

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
المركز الجامعي لعين تموشنت  
Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent  
Institut des Sciences et de la Technologie  
Département de Génie Electrique



Projet de fin d'études  
Pour l'obtention du diplôme de Master en :  
Domaine : SCIENCE ET TECHNOLOGIE  
Filière : GENIE ELECTRIQUE  
Spécialité : ELECTRONIQUE BIOMEDICALE  
Thème

## **Contrôle d'un afficheur graphique LCD par le microcontrôleur 8051**

**Présenté Par :**

- 1) RACHIDI Hanane
- 2) FEKROUN Sihem

**Devant les jury composé de :**

Dr BENYAHIA Karima	M CA	C.U.B.B (Ain Temouchent	Présidente
Mr BENCHERIF Kaddour	M A.A	C.U.B.B (Ain Temouchent	Encadrant
Mr. MECIRDI Nour eddine	M A.A	C.U.B.B (Ain Temouchent	Examineur

*Année universitaire 2015/2016*



*★ REMERCIEMENTS ★*

*En préambule à ce mémoire nous remercions **ALLAH** qui nous aide et nous donne la patience et le courage durant ces longues années.*

*En premier lieu, je tiens à remercier mon encadrant **Mr BEN Cherif***

*Enseignant au **CUBB** pour avoir accepté de diriger ce travail. Son soutien, ses compétences et sa riche expérience m'ont été d'une aide inestimable ses précieuses directives tout au long de la réalisation De ce travail m'ont beaucoup impressionnée.*

*Je tiens à remercier sincèrement les membres du jury composé de monsieur le président de jury **Dr Benyahia Karima** et **Mr Mecirdi Nour Eddine** me font le grand honneur d'évaluer Ce travail.*

*Nous tenons aussi à remercier tout particulièrement et à témoigner toute notre reconnaissance aux enseignants qui nous ont donné les bases de l'électronique biomédicale.*

*Nos remerciements vont également à tous le staff administratif, en particulier **Mr Benaza** le chef du département.*

# DÉDICACES

*A mes chers parents*

*Qui m'ont soutenu et encouragé durant ces années d'études.*

*Qu'ils trouvent ici le témoignage de mon profonde reconnaissance.*

*A mes frères et ma sœur*

*A toutes ma famille et mes amis*

*A mon binôme*

*Je vous dois pour votre sympathie et votre appui continu,*

*Des remerciements que les mots ne peuvent traduire.*

*A tous ceux qui me nomment dans le secret de leur cœur*

*A tous ceux qui m'aime*

*A tous ceux que j'aime.*

**HANANE**

# DÉDICACES

*Je dédie ce modeste travail*

*Ames parents pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.*

*A ma sœur **FATIMA** et mon frère **MOHAMED** pour leur tendresse et leur complicité.*

*Ames camarades **SIHAM** et **SOUMIA** pour leurs encouragements, pour l'ambiance agréable tout au long de ce mémoire et pour ses présences dans les moments difficiles et grâce à ceux j'ai passé d'excellents moments.*

*Ames amies **ZINEB** et **ASSIA** plus grande source de bonheur je n'espère que la vie lui réserve le meilleur.*

*A toute ma famille ainsi qu'à mes amis et voisins pour leur collaboration et leur encouragement.*

*A mon binôme **HANAN** qui m'a apporté son aide à la réalisation de ce travail.*

**SIHEM**

# TABLE DE MATIERES

---

## TABLE DE MATIERES

LISTE DES FIGURES .....	i
LISTE DES TABLEAUX.....	ii
INTRODUCION GENERALE.....	1

### **Chapitre I : Les afficheurs graphiques**

I-1) afficheurs à cristaux liquides (LCD) .....	2
I-1-1) introduction .....	2
I- 1-2) constitutions électriques d'un afficheu.....	3
I-1-3) Constitution d'un afficheur à cristaux liquides à transmission.....	3
I-1-4) Principe de fonctionnement d'un afficheur à cristaux liquides.....	4
I-1-5) L'arrangement des cristaux liquides .....	5
I-1-5-1) Afficheur LCD à transmission ou rétro éclairés.....	5
I-1-5-2) Afficheur LCD à réflexion .....	5
I-1-5-3) Afficheur LCD transflectif .....	6
I-1-5-4) Afficheur numérique .....	7
I-1-5-5) Afficheur semi-graphique ou alphanumérique.....	7
I-1-5-6) Afficheur graphique .....	8
I-1-6) Commande d'un afficheur LCD .....	8
I-1-7) Principe de contrôle d'un LCD.....	9
I-2) Conclusion.....	9

### **Chapitre II : étude de microcontrôleur 8051 et ces jeux d'instruction**

II-1Introduction .....	10
II-2Généralité .....	10
II-2-1 définition d'un microcontrôleur .....	10
II-2-2) Avantages des systèmes à base de microcontrôleurs .....	11
II-2-3) Inconvénients des systèmes à base de microcontrôleurs .....	12
II-3 Architecture générale d'un microcontrôleur .....	12
II-3-1) Structure générale d'un système programmable.....	12
II-3-2) Organisation interne d'un microcontrôleur.....	14
II-4 Les microcontrôleurs de la famille 8051 .....	14

## TABLE DE MATIERES

---

II-4-1 Le noyau 80C51 .....	15
II-5) L'unité centrale de traitement (Central Processor Unit).....	17
II-5-1 Mémoire programme et mémoire de données .....	17
II-6 Structure de processeur C51 .....	19
II-6-1 Traitement des données .....	19
II-6-2 Gestion des données .....	19
II-6-3 Traitement des instructions .....	19
II-6-4 Port d'entrée /sortie:( E/S).....	20
II-6-5 Les compteurs / temporisateurs (timer 0 et 1).....	20
II-6-6 Communication série.....	21
II-6-7 Les interruptions.....	21
II-6-8 Circuit d'horloge: .....	21
II-7 Présentation matérielle .....	22
II-7-1 Notion de fonction secondaire.....	23
II-7-2 Description des broches .....	23
II-8 CONCLUSION .....	25
II-9 INTRODUCTION .....	26
II-10 Les modes d'adressages .....	26
II-10-1 Adressage immédiat.....	26
II-10-2 Adressage direct.....	27
II-10-3 Adressage par registre.....	27
II-10-4 Adressage indirect par registre.....	28
II-10-5 Adressage indirect indexé par registre .....	28
II-10-6 Adressage au niveau du bit .....	29
II-11 JEU D'INSTRUCTIONS .....	29
II -11-1 Définition .....	29
II-11-2 Classification des instructions.....	30
II-11-2-1 Instructions de transfert de données .....	30
II-11-2-2 Instructions arithmétiques.....	32
II-11-2-3 Instructions logiques et booléennes .....	32
II-11-2-4 Les instructions de branchement .....	33
II-11-2-5 Les instructions de branchement conditionnel .....	35
II-11-2-6 Instruction Diverses .....	36
II-11-2-7 Les interruptions .....	36
II-11-2-7-1 Gestion des interruptions.....	36

## TABLE DE MATIERES

---

II-11-2-7-2 Déroulement d'une interruption .....	37
II-11-2-8 : Les timers .....	38
II-12 Conclusion.....	39

### Chapitre III : Etude et conception du circuit électronique sous proteus

III-1) Introduction .....	40
<b>III-2) Présentation générale du proteus :</b> .....	40
III-2-1) ISIS .....	41
III-2-2) ARES .....	41
III-3) Liste de matériel .....	41
III-4) principe de fonctionnement de chaque étage.....	42
III-4-1) Afficheur graphique LCD.....	42
III-4-1-1) La mise en œuvre d'un afficheur LCD .....	42
III-4-2) Microcontrôleur 8051 .....	44
III-4-3) Quartz 1.2 MHZ .....	46
III-4-3-1) Applications.....	47
III-4-4) La capacité.....	49
III-4-4-1) Caractéristiques.....	50
III-4-5) La porte logique NON-ET(NAND).....	50
III-4-5-1) Equation.....	51
III-4-5-2) Fonction (ET-NON ou NAND).....	52
III-4-6) Circuit intégré 74LS373 .....	52
III-5 partie software.....	53
III-5-1 mikroC PRO pour 8051:.....	53
III-5-2 programme en langage C .....	53
III-5-3 programme en HEXA .....	57
III.6) réalisation de circuit imprimé .....	58
III-9 conclusion .....	60
Conclusion générale .....	61



## LISTE DES FIGURES

---

### LISTE DES FIGURES

Figure I-1 : constitution d'un afficheur graphique LCD .....	3
Figure I-2 : cristaux liquides .....	4
Figure I-3 : fonctionnement de base.....	5
Figure I-4 : afficheur LCD à transmission et à réflexion .....	6
Figure I-5 : afficheur LCD transflectif .....	7
Figure I-6 : afficheur numérique .....	7
Figure I-7 : afficheur alphanumérique .....	8
Figure I-8 : afficheur graphique .....	8
Figure II-1 : un microcontrôleur réputé.....	11
figureII -2:structure d'un système programmable.....	13
Figure II.3: Architecture détaillée du 8051 .....	14
Figure II-4 : organisation interne de microcontrôleur .....	14
Figure II-5 : Schéma bloc du noyau 80C51 .....	17
Figure II-6 : Organisation de la mémoire interne.....	18
Figure II-7: Organisation des espaces mémoires du 8051 .....	18
Figure II-8- : Brochage de 8051 .....	23
Figure II-9 Signification des symboles utilisés dans la classification des instructions.....	30
Figure II-10 : interruptions de 8051/8052 .....	37
Figure III.1 : circuit réalisé sous proteus ISIS.....	40
Figure III.1 : circuit réalisé sous proteus ISIS.....	40
Figure III-2 : Afficheur graphique LCD .....	43
Figure III-3 : schéma de principe d'un afficheur LCD .....	44
Figure III-4 : Structure d'un système à microprocesseur.....	44
Figure III-5 : Le microcontrôleur 8051 .....	46
Figure III.6 : Schéma du microcontrôleur 8051 .....	46
Figure III-7 : Quartz 1.2 MHZ .....	47
Figure III- 8 Composition d'un filtre à quartz.....	47
Figure III-9 : Symbole et circuit équivalent d'un quartz .....	49
Figure III-10 Les différents types de capacité.....	50
Figure III-11 :4 portes NAND.....	51
Figure III-12 : Chronogramme de la fonction NON-ET .....	52



## LISTE DES FIGURES

---

Figure III-13 : Le 74L373 .....	53
Figure III-14 : schéma du circuit sous ARES .....	59
Figure III-15 : schéma négatif du circuit sous ARES .....	60
Figure III-16 : visualisation 3D du circuit sous ARES .....	60

**LISTE DES TABLEAUX**

**Tableau II-2** Signification des symboles utilisés dans la classification des instructions.....30

**Tableau II-3** : interruptions de 8051/8052.....37

# **INTRODUCTION GENERALE**

### Introduction Générale :

Tous les photoscopes modernes (ceux de moins de deux ans !) possèdent comme interface avec l'utilisateur, un, voire deux écrans de contrôle. Les modèles les plus simples se contentent de l'affichage minimum des informations vitales : état des piles, nombre de vues restantes, mode de fonctionnement du flash, qualité et format des images etc...

Les indications apparaissent dans un petit écran à cristaux liquides comme celui utilisé pour les calculatrices ou les montres. Pour la suite, nous appellerons cet écran, "écran d'état" ou "écran de mode". Dans les modèles plus évolués (et aussi plus chers) un deuxième écran de dimensions plus importantes est utilisé pour visualiser les photographies prises. Cet affichage en couleur s'effectue aussi sur un écran à cristaux liquides, que nous qualifierons d'"écran de contrôle".

Les deux types d'écran sont basés sur le même principe : l'utilisation de cristaux liquides, mais la technologie mise en œuvre diffère considérablement pour les deux modèles. Ainsi, les écrans à cristaux liquides sont appelés ACL, ou en anglais, LCD (liquide cristal display.)

Un deuxième sigle est rajouté, qui qualifie le type d'écran : les petits écrans d'état sont dits TN (Twist Nematic) ou STN (Super Twist Nematic,) les écrans de contrôle sont eux dénommés TFT (Thin Film Transistor). Les premiers entrent dans la catégorie d'afficheurs "passifs" et les seconds sont, par opposition, dans la catégorie "actifs", plus précisément à "matrice active".

L'objectif de notre travail est d'étudier en détail les afficheurs et la conception d'un circuit qui sera contrôlé par le microcontrôleur 8051 sous environnement de Proteus.

Nous avons divisé notre mémoire en quatre chapitres : le premier chapitre sera plus particulièrement consacré à l'étude des afficheurs graphiques LCD.

Le deuxième chapitre va traiter la programmation du microcontrôleur 8051.

Dans le troisième chapitre on entamera l'étude des différentes parties du circuit. Le quatrième chapitre sera consacré sur la réalisation du circuit et programmation sous environnement Proteus ISIS et en passant au circuit imprimé et la visualisation 3D sous ARES.

Nous achèverons notre mémoire par une conclusion générale.

# CHAPITRE I

## **I-1) afficheurs à cristaux liquides (LCD) :**

### **I-1-1) Introduction :**

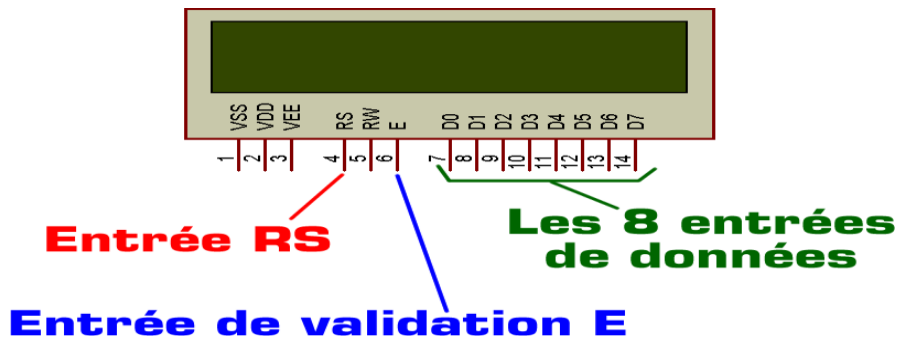
Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquide Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).

Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables.

Un afficheur LCD contient :

- une entrée de contrôle RS (Register Select)
- une entrée de validation E (Enable)
- 8 entrées de données D0 à D7 [3] [4].



**Figure I-1** : Constitution d'un Afficheur Graphique LCD

### I 1-2) Constitutions électriques d'un Afficheur :

Chaque élément de l'afficheur (segment, pictogramme, point, etc.) nécessite 2 surfaces conductrices (électrodes) en regard.

Pour le détail sur le principe de fonctionnement d'un afficheur à cristaux liquides, voir l'annexe.

Une des surfaces est située sur la face arrière de l'afficheur : « back plane »

Selon le nombre d'éléments à afficher, on rencontre 2 types de configurations :

- 1 commun (back plane) pour tous les éléments + 1 électrode pour chaque élément
- plusieurs communs + 1 électrode pour chaque groupe d'éléments interconnectés

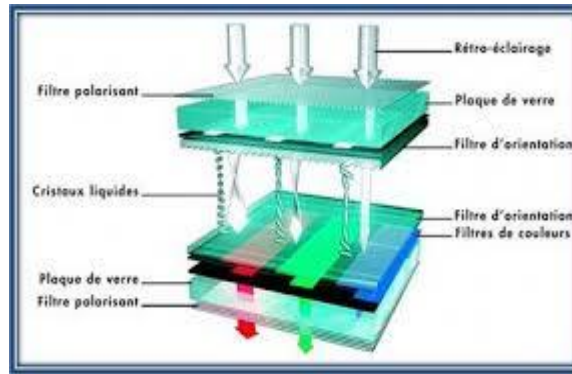
La dernière méthode permet de réduire le nombre de broches de l'afficheur.

Les commandes dépendent directement de l'organisation des électrodes [5].

### I-1-3) Constitution d'un Afficheur à Cristaux Liquides à transmission :

Le cristal liquide est contenu entre deux plaques de Verre très proches. Des électrodes transparentes Ayant la forme d'un segment de l'afficheur sont déposées à la surface de chacune des plaques de verre. Elles permettent de créer un champ électrique. Ce dispositif est inséré entre deux polariseurs (filtres polarisants croisés) [5].





**Figure I-2 :** Cristaux liquides.

#### **I-1-4) Principe de fonctionnement d'un Afficheur à Cristaux Liquides :**

Le premier filtre polarisant ne laisse passer que l'onde électromagnétique horizontale. En l'absence de champ électrique externe, les molécules du cristal liquide s'orientent aléatoirement dans toutes les directions de l'espace. Dans cet état, le liquide fait tourner la direction de polarisation de toute lumière incidente, d'un angle de  $90^\circ$ . Le second filtre polarisant laisse passer l'onde électromagnétique verticale. La lumière traverse le segment transparent.

Le cristal liquide soumis à un champ électrique perd sa propriété de rotation de polarisation d'une onde lumineuse. Le second filtre polarisant ne laisse pas passer l'onde électromagnétique horizontale. La lumière ne traverse pas le segment devenu opaque (noir) [5].

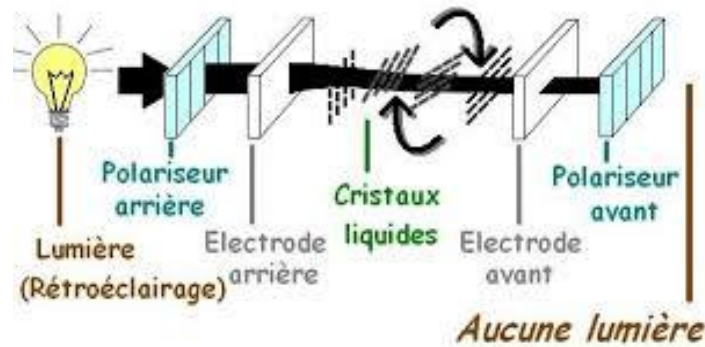


Figure I-3 : Fonctionnement de base.

### I-1-5) L'arrangement des Cristaux Liquides :

#### I-1-5-1) Afficheur LCD à transmission ou rétro éclairés :

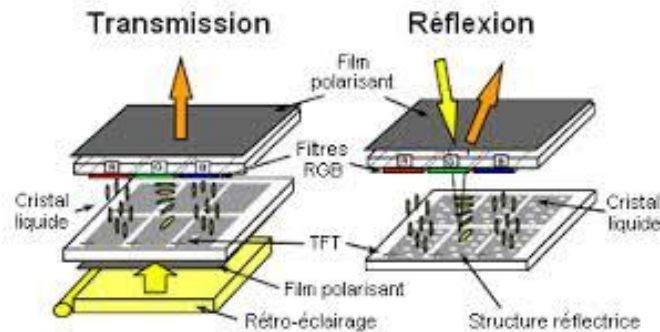
Ces afficheurs LCD sont constitués de deux électrodes transparentes. Ils nécessitent un Éclairage par l'arrière. Cette lumière est incorporée par le fabricant à son panneau LCD. La consommation du panneau d'éclairage n'est pas négligeable dans le cas d'une Alimentation autonome. Cet afficheur présente l'avantage d'être visible dans l'obscurité. Ils sont utilisés pour les écrans d'ordinateurs portables, téléphones portables... [6].

#### I-1-5-2) Afficheur LCD à réflexion :

Dans ce type d'afficheur le plan arrière porte une surface réfléchissante à la suite du Filtre polarisant. La lumière provenant de la face avant traverse successivement : le filtre polarisant avant, le cristal liquide puis le deuxième polariseur. En l'absence de tension entre les deux électrodes les cristaux liquides sont transparents et la lumière Est totalement réfléchi par la surface métallique arrière. En présence de tension entre les deux électrodes le liquide est opacifié, la lumière n'est plus réfléchi, le segment est affiché.

Ce type d'afficheur est conçu pour être éclairé par l'avant. La consommation électrique réduite à celle de la commande est très faible. Ces afficheurs sont utilisés pour les applications où

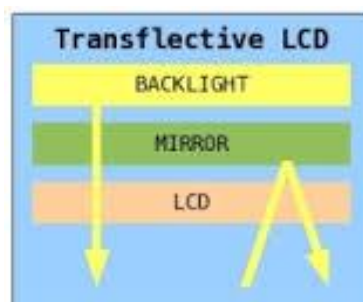
l'autonomie en alimentation doit être grande : calculettes, montres, thermomètres.... La lecture de l'afficheur n'est possible que dans un environnement suffisamment éclairé [6].



**Figure I-4 :** Afficheur LCD à transmission et à réflexion.

### I-1-5-3) Afficheur LCD transflectif :

Dans ce type d'afficheur le plan arrière porte une surface semi-transparente. Cette surface transmet environ 47 % de la lumière de la face arrière et réfléchit environ 52 % de la lumière reçue par la face avant [6].



**Figure I-5 :** Afficheur LCD transflectif.

**I-1-5-4) Afficheur numérique :**

Un afficheur numérique permet d'afficher uniquement des chiffres. Chaque chiffre est constitué de 7 ou 8 segments à cristaux liquides. Ce type d'afficheur est essentiellement utilisé dans les montres digitales, les réveils, les appareils de mesure [6].



**Figure I-6 :** Afficheur numérique.

**I-1-5-5) Afficheur semi-graphique ou alphanumérique :**

Un afficheur alphanumérique permet d'afficher des chiffres, des lettres et quelques caractères spéciaux. Ce type d'afficheur est constitué d'un certain nombre de lignes, constituées elles-mêmes par un certain nombre de caractères (ex : 2 lignes de 16 caractères).

Chaque caractère est composé d'une matrice de points (généralement 5 x 7) [6].



**Figure I-7 :** Afficheur alphanumérique.

### I-1-5-6) Afficheur graphique :

Dans ce type d'afficheur est constitué d'un certain nombre de pixels fonction de la définition de l'écran. Chaque pixel est commandé séparément les uns des autres. Il permet d'afficher des dessins beaucoup plus évolués [6].



**Figure I-8 :** Afficheur graphique

### I-1-6) Commande d'un Afficheur LCD

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur (voir plus bas).

#### **Mode 8 bits :**

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches D0 à D7. On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7. L'afficheur lira la donnée sur le front descendant de cette entrée.

#### **Mode 4 bits :**

Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture. Les données seront valides sur les lignes D0 à D7 lors de l'état haut de la ligne E.

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches

d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble.

Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy BF. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée [2].

#### **I-1-7) Principe de contrôle d'un LCD :**

Les écrans LCD se pilotent avec 4 ou 8 pattes pour les données, et 3 pattes de contrôle. Il existe un simulateur très pratique pour s'essayer à la programmation (mais c'est facultatif, des bibliothèques de code se trouvant facilement quelque soit votre plateforme de programmation) [7].

#### **I-2) Conclusion :**

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage de paramètres de fonctionnement. Grâce à la commande par un microcontrôleur ces afficheurs permettent de réaliser un affichage de messages aisés. Ils permettent également de créer ses propres caractères.

# CHAPITRE II



## **II-9) Introduction :**

Tous les membres de la famille **MCS-51** exécutent le même jeu d'instructions. Ces instructions ont été optimisées pour convenir à la commande et au contrôle de processus se suffisant d'une architecture 8 bits. Il autorise de nombreux modes d'adressage permettant entre des accès rapides aux données logées en RAM interne facilitant ainsi des opérations portant sur de petites structures données.

Cette organisation du type processeur de données 8 bits est complétée par la possibilité de manipuler directement des variables réduites au formant du bit (variables booléennes) .Le constructeur parle d'un processeur booléen pour définir cette possibilité supplémentaire .Ce processeur n'est en fait, vu de l'utilisateur, qu'une mode d'adressage supplémentaire. Le processeur booléen utilise l'indicateur de report repéré par le symbole CY comme accumulateur. On peut alors effectuer toutes les opérations logiques classiques directement entre cet accumulateur et des variables binaires en évitant les opérations préalables d'isolement [12].

En introduction à la description du jeu d'instructions, il est proposé dans les paragraphes suivants, d'une part, de préciser les modes d'adressage possibles ainsi que leur syntaxe et, d'autre part, de définir les groupes d'instruction existant pour cette famille de microcontrôleurs [12].

## **II-10) Les Modes d'adressages :**

Le jeu d'instructions utilise six modes d'adressage qui sont les suivant:

### **II-10-1) Adressage Immédiat :**

Au niveau code machine ce mode d'adressage se traduit par le fait que le code opération est suivi d'un deuxième octet représentant une donnée. Du fait que cette donnée est logée dans la mémoire programme, en général mémoire seulement accessible en lecture, elle constitue donc par définition une constante. Au niveau la syntaxe dans le code source, l'adressage immédiat utilise le caractère # [12].

**Exemple:**

Placer dans A la valeur 30H/

```
MOV A, #30H
```

Placer dans A la valeur fixée au symbole VAL:

```
MOV A, #VAL
```

**II-10-2) Adressage Direct :**

Un des opérandes de l'instruction est une adresse sur le contenu de laquelle va porter l'opération. Aucun signe particulier n'est employé dans la syntaxe du code source [12].

**Exemple:**

```
MOV A, 30H
```

Signifie placer dans A la donnée situé à l'adresse 30H.

Ce mode d'adressage est le moyen d'accès aux registres à fonction spéciale.

D'une façon générale, l'adressage direct ne concerne que les 128 octets de poids faible de RAM interne et les SFR [12].

**II-10-3) Adressage par Registre :**

Le (ou les) opérande(s) de l'opération sont représentée par des noms de registres. En fait l'opération porte sur les valeurs contenues dans ces registres [12].

**Exemple:**

Transfert dans A de la valeur contenu dans R1:

```
MOV A, R1
```

Additionner à cette valeur le contenu de R0

```
ADD A, R0
```

**II-10-4) Adressage Indirect par Registre :**

L'adressage à indirect par registre utilise le contenu des registres R0 ou R1 comme une adresse 8 bits. Ces pointeurs permettent de travailler sur des données situées dans les 256 premiers octets de la RAM interne, ou sur une page de 256 octets de la mémoire externe. Dans ce cas la page est définie par le contenu du PORT P2 .Ce mode d'adressage ne permet pas un accès aux registres à fonction spéciale (SFR).

Le signe @ est utilisé dans le code source pour indiquer à l'assembleur ce mode d'adressage [12].

**Exemple:**

Placer dans R0 la valeur 30H par adressage immédiat:

```
MOV R0, #30H
```

Le contenu de l'adressage 30H est placé dans A

```
MOV A, @R0
```

Le même type d'adressage est possible avec le registre 16 bits DPTR; l'adressage porte alors sur les 64 KO octets de la mémoire externe. Les instructions de sauvegarde sur la pile et récupération depuis la pile utilisent implicitement ce mode adressage grâce au pointeur SP [12].

**II-10-5 Adressage Indirect Indexé par Registre :**

Ce mode d'adressage ne permet d'accéder qu'en mode lecture à la zone de mémoire programme. Il permet une récupération de données structurées sous forme de table placée dans le code programme.

Dans ce mode d'adressage, le compteur programme PC ou le pointeur ou le pointeur de données DPTR servent de pointeur .Le déplacement ajouté à la valeur de ce pointeur est le contenu de l'accumulation, et la valeur récupérée est placée dans l'accumulateur. Ce mode d'adressage facilite l'accès à des tables de données mais la valeur du déplacement est perdue [12].

**Exemple:**

On ajoute 1 à la valeur de A pour compenser l'instruction RET :

```
INC A
```

Puis A prend la valeur 50H si A valait 1 avant appel de ce sous-programme:

```
MOV A, @ A+PC
```

```
RET
```

```
DB 50H
```

```
DB 60H
```

```
Etc.
```

**II-10-6) Adressage au Niveau du Bit :**

Un adressage au niveau du bit est possible. L'adresse étant spécifié sur un octet, le nombre de bits directement adressables est donc limité à 256 .L'adresse de ces bits est définie d'une part, dans le tableau des registres à fonction spéciale, d'autre part dans le tableau détaillant la RAM interne [12].

**II-11) JEU D'INSTRUCTIONS :****II -11-1) Définition :**

- Partie de l'architecture avec laquelle le programmeur ou le concepteur de compilateur est en contact.
- Ensemble des instructions que peut exécuter un processeur.
- Ensemble des circuits logiques câblés dans un processeur [18].

**II-11-2) Classification des Instructions**

Signification des symboles utilisés :

<b>Rn</b>	Un des registres actifs R0 à R7
<b>direct</b>	Adresse d'un octet de RAM interne, d'un port, ou SFR
<b>@Ri</b>	Adressage indirect par registre R0 ou R1
<b>#data</b>	Donnée 8 bits
<b>#data 16</b>	Donnée 16 bits
<b>Bit</b>	Adresse au niveau du bit
<b>Rel</b>	déplacement 8 bits signé relatif au PC : de -128 à +127
<b>addr11</b>	Adresse limitée au bloc de 2Ko dans lequel figure l'instruction
<b>Addr16</b>	Adresse sur l'espace de 64Ko

**Figure II-9** Signification des Symboles Utilisés dans la Classification des Instructions

Le jeu d'instructions du microcontrôleur 8051 peut être réparti dans les groupes suivants:

- Instructions de transfert de données.
- Instructions arithmétiques.
- Instructions logiques (booléens).
- Instructions de manipulation de bits.
- Programme de contrôle des instructions (saut) [17].

**II-11-2-1) Instructions de Transfert de Données**

Les transferts affectant l'accumulateur s'effectuent en 1 $\mu$ S, les autres en 2 $\mu$ s.

**MOV : déplacement de donnée dans la mémoire interne**

MOV Dest, Source : recopie Source dans Dest, aucun drapeau n'est positionné.

Notons que l'on peut transférer des données directement d'une case mémoire interne dans une autre sans passer par l'accumulateur.

Les instructions du genre MOV Rn,Rn ne sont pas autorisées. Pour réaliser MOV R2, R4 avec le bank 0 actif, il faut faire MOV 02,04. Si c'est le bank 1 qui est actif, pour réaliser MOV R2, R4 il faut faire MOV 0Ah, 0Ch [19].

### **PUSH et POP**

Sauvegarder ou récupérer une valeur dans la pile. PUSH incrémente SP puis écrit l'opérande dans la case pointée par ce dernier POP lit la valeur pointée par SP avant de décrémenter ce dernier

XCH XCH A, : échange les données de l'accumulateur A et de l'octet adressé.

XCHD XCHD A,@Ri : échange les quartets de poids faible entre l'accum A et l'octet adressé [19].

Syntaxe	Code	Octets	cycles
PUSH direct	C0	2	2
POP direct	D0	2	2
XCH A, Rn	C8+n	1	1
XCH A, direct	C5	2	1
XCH A, @Ri	C6+i	1	1
XCHD A, @Ri	D6+i	1	1

MOVX MOVX , permet la lecture ou l'écriture d'un octet en RAM externe en passant toujours par l'accumulateur . L'adressage indirect est utilisé MOVC MOVC A, @A+ permet de lire un octet dans la mémoire programme [19].

Syntaxe	Code	Octets	cycles
MOVX A,@Ri	E2+i	1	2
MOVX A,@DPTR	E0	1	2
MOVX @Ri,A	F2+i	1	2
MOVX @DPTR,A	F0	1	2
MOVC A,@A+DPTR	93	1	2
MOVC A,@A+PC	83	1	2

**II-11-2-2) Instructions Arithmétiques**

**ADD A**, additionne un octet avec l'accum A, résultat dans A.

**ADDC A**, additionne un octet, l'accum A et la retenue, résultat dans A.

**SUBB A**, soustrait un octet et la retenue du contenu de A, résultat dans A.

**INC** : incrémente un octet ou DPTR.

**DEC** : décrémente un octet.

**MUL AB** : multiplie l'accum A et l'accum B, résultat : octet faible dans A et octet fort dans B

**DIV AB** : divise le contenu de A par le contenu de B, quotient dans A et reste dans B.

**DA A** : ajustement décimal de A [19].

**II-11-2-3) Instructions Logiques et Booléennes :**

**ANL** , : réalise un ET logique entre source et dest, résultat dans dest.

**ORL** , : réalise un OU logique entre source et dest, résultat dans dest.

**XRL** , : réalise un OU exclusif entre source et dest, résultat dans dest.

**CLR** met l'accumulateur ou un bit à 0

**CPL** Complémente l'accumulateur ou uRL,

**RLC, RR, RRC, SWAP**

**RL A** : rotation vers la gauche du contenu de A

**RLC A** : rotation vers la gauche du contenu de A à travers la retenue

**RR A** : rotation vers la droite du contenu de A

**RRC A** : rotation vers la droite du contenu de A à travers la retenue

**SWAP A** : échange le n bite quartet de poids faible avec celui de poids fort de A



**SETB**

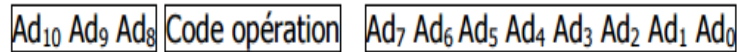
**SETB** : met un bit [19].

ANL A, #data	54	2	1	-
ANL A, @Ri	56+i	1	1	-
ANL A, direct	55	2	1	-
ANL A, Rn	58+n	1	1	-
ANL direct, #data	53	3	2	-
ANL direct, A	52	2	1	-
ANL C, /bit	B0	2	2	C
ANL C, bit	82	2	2	C
ORL A, #data	44	2	1	-
ORL A, @Ri	46+i	1	1	-
ORL A, direct	45	2	1	-
ORL A, Rn	48+n	1	1	-
ORL direct, #data	43	3	2	-
ORL direct, A	42	2	1	-
ORL C, /bit	A0	2	2	C
ORL C, bit	72	2	2	C
XRL A, #data	64	2	1	-
XRL A, @Ri	66+i	1	1	-
XRL A, direct	65	2	1	-
XRL A, Rn	68+n	1	1	-
XRL direct, #data	63	3	2	-
XRL direct, A	62	2	1	-
CLR A	E4	1	1	-
CLR bit	C2	2	1	-
CLR C	C3	1	1	C=0
CPL A	F4	1	1	-
CPL bit	B2	2	1	-
CPL C	B3	1	1	C
RL A	23	1	1	-
RLC A	33	1	1	C
RR A	03	1	1	-
RRC A	13	1	1	C
SWAP A	C4	1	1	-
SETB bit	D2	2	1	-
SETB C	D3	1	1	-

**II-11-2-4 ) Les Instructions de Branchement :**

**ACALL addr11** : réalise un saut absolu incondtionnel vers un sous programme commençant à l'adresse addr11. Les 11 bits de addr11 sont substitués au 11 bits de poids faible du PC, ce qui signifie que la destination reste dans le même bloc de 2K que l'instruction courante.

Le codage se fait d'une façon particulière : les 3 bits de poids fort de addr11 complètent le code Opération dont les 5 bits bas sont égaux 11h [19].



Si on note  $Ad_{10} Ad_9 Ad_8 = \text{page} = P$  et  $Ad_7 Ad_6 Ad_5 Ad_4 Ad_3 Ad_2 Ad_1 Ad_0 = \text{Addr8}$  On obtient le code instruction :  $11h+20hxP \text{ Addr8}$

(P variant entre 0 et 7)

AJMP  $addr_{11}$  : réalise un saut absolu inconditionnel vers l'instruction d'adresse  $addr_{11}$ . La différence avec ACALL est que ici on ne saute pas vers un sous programme, il n'est pas nécessaire d'empiler le PC pour pouvoir retourner après l'exécution du sous programme. Le codage se fait de la même façon que ACALL sauf que COL =  $01h : 01h+20hxP \text{ Addr8}$  (P variant entre 0 et 7) [19].

LCALL  $addr_{16}$  : réalise un saut long absolu inconditionnel vers un sous programme commençant à l'adresse  $addr_{16}$ . Pour repérer la position de retour, le PC est incrémenté de 3 puis empilé. Il sera dépilé au retour du sous programme.

LJMP  $addr_{16}$  : réalise un saut long absolu inconditionnel vers la position d'adresse  $addr_{16}$ .

SJMP rel : réalise un saut court relatif au PC. On peut donc sauter dans l'intervalle : [PC-128, PC+127]

JMP @A+DPTR : réalise un saut long absolu inconditionnel vers la position d'adresse A+DPTR. (Adressage indirect) [19].

Syntaxe	Code instruction	Octets	cycles
ACALL $adr_{11}$	$11h+20hxP \text{ Addr8}$	2	2
AJMP $addr_{11}$	$01h+20hxP \text{ Addr8}$	2	2
LCALL $addr_{16}$	$12h \text{ addr}_{16}$	3	2
LJMP $addr_{16}$	$02h \text{ addr}_{16}$	3	2
SJMP rel	$80h \text{ rel}$	2	2
JMP @A+DPTR	$73h$	1	2

**2- 5) Les Instructions de Branchement Conditionnel :**

- JZ rel : saut si A=0
- JNZ rel : saut si A $\neq$ 0
- JC rel : saut si retenue à 1
- JNC rel : saut si retenue à 0
- JB bit,rel : saut si bit à 1
- JNB bit,rel : saut si bit à 0
- JBC bit,rel : saut si le bit est à 1 et mise à zero de celui-ci
- CJNE , : saut si byte1 et byte2 sont différents
- DJNZ , rel : décrémente byte et saut si résultat différent de0 [19].

JZ rel	60	22
JNZ rel	70	22
JC rel	40	22
JNC rel	50	22
JB bit, rel	20	32
JNB bit, rel	30	32
JBC bit,rel	10	32
CJNE @Ri, #data, rel	B6+i	32
CJNE A, #data, rel	B4	32
CJNE A, direct, rel	B5	32
CJNE Rn, #data, rel	E8+n	32
DJNZ direct, rel	D5	32
DJNZ Rn, rel	D8+n	22

**II-11-2-6) Instruction Diverses :**

RET : Retour de sous programme

RETI : Retour d'interruption

NOP : No Operation [19].

Syntaxe	CO	Octets	cycles
RET	22h	1	2
IRET	32h	1	2
NOP	00	1	1

**II-11-2-7) Les Interruptions :**

Le 8051 possède 5 sources d'interruption, le 8052 en possède une de plus : • Evénement externe sur l'entrée INT0 (P3.2) • Evénement externe sur l'entrée INT1 (P3.3) • Débordement du TIMER0 • Débordement du TIMER1 • Emission ou réception d'un caractère sur le port série • Débordement du Timer 2 ou déclenchement du mode chargement/capture de ce dernier par l'entrée T2EX (P1.1) [19].

**II-11-2-7-1) Gestion des Interruptions :**

D'une manière générale, chaque interruption est gérée par 3 bits situés dans des registres SFR :

- Un drapeau, qui est un bit positionné quand l'événement déclencheur intervient,
- Un bit de validation : Quand l'événement déclencheur intervient si le bit de validation est positionné, l'interruption peut avoir lieu sinon elle est ignorée,
- Un bit de priorité : Avec un bit, chaque interruption a 2 niveaux de priorité, (0=basse), (1=haute). Quand l'événement déclencheur intervient, même si l'interruption en question est validée, si une interruption de même ou de niveau supérieur est en train d'être exécutée, alors notre l'interruption devra attendre. Le bit EA situé à la position 7 du registre SFR IE permet la validation/inhibition globale de toutes les interruptions [19].

**II-11-2-7-2) Déroulement d'une Interruption :**

Quand un événement déclencheur d'une interruption se produit, le drapeau correspondant est positionné (voir tableau ci-dessous), si le bit de validation est positionné et l'interruption est prioritaire alors l'interruption est déclenchée et les tâches suivantes sont réalisées : - La valeur du Program Counter est empilée- Les interruptions de même ou de priorité inférieures sont masquées - Dans le cas des interruptions INT0/1 ou TIMER0/1, le drapeau correspondant est remis à zéro. Pour les autres interruptions, le programmeur doit remettre lui même ce drapeau à zéro au début de la routine d'interruption. - Le PC est chargée par l'adresse du vecteur d'interruption correspondant ce qui transfère l'exécution au début de ce vecteur ou doit résider la routine d'interruption. - A l'exécution de l'instruction RETI qui doit terminer chaque routine d'interruption, le processeur dépile le PC pour retrouver sa place dans le programme principal et remet le statut des interruptions dans l'état où il l'a trouvé avant de commencer cette interruption [19].

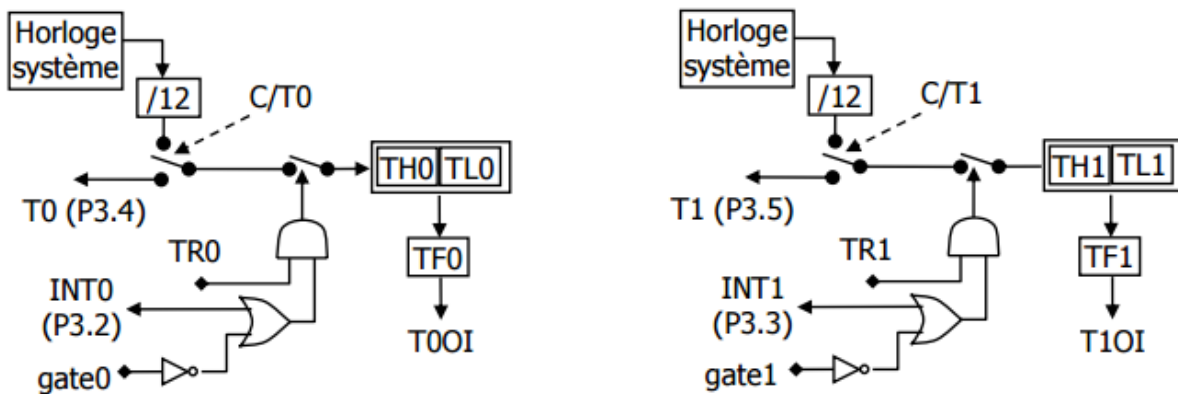
d'interruption	Flag	Validation	vecteur	Priorité	EDG
INT0 (P3.2)	IE0 (TCON.1)	EX0 (IE.0)	0003H	PX0 (IP.0)	IT0 (TCON.0)
INT1 (P3.3)	IE1 (TCON.3)	EX1 (IE.2)	0013H	PX1 (IP.2)	IT1 (TCON.2)
TIMER0 Overflow	TF0 (TCON.5)	ET0 (IE.1)	000BH	PT0 (IP.1)	
TIMER1 Overflow	TF1 (TCON.7)	ET1 (IE.3)	001BH	PT1 (IP.3)	
PORT SERIE	RI ou TI (SCON 0 et 1)	ES (IE.4)	0023H	PS (IP.4)	
TIMER2 Overflow	TF2 (T2CON.7)	ET2 (IE.5)			
TIMER2 capture/reload	EXF2 (T2CON.6)	EXEN2(T2CON.3) ET2(IE.5)	002BH	PT2 (IP.5)	↓ sur T2EX(P1.1)
Validation globale de toutes les interruptions		EA (IE.7)			

**Figure II-10 :** Interruptions de 8051/8052**II-11-2-8) Les Timers :**

Le 8051 possèdent deux timers 16 bits TIMER0 et TIMER1 dont le fonctionnement est déterminé par les registres de configuration TMOD et TCON. Les signaux horloges de ces circuits proviennent soit de l'horloge interne du système (mode Timer) soit des entrées externes T0 (P3.4) et T1 (P3.5) (mode Compteur). Le débordement de ces timers positionne

les drapeaux TF0 ou TF1 qui déclenchent les interruptions correspondantes si elles ne sont pas masquées. Les deux Timers ont un fonctionnement identique. il sont accessibles par l'intermédiaire des registres 8 bits TLx et THx qui constituent leurs parties basse et haute [19].

SFR	Description	Adresse
TH0	Timer 0 High Byte	8Ch
TLO	Timer 0 Low Byte	8Ah
TH1	Timer 1 High Byte	8Dh
TL1	Timer 1 Low Byte	8Bh
TCON	Timer Control	88h
TMOD	Timer Mode	89h



TMOD (89h) SFR

Bit	Nom	description
7	GATE1	Quand ce bit est positionné, TIMER1 ne peut fonctionner que si l'entrée INT1 (P3.3) est haute. Quand il est à zéro, TIMER1 peut fonctionner indépendamment de l'entrée INT1
6	C/T1	Si bit est positionné, TIMER1 fonctionne en compteur d'événements sur l'entrée T1 (P3.5). Si ce bit est à zéro, TIMER1 est incrémenté par l'horloge système
5	T1M1	Timer1 mode (voir ci-dessous)
4	T1M0	Timer1 mode (voir ci-dessous)
3	GATE0	Quand ce bit est positionné, TIMER0 ne peut fonctionner que si l'entrée INT0 (P3.2) est haute. Quand il est à zéro, TIMER0 peut fonctionner indépendamment de l'entrée INT0
2	C/T0	Si bit est positionné, TIMER0 fonctionne en compteur d'événements sur l'entrée T0 (P3.4). Si ce bit est à zéro, TIMER0 est incrémenté par l'horloge système
1	T0M1	Timer0 mode (voir ci-dessous)
0	T0M0	Timer0 mode (voir ci-dessous)

TxM1	TxM0	Mode	Description
0	0	0	13-bit Timer.
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

**II-12) Conclusion :**

Le jeu d'instructions est l'ensemble des instructions machines qu'un processeur d'ordinateur peut exécuter. Ces instructions machines permettent d'effectuer des opérations élémentaires (addition, ET logique...) ou plus complexes (division, passage en mode basse consommation...). Le jeu d'instructions définit quelles sont les instructions supportées par le processeur. Le jeu d'instructions précise aussi quels sont les registres du processeur manipulables par le programmeur (les registres architecturaux).

# CHAPITRE III



### III-1) Introduction :

Afin de réaliser notre circuit électronique on utilise le logiciel « Proteus » est une suite logicielle destinée à l'électronique et développé par la société Labcenter Electronics, elle permet de dessiner des schémas électronique, de les stimuler et de réaliser des circuits imprimés les logiciels incluent dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES [20].

ISIS, permettant entre autres la création de schémas et la simulation électrique, et ARES, dédié à la création d'imprimés, ce logiciel est très important dans le domaine de l'électronique car elle nous permet de réaliser nos projets de manière rapide et efficace [12].

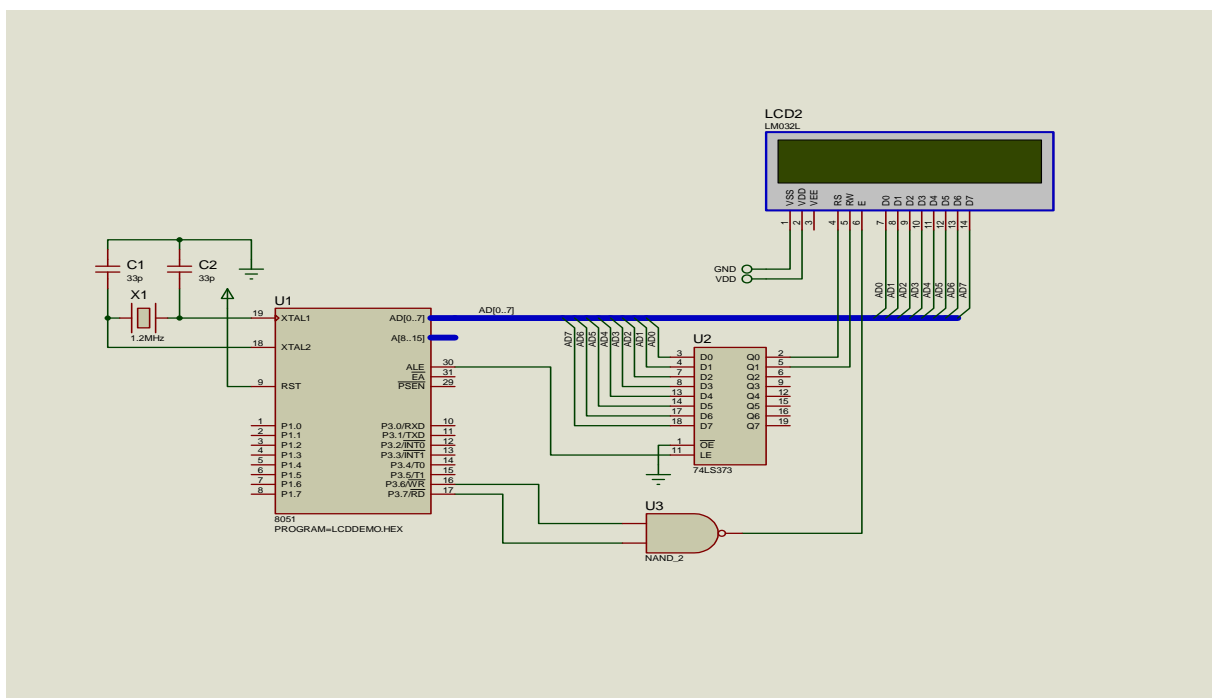


Figure III.1 : Circuit Réalisé Sous Proteus ISIS

### III-2) Présentation générale du proteus :

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels facile et rapide à comprendre et utiliser

- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet [21].

### **III-2-1) ISIS**

Le logiciel ISIS de Proteus (Intelligent schématique input system) est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits [21].

### **III-2-2) ARES**

Le logiciel ARES (Automatique Router Equipements System) est un outil d'édition et de routage qui complèterait parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement [21].

### **III-3) Liste de Matériel :**

- Afficheur graphique LCD.
- Microcontrôleur 8051.
- Quartz 1.2 MHZ.
- Deux capacités 33 P.
- Porte logique (NON ET).
- 74L373.

### III-4) principe de Fonctionnement de Chaque Etage :

#### III-4-1) Afficheur Graphique LCD :

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité [22].

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA) [22].

#### III-4-1-1) La Mise en œuvre d'un Afficheur LCD :

Un afficheur LCD sera commandé par un microcontrôleur.

Il faut donc penser aux mises en œuvre :

- **matérielle** : connexion des broches du microcontrôleur à l'afficheur LCD,
- **logicielle** : utilisation de sous programmes permettant de commander l'afficheur LCD (initialisation, effacement de l'afficheur, affichage d'un caractère, affichage d'une variable,...) ères.

Il existe de très nombreux types de LCD.

Un afficheur peut être constitué de :

- Segments (affichage de chiffres ou lettres)
- Pictogrammes, dessins, icônes
- Mots ou groupes de mots (ex : Low Batt)
- Points (affichage alphanumérique ou graphique) [22],

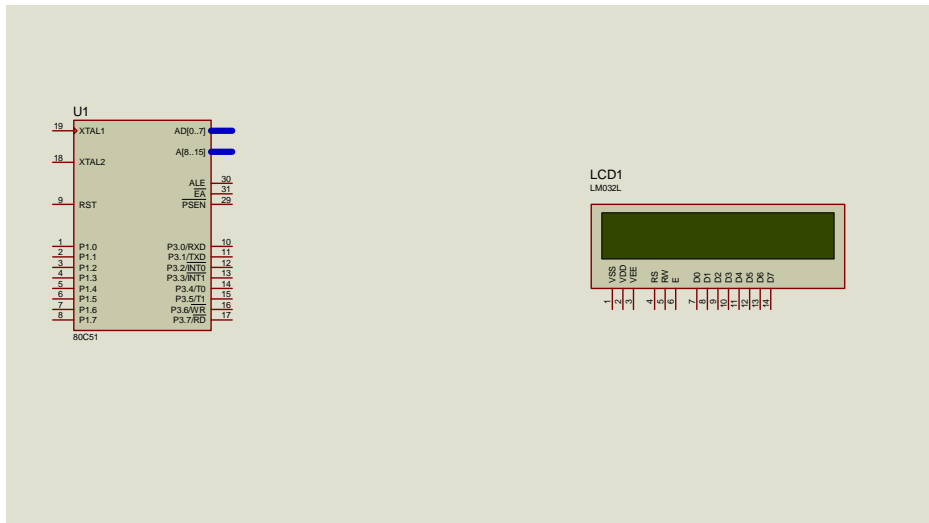


**Figure III-2** : Afficheur Graphique LCD

Un afficheur à cristaux liquides (**LCD**) est constitué de :

- 2 plaques transparentes finement gravées pour l'orientation des cristaux
- Électrodes conductrices transparentes déposées sur les plaques pour appliquer une tension
- Cristaux liquides.

Un exemple de réalisation technologique est donné plus loin [22].

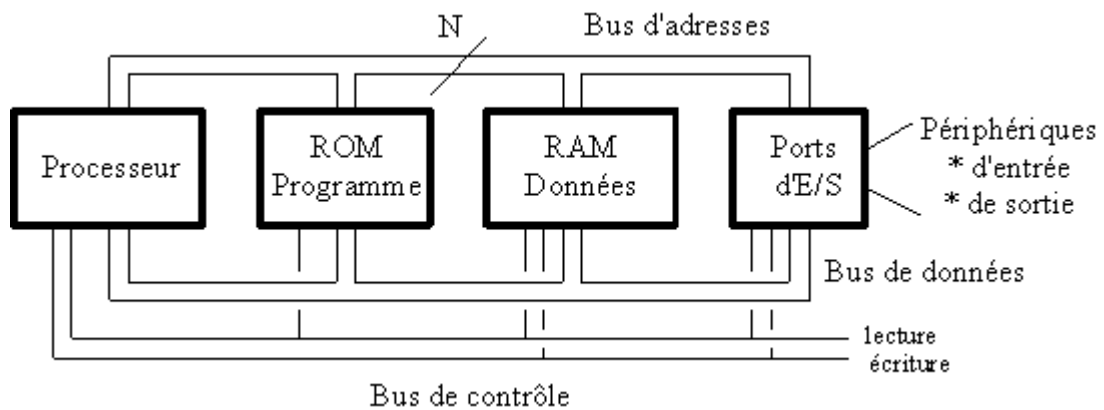


**Figure III-3** : Schéma de Principe d'un Afficheur LCD

### III-4-2) Microcontrôleur 8051 :

Intel 8051 ou 8051 est un microcontrôleur (MC) développé par Intel en 1980 pour être utilisé dans des produits embarqués. C'est encore une architecture populaire ; de nombreux microcontrôleurs plus récents incorporent un cœur 8051, complété par un certain nombre de circuits périphériques intégrés sur la même puce, et dotés de mémoires de plus grande capacité [23].

La structure d'un système à microprocesseur (MP) est indiquée dans la figure ci-après.



**Figure III-4** : Structure d'un Système à Microprocesseur

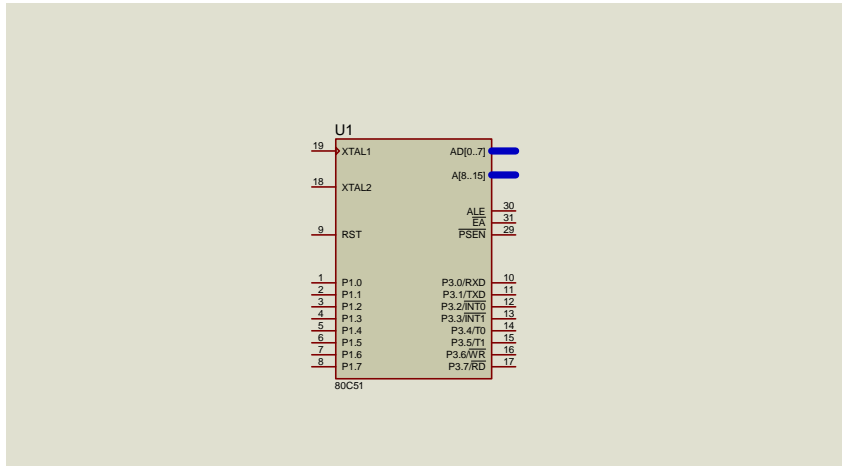
- Le MP, parfois appelé unité centrale, est le cœur du système, puisque c'est lui qui est chargé d'exécuter les instructions du programme.
- Le programme est inscrit dans la mémoire ROM du système ; il s'agit d'une mémoire non volatile, c'est-à-dire qui préserve les informations qu'elle contient même en l'absence d'alimentation électrique.

- La mémoire vive ou RAM sert à stocker des informations durant le fonctionnement du programme : informations en provenance de l'extérieur, données intermédiaires dans les calculs...
- Les ports d'entrées/sorties permettent la communication du système avec le monde extérieur ; les périphériques d'entrée seront par exemple des boutons-poussoirs, des interrupteurs, des commutateurs, des compteurs, des convertisseurs analogiques/numériques... ; les périphériques de sortie seront des témoins à leds, des relais, des moteurs, des convertisseurs numériques/analogiques... En général, un circuit d'interface de puissance est placé entre le port de sortie et le périphérique commandé [23].

Un microcontrôleur (MC) réunit ces différentes fonctions électroniques sur une même puce. Dans les paragraphes qui suivent, nous allons décrire les principales parties du microcontrôleur 8051. Le 8051 est un MC 8 bits car il traite des données sur 8 bits ; le bus de données comporte donc 8 lignes ; comme la plupart des MC 8 bits, le 8051 gère des adresses en 16 bits, ce qui donne un espace adressable de  $2^{16}$  octets soit 64 ko [23].



**Figure III-5** : Le Microcontrôleur 8051



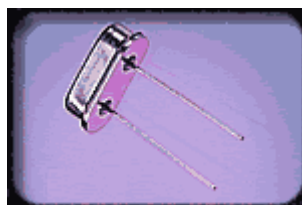
**Figure III.6 :** Schéma du Microcontrôleur 8051

### III-4-3) Quartz 1.2 MHZ :

En électronique, un quartz est un composant qui possède comme propriété utile d'osciller à une fréquence stable lorsqu'il est stimulé électriquement [24].

Les oscillateurs à quartz sont destinés à produire des signaux de fréquence de haute précision et de grande stabilité dans un intervalle de température donné et dans le temps. L'ordre de grandeur de la précision sur la fréquence d'oscillation avec un quartz courant de bonne qualité et dans des conditions déterminées (voir plus loin) est de +/- 30 ppm soit 0,003 % sans aucun ajustement [25].

Les propriétés piézoélectriques remarquables du minéral de quartz permettent d'obtenir des fréquences d'oscillation très précises, ce qui en font un élément important en électronique numérique ainsi qu'en électronique analogique [24].



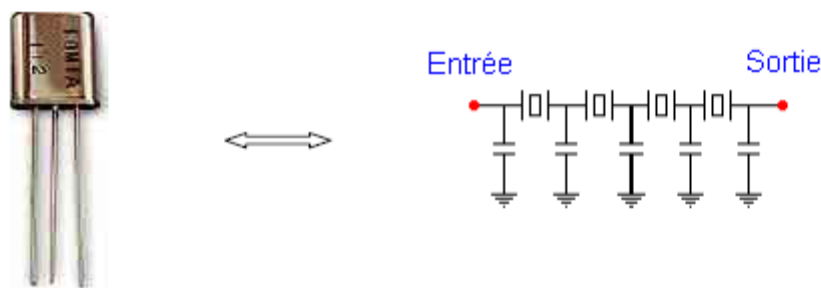
**Figure III-7 :** Quartz 1.2 MHZ

**III-4-3-1) Applications :**

Les quartz sont conçus pour vibrer à des fréquences allant de la dizaine de kilohertz, à quelques dizaines de mégahertz. La production mondiale de quartz électronique est de deux milliards ( $2 \times 10^9$ ) chaque année. La plupart sont destinés aux montres à quartz, et pour fournir une base de temps dans les circuits électroniques. On trouve des quartz dans les équipements de test et de mesure, tels que compteurs, générateurs de signaux à basses fréquences, oscillateurs à hautes fréquences ou oscilloscopes [24].

Les quartz sont également très utilisés dans les systèmes de radiocommunication, pour les références de fréquences, mais aussi pour réaliser des filtres de bande étroits [24].

Le quartz permet de réaliser des oscillateurs de précision et des filtres sélectifs [26].



**Figure III- 8** Composition d'un Filtre à Quartz

**Oscillateur a quartz :**

Voici les différents types d'oscillateurs à quartz :

- **ATCXO** : Analogue Temperature-compensated Crystal Oscillator
- **CDXO** : Calibrated Dual Crystal Oscillator
- **MCXO** : Microcomputer-compensated Crystal Oscillator
- **OCVCXO** : Oven-Controlled Voltage-Controlled Crystal Oscillator
- **OCXO** : Oven-Controlled Crystal Oscillator



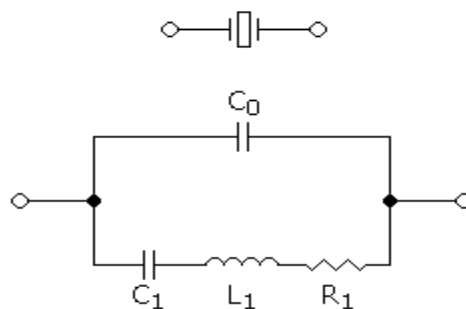
- **RbXO** : Rubidium Crystal Oscillator, un quartz qui peut être de la famille MCXO, synchronisé par une horloge au rubidium, et activée occasionnellement pour économiser l'énergie.
- **TCVCXO** : Temperature-Compensated Voltage-Controlled Crystal Oscillator
- **TCXO** : Temperature-Compensated Crystal Oscillator
- **TSXO** : Temperature-Sensing Sensing Crystal Oscillator, une variante du TCXO
- **VCXO** : Voltage-Controlled Crystal Oscillator
- **DTCXO** : Digital Temperature Compensated Crystal Oscillator, le même principe que le MCXO [24].

#### Filtre à quartz :

Les quartz permettent de réaliser des filtres avec un facteur de qualité élevé. Ce facteur peut devenir un inconvénient quand le quartz continue à vibrer alors qu'on ne le souhaite pas, car cela déforme le signal d'origine [24].

#### Capteur à quartz :

Certains capteurs exploitent la stabilité du quartz pour détecter l'évaporation de composés chimiques (microbalance à quartz). Le principe est de détecter la variation de fréquence du quartz en oscillation causée par le dépôt du composé en cours d'évaporation [24].



**Figure III-9** : Symbole et Circuit Equivalent d'un Quartz

### III-4-4) La capacité :

La capacité représente la quantité de charge électrique portée par un conducteur pour un potentiel électrique donné. Elle est alors définie comme étant la somme des charges électriques du conducteur divisée par le potentiel du conducteur [27] :

$$C = \frac{Q}{U}$$

La capacité peut être également exprimée à l'aide du flux électrique (voir théorème de Gauss) :

$$C = \epsilon_0 \frac{\Phi}{U}$$

Où

C est la capacité (en farads (F)) ;

Q est la charge (en coulombs (C)) ;

U est la différence de potentiels aux bornes de l'élément (en volts (V)) ;

$\epsilon_0$  La permittivité électrique du vide (qui vaut  $8,854 \times 10^{-12} \text{ F} \cdot \text{m}^{-1}$ ) ;

$\Phi$  Est le flux électrique associé à la charge Q [27].

Pour un condensateur, la capacité électrique est le rapport de la quantité de charge portée par l'armature positive sur la différence des potentiels entre les armatures [27].



**Figure III-10** Les Différents Types de Capacité.

**III-4-4-1) Caractéristiques :**

- stocke les charges électriques (en DC),
- s'oppose aux variations de tension (en AC)

- calcul de la réactance totale :

En // les capacités s'ajoutent, [28]

- en série

$$\frac{1}{C_{eq}} = \frac{1}{C_1} + \frac{1}{C_2}$$

**III-4-5) La Porte Logique NON-ET(NAND) :**

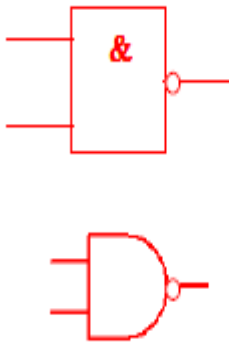
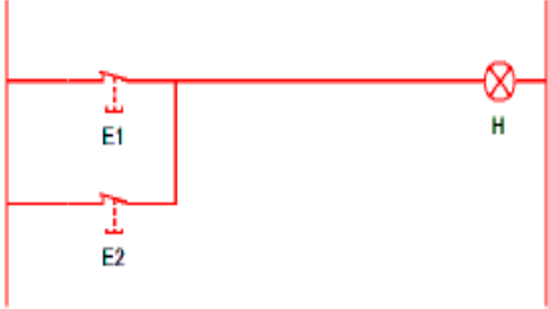
La fonction NON-ET (*NAND* en anglais) est un opérateur logique de l'algèbre de Boole. À deux opérandes, qui peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si au moins l'un des deux opérandes a la valeur FAUX [29].



**Figure III-11** :4 Portes NAND

III-4-5-1) Equation :

$$L = \overline{a \cdot b} = \bar{a} + \bar{b} \text{ [30]}$$

Fonction NON-ET (NAND)																	
Symbole logique : Européen Américain	Schéma électrique	Table de vérité															
		<table border="1"> <thead> <tr> <th>E1</th> <th>E2</th> <th>H</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	E1	E2	H	0	0	1	0	1	1	1	0	1	1	1	0
E1	E2	H															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
Équation logique																	
$H = \overline{E1.E2}$																	

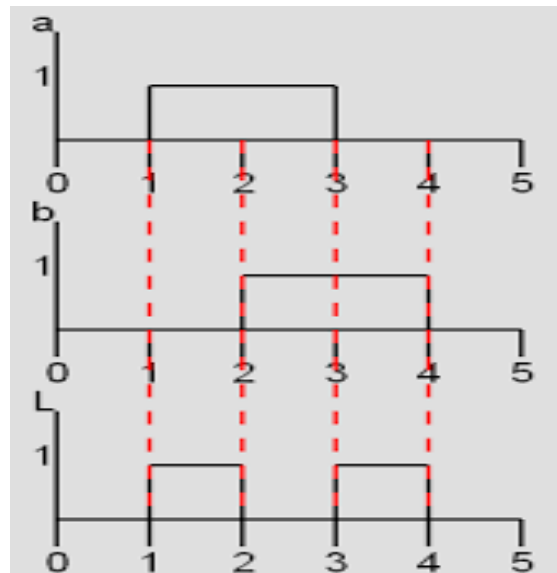


Figure III-12 : Chronogramme de la Fonction NON-ET.

**III-4-5-2) Fonction (ET-NON ou NAND) :**

L'obtention de la fonction nand3 se fait avec 2 variables au moins. Elle correspond à  $V_{14}$  du tableau des 16 fonctions à 2 variables [29].

C'est une fonction ET dont la sortie a été inversée.

Le théorème DE MORGAN permet encore d'écrire l'équation sous la forme suivante [30] :

$$H = \overline{E1} + \overline{E2}$$

**III-4-6) Circuit Intégré 74LS373 :**

Le SN54 / 74LS373 est un verrou transparent se compose de huit verrous avec les 3 sorties de l'état pour bus d'applications de système organisé. Il est un 20 broches IC comprend huit lignes d'entrée de données (D7-D0) et 8 ligne de sortie (O0-O7). Les bascules sont transparentes aux données [31].

Ces registres 8 bits comportent des sorties 3-state conçu spécifiquement pour la conduite hautement capacitive ou relativement chargés de faible impédance.

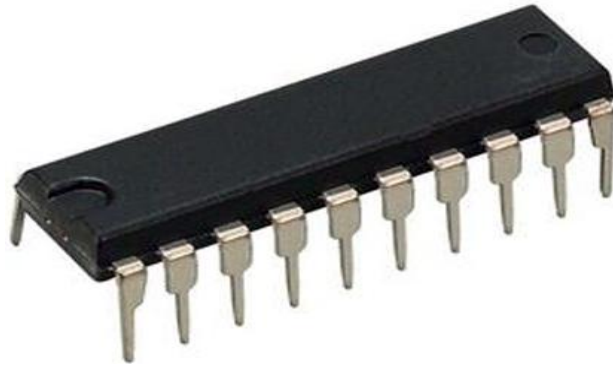
La Haute impédance 3 état et augmenté lecteur de niveau haut logique fournissent ces registres avec la capacité d'être connecté directement à et conduire les lignes de bus dans un système de bus organisé sans avoir besoin d'interface ou de pull up composants.

Ces dispositifs sont particulièrement attrayants pour, la mise en œuvre des registres tampons, les ports d'E / S, conducteurs de bus bidirectionnel et registres de travail.

Les huit loquets du ' LS373 et ' S373 sont transparentes bascules de type D, ce qui signifie que, si la validation (C ou CLK) d'entrée est élevée, les sorties Q suivent les données (D) entrées [32].

Lorsque C est prise ou CLK bas, la sortie est bloquée au niveau des données qui a été mis en place. Les huit bascules du ' LS374 et ' S374 sont D- bascules de bord déclenché.

Sur le positif la transition de l'horloge, les sorties Q sont réglées sur les états logiques qui ont été mis en place au niveau des entrées D [32].



**Figure III-13** : Le 74L373

### **III-5 Partie Software :**

#### **III-5-1 MikroC PRO pour 8051:**

On peut donc simuler dans ISIS un programme compilé en basic c ou pascal, Cofmaker est une passerelle entre les environnements de développement intégrés "mikroc mikropascal et mikro basic" [+++] et le logiciel de CAO "proteus». La construction d'un fichier cof Common Object file et son utilisation dans proteus ISIS sont automatisées | On peut donc simuler dans ISIS un programme compilé a en basic c ou pascal ; Cofmaker supporte le micro contrôleurs microcontrôleur 8051: Le logiciel peut produire un arbre d'appel des différents sous programmes Cofmaker est gratuit jusqu'à 2ko de code exécutable [33].

#### **III-5-2 programme en langage C :**

```
; LCD Display Driver Demo.  
; Timing code assumes 1.2MHz Clock  
  
;LCD Registers addresses  
LCD_CMD_WR equ 0  
LCD_DATA_WReq 1  
LCD_BUSY_RDequ 2  
LCD_DATA_RDequ 3  
  
;LCD Commands  
LCD_CLS equ 1  
LCD_HOME equ 2
```

```

LCD_SETMODE equ 4
LCD_SETVISIBLE equ 8
LCD_SHIFT equ 16
LCD_SETFUNCTION equ 32
LCD_SETCGADDR equ 64
LCD_SETDDADDR equ 128

;Reset vector
    org 0000h
    jmp start

;Start of the program
    org 0100h

string1a:db ' !! CENTRE UNIVERSITAIRE'
          db 0
string1b:db ' !! AIN TEMOUCHENT!! '
          db 0
string2:db ' BELAHADJ BOUCHAIBA '
          db 0

string3:db ' MASTER ELECTRONIQUE '
          db 0
string4:db ' BIOMIDICAL!'
          db 0

start:    mov A,#038h
          call wrcmd

loop:     mov A,#LCD_SETVISIBLE+6      ;Make the display & blink visible:
          call wrcmd

          mov R7,#2
loop2:   mov DPTR,#string1a
          call wrstr

          mov DPTR,#200
          call wtms

          mov A,#LCD_CLS                ;Clear screen
          call wrcmd

          mov DPTR,#string1b
          call wrstr

          mov DPTR,#200
          call wtms

          mov A,#LCD_CLS                ;Clear screen
          call wrcmd

          djnz R7,loop2

          mov DPTR,#string1a
          call wrstr

```

```

mov DPTR,#400
call wtms

mov A,#LCD_SETDDADDR+64
call wrcmd

mov DPTR,#string2
call wrslow

mov DPTR,#200
call wtms

mov A,#LCD_CLS ;Clear screen
call wrcmd

mov DPTR,#string3
call wrslow

    mov A,#LCD_SETDDADDR+64
    call wrcmd

mov DPTR,#string4
call wrslow

mov A,#LCD_SETVISIBLE+7 ;Show the blink cursor as well.
call wrcmd

mov DPTR,#2000
call wtms

mov A,#LCD_CLS ;Clear screen
call wrcmd

jmp loop

;Sub routine to write null terminated string at DPTR in program ram.
wrstr:    mov R0,#LCD_DATA_WR
wrstr1:   clr A
          movc A,@A+DPTR
          jz wrstr2
          movx @R0,A
          call wtbody
          inc DPTR
          push DPL
          push DPH
          pop DPH
          pop DPL
          jmp wrstr1
wrstr2:   ret

;Sub routine to write null terminated string at DPTR in program ram.
Slowly
wrslow:   mov R0,#LCD_DATA_WR
wrslw1:   clr A
          movc A,@A+DPTR
          jz wrslw2
          movx @R0,A

```



```

    call wtbody
    inc DPTR
    push DPL
    push DPH
        mov DPTR,#100
        call wtms
    pop DPH
    pop DPL
    jmp wrslw1
wrslw2:    ret

;Sub routine to write command:
wrcmd:    mov R0,#LCD_CMD_WR
        movx @R0,A
        jmp wtbody

;Sub routine to write character:
wrchar:   mov R0,#LCD_DATA_WR
        movx @R0,A

;Subroutine to wait for busy clear
wtbody:   mov R1,#LCD_BUSY_RD
        movx A,@r1
        jb ACC.7,wtbody
        ret

;Wait for number of seconds in A
wtsec:    push ACC
        call wtms
        pop ACC
        dec A
        jnz wtsec
        ret

;Wait for number of milliseconds in DPTR
wtms:    xrl DPL,#0FFh                ;Can't do DEC DPTR, so do the loop by
forming 2's complement
        xrl DPH,#0FFh                ;and incrementing instead.
        inc DPTR
wtms1:    mov TL0,#09Ch                ;100 ticks before overflow = 1ms
at 1.2MHz Clock
        mov TH0,#0FFh
        mov TMOD,#1                ;Timer 0 mode 1
        setb TCON.4                ;Timer 0 runs
wtms2:    jnb TCON.5,wtms2
        clr TCON.4                ;Timer 0 stops
        clr TCON.5
        inc DPTR
        mov A,DPL
        orl A,DPH
        jnz wtms1
        ret

END

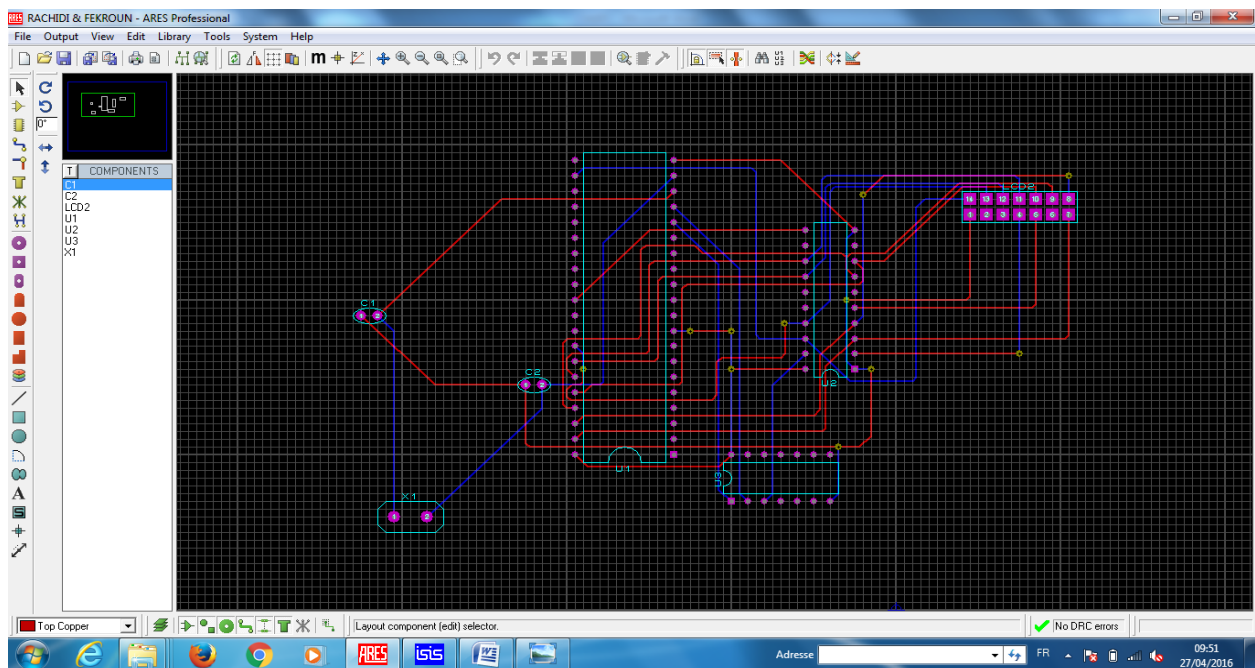
```

**III-5-3 Programme en HEXA:**

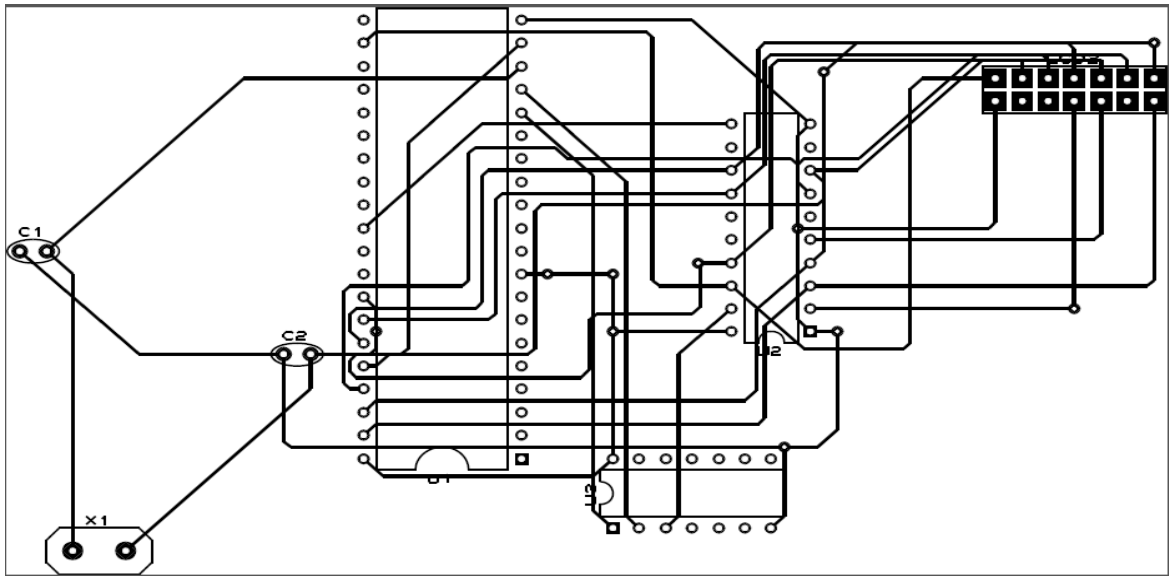
```
:0300000002018F6B
:100100002021212043454E54524520554E49564505
:10011000534954414952452041494E2054454D4F81
:10012000554348454E54202121200021212042459D
:100130004C4841444A4520424F554348414942209A
:1001400021212000204D415354455220454C454328
:1001500054524F4E495155452042494F4D4544490F
:1001600043414C2000204D4943524F434F4E54527F
:100170004F4C455552203830353120002020203852
:100180003035312070726F636573736F7221007444
:1001900038120236740E1202367F029001001202EB
:1001A000049000C8120251740112023690012B1201
:1001B00002049000C81202517401120236DFDC9072
:1001C000010012020490019012025174C012023612
:1001D00090014412021A9000C812025174011202D6
:1001E0003690016512021A74C012023690017C1218
:1001F000021A740F1202369007D0120251740112C3
:10020000023680907801E493600FF212023FA3C09F
:1002100082C083D083D08280ED227801E493601580
:10022000F212023FA3C082C083900064120251D038
:1002300083D08280E7227800F202023F7801F279CF
:1002400002E320E7FA22C0E0120251D0E01470F677
:10025000226382FF6383FFA3758A9C758CFF758977
:1002600001D28C308DFDC28CC28DA3E58245837096
:02027000E72283
:00000001FF
```

### III.6) Réalisation de Circuit Imprimé :

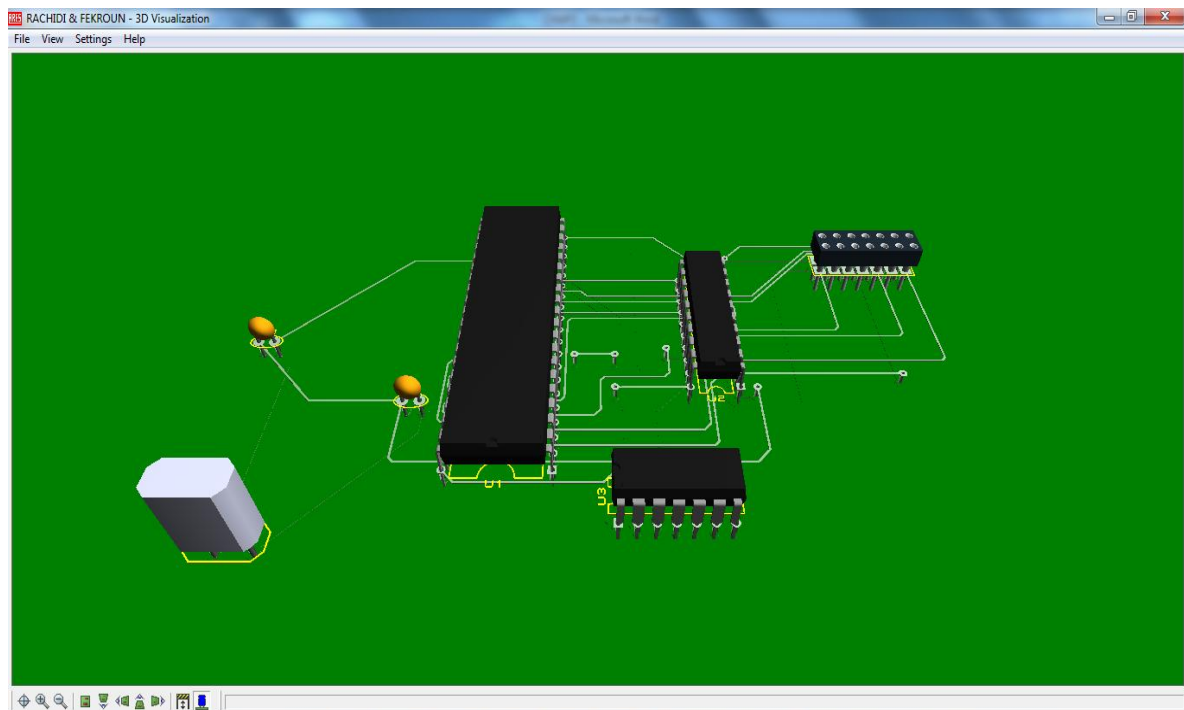
On utilisant la partie du logiciel ARES pour réaliser notre typon comme il est indiqué dans la Figure III-14 :



**Figure III-14** : Schéma du Circuit Sous ARES



**Figure III-15 :** Schéma Négatif du Circuit Sous ARES



**Figure III-16 :** Visualisation 3D du Circuit Sous ARES

**III-9) Conclusion :**

La réalisation de notre circuit nous permet de bien comprendre le logiciel Proteus qui est effectivement entièrement destiné à la conception et à la réalisation de circuit imprimé et de schéma électronique. Comme on signale l'utilisation du programmeur du microcontrôleur 8051 mikro C PRO pour 8051 qui nous a facilité le passage du programme C vers le programme en HEXA qui est nécessaire pour le fonctionnement du 8051.

# **CONCLUSION GENERAL**

### Conclusion Générale :

L'objectif de notre étude est de développer un circuit électronique pour contrôler un afficheur graphique LCD par le microcontrôleur 8051 sous environnement de Proteus qui peut afficher la légende suivante : CENTRE UNIVERSITAIRE BELHADJ BOUCHAIB AIN'TEMOUCHENT MASTER ELECTRONIQUE BIOMIDICALE ce qui nous a obligé de faire une étude bibliographique pour la conception de notre circuit et la programmation du microcontrôleur 8051.

A cet effet ; Nous avons consacré notre travail sur le savoir d'utiliser le logiciel Proteus qui nous a permis de bien construire notre maquette avec l'éditeur de schéma ISIS et le passage vers la conception du typon du circuit imprimé par l'outil ARES.

Au cours de cette étude nous avons fréquenté quelques problèmes lors de l'installation du logiciel Proteus et ceci suite à la non disponibilité de ce logiciel dans notre établissement.

Ce projet nous a permis d'élargir nos connaissances dans le domaine d'électronique et bien comprendre le fonctionnement des microcontrôleurs. On signale aussi le manque des composants (microcontrôleur 8051, décodeur BCD -> 7 segments, afficheurs à LED 7 segments) et quelques équipements et en particulier le programmeur assembleur dans notre centre ce qui nous a empêché et entravé pour passer nos bons résultats théoriques obtenus aux domaines de réalisation pratique, mais nous espérons que cette étude sera utile pour les prochains travaux et sera complétée à l'avenir par d'autres étudiants.

Enfin, nous estimons avoir fourni dans ce rapport les éléments nécessaires à la clarification de l'utilité et de l'élaboration de notre projet.

**ANNEXE**



verrou



## OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

The SN54/74LS373 consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE is LOW, the data that meets the setup times is latched. Data appears on the bus when the Output Enable (OE) is LOW. When OE is HIGH the bus output is in the high impedance state.

The SN54/74LS374 is a high-speed, low-power Octal D-type Flip-Flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. A buffered Clock (CP) and Output Enable (OE) is common to all flip-flops. The SN54/74LS374 is manufactured using advanced Low Power Schottky technology and is compatible with all Motorola TTL families.

- Eight Latches in a Single Package
- 3-State Outputs for Bus Interfacing
- Hysteresis on Latch Enable
- Edge-Triggered D-Type Inputs
- Buffered Positive Edge-Triggered Clock
- Hysteresis on Clock Input to Improve Noise Margin
- Input Clamp Diodes Limit High Speed Termination Effects

**PIN NAMES**

D <sub>0</sub> -D <sub>7</sub>	Data Inputs
LE	Latch Enable (Active HIGH) Input
CP	Clock (Active HIGH going edge) Input
OE	Output Enable (Active LOW) Input
O <sub>0</sub> -O <sub>7</sub>	Outputs (Note b)

**LOADING (Note a)**

	HIGH	LOW
D <sub>0</sub> -D <sub>7</sub>	0.5 U.L.	0.25 U.L.
LE	0.5 U.L.	0.25 U.L.
CP	0.5 U.L.	0.25 U.L.
OE	0.5 U.L.	0.25 U.L.
O <sub>0</sub> -O <sub>7</sub>	65 (25) U.L.	15 (7.5) U.L.

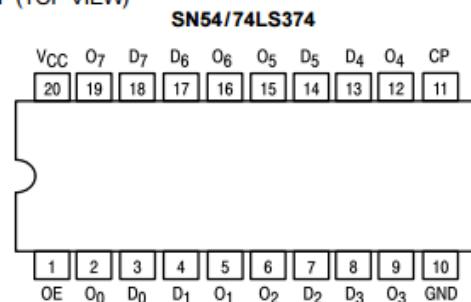
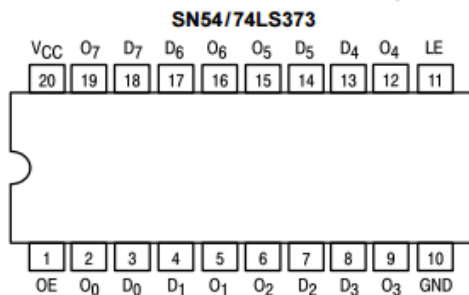
**NOTES:**

- a) 1 TTL Units Load (U.L.) = 40 μA HIGH/1.6 mA LOW.  
 b) The Output LOW drive factor is 7.5 U.L. for Military (54) and 25 U.L. for Commercial (74) Temperature Ranges. The Output HIGH drive factor is 25 U.L. for Military (54) and 65 U.L. for Commercial (74) Temperature Ranges.

**ORDERING INFORMATION**

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

**CONNECTION DIAGRAM DIP (TOP VIEW)**



**NOTE:**  
 The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

SN54/74LS373

AC WAVEFORMS

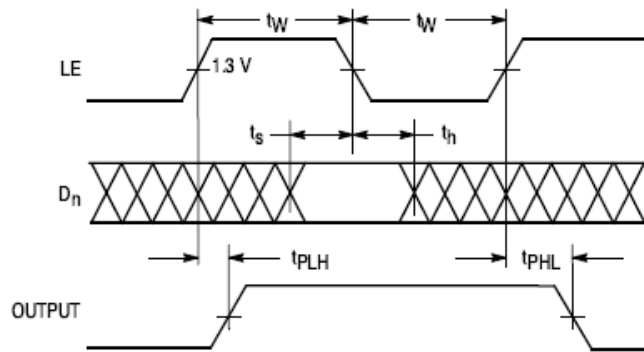


Figure 1

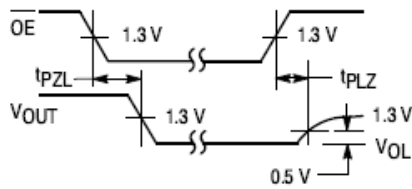


Figure 2

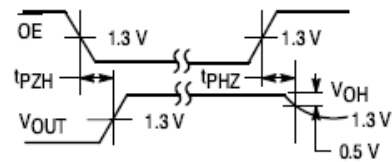


Figure 3

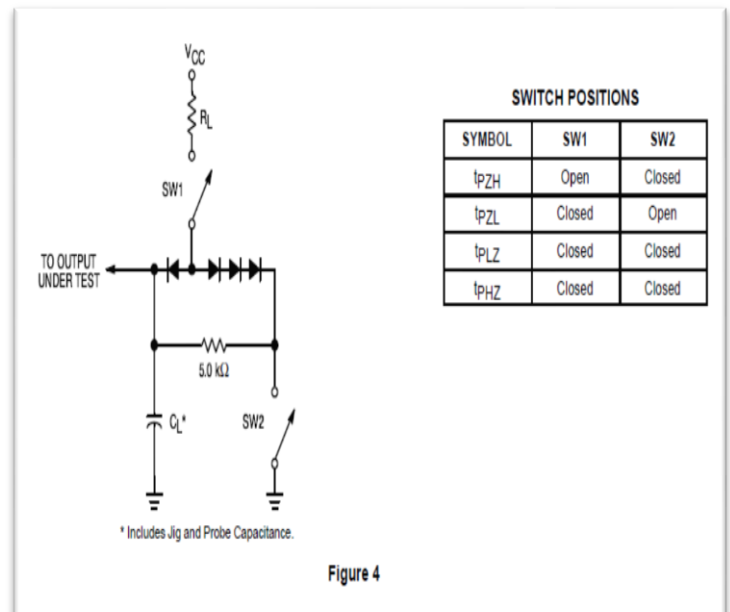
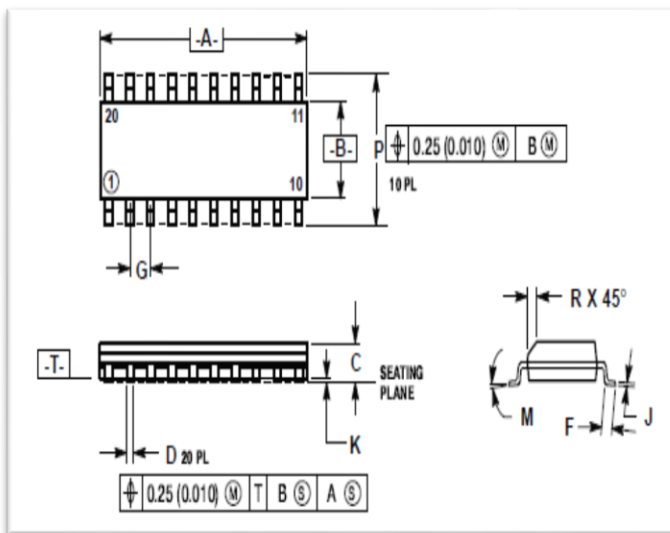


Figure 4



PRELIMINARY

## 87C51/80C51BH/80C31BH CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER

*Commercial/Express*

87C51/80C51BH/80C51BHP/80C31BH

\*See Table 1 for Proliferation Options

- |  |   |
|--|---|
| ■ High Performance CHMOS EPROM                   | ■ 5 Interrupt Sources                   |
| ■ 24 MHz Operation                               | ■ Programmable Serial Port              |
| ■ Improved Quick-Pulse Programming Algorithm     | ■ TTL- and CMOS-Compatible Logic Levels |
| ■ 3-Level Program Memory Lock                    | ■ 64K External Program Memory Space     |
| ■ Boolean Processor                              | ■ 64K External Data Memory Space        |
| ■ 128-Byte Data RAM                              | ■ ONCE Mode Facilitates System Testing  |
| ■ 32 Programmable I/O Lines                      | ■ Power Control Modes                   |
| ■ Two 16-Bit Timer/Counters                      | — Idle                                  |
| ■ Extended Temperature Range<br>(-40°C to +85°C) | — Power Down                            |

### MEMORY ORGANIZATION

**PROGRAM MEMORY:** Up to 4 Kbytes of the program memory can reside on-chip (except 80C31BH). In addition the device can address up to 64K of program memory external to the chip.

**DATA MEMORY:** This microcontroller has a 128 x 8 on-chip RAM. In addition it can address up to 64 Kbytes of external data memory.

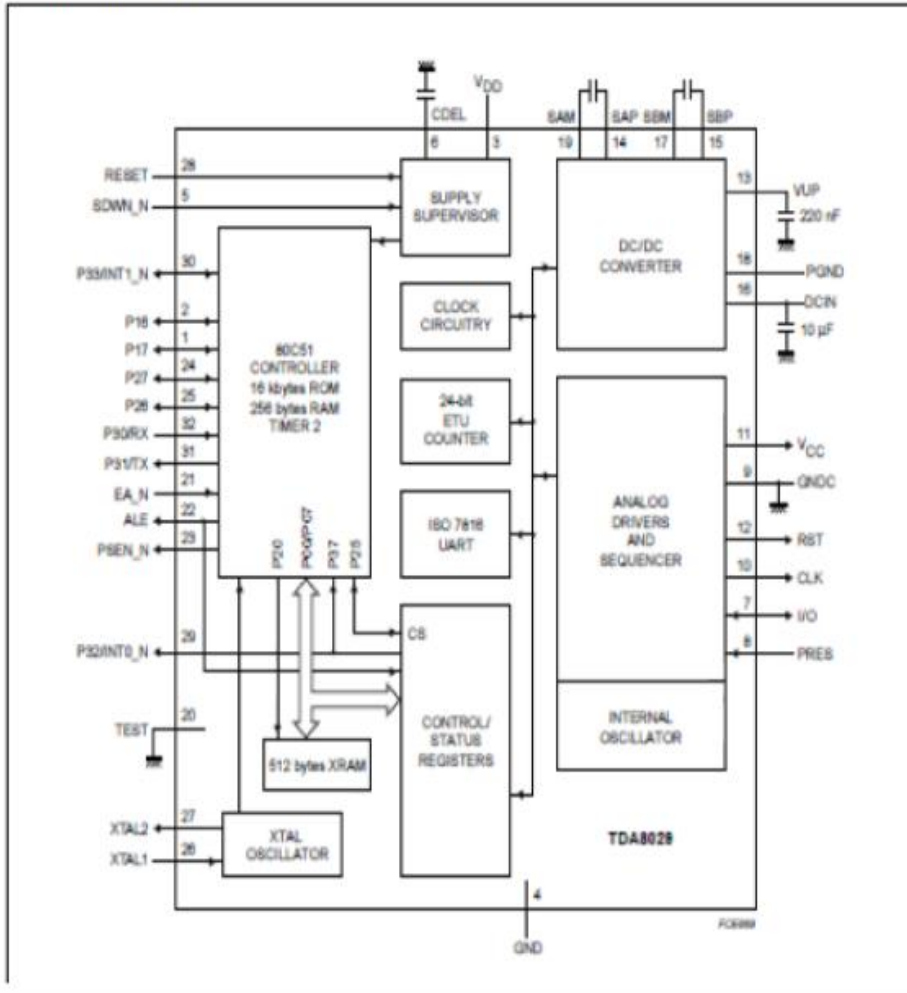
The Intel 87C51/80C51BH/80C31BH is a single-chip control-oriented microcontroller which is fabricated on Intel's reliable CHMOS II-E technology. Being a member of the MCS<sup>®</sup> 51 controller family, the 87C51/80C51BH/80C31BH uses the same powerful instruction set, has the same architecture, and is pin-for-pin compatible with the existing MCS 51 controller family of products.

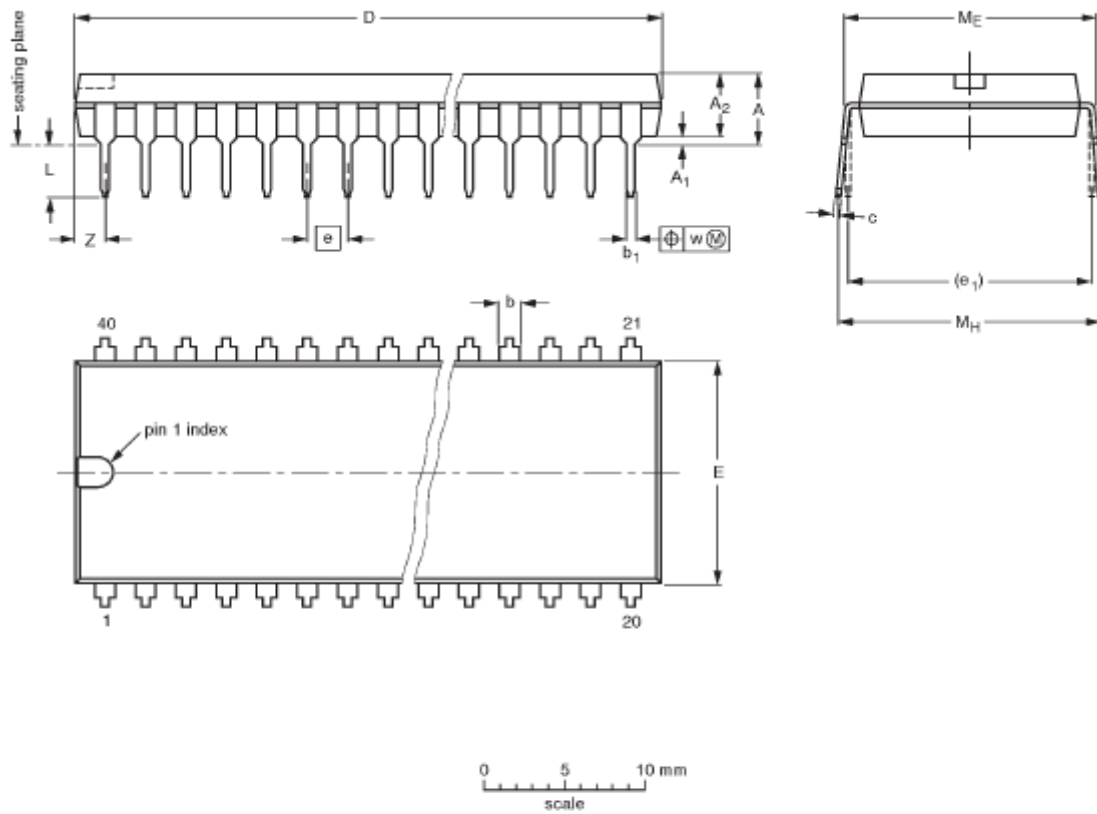
The 80C51BHP is identical to the 80C51BH. When ordering the 80C51BHP, customers must submit the 64 byte encryption table together with the ROM code. Lock bit 1 will be set to enable the internal ROM code protection and at the same time allows code verification.

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
TDA8029HL/C1	LQFP32	plastic low profile quad flat package; 32 leads; body 7 × 7 × 1.4 mm	SOT358-1
TDA8029HL/C2			

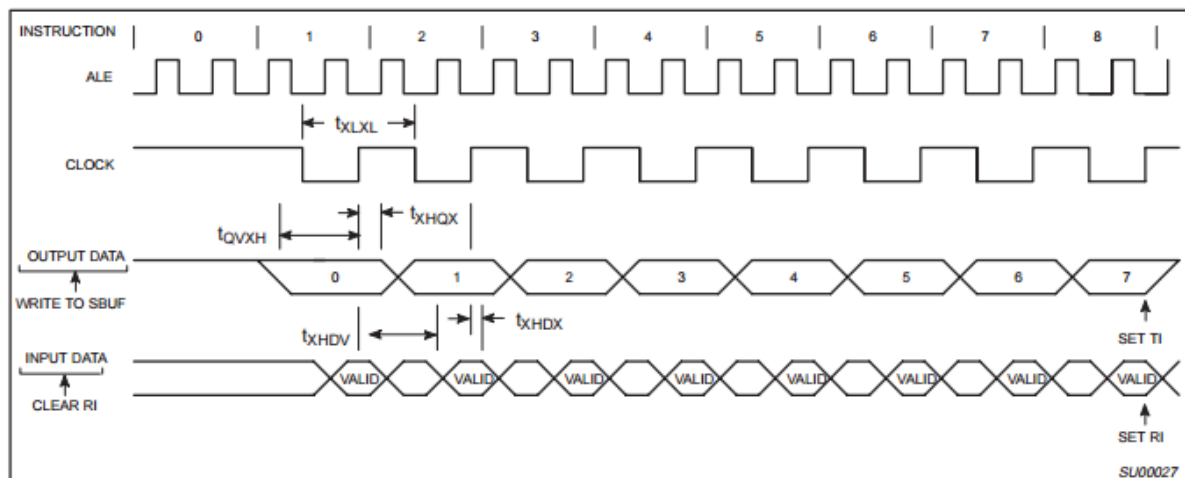
BLOCK DIAGRAM





DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A <sub>1</sub> min.	A <sub>2</sub> max.	b	b <sub>1</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	e <sub>1</sub>	L	M <sub>E</sub>	M <sub>H</sub>	w	Z <sup>(1)</sup> max.
mm	4.7	0.51	4.0	1.70 1.14	0.53 0.38	0.36 0.23	52.50 51.50	14.1 13.7	2.54	15.24	3.60 3.05	15.80 15.24	17.42 15.90	0.254	2.25
inches	0.19	0.020	0.16	0.067 0.045	0.021 0.015	0.014 0.009	2.067 2.028	0.56 0.54	0.10	0.60	0.14 0.12	0.62 0.60	0.69 0.63	0.01	0.089



## QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$V_{DD}$	supply voltage		2.7	–	6.0	V
$V_{DD(IN)}$	input voltage for the DC/DC converter		$V_{DD}$	–	6.0	V
$I_{DD(SH)}$	supply current in shut-down mode	$V_{DD} = 3.3$ V	–	–	20	$\mu$ A
$I_{DD(PD)}$	supply current in Power-down mode	$V_{DD} = 3.3$ V; card inactive; microcontroller in Power-down mode	–	–	110	$\mu$ A
$I_{DD(SL)}$	supply current in Sleep mode	$V_{DD} = 3.3$ V; card active at $V_{DD} = 5$ V; clock stopped; microcontroller in Power-down mode; $I_{CC} = 0$ $\mu$ A	–	–	675	$\mu$ A
$I_{DD(OP)}$	supply current in operating mode	$I_{CC} = 65$ mA; $f_{XTAL} = 20$ MHz; $f_{CLK} = 10$ MHz; 5 V card; $V_{DD} = 2.7$ V	–	–	260	mA
$V_{CC}$	card supply voltage	active mode including static loads; $I_{CC} < 65$ mA; 5 V card	4.75	5.0	5.25	V
		active mode; current pulses of 40 nAs with $I < 200$ mA, $t < 400$ ns, $f < 20$ MHz; 5 V card	4.6	–	5.4	V
		active mode including static loads; $I_{CC} < 65$ mA; $V_{DD} > 3.0$ V; 3 V card	2.78	3	3.22	V
		active mode; current pulses of 24 nAs with $I < 200$ mA, $t < 400$ ns, $f < 20$ MHz; 3 V card	2.75	–	3.25	V
		active mode including static loads; $I_{CC} < 30$ mA; 1.8 V card	1.62	1.8	1.98	V
		active mode; current pulses of 12 nAs with $I < 200$ mA, $t < 400$ ns, $f < 20$ MHz; 1.8 V card	1.62	–	1.98	V
$I_{CC}$	card supply current	5 V card; $V_{DD} = 0$ to 5 V	–	–	65	mA
		3 V card; $V_{DD} = 0$ to 3 V; $V_{DD} > 3.0$ V	–	–	65	mA
		1.8 V card; $V_{DD} = 0$ to 1.8 V	–	–	30	mA
$I_{CC(OV)}$	overload detection current		–	100	–	mA
$SR_r, SR_f$	rise and fall slew rate on $V_{CC}$	maximum load capacitor 300 nF	0.05	0.16	0.22	V/ $\mu$ s
$t_{de}$	deactivation sequence duration		–	–	100	$\mu$ s
$t_{ad}$	activation sequence duration		–	–	130	$\mu$ s
$f_{XTAL}$	crystal frequency	$V_{DD} = 5$ V	4	–	27	MHz
		$V_{DD} < 3$ V	4	–	16	MHz
$T_{AMB}$	ambient temperature		–40	–	+90	$^{\circ}$ C

# LM032L·LM032XMBL

- 20 character x 2 lines
- Controller LSI HD44780 is built-in (See page 79).
- +5V single power supply
- Display color: LM032L : Gray  
LM032XMBL : New-gray

### MECHANICAL DATA (Nominal dimensions)

Module size . . . . .116W x 39H (max.) x 10.5T (max.) mm  
 Effective display area . . . . . 83W x 18.6H mm  
 Character size (5 x 7 dots) . . . . . 3.2W x 4.85H mm  
 Character pitch . . . . . 3.7 mm  
 Dot size . . . . . 0.6W x 0.65H mm  
 Weight . . . . . about 50 g

### ABSOLUTE MAXIMUM RATINGS

	min.	max.
Power supply for logic ( $V_{DD}-V_{SS}$ )	0	6.5 V
Power supply for LCD drive ( $V_{DD}-V_O$ )	0	6.5 V
Input voltage ( $V_i$ )	$V_{SS}$	$V_{DD}$ V
Operating temperature ( $T_a$ )	0	50 40*°C
Storage temperature ( $T_{stg}$ )	-20	70 60*°C

\* Shows the value of type LM032XMBL.

### ELECTRICAL CHARACTERISTICS

$T_a = 25^\circ\text{C}$ ,  $V_{DD} = 5.0 \text{ V} \pm 0.25 \text{ V}$   
 Input "high" voltage ( $V_{IH}$ ) . . . . . 2.2 V min.  
 Input "low" voltage ( $V_{IL}$ ) . . . . . 0.6 V max.  
 Output "high" voltage ( $V_{OH}$ ) ( $-I_{OH} = 0.2 \text{ mA}$ ) . . . 2.4V min.  
 Output "low" voltage ( $V_{OL}$ ) ( $I_{OL} = 1.2 \text{ mA}$ ) . . . 0.4V max.  
 Power supply current ( $I_{DD}$ ) ( $V_{DD} = 5.0 \text{ V}$ ) . . . 2.0 mA typ.  
 . . . . . 3.0 mA max.

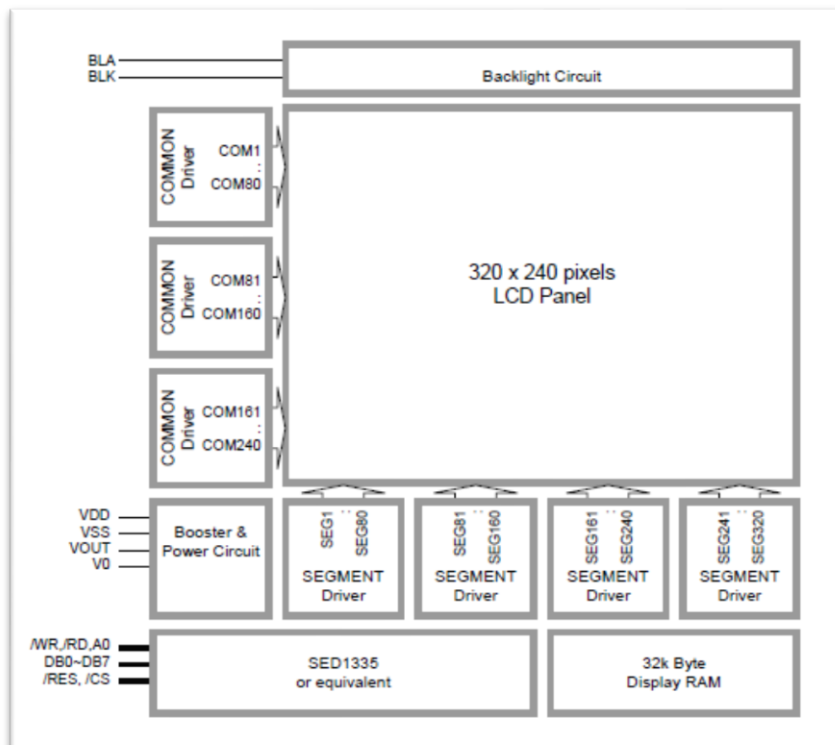
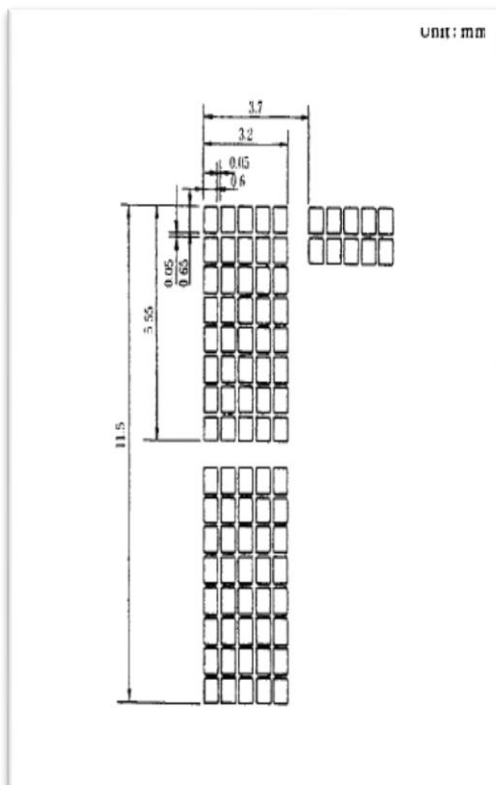
### INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function
1	$V_{SS}$	-	0V
2	$V_{DD}$	-	+5V
3	$V_O$	-	-
4	RS	H/L	L: Instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module→MPU) L: Data write (LCD module←MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	

### Notes:

In the HD44780, the data can be sent in either 4-bit 2-operation or 8-bit 1-operation so that it can interface to both 4 and 8 bit MPU's.

- (1) When interface data is 4 bits long, data is transferred using only 4 buses of  $DB_4 \sim DB_7$ , and  $DB_0 \sim DB_3$  are not used. Data transfer between the HD44780 and the MPU completes when 4-bit data is transferred twice. Data of the higher order 4 bits (contents of  $DB_4 \sim DB_7$  when interface data is 8 bits long) is transferred first and then lower order 4 bits (contents of  $DB_0 \sim DB_3$  when interface data is 8 bits long).
- (2) When interface data is 8 bits long, data is transferred using 8 data buses of  $DB_0 \sim DB_7$ .





## Terminal Functions

Pin No.	Pin Name	I/O	Descriptions
1	VSS	Power	0V Power Supply, GND
2	VDD	Power	Positive Power Supply
3	V0	Input	LCD Contrast Reference Input
4	/WR(R/W)	Input	Write enable input, active LOW
5	/RD(E)	Input	Read enable input, active LOW
6	/CS	Input	Chip Select Signal /CS=LOW: Data IO is enabled
7	A0	Input	Data Type Select A0=H: command write, display data or cursor add read A0=L: status flag read, display data or parameter write
8	/RES	Input	Reset Signal: /RES = L, Reset the LCD Module /RES = H, Normal Running
9	DB0	I/O	8-bit bi-directional data bus
:	:		
16	DB7		
17	NC	-	No connection (leave open)
18	VOOUT	Power	Power Booster Output for V0
19	BLA	Power	Positive Power Supply for LED backlight
20	BLK	Power	Negative Power Supply for LED backlight

## 2. Absolute Maximum Ratings

Items	Symbol	Min.	Max.	Unit	Condition
Supply Voltage	$V_{DD}$	0	+6.0	V	$V_{SS} = 0V$
Input Voltage	$V_{IN}$	$V_{SS}-0.3$	$V_{DD}+0.3$	V	$V_{SS} = 0V$
Operating Temperature	$T_{OP}$	0	+50	°C	No Condensation
Storage Temperature	$T_{ST}$	-20	+70	°C	No Condensation

## Cautions:

Any Stresses exceeding the Absolute Maximum Ratings may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

## 3. Electrical Characteristics

### 3.1 DC Characteristics

 $V_{SS}=0V, V_{DD}=5.0V, T_{OP}=25^{\circ}C$ 

Items	Symbol	MIN.	TYP.	MAX.	Unit	Applicable Pin
Operating Voltage	$V_{DD}$	4.5	5.0	5.5	V	VDD
Input High Voltage	$V_{IN}$	0.6xVDD	-	VDD	V	DB0~DB7, /WR, /RD, /CS, A0, /RES
Input Low Voltage	$V_{IN}$	VSS	-	0.15xVDD	V	DB0~DB7, /WR, /RD, /CS, A0, /RES
LCD Contrast Reference Voltage	$V_0$	-	-18.5	-	V	V0
Operating Current	$I_{DD}$	-	30.0	90.0	mA	VDD



## Jeux d'instructions de 8051 :

## Syntaxe:

Rn – registres R0-R7

@Ri – adressage indirect RAM interne ou externe par les registres R0 ou R1

#data 8 – adressage immédiat (8 ou 16 bits)

bit – 128 drapeaux généraux et bits des SFR

A – Accumulateur ACC

Mnemonic	Description	Byte	Cycle
<b>Logic Operations</b>			
ANL	A,Rn	AND register to accumulator	1 1
ANL	A,direct	AND direct byte to accumulator	2 1
ANL	A,@Ri	AND indirect RAM to accumulator	1 1
ANL	A,#data	AND immediate data to accumulator	2 1
ANL	direct,A	AND accumulator to direct byte	2 1
ANL	direct,#data	AND immediate data to direct byte	3 2
ORL	A,Rn	OR register to accumulator	1 1
ORL	A,direct	OR direct byte to accumulator	2 1
ORL	A,@Ri	OR indirect RAM to accumulator	1 1
ORL	A,#data	OR immediate data to accumulator	2 1
ORL	direct,A	OR accumulator to direct byte	2 1
ORL	direct,#data	OR immediate data to direct byte	3 2
XRL	A,Rn	Exclusive OR register to accumulator	1 1
XRL	A direct	Exclusive OR direct byte to accumulator	2 1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1 1
XRL	A,#data	Exclusive OR immediate data to accumulator	2 1
XRL	direct,A	Exclusive OR accumulator to direct byte	2 1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3 2
CLR	A	Clear accumulator	1 1
CPL	A	Complement accumulator	1 1
RL	A	Rotate accumulator left	1 1
RLC	A	Rotate accumulator left through carry	1 1
RR	A	Rotate accumulator right	1 1
RRC	A	Rotate accumulator right through carry	1 1
SWAP	A	Swap nibbles within the accumulator	1 1

Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@R1	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@R1	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@R1,A	Move accumulator to indirect RAM	1	1
MOV	@R1,direct	Move direct byte to indirect RAM	2	2
MOV	@R1,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC	A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX	A,@R1	Move external RAM (8-bit addr.) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX	@R1,A	Move A to external RAM (8-bit addr.)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@R1	Exchange indirect RAM with accumulator	1	1
XCHD	A,@R1	Exchange low-order nibble indir. RAM with A	1	1

# **BIBLIOGRAPHIE**

**BIBLIOGRAPHIE**

- [1]. [http : //www.panneaux-multilignes.fr/afficheurs-graphiques.ht](http://www.panneaux-multilignes.fr/afficheurs-graphiques.ht).
- [2]. [http : //www.google.fr/search?q=les+types+d%27afficheur+graphique](http://www.google.fr/search?q=les+types+d%27afficheur+graphique).
- [3].[https://www.google.dz/ ?gws\\_rd=Cr,ssl&ei= 3IGBVvHGOYfcO6 +LCD](https://www.google.dz/?gws_rd=Cr,ssl&ei=3IGBVvHGOYfcO6+LCD).
- [4]. [http : //www.gecif.net/articles/genie\\_electrique/logiciels/proteus/lcd/#6](http://www.gecif.net/articles/genie_electrique/logiciels/proteus/lcd/#6).
- [5]. <http://www.technologuepro.com/tags/afficheurlcd.html>.
- [6]. <http://pedagogie2.acreunion.fr/colBretagne/bouquin/afficheur.PDF>.
- [7]. [http://www.pobot.org/IMG/article\\_PDF\\_AfficheurLCD.pdf](http://www.pobot.org/IMG/article_PDF_AfficheurLCD.pdf).
- [8]. <http://www.google.dz>.
- [9]. <http://www.decitre.fr/livres/microcontrolleurs-8051-et-8052-9782100048526.html>.
- [10]. <https://fr.wikipedia.org/wiki/Microcontrôleur>.
- [11]. <http://web.mit.edu/6.115/www/document/8051>.
- [12]. Commande de trois moteurs par un microcontrôleur 8051mémoire(2008) de Melle BERRZIGA Madiha et Mr NEGMARI Rafik. Université de science et de la technologie Mohamed Boudiaf d’Oran.
- [13].<http://www.uvt.rnu.tn/resources-uvt/cours/microprocesseurs-microcontrolleurs/microcontrolleurs/pdf/chapitre4.pdf>.
- [14]. [http://www.lycee-ferry-versailles.fr/ssi/ci11/systemes\\_numeriques.pdf](http://www.lycee-ferry-versailles.fr/ssi/ci11/systemes_numeriques.pdf).
- [15]. <http://btselt.e-monsite.com/medias/files/presentation-du-pic-16f877.pdf>.
- [16]. [http://ssi.stjo.free.fr/content/fichiers/cours\\_Elec/C\\_Experimenter/Prog\\_%C2%B5c/microprocesseur.pdf](http://ssi.stjo.free.fr/content/fichiers/cours_Elec/C_Experimenter/Prog_%C2%B5c/microprocesseur.pdf).

- [17]. [http://www.alciro.org/alciro/microcontroladores-8051\\_24/tipos-instrucciones-microcontroladores-8051\\_284\\_fr.htm](http://www.alciro.org/alciro/microcontroladores-8051_24/tipos-instrucciones-microcontroladores-8051_284_fr.htm).
- [18]. [https://deptmedia.cnam.fr/new/spip.php?pdoc8995/Downloades/002\\_pipeline\\_1-2.pdf](https://deptmedia.cnam.fr/new/spip.php?pdoc8995/Downloades/002_pipeline_1-2.pdf).
- [19]. <http://www.oumnad.123.fr/Microcontrolleurs/Intel8051.pdf>.
- [20]. [https://fr.wikipedia.org/wiki/Proteus\\_\(%C3%A9lectronique\)](https://fr.wikipedia.org/wiki/Proteus_(%C3%A9lectronique)).
- [21]. <http://www.elektronique.fr/logiciels/proteus.php>.
- [22]. <https://www.aurel32.net/elec/lcd.php>.
- [23]. [https://fr.wikipedia.org/wiki/Intel\\_8051](https://fr.wikipedia.org/wiki/Intel_8051).
- [24]. [https://fr.wikipedia.org/wiki/Quartz\\_\(%C3%A9lectronique\)](https://fr.wikipedia.org/wiki/Quartz_(%C3%A9lectronique)).
- [25]. [http://sti.tice.ac-orleanstours.fr/spip2/IMG/pdf/Oscillateurs\\_a\\_quartz\\_et\\_resonateur\\_ceramique.pdf](http://sti.tice.ac-orleanstours.fr/spip2/IMG/pdf/Oscillateurs_a_quartz_et_resonateur_ceramique.pdf).
- [26]. [https://www.scei-concours.fr/tipe/TIPE\\_2002/sujets\\_2002/si\\_tsi\\_2002.pdf](https://www.scei-concours.fr/tipe/TIPE_2002/sujets_2002/si_tsi_2002.pdf) PDF DE QUARTZ/SI-TSI-2002 PDF.
- [27]. [https://fr.wikipedia.org/wiki/Capacit%C3%A9\\_%C3%A9lectrique](https://fr.wikipedia.org/wiki/Capacit%C3%A9_%C3%A9lectrique).
- [28]. [ecoinfo.cnrs.fr/IMG/pdf/Cours-electricite-V1-4.pdf](http://ecoinfo.cnrs.fr/IMG/pdf/Cours-electricite-V1-4.pdf).
- [29]. [https://fr.wikipedia.org/wiki/Fonction\\_NON-ET](https://fr.wikipedia.org/wiki/Fonction_NON-ET).
- [30]. <http://robert.cireddu.free.fr/SI/Cours%20sur%20les%20fonctions%20logiques.pdf>.
- [31]. <http://ecee.colorado.edu/~mcclurel/sn74ls373rev5.pdf>.
- [32]. <http://www.tme.eu/fr/Document/ba53759a6cfefeecee1741704488c7d81/SN74LS373DW.pdf>
- [33]. <http://telecharger.logitheque.com/mikroc.html>.

## RESUME

Les types d'écran sont basés sur l'utilisation de cristaux liquides, et la technologie mise en œuvre diffère considérablement pour les modèles. Ainsi, les écrans à cristaux liquides sont appelés ACL, ou en anglais, LCD (liquid cristal display.) Les techniques de la programmation des microcontrôleurs pour les applications dans la conception des circuits électroniques sont devenus actuellement très importantes afin d'atteindre rapidement les objectifs des cahiers de charges et d'assurer des bons résultats L'objectif de notre projet consiste à faire l'étude d'un circuit électronique à base de microcontrôleur 8051 pour contrôler un afficheur LCD ; La conception du circuit électronique, circuit imprimé et la visualisation 3D du projet doivent être effectuées sous environnement du logiciel Proteus ISIS et ARES.

**Mots clés :** LCD, microcontrôleur 8051, alimentation, Proteus.

## ملخص

تستند أنواع الشاشة على استخدام البلورات السائلة، و تنفيذ التكنولوجيا يختلف اختلافا كبيرا عن النماذج. وهكذا ، وتسمى شاشات الكريستال السائل AC L ، LCD الإنجليزية ( عرض الكريستال السائل ) . تقنيات برمجة للتطبيقات المتحكم في تصميم الدوائر الإلكترونية أصبحت الآن مهمة جدا للوصول بسرعة لتحديد الأهداف و ضمانا للحصول على نتائج جيدة. والهدف من مشروعنا هو دراسة الدوائر الإلكترونية على أساس 8051 متحكم للسيطرة على شاشة LCD . يجب أن يتم تصميم الدوائر الإلكترونية دائرة كهربائية المطبوعة والتصوير ثلاثي الأبعاد للمشروع في ظل البيئة المتقلبة ISIS والبرمجيات ARES.

**كلمات البحث:** شاشات الكريستال السائل ، 8051 متحكم ، والطاقة ، بروتينوس

## ABSTRACT

Screen types are based on the use of liquid crystals, and implementing technology significantly different for the models. Thus, liquid crystal displays are called ACL in French, or English LCD (liquid crystal display). The techniques of programming microcontrollers for applications in the design of electronic circuits have become now very important to quickly reach the specification of objectives and ensure good results. The aim of our project is to study an electronic circuit based on 8051 microcontroller to control an LCD display; the design of the electronic circuit, PCB and 3D visualization of the project must be carried out under the environment Proteus ISIS and ARES software.

**Keywords:** LCD, 8051 microcontroller, power, Proteus.