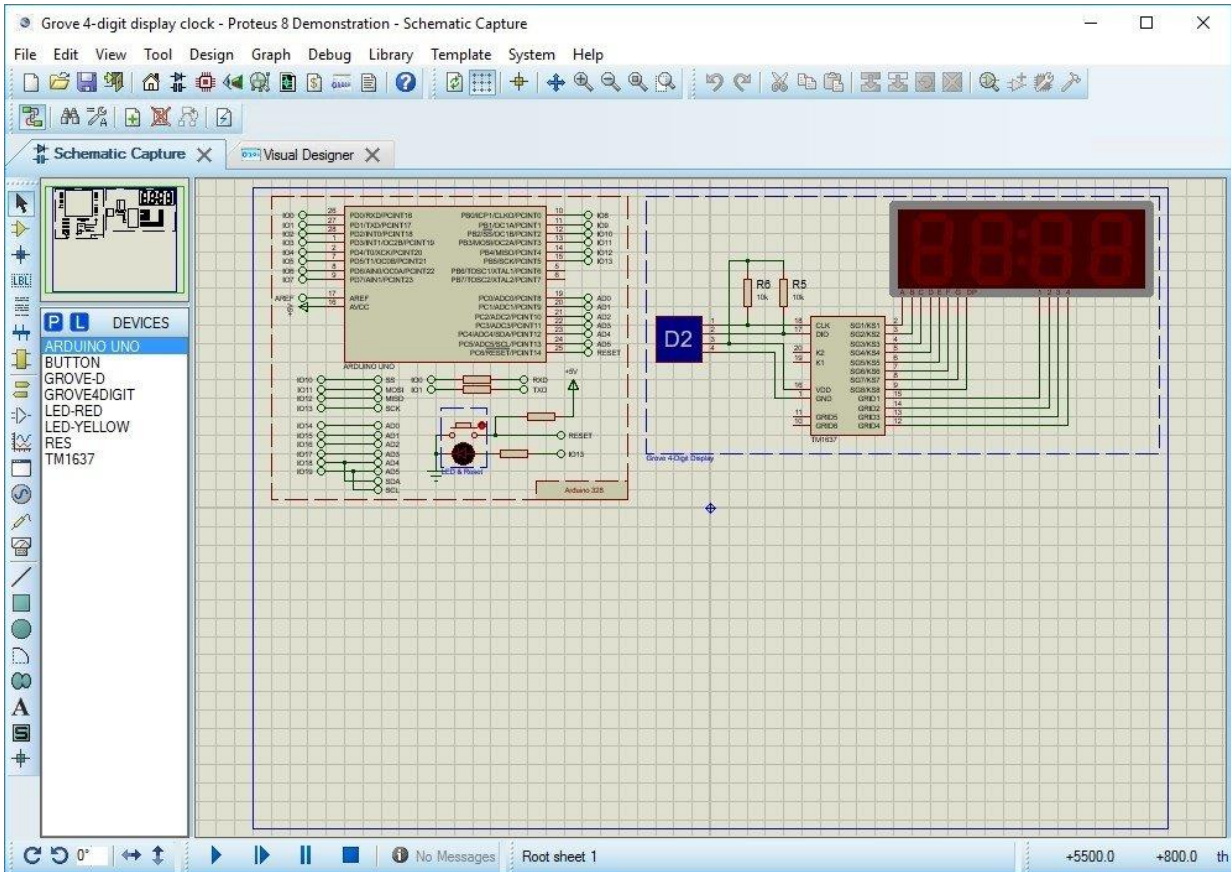


Figure 2 : interface du simulateur isis proteus

Avant de lancer la simulation, on doit charger le fichier. Hex. Pour procéder, on double clique sur le PIC, et une fenêtre qui permet de rechercher le fichier. Hex et on peut également ajuster la fréquence d'oscillation. Par défaut, cette valeur est de 4 MHz.[5]

Conclusion

Cette étude a permis de constater que les deux logiciels peuvent interagir plus facilement entre eux depuis qu'ils utilisent des protocoles de communication, il est maintenant clair que l'évolution qui a mené rend le travail plus simple.

Chapitre II : ETUDE DES DIFFERENTS ELEMENTS UTILISES

II.1. Microcontrôleur

II.1.1 Introduction

Les microcontrôleurs se trouvent dans maintes applications publiques ou professionnelles, avec plusieurs familles.

La société Américaine Microchip Technologie est créée dans les années 90 sur un microcontrôleur CMOS : le PIC (Peripheral Interface Controller). Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

- Les PIC existent dans plusieurs versions :
- les UVROM qui sont effaçable par une source de rayonnements ultraviolets
- les OTPROM programmable une seule fois
- les E²PROM et flash EPROM qui sont effaçables électriquement.

II.1.2 Classification des PICs de Microchip

les modèles Microchip, sont classés en 3 familles, contient chacune différentes références. Ces familles sont :

- Base-line : les instructions sont codées sur 12 bits.
- mid-line : les instructions sont codées sur 14 bits.
- High-line : les instructions sont codées sur 16 bits.

II.1.3 Identification des PICs

La forme générale de la référence du pic se présente comme ça : **xx(L)Xxyy-zz**

Xx : famille du composant, actuellement « 12, 14, 16, 17 et 18 ».

L : tolérance plus importante de la plage de tension

XX : type de mémoire programme

C : EPROM ou EEPROM

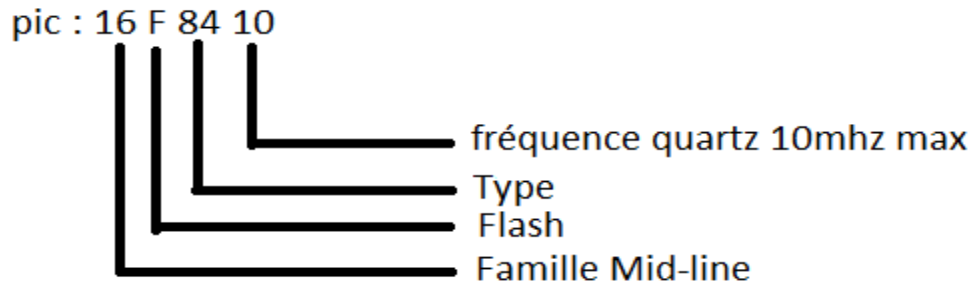
CR : PROM

F : Flash

yy : Identificateur.

Zz : vitesse maximale du quartz de pilotage.[6]

Exemple :



II.1.4 Pic 18F2550



Figure 3 : image réelle du pic 18F2550

28-Pin PDIP, SOIC

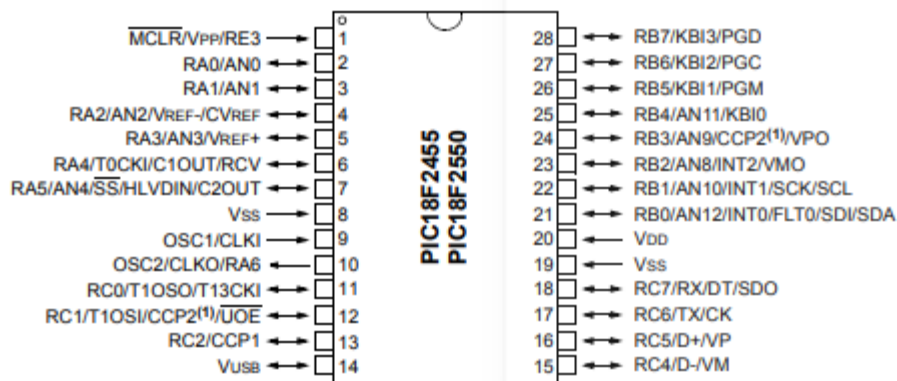


Figure 4 : différents pins du pic 18F2550

II.1.4.1 Caractéristiques

Fabriquant : Microchip

Catégorie du produit : Microcontrôleurs 8 bits – MCU

Série : PIC18(L)Fxx50

Style de montage : Through Hole

Package/Boîte : SPDIP-28

Cœur : PIC18

Taille de la mémoire du programme : 32 kB

Largeur du bus de données : 8 bit

Résolution CAN : 10bit

Fréquence de l'horloge max. : 48 MHz

Nombre d'E/S : 24 I/O

Taille de la RAM de données : 2 kB

Tension d'alimentation de fonctionnement : 2 V to 5.5 V

Température de fonctionnement min. : - 40 C

Température de fonctionnement max. : + 85 C

Conditionnement : Tube

Marque : MicrochipTechnology / Atmel

Type de Ram de données : SRAM

Taille de la ROM de données : 256 B

Type de Rom de données : EEPROM

Hauteur : 3.3 mm

Type d'interface : EAUSART, I2C, SPI

Longueur : 34.67 mm

Nombre de canaux CAN : 10 Channel

Nombre d'horloges/de compteurs : 4 Timer

Gamme de processeur : PIC18

Produit : MCU

Type de produit : 8-bit Microcontrollers – MCU

Type de la mémoire programme : Flash 15

Sous-catégorie : Microcontrollers – MCU

Tension d'alimentation – Max. : 5.5 V

Tension d'alimentation – Min. : 4.2 V

Nom commercial : PIC

Largeur : 7.24 mm

Poids de l'unité : 22 g

II.2 Registre à décalage 74HC595

II.2.1 Qu'est ce que c'est 74HC595 ?

C'est un circuit intégré qui implémente un circuit CMOS à l'intérieur. Plus clairement, il s'agit d'un registre à décalage de 8 bits, ses valeurs en sortie correspondent à des valeurs d'entrée et des valeurs mémorisées avant.

II.2.2 Pourquoi on utilise 74HC595 ?

Leur importance est de convertir une information série en parallèle. Et aussi pour économiser le nombre de broche utilisée en sortie.[7]

6.2 Pin description

Table 2. Pin description

Symbol	Pin	Description
Q1	1	parallel data output 1
Q2	2	parallel data output 2
Q3	3	parallel data output 3
Q4	4	parallel data output 4
Q5	5	parallel data output 5
Q6	6	parallel data output 6
Q7	7	parallel data output 7
GND	8	ground (0 V)
Q7S	9	serial data output
MR	10	master reset (active LOW)
SHCP	11	shift register clock input
STCP	12	storage register clock input
OE	13	output enable input (active LOW)
DS	14	serial data input
Q0	15	parallel data output 0
Vcc	16	supply voltage

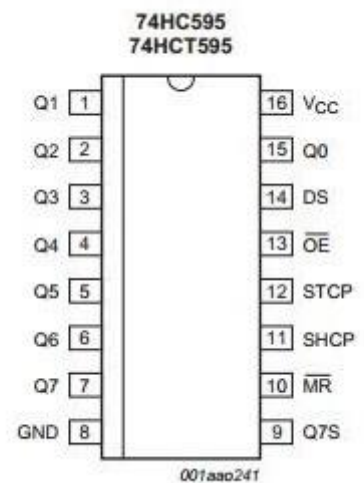


Figure 5 : les pins du 74HC595

II.2.2.1 Caractéristiques

- 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, 3 états
- Compatible TTL, LSTTL
- Câblage 2*8 broches
- Décalages jusqu'à 100 MHz
- Température de -40 à +85°C

II.2.2.2 Utilisations

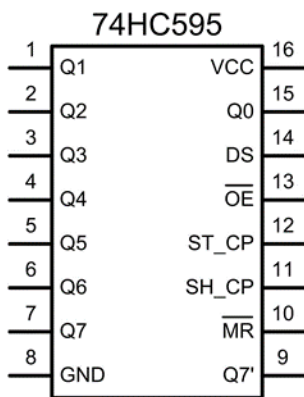
- Conversion série / parallèle
- Chenillards de LED ou d'éclairages
- Registre de stockage de télécommande
- Commande de réseaux de diodes LEDE
- Multiplexage

Alimentation 5V

Alimentation Vcc : 7 V maxi

Courant de sortie : pinQ7S pour montage en cascade 25mA, pins Qn 35 mA.[8]

II.2.3 Comment fonctionne t-il ?



Etape 1 : on place la valeur sur la broche DS

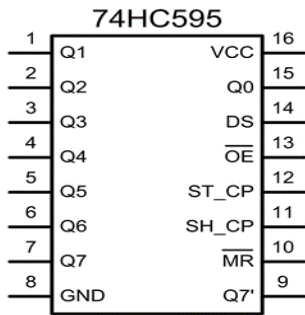
1

DS

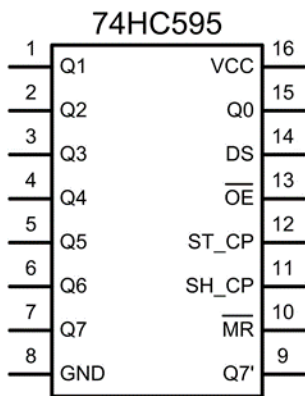
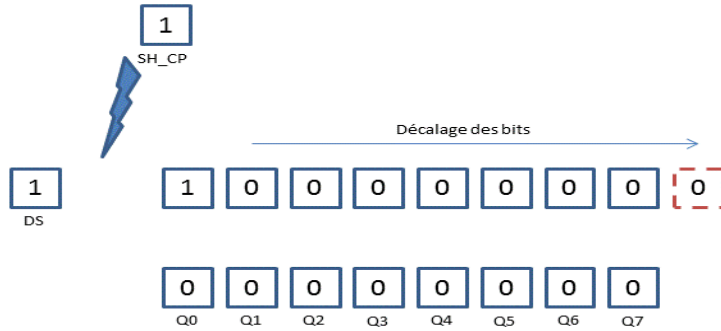
0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

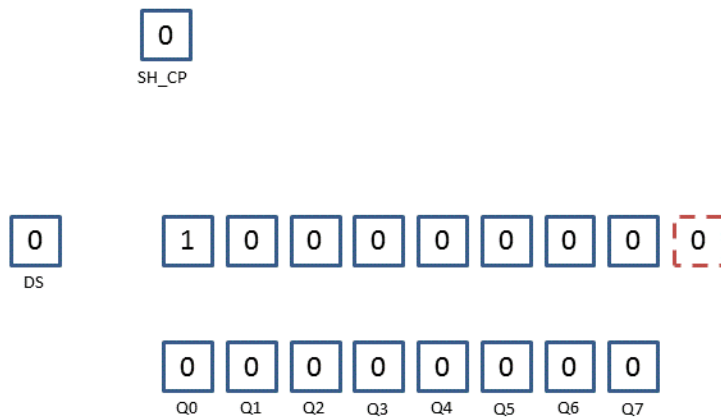
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7

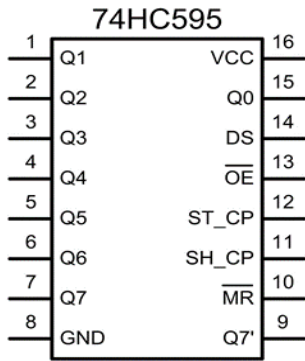


Etape 2 : la broche SH_CP est mise à 1
La valeur de la broche DS avance donc dans le registre

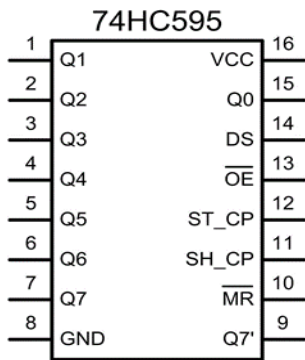
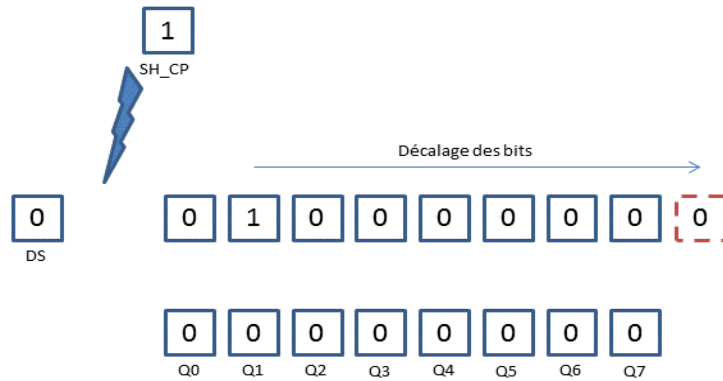


Etape 3 : remise à 0 de la broche SH_CP
On place la valeur du bit suivant sur la broche DS

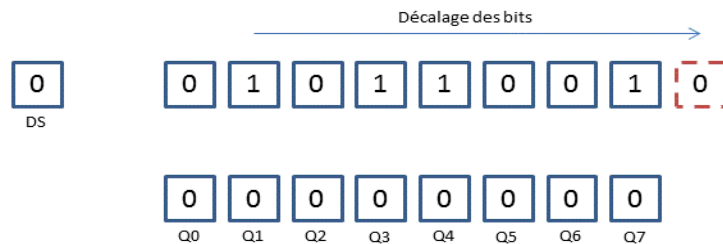


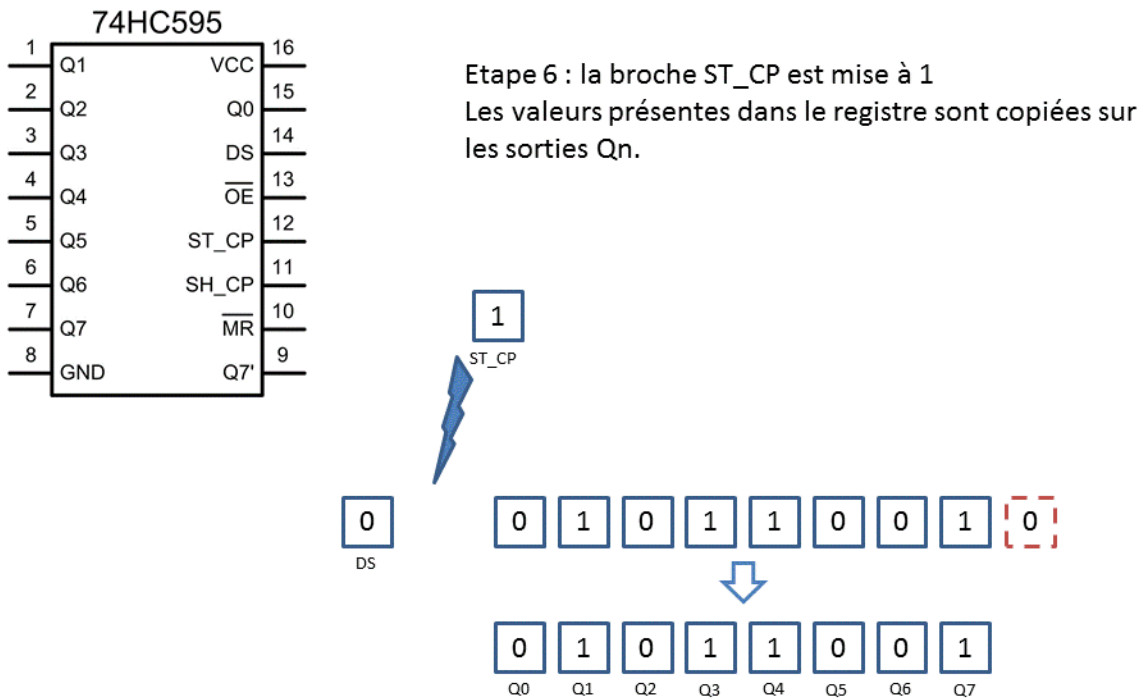


Etape 4 : la broche SH_CP est mise à nouveau à 1
La valeur de la broche DS avance donc dans le registre



Etape 5 : Les étapes 3 et 4 sont répétées autant de fois que nécessaire afin de remplir le registre. Toutefois si l'on répète plus de 8 fois l'opération, seuls les derniers 8 bits seront retenus.





il y a 2 grandes phases :

- La première on met les bits que l'on veut
- La deuxième est la confirmation des données.[9]

II.3 Module Horloge DS 1307

II.3.1 Introduction

Le module horloge DS 1307 est une horloge/calendrier en temps réel RTC (real time clock) qui permet de donner :

- L'Heure courante :
 - L'Heure au format de 12 ou 24 heures
 - Les Minutes.
 - Les secondes.
- Les dates :
 - jours
 - mois (avec ajustement des années bissextiles).
 - Années (jusqu'au 2100).

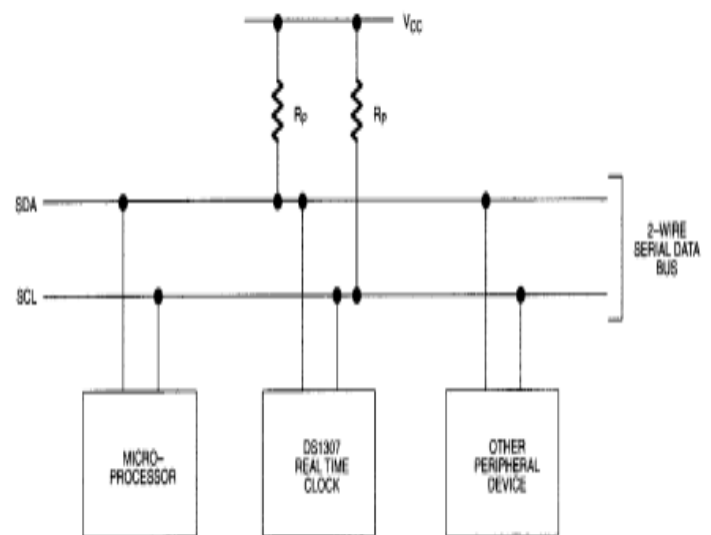
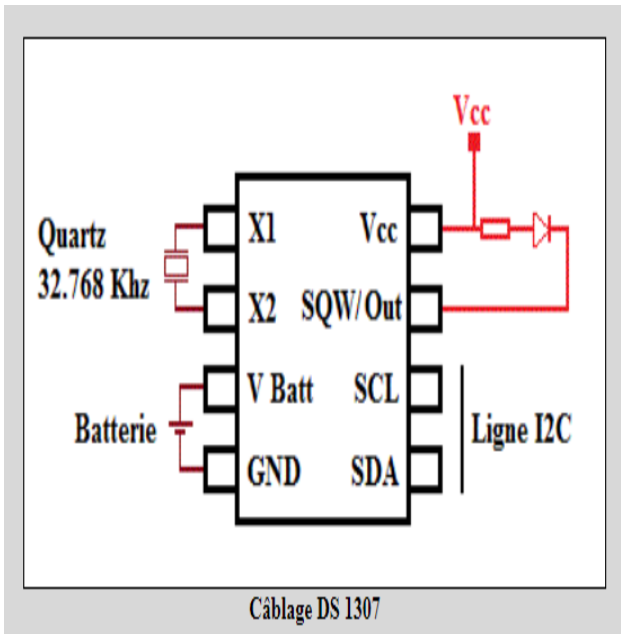
Il existe plusieurs modules contenant un circuit DS 1307.

- Certains modules sont constitués uniquement du DS 1307.
- Sur d'autres une mémoire externe est ajoutée.

II.3.2 Caractéristique du circuit DS 1307

- La commande appel ou programmation s'effectue via un bus I2C.
- Le circuit possède 56 octets de RAM intégrés et disponibles pour l'utilisateur.
- Le DS 1307 possède un circuit de détection de puissance intégré qui détecte les pannes de courant et passe automatiquement sur l'alimentation de secours.
- L'alimentation de secours peut être constituée d'une pile ou d'une batterie. Sur certains modules le réceptacle de la pile est intégré.
- Le circuit possède une sortie qui génère un signal carré.
- L'adresse du composant sur le bus I2C n'est pas réglable (Adresse DS 1307 : 1101000).si d'autres composants sont montés sur le bus I2C il faudra leur choisir une adresse différente.
- La programmation du circuit s'effectue en **BCD** (Binary Coded Decimal) : **DCB** (Décimal Codé Binaire).
- L'alimentation du circuit doit être comprise entre 4.5 et 5.5 volts.

L'appel et la programmation de la date ou de l'heure se fait en utilisant le protocole du bus I2C.



II.3.3 Le décimal codé binaire

Le code BCD (Binary Coded Decimal) représente un chiffre par un ensemble de 4 bits au lieu de 8 habituellement.

Pour rapprocher de l'écriture d'un chiffre en base dix, en BCD ce chiffre est codé sur 4 bits seulement.

CHIFFRE	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Représentation BCD

Ainsi un nombre est représenté par des mots de 4 bits représentant chaque chiffre des nombres indépendamment les uns des autres.

Exemple :

CHIFFRE	BCD		
5	0101		
	5		
15	0001	0101	
	1	5	
153	0001	0101	0011
	1	5	3

Représentation des chiffres et nombres en BCD

En utilisant le **BCD**, la manipulation des données numériques se fait sur 4 bits au lieu de 8, on peut simplifier l'électronique nécessaire pour le fonctionnement du circuit.

II.3.4 le DS 1307

Le DS 1307 possède en interne des registres de 8 bits qui contient l'heure, la date,...

Leurs utilisations se fait en utilisant le codage BCD.

Ainsi comme les 8 registres contenus dans les adresses allant de 00h à 07h (h= Hexadécimale)

sont des registres 8 bits , ils sont programmés comme si les registres ne faisaient que 4 bits

Les bits 0 à 3 contiennent une partie de la valeur (les unités) et les bits 4 à 7 l'autre partie de la valeur (les dizaines).

II.3.5 les registres du DS 1307

La programmations ou la lecture du DS 1307 s'effectue via des registres 8 bits codées 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 et 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 décimal).

	BIT7							BIT0
00H	CH	10 SECONDS		SECONDS				00-59
	0	10 MINUTES		MINUTES				00-59
	0	12 / 24	10 HR / A/P	10 HR	HOURS			01-12 / 00-23
	0	0	0	0	0	DAY		1-7
	0	0	10 DATE		DATE			01-28/29 / 01-30 / 01-31
	0	0	0	10 MONTH	MONTH			01-12
	10 YEAR			YEAR				00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0

Figure 6 : les registres du ds1307

Registres des secondes : 00h

Les valeurs de ce registres est compris entre 00 à 59 il est nécessaire d'utiliser les deux parties de ce registre :

Les bits 0 à 3 contiennent les valeurs des unités allant de 0 à 9.

Les bits 4 à 6 contiennent les valeurs des dizaines allant 1 à 5.

Le bit 7 est utilisé pour permettre le fonctionnement de l'horloge ou pour l'arrêter :

Lorsque le bit est à 1 l'horloge fonctionne.

Lorsque le bit est à 0 (par default) l'horloge est arrêtée.

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	

Registre 00h

Registre des Minutes : 01h

Identique au registre des secondes sauf pour le bit 7 qui n'est pas utilisé.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
01h	0	(10) Minutes			Minutes			Minutes	00 à 59	

Registre 01h

Registre des Heures : 02h

Le DS 1307 permet un réglage de l'heure en mode 12 ou 24 heures.

*Les bits 0 à 3 contiennent les valeurs des unités allant de :

- 0 à 9 en mode 12 heures.
- 0 à 3 en mode 24 heures.

*Le bit 6 contient le mode de fonctionnement de l'horloge. Ce mode est défini par :

- 1 pour le mode 12 heures.
- 0 Pour le mode 24 heures.

*Le bit 5 est fonction du bit 6. Il permet soit d'indiquer le mode de fonctionnement de l'horloge , AM/PM en mode 12 heures, soit l'indication des dizaines d'heures en mode 24 heures :

- Pour le mode 12 heures le contenu de ce bit indique :
 - 0 pour AM.
 - 1 pour PM.
- Pour le mode 24 heures les bits 4 et 5 sont utilisés pour indiquer les dizaines d'heures :
 - 0,1 et 2.

Chapitre II : étude des différents éléments utilisés

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
02h	0	12	PM / AM	(10) Heure	HEURES				Heure	1 à 12 +AM/PM 00 à 23
		24	(10) Heure							

Registre 02h

Registre des jours : 03h

Les jours de la semaine sont indiqués comme suit : 1 pour Dimanche, 2 pour Lundi,
Ainsi seulement 3 bits (0,1,2) sont nécessaires pour coder les jours de la semaine.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
03h	0	0	0	0	0	JOURS			Jour	01 à 07

Registre 03h

Registre de la date : 04h

Le principe est le même que pour le registre des minutes ou d'heures.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
04h	0	0	(10) Date		Date				Date	01 à 31

Registre 04h

Registre des mois : 05h

Le principe est le même que pour le registre des minutes ou d'heures.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
05h	0	0	0	(10) Mois	MOIS				Mois	01 à 12

Registre 05h

Registre des années : 06h

Le principe est le même que pour le registre des minutes ou d'heures.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
06h	(10) Année				Année				Année	00 à 99

Registre 06h

Registre du contrôle : 07h

Ce registre permet de contrôler une sortie qui peut :

- être utilisée en tant que sortie logique.
- générer un signal carré de fréquence variable.

ADRESSE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FONCTION	CONTENU
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Contrôle	-----

Registre 07h

Sortie logique

Pour être utilisée en tant que sortie logique :

Le bit 4 (SQWE) du registre de contrôle doit impérativement être à 0.

Le niveau logique est alors déterminé par le bit 7 (OUT) :

- Si OUT est à 0, la sortie est au niveau bas.
- Si OUT est à 1, la sortie est au niveau haut.

Par défaut OUT est à 0.

Générer un signal carré de fréquence variable

Pour être utilisée en tant que générateur de signal carré :

Le bit 4 (SQWE) du registre de contrôle doit impérativement être à 1 pour activer la sortie de l'oscillateur.

Le bit 4 (SAQWE) du registre de contrôle est par défaut à 0.

La fréquence du signal carré dépend alors de la valeur des bits 0 et 1 (RS0 ET RS1)

Lors de la première mise sous tension de l'appareil ces bits sont généralement mis à 1.

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

Sortie à fréquence variable

II.3.6 La Programmation du DS 1307

La programmation du DS 1307 s'effectue via le bus I2C.

Ce bus permet la programmation et la lecture du DS 1307 en utilisant seulement 2 fils.

L'adresse du composant sur le bus I2C n'est pas réglable : Adresse DS 1307 : 1101000.

La programmation du circuit s'effectue en BCD (Binary Coded Decimal) : DCB(Décimal Codé Binaire).

II.3.6.1 Principe de programmation

- 1) Déclarer la bibliothèque I2C.
- 2) Déclarer l'adresse du DS 1307 sur le bus I2C.
- 3) Déclarer les fonctions de conversion.
Fonction décimale vers décimale codé binaire.
Fonction décimale codé binaire vers décimale.
- 4) Envoyer la trame de lecture ou de programmation.
- 5) Libérer le bus I2C.[10]

II.4 Le M24C64

II.4.1 Caractéristique du circuit M24C64

Fabricant : STMicroelectronics

Catégorie du produit : EEPROM

Style de montage : SMD/SMT

Taille de la mémoire : 64 kbit

Tension d'alimentation-min : 1.8v

Tension d'alimentation-max : 5.5v

Température de fonctionnement min : -40 c

Température de fonctionnement max : +85 c

Fréquence de l'horloge max : 1 MHz

Temps d'accès : 450 ns

Rétention des données : 200 ans

Courant d'alimentation maximale : 2.5 ma

Hauteur : 1.65 mm

Longueur : 5mm

Courant d'alimentation de fonctionnement : 3mA

Tension d'alimentation de fonctionnement : 2.5v , 3.3v , 5v

Tension de programmation : 1.8 v to 5.5 v

Largeur : 4mm

II.5 Le 7SEG-MPX2-CA

II.5.1 Affichage à sept segments

L'affichage à sept segments est une technologie d'affichage basée sur sept segments qui peut être activée ou désactivée en fonction du mode graphique à générer. Ce type d'affichage est principalement utilisé pour afficher des nombres décimaux et hexadécimaux. C'est une alternative aux affichages matriciels plus complexes. L'affichage à sept segments est largement utilisé dans les horloges numériques, les montres électroniques, les calculatrices et autres appareils électroniques qui affichent des informations numériques.[12]

II.5.2 Représentations des caractères

Les sept segments sont disposés en un rectangle composé de deux sections verticales de chaque côté et d'une section horizontale en haut, au milieu et en bas. Dans la plupart des applications, ces segments ont une forme et une taille presque uniformes (généralement des hexagones allongés, mais des trapèzes et des rectangles peuvent également être utilisés). Dans les calculatrices, les segments de ligne verticale sont longs et ont des extrémités de forme étrange pour améliorer la lisibilité. Les sept éléments de l'affichage peuvent s'allumer dans différentes combinaisons pour représenter les chiffres arabes. Les segments sont désignés par les lettres A à G. Un point décimal facultatif (le huitième segment, appelé DP) est utilisé pour afficher les nombres non entiers 2,3. Un octet peut coder l'état complet d'un afficheur 7 segments, inclure la virgule décimale. Les encodages de bits les plus populaires sont gfedcba et abcdefg. Dans la représentation gfedcba, la valeur d'octet 0x06 active les segments « c » et « b », qui afficheront « 1 »

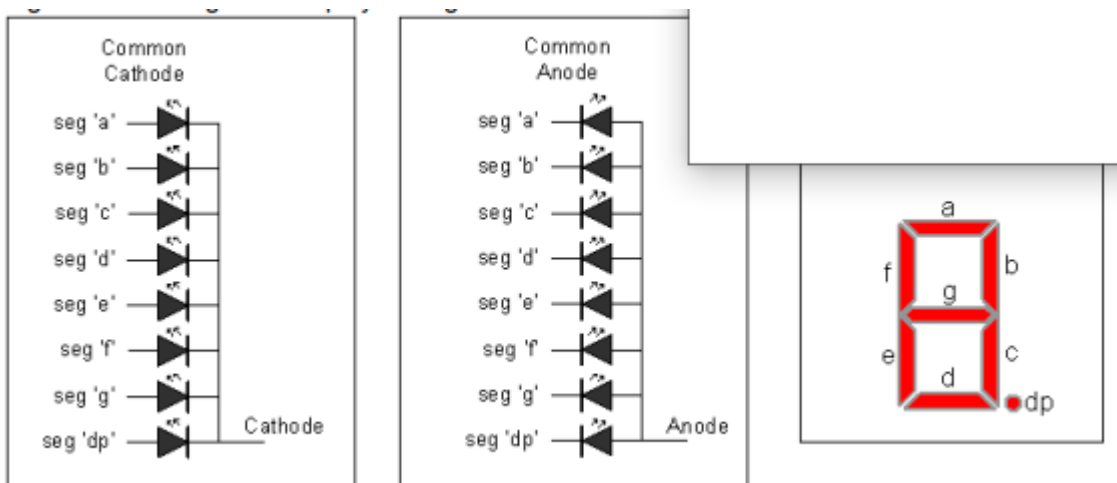


Figure 7 : configuration afficheur 7 segments

II.5.2.1 Caractères décimaux

Les chiffres de 0 à 9 sont les caractères les plus courants sur les afficheurs à sept segments.

Voici les modèles les plus couramment utilisés (figure 7) :



Figure 8 : les chiffres de 0 à 9

II.5.2.2 Caractères hexadécimaux

L'alphabet de A à F est représenté comme suit :



Figure 9 : l'alphabet de A à F. [12]

II.5.3 Programmation

Dans le cas d'un afficheur 7 segments piloté par 8 bits, la table de vérité donne (le segment G correspond au 7^{ème} bit, le segment A correspond au 1^{er} bit) :

Programmation

Affichage	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Hexadécimal
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

Figure 10 : la table de vérité de 8 bits. [12]

II.6 Les batteries

Un bloc-batterie, ou plus communément le bloc-batterie 1, est un groupe de batteries connectées ensemble pour produire un générateur de tension et de la capacité souhaitées. Ces batteries sont parfois appelées cellules ou cellules 2. Aussi connus sous le nom de piles, on entend les accumulateurs rechargeables utilisés dans les équipements électriques et électroniques domestiques. Les batteries peuvent stocker chimiquement de l'énergie électrique de manière contrôlée et la restituer sous forme de courant continu.

II.7 bouton poussoir

Un bouton (ou bouton et bouton 1) est un simple interrupteur qui contrôle la fonction d'un processus, Les boutons sont généralement fabriqués dans un matériau dur, généralement en plastique ou en métal, mais peuvent également être en caoutchouc. Il existe deux types de boutons poussoirs : les boutons poussoirs normalement ouverts et les boutons poussoirs normalement fermés (contact à ouverture (CR) ou contact normalement ouvert (CT))4.

Dans un bouton normalement ouvert, une connexion électrique est établie lorsque le bouton est enfoncé, tandis que dans un bouton normalement fermé, le circuit s'ouvre lorsque le bouton est enfoncé. Le mécanisme du bouton est principalement équipé d'un ressort ou d'un système qui permet un retour automatique à la position initiale.

II.8 résistance

La résistance est la propriété physique d'un matériau qui limite la circulation du courant dans un circuit.

Les composants ayant cette propriété sont utilisés pour limiter le passage des électrons dans un circuit

Formule : $R=U/I$. Son unité est l'ohm et l'équation est la suivante :

$$[R] = L^2 \cdot M \cdot T^{-3} \cdot I^{-2}$$

$$1 \Omega = 1 \text{ m}^2 \cdot \text{kg} \cdot \text{s}^{-3} \cdot \text{A}^{-2}$$

conclusion

Ce chapitre avait pour ambition d'étudier les éléments utilisés, il a fallu dans un premier temps définir chaque module, examiner ses caractéristiques et ne sélectionner que les plus pertinentes.

Ce travail minutieux, il convenait alors de s'intéresser à la réception de programme, une partie a été adopté dans cette étape, celui de rechercher et de confronter les différents problématiques.

Chapitre III : conception & programmation & simulation

III.1 conception

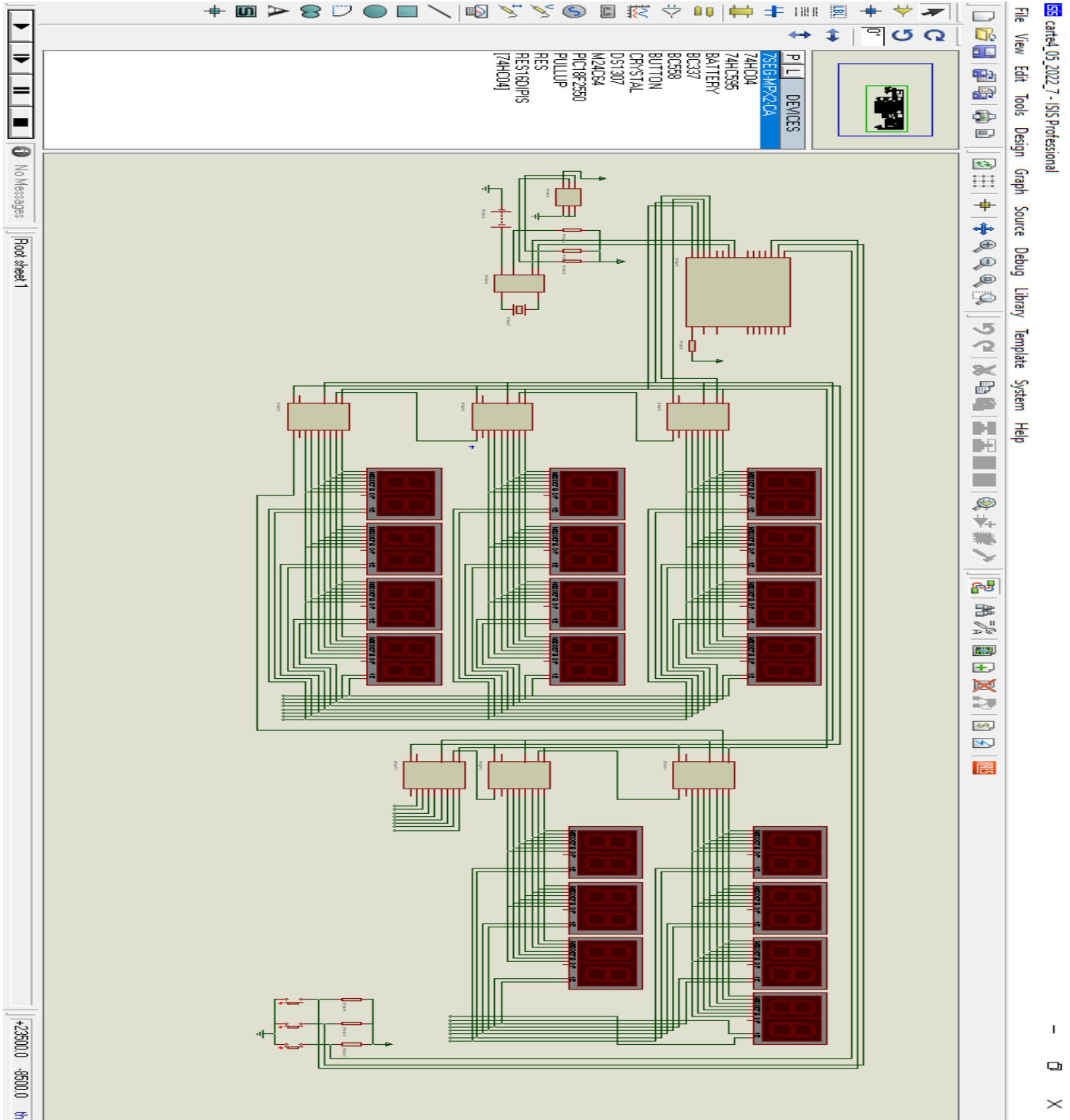


Figure 11 : Schéma électrique complet du notre projet

- **Commentaire :** nous avons un microcontrôleur, cinq registres à décalage , un ds1307, m24c64, 38 afficheurs 7 segments et des boutons poussoir pour régler l’heure et la date

III.2 programmations

```
//#include <built_in.h>
```

déclaration les pins pour simplifier la nomination

```
sbit ds_s at RB4_bit;
sbit oe_s at RB5_bit;
sbit st_s at RB6_bit;
sbit sh_s at RB7_bit;

sbit i2c_d at RB0_bit;
sbit i2c_h at RB1_bit;
```

```
void config()
```

configuration portb et la bibliothèque I2C

```
{
    trisb=0;
    portb=0;
    I2C1_Init(100000);
    CMCON = 0x07;
    ADCON1 = 0x0f;
    UART1_Init(9600);
    Delay_ms(1000);
}
```

```
void lecturedate()
```

lecture date et l'heure sur DS1307

```
{
    I2C1_Start();
    I2C1_Wr(0xD0);
    I2C1_Wr(0);
    I2C1_Repeated_Start();
    I2C1_Wr(0xD1);
    seco=I2C1_Rd(1);seco=bcd2dec(seco);
    minu=I2C1_Rd(1);minu=bcd2dec(minu);
    heur=I2C1_Rd(1);heur=bcd2dec(heur);
    jour=I2C1_Rd(1);
    jour=I2C1_Rd(1);jour=bcd2dec(jour);
    mois=I2C1_Rd(1);mois=bcd2dec(mois);
    anne=I2C1_Rd(0);anne=bcd2dec(anne);
    I2C1_Stop();
}
```



```
void serie(char n1)
{
    ds_s=n1.f7;sh_s=1;sh_s=0;
    ds_s=n1.f6;sh_s=1;sh_s=0;
    ds_s=n1.f5;sh_s=1;sh_s=0;
    ds_s=n1.f4;sh_s=1;sh_s=0;
    ds_s=n1.f3;sh_s=1;sh_s=0;
    ds_s=n1.f2;sh_s=1;sh_s=0;
    ds_s=n1.f1;sh_s=1;sh_s=0;
    ds_s=n1.f0;sh_s=1;sh_s=0;
}
```

l'affectation

```
void visible ()
{
    st_s=1;
    st_s=0;
    oe_s=0;
    delay_us(500); //2 200
    oe_s=1; //extinction
    delay_us(1000); //2 200
}
```

affectation de 0 ou 1 au registre à décalage 74HC595

```
void send1(char n1,char n2,char n3,char n4,char n5,char n6) //send1(255-1 ,heur ,jour ,h_ma ,h_do, h_fa);
{
    char tab[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};
    serie(n1);

    if ((tour.f3==0) && (n2.f7==1))
        {n2=10;}
    else
    {
        n2.f7=0;
        n2=n2/10;
    }
    serie(tab[n2]);

    if ((tour.f3==0) && (n3.f7==1))
        {n3=10;}
    else
    {
        n3.f7=0;
        if (n3==100) n3=10;
        else n3=n3/10;
    }
}
```

```
serie (tab[n3]);

n4=n4/10;serie (tab[n4]);
n5=n5/10;serie (tab[n5]);
n6=n6/10;serie (tab[n6]);
visible ();
}

void sende2 (char n1, char n2, char n3, char n4, char n5, char n6)
{
char tab[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};
serie (n1);

if ((tour.f3==0) && (n2.f7==1))
    {n2=10;}
    else
    {
    n2.f7=0;
    n2=n2%10;
    }
}
```

```
serie (tab[n2]);

if ((tour.f3==0) && (n3.f7==1))
    {n3=10;}
    else
    {
    n3.f7=0;
    if (n3==100) n3=10;
    else n3=n3%10;
    }
serie (tab[n3]);

n4=n4%10;serie (tab[n4]);
n5=n5%10;serie (tab[n5]);
n6=n6%10;serie (tab[n6]);
visible ();
}
```

```
void affichesalatG()
{
sendel(1 ,heur ,jour ,h_ma ,h_do, h_fa);
sende2(2 ,heur ,jour ,h_ma ,h_do, h_fa);

sendel(4 ,minu ,mois ,m_ma ,m_do, m_fa);
sende2(8 ,minu ,mois ,m_ma ,m_do, m_fa);

sendel(16 ,seco ,20 ,h_ic ,h_as, h_ch);
sende2(32 ,seco ,20 ,h_ic ,h_as, h_ch);

sendel(64 ,anne ,anne, m_ic, m_as, m_ch);
sende2(128,anne ,anne, m_ic, m_as, m_ch);
}
```

```
void affichesalatH()
{
sendel(1 ,heur ,jouH ,h_ma ,h_do, h_fa);
sende2(2 ,heur ,jouH ,h_ma ,h_do, h_fa);

sendel(4 ,minu ,moiH ,m_ma ,m_do, m_fa);
sende2(8 ,minu ,moiH ,m_ma ,m_do, m_fa);

sendel(16 ,seco ,14 ,h_ic ,h_as, h_ch);
sende2(32 ,seco ,14 ,h_ic ,h_as, h_ch);

sendel(64 ,anne ,annH, m_ic, m_as, m_ch);
sende2(128,anne ,annH, m_ic, m_as, m_ch);
}
```

```
void affichesalatN()
{
sendel(1 ,heur ,100 ,h_ma ,h_do, h_fa);
sende2(2 ,heur ,100 ,h_ma ,h_do, h_fa);

sendel(4 ,minu ,100 ,m_ma ,m_do, m_fa);
sende2(8 ,minu ,100 ,m_ma ,m_do, m_fa);

sendel(16 ,seco ,100 ,h_ic ,h_as, h_ch);
sende2(32 ,seco ,100 ,h_ic ,h_as, h_ch);

sendel(64 ,anne ,100 ,m_ic, m_as, m_ch);
sende2(128,anne ,100 ,m_ic, m_as, m_ch);
}
```

```
void read_24c64()
{
    int adresa;char addressH;char addressL;
    adresa=0x210+((mois-1)*32+jour-1)*16;
    addressH=adresa/256;
    addressL=adresa%256;
    I2C1_Start();
    I2C1_Wr(0xa0);
    I2C1_Wr(addressH);
    I2C1_Wr(addressL);
    I2C1_Repeated_Start();
    I2C1_Wr(0xa1);
    h_fa=I2C1_Rd(1); //j
    m_fa=I2C1_Rd(1); //m
    h_fa=I2C1_Rd(1); //ims
    m_fa=I2C1_Rd(1); //ims
    h_fa=I2C1_Rd(1);
    m_fa=I2C1_Rd(1);
    h_ch=I2C1_Rd(1);
    m_ch=I2C1_Rd(1);
    h_do=I2C1_Rd(1);
    m_do=I2C1_Rd(1);
    h_as=I2C1_Rd(1);
    m_as=I2C1_Rd(1);
    h_ma=I2C1_Rd(1);
    m_ma=I2C1_Rd(1);
    h_ic=I2C1_Rd(1);
    m_ic=I2C1_Rd(0);
    I2C1_Stop();
}
```

```
    m_as=I2C1_Rd(1);
    h_ma=I2C1_Rd(1);
    m_ma=I2C1_Rd(1);
    h_ic=I2C1_Rd(1);
    m_ic=I2C1_Rd(0);
    I2C1_Stop();
/*
    h_fa=4;
    m_fa=15;
    h_ch=7;
    m_ch=14;
    h_do=12;
    m_do=56;
    h_as=16;
    m_as=6;
    h_ma=19;
    m_ma=25;
    h_ic=20;
    m_ic=3;
*/
```

```
void calc_hcg()
{
char tab[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
char jj[]={0,0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf};
char hhR[]={0,0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0xff,0xff,0xff,0xff,0xff};
char hhL[]={0,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xfe,0xfd,0xfb,0xf7,0xef};
annH=anne;moiH=mois;
nj=jour-1;
while(mois>1)
{
    nj=nj+tab[mois-1];
    if ((mois==3) && (anne%4==0)) nj++;
    mois--;
}
while(anne>0)
{
    anne=anne-1;
    nj=nj+365;
    if (anne%4==0) nj=nj+1;
}
same=(nj-2)%7+1;
same=jj[same];
anne=annH;mois=moiH;
    nj=nj+off-3;
annH=nj/354.3333333;
nj=nj-annH*354.3333333;
moiH=nj/29.52;
jouH=nj-moiH*29.52;
```

```
jouH=24+jouH;
if (jouH>30) {jouH=jouH-30;moiH++; }
moiH=9+moiH;
if (moiH>12) {moiH=moiH-12;annH++;}
annH=annH+20; //heg[15][12]
harR=hhR[moiH];
harL=hhL[moiH];
}

////////////////////////////////////
void actu_heg()
{
read_24c64();
off=EEPROM_Read(0);
calc_heg();
}

////////////////////////////////////
void write_ds1307(char address,char w_data)
{
I2C1_Start();
I2C1_Wr(0xD0);
I2C1_Wr(address);
I2C1_Wr(w_data);
I2C1_Stop();
}
```

```
void ecriredate()
{
write_ds1307(0,0x00);
write_ds1307(1,dec2bcd(minu));
write_ds1307(2,dec2bcd(heur));
write_ds1307(4,dec2bcd(jour));
write_ds1307(5,dec2bcd(mois));
write_ds1307(6,dec2bcd(anne));
}
```

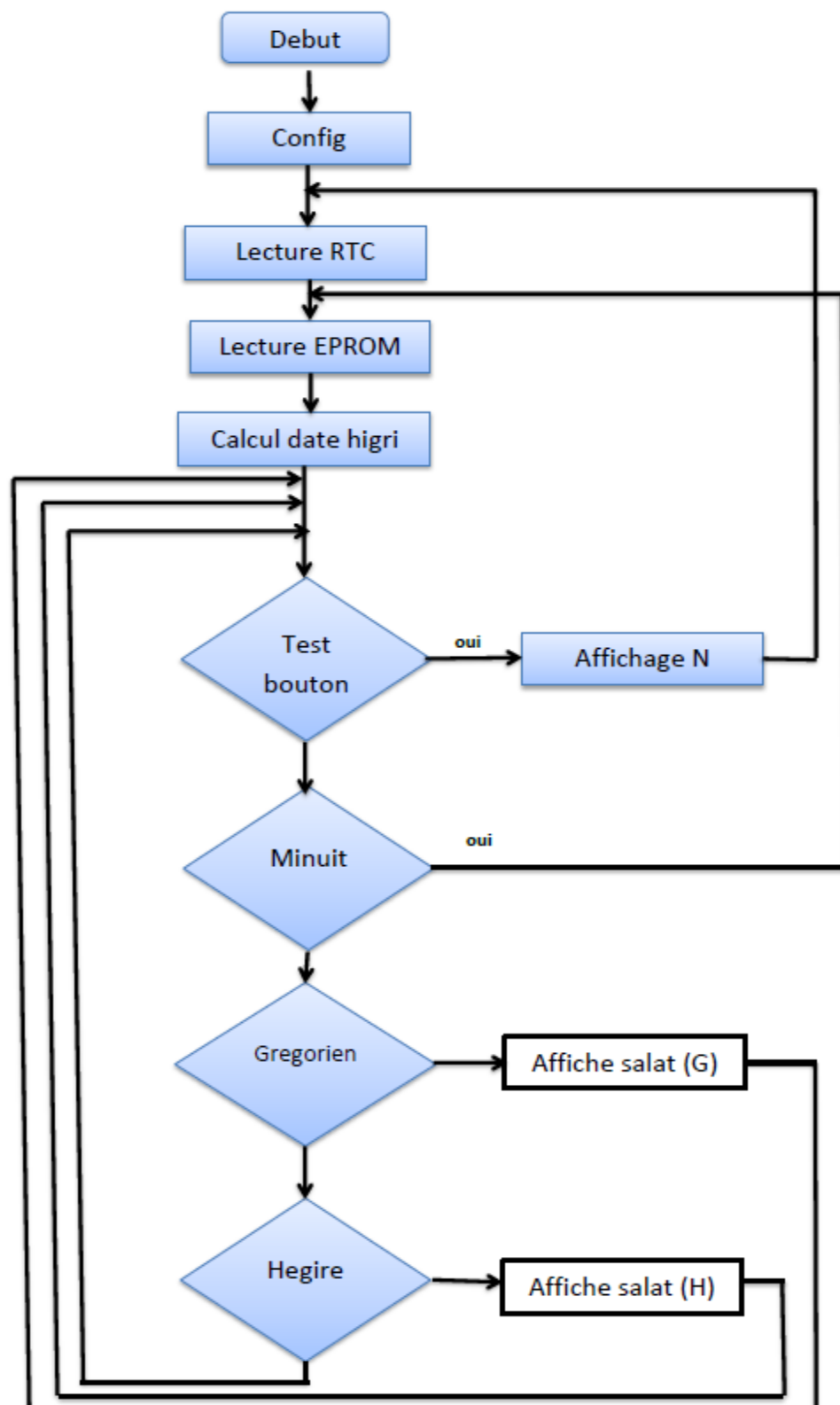
écriture du ds1307 les minutes , heure , jour ,mois ,
année (transformation décimal au binaire code)

```
void testbouton()
{
if (porta.f0==0)
{
tour=0;action=1;
bt++;
if (bt==7) {bt=0;}
jour.f7=0;
mois.f7=0;
anne.f7=0;
heur.f7=0;
minu.f7=0;
jouH.f7=0;
if (bt==1) jour.f7=1;
if (bt==2) mois.f7=1;
if (bt==3) anne.f7=1;
if (bt==4) heur.f7=1;
if (bt==5) minu.f7=1;
if (bt==6) jouH.f7=1;
if (bt==0)
{
EEPROM_Write(0,off);
ecriredate();
lecturedate();
actu_heg();
}
}
```

```
if (porta.f1==0)
{
    tour=0;action=1;
    if (bt==1) {jour++;if (jour==160) jour=129; }
    if (bt==2) {mois++;if (mois==141) mois=129; } //81
    if (bt==3) {anne++;if (anne==228) anne=128; }
    if (bt==4) {heur++;if (heur==152) heur=128; }
    if (bt==5) {minu++;if (minu==188) minu=128; } //80
    if (bt==6)
    {
        off++;if (off==7) off=6;
        calc_heg();
        jouH.f7=1;
    } //80
}
```

```
if (porta.f2==0)
{
    tour=0;action=1;
    if (bt==1) {jour--;if (jour==128) jour=159; }
    if (bt==2) {mois--;if (mois==128) mois=140; }
    if (bt==3) {anne--;if (anne==127) anne=227; }
    if (bt==4) {heur--;if (heur==127) heur=151; }
    if (bt==5) {minu--;if (minu==127) minu=187; }
    if (bt==6)
    {
        off--;if (off==255) off=0;
        calc_heg();
        jouH.f7=1;
    } //80
}
}
```


Organigramme principal



Programme principal, on appelle les fonctions précédentes pour simplifier l'écriture du programme

III .3 simulation

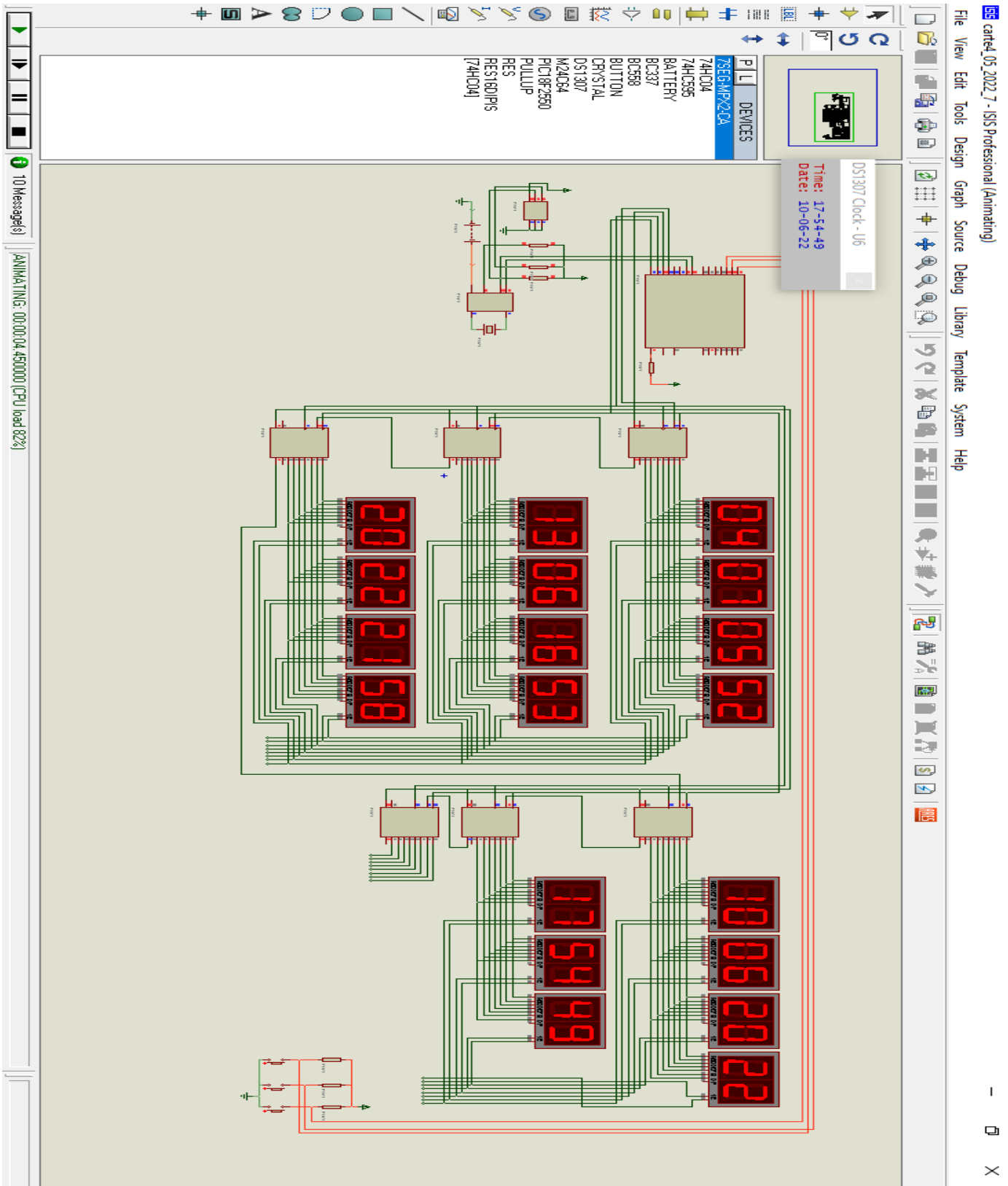


Figure 12 : simulation du circuit affichage de la date miladi et les horaires de prières

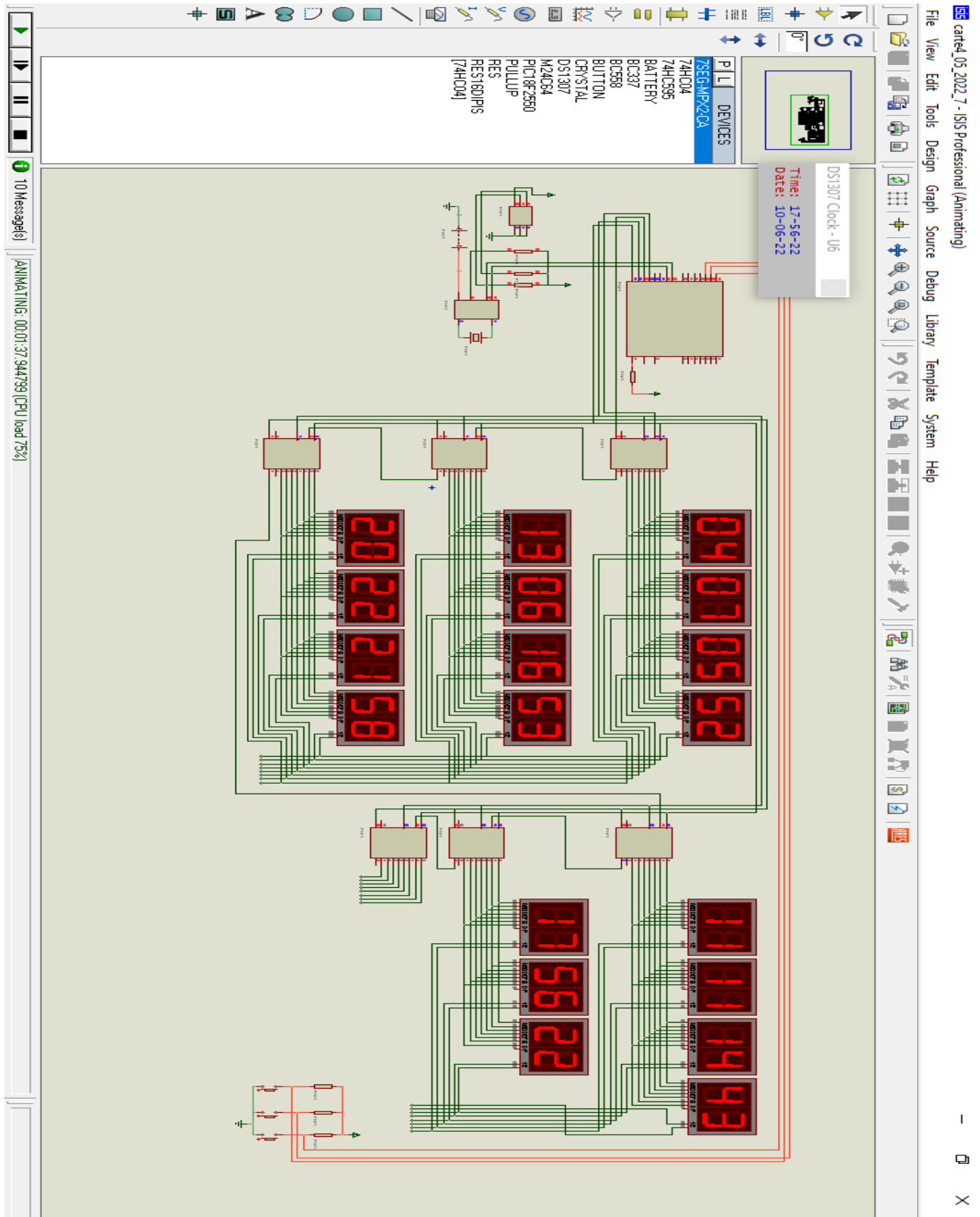


Figure 13 : simulation du circuit affichage de la date hijri et les horaires de prières

- **Sobh : 04 :07**
- **Chourouk : 05 :52**
- **Dohr : 13 :06**
- **Aser : 16 :53**
- **Maghrib : 20 :22**
- **Icha : 21 :58**
- **Date miladi : 10/06/2022**
- **Date hijri : 11/11/1443**
- **Heure : 17 :54 :49 et 17 :56 :22**

conclusion

après une programmation sur logiciel mikroc, on téléverse le programme sur le pic 18F2550 qui se trouve dans un fichier hex puis on simule sur logiciel isis donc les afficheurs s'allument sur les horaires de prières, la date et l'heure actuel.

Conclusion générale

Conclusion générale

Conclusion générale

Ce travail illustre le fonctionnement d'un système d'affichage d'horloge-calendrier en temps réel avec l'affichage des horaires de prière à base d'un microcontrôleur 18f2550 et d'autres modules . Les aspects de ce système sont :

- Affichage des horaires de prière (sobh,chourouk,dohr,aser,maghreb,ichaa)
- Affichage de la date actuelle (date hijri,date miladi)
- Affichage de l'heure.

Nous avons rencontré quelques obstacles dans ce travail, tels que l'existence d'un défaut de logiciel, les versions de logiciel ne sont pas compatibles, d'où la nécessité de les chercher ailleurs, le flux d'Internet faible au niveau des laboratoires, le langage de programmes que nous n'avions pas connus auparavant. Mais nous avons tiré parti de nouvelles informations et avons travaillé avec un programmes (mikroc). Nous nous sommes familiarisés avec les différents types de ce système.

Cependant, ce projet peut être amélioré, nous pouvons ajouter plus de paramètres à l'utilisateur pour lui donner plus de contrôle, et lui permettre aussi de contrôler les accès à partir des boutons poussoirs, ainsi de commander le système et le changer

Références Biographiques

[1] site internet <https://fr.wikipedia.org/wiki/Temps>

[2],[3],[4],[5] NIZAR TOUJENI

Site internet <https://www.technologuepro.com/cours-microcontrolleurs-mikroc/chapitre-3-compileur-mikroc-et-simulateur-isis.html>

[6] site internet <https://www.technologuepro.com/microcontrolleur/les-microcontrolleurs-pic.htm>

[7] site internet <https://www.robot-maker.com/forum/tutorials/article/117-registre-a-decalage-74hc595/>

[8] site internet <http://tiptopboards.com/136-registre-%C3%A0-d%C3%A9calage-8-bits-74hc595-x2.html>

9] site internet <https://www.robot-maker.com/forum/tutorials/article/117-registre-a-decalage-74hc595/>

[10] site internet <https://lesiteferroviaire.pagesperso-orange.fr/Module%20Horloge%20DS1307.htm#Ancre1>

[11] site internet <https://www.mouser.fr/ProductDetail/STMicroelectronics/M24C64-RMN6TP?qs=%252B9pW1iD4MXUPHPq0%252B2Ayg%3D%3D>

[12] « Seven Segment Displays » [archive du 4 avril 2012]

« Seven Segment Displays » [archive du 5 janvier 2012] (consulté le 14 novembre 2012)

ElektrotechnikTabellenKommunikationselektronik, Braunschweig, Germany, Westermann Verlag, 1999, 3rd éd. (ISBN 3142250379), p. 110

(de) Dieter Nährmann, WerkbuchElektronik, Achim, Bremen, Germany, Franzis-VerlagGmbH, 1981, 3e éd. (ISBN 3-7723-6543-4), p. 695

Résumé

L'affichage l'horloge-calendrier et les horaires de prière est un système compliqué qui utilise plusieurs modules .C'est un domaine de la programmation par ordinateur consistant à reconnaître automatiquement la date et l'heure actuel et aussi l'horaire de chaque prière à partir d'un microcontrôleur 18f2550. Il s'agit d'un sujet particulièrement étudié (isis proteus et mikroc),avec de très nombreuses publications et brevets, et des conférence spécialisées.

Ce système utilise plusieurs composants tels que le registre à décalage le 74HC595 qui sert à donner 8 sorties pour 4 entrées, il facilite vraiment la tâche et rend le circuit plus simple , et on a aussi trois autres composants, le premier est le DS1307 qui est responsable de l'horloge et le calendrier en temps réel RTC (real time clock), puis le deuxième qui est le M24C64 qui sert de stocker les horaires exacts des prières, et les afficheurs à sept segments qui affichent le résultat. Le fonctionnement de se système se base sur ces composants.

Après des études qui ont été détaillé dans notre mémoire, on obtient un circuit et un programme final. Reste juste de compiler le programme sur le pic 18F2550 et le simuler sur isis et puis voir le résultat.

Il est également utilisé pour faciliter la vie de l'utilisateur, on trouve généralement ce système dans des applications sur mobile, mosquées.

Mots clés BCD,EPROM ;I2C;Pwm;PIC;CMOS;LCD ;RAM ;CPU

ملخص

يعد عرض التقويم على مدار الساعة وأوقات الصلاة نظامًا معقدًا يستخدم عدة وحدات ، فهو مجال برمجة كمبيوتر يتكون من التعرف التلقائي على التاريخ والوقت الحاليين وأيضًا وقت كل صلاة من متحكم 18F2550. هذا موضوع تمت دراسته بشكل خاص في البرمجيات (ISIS PROTEUR ET MIKROC) ، مع العديد من المنشورات وبراءات الاختراع ، والمؤتمرات المتخصصة.

يستخدم هذا النظام عدة مكونات مثل سجل التحول 74HC595 الذي يستخدم لإعطاء 8 مخرجات لـ 4 مدخلات ، إنه يسهل المهمة حقًا ويجعل الناظر أبسط ، ولدينا أيضًا ثلاثة مكونات أخرى ، الأول هو DS1307 وهو واحد هو المسؤول عن إعطاء الساعة والتقويم في الوقت الحقيقي RTC (ساعة الوقت الحقيقي) ، ثم الثاني وهو M24C64 الذي يستخدم لإعطاء أوقات الصلاة بالضبط ، والثالث هو العرض المكون من سبعة أجزاء الذي يعرض النتيجة. يعتمد تشغيل هذا النظام على هذه المكونات. بعد الدراسات التي تم تفصيلها في مذكرتنا ، نحصل على دائرة وبرنامج نهائي ، يبقى فقط تجميع البرنامج على المتحكم 18F2550 ومحاكته على ISIS ثم يتم عرض النتيجة. يستخدم لتسهيل الحياة على المستخدم ، وهذا النظام موجود بشكل عام في تطبيقات الهاتف المحمول وعلى شكل لوحات في المساجد.

الكلمات المفتاحية: BCD,EPROM ;I2C;Pwm;PIC;CMOS;LCD ;RAM ;CPU

Abstract

Displaying the clock-calendar and prayer times is a complicated system that uses several modules, it is a field of computer programming consisting of automatically recognizing the current date and time and also the time of each prayer from an 18f2550 microcontroller. This is a subject particularly studied in software (isis proteus and mikroc), with numerous publications and patents, and specialized conferences.

This system uses several components such as the 74HC595 shift register which is used to give 8 outputs for 4 inputs, it really facilitates the task and makes the circuit simpler, and we also have three other components, the first is the DS1307 which is the one responsible for giving the clock and the calendar in real time RTC (real time clock), then the second which is the M24C64 which is used to give the exact prayer times, and the third the seven-segment display which displays the result . The operation of this system is based on these components.

After studies which are detailed in our memory, we obtain a circuit and a final program, it remains just to compile the program on the peak 18F2550 and simulate it on isis and then the result is displayed.

It is also used to make life easier for the user, this system is generally found in mobile applications, and on a plate in mosques.

Keywords: BCD,EPROM ; I2C; Pwm; PIC; CMOS; LCD ; RAM ; CPU