

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

« وَقُلْ رَبِّ زِدْنِي عِلْمًا »

- طه 114 -

# Remerciements

C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à l'égard de tous ceux qui nous ont aidés à la réalisation de notre projet de fin d'études.

Nous tenons en premier lieu à exprimer notre gratitude envers notre encadreur **Mr NEMICHE AHMED** qui nous a permis de mener à terme ce travail, par ses soutiens, ses précieux conseils et ses bienveillances.

Sa disponibilité et sa qualité humaine sont autant d'éléments qui ont favorisé le développement de ce mémoire.

Que les membres de jury **Mr BELAID ALI** et **Mr HACHIM ABDALLAH** et **Mr BENALLAL MOHAMMED** trouvent nos profondes gratitudees pour l'honneur qu'ils nous font en examinant notre exposé.

Nos pensées vont également à tous les membres de nos familles, leurs encouragements, leurs soutiens qui nous ont permis de franchir toutes les difficultés envisagées.

# Didicace: Didicace:

*Au non 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 de mes études.*

*Je dédie ce modeste travail à :*

*A mon père Mohamed que Dieu bénisse son âme.*

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

*Mes frères : Miloud, Aissa , Djeloule , Youcef , Hossine.*

*Mes sœurs : Amaria , Fatiha , Fatima .*

*A toutes les familles : MAHI.*

*A toutes les familles : ABESS.*

*A mon binôme, notre encadreur et tous mes amis.*

*A ma camarade et particulièrement : **FATIHA BOUCHA.***

*En fin à moi-même et toute la promotion master d'instrumentation électronique **2014.***

**MAHI KHADIDJA.**

# Didicace:

*Au non 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 Ahmed**

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

**Mes frères : Yousef, Rachid, Ibrahim.**

**Mes sœurs : Fatima, Saliha.**

**A mon marié : OMAR**

**A toutes les familles : GUITOUN.**

**A toutes les familles : HOCINI.**

**A mon binôme, notre encadreur et tous mes amis.**

**A ma camarade et particulièrement : KARIMA HAFS.**

**En fin à moi-même et toute la promotion master d'instrumentation électronique 2014.**

**GUITOUN KHADIDJA.**

# Sommaire

## Chapitre I:

**I / Objectif .....8**

**II/ Introduction Générale .....9**

## Chapitre II:

**II.1/ Les microcontrôleurs .....11**

II.1.1/ Définition d'un microcontrôleur .....11

II.1.2/ Les avantages du microcontrôleur .....11

II.1.3/ Contenu d'un microcontrôleur .....11

**II.2 / Les PICS .....12**

II.2.1/ Définition d'un PIC .....12

II.2.2/ Les différentes familles de PIC .....12

- Structure d'un PIC .....12
- Structure minimale d'un PIC .....12
- Caractéristiques générales.....13
- Schéma fonctionnel.....14
- Identification d'un PIC .....15

## Chapitre III:

**III/ Afficheur 7 SEGMENTS .....17**

III.1/ Définition .....17

III.2/ Des LED, encore des LED .....19

III.3/ Cathode commune ou Anode commune .....19

III.4 / Choix de l'afficheur .....19

III.5 /Présentation du boîtier .....20

## Chapitre IV:

**IV.1/Circuit décalage 74HC164 .....22**

IV.1.1/ Définition .....22

IV.1.2/ Diagramme fonctionnel .....22

**IV.2/Horloge temps réel DS1302 .....23**

IV.2.1 /Définitions .....23

IV.2.2 / Caractéristiques .....23

IV.2.3 / Structure interne .....23

IV.2.4 / Fonctionnement.....24

IV.2.5 / Application.....24

IV.2.6/ Branchement d'horloge .....24

## Chapitre V:

<b>V.1/ Etude Pratique.....</b>	<b>27</b>
V .1.1/schéma synoptique .....	27
V .1.2/ Branchement d'horloge temps réel : <b>DS1302</b> .....	28
• Fiche technique DS1302.....	28
V.1. 3/ Simulation de circuit par ISIS .....	29
V .1.4/ création du circuit imprimé .....	30
V.1.5/ Le circuit en 3D .....	31
V .1.6/ Carte principale coté composants.....	32
<b>V.2/L'organigramme.....</b>	<b>33</b>

## Chapitre VI:

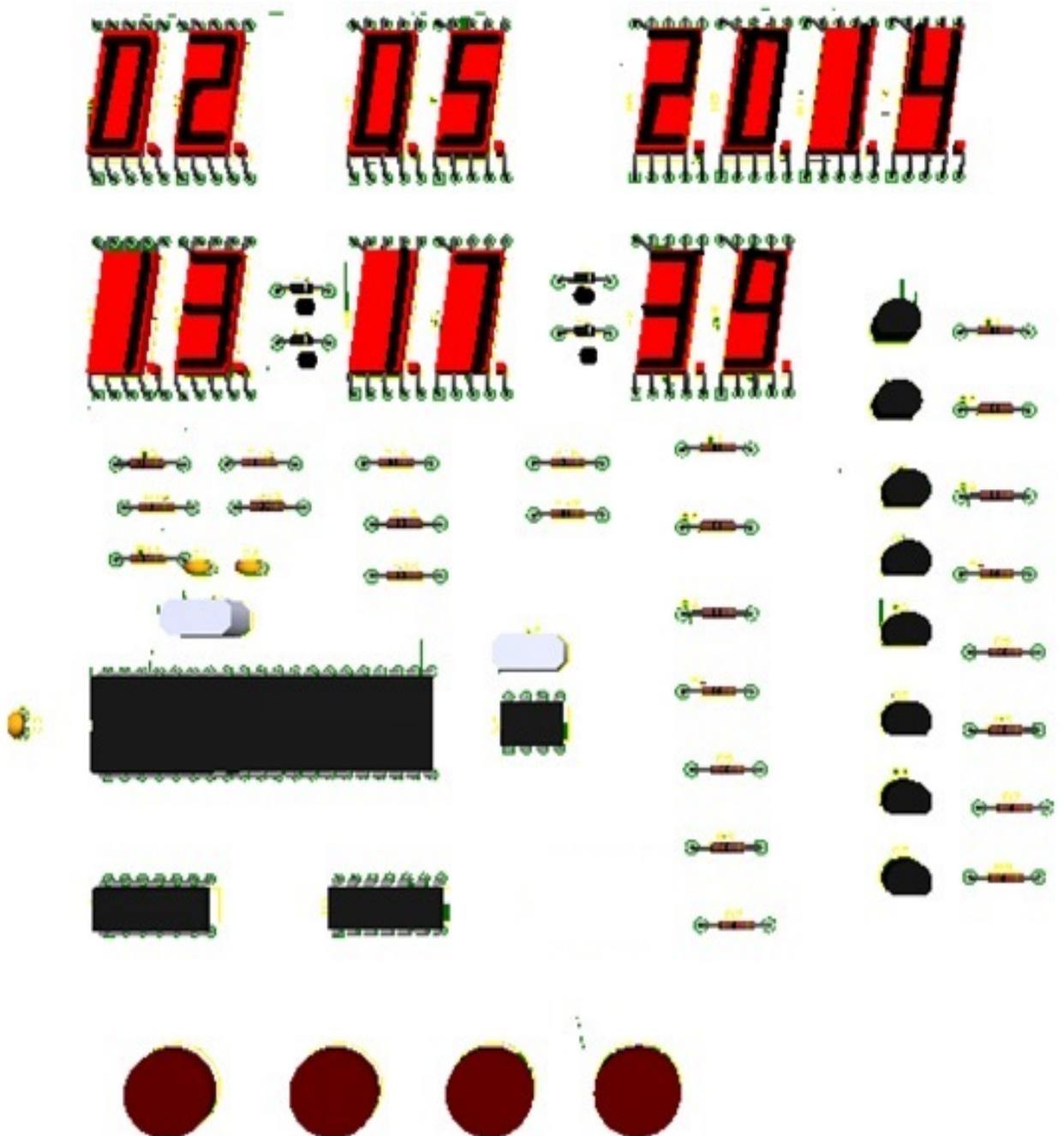
<b>VI.1 / Conclusion générale .....</b>	<b>38</b>
<b>VI.2 / Bibliographie.....</b>	<b>39</b>
<b>VI.3 / Annexes.....</b>	<b>40</b>
❖ Liste des composants .....	40

# CHAPITRE I

## 1 / OBJECTIF :

Notre travail consiste à réaliser une horloge temps réel qui permet d'afficher :

- ✚ date (jour, mois, année) .
- ✚ horaire (heure, minute, seconde) .
- ✚ réglage de ces valeurs par 4 boutons poussoirs.
- ✚ utilisation du  $\mu\text{C}$  18F458 et DS1302 (sauvegarde par pile).



## **II / 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'une des deux configurations suivantes :

- horloge externe RTC (\*) de type PCF8583

- horloge externe RTC (\*) de type DS1302

(\*) RTC = **R**ea**I** **T**ime **C**lock en anglais, horloge temps réel en français

Le dialogue avec le PCF8583 ou DS1302 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. Les horloges externes PCF8583 et DS1302 présentent l'avantage de tourner moins vite (oscillateur 32 KHz) et consomment bien moins d'énergie que le PIC. La sauvegarde en énergie est pour cette raison bien plus aisée à implémenter. En cas de coupure secteur, seul l'horloge externe (PCF8583 ou DS1302) 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 DS1302.

Nous allons travailler selon une méthode de travail bien déterminée :

- 1- Etude théorique des différents systèmes qui composent notre enregistreur de données.
- 2- Conception et simulation du système.
- 3- Réalisation pratique et diagnostique.

*Ce rapport reflète et décrit le travail effectué.*

# CHAPITRE II

## II.1- Les microcontrôleurs :

### II.1.1/ 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.

### II.1.2/ 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.

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 :
  - 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.

### II.1.3. 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 suivante :

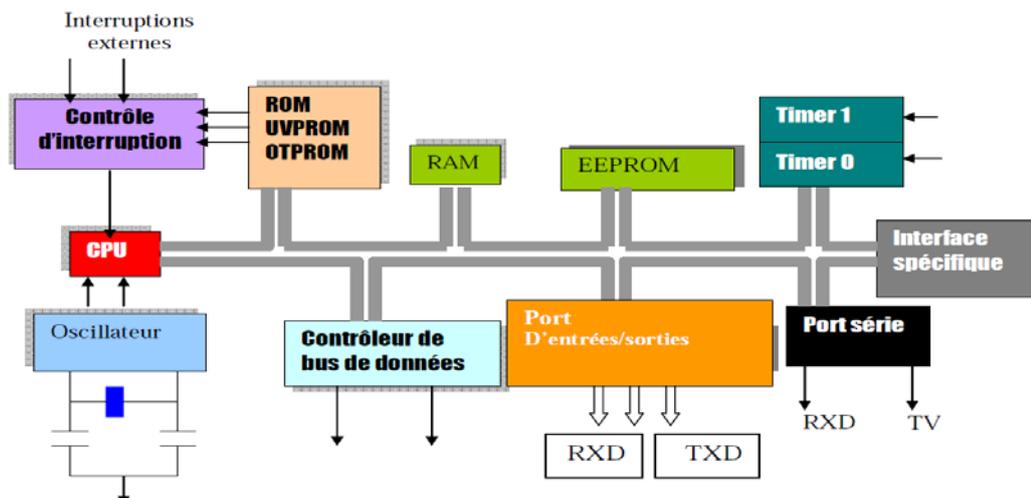


Figure 1 : Structure interne d'un microcontrôleur.

## II.2 / Les Pics :

### II.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.

### II.2 .2/ Les différentes familles de PIC :

Les PICs sont subdivisés en 3 grandes familles :

- La famille Base-Line, qui utilise des mots d'instructions de 12 bits.
- la famille Mid-Range, qui utilise des mots de 14 bits (et dont font partie les 16F84 et 16F876)
- 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.

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 :

- 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 :

- 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.

- **Caractéristiques générales:**

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	Comparators	CCP/ ECCP (PWM)	MSSP		USART	Timers 8/16-bit
	FLASH (bytes)	# Single Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I <sup>2</sup> C		
PIC18F458	32K	16384	1536	256	33	8	2	1/1	Y	Y	Y	1/3

**Tableau :** *Caractéristiques du PIC 18F458.*

- Schéma fonctionnel:

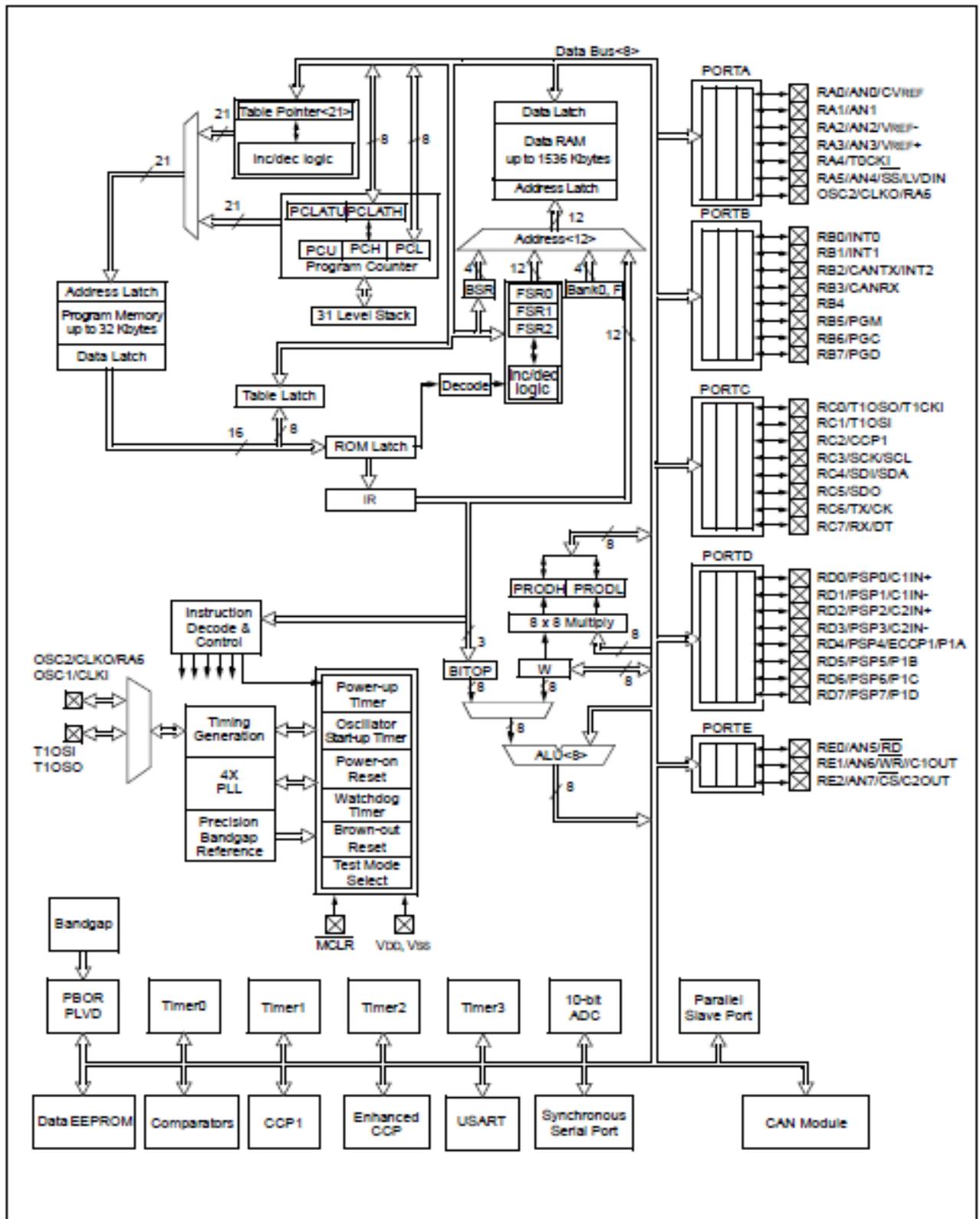
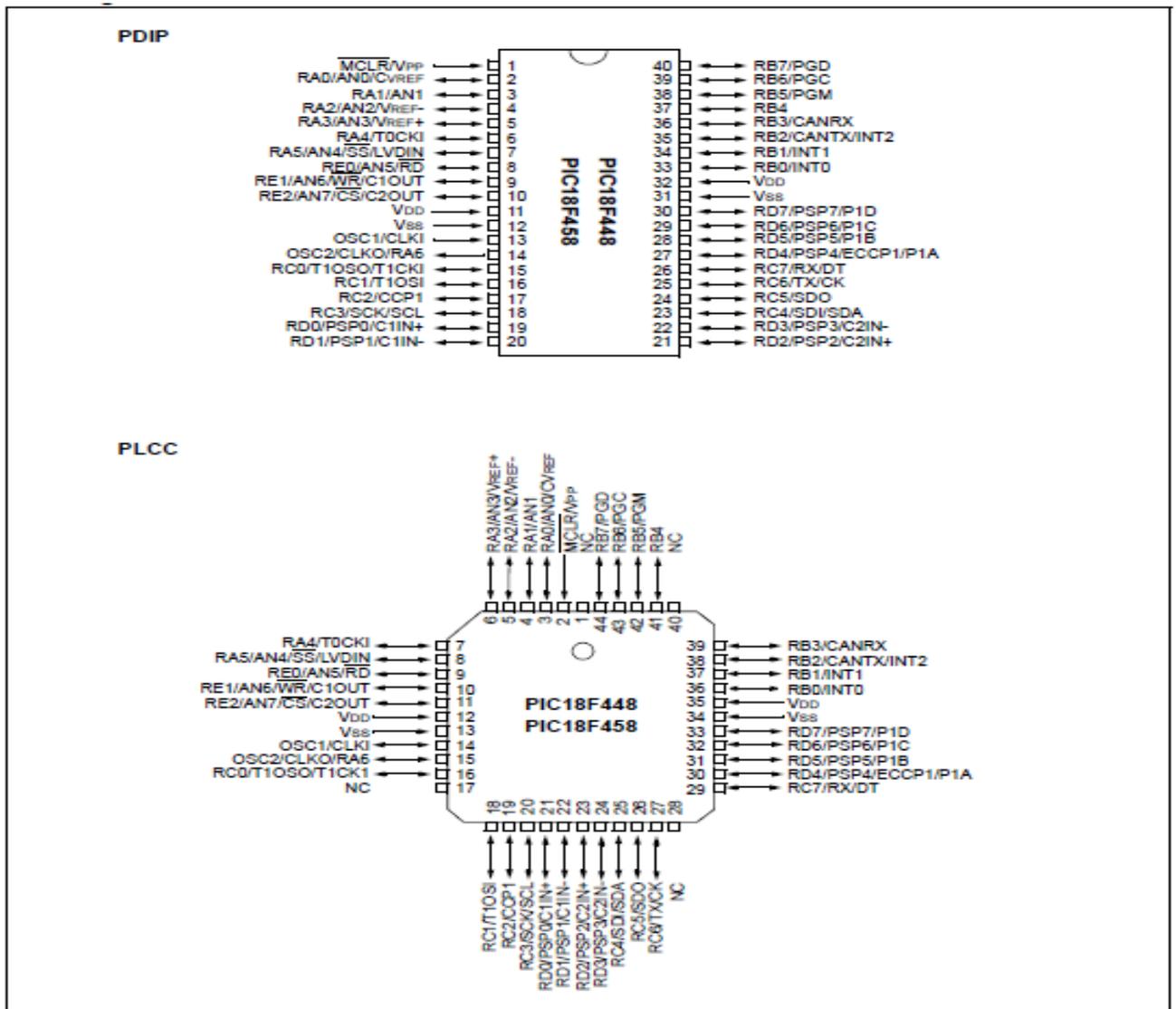


Figure 2 : Schéma fonctionnel d'un PIC



- **Identification d'un PIC :**

Pour identifier un PIC, nous utilisons simplement son numéro : les 2 premiers chiffres indiquent la catégorie du PIC, **16** indique un PIC Mid-Range, **18** indique PIC High-Range.

Vient ensuite parfois une lettre **L** : Celle-ci indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante. Ensuite, nous trouvons : **C** indique que la mémoire programme est une EPROM ou plus rarement une EEPROM, **CR** pour indiquer une mémoire de type ROM ou **F** pour indiquer une mémoire de type *FLASH*.

À ce niveau, on rappelle que seule une mémoire FLASH ou EEPROM est susceptible d'être effacé.

Finalement nous trouvons sur les boîtiers le suffixe « -XX » dans lequel XX représente la fréquence d'horloge maximale que le PIC peut recevoir. Par exemple -04 pour un 4MHz.

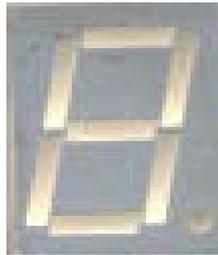
Notons dès à présent que les PICs sont des composants STATIQUES, c'est à dire que la fréquence d'horloge peut être abaissée jusqu'à l'arrêt complet sans perte de données et sans dysfonctionnement.

# CHAPITRE III

## III/ Afficheur 7 SEGMENTS:

### III.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 boîtier. 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). Regardez à quoi ça ressemble :



Afficheur à 7 segments

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

Programmation :

**Affichage Entrée 1 Entrée 2 Entrée 3 Entrée 4**

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

En notant les entrées 1, 2, 3, 4 du tableau ci-dessus respectivement  $i_1$ ,  $i_2$ ,  $i_3$  et  $i_4$ , les équations des segments (pour afficher les nombres de 0 à F) sont :

- $a = (\text{not } (i_1) \text{ and } i_3) \text{ or } (i_1 \text{ and not } (i_4)) \text{ or } (i_2 \text{ and } i_3) \text{ or not}(i_2 \text{ or } i_4) \text{ or } (i_1 \text{ and not}(i_2) \text{ and not}(i_3)) \text{ or } (\text{not}(i_1) \text{ and } i_2 \text{ and } i_4)$
- $b = \text{not } (i_1 \text{ or } i_2) \text{ or not } (i_2 \text{ or } i_3) \text{ or not}(i_2 \text{ or } i_4) \text{ or } (\text{not}(i_1) \text{ and not}(i_3 \text{ xor } i_4)) \text{ or } (i_1 \text{ and not}(i_3) \text{ and } i_4)$
- $c = (i_1 \text{ xor } i_2) \text{ or } (\text{not } (i_3) \text{ and } i_4) \text{ or } (\text{not } (i_3 \text{ xor } i_4) \text{ and not}(i_2))$
- $d = (i_1 \text{ and not } (i_3)) \text{ or not}(i_1 \text{ or } i_2 \text{ or } i_4) \text{ or } (i_2 \text{ and } (i_3 \text{ xor } i_4)) \text{ or } (\text{not}(i_2) \text{ and } i_3 \text{ and } i_4)$
- $e = \text{not } (i_2 \text{ or } i_4) \text{ or } (i_3 \text{ and not}(i_4)) \text{ or } (i_1 \text{ and } i_2) \text{ or } (i_1 \text{ and } i_3)$
- $f = (i_1 \text{ and not } (i_2)) \text{ or not } (i_3 \text{ or } i_4) \text{ or } (\text{not } (i_3) \text{ and } (i_1 \text{ xor } i_2)) \text{ or } (i_1 \text{ and } i_3) \text{ or } (i_2 \text{ and not}(i_4))$
- $g = (i_3 \text{ and } (i_1 \text{ or not } (i_2) \text{ or not } (i_4))) \text{ or } (i_1 \text{ and } i_4) \text{ or } (\text{not}(i_3) \text{ and } (i_1 \text{ xor } i_2))$

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

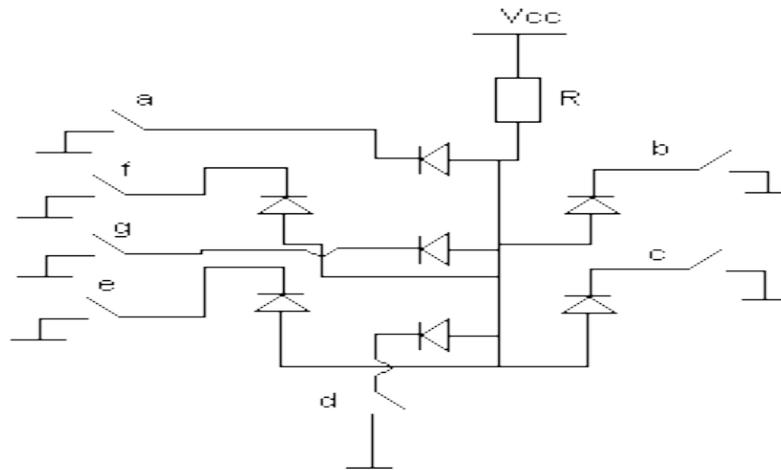
Dans le cas d'un afficheur 7 segments commandé par 8 bits, la table de vérité donne (segment G correspondant à bit 7 et A à bit 1) :

### 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

### III.2/Des LED, encore des 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 :



***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).

### III.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.

### III. 4/Choix de l'afficheur :

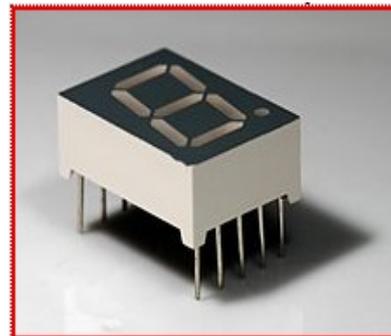
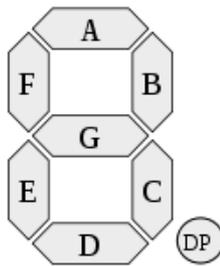
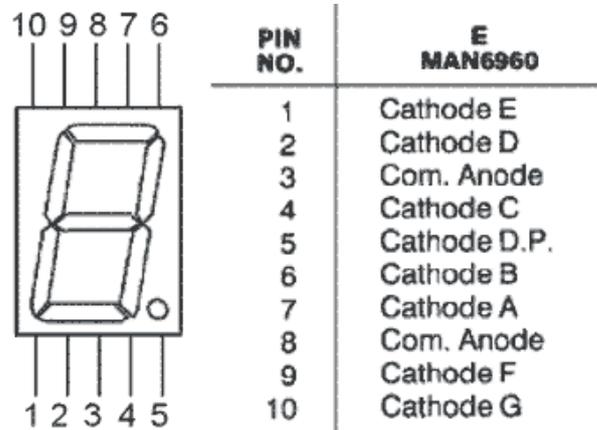
Pour la rédaction nous avons fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin

. Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment on mettra la broche de signal à 0 et on l'éteindra en mettant le signal à 1. On a toujours fait comme ça depuis le début, ça ne vous posera donc aucun problème.

Nous allons maintenant voir comment brancher l'afficheur à anode commune.

### III.5 /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 boîtier). Voici une représentation de ce dernier (à gauche) :



Voici la signification des différentes broches :

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.

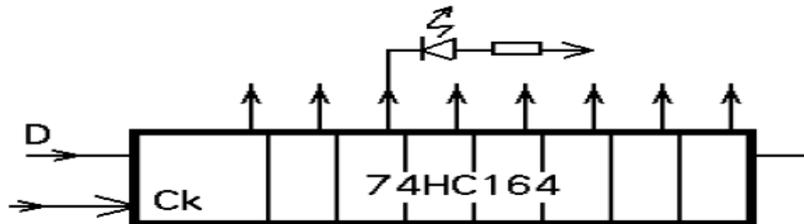
Pour allumer un segment c'est très simple, il suffit de le relier à la masse.

# CHAPITRE IV

## IV .1/Circuit décalage 74HC164 :

### IV.1 .1/ Définition:

Un registre à décalage est un circuit électronique formé de bascules (flip-flops) en cascade, avec un signal dit d'horloge (clock) qui pousse l'information d'une bascule à la suivante :



Il existe des circuits registres avec 8 ou 16 sorties simples ou amplifiées, ou avec 8 entrées. On peut les cascader. L'intérêt est que quelque soit la longueur, il suffit de 2 ou 3 broches sur le processeur pour en avoir beaucoup plus pour commander des LEDs ou lire des pushsoirs.

A noter qu'avec le circuit 74HC164 pendant que l'information se décale, les sorties changent et un registre parallèle doit être ajouté sur les sorties si cela est gênant.

### IV.1.2 /Diagramme fonctionnel :

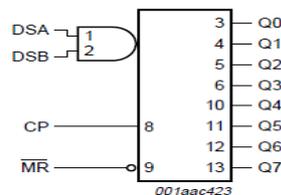


Fig 1. Logic symbol

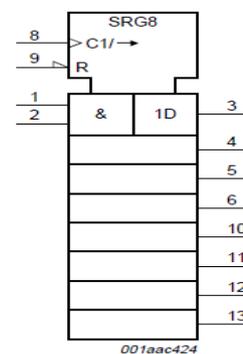


Fig 2. IEC logic symbol

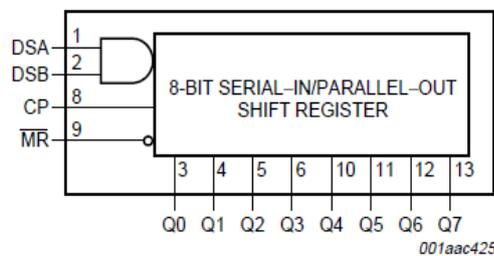


Fig 3. Logic diagram

## IV .2/Horloge temps réel DS1302 :

### IV.2.1 /Définitions :

#### Qu'est-ce qu'une horloge temps réel ?

[ RTC = **R**eal **T**ime **C**lock = Horloge temps réel ]

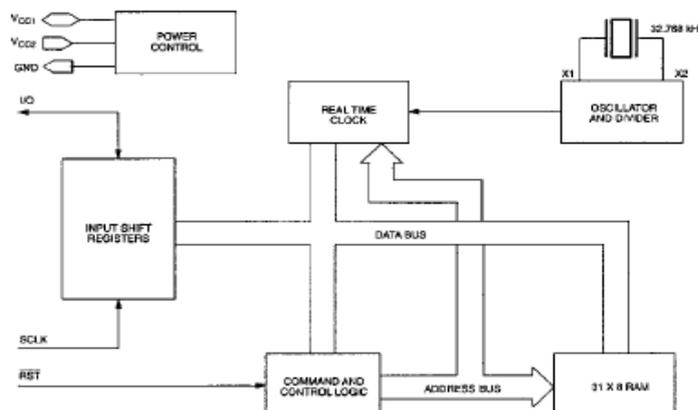
Une horloge temps réel est un composant de type "oscillateur / base de temps" qui fonctionne de façon autonome, avec un quartz pour une bonne précision de la fréquence d'oscillation. Ce type de composant équipe entre autres les cartes mère des ordinateurs, et permet de disposer à tout instant (et sur requête) des informations Date et Heure. Si l'alimentation du composant est assurée lors des coupures secteur (avec une pile, un accu ou un condensateur de forte valeur), les informations sont conservées et continuent d'être régulièrement mises à jour. C'est grâce à ce genre de composant que votre PC conserve la bonne date et la bonne heure même si vous débranchez son cordon secteur pendant plusieurs jours... à condition bien sûr que la pile de sauvegarde (en général une CR2032) soit en bon état. Ce type de composant peut être utilisé ailleurs que dans un ordinateur, en fait dans tout système où la date et/ou l'heure sont utilisées. Par exemple dans un enregistreur d'événements avec horodatage, ou encore dans un programmeur journalier.

**L'horloge temps réel PCF8583 ou DS1302** dont il est question ici dispose d'une interface de type I2C, et le dialogue avec elle s'effectue grâce à un microcontrôleur utilisé en maître.

### IV.2.2 / Caractéristiques :

- Circuit d'horloge en temps réel donnant la seconde, minute, heure, jour du mois, mois, jour de la semaine, et année (avec compensation des années bissextiles jusqu'en 2100).
- RAM statique de 31 octets de large avec sauvegarde par pile.
- Réduction au strict indispensable du nombre des lignes d'E/S par transmission sérielle
- Fonctionne à une tension d'alimentation comprise entre 2,0 et 5,5 V
- Consommation de courant faible, inférieure à 200 nA sous 2,0 V
- Mode salve (*burst*) pour la lecture/écriture de n'adresses dans la – RAM de l'horloge
- Boîtier à 8 broches
- Interface 3 fils
- Compatible TTL

### IV.2.3 / Structure interne :

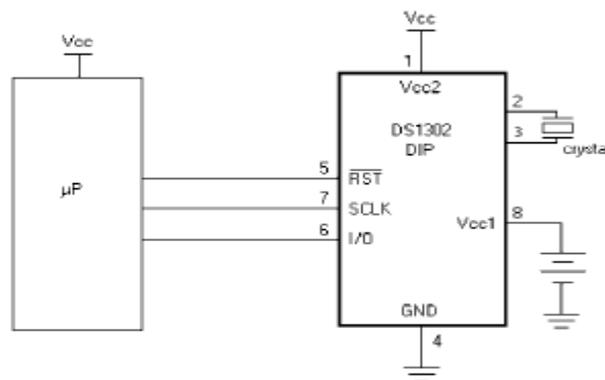


#### IV.2.4 / Fonctionnement :

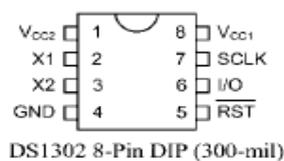
Le DS1302, conçu à l'origine pour le suivi chronologique de la fonction de charge d'appareils alimentés par accus, intègre une horloge en temps réel avec fonction de calendrier et une RAM statique de 31 octets. Il communique avec un microcontrôleur par le biais d'une interface sérielle dépouillée. La transmission des données se fait par le biais de 3 lignes seulement : RAZ (*Reset*), horloge sérielle (SCL = *Serial Clock*) et E/S (*I/O*). Il est possible de Faire transiter les données dans les 2 sens, soit octet par octet soit en mode salve qui permet d'envoyer des groupes de 32 octets maximum. L'horloge en temps réel/calendrier égrène les secondes, minutes, heures, jours du mois, mois, jours de la semaine et année. L'horloge travaille en mode 24 ou 12 heures avec mention AM/PM. En fonctionnement, le DS1302 ne requiert que très peu d'énergie et consomme moins de 1  $\mu$ W pour garder à jour les informations de date et d'heure. Le DS1302 remplace le standard de l'industrie qu'était devenu son prédécesseur, le DS1202, mais contrairement à ce dernier, utilise 2 broches pour son alimentation sachant qu'il utilise celle dont le niveau de tension (VCC ou tension de pile) est le plus élevé. Cela permet d'avoir une recharge d'entretien de l'accu de sauvegarde...

Le DS1302 possède en outre quelques octets de mémoire de brouillon (*scratchpad*).

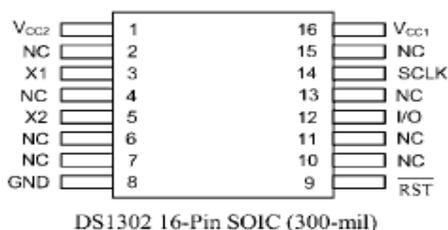
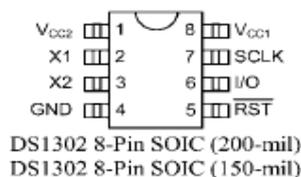
#### IV.2.5 / Application :



#### IV.2.6 / Branchement d'horloge :

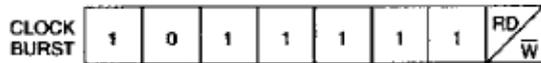
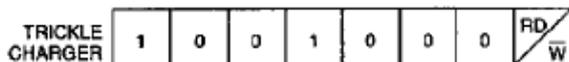
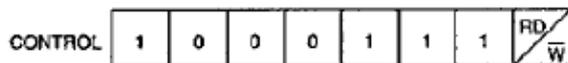
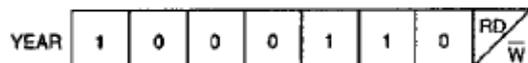
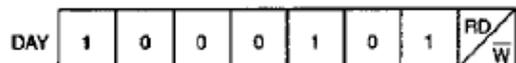
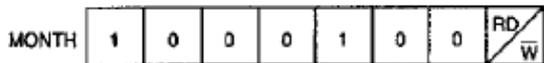
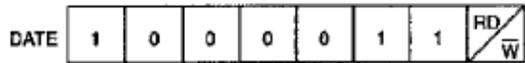
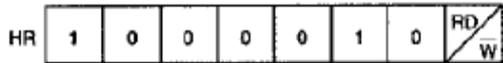
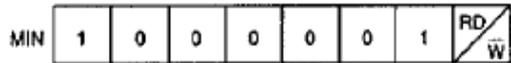
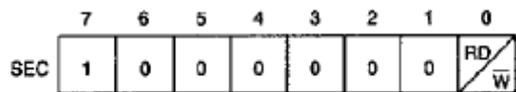


Broche	Fonction
X1, X2	Connexion du quartz 32,768 kHz
GND	Masse
RST	Remise à zéro
I/O	Entrée/Sortie des données
SCLK	Horloge sérielle
V <sub>CC1</sub> , V <sub>CC2</sub>	Alimentation

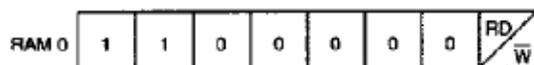


REGISTER ADDRESS

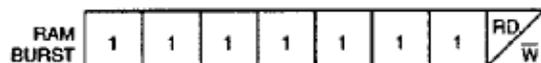
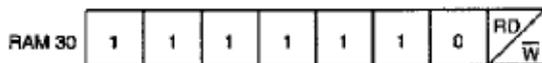
A. CLOCK



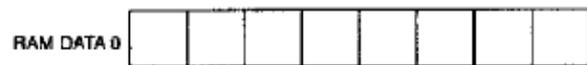
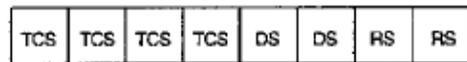
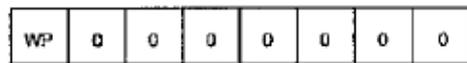
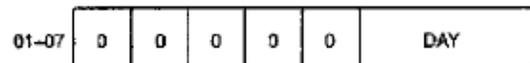
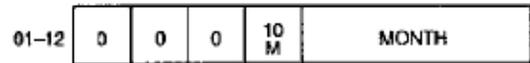
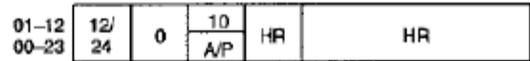
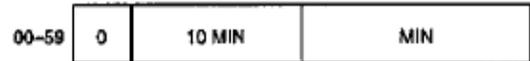
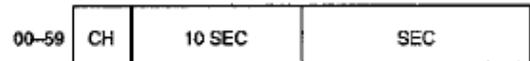
B. RAM



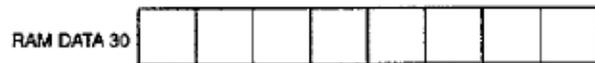
⋮



REGISTER DEFINITION



⋮

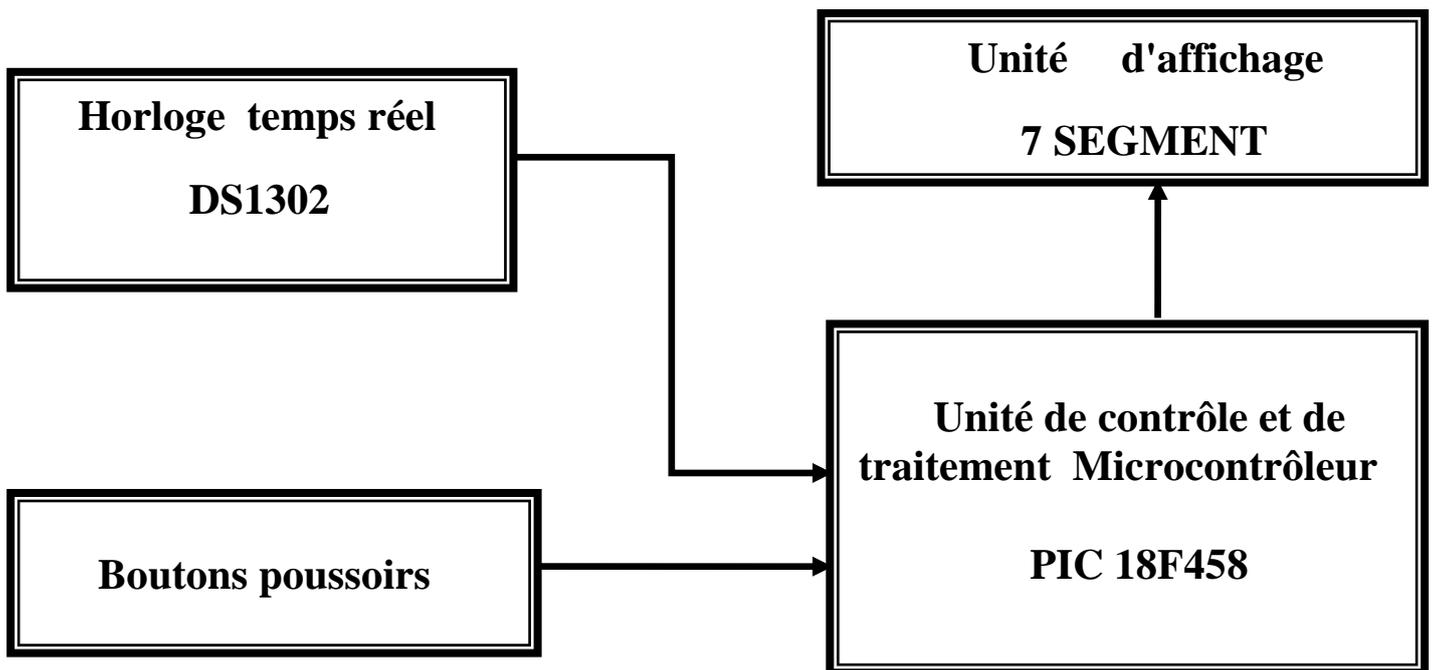


# CHAPITRE V

## V .1/ Etude pratique:

### V.1.1/schéma synoptique :

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

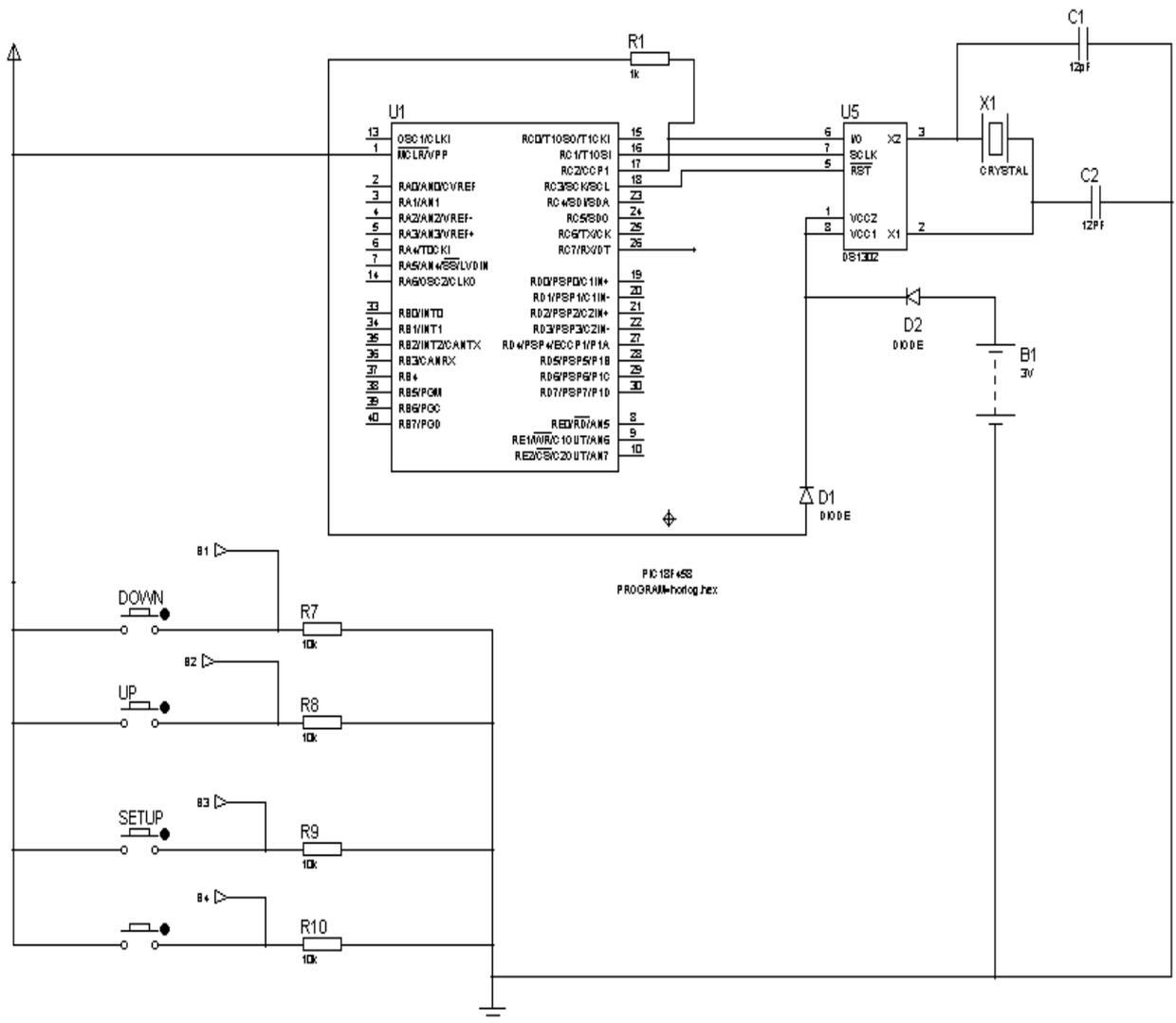


**Figure 3:** Schéma synoptique.

En effet, on trouve :

- l'unité de traitement et de contrôle des données qui est le microcontrôleur (PIC 18F458).
- Un afficheur de 7SEGMENT pour afficher les données.
- 4 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.

## V .1.2/Branchement d'horloge temps réel : DS1302



### ❖ Fiche technique DS1302:

Circuit intégré RTC (Real Time Clock - Calendrier temps réel permanent). Interface: 3-wire . Type de mémoire: NV SRAM. Capacité: 31B. Tension d'alimentation: 2 à 5,5V DC. Boîtier: DIP8. Montage: traversant. Courant d'alimentation DC: 1.28mA. Format d'horloge: HH:MM:SS. Format de date: YY-MM-DD. Courant de veille: 200nA. Référence d'origine du constructeur (MAXIM-DALLAS): DS1302+.

### V .1.3/ 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 utilisé 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, et ce dépend de la précision des modèles SPICE1 des Composants et de la complication des montages.

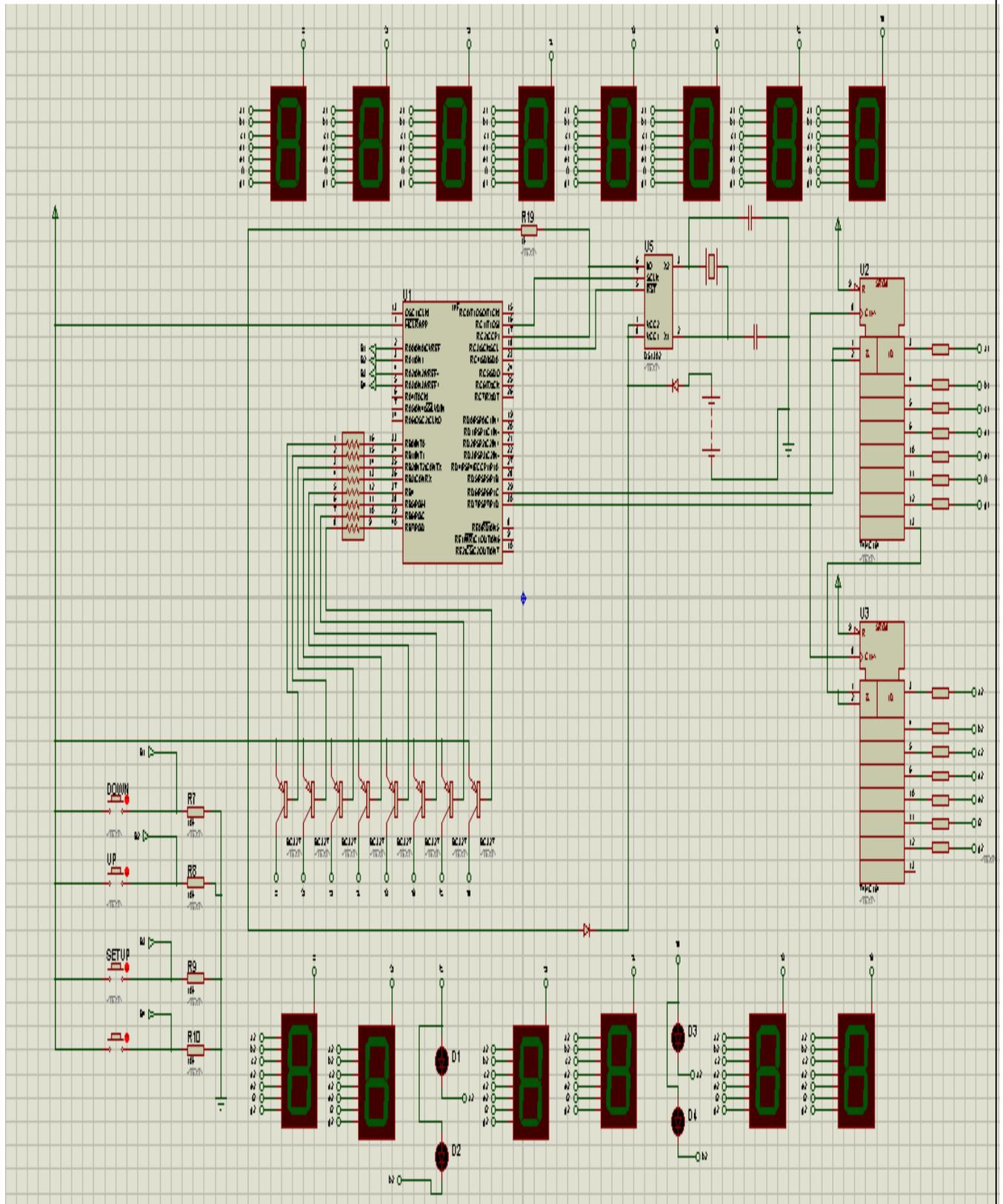
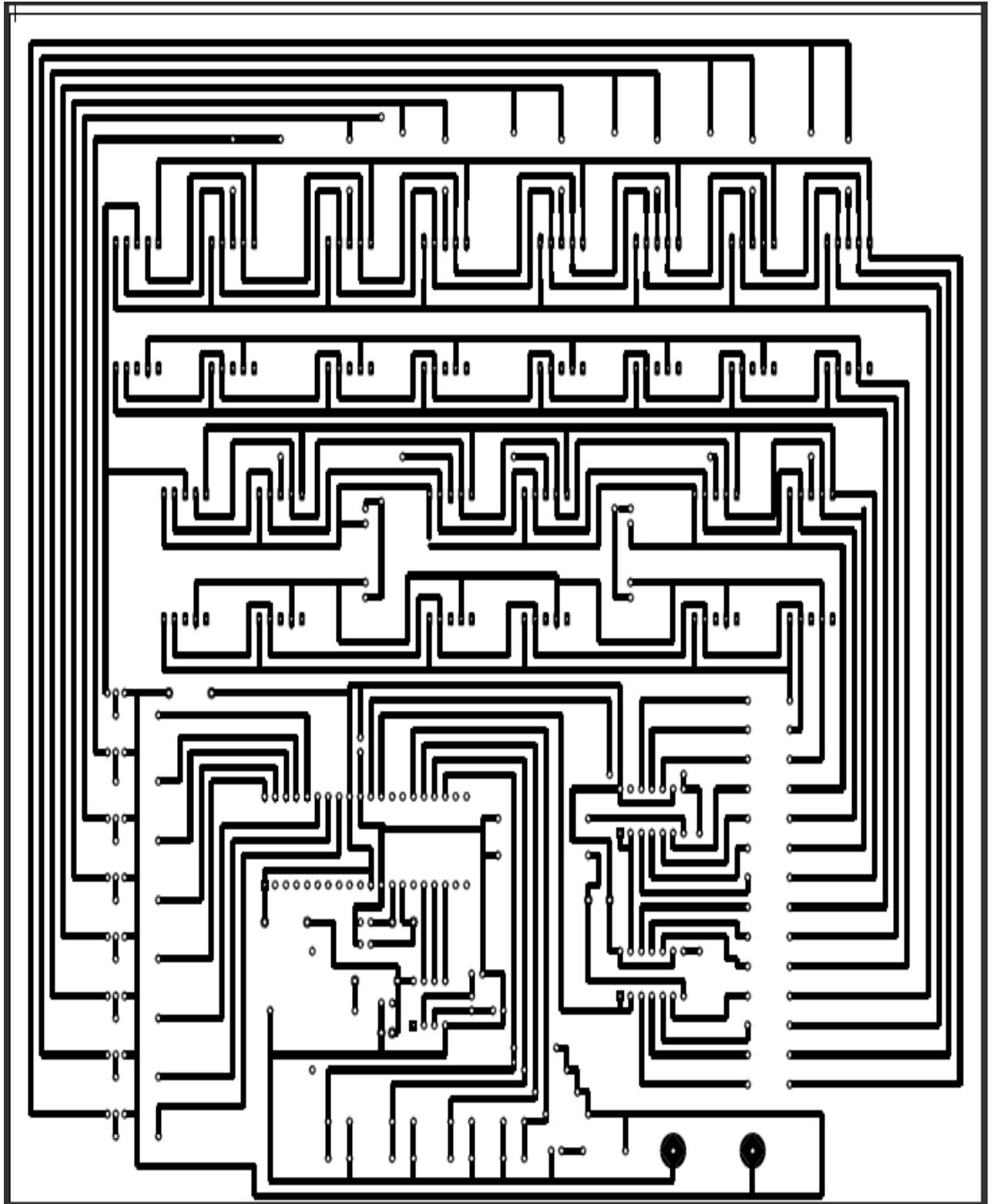
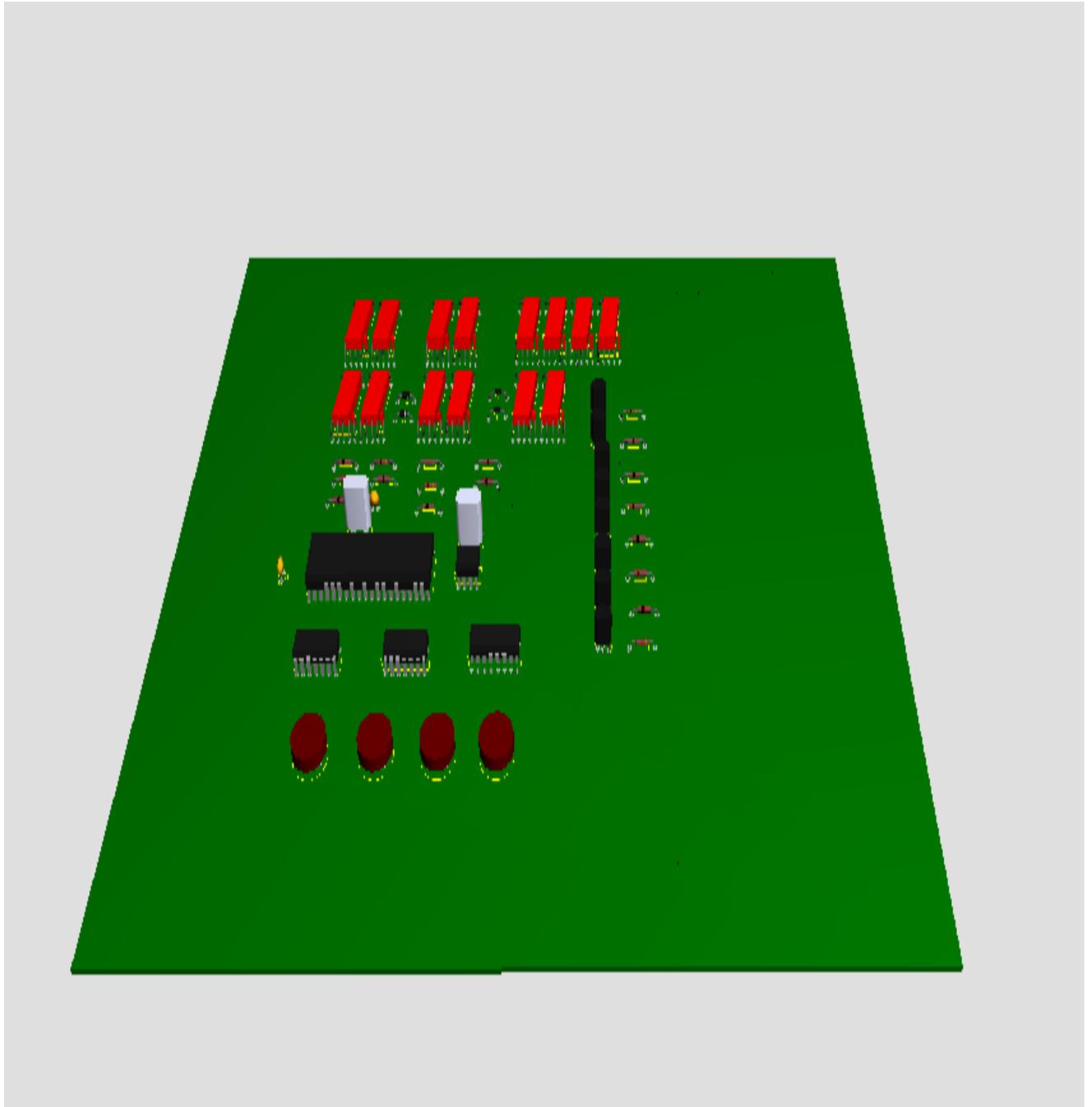


Figure 4: Fenêtre du logiciel de simulation Proteus ISIS.

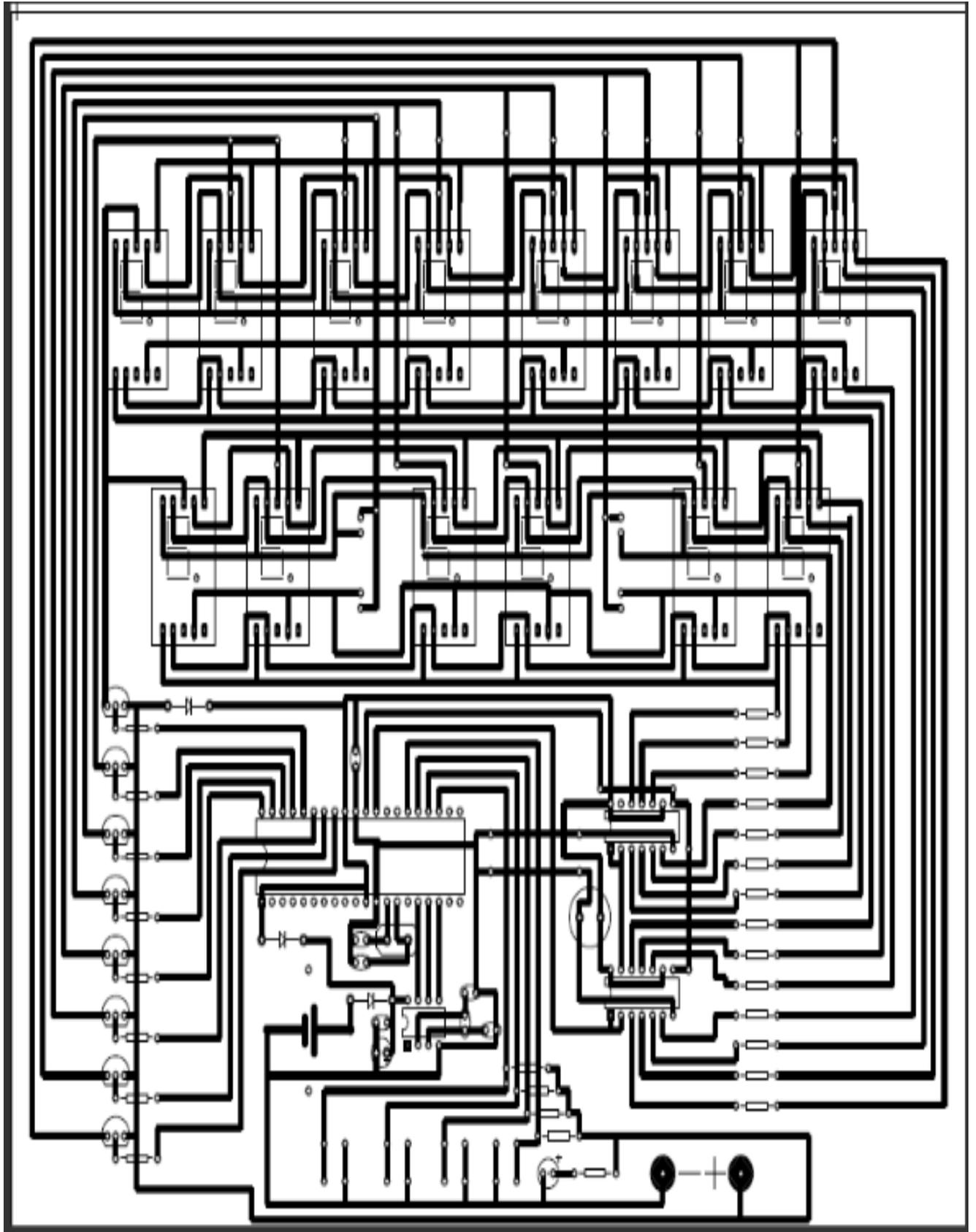
V .1.4/ création du circuit imprimé :



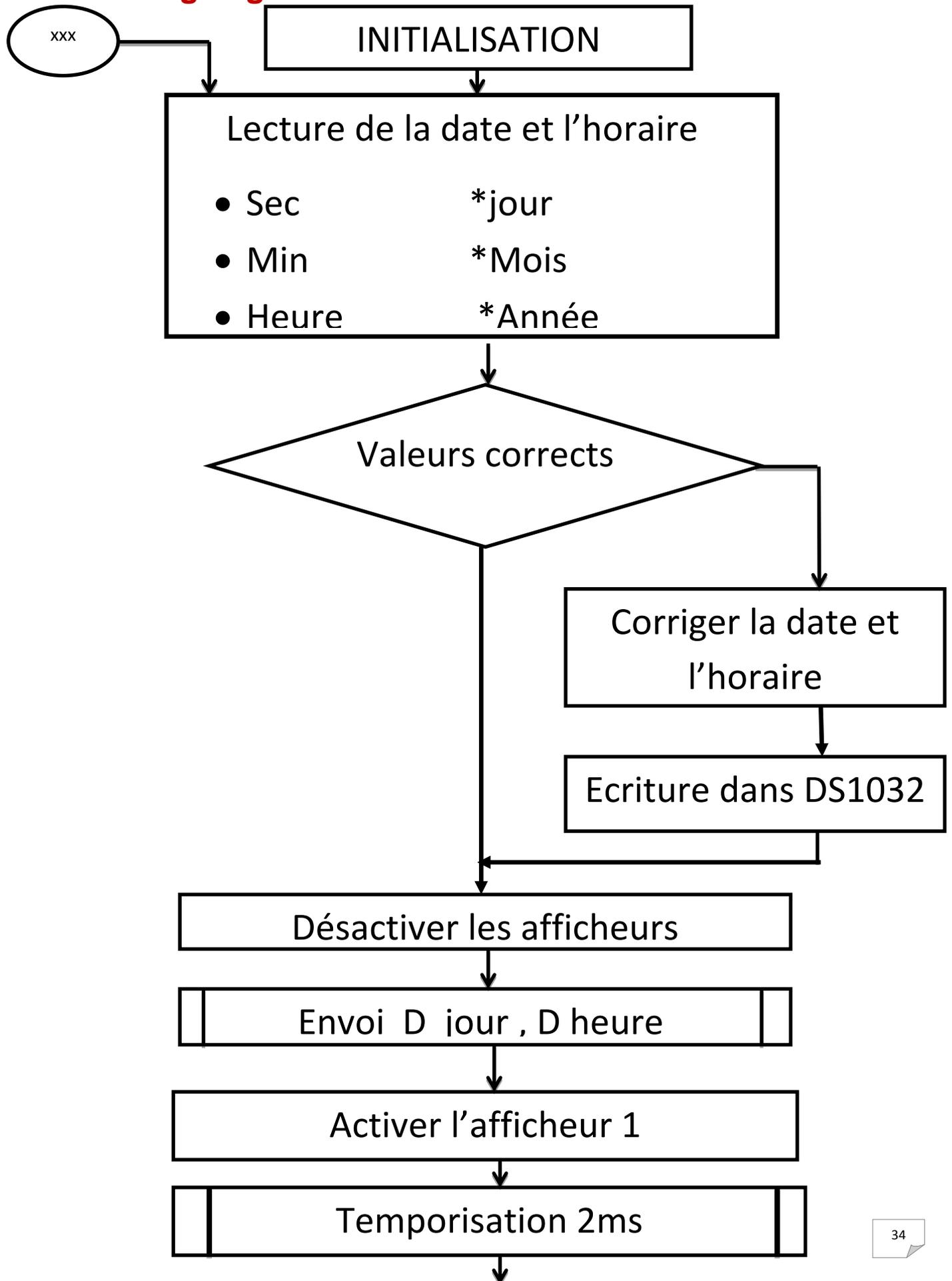
V.1 .5/ Le circuit en 3D :

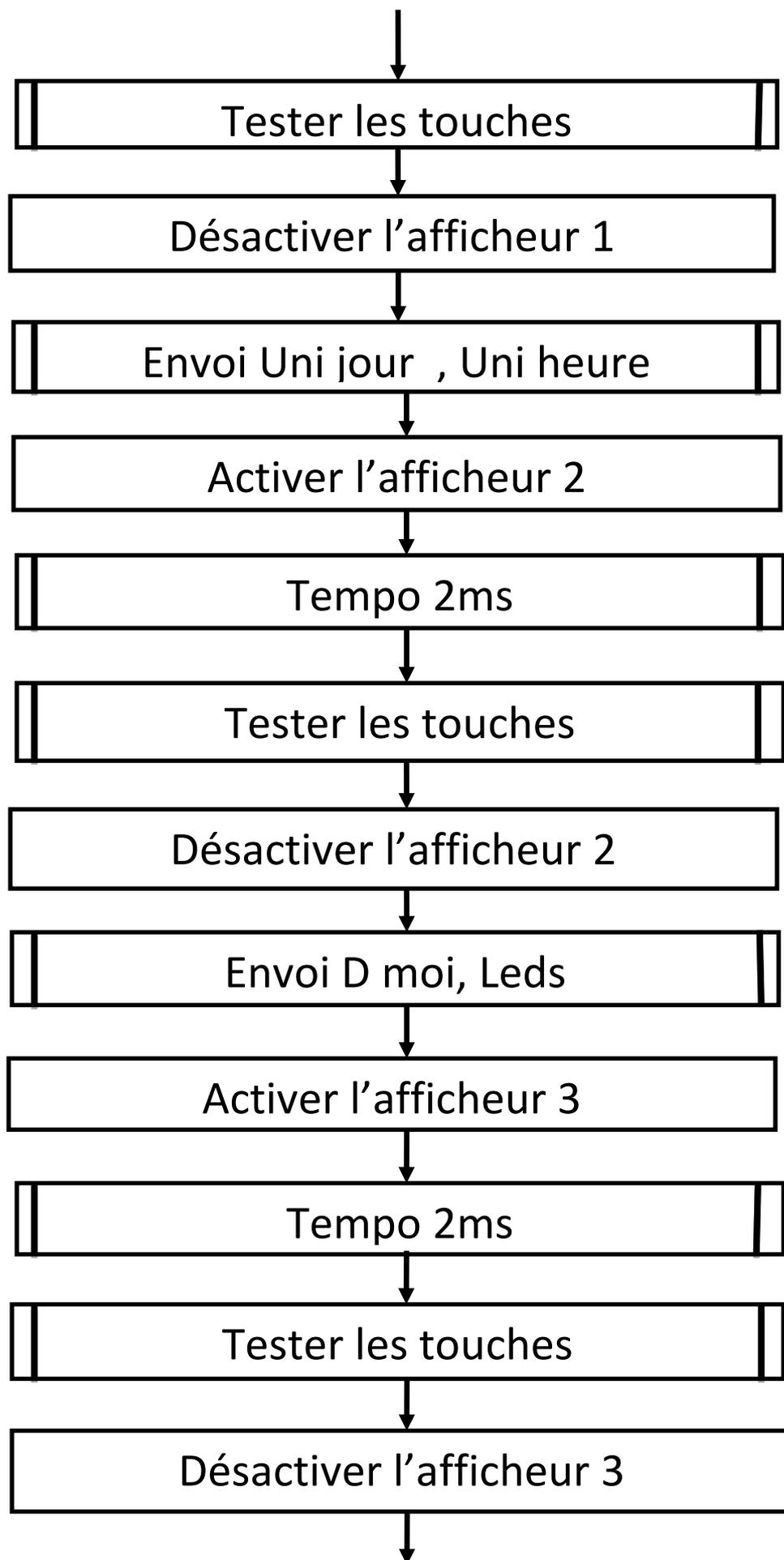


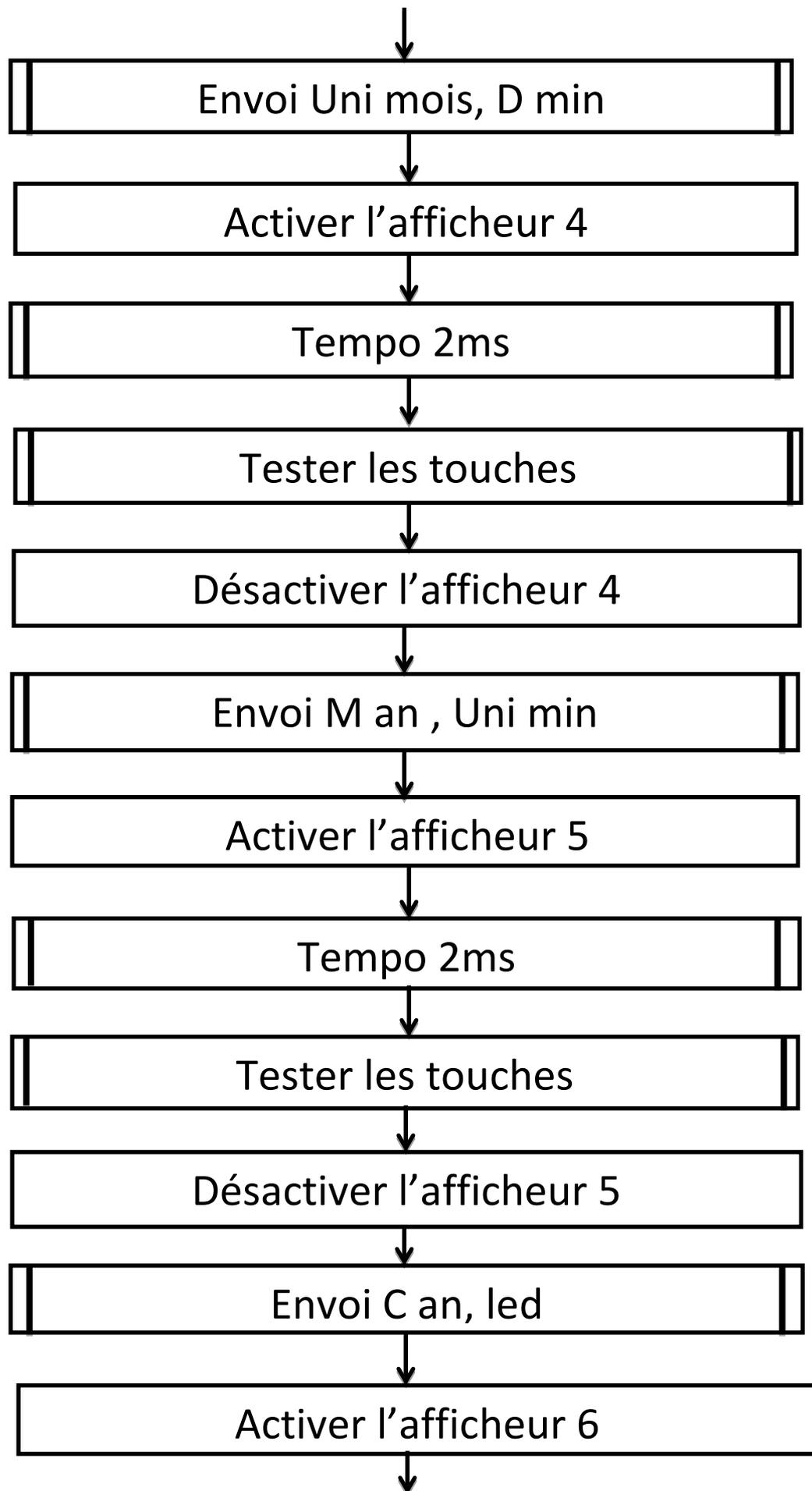
V.1 .6/ *carte principale coté composants :*

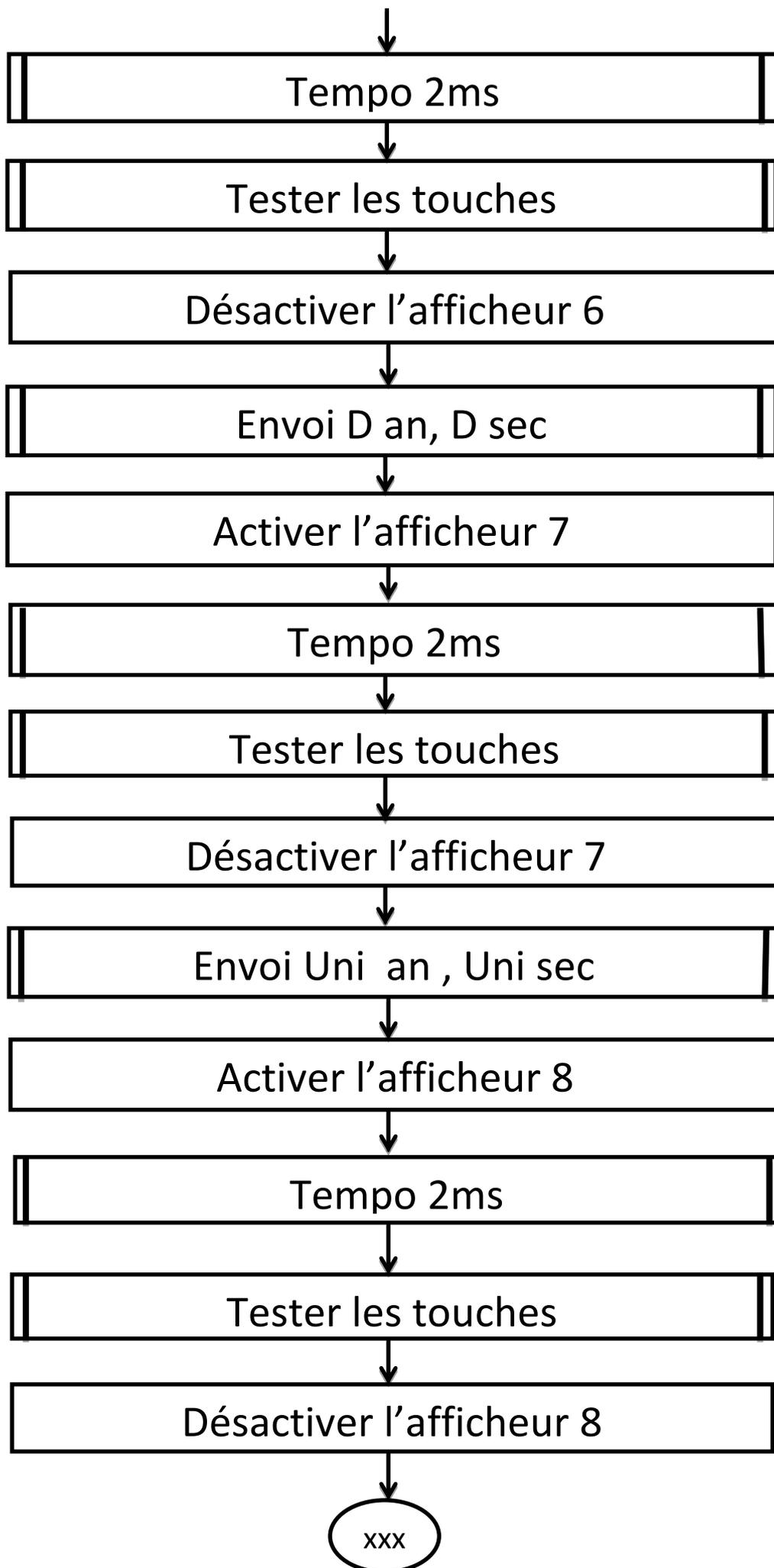


## V.2/L'organigramme :









# CHAPITRE VI

## VI.1/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'en 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.

## **VI .2/Bibliographie :**

**[1] P.F.E : Conception et réalisation d'un enregistreur de données**

**Réalisé par: Alibi Elmehdi Jawadi Sami.**

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

**[3] [http://fr.wikipedia.org/wiki/Afficheur\\_7segments](http://fr.wikipedia.org/wiki/Afficheur_7segments).**

**[4] <http://pdfserv.maxim-ic.com/arpdf/DS1302.pdf>.**

**[5] [http : // www.Microchip .fr .pdf](http://www.Microchip.fr.pdf).**

## VI.3/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 :PIC18F458
- U2 ,U3 :SN74HC164
- U4 : DS1302
- LA PILE CR2032.
- Quartz :12MHZ
- Quartz :3268 HZ
- C1,C2,C3,C4 :12pf
- C5 :100n
- C6 :10 $\mu$ f
- 14 chiffres d'afficheurs 7SEGMENTS
- Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9 :BC327
- 4 boutons poussoirs
- 5 Leds