

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Année : 2019

Faculté : Sciences de l'Ingénierat
Département : Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :

Détection du complexe QRS par la Méthode de
Laguna

Domaine : Sciences et Technologie

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Par :

COULIBALY Mohamed

DEVANT Le JURY

Président : M. BOUGAZI	Grade	UBM Annaba
Directeur de mémoire : BOUSBIA SALAH Mounir	Pr	UBM Annaba
Examineurs : T. Hafs	Grade	UBM Annaba
K. Boukhari	Grade	UBM Annaba

DÉDICASES

A la pensée de ma mère, mon père et tous leurs

Sacrifices, leur tendresse, leur soutien et leurs

Prières tout au long de nos vies,

A mes chères frères, chihab el dine, soundouse et ranim pour leur appui et leur

encouragement, et leur soutien morale ;

Que ce travail soit l'accomplissement de vos vœux tant allégués pour votre
soutien

Infaillible

Merci d'être toujours là pour moi

A tous le nombre de la section d'électronique des systèmes embarqué 2019, et a
mes amis

A tous ceux qui me sont chers et proches,

A tous ceux qui ont semé en moi à tout point de vue,

A tous(te) les proches et les ami(e)s que le destin a arrachés à la vie.

REMERCIEMENT

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

En second lieu, je tiens à remercier mon encadreur MADAME N.ZERMI et son précieux conseil et son aide durant toute la période du travail.

Nous remercions également :

Mr le professeur M.ZADAM de l'université de BADJI MOKHTAR ANNABA qui nous a fait le grand honneur de présider notre jury de mémoire. Qu'il trouve ici l'expression de notre gratitude pour l'intérêt qu'il a accordé à nos travaux.

Mr N.DOGHMANE professeur de l'université de de BADJI MOKHTAR ANNABA ainsi qu'au Mr F.Amara MCB de l'université de de BADJI MOKHTAR ANNABA, d'avoir acceptés de participer à mon jury comme examinateurs.

Je remercie, tous les enseignants de l'université de BADJI MOKHTAR ANNABA, pour avoir bien voulu me faire profiter pleinement de leurs compétences scientifiques et encore de leurs idées illuminatives.

Je remercie également mes collègues de la promo 2019.

Liste des abréviations

SRAM: (static read access memory) ,mémoire vive.

EEPROM: Electrically Erasable Programmable Read Only Memory (mémoire morte destinée aux données).

PWM: Pulse Width Modulation.

Proteus Professional: logicielle destinée à l'électronique.

CAO : Construction Assistée par Ordinateur.

ISIS : logiciel permet également de simuler ces schémas.

ARES : Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS.

(MAX7219/MAX7221) : c'est un microprocesseur ou contrôleur des afficheurs.

Liste des abréviations

SRAM: (static read access memory) ,mémoire vive.

EEPROM: Electrically Erasable Programmable Read Only Memory (mémoire morte destinée aux données).

PWM: Pulse Width Modulation.

Proteus Professional: logicielle destinée à l'électronique.

CAO : Construction Assistée par Ordinateur.

ISIS : logiciel permet également de simuler ces schémas.

ARES : Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS.

(MAX7219/MAX7221) : c'est un microprocesseur ou contrôleur des afficheurs.

Introduction

Chapitre I

I-Introduction sur l'Arduino :	2
I.1 Présentation de la carte	3
I.1.a Qu'est-ce qu'un microcontrôleur	3
I.1.b Caractéristiques techniques de l'Arduino UNO	3
I.2 Présentation du logiciel	4
I.2.a IDEArduino :	4
I.2.b Langage Arduino	5
I.3 Fonctionnalité de base	6
I.3.a Les entrées/sorties	6
I.4 Quelques bibliothèques	8
I.5 Présentation générale sur Proteus Professional:	8
I.5.1 ISIS :	9
I.5.2 ARES :	9
I.6 Gestion d'un projet et création d'un nouveau projet :	11
I.7 Edition d'un schéma	12
I.7.1 Ajout d'un composant	12
I.7.1.a Chargement d'un composant	12
I.8.2 DEBUT DE CREATION DU CIRCUIT IMPRIME :	14
8.2.a UN COMPOSANT	14
8.2.b UNE PISTE	14
8.2.c CREATION D'UN TYPON	14
I.8.3 SELECTION DES EMPREINTES A UTILISER :	14
8.4.a placement automatique des empreinte	15
8.4.b placement manuel des empreinte	15
I.8.5. ROUTAGE	15
I.9 Conclusion	16

Chapitre II:

II. LES AFFICHEUR 7 SEGMENTS	17
II.1. Définition d'un afficheur à 7 segments	17
II.2. Des LED :	18
II.3. Cathode commune ou Anode commune :	19
II.4. Choix de l'afficheur :	19
II.5. Présentation du boîtier :	19

II.6.Problématiques liée au score board avec l'arduinouno :	21
II.7.le contrôleur d'affichage max (7219/7222) :	21
II.7.a- description Générale.....	21
II.7.b. la configuration des pins ou des broches du max7219 :.....	22
II.8 Conclusion :	22
Chapitre III:	
III.1 Objectif du travail :	23
III.2 les avantages et les inconvénients des choix ou les prototypes proposer dans ce projet : 24	
III.2.1.a les avantages du premier prototype :	24
III.2.1.b les inconvénients du premier prototype :.....	24
III.2.2.a les avantages du deuxième prototype :	24
III.2.2.b les inconvénients du deuxième prototype :.....	24
III.3 La conception pratique :	24
III.4 Simulationdu système et test du programme :	24
III.4.2. b- Système de l'horloge du score board :.....	25
III.6.les photos de système de pointage prototype 1:	26
Conclusion :.....	27

I-Introduction sur l'Arduino :

Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques Pour l'informatique embarquée Ces cartes permettent un accès simple et peu couteux. De Plus, elles sont entièrement libres de droit, autant sur l'aspect du code source (Open Source) Que sur l'aspect matériel (Open Hardware). Ainsi, il est possible de refaire sa propre carte Arduino dans le but de l'améliorer ou d'enlever des fonctionnalités inutiles au projet. Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique Embarquée de par sa simplicité. En effet, beaucoup de librairies et de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée afin de gagner en Simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications dans le cadre de hobby. Les possibilités des cartes Arduino sont énormes, un grand nombre d'application ont déjà été réalisée et testées par bon nombre d'internautes. On retrouve par exemple diverse Forme de robot (**voir Fig.1**), des stations météo (**voir Fig.2**). [1]



Figure1 : robot arduino



Figure 2 : station météo (contrôler par Arduino)

I.1) Présentation de la carte

I.1.a) Qu'est-ce qu'un microcontrôleur

Les cartes Arduino font partie de la famille des microcontrôleurs. Un microcontrôleur est une petite unité de calcul accompagné de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement. Parmi les périphériques, on recense généralement des Timers, des convertisseurs analogique-numérique, des liaisons Séries, etc. On peut comparer un micro contrôleur à un ordinateur classique, mais pour système d'exploitation et avec une puissance de calcul considérablement plus faible. Les microcontrôleurs sont inévitables dans les domaines de l'informatique embarquée, De l'automatique et de l'informatique industrielle. Ils permettent de réduire le nombre de Composant et de simplifier la création de cartes électroniques logiques.

I.1.b) Caractéristiques techniques de l'Arduino UNO

Un des modèles les plus répandu de carte Arduino est l'Arduino UNO (**voir Fig.3**), [3] C'est La première version stable de carte Arduino. Elle possède toutes les fonctionnalités d'un Microcontrôleur classique en plus de sa simplicité d'utilisation. Elle utilise une puce ATmega328P cadencée à 16Mhz. Elle possède 32ko de Mémoire flash destinée à recevoir le programme, 2ko de SRAM (mémoire vive) et 1 ko D'EEPROM (mémoire morte destinée aux données). Elle offre 14 pins (broches) D'entrée/sortie numérique (données acceptée 0 ou 1) dont 6 pouvant générer des PWM (Pulse Width Modulation, détaillé plus tard). Elle permet aussi de mesurer des grandeurs Analogiques grâce à ces 6 entrées analogiques. Chaque broche est capable de délivrer un Courant de 40mA pour une tension de 5V Cette carte Arduino peut aussi s'alimenter et communiquer avec un ordinateur grâce à Son port USB. On peut aussi

l'alimenter avec unes alimentations comprise en 7V et 12V Grâce à sa connecteur Power

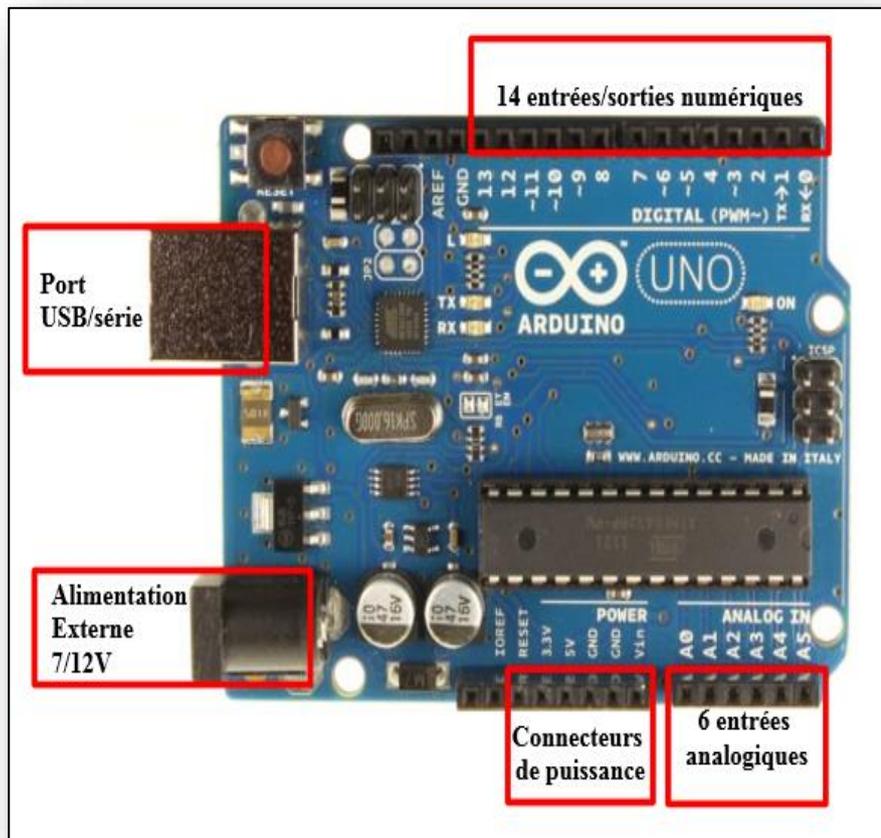


Figure 3:Arduino uno

I.2) Présentation du logiciel

I.2.a) IDE Arduino :

UN IDE (environnement de développement) libre et gratuit est distribué sur le site D'Arduino(compatible.Windows,.Linuxet.Mac).à.l'adresse'),[3],[4]. D'autres alternatives existent pour développer pour Arduino (extensions pour CodeBlocks, Visual Studio, Eclipse, XCode, etc.) mais nous N'aborderons dans ce modeste travail que l'IDE officiel. L'interface de l'IDE Arduino est plutôt simple (**voir Fig.4**), il offre une interface Minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un Éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les Deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus plus classique qui est utilisé pour accéder aux Fonctions avancées de l'IDE. Enfin, une console

affichant les résultats de la compilation Du code source, des opérations sur la carte, etc.

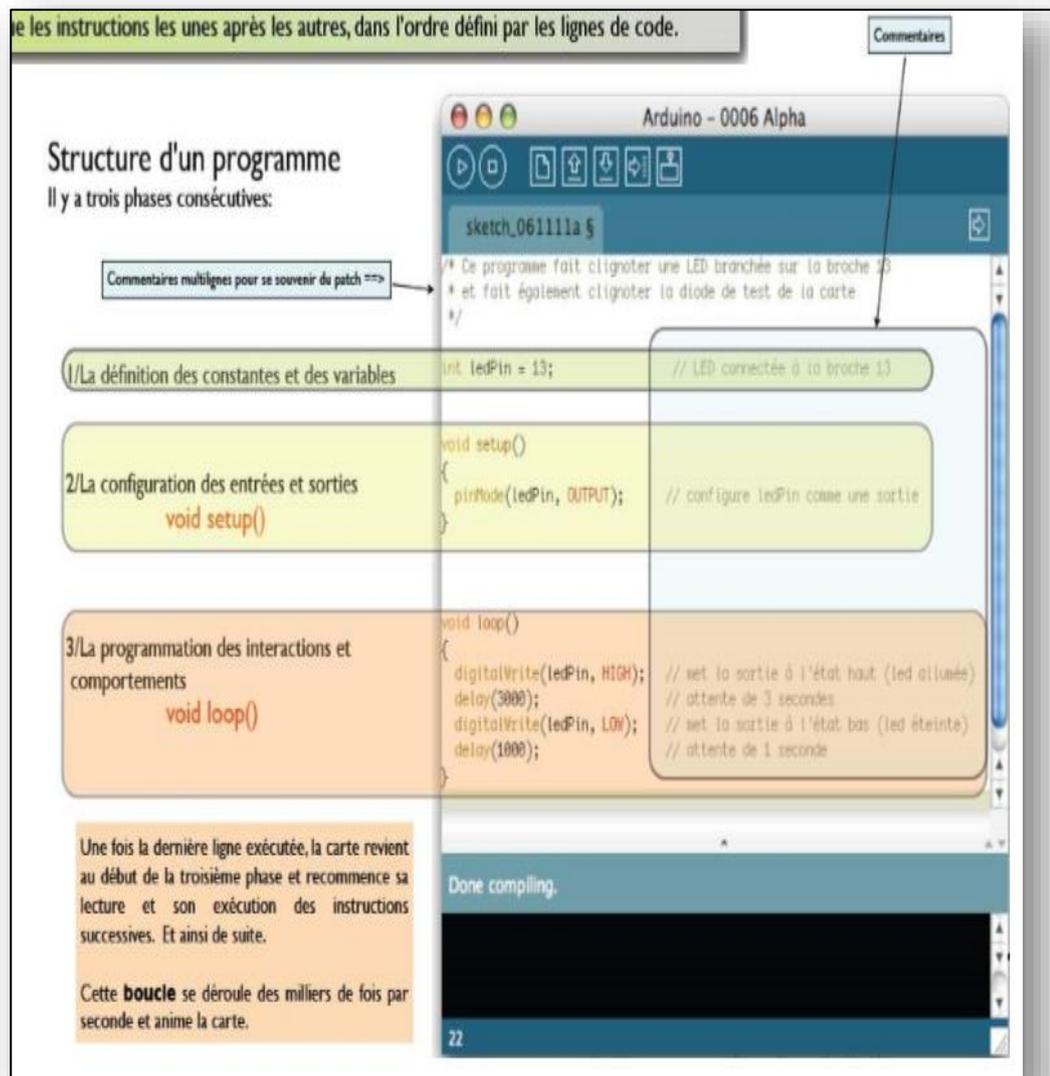


Figure4 : IDE Arduino

I.2.b Langage Arduino

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des Similarités avec le C, le C++, le Java et le Processing. Le langage impose une structure Particulière typique de l'informatique embarquée. La fonction setup contiendra toutes les Opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de Communications série, etc.). La fonction loop elle, est exécutée en boucle après l'exécution De la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset). Cette boucle est absolument nécessaire sur les

Microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation. En effet, si l'on Omettait cette boucle, à la fin du code produit, il sera impossible de reprendre la main sur la Carte Arduino qui exécuterait alors du code aléatoire. Au niveau de la syntaxe, on retrouve des similarités avec les langages précédemment Cités. La déclaration des variables se fait généralement dans l'espace global (de façon à Partager les variables les plus importantes entre les deux fonctions principales). On retrouve Les types de base suivant : Il existe d'autres types de base mais ils ne sont qu'un alias de ceux cités Précédemment.

I.3) Fonctionnalité de base

I.3.a) Les entrées/sorties

Le langage Arduino vient avec un nombre important de fonction de base permettant d'interagir avec son environnement. Les fonctions les plus utilisée sont les fonctions D'entrée/sorties. Ce sont elles qui permettent d'envoyer ou de mesurer une tension sur une Des broches de la carte. Dans un premier temps, avant d'effectuer une mesure ou d'envoyer une commande. Il Est nécessaire de définir la direction des broches utilisées. Pour cela on fait appel à la Fonction pinMode en lui donnant d'une part, la broche concernée, et d'autre part, la Direction :voidsetup(){

```
pinMode(1,OUTPUT);// Broche 1 en sortie
```

```
pinMode(2,INPUT) ; // Broche 2 en entrée
```

```
}
```

Une fois cette configuration faite, on peut procéder à l'utilisation des broches. ToutesLes broches sont capables d'écrire et de lire des données numériques (c'est-à-dire des 0 (0V) Ou des 1 (5V)). Mais, certaines disposent de fonctionnalité supplémentaire. Tout d'abord, toutes les cartes Arduino possèdent des entrées analogiques. Ce sont Les broches A0-A1-A2 etc. Elles permettent de lire des tensions analogiques (comprise entre 0 et 5V) et de le convertir en entier (compris entre 0 et 1023) proportionnellement à la Tension mesurée. Certaines cartes Arduino possède des sorties analogique faisant L'opération inverse (met une tension sur la broche proportionnellement à l'entier donné), Mais ce n'est pas le cas pour l'Arduino UNO. Pour pouvoir tout de même contrôler des composants autrement qu'en « tout ou rien » il est possible d'utiliser des broches PWM. Ce sont les broches annotés par un tilde ~ sur la carte. Les PWM (Pulse Width Modulation) sont utilisées pour synthétiser des signaux analogiques

en modulant le temps passé à l'état 1 (5V). Le signal obtenu est représenté figure 5. En utilisant une fréquence relativement élevée, les PWM permettent de commander certains composants comme si il recevait une tension analogique. Cela provient du fait que les composants utilisés dans l'électronique analogique, ne changes pas d'états instantanément. Par exemple, une ampoule à incandescence reste chaude et éclaire un court instant après avoir été éteinte. Ce phénomène est généralement invisible à l'œil nu. Grâce à elles, on pourra par exemple faire varier l'intensité d'une LED. La plupart des cartes Arduino utilisent des PWM cadencées à 490Hz environ. Toutes ces fonctionnalités sur les broches d'entrées sorties sont utilisables par le biais de quatre fonctions :

1-digitalRead(pin) : mesure une donnée numérique sur une des broches, la broche en question doit être réglée en entrée.

2-digitalWrite(pin, value) : écrit une donnée numérique sur une des broches, la broche concernée doit être réglée en sortie. Le paramètre value doit être égal à HIGH (état 1 soit 5V) ou LOW (état 0 soit 0V).

3-analogRead(pin) : mesure une donnée analogique sur une des broches (compatible seulement), la broche doit être réglée sur entrée.

4- analogWrite(pin, value) : écrit une donnée sous forme de PWM sur une des broches (Compatible uniquement), la broche doit être réglée en sortie. Le paramètre valu doit être compris dans l'intervalle [0;255]

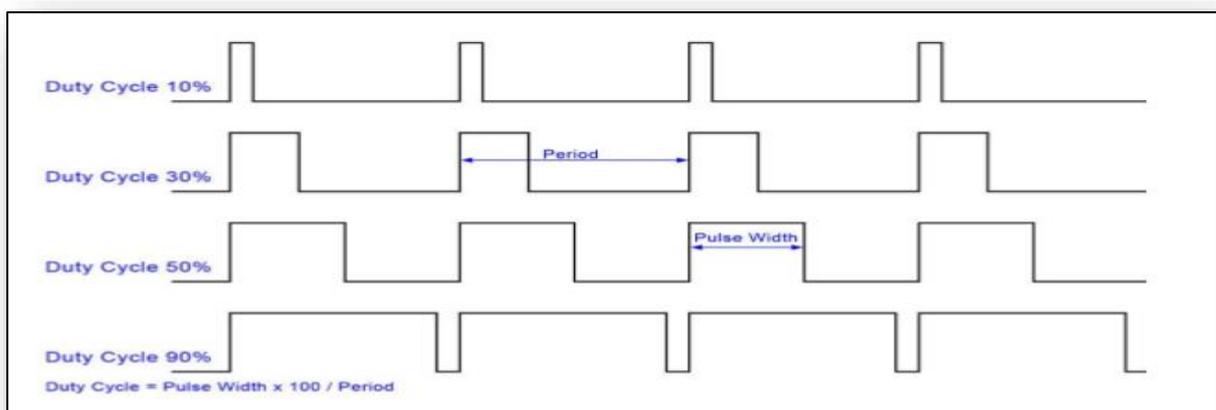


Figure5 : signal PWM

I.4) Quelques librairies

En plus de la simplicité du langage et des nombreuses fonctionnalités qu'offre. L'IDE vient avec un nombre important de librairies évitant ainsi d'implémenter des fonctions courantes dans l'informatique embarquée. Une des librairies les plus utilisée est celle implémentant la communication série. La majorité des cartes Arduino possède un émulateur de connexion série pour communiquer au travers de l'USB. Ainsi, on peut communiquer avec l'ordinateur sur lequel la carte Arduino est connectée. Cela permet, par exemple, de déboguer un programme en affichant la valeur des variables ou simplement afficher la valeur des capteurs. Cette librairie a été directement implémentée dans le langage Arduino. On peut accéder à la communication série (au travers l'USB) grâce à l'objet Serial. Une autre librairie existe pour communiquer par liaison série via une des broches de la carte. Il est parfois nécessaire de stocker des informations même après l'arrêt de la carte. Il est possible de stocker une petite quantité d'information sur la mémoire EEPROM (Electrically Erasable Programmable Read Only Memory) intégrée. Une librairie est aussi fournie pour dialoguer avec cette mémoire. Il est possible de lire et d'écrire sur cette mémoire sans rien ajouter sur la carte. Cette mémoire est accessible via l'objet EEPROM et en ajoutant la librairie du même nom. Attention, la mémoire EEPROM à une durée de vie limitée (environ 100 000 cycles d'écritures). Il faut donc veiller à ne pas écrire répétitivement les données. D'autres alternatives existent pour ajouter de plus grandes quantités de mémoires mortes à l'Arduino (carte SD notamment) mais cela demande l'ajout de composants externes. Pour terminer, il existe aussi des librairies permettant de contrôler des composants externes. Parmi ces librairies, deux plus populaires

a-`<sevseg.h>` : qui est la librairie qui sert le contrôle des afficheurs 7 segments.

b-`<Controleled.h>` : elle a le même rôle de la librairie précédente et aussi permet de gérer et contrôler des microprocesseurs comme notre contrôleur d'affichage le Maxim «max 7219 cng ou le max 7219 eng ».

I.5) Présentation générale sur Proteus Professional:

Proteus Professional est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus Professional permettent la CAO (Construction Assistée par Ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: (ISIS, ARES, PROSPICE) et VSM. 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 Professional possède d'autres avantages :

- a• Pack contenant des logiciels facile et rapide à comprendre et utiliser.
- b• Le support technique est performant.
- c• L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet

I.5.1) ISIS :

ISIS de Proteus Professional 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.

I.5.2) ARES :

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB (Printed circuit board) 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.

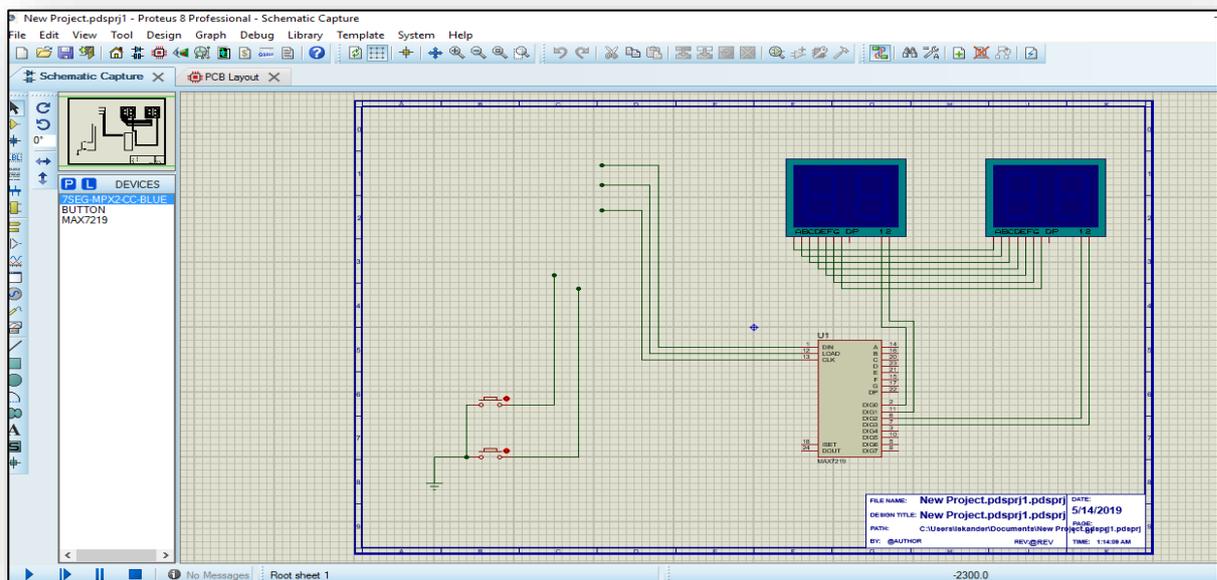


Figure 6 : Un schéma électrique par isis proteus

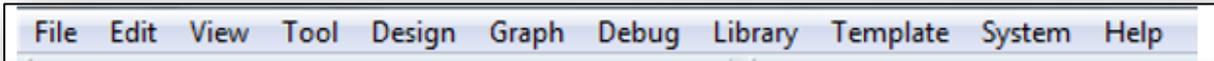


Figure 7 : Barre de menus

Cette barre permet de gérer votre travail (ouverture, sauvegarde, impression, mode d'affichage, etc.).



Figure 8 : Barres d’outils de commande

Ces barres fournissent un accès équivalent aux commandes des menus. Elles peuvent être masquées par la commande "Barre d’outils" du menu "Affichage"



Figure 9 : Barre d’outils de sélection de mode

Cette barre permet de sélectionner un outil parmi les 3 modes d’édition disponibles



Figure 10 : Barre d’orientation

Cette barre permet d'afficher et de contrôler la rotation et la réflexion d'un objet placé ou à placer.

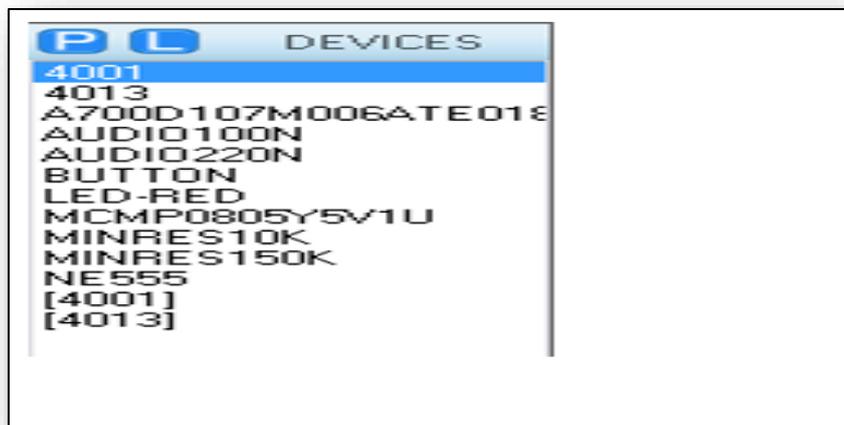
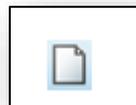


Figure 11 : Sélecteur d'objets

Le sélecteur d'objets liste les différents éléments, selon le mode de travail choisi. Les types d'objets qui peuvent y apparaître sont les composants, les terminaux, les pattes, les symboles graphiques, les marqueurs, les graphes.

I.6) Gestion d'un projet et création d'un nouveau projet :



a• Menu Fichier Nouveau Projet ou appuyer sur l'icône



b• Menu Fichier Ouvrir projet ou appuyer sur l'icône



c• Menu Fichier Enregistrer projet sous



- d• Menu Fichier Enregistrer projet ou appuyer sur l'icône

I.7) Edition d'un schéma

I.7.1) Ajout d'un composant

I.7.1.a) Chargement d'un composant

Avant de commencer un projet, vous devez d'abord pré-charger et mettre en attente les différents composants que vous comptez utiliser. Il est néanmoins possible de les charger plus tard. Les composants disponibles sont regroupés dans des bibliothèques classées par thèmes, Pour charger un composant :



- a• Cliquer sur l'icône "Composant de la barre d'outils de sélection démodés.



- b• Appuyer sur la touche P du sélecteur d'objets
- c• La fenêtre "Pickdevices" apparaît : Dans le cas où vous ne connaissez pas la catégorie où se trouve le composant désiré, il suffit de taper son nom ou quelques lettres dans la zone "Mots clés" afin que l'application sélectionne certaines catégories susceptibles d'accueillir le composant recherché. Une fois ce choix fait ou si vous connaissez la catégorie, il suffit de :
- d• Sélectionner la catégorie
- e• Cliquer sur l'objet, son aperçu apparaît dans la fenêtre de droite
- f• Double-cliquer sur l'objet désiré pour le charger.
Celui-ci sera ajouté dans la liste "Device" du sélecteur d'objets.

g• Une fois tous les objets chargés, fermer la fenêtre "PickDevices".

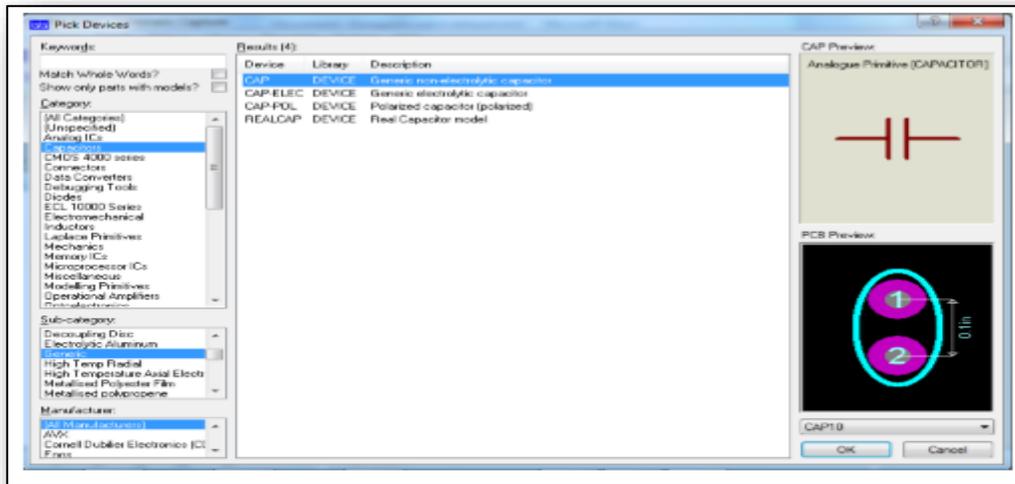


Figure 12 : Chargement des composants sur isis protus

I.8) EDITION DE CIRCUIT IMPRIME :

I.8.1) L'ECRAN ARES

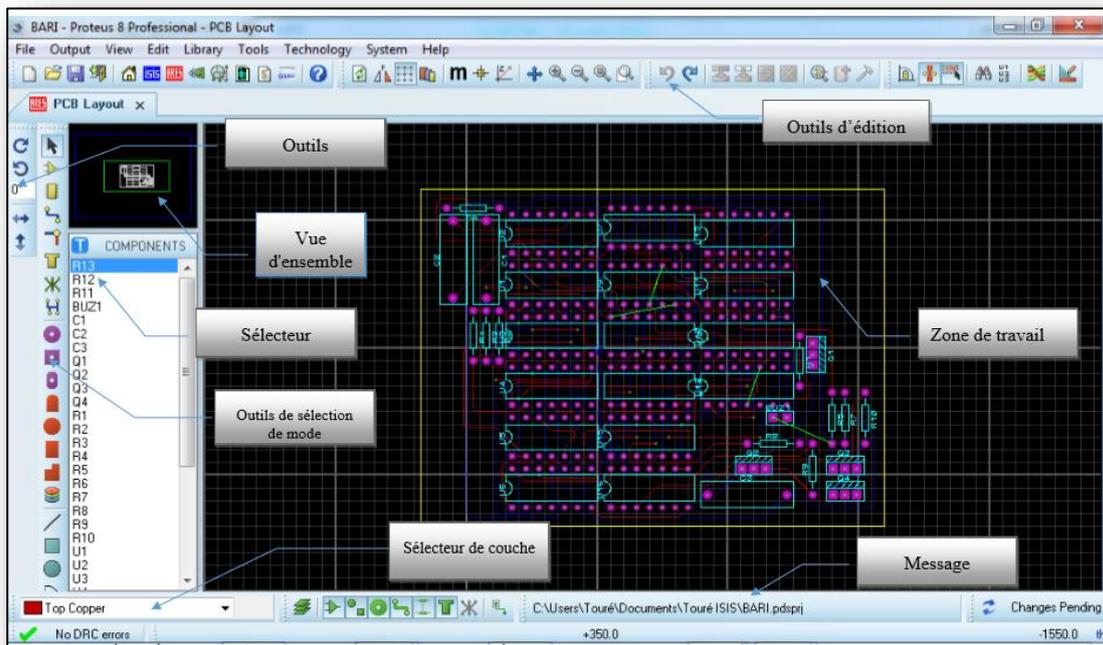


Figure13 : illustration d'un schéma PCB par isis protus

I.8.2) DEBUT DE CREATION DU CIRCUIT IMPRIME :**8.2.a) UN COMPOSANT**

Cliquer sur le bouton

« Composants », puis le placer dans la zone de travail

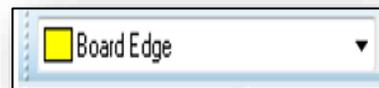
8.2.b) UNE PISTE

Pour tracer une piste, cliquer sur bouton piste
et de routage

des outils de placement

8.2.c) CREATION D'UN TYPON

a) Cliquer sur l'un des boutons du mode graphique



b) Sélectionner le mode de traçage des bordures

c) Utiliser les outils graphiques pour dessiner les contours du circuit (la forme doit être fermée).

I.8.3) SELECTION DES EMPREINTES A UTILISER :

Si vous n'avez pas préalablement saisi le schéma avec ISIS ou si vous souhaitez rajouter des empreintes des composants (points test, connecteurs...), procéder comme suit:



a) Cliquer sur le bouton

de placement de boîtiers



b) Cliquer sur le bouton

“Prendre des boîtiers”.

- c) Tapez un mot clef ou sélectionnez une catégorie.
- d) Cliquer 2× sur le boîtier pour l'ajouter à la sélection.
- e) Recommencer en (C) pour sélectionner d'autres boîtiers.
- f) Fermer la fenêtre

I.8.4) PLACEMENT DES EMPREINTES DE COMPOSANTS

8.4.a) placement automatique des empreinte

Cliquer sur le menu outils, choisir Placement Automatique
Vérifier que tous les composants sont choisis puis valider par ok

8.4.b) placement manuel des empreinte



Cliquer sur le bouton "Composants".

Cliquer sur le composant choisi.

Placer la souris où le composant doit être positionné. Cliquer, le composant est placé.

I.8.5). ROUTAGE

8.5.a) LANCEMENT DU ROUTAGE AUTOMATIQUE



- a. Cliquer sur « Outils / Routeur automatique » ou le bouton
- b. Cliquez sur commencer les routages

8.5. b.)ROUTAGE MANUEL



✓ Cliquer sur le bouton "Pistes".

✓ Cliquer 2× sur « DEFAULT »

✓ Choisir la taille des pistes

✓ Cliquer sur « OK »

I.9) Conclusion :

Ce chapitre est portail pour la réalisation et la conception matérielle .Ce travail m'a permis de maitriser les options de la carte arduino uno avec les afficheurs 7segments et le contrôleur d'affichage (max 7219 cng/eng), afin de les exploiter d'une manière correcte .J'ai présenté également des principes et des fonctions de la programmation dédiée à la carte arduino uno. J'ai choisis la carte arduino uno pour son aspect économique, sa popularité.

II. LES AFFICHEUR 7 SEGMENTS

II.1. Définition d'un afficheur à 7 segments

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 [5], [6] :

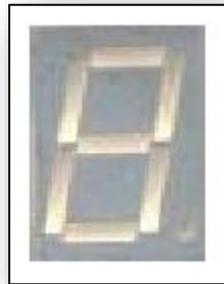


Figure 6 : Afficheur 7 segment

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

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

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)

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

II.2. Des LED :

Les LED, il y en a ! Entre 7 et 8 selon les modèles. Voici un schéma vous présentant un modèle d'afficheur sans le point :

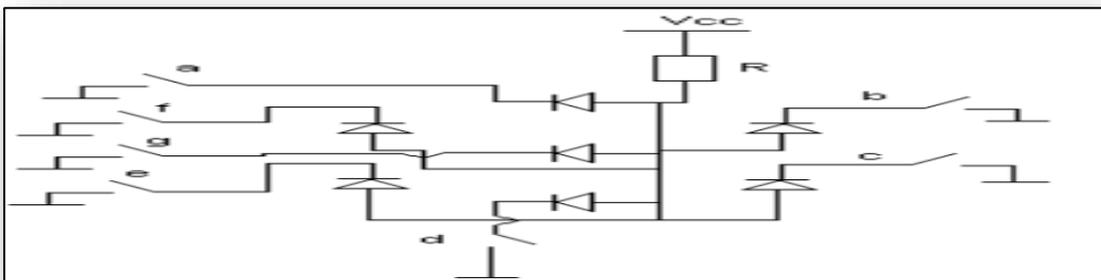


Figure 7: schéma électrique interne d'un afficheur 7 segments anode commune

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).

II.3. Cathode commune ou Anode commune :

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

II.4. Choix de l'afficheur :

Pour la rédaction nous avons fait le choix d'utiliser des afficheurs à anode commune pour réaliser la tâche de l'horloge du système. Ainsi, dans la deuxième tâche on a utilisé les afficheurs à cathode commune. Les LED seront branchées d'un côté à +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. Nous allons maintenant voir comment brancher l'afficheur à cathode commune

II.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) :

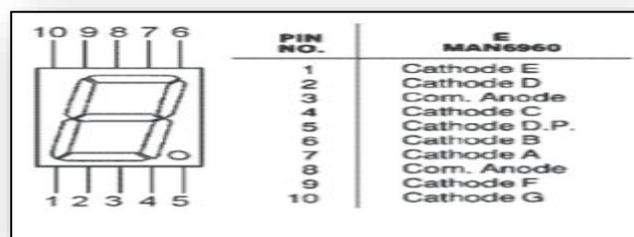


Figure 8: Représentation des segments et les broches par numéro d'un afficheur 7 segment cathode commun

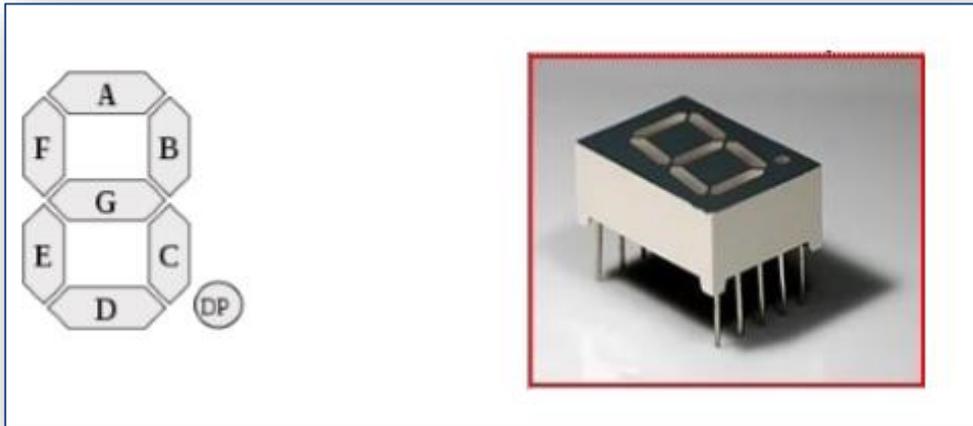


Figure 9 : la représentation des segments et les broches d'un afficheur 7 segment anodecommun

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 pour l'afficheur cathode commun et pour allumer un segment d'un afficheur a anode commun il suffit le relier à 5 v.

II.6.Problématiques liée au score board avec l'arduinouno :

Si on utilisant l'afficheur 7 segment dans notre projet qui consiste de faire une table d'affichage semi professionnel basé à des afficheurs 7 segments notamment qu'on a utilisés le microcontrôleur arduino uno qui a un nombre des ports illimité et un seule afficheur a 7 broches ou 7 sorties. En effet pour résoudre cette problème ont besoin d'une procédure de multiplexage qui minimise le nombre des ports utilisés possible, cette solution est valable lorsque on utilisant le circuit max7219 cng ou max7219 eng, qu'appelle aussi le contrôleur d'affichage

II.7.le contrôleur d'affichage max (7219/7222) :**II.7.a- description Générale**

Les MAX7219/MAX7221 sont des interfaces pilotes compacts pour, bar - graphe, ou 64 LED individuels ou des afficheurs 7 segments numérique à LEDS de cathode commune. Il peut afficher jusqu'à 8 chiffres via une commande série piloté par les d'entrée/sortie d'un microprocesseurs (Ps). μ Sont Inclus dans la puce un BCD Code – B décodeur, un circuit de balayage multiplex, un pilotes de segment et chiffres, et une mémoire RAM statique 8x8 qui stocke chaque chiffre. Une seule résistance externe est nécessaire pour appliquer le courant sur toutes les LEDS du segment. Le MAX7221 est compatible SPI™, QSPI™, et MICROWIRE™, et pilotes de vitesse de limite de balayage segments pour réduire les EMI (interférence électromagnétique). Une interface série 4 fils pratique qui se connectent à tous Ps communs. Les Chiffres individuels peuvent être adressés μ et mise à jour sans avoir à réécrire la totalité de l'affichage. La MAX7219 / MAX7221 permettent également à l'utilisateur de sélectionner code-B décodage ou de non-décodage pour chaque chiffre. Les dispositifs comprennent une faible puissance à 150 A , arrêt Mode, le contrôle de la luminosité analogique et μ numérique, Un registre de numérisation de fin de course qui permet à l'utilisateur d'afficher de 1 à 8 chiffres, et un mode de test qui force tous les voyants. [7]

II.7.b. la configuration des pins ou des broches du max7219 :

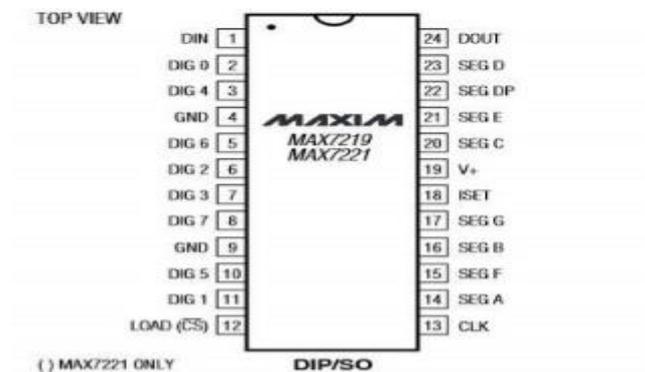
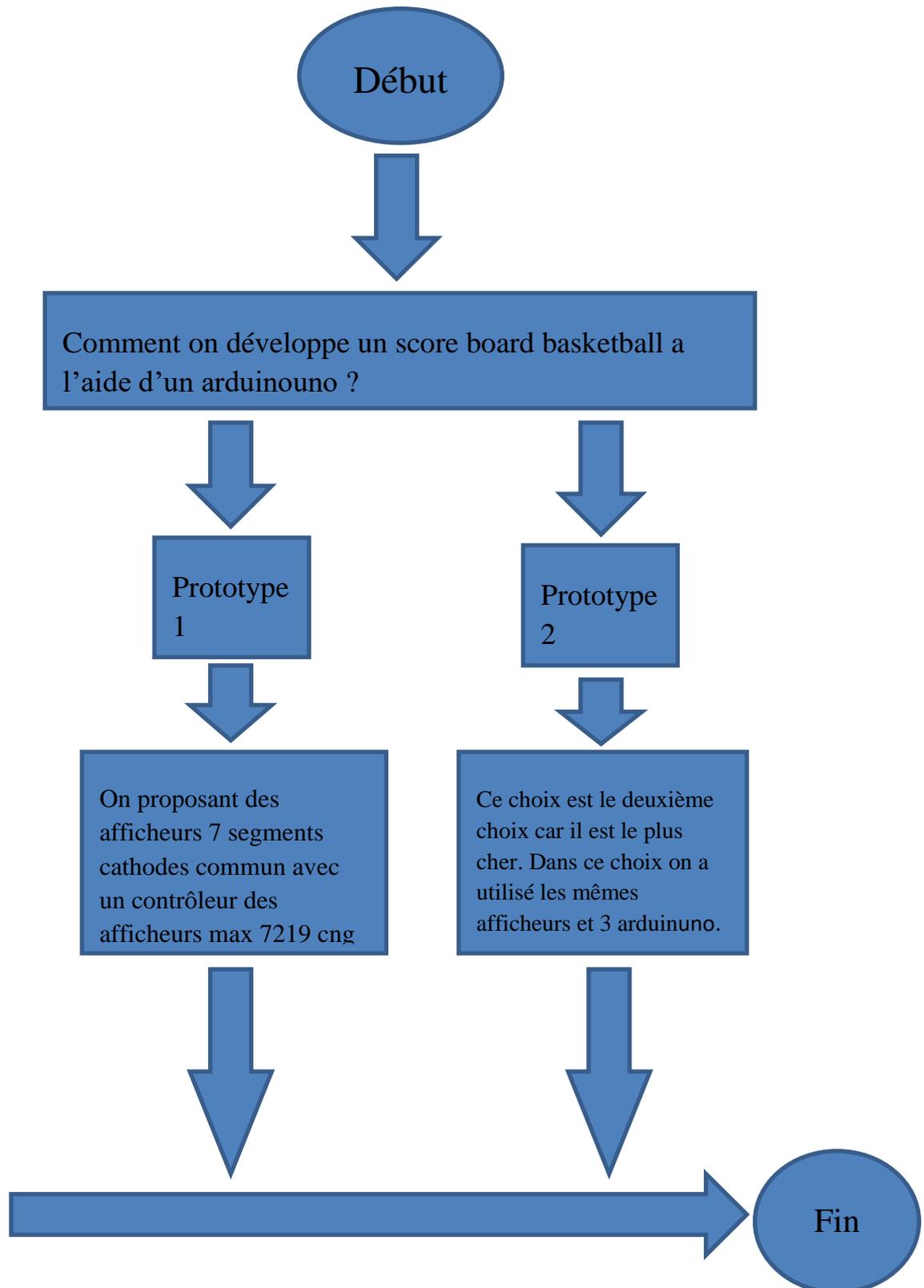


Figure 10: Configuration des pins du max 7219

II.8) Conclusion :

Dans ce chapitre on a bien définie les afficheurs 7 segments selon leur caractéristiques et leurs types (cathode commun ou anode commun) et comment ça fonctionnent ces deux dernier afficheurs et puis on a donné aussi une description sur le contrôleur d'affichage (max 7219 cng/eng).

III.1) Objectif du travail :



III.2) les avantages et les inconvénients des choix ou les prototypes proposer dans ce projet :**III.2.1.a) les avantages du premier prototype :**

- 1-Il range tous les composants du travail à la même carte.
- 2-Il utilise un seul arduinouno, il consomme moins d'énergie.
- 3- Plus couteux.

III.2.1.b) les inconvénients du premier prototype :

- 1-le contrôleur d'affichage sert de commander seulement les afficheurs 7 segments cathode commun.

III.2.2.a) les avantages du deuxième prototype :

- 1-il sert de réaliser notre score board avec n'importe afficheur (7segment cathode commun ou anode commun, LCD etc.....).

III.2.2.b) les inconvénients du deuxième prototype :

- 1-il utilise trois arduinouno, donc il consomme beaucoup d'énergie.
- 2-plus cher.

III.3) La conception pratique :

Dans ce chapitre on a introduit l'éditeur ou le logicielle qui nous utilisons pour créer et gérer notre système (scoreboard basketballwitharduinouno) et illustrer ce système avec ses composants électroniques dans une plaque d'essai et leur PCB on basant sur cette logicielle avantde le faire au réel.

III.4) Simulationdu système et test du programme :

Pour réaliser notre système on a décidé de réaléser les de prototypes (1 et 2) par conséquent ont distingués deux taches matérielle et logicielle :

- a- la tache de pointage du score boardbasketball.
- b- la tache de l'horloge. Ou le période

III.4.1.a). –le système de pointage du score board prototype 1 :

C'est système utilisé par un opérateur (le personne qui faire l'opération de pointage sur score board) généralement dans un partie ou match et dans notre cas c'est le basketball, et dans notre projet le système du pointage est constitué d'un deux afficheurs 7 segments cathode commun et deux boutons poussoir sert de faire l'opération de pointage et comme on a choisit le basketball l'opérateur doit marquer (1 point ou 2 points ou 3 trois points).la figure 19 ce dessous montre bien comment ce système ça marche

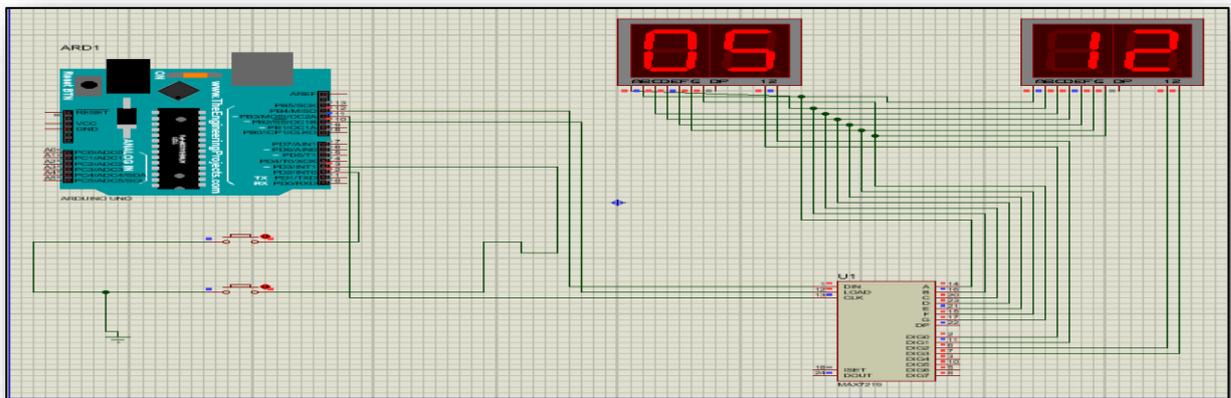


Figure 19 : Simulation du système de pointage score board

III.4.2. b)- Système de l'horloge du score board :

Ce système est tout simplement affiche le temp ou période d'un partie ou match dans notre cas un partie de basketball.

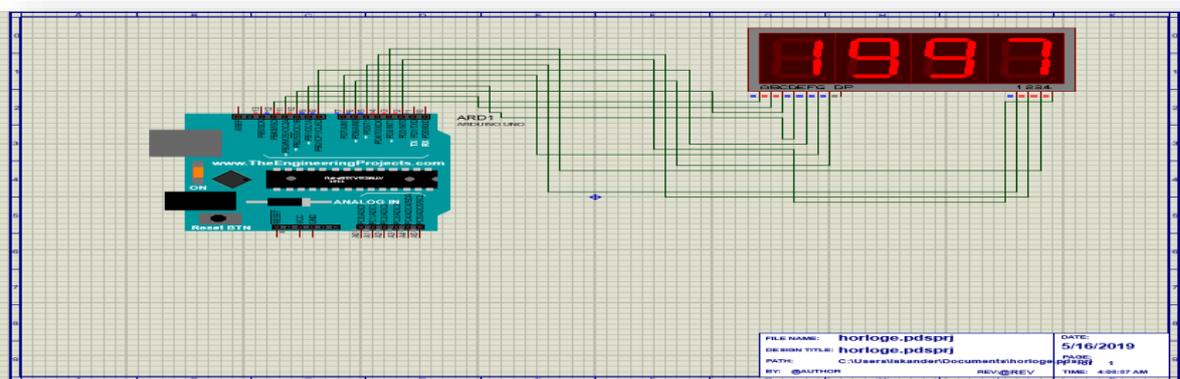


Figure 20: Simulation de l'horloge de score board

III.6. les photos de système de pointage prototype 1:

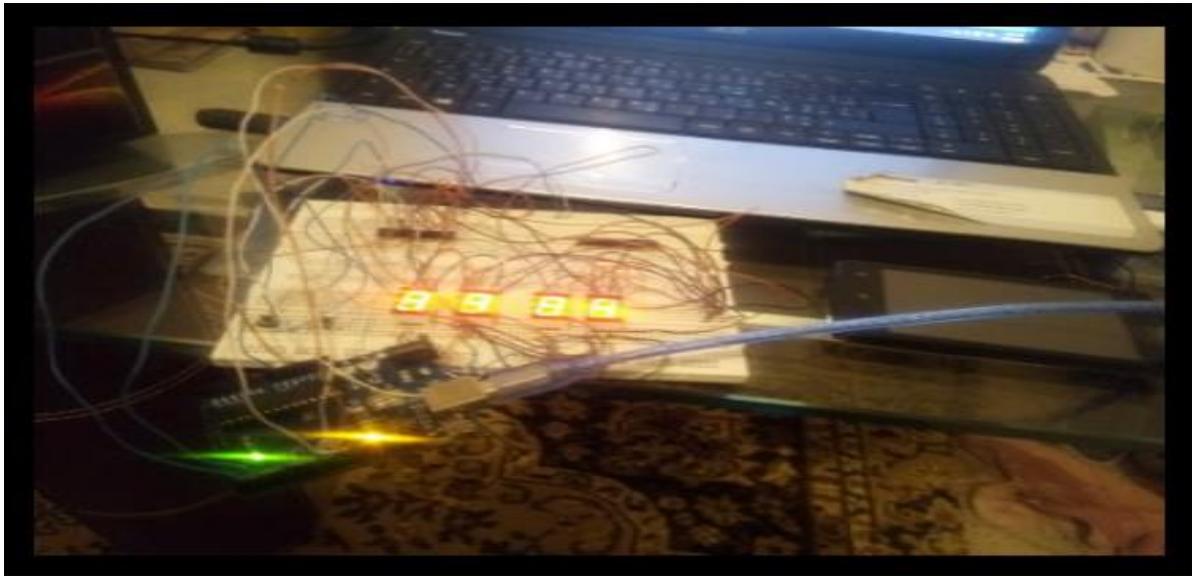


Figure 21 : Photo du système de pointage 1ere prototype

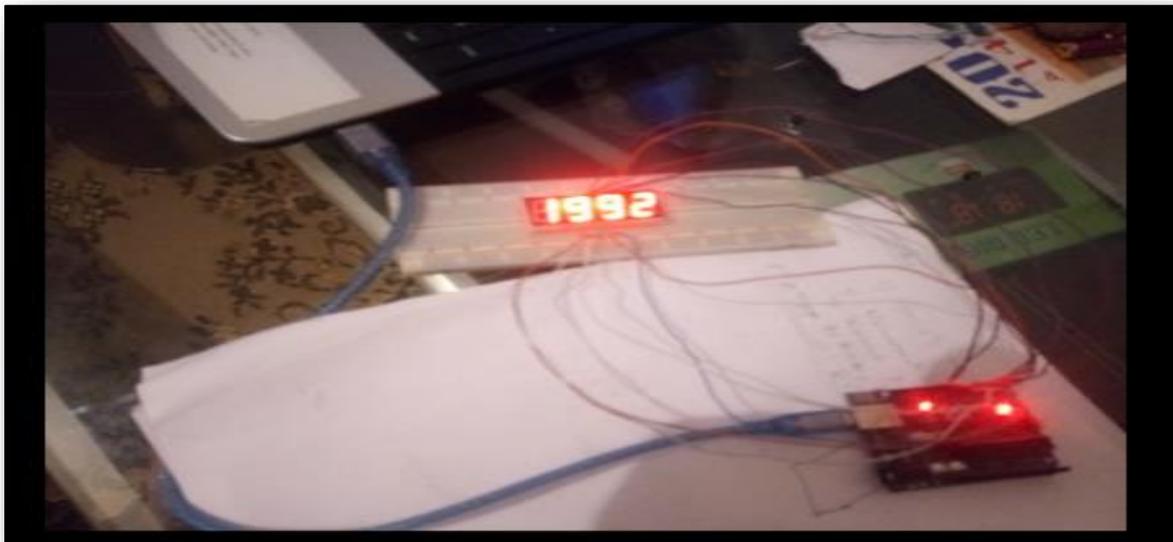


Figure 21 : photo du système de l'horloge

III.7) le système de score boardbasketball avec le deuxième prototype :

Figure22 : photo du score boardbaske



Figure23 : photo du Système de pointage du score board

Conclusion :

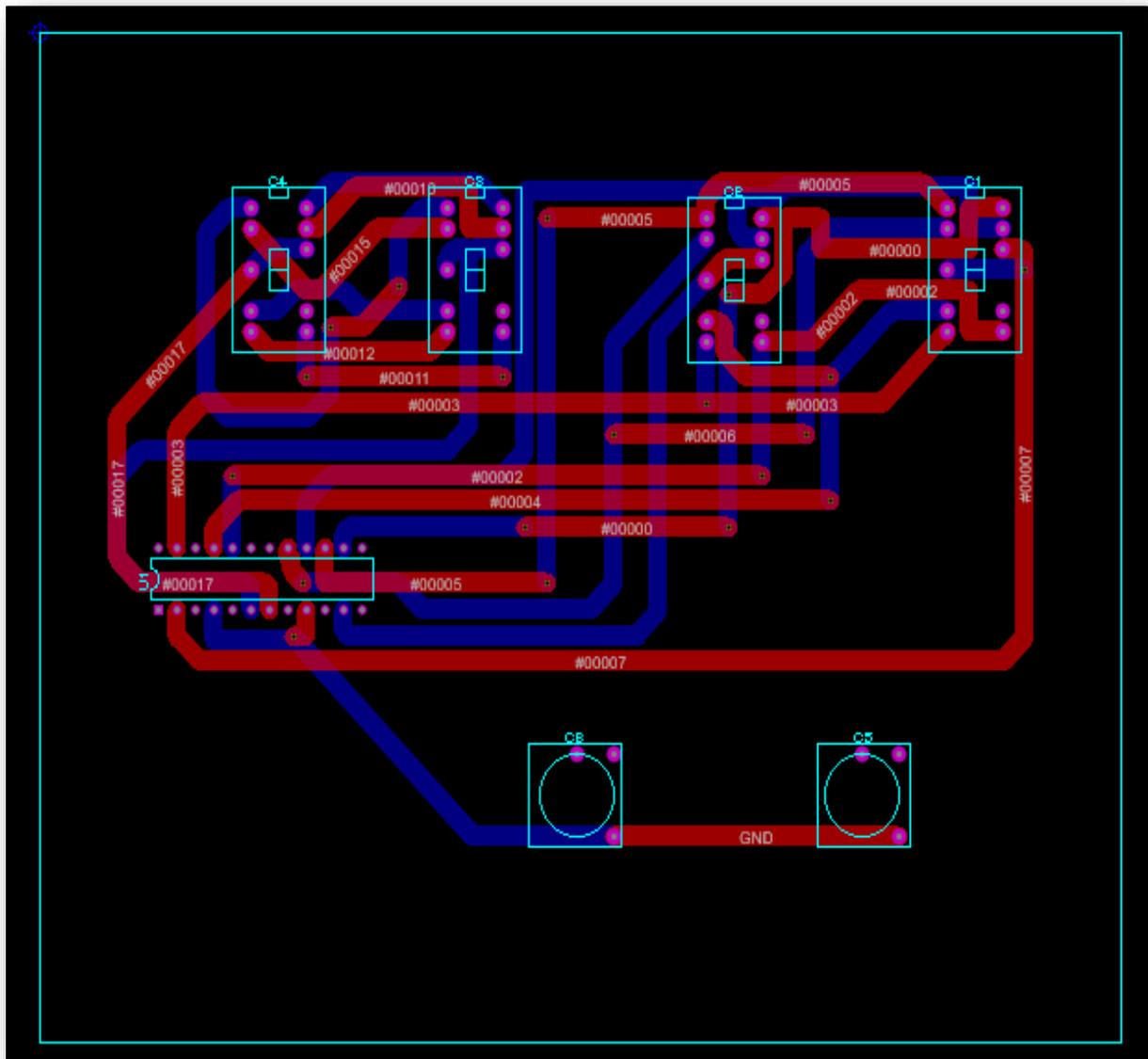
Comme nous venons de le voir dans ce chapitre, beaucoup des travaux ont été réalisés. Concernant le score board, j'ai présenté deux logiciels utilisés pour la simulation et la commande de score board .Pour réaliser cette commande je dispose l'utilisation d'une carte microcontrôleur "Arduinouno" qui possède un espace de controler les afficheurs.A l'aide du logiciel "Arduino " j'ai fait l'essai du codes et puis les compiler en suite j'ai les copier dans un fichier et puis les téléviser dans la carte arduino.

Conclusion générale :

L'élaboration de ce travail dans le cadre du projet de la fin d'étude, m'a permis d'approfondir mes connaissances théoriques en électronique et d'acquérir une bonne expérience au niveau de la réalisation pratique. Lors de cette manipulation, j'ai essayé de fournir une commande sert de gérer les afficheurs 7segments du score board basketball. Mon projet de fin d'étude consiste à détailler ainsi qu'à réaliser la carte électronique "score board basketball whit arduino uno". Ce projet m'a donné une meilleure idée sur la complémentarité entre le volet théorique et le volet pratique. J'ai rencontré plusieurs problèmes durant la conception du score board et malheureusement la partie du pointage ne fonctionne pas correctement dans le premier prototype a cause les afficheurs que j'ai l'utilise et que j'ai les cherchés et j'ai les ne trouve pas au marché. En fait, ce projet a été une source de découverte de plusieurs domaines d'étude telle que L'information pour la programmation embraquée et le design des schémas électroniques, Sans oublier le savoir-faire dans le domaine électronique qui consiste à réaliser pratiquement Les circuits électroniques.

Le circuit imprimé :

Le système de pointage



Les data sheet des composants électriques :

MAXIM

Serially Interfaced, 8-Digit LED Display Drivers

19-4452; Rev 4; 7/03

General Description

The MAX7219/MAX7221 are compact, serial input/output common-cathode display drivers that interface microprocessors (μ Ps) to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 individual LEDs. Included on-chip are a BCD code-B decoder, multiplex scan circuitry, segment and digit drivers, and an 8x8 static RAM that stores each digit. Only one external resistor is required to set the segment current for all LEDs. The MAX7221 is compatible with SPI™, QSPI™, and MICROWIRE™, and has slew-rate-limited segment drivers to reduce EMI.

A convenient 4-wire serial interface connects to all common μ Ps. Individual digits may be addressed and updated without rewriting the entire display. The MAX7219/MAX7221 also allow the user to select code-B decoding or no-decode for each digit.

The devices include a 150 μ A low-power shutdown mode, analog and digital brightness control, a scan-limit register that allows the user to display from 1 to 8 digits, and a test mode that forces all LEDs on.

For applications requiring 3V operation or segment blinking, refer to the MAX6951 data sheet.

Applications

Bar-Graph Displays

Industrial Controllers

Panel Meters

LED Matrix Displays

Pin Configuration

Features

- ◆ 10MHz Serial Interface
- ◆ Individual LED Segment Control
- ◆ Decode/No-Decode Digit Selection
- ◆ 150 μ A Low-Power Shutdown (Data Retained)
- ◆ Digital and Analog Brightness Control
- ◆ Display Blanked on Power-Up
- ◆ Drive Common-Cathode LED Display
- ◆ Slow-Rate Limited Segment Drivers for Lower EMI (MAX7221)
- ◆ SPI, QSPI, MICROWIRE Serial Interface (MAX7221)
- ◆ 24-Pin DIP and SO Packages

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX7219CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX7219CWG	0°C to +70°C	24 Wide SO
MAX7219C/D	0°C to +70°C	Dice*
MAX7219ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX7219EWG	-40°C to +85°C	24 Wide SO
MAX7219ERG	-40°C to +85°C	24 Narrow CERDIP

Ordering information continued at end of data sheet.
*Dice are specified at $T_A = +25^\circ\text{C}$.

Typical Application Circuit

MAX7219/MAX7221

MAX7219/MAX7221

Serially Interfaced, 8-Digit LED Display Drivers

ABSOLUTE MAXIMUM RATINGS

Voltage (with respect to GND)		Operating Temperature Ranges (T _{MIN} to T _{MAX})
V+	-0.3V to 6V	MAX7219C_G/MAX7221C_G
DIN, CLK, LOAD, CS	-0.3V to 6V	MAX7219E_G/MAX7221E_G
All Other Pins	-0.3V to (V+ + 0.3V)	Storage Temperature Range
Current		Lead Temperature (soldering, 10s)
DIG0-DIG7 Sink Current	500mA	
SEGA-G, DP Source Current	100mA	
Continuous Power Dissipation (T _A = +85°C)		
Narrow Plastic DIP (derate 13.3mW/°C		
above +70°C)	1066mW	
Wide SO (derate 11.8mW/°C above +70°C)	941mW	
Narrow CERDIP (derate 12.5mW/°C above +70°C)	1000mW	

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V+ = 5V ± 10%, R_{SET} = 9.53kΩ ± 1%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Supply Voltage	V+		4.0		5.5	V
Shutdown Supply Current	I+	All digital inputs at V+ or GND, T _A = +25°C			150	μA
Operating Supply Current	I+	R _{SET} = open circuit			8	mA
		All segments and decimal point on, I _{SEG-} = -40mA		330		
Display Scan Rate	f _{OSC}	8 digits scanned	500	800	1300	Hz
Digit Drive Sink Current	I _{DIGIT}	V+ = 5V, V _{OUT} = 0.65V	320			mA
Segment Drive Source Current	I _{SEG}	T _A = +25°C, V+ = 5V, V _{OUT} = (V+ - 1V)	-30	-40	-45	mA
Segment Current Slew Rate (MAX7221 only)	ΔI _{SEG} /Δt	T _A = +25°C, V+ = 5V, V _{OUT} = (V+ - 1V)	10	20	50	mA/μs
Segment Drive Current Matching	ΔI _{SEG}			3.0		%
Digit Drive Leakage (MAX7221 only)	I _{DIGIT}	Digit off, V _{DIGIT} = V+			-10	μA
Segment Drive Leakage (MAX7221 only)	I _{SEG}	Segment off, V _{SEG} = 0V			1	μA
Digit Drive Source Current (MAX7219 only)	I _{DIGIT}	Digit off, V _{DIGIT} = (V+ - 0.3V)	-2			mA
Segment Drive Sink Current (MAX7219 only)	I _{SEG}	Segment off, V _{SEG} = 0.3V	5			mA

Serially Interfaced, 8-Digit LED Display Drivers

ELECTRICAL CHARACTERISTICS (continued)

(V+ = 5V ±10%, RSET = 9.53kΩ ±1%, TA = TMIN to TMAX, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
LOGIC INPUTS						
Input Current DIN, CLK, LOAD, CS	I _{IH} , I _{IL}	V _{IN} = 0V or V+	-1		1	μA
Logic High Input Voltage	V _{IH}		3.5			V
Logic Low Input Voltage	V _{IL}				0.8	V
Output High Voltage	V _{OH}	DOUT, I _{SOURCE} = -1mA	V+ - 1			V
Output Low Voltage	V _{OL}	DOUT, I _{SINK} = 1.6mA			0.4	V
Hysteresis Voltage	ΔV _I	DIN, CLK, LOAD, CS		1		V
TIMING CHARACTERISTICS						
CLK Clock Period	t _{CP}		100			ns
CLK Pulse Width High	t _{CH}		50			ns
CLK Pulse Width Low	t _{CL}		50			ns
CS Fall to SCLK Rise Setup Time (MAX7221 only)	t _{CSS}		25			ns
CLK Rise to CS or LOAD Rise Hold Time	t _{CSH}		0			ns
DIN Setup Time	t _{DS}		25			ns
DIN Hold Time	t _{DH}		0			ns
Output Data Propagation Delay	t _{DO}	C _{LOAD} = 50pF			25	ns
Load-Rising Edge to Next Clock Rising Edge (MAX7219 only)	t _{LDCK}		50			ns
Minimum CS or LOAD Pulse High	t _{CSW}		50			ns
Data-to-Segment Delay	t _{DSPD}				2.25	ms

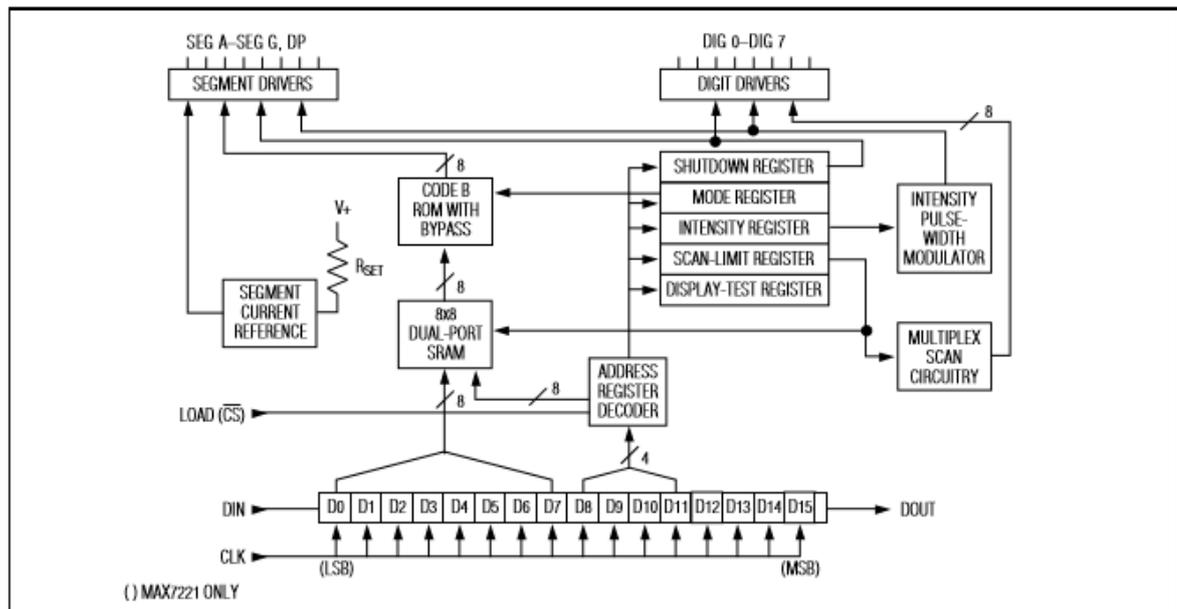
Data-to-Segment Delay	t _{DSPD}				2.25	ms
High						
Minimum CS or LOAD Pulse	t _{CSW}		50			ns
Rising Edge (MAX7219 only)	t _{LDCK}		50			ns
Load-Rising Edge to Next Clock						
Output Data Propagation Delay	t _{DO}	C _{LOAD} = 50pF			25	ns
DIN Hold Time	t _{DH}		0			ns
DIN Setup Time	t _{DS}		25			ns
CS Fall to SCLK Rise Setup Time (MAX7221 only)	t _{CSS}		25			ns
CLK Rise to CS or LOAD Rise Hold Time	t _{CSH}		0			ns
CLK Pulse Width High	t _{CH}		50			ns
CLK Pulse Width Low	t _{CL}		50			ns
CLK Clock Period	t _{CP}		100			ns

Serially Interfaced, 8-Digit LED Display Drivers

Pin Description

PIN	NAME	FUNCTION
1	DIN	Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK's rising edge.
2, 3, 5-8, 10, 11	DIG 0-DIG 7	Eight-Digit Drive Lines that sink current from the display common cathode. The MAX7219 pulls the digit outputs to V+ when turned off. The MAX7221's digit drivers are high-impedance when turned off.
4, 9	GND	Ground (both GND pins must be connected)
12	LOAD (MAX7219)	Load-Data Input. The last 16 bits of serial data are latched on LOAD's rising edge.
	\overline{CS} (MAX7221)	Chip-Select Input. Serial data is loaded into the shift register while \overline{CS} is low. The last 16 bits of serial data are latched on \overline{CS} 's rising edge.
13	CLK	Serial-Clock Input. 10MHz maximum rate. On CLK's rising edge, data is shifted into the internal shift register. On CLK's falling edge, data is clocked out of DOUT. On the MAX7221, the CLK input is active only while \overline{CS} is low.
14-17, 20-23	SEG A-SEG G, DP	Seven Segment Drives and Decimal Point Drive that source current to the display. On the MAX7219, when a segment driver is turned off it is pulled to GND. The MAX7221 segment drivers are high-impedance when turned off.
18	ISET	Connect to VDD through a resistor (RSET) to set the peak segment current (Refer to <i>Selecting RSET Resistor</i> section).
19	V+	Positive Supply Voltage. Connect to +5V.
24	DOUT	Serial-Data Output. The data into DIN is valid at DOUT 16.5 clock cycles later. This pin is used to daisy-chain several MAX7219/MAX7221's and is never high-impedance.

Functional Diagram

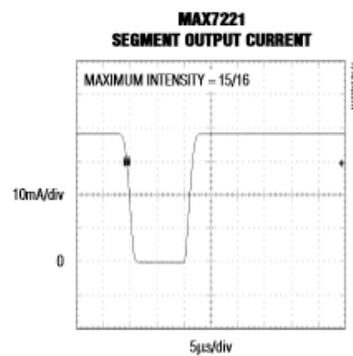
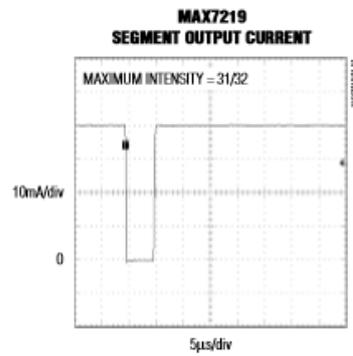
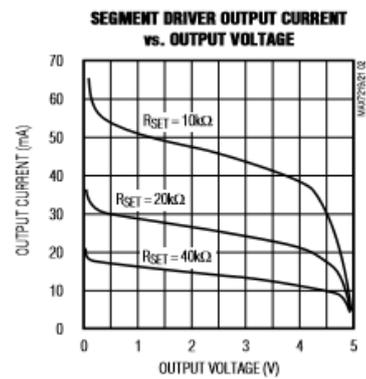
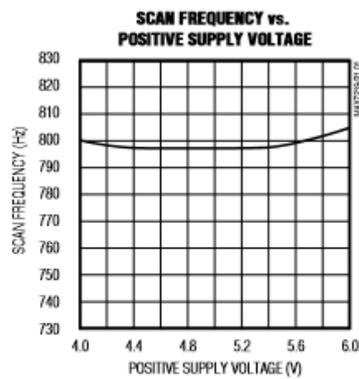


Serially Interfaced, 8-Digit LED Display Drivers

MAX7219/MAX7221

Typical Operating Characteristics

(V+ = +5V, TA = +25°C, unless otherwise noted.)



Serially Interfaced, 8-Digit LED Display Drivers

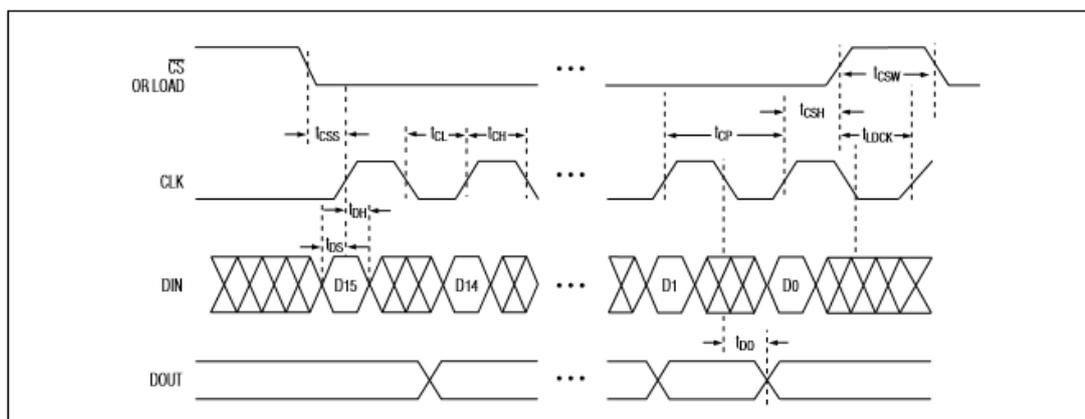


Figure 1. Timing Diagram

Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	ADDRESS				MSB				DATA				LSB

Detailed Description

MAX7219/MAX7221 Differences

The MAX7219 and MAX7221 are identical except for two parameters: the MAX7221 segment drivers are slew-rate limited to reduce electromagnetic interference (EMI), and its serial interface is fully SPI compatible.

Serial-Addressing Modes

For the MAX7219, serial data at DIN, sent in 16-bit packets, is shifted into the internal 16-bit shift register with each rising edge of CLK regardless of the state of LOAD. For the MAX7221, CS must be low to clock data in or out. The data is then latched into either the digit or control registers on the rising edge of LOAD/CS. LOAD/CS must go high concurrently with or after the 16th rising clock edge, but before the next rising clock edge or data will be lost. Data at DIN is propagated through the shift register and appears at DOUT 16.5 clock cycles later. Data is clocked out on the falling edge of CLK. Data bits are labeled D0–D15 (Table 1). D8–D11 contain the register address. D0–D7 contain the data, and D12–D15 are “don’t care” bits. The first received is D15, the most significant bit (MSB).

Digit and Control Registers

Table 2 lists the 14 addressable digit and control registers. The digit registers are realized with an on-chip, 8x8 dual-port SRAM. They are addressed directly so that individual digits can be updated and retain data as long as V+ typically exceeds 2V. The control registers consist of decode mode, display intensity, scan limit (number of scanned digits), shutdown, and display test (all LEDs on).

Shutdown Mode

When the MAX7219 is in shutdown mode, the scan oscillator is halted, all segment current sources are pulled to ground, and all digit drivers are pulled to V+, thereby blanking the display. The MAX7221 is identical, except the drivers are high-impedance. Data in the digit and control registers remains unaltered. Shutdown can be used to save power or as an alarm to flash the display by successively entering and leaving shutdown mode. For minimum supply current in shutdown mode, logic inputs should be at ground or V+ (CMOS-logic levels).

Typically, it takes less than 250µs for the MAX7219/MAX7221 to leave shutdown mode. The display driver can be programmed while in shutdown mode, and shutdown mode can be overridden by the display-test function.

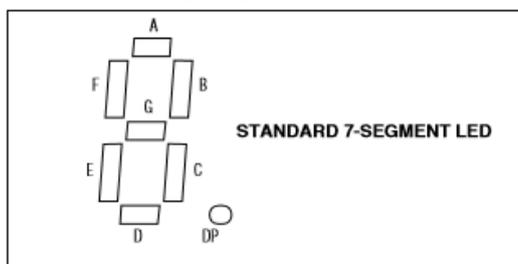
Serially Interfaced, 8-Digit LED Display Drivers

Table 5. Code B Font

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6-D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

*The decimal point is set by bit D7 = 1

Table 6. No-Decode Mode Data Bits and Corresponding Segment Lines



	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Corresponding Segment Line	DP	A	B	C	D	E	F	G

Intensity Control and Interdigit Blanking

The MAX7219/MAX7221 allow display brightness to be controlled with an external resistor (RSET) connected between V+ and ISET. The peak current sourced from the segment drivers is nominally 100 times the current entering ISET. This resistor can either be fixed or variable to allow brightness adjustment from the front panel. Its minimum value should be 9.53kΩ, which typically sets the segment current at 40mA. Display brightness can also be controlled digitally by using the intensity register.

Digital control of display brightness is provided by an internal pulse-width modulator, which is controlled by the lower nibble of the intensity register. The modulator scales the average segment current in 16 steps from a maximum of 31/32 down to 1/32 of the peak current set by RSET (15/16 to 1/16 on MAX7221). Table 7 lists the intensity register format. The minimum interdigit blanking time is set to 1/32 of a cycle.

Serially Interfaced, 8-Digit LED Display Drivers

Table 7. Intensity Register Format (Address (Hex) = 0xXA)

DUTY CYCLE		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
MAX7219	MAX7221									
1/32 (min on)	1/16 (min on)	X	X	X	X	0	0	0	0	0xX0
3/32	2/16	X	X	X	X	0	0	0	1	0xX1
5/32	3/16	X	X	X	X	0	0	1	0	0xX2
7/32	4/16	X	X	X	X	0	0	1	1	0xX3
9/32	5/16	X	X	X	X	0	1	0	0	0xX4
11/32	6/16	X	X	X	X	0	1	0	1	0xX5
13/32	7/16	X	X	X	X	0	1	1	0	0xX6
15/32	8/16	X	X	X	X	0	1	1	1	0xX7
17/32	9/16	X	X	X	X	1	0	0	0	0xX8
19/32	10/16	X	X	X	X	1	0	0	1	0xX9
21/32	11/16	X	X	X	X	1	0	1	0	0xXA
23/32	12/16	X	X	X	X	1	0	1	1	0xXB
25/32	13/16	X	X	X	X	1	1	0	0	0xXC
27/32	14/16	X	X	X	X	1	1	0	1	0xXD
29/32	15/16	X	X	X	X	1	1	1	0	0xXE
31/32	15/16 (max on)	X	X	X	X	1	1	1	1	0xFF

Table 8. Scan-Limit Register Format (Address (Hex) = 0xXB)

SCAN LIMIT	REGISTER DATA								HEX CODE	
	D7	D6	D5	D4	D3	D2	D1	D0		
Display digit 0 only*	X	X	X	X	X	0	0	0	0	0xX0
Display digits 0 & 1*	X	X	X	X	X	0	0	1	0	0xX1
Display digits 0 1 2*	X	X	X	X	X	0	1	0	0	0xX2
Display digits 0 1 2 3	X	X	X	X	X	0	1	1	0	0xX3
Display digits 0 1 2 3 4	X	X	X	X	X	1	0	0	0	0xX4
Display digits 0 1 2 3 4 5	X	X	X	X	X	1	0	1	0	0xX5
Display digits 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	0	0xX6
Display digits 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	0	0xX7

*See *Scan-Limit Register* section for application.

Scan-Limit Register

The scan-limit register sets how many digits are displayed, from 1 to 8. They are displayed in a multiplexed manner with a typical display scan rate of 800Hz with 8 digits displayed. If fewer digits are displayed, the scan rate is $8f_{OSC}/N$, where N is the number of digits

scanned. Since the number of scanned digits affects the display brightness, the scan-limit register should not be used to blank portions of the display (such as leading zero suppression). Table 8 lists the scan-limit register format.

Serially Interfaced, 8-Digit LED Display Drivers

If the scan-limit register is set for three digits or less, individual digit drivers will dissipate excessive amounts of power. Consequently, the value of the RSET resistor must be adjusted according to the number of digits displayed, to limit individual digit driver power dissipation. Table 9 lists the number of digits displayed and the corresponding maximum recommended segment current when the digit drivers are used.

Display-Test Register

The display-test register operates in two modes: normal and display test. Display-test mode turns all LEDs on by overriding, but not altering, all controls and digit registers (including the shutdown register). In display-test mode, 8 digits are scanned and the duty cycle is 31/32 (15/16 for MAX7221). Table 10 lists the display-test register format.

Table 9. Maximum Segment Current for 1-, 2-, or 3-Digit Displays

NUMBER OF DIGITS DISPLAYED	MAXIMUM SEGMENT CURRENT (mA)
1	10
2	20
3	30

Table 10. Display-Test Register Format (Address (Hex) = 0xXF)

MODE	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Normal Operation	X	X	X	X	X	X	X	0
Display Test Mode	X	X	X	X	X	X	X	1

Note: The MAX7219/MAX7221 remain in display-test mode (all LEDs on) until the display-test register is reconfigured for normal operation.

No-Op Register

The no-op register is used when cascading MAX7219s or MAX7221s. Connect all devices' LOAD/CS inputs together and connect DOUT to DIN on adjacent devices. DOUT is a CMOS logic-level output that easily drives DIN of successively cascaded parts. (Refer to the *Serial Addressing Modes* section for detailed information on serial input/output timing.) For example, if four MAX7219s are cascaded, then to write to the

fourth chip, send the desired 16-bit word, followed by three no-op codes (hex 0xXX0X, see Table 2). When LOAD/CS goes high, data is latched in all devices. The first three chips receive no-op commands, and the fourth receives the intended data.

Applications Information

Supply Bypassing and Wiring

To minimize power-supply ripple due to the peak digit driver currents, connect a 10µF electrolytic and a 0.1µF ceramic capacitor between V+ and GND as close to the device as possible. The MAX7219/MAX7221 should be placed in close proximity to the LED display, and connections should be kept as short as possible to minimize the effects of wiring inductance and electromagnetic interference. Also, both GND pins must be connected to ground.

Selecting RSET Resistor and Using External Drivers

The current per segment is approximately 100 times the current in ISET. To select RSET, see Table 11. The MAX7219/MAX7221's maximum recommended segment current is 40mA. For segment current levels above these levels, external digit drivers will be needed. In this application, the MAX7219/MAX7221 serve only as controllers for other high-current drivers or transistors. Therefore, to conserve power, use RSET = 47kΩ when using external current sources as segment drivers.

The example in Figure 2 uses the MAX7219/MAX7221's segment drivers, a MAX394 single-pole double-throw analog switch, and external transistors to drive 2.3" AND2307SLC common-cathode displays. The 5.6V zener diode has been added in series with the decimal point LED because the decimal point LED forward voltage is typically 4.2V. For all other segments the LED forward voltage is typically 8V. Since external transistors are used to sink current (DIG 0 and DIG 1 are used as logic switches), peak segment currents of 45mA are allowed even though only two digits are displayed. In applications where the MAX7219/MAX7221's digit drivers are used to sink current and fewer than four digits are displayed, Table 9 specifies the maximum allowable segment current. RSET must be selected accordingly (Table 11).

Refer to the Power Dissipation section of the Absolute Maximum Ratings to calculate acceptable limits for ambient temperature, segment current, and the LED forward-voltage drop.

Serially Interfaced, 8-Digit LED Display Drivers

Table 2. Register Address Map

REGISTER	ADDRESS					HEX CODE
	D15-D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xFF

Initial Power-Up

On initial power-up, all control registers are reset, the display is blanked, and the MAX7219/MAX7221 enter shutdown mode. Program the display driver prior to display use. Otherwise, it will initially be set to scan one digit, it will not decode data in the data registers, and the intensity register will be set to its minimum value.

Decode-Mode Register

The decode-mode register sets BCD code B (0-9, E, H, L, P, and -) or no-decode operation for each digit. Each bit in the register corresponds to one digit. A logic high selects code B decoding while logic low bypasses the decoder. Examples of the decode mode control-register format are shown in Table 4.

When the code B decode mode is used, the decoder looks only at the lower nibble of the data in the digit registers (D3-D0), disregarding bits D4-D6. D7, which sets the decimal point (SEG DP), is independent of the decoder and is positive logic (D7 = 1 turns the decimal point on). Table 5 lists the code B font.

When no-decode is selected, data bits D7-D0 correspond to the segment lines of the MAX7219/MAX7221. Table 6 shows the one-to-one pairing of each data bit to the appropriate segment line.

Table 3. Shutdown Register Format (Address (Hex) = 0xXC)

MODE	ADDRESS CODE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode	0xXC	X	X	X	X	X	X	X	0
Normal Operation	0xXC	X	X	X	X	X	X	X	1

Table 4. Decode-Mode Register Examples (Address (Hex) = 0xX9)

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7-0	0	0	0	0	0	0	0	0	0x00
Code B decode for digit 0 No decode for digits 7-1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3-0 No decode for digits 7-4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7-0	1	1	1	1	1	1	1	1	0xFF

Serially Interfaced, 8-Digit LED Display Drivers

Table 11. RSET vs. Segment Current and LED Forward Voltage

ISEG (mA)	VLED (V)				
	1.5	2.0	2.5	3.0	3.5
40	12.2	11.8	11.0	10.6	9.69
30	17.8	17.1	15.8	15.0	14.0
20	29.8	28.0	25.9	24.5	22.6
10	66.7	63.7	59.3	55.4	51.2

Computing Power Dissipation

The upper limit for power dissipation (PD) for the MAX7219/MAX7221 is determined from the following equation:

$$PD = (V_+ \times I_{SEG}) + (V_+ - V_{LED})(DUTY \times I_{SEG} \times N)$$

where:

V_+ = supply voltage

DUTY = duty cycle set by intensity register

N = number of segments driven (worst case is 8)

V_{LED} = LED forward voltage

I_{SEG} = segment current set by RSET

Dissipation Example:

$$I_{SEG} = 40\text{mA}, N = 8, DUTY = 31/32, V_{LED} = 1.8\text{V at } 40\text{mA}, V_+ = 5.25\text{V}$$

$$PD = 5.25\text{V}(40\text{mA}) + (5.25\text{V} - 1.8\text{V})(31/32 \times 40\text{mA} \times 8) = 1.11\text{W}$$

Thus, for a CERDIP package ($\theta_{JA} = +80^\circ\text{C/W}$ from Table 12), the maximum allowed ambient temperature T_A is given by:

$$T_{J(\text{MAX})} = T_A + PD \times \theta_{JA} + 150^\circ\text{C} = T_A + 1.11\text{W} \times 80^\circ\text{C/W}$$

where $T_A = +61.2^\circ\text{C}$.

The T_A limits for PDIP and SO Packages in the dissipation example above are $+66.7^\circ\text{C}$ and $+55.6^\circ\text{C}$, respectively.

Table 12. Package Thermal Resistance Data

PACKAGE	THERMAL RESISTANCE (θ_{JA})
24 Narrow DIP	$+75^\circ\text{C/W}$
24 Wide SO	$+85^\circ\text{C/W}$
24 CERDIP	$+80^\circ\text{C/W}$
Maximum Junction Temperature (T_J) = $+150^\circ\text{C}$	
Maximum Ambient Temperature (T_A) = $+85^\circ\text{C}$	

Cascading Drivers

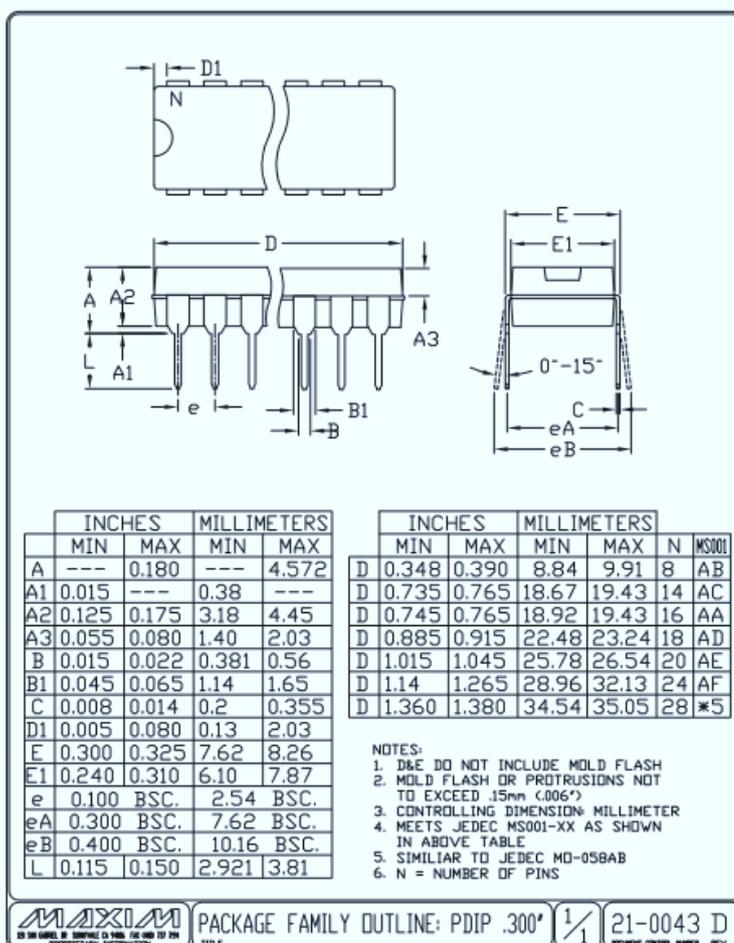
The example in Figure 3 drives 16 digits using a 3-wire μP interface. If the number of digits is not a multiple of 8, set both drivers' scan limits registers to the same number so one display will not appear brighter than the other. For example, if 12 digits are needed, use 6 digits per display with both scan-limit registers set for 6 digits so that both displays have a 1/6 duty cycle per digit. If 11 digits are needed, set both scan-limit registers for 6 digits and leave one digit driver unconnected. If one display for 6 digits and the other for 5 digits, the second display will appear brighter because its duty cycle per digit will be 1/5 while the first display's will be 1/6. Refer to the *No-Op Register* section for additional information.

Serially Interfaced, 8-Digit LED Display Drivers

MAX7219/MAX7221

Package Information (continued)

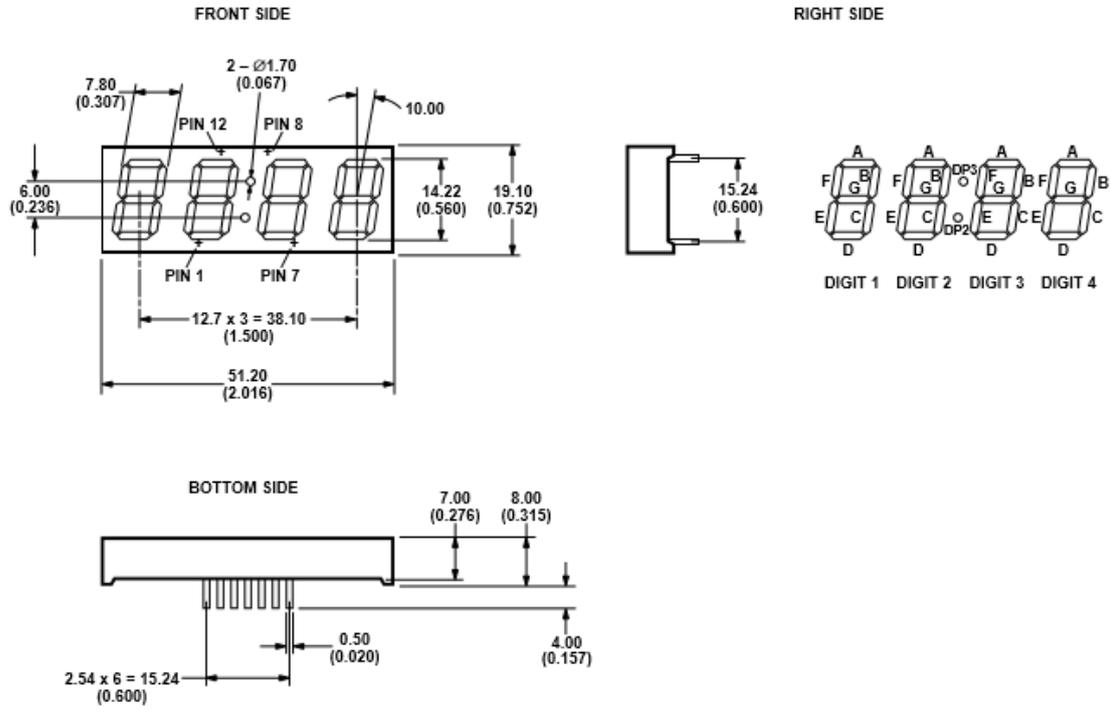
(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to www.maxim-ic.com/packages.)



Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Package Dimensions

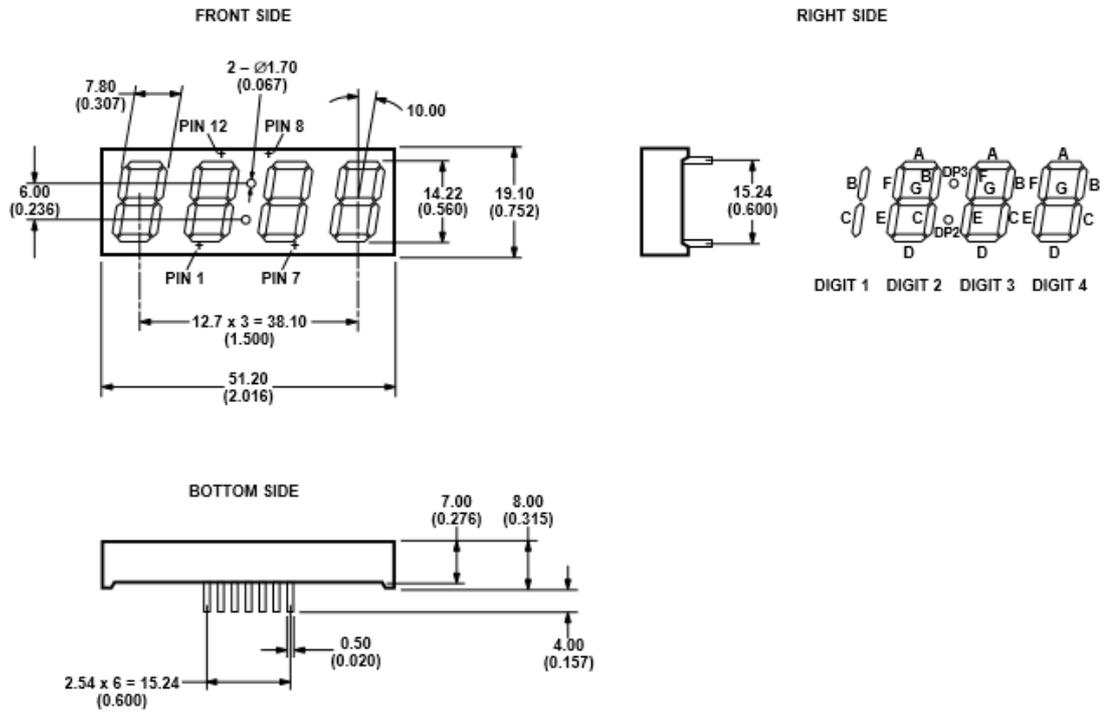
88:88 0.56" Four Digit Display



NOTE:
 ALL DIMENSIONS ARE IN MILLIMETERS (INCHES).
 UNLESS OTHERWISE STATED, TOLERANCES ARE ± 0.25 mm.

Package Dimensions

18:88 0.56" Four Digit Display



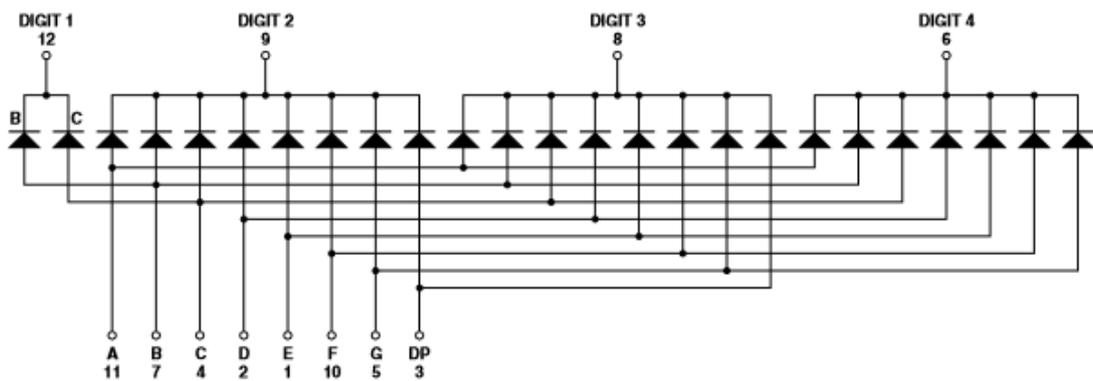
NOTE:
 ALL DIMENSIONS ARE IN MILLIMETERS (INCHES).
 UNLESS OTHERWISE STATED, TOLERANCES ARE ± 0.25 mm.

Pin Configuration

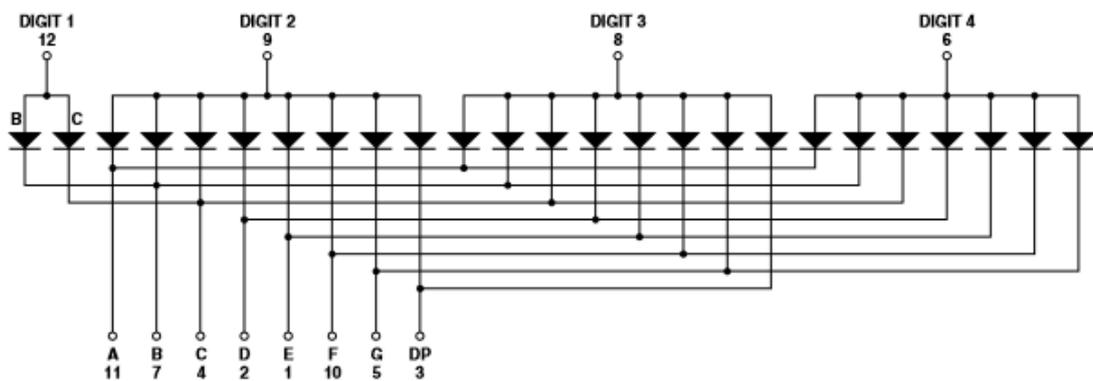
Pin	Function	
	HDSP-B01E/B03E	HDSP-B02E/B04E
1	Anode E	Cathode E
2	Anode D	Cathode D
3	Anode DP	Cathode DP
4	Anode C	Cathode C
5	Anode G	Cathode G
6	Digit 4 Common Cathode	Digit 4 Common Anode
7	Anode B	Cathode B
8	Digit 3 Common Cathode	Digit 3 Common Anode
9	Digit 2 Common Cathode	Digit 2 Common Anode
10	Anode F	Cathode F
11	Anode A	Cathode A
12	Digit 1 Common Cathode	Digit 1 Common Anode

Internal Circuit Diagram

HDSP-B01E (Common Cathode)

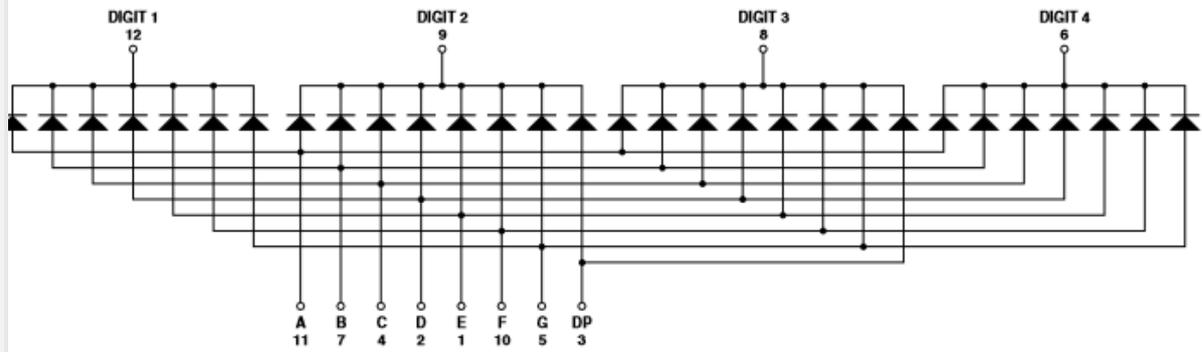


HDSP-B02E (Common Anode)

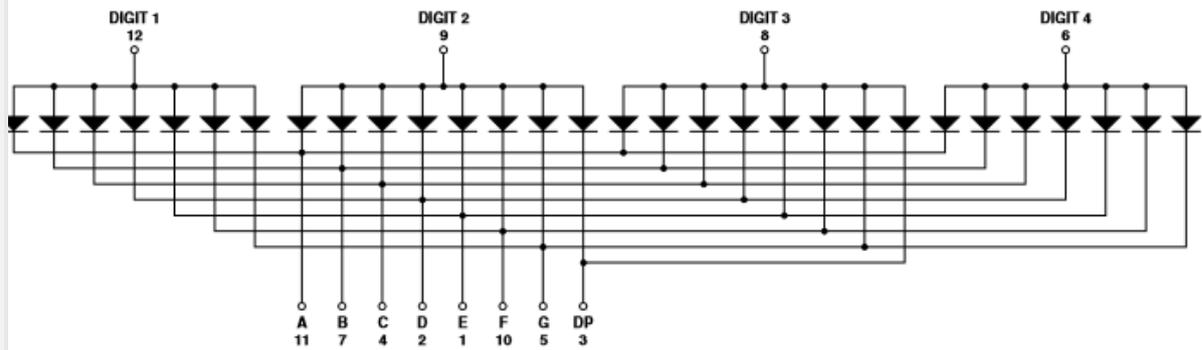


Internal Circuit Diagram

HDSP-B03E (Common Cathode)



HDSP-B04E (Common Anode)



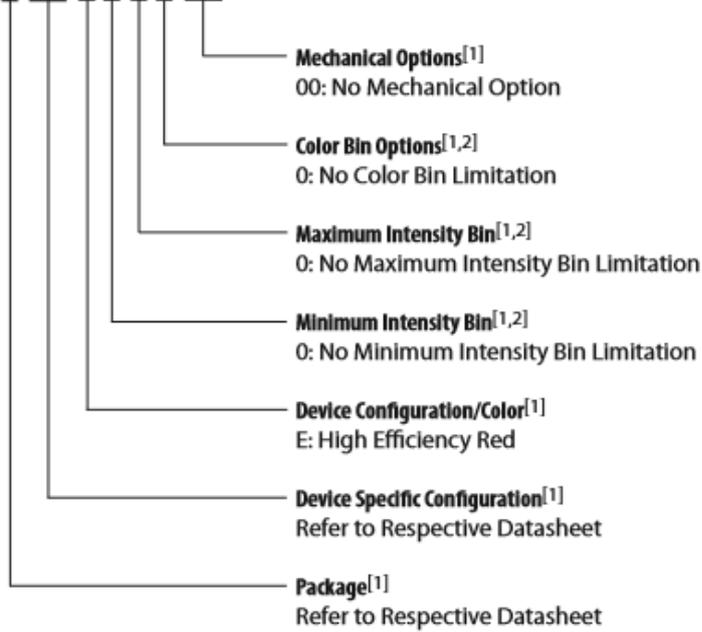
Devices

HDSP-	Description
B01E	88:88 0.56" HER 4 Digit, Untinted, Common Cathode, Display 18:88, Gray Color Surface
B02E	88:88 0.56" HER 4 Digit, Untinted, Common Anode, Display 18:88, Gray Color Surface
B03E	88:88 0.56" HER 4 Digit, Untinted, Common Cathode, Display 88:88, Gray Color Surface
B04E	88:88 0.56" HER 4 Digit, Untinted, Common Anode, Display 88:88, Gray Color Surface

Part Numbering System

5082 -X X X X-X X X X X

HDSP-X X X X-X X X X X



Notes:

1. For codes not listed in the figure above, please refer to the respective datasheet or contact your nearest Avago representative for details.
2. Bin options refer to shippable bins for a part number. Color and Intensity Bins are typically restricted to 1 bin per tube (exceptions may apply). Please refer to respective datasheet for specific bin limit information.

Absolute Maximum Ratings

Description	HER	Units
Average Power per Segment or DP	65	mW
Peak Forward Current per Segment or DP	100	mA
DC Forward Current per Segment or DP	25	mA
Operating Temperature Range	-40 to +105	°C
Storage Temperature Range	-40 to +105	°C
Reverse Voltage per Segment or DP	5	V
Wavesoldering Temperature for 3 seconds 1.59 mm below body	250	°C

Note:

1. Derate above 40°C at 0.33 mA/°C for HER.

Electrical/Optical Characteristics at T_A = 25°C

Device Series	Parameter	Symbol	Min.	Typ.	Max.	Units	Test Conditions
B01E B02E	Luminous Intensity/Segment (Digit Average)	I _v	3200	5500		μcd	I _F = 10 mA
B03E B04E	Forward Voltage/Segment or DP	V _F		2.05	2.60	V	I _F = 20 mA
	Peak Wavelength	λ _{PEAK}		632		nm	I _F = 20 mA
	Dominant Wavelength	λ _d		622		nm	I _F = 20 mA
	Luminous Intensity Matching Ratio	I _{v-M}			2:1		I _F = 10 mA
	Reverse Current	I _R			100	μA	V _R = 5 V

Notes:

1. Typical specification for reference only. Do not exceed absolute maximum ratings.
2. The dominant wavelength, λ_d, is derived from the CIE chromaticity diagram and is that single wavelength which defines the color of the device.

Intensity Bin Limits (Tolerance ± 10%)

Rank	Symbol	Condition	Min.	Max.	Unit
L	I _v	I _F = 10 mA	3200	5050	μcd
M	I _v	I _F = 10 mA	5050	8000	μcd

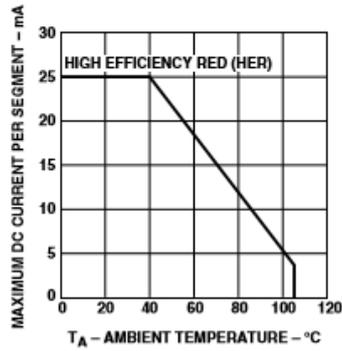


Figure 1. Maximum allowable average or DC current vs. ambient temperature.

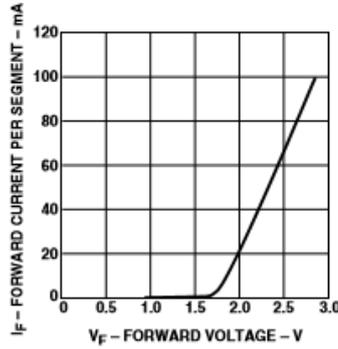


Figure 2. Forward current vs. forward voltage.

Contrast Enhancement

For information on contrast enhancement, please see Application Note 1015.

Soldering/Cleaning

Cleaning agents from the ketone family (acetone, methyl ethyl ketone, etc.) and from the chlorinated hydrocarbon family (methylene chloride, trichloroethylene, carbon tetrachloride, etc.) are not recommended for cleaning LED parts. All these various solvents attack or dissolve the encapsulating epoxies used to form the package of plastic LED parts.

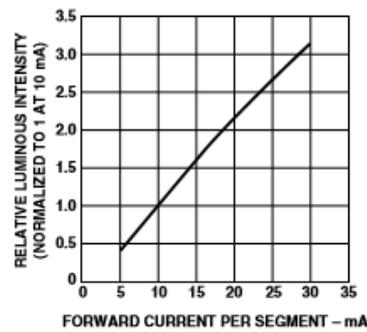


Figure 3. Relative luminous intensity vs. DC forward current.

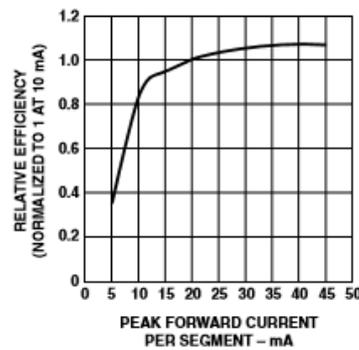


Figure 4. Relative efficiency (luminous intensity per unit current) vs. peak current.

Bibliographie

Arduino

[1]-<http://arduino.cc/en/main/software>

[2]-<http://www.hackaday.com/category/arduino-hacks/>

[3]-<http://www.instructables.com/tag/type-id/category-technology/channel-arduino/>

[4]-<http://www.arduino.cc/>

Afficheur 7 segments

[5]- http://fr.wikipedia.org/wiki/Afficheur_7segments.

[6]-<http://www.Microchip.fr.pdf>.

Maxim (7219/7222)

[7]-www.maxim-ic.com