

Table des matières

Introduction générale.....	2
chapitre I	
I. 1.Introduction.....	4
I. 2.Utilité.....	4
I. 3.Cahier de charge	5
I. 4. Schéma synoptique	6
I. 5. Programmation.....	7
I.6. Structure d'un programme Arduino	7
I. 7 Multiplexage	8
chapitreII	
II.1. Arduino.....	10
II.1.1. Historique	10
II.1.2. Définition.....	10
II.1.3. Constitution d'une carte Arduino UNO	11
II.1.4. Brochage de la carte Uno	13
II.1. 5. Alimentation	13
II.1.6. Mémoires de la carte Arduino	14
II.1.7. Programmation	14
II.18. Programmation des modules Arduino (Shield)	15
II.2. Registres de décalage.....	15
II.2.1. Brochage.....	16
II.2. 2. Principe de fonctionnement	17
II.3. LEDs (Light Emitting Diode)	18
II.3. 1. Caractéristiques électriques principales d'une LED.....	19
II.3.2.Caractéristique optiques.....	19
II.3.3. Présentation d'une matrice à LEDs.....	20
II.3. 4. Le modèle HS-2088A	21
II.4. Module Bluetooth	22
chapitreIII	
III. 1.HARDWARE.....	24
III.1.1. Schéma électrique.....	25

III.1.2. Description de schéma électrique	26
III.1.3. Calcul de la résistance associée avec une LED	26
III.1.4. Conception du circuit imprimé	27
III.1.4.1. Le circuit réel.....	28
III. 2. SOFTWARE	30
III.2. 1. Simulation par PROTEUS(ISIS)	30
III.2. 2. Programmation d'une matrice à LEDs.....	31
III.2.2.1. Le multiplexage	32
III.2.3. Exemple d'affichage.....	33
III.2.3.1.Explication.....	35
III.2.3.. Résultats obtenus	37
III.2.4. Organigrammes	39
Conclusion générale	43

Table des matières des figures

Figure I. 1. Schéma synoptique	5
Figure II. 1. Constitution de la carte Arduino	10
Figure II. 2. Brochage de la carte Arduino	12
Figure II. 3. Présentation du 74HC595	15
Figure II. 4. Constitution d'une LED.....	17
Figure II. 5. La longueur d'onde du spectre	19
Figure II. 6. Structure d'une matrice à LEDs	20
Figure II. 7. Matrice à LEDs modèle B.....	21
Figure II. 8. Le module Bluetooth	22
Figure III. 1. Exemples de caractères	24
Figure III. 2. Schéma électrique.....	26
Figure III. 3. Conception du schéma électrique	27
Figure III. 4. Le circuit imprimé	27
Figure III. 5. Platine	28
Figure III. 6. Le circuit imprimé complet	29
Figure III. 7. Exemple de simulation.....	30
Figure III. 8. Illustration d'un émoticône.....	31
Figure III. 9. Etapes pour afficher l'émoticône	31
Figure III. 10. Exemple d'affichage (émoticône).....	32
Figure III.11. Les différentes étapes effectuées pour afficher le mot 'LED'	37
Figure III. 12. Affichage du mot 'LED'	38
Figure III.13. Organigramme du programme principal.....	40
Figure III.14. Organigramme de procédure affichage fixe	42

Table des matières des tableaux

Table II. 1. Caractéristiques de la carte Arduino UNO	11
Table II. 2. Brochage de 74HC595	15
Table II. 3. Brochage de la matrice HS-2088A	20
Table III. 1 . Structure et codes des caractères	34
Table III. 2. Sorties des registres de décalage	35

CHAPITRE I : Généralités sur les journaux lumineux.

CHAPITRE II : Etude théorique sur les composants du circuit.

CHAPITRE III :Réalisation Pratique.

Résumé

INTRODUCTION GENERALE

CONCLUSION GENERALE

ANNEXES

Introduction générale

Le temps nous a tous apporté une très riche moisson de progrès dans les connaissances techniques et scientifiques ; parmi celles-ci l'électronique, qui intervient aujourd'hui d'une façon de plus en plus importante non seulement dans l'industrie mais dans la vie quotidienne destiné à un large public.

De ce fait, nous avons choisi un projet qui représente un moyen moderne de communication qui a pour objectif la réalisation d'un panneau à LEDs.

Ce dernier, appelé encore '**journal lumineux**', c'est un afficheur électronique qui permet de diffuser les informations en temps réel.

Dans notre projet, nous étudierons les différentes parties qui constituent un panneau à LEDs ainsi les étapes de réalisation.

Pour commander ce panneau nous utiliserons une carte Arduino, géré par son libre interface de programmation. L'affichage d'un texte sera réalisé selon le principe de multiplexage en utilisant les registres à décalage sous la référence 74HC595.

Le premier chapitre est consacré à quelques généralités de base sur le journal lumineux, son utilisation ainsi l'élaboration de notre cahier de charges.

Dans le deuxième chapitre, nous aborderons l'étude théorique des différents compartiments (étages) qui entrent en jeu dans notre réalisation.

Le dernier chapitre, est consacré au développement du code Arduino (software) ainsi que les cartes électroniques (hardware) nécessaires pour la réalisation du circuit.

I.1. Introduction

Un panneau lumineux est une véritable borne d'information, celui-ci permet de diffuser et informer, en intérieur comme en extérieur, toutes les informations destinées au public (informations routières, communales,...etc.). Parmi ses options, la possibilité d'appliquer un défilement qui est utile pour des opérations commerciales et publicitaires.

Un journal lumineux est un panneau électronique composé de LEDs, de différentes formes, sa taille varie en fonction de plusieurs paramètres (le lieu, le coût et la fonctionnalité). Le message qui circule sur ce journal lumineux peut être statique, ou bien dynamique comme dans le cas d'un affichage de température ou d'une date.

I.2. Utilité

Pour faire passer vos messages avec un impact, il faut choisir les bons outils de communication.

Les panneaux lumineux possèdent plusieurs performances (conception, dimensions et technologies) ce qui rend l'information plus claire, lisible, valorisante et facilement modulable. Généralement ils servent à effectuer plusieurs fonctionnalités telles que l'accueil, orientation, information, sensibilisation et la communication.

Citons parmi ses utilisations :

- Le journal lumineux permet d'informer les gens de manière rapide et efficace, que ce soit sur des informations municipales, des événements culturels, des manifestations sportives ou des rencontres associatives telles que l'affichage des scores lors d'un match.
- Les villes et collectivités locales qui ont besoin de transmettre des informations pratiques à leurs administrés.

Chapitre I - Généralités sur les journaux lumineux.

- Les sociétés de transports en commun (les aéroports, les gares), utilisent les panneaux lumineux pour transmettre des informations en temps réelles à leurs usagers (temps d'attente, provenance, destination... etc.).
- Ils peuvent également communiquer sur des spectacles locaux
- Il peut être situé au niveau de chaque structure tel que :
 - les Hôtels, les restaurants, les cafétérias.
 - les hôpitaux.
 - les magasins.
 - les universités.

I.3. Cahier de charges

L'objectif principal est de réaliser un journal lumineux qui affiche des messages fixes ou défilants selon une vitesse souhaitée. L'afficheur est ainsi constitué de 768 LEDs dont 8 lignes et 96 colonnes.

L'affichage utilise la technique du multiplexage ; une seule des 8 lignes est alimentée, et le balayage rapide et cyclique des lignes permet d'obtenir un affichage lumineux continu.

Par la suite Nous proposons de réaliser des effets lumineux qui se succèdent automatiquement tel que :

- Le défilement de la gauche vers la droite ou de la droite vers la gauche.
- L'affichage fixe.

I.4. Schéma synoptique

Afin de simplifier la conception du circuit électrique nous proposons un schéma synoptique illustré dans la figure I.1.

Les différents modules qui entrent dans la réalisation de ce système d'affichage sont:

- L'alimentation qui fournit l'énergie nécessaire au système pour son fonctionnement.
- Le périphérique d'entrée (modules Bluetooth) qui permet de recevoir les consignes de l'utilisateur.
- La carte Arduino qui commande l'affichage du journal.

- Le circuit d'interface (registres à décalage et résistances) qui permet d'optimiser le nombre de pins utilisés par l'Arduino.
- Les matrices à LEDs qui affiche les caractères (latin, arabe, signes...).

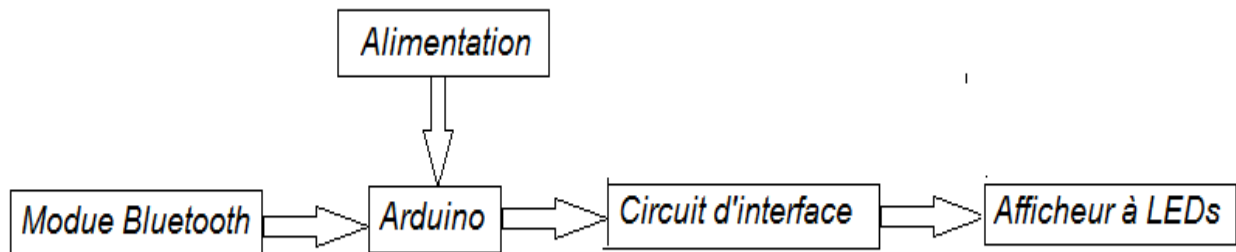


Figure I.1. Schéma synoptique

I. 5. Programmation

Un programme permet de traduire le cahier des charges en une suite ordonnée d'actions que doit réaliser le processus de commande, il se base sur un algorithme qui est une procédure composée d'une séquence d'opérations qui sera traduite en instructions élémentaires. Par la suite, il suffit de transformer ces actions en un langage évolué tel que le langage java ou le langage C [7,8].

Un langage de programmation permet d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur [18].

Le logiciel Arduino, basé sur le langage C++, possède une bibliothèque de développement riche. L'exécution s'effectue d'une manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

I.6. Structure d'un programme Arduino

Le programme Arduino est composé de 3 parties principales :

1. la partie optionnelle : déclaration des constantes et des variables.
2. la partie initialisation et configuration des entrées/sorties dans la fonction **setup ()**.
3. la partie principale qui s'exécute en boucle (la fonction **loop**), elle est réservée aux instructions à effectuer par le programme [5,1].

I.7. Multiplexage

Multiplexer est une technique très efficace pour contrôler plusieurs composants câblés dans une matrice [15]. Dans notre cas, nous allons parler de l'affichage multiplexé, mais les mêmes principes de base sont applicables à d'autres composants multiplexés (capteurs, boutons, etc.).

Dans un réseau multiplexé de LEDs, à un moment donné, une seule rangée est activée.

Alors, comment pouvons-nous envoyer des données à une ligne à la fois?

Pour répondre à cette question, il faut mettre une seule ligne à l'état haut (5 V) et les autres à l'état bas (la masse) et défiler chaque ligne, une après l'autre.

Par la suite, pour activer les LEDs, il suffit de brancher les cathodes à la masse [4].

II.1. Arduino

Au fil des années, l'Arduino est devenu un élément clé de milliers de projets, ces derniers varient du plus simple au plus complexe. De ce fait, une large communauté profite de cette plateforme à source libre. Il représente un pont tendu entre le monde réel et le monde numérique, et permet d'étendre les capacités de relations humaines/machines ou environnement/machine [1].

II.1.1. Historique

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie). Ils rencontraient un problème majeur à cette période (2003-2004) ; les outils nécessaires à la création de projet d'interactivité étaient complexes et onéreux (entre 80 et 100 euros). Les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Leur préoccupation se concentre alors sur la réalisation d'un matériel moins cher et plus facile à utiliser [2].

En 2003, Hernando Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée « Wiring », accompagnée d'un environnement de programmation libre et ouvert. Cette carte a donc inspiré le projet Arduino (2005) et conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Caurtielles, Massimo Banzi et Nicholas Zambetti) [3].

II.1.2. Définition

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif, c'est un circuit imprimé en matériel libre (dont les plans sont publiés en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électrique, de manière d'effectuer des tâches diverses. Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe.

II.1.3 . Constitution d'une carte Arduino UNO

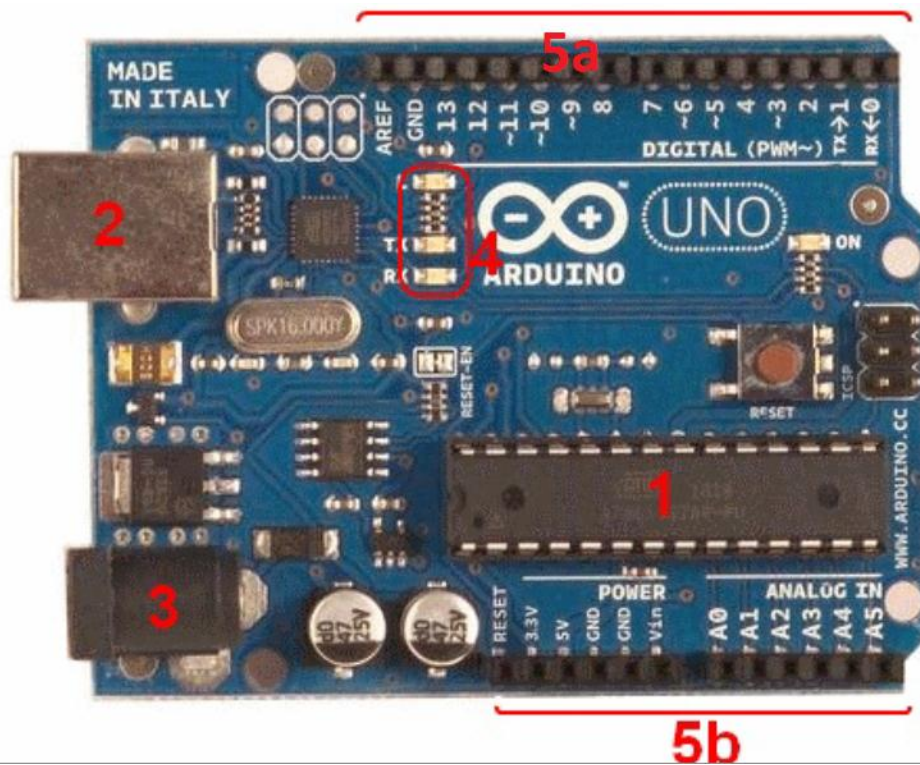


Figure II.1. Constitution de la carte Arduino.

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle possède tout ce qui est nécessaire pour le bon fonctionnement du microcontrôleur. Pour pouvoir l'utiliser il suffit de la connecter à un ordinateur à l'aide d'un câble USB [3].

Chapitre II - Etude théorique sur les composants du circuit.

Synthèse des caractéristiques

La table II.1 résume les caractéristiques d'une carte Arduino Uno

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

Table II.1. Caractéristiques de la carte Arduino UNO [16]

II.1.4. Brochage de la carte Uno

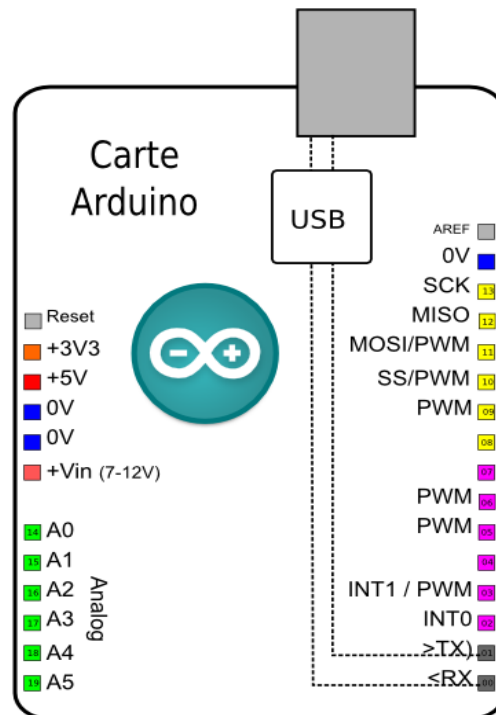


Figure II.2. Brochage de la carte Arduino [16]

II.1.5. Alimentation

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte [2].

Les broches d'alimentation sont les suivantes:

- La borne V_{IN} : représente La tension d'entrée positive lorsque la carte Arduino est alimentée par une source de tension externe.
- La borne de 5V : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte.
- GND : Broche de masse (ou 0V).

Chapitre II - Etude théorique sur les composants du circuit.

- La borne de +3V3 : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de l'ordinateur et le port série de l'ATmega). Ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA [17].

II.1.6. Mémoires de la carte Arduino

Les 3 types de mémoires qui existent au sein d'une Arduino Uno à base ATmega328 sont :

- La mémoire **FLASH** : c'est une mémoire volatile, sa capacité est de 32 kilooctets, elle est rapide [6].
- La mémoire **SRAM** (Static Read Access Memory) : C'est une mémoire volatile. Sa capacité est de 2 kilooctets, elle est très rapide [6].
- La mémoire **EEPROM** : (Electrically-Erasable Programmable Read-Only Memory) c'est une mémoire non volatile, sa capacité est de 1 kilooctets, elle est lente. [6].

II.1.7. Programmation

La carte Arduino Uno est programmée avec le **logiciel Arduino**. Il suffit de sélectionner "Arduino Uno" dans le menu Tools >Board (en fonction du microcontrôleur présent sur la carte) [7].

Le microcontrôleur ATmega328 est livré avec un bootloader (petit programme de démarrage) préprogrammé qui permet de transférer le nouveau programme dans le microcontrôleur en utilisant le Protocole original STK500, sans avoir à utiliser un matériel de programmation externe. Nous pouvons en outre, injecter le programme via le connecteur ICSP (In-Circuit Serial Programming, "Programmation Série Dans le circuit" en français) [17].

II.1. 8. Programmation des modules Arduino (Shield)

Le logiciel de programmation des modules Arduino est une application Java, il sert à la fois d'un éditeur de code et d'un compilateur, il peut transférer le firmware et le programme à travers la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler les programmes via l'interface en ligne de commande [11,14].

II.2. Registres de décalage

L'élément principal dans le circuit d'interface est les registres de décalage 74HC595, ce dernier est un système logique permettant la mémorisation des données depuis une connexion série via une interface série. Il est composé d'un ensemble de bascules de type D misent en cascade, à chaque front (montant ou descendant) l'état de ces bascules est mise à jour [18].

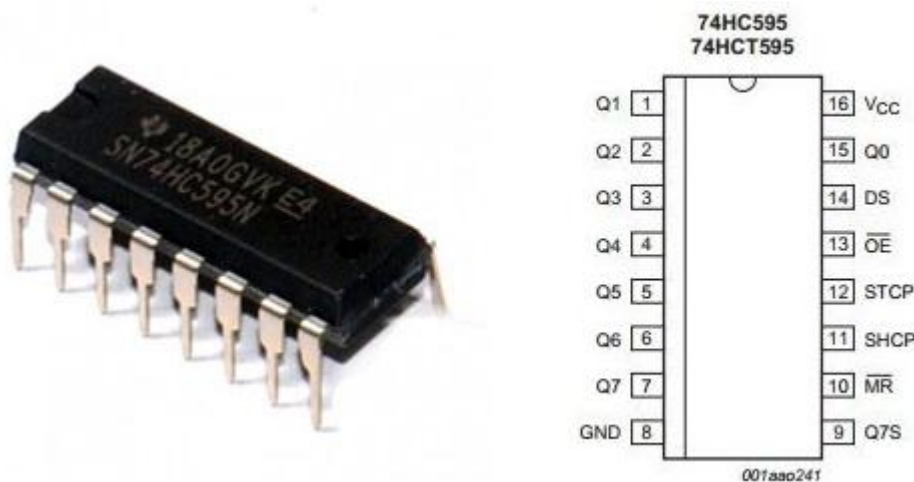


Figure II.3. Présentation du 74HC595

Chapitre II - Etude théorique sur les composants du circuit.

II.2.1. Brochage

Le registre de décalage 74HC595 se présente sous la forme d'un petit Circuit intégré (empaqueté DIP16), il possède 16 pattes.

Les sorties sont : les broches de 1 à 7 et la broche 15 (Q0).

L'entrée des données série qui commande les sorties du composant se trouve sur la broche 14 (serial data input).

La patte 13 (Output enable) est une broche de sélection qui permet d'autoriser les sorties ; cela signifie que lorsqu'elle n'est pas à l'état logique requis (niveau BAS), les sorties du 74HC595 ne seront pas utilisables. Sur la broche 10 il y a le Master Reset(MR) qui a pour fonction de mettre à zéro toutes les sorties, elle est active à l'état BAS.

Les broches 11 et 12 sont des 'horloges'.

Le tableau ci-dessous résume la fonctionnalité de chaque pin

Pins	Symbole	Description
1-7	Q1 à Q7	Les sorties parallèles 1-7.
15	Q0	Sortie parallèle 0.
9	Q7 ‘	Sortie série.
10	MR	Remise à zéro (active au niveau bas).
11	SH_CP	Horloge de décalage (clock shift).
12	ST_CP	Stockage (latch).
13	OE	Sorties actives (active au niveau bas).
14	DS	Entrée série.
8	GND	La masse.
16	Vcc	Alimentation.

Table II.2. Brochage de 74HC595 [19]

II.2. 2. Principe de fonctionnement

Le registre série-parallèle permet de décaler vers la droite des informations en les appliquant une à une sur l'entrée série (DS) qui travaille en corrélation avec l'horloge (SH_CP), c'est-à-dire la première impulsion transfère l'information de l'entrée à la sortie de la première bascule, la seconde la transmet à la sortie de la deuxième bascule et ainsi de suite jusqu'à la huitième. L'information se décale donc en se propageant de l'entrée de la première bascule à la sortie de la huitième bascule au bout de huit impulsions d'horloge [13].

La seconde horloge sert à déterminer si le composant doit mettre à jour les états de ses sorties ou non, en fonction de l'ordre qui est transmis. Lorsque ce signal passe de l'état BAS à l'état HAUT, le composant change les niveaux logiques de ses sorties en fonction des bits de données reçues. En clair, il copie les huit derniers bits transmis sur ses sorties

Le pin 9 (sortie série) est utilisé pour chainer les 74HC595.

II.3. LEDs (Light Emitting Diode)

On appelle électroluminescente l'émission d'un rayonnement lumineux due à une excitation électronique dans un matériau. Il s'agit de l'émission spontanée de lumière provoquée par l'injection des électrons à travers une jonction PN particulière polarisée en direct. Les matériaux utilisés pour convertir l'énergie électrique en énergie lumineuse sont les semi-conducteurs. La tension de seuil de la LED est supérieure à celle d'une diode classique (environ 1,6 V pour l'arséniure de gallium). Elle est en relation directe avec l'énergie du photon émis [9,10].

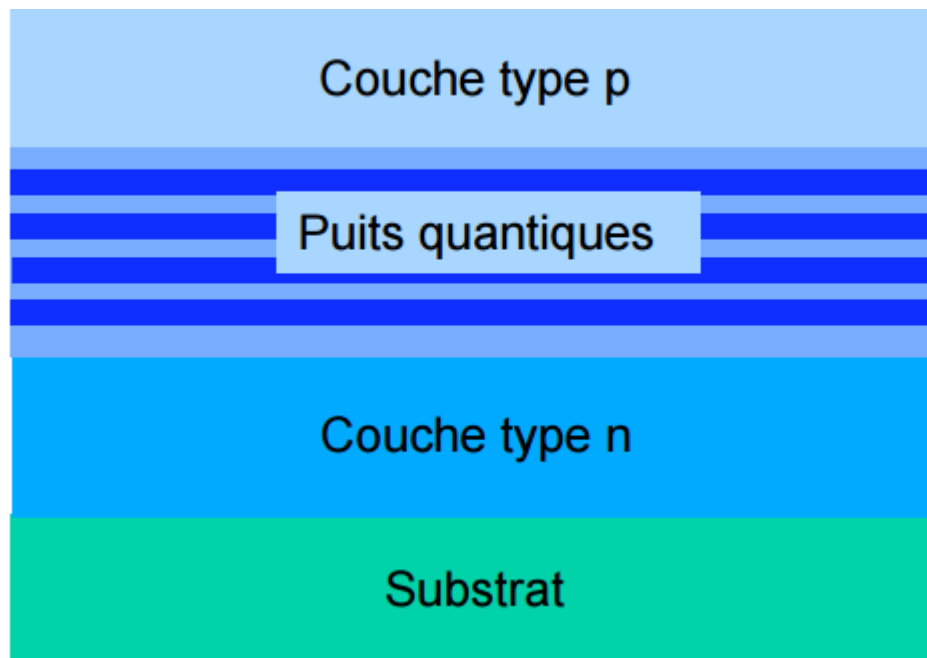


Figure II. 4. Constitution d'une LED [20]

II.3.1. Caractéristiques électriques principales d'une LED

La LED possède un comportement électrique sensiblement identique à celui d'une jonction PN. Son seuil de conduction V_{d0} est cependant plus élevé et dépend de la longueur d'onde lumineuse dominante [12].

Les autres caractéristiques électriques importantes sont :

- Le courant direct moyen qu'elle peut supporter en permanence. Il est compris entre 5 et 20 mA pour une LED à usage générale.
- Le courant direct crête qu'elle peut supporter en régime impulsionnel.
- La tension inverse maximale V_{Rmax} admissible sans dommage. Vis-à-vis de la tension inverse ; la LED est plus fragile qu'une jonction PN et la limite courante de V_{Rmax} est de 5 volts.

II.3.2. Caractéristique optiques

En pratique, les caractéristiques optiques que l'on doit prendre en compte pour choisir une LED sont :

- **Correspondance couleurs, longueurs d'onde et énergie des photons :** La couleur liée à la longueur d'onde dominante du spectre d'émission lumineuse ; On trouve des LEDs à usage générale dans les teintes bleues, verte, jaune, orange, rouge et proche infrarouge. On construit actuellement des LEDs à émission lumineuse.
- **Longueur d'onde du pic d'émission :** Cette valeur nous indique la longueur d'onde (λ), en nanomètre, à laquelle est émis la plus importante partie du rayonnement (wavelength).
- **Spectre ou largeur spectrale à mi-intensité:** Le spectre d'émission d'une diode LED est relativement étroit. Il existe actuellement plusieurs types de LEDs donnant chacun des spectres différents. Cela est obtenu par la variété des semi-conducteurs utilisés pour fabriquer les jonctions PN.

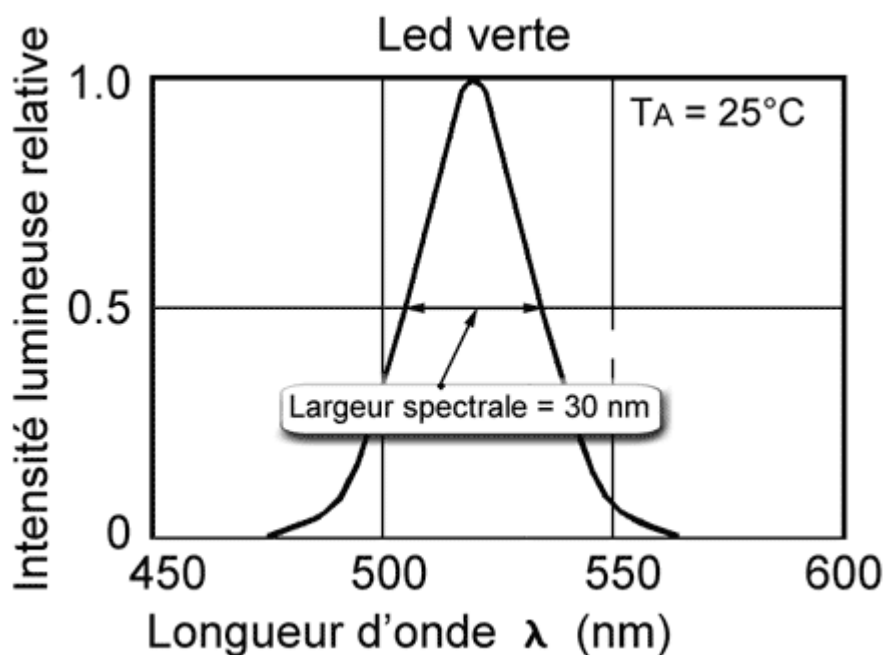


Figure II.5. La longueur d'onde du spectre [21]

- **Diagramme de rayonnement:** Le flux lumineux n'est pas homogène tout autour de la LED. La répartition spatiale de la puissance émise dépend de la forme de la diode LED.
- **Angle d'émission à mi-intensité:** Les fabricants précisent souvent l'angle pour lequel l'intensité lumineuse a été réduite de moitié.
- **Intensité lumineuse:** L'intensité lumineuse (mesurée en candelas) est la quantité de lumière émise dans une certaine direction à 1 mètre de distance [21].

II.3.3. Présentation d'une matrice à LEDs

Une matrice à LEDs est un arrangement particulier d'un ensemble de LEDs pour constituer un tableau où on peut activer chaque LED indépendamment.

Les LEDs sont rangées en lignes et colonnes. Il existe deux types de matrice :

- Matrices à anodes communes.
- Matrices à cathodes communes.

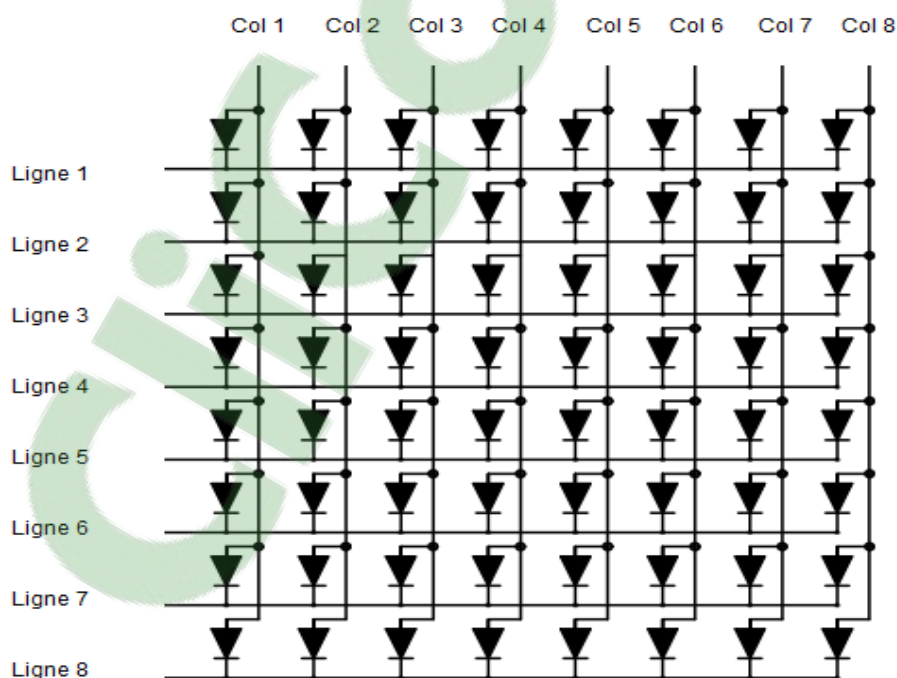


Figure II. 6. Structure d'une matrice à LEDs

Avec Col représente les Colonnes.

Chapitre II - Etude théorique sur les composants du circuit.

II.3.4. Le modèle HS-2088A

Le modèle des matrices à LEDs que nous allons utiliser est HS-2088A (figure II. 7) est une matrice 8x8 (64 LEDs) utilise au total 16 entrées (8 pour les lignes et 8 pour les colonnes) ; mais elle est construit d'une manière spécifique ; la table II.3 décrit son brochage :

Pin conceptuel	1	2	3	4	5	6	7	8
Pin physique	5 L	7 L	2 C	3 C	8 L	6 L	6 L	3 L

9	10	11	12	13	14	15	16
1 L	4 C	6 C	4 L	1 C	2 L	7 C	8 C

Table II.3. Brochage de la matrice HS-2088A

Avec L représente les lignes et C représente les colonnes.

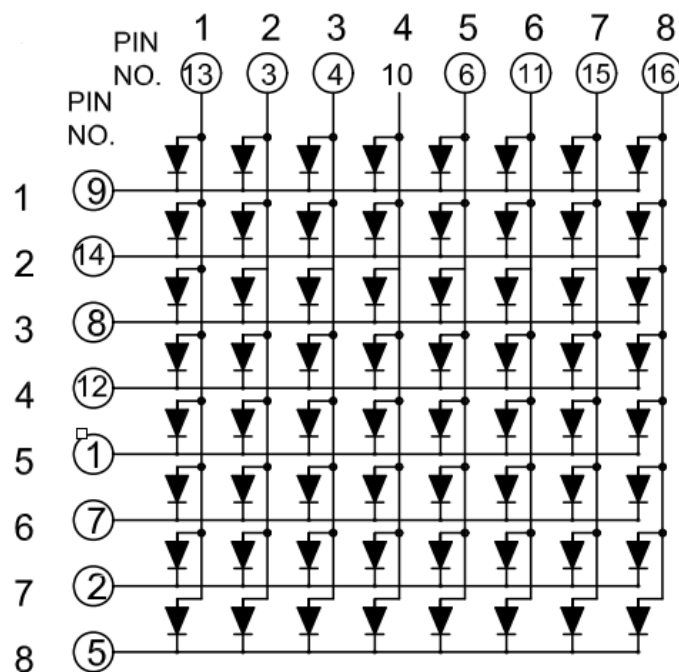


Figure II.7. Matrice à LEDs modèle B

II.4.Module Bluetooth

Bluetooth est un outil de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF. Son objectif est de simplifier les connexions entre les appareils en supprimant les liaisons câblées.

Une communication Bluetooth s'appuie sur un émetteur-récepteur, la bande de fréquence utilisée est l'ISM (Industrial, Scientific and Material) réservée pour l'industriel et la science.

Au niveau mondial la bande de fréquences s'étend sur 83,5 MHz (de 2,4 à 2,4835 GHz). L'émission à cette fréquence est libre (il n'est pas nécessaire d'acheter la licence).

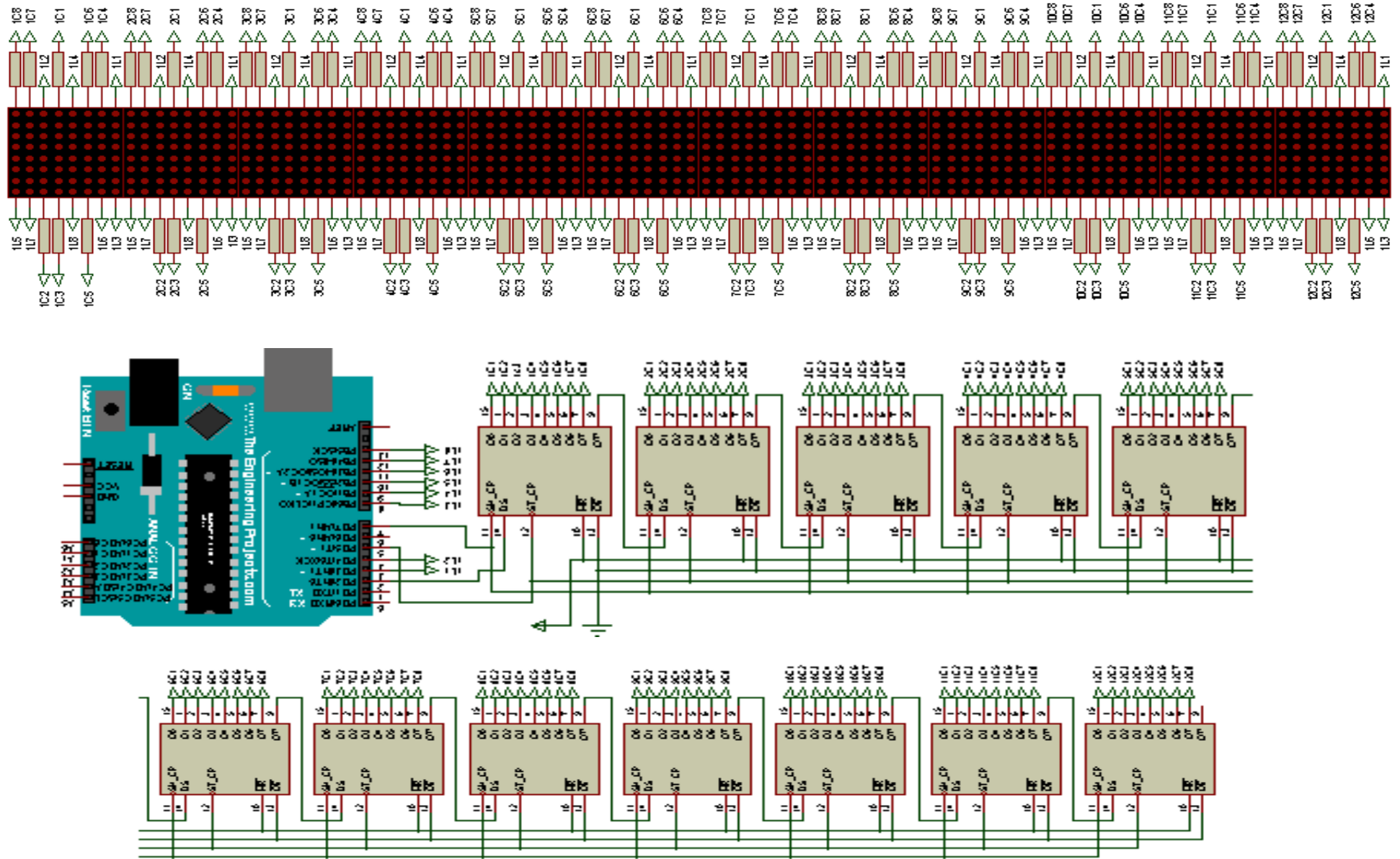
Un module Bluetooth contient un contrôleur de liaisons : Il encode et décode les paquets Bluetooth selon la charge utile et les paramètres liés au canal physique [14].

Le module HC-05, il est représenté dans la figure suivante :



Figure II.8. Le module Bluetooth

Chapitre III : Réalisation pratique



III.1.HARDWARE

Dans cette partie nous allons décrire les méthodes adoptées pour répondre aux spécifications de notre cahier de charge, en abordant la conception détaillée de chaque partie du circuit afin d'obtenir une schématisation complète et précise.

Notre type d'afficheur possède 64 LEDs en forme de points, disposés selon une matrice de 8 lignes et 8 colonnes, l'ensemble de matrices forme un journal défilant dont sa capacité d'affichage est limitée à 147 caractères.

Les caractères latins sont codés par 8 lignes et 6 colonnes et les autres sont codés par des matrices de 8x8 LEDs.

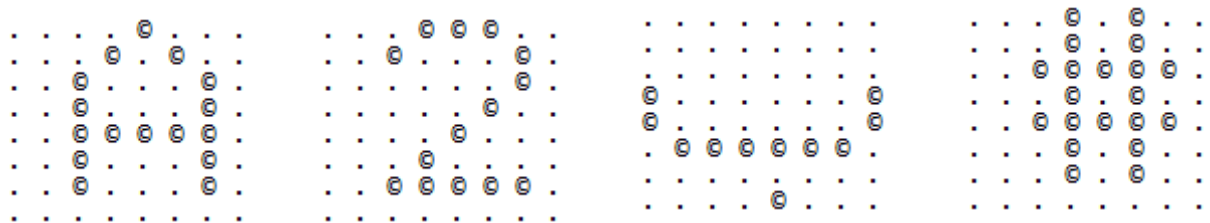


Figure III.1. Exemples de caractères

III.1.1. Description de schéma électrique

Pour faire fonctionner le circuit il est nécessaire d'utiliser une alimentation. Les broches d'alimentation sont les broches 16 (VCC) et 8 (masse) et la tension à appliquer entre 2V et 5.5 V. (Voir la table II.2).

La carte Arduino utilise 3 sorties numériques afin de transmettre les données via les registres à décalage.

Nous avons pris 8 pins (3, 4, 8, 9, 10, 11, 12, 13) de l'Arduino pour relier les lignes.

Chapitre III - Réalisation pratique

Les registres à décalage sont montés en cascade c'est-à-dire les données série provenant de la carte Arduino (pin 2) arrivent à l'entrée DS (Serial Data Input) du premier 74HC595 et sa sortie Q7' (Serial Data Output) est connectée à l'entrée DS du second, et ainsi de suite.

L'horloge des registres 74HC595 (broche SH-CP : Shift register clock input) est générée par l'Arduino (pin 7).

Le pin 5 de l'Arduino est relié aux entrées ST-CP (Storage register clock input) des registres 74HC595.

III.1.2. Calcul de la résistance associée avec une LED

Le courant maximum fourni par chaque broche de l'Arduino ne doit pas excéder 40 mA (table II.1). Chaque colonne reliée à une sortie de registre à décalage alimentant les huit LEDs, le courant maximum passant par une LED doit être moins d'un quart de 40 mA, soit entre 5 et 10 mA.

Chaque LED rouge dans la matrice a une tension directe d'environ 1.8 V. En calculant la résistance nécessaire avec une tension d'alimentation de 5V.

La tension aux bornes de R doit être de $5 - 1,8 = 3,2$ V.

Selon la loi d'ohm ($V = R * I$)

$$R = 3,2 / 0.01 = 320 \text{ ohm.}$$

Mais nous devons utiliser les valeurs normalisées, donc nous plaçons une résistance de 220 ohms en série avec chacune des colonnes, pour que celle-ci puisse fonctionner correctement sous une tension d'alimentation de 5 volts.

III.1.4. Conception du circuit imprimé

Notre circuit imprimé composé de 3 parties ; chacune contient 4 matrices à LEDs.

Par la suite, nous allons décrire la conception d'une seule partie (étage d'entrée). Les deux autres circuits imprimés sont identiques.

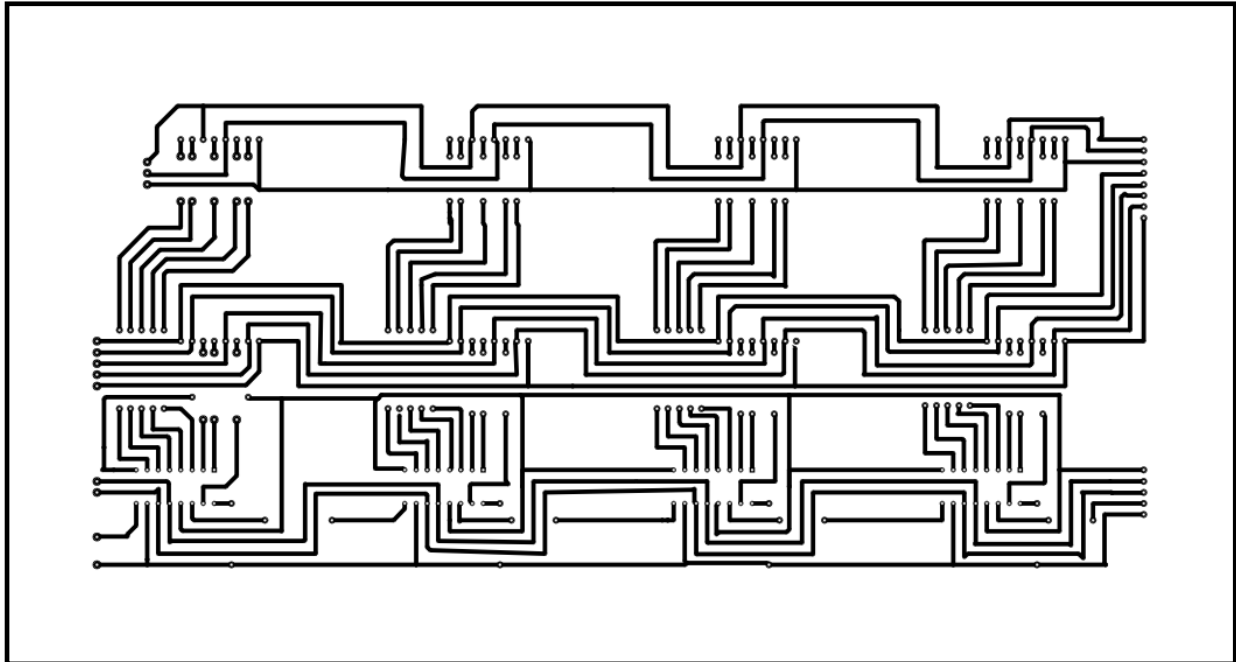


Figure III.3. Conception du schéma électrique

III.1.4.1. Le circuit réel

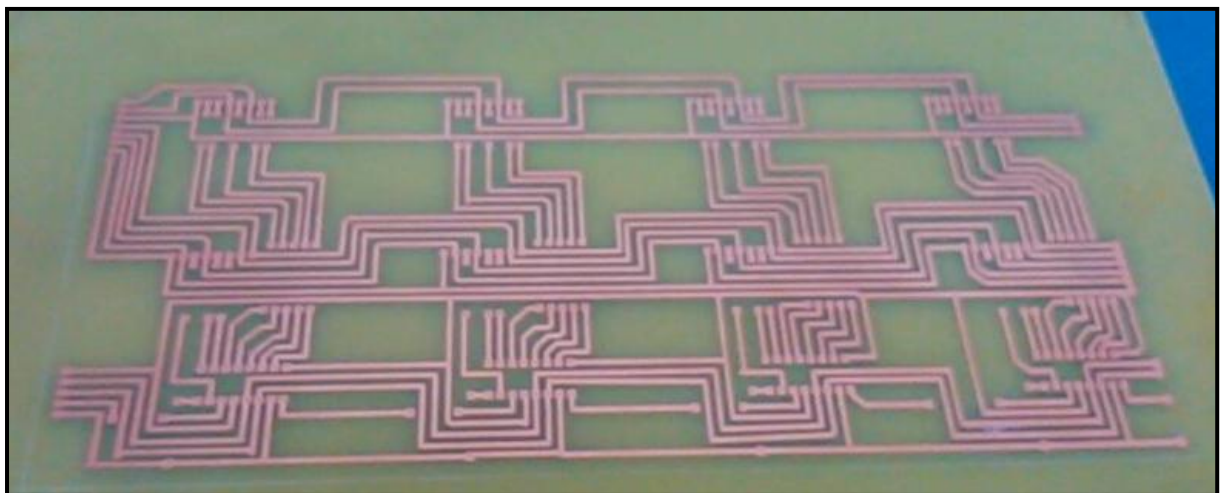


Figure III.4. Le circuit imprimé

Après l'implantation de composants, nous avons obtenu le circuit suivant (figure III5):

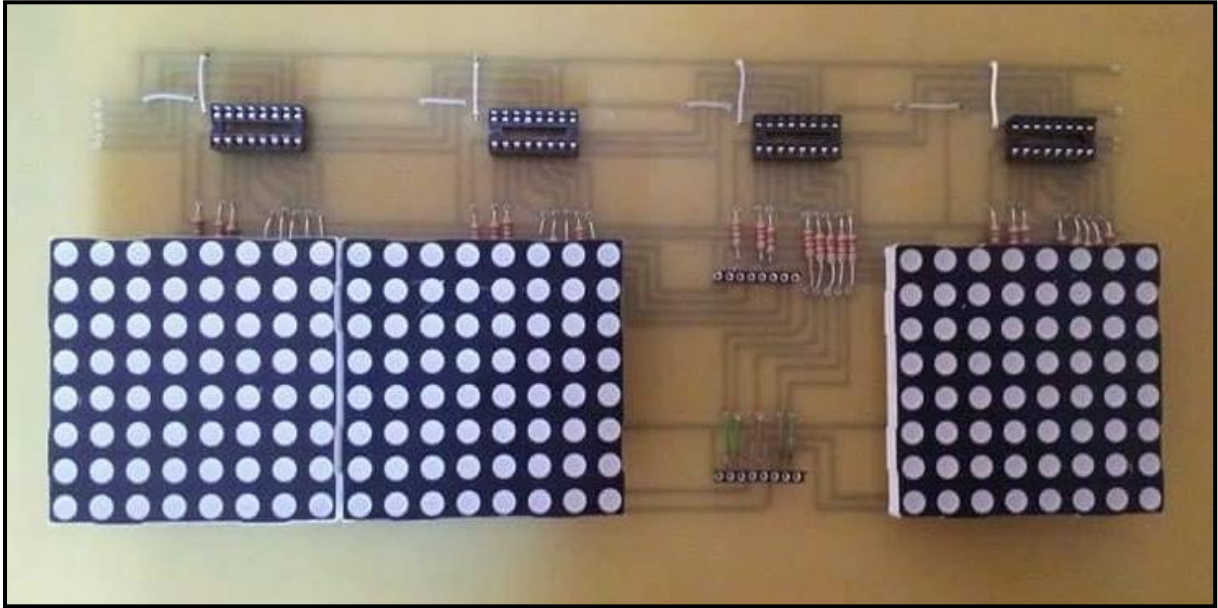


Figure III.5. Platine

Après la conception des trois circuits imprimés, nous allons les intégrer pour avoir un seul circuit. Le circuit final est illustré dans la figure ci-dessous.



Figure III.6. Le circuit imprimé complet

III. 2. SOFTWARE

Comme c'est déjà noté, notre projet se divise essentiellement en deux parties, simulation et réalisation.

III.2. 1. Simulation par PROTEUS (ISIS)

Avant de passer à la réalisation pratique, il était nécessaire de simuler les différentes parties du système, pour cela nous avons utilisé le logiciel « PROTEUS », qui nous a permis d'ajuster et modifier le circuit afin d'accélérer le prototypage.

La figure suivante présente un exemple de simulation de six matrices :

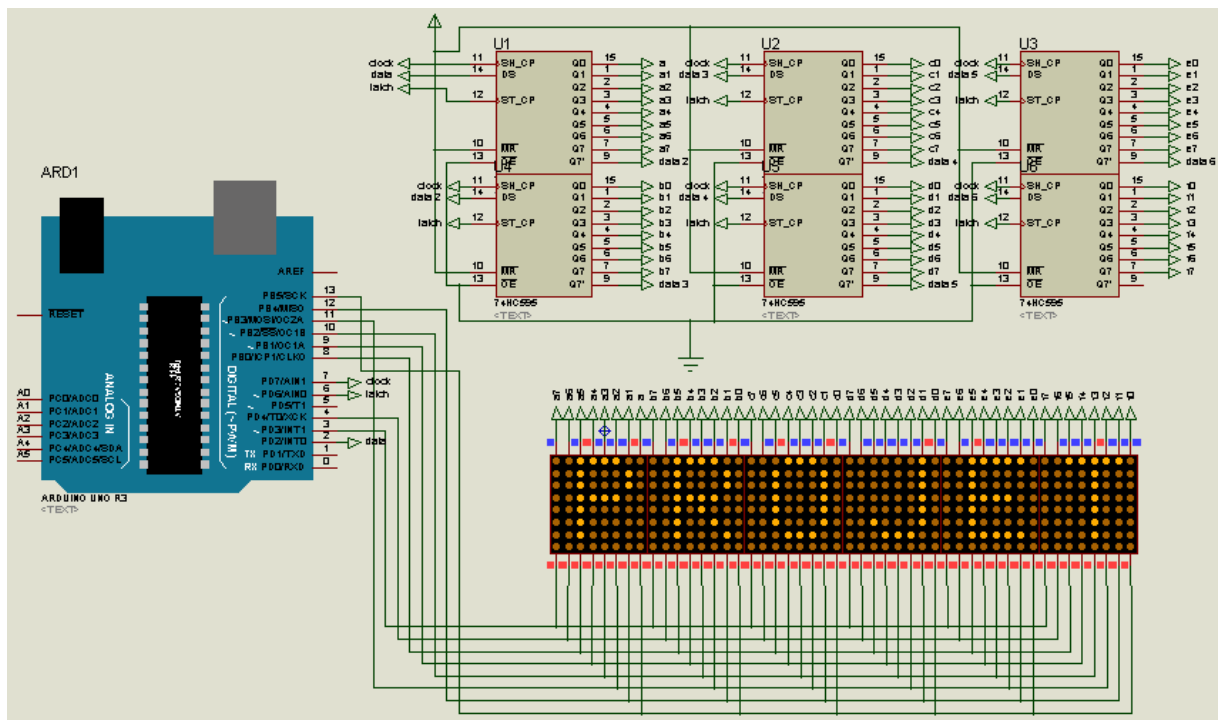


Figure III.7. Exemple de simulation

III.2. 2. Programmation d'une matrice à LEDs (anodes communes)

Pour un premier essai nous allons afficher un émoticône souriant, pour cela il faut tout d'abord introduire le tableau des constantes suivant;

Byte Souriant [8] = {0b11111111, 0b10011001, 0b10011001, 0b11111111, 0b11100111, 0b10111101, 0b11000011, 0b11111111};

Byte : une unité d'adressage varie de (0-255).

Souriant: le nom de tableau (c'est un identificateur).

[8] : la taille de tableau.

0bXXXXXXXX : « 0b » signifie que la variable est en binaire (X peut avoir 2 valeurs: 0 ou 1).

Les colonnes (cathodes) de la matrice sont connectées à la sortie des registres à décalage, donc une colonne doit être mise à l'état BAS pour pouvoir allumer une LED.

Les lignes (anodes) sont connectées aux pins de l'Arduino, donc une ligne doit être mise à l'état HAUT pour que celle-ci fonctionne. Si la ligne et la colonne sont toutes les deux aux mêmes niveaux (état HAUT ou BAS), la LED ne s'allume pas.

Pour contrôler plusieurs LEDs sur une même ligne, il faut mettre celle-ci à l'état HAUT, et mettre les colonnes à l'état BAS en allumant les LEDs désirées.

III.2.2.1. Le multiplexage

Pour réaliser l'exemple précédent (émoticône souriant) il faut effectuer plusieurs étapes :

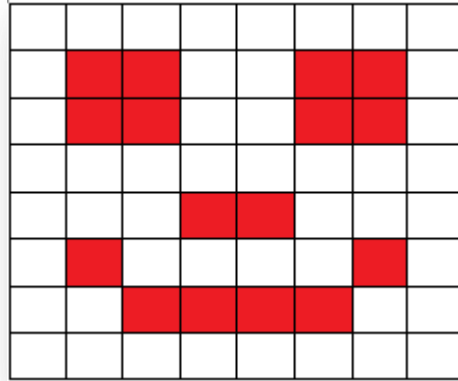


Figure III. 8. Illustration d'un émoticône

Les étapes nécessaires pour afficher l'émoticône sont décrites dans la figure III.9 :

étape-1-	étape-2-	étape-3-	étape-4-
étape-5-	étape-6-	étape-7-	étape-8-

Figure III.9. Etapes pour afficher l'émoticône

Les huit étapes montrent qu'on active une seule ligne à la fois; les lignes sont activées les unes après les autres, et le balayage se fait rapidement, à une fréquence de **125 Hz**.

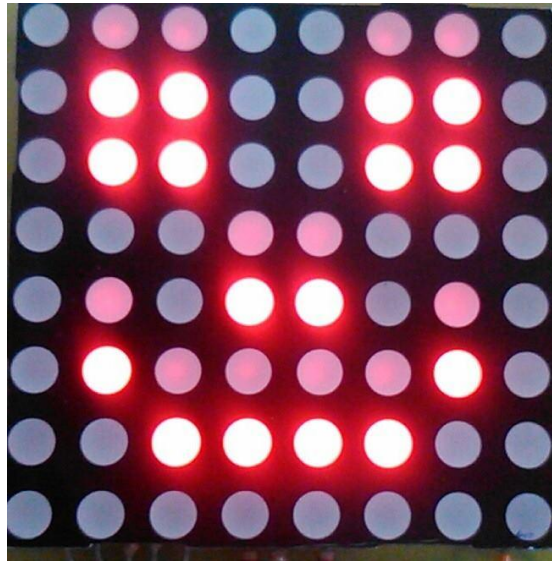


Figure III.10. Exemple d’affichage (émoticône)

III.2. 3. Exemple d’affichage

Nous voulons afficher le mot « LED » dans notre journal lumineux, utilisant trois matrices au maximum, les caractères « L », « E », « D » et « Blanc » sont codés selon des matrices de 8 lignes et 6 colonnes, nous allons décrire la méthode à suivre :

Chaque LED de la matrice ne peut avoir que deux états (éteinte ou allumée), tout comme un bit ('1' ou '0').

Partant de ce principe, il faut faire correspondre les bits de commande (HAUT ou BAS) sur les LEDs, pour cela nous allons nous servir d'un tableau de bits, dans lequel nous allons stocker les données ligne par ligne les états souhaité des LEDs, une fois les caractères sont dessinés (littéralement), il nous faut les envoyer vers les registres à décalage, le tableau suivant (tableau III.1) montre en détails la structure et les codes des caractères.

Pour cet exemple, il nous faut trois registres de décalage et trois matrices à LEDs.

Chapitre III - Réalisation pratique

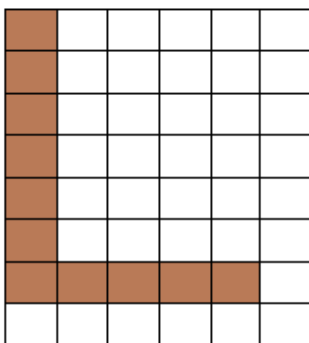
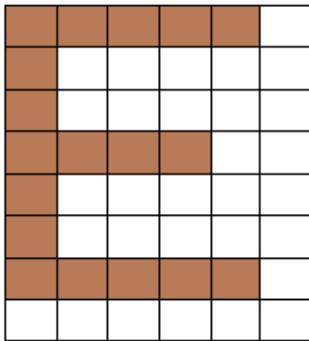
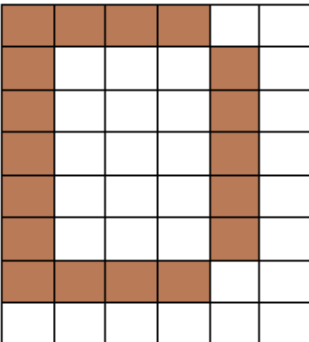
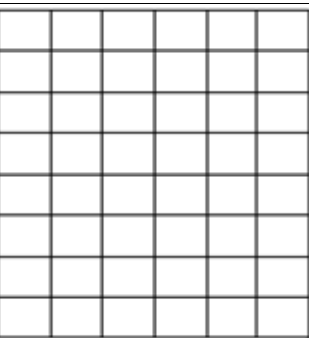
Caractère	Code	Codes en binaire et en hexadé- cimal																																																								
	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	<table><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 000001 = 0x01</td></tr><tr><td>0b 111111 = 0x3F</td></tr></table>	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 000001 = 0x01	0b 111111 = 0x3F
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	0	0	0	0	1																																																					
1	1	1	1	1	1																																																					
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 000001 = 0x01																																																										
0b 111111 = 0x3F																																																										
	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	<table><tr><td>0b 000001 = 0x01</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 000011 = 0x03</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 011111 = 0x1F</td></tr><tr><td>0b 000001 = 0x01</td></tr><tr><td>0b 111111 = 0x3F</td></tr></table>	0b 000001 = 0x01	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 000011 = 0x03	0b 011111 = 0x1F	0b 011111 = 0x1F	0b 000001 = 0x01	0b 111111 = 0x3F
0	0	0	0	0	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	0	0	0	1	1																																																					
0	1	1	1	1	1																																																					
0	1	1	1	1	1																																																					
0	0	0	0	0	1																																																					
1	1	1	1	1	1																																																					
0b 000001 = 0x01																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 000011 = 0x03																																																										
0b 011111 = 0x1F																																																										
0b 011111 = 0x1F																																																										
0b 000001 = 0x01																																																										
0b 111111 = 0x3F																																																										
	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	0	1	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	<table><tr><td>0b 000011 = 0x03</td></tr><tr><td>0b 011101 = 0x1D</td></tr><tr><td>0b 011101 = 0x1D</td></tr><tr><td>0b 011101 = 0x1D</td></tr><tr><td>0b 011101 = 0x1D</td></tr><tr><td>0b 011101 = 0x1D</td></tr><tr><td>0b 000011 = 0x03</td></tr><tr><td>0b 111111 = 0x3F</td></tr></table>	0b 000011 = 0x03	0b 011101 = 0x1D	0b 011101 = 0x1D	0b 011101 = 0x1D	0b 011101 = 0x1D	0b 011101 = 0x1D	0b 000011 = 0x03	0b 111111 = 0x3F
0	0	0	0	1	1																																																					
0	1	1	1	0	1																																																					
0	1	1	1	0	1																																																					
0	1	1	1	0	1																																																					
0	1	1	1	0	1																																																					
0	1	1	1	0	1																																																					
0	0	0	0	1	1																																																					
1	1	1	1	1	1																																																					
0b 000011 = 0x03																																																										
0b 011101 = 0x1D																																																										
0b 011101 = 0x1D																																																										
0b 011101 = 0x1D																																																										
0b 011101 = 0x1D																																																										
0b 011101 = 0x1D																																																										
0b 000011 = 0x03																																																										
0b 111111 = 0x3F																																																										
	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	<table><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr><tr><td>0b 111111 = 0x3F</td></tr></table>	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F	0b 111111 = 0x3F
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
1	1	1	1	1	1																																																					
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										
0b 111111 = 0x3F																																																										

Tableau III. 1 . Structure et codes des caractères

Chapitre III - Réalisation pratique

III.2.3.1. Interprétation

Chaque octet va être lu du bit de poids faible vers le bit de poids fort (de la droite vers la gauche). L'envoi des données soit **en série**, commençant par le premier bit (dans la broche 14), après, nous donnons un front montant à l'horloge (la broche 11), ensuite, nous envoyons le deuxième bit et nous donnons un front montant à l'horloge et ainsi de suite jusqu'à l'envoi des 24 bits.

Après l'envoi des données, nous activons la proche ST-CP (La broche numéro 12) puis nous la désactivons (donner une impulsion), afin de faire passer les données vers les sorties de registre à décalage.

Le premier registre prend les six bits du premier caractère et les deux autres du deuxième (selon le programme), l'ensemble vaut un octet. Le deuxième registre prend le reste des bits (les quatre derniers) du deuxième caractère et les quatre de troisième caractère. Le troisième registre prend les deux bits restant et le reste des bits sont réservés pour le caractère vide.

L'entrée data du troisième registre de décalage est reliée avec la sortie série **Q7'** du second ainsi, l'entrée data du second est reliée avec le **Q7'** du premier. A la fin, les données de l'entrée **série** sont converties en **sorties parallèles**.

Le tableau suivant montre en détails les états des sorties des registres, et la ligne activée à chaque étape.

Etapes	Numéro du registre	Sorties des registres à décalage								La ligne activée
1	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	1
		0	1	1	1	1	1	0	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	0	0	1	0	0	0	0	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	1	1	1	1	
2	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	2
		0	1	1	1	1	1	0	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	0	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	1	1	1	1	1	1	1	

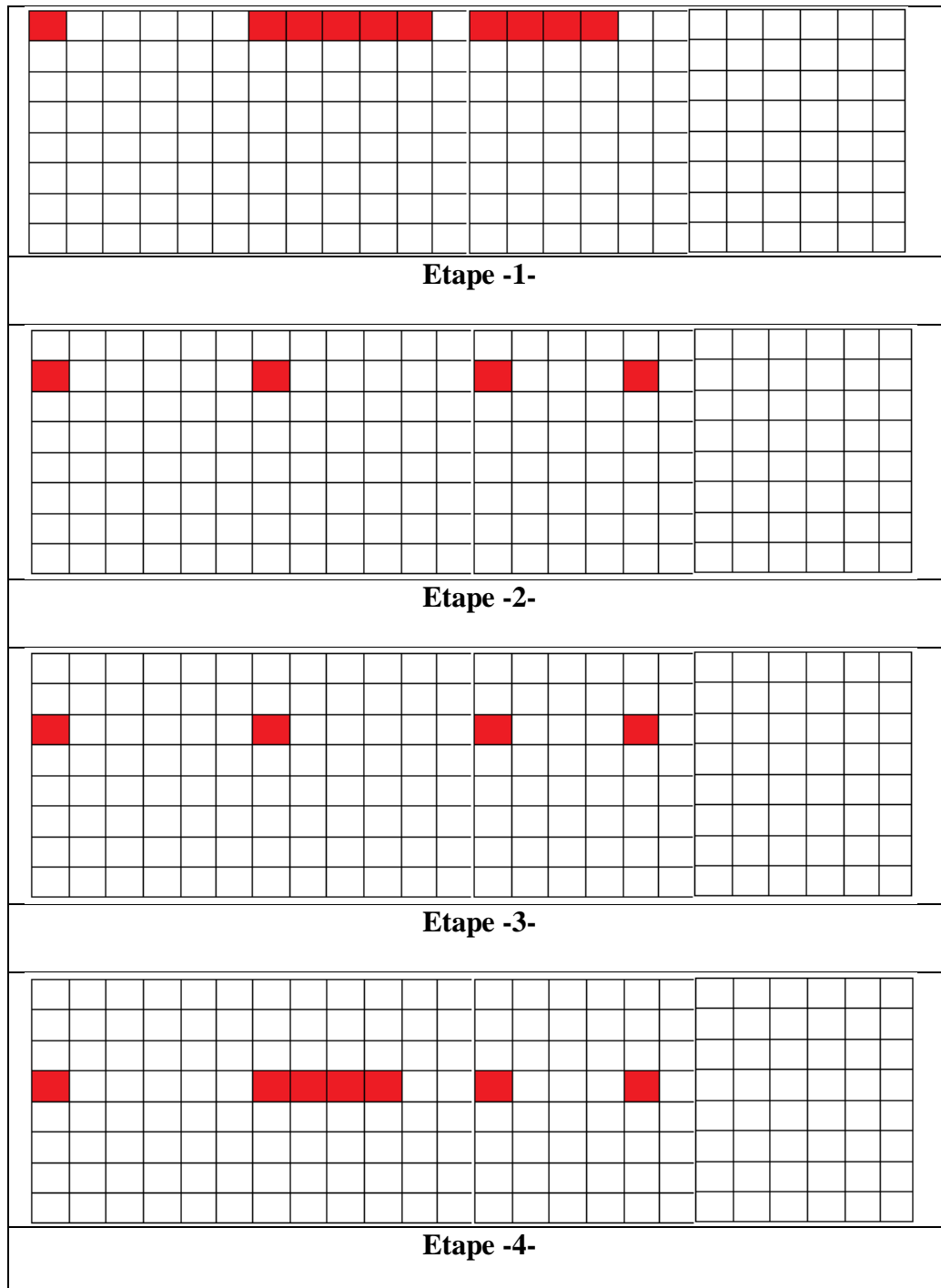
Chapitre III - Réalisation pratique

3	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	3
		0	1	1	1	1	1	0	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	0	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	1	1	1	1	1	1	1	
4	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	4
		0	1	1	1	1	1	0	0	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	0	1	1	0	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	1	1	1	1	1	1	1	
5	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	5
		0	1	1	1	1	1	0	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	0	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	1	1	1	1	1	1	1	
6	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	6
		0	1	1	1	1	1	0	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	0	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	1	1	1	1	1	1	1	
7	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	7
		0	0	0	0	0	1	0	0	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		0	0	0	1	0	0	0	0	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	1	1	1	1	
8	1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	8
		1	1	1	1	1	1	1	1	
	2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	1	1	1	1	
	3	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
		1	1	1	1	1	1	1	1	

Table III. 2. Sorties des registres de décalage

III.2.3. 2. Résultats obtenus

Les résultats obtenus après chaque envoi sont illustrés dans la figure III.11 :



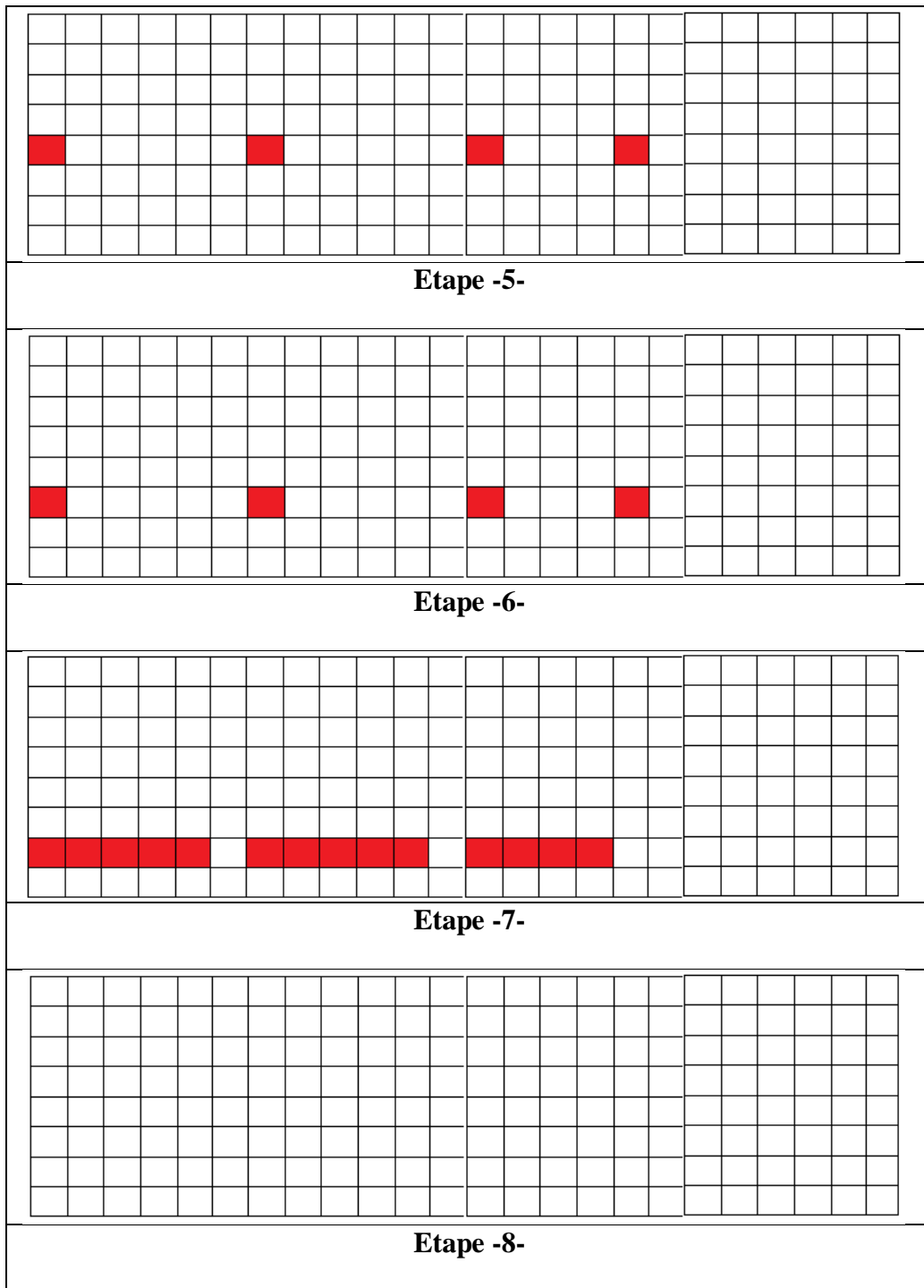


Figure III.11. Les différentes étapes effectuées pour afficher le mot 'LED'

Chapitre III - Réalisation pratique

A chaque instant donné, une seule des 8 lignes des matrices est alimentée. Le balayage rapide et cyclique des lignes nous permet de visualiser le mot 'LED' (figure III.12).

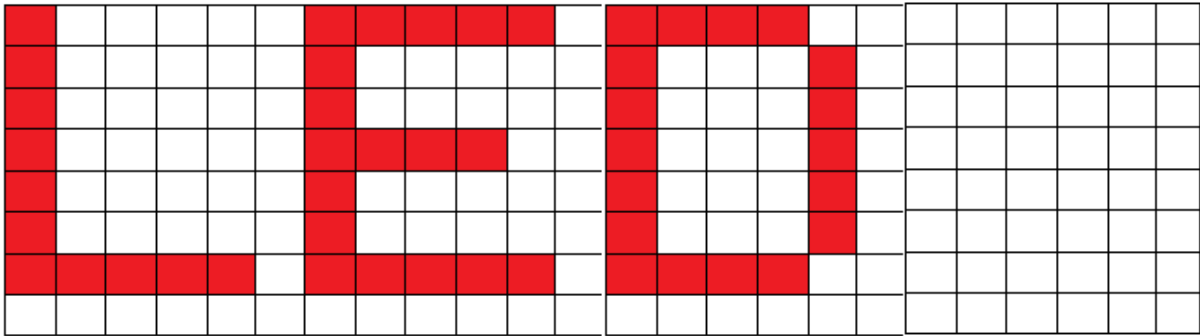


Figure III. 12. Affichage du mot 'LED'



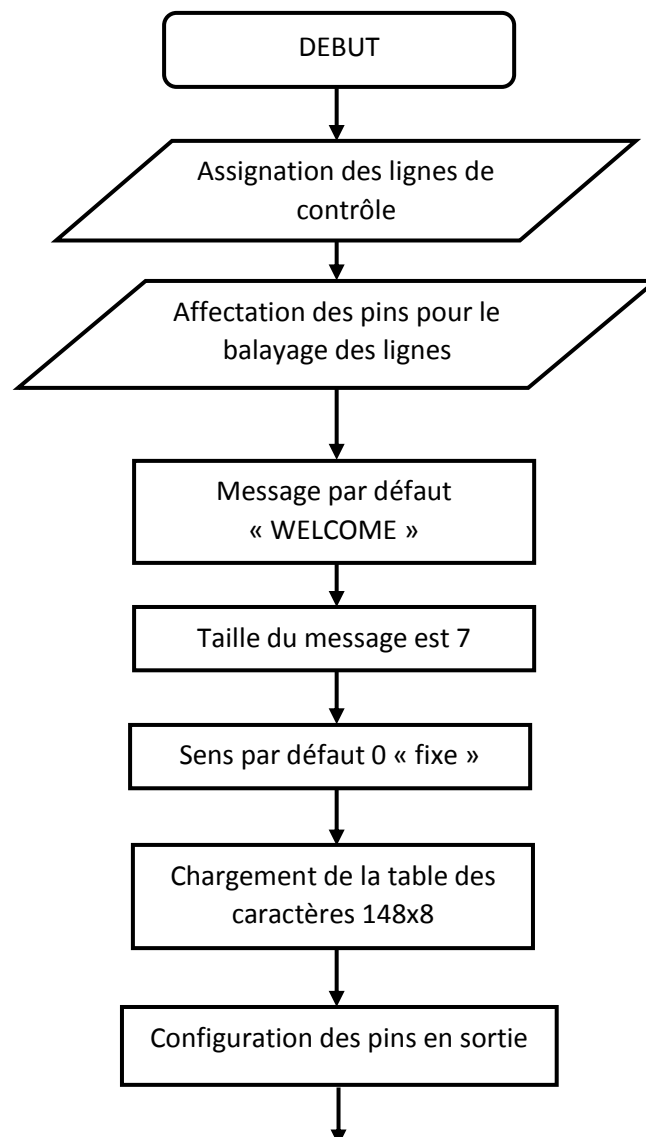
III.2.4. Organigrammes

Afin de simplifier le principe de fonctionnement, nous allons établir les organigrammes dressé en figure III.13 et figure III.14.

Le logiciel de gestion est composé de 3 tâches principales s'exécutant en série.

1. Envoi d'un octet au 74HC595.
2. Calcul des octets issues des sorties des douze registres à décalages.
3. L'affichage, contient trois modes :
 - . Fixe.
 - Défilement de la gauche vers la droite.
 - Défilement de la droite vers la gauche.

L'organigramme général est présenté comme suit :



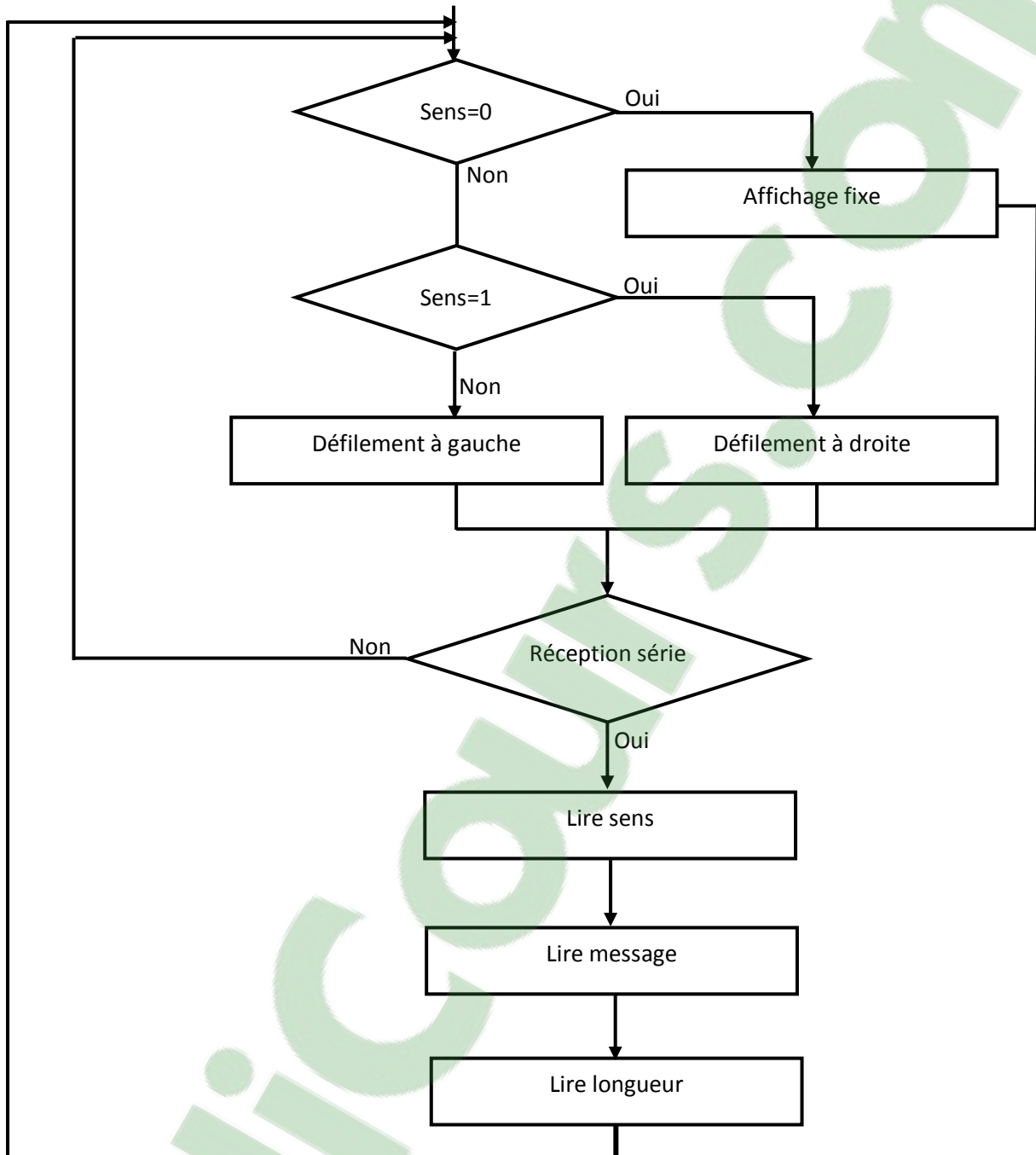


Figure III.13. Organigramme du programme principal

Chapitre III : Réalisation pratique

L'affichage représente une des parties les plus importantes de notre projet.
L'organigramme illustré dans la figure ci-dessous permet d'effectuer un affichage fixe.

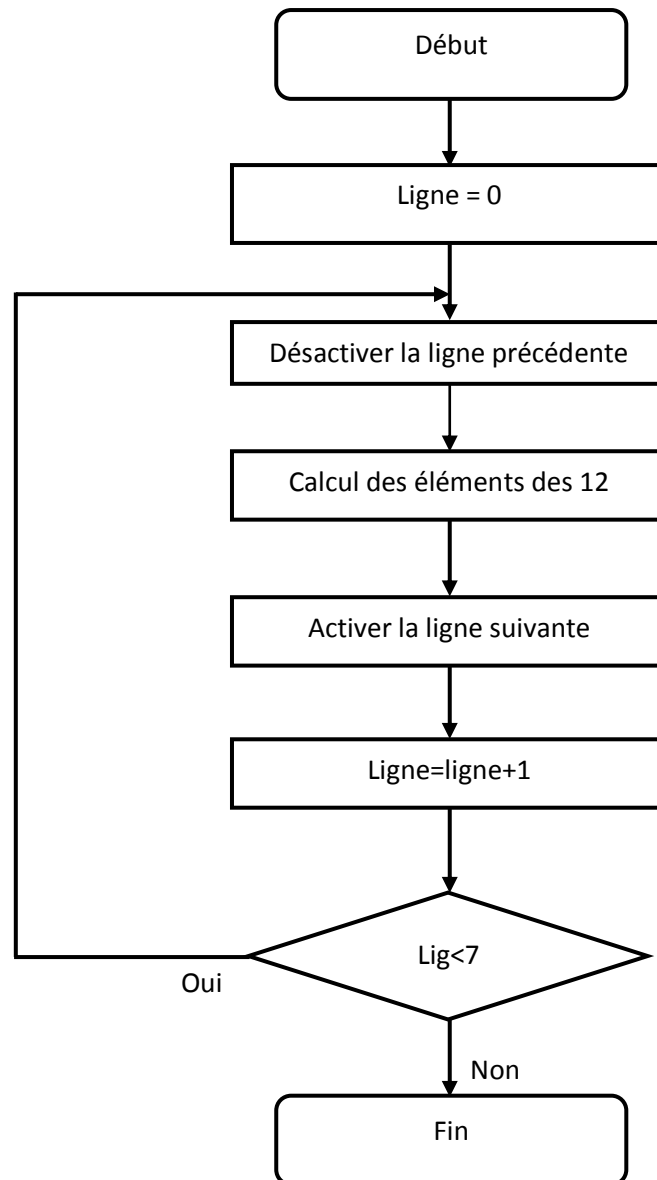


Figure III.14. Organigramme de procédure affichage fixe

Conclusion générale et perspectives

Dans notre projet, nous avons pu réaliser un prototype d'un journal lumineux qui contient sept cent soixante-huit LEDs, câblés en douze matrices, sur huit lignes et quatre-vingt-seize colonnes, commandé par la carte Arduino, commençant par la conception du circuit électrique, passant par la simulation sous ISIS qui nous a permis d'ajuster et modifier le circuit, puis la réalisation des différents étages, par la suite, nous avons élaboré des organigrammes pour aboutir au programme arduino.

Nous sommes arrivés dans le cadre du présent travail à acquérir des connaissances en programmation pour arduino ainsi que se familiariser en électronique.

Néanmoins, afin d'améliorer notre prototype nous proposons d'intégrer un module Bluetooth pour faciliter l'envoi des messages.

BIBLIOGRAPHIE

[1] Arduino Maitriser sa programmation et ses cartes d'interface

CHRISTAN TAVERNIER 2014, Paris édition DUNOD.

[2] BOITE A OUTILS ARDUINO

MARGOLIS Michael 2012, édition DUNOD.

[3] Arduino pour bien commencer en électronique et en programmation

ASTALASEVEN, ESKIMON et OLYTE, www.site.duzero.com

[4] Begening Arduino

MCROBERTS Michael 2013, édition Apress

[5] Atelier Arduino

JEAN-Noel 2016, Centre de Ressources ART Sensitif.

[6] LES CARTES A PUCE Théorie et mise en œuvre

TAVERNIER Christian 2011, 2^e édition DUNOD.

[7] S'initier à la programmation

DELANNOY Llaude 2008, édition EYROLLES.

[8] Arduino : Premier pas en informatique embarquée.

LANDRAULT Simon et Olyte 2014, Le blog d'Eskimon.

[9] Micro et nano-électronique Bases Composants Circuits

FANET Herve 1992, édition DUNOD.

[10] L'ELECTRONIQUE DE A à Z

NEFFATU Taher 2006, édition DUNOD.

[11] ARDUINO AVENTURES

KELLY Tames Floyd, www.it-ebodes.info.

[12] Notions Fondamentales pour l'Electricité et l'Electronique.

HAMDOUNE Abdelkader 2012.

[13] Electronique

MARET, MOREAU, ALLAY, DUBOIS, LAFARGUE et LE GOLF, édition BERTI.

[14] Rapport de stage

Implantation d'un module Bluetooth

TEBOUL Nicolas, société SLE.

[15] <http://www.instructables.com/id/Multiplexing-with-Arduino-and-the-74HC595/>

[16] http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno

[17] <http://fr.flossmanuals.net/arduino/programmer-arduino/>

[18] <http://blog.matael.org/writing/arduino-et-registres/>

[19] <http://eskimon.fr/269-arduino-annexe-1-ajouter-des-sorties-a-arduino-74hc595>

[20] http://atela.univ-lorraine.fr/docs/documents/Les_diodes.pdf

[21] <http://www.positron-libre.com/cours/electronique/diode/led/diode-led.php>

8 BIT SHIFT REGISTER WITH OUTPUT LATCHES (3 STATE)

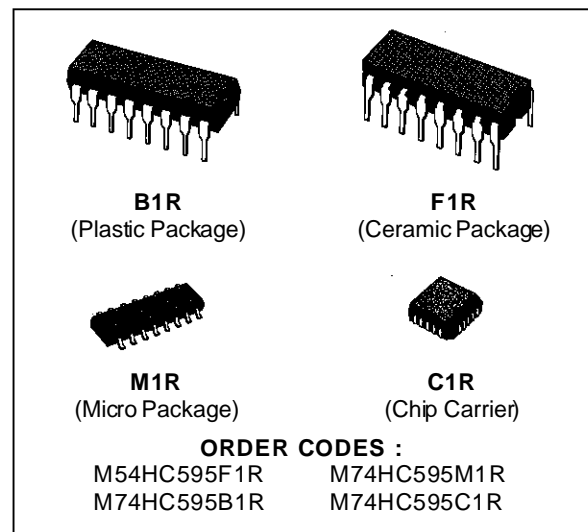
- HIGH SPEED
 $f_{MAX} = 55 \text{ MHz (TYP.) AT } V_{CC} = 5 \text{ V}$
- LOW POWER DISSIPATION
 $I_{CC} = 4 \mu\text{A (MAX.) AT } T_A = 25^\circ\text{C}$
- HIGH NOISE IMMUNITY
 $V_{NIH} = V_{NIL} = 28 \% V_{CC} \text{ (MIN.)}$
- OUTPUT DRIVE CAPABILITY
15 LSTTL LOADS FOR QA TO QH
10 LSTTL LOADS FOR QH'
- SYMMETRICAL OUTPUT IMPEDANCE
 $|I_{OH}| = |I_{OL}| = 6 \text{ mA (MIN.) FOR QA TO QH}$
 $|I_{OH}| = |I_{OL}| = 4 \text{ mA (MIN.) FOR QH'}$
- BALANCED PROPAGATION DELAYS
 $t_{PLH} = t_{PHL}$
- WIDE OPERATING VOLTAGE RANGE
 $V_{CC} \text{ (OPR)} = 2 \text{ V TO } 6 \text{ V}$
- PIN AND FUNCTION COMPATIBLE
WITH LSTTL 54/74LS595

DESCRIPTION

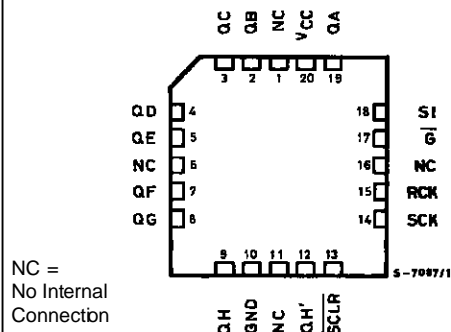
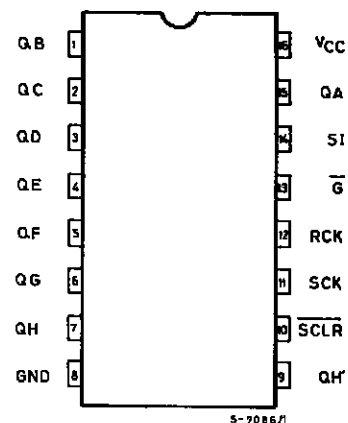
The M54/74HC595 is a high speed CMOS 8-BIT SHIFT REGISTERS/OUTPUT LATCHES (3-STATE) fabricated in silicon C²MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. This device contains an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The storage register has 8 3-STATE outputs. Separate clocks are provided for both the shift register and the storage register.

The shift register has a direct-overriding clear, serial input, and serial output (standard) pins for cascading. Both the shift register and storage register use positive-edge triggered clocks. If both clocks are connected together, the shift register state will always be one clock pulse ahead of the storage register.

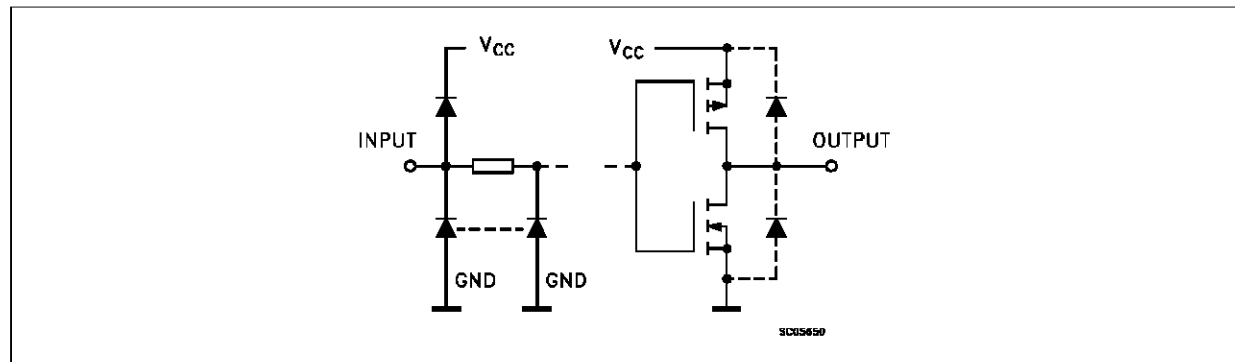
All inputs are equipped with protection circuits against static discharge and transient excess voltage.



PIN CONNECTIONS (top view)



INPUT AND OUTPUT EQUIVALENT CIRCUIT

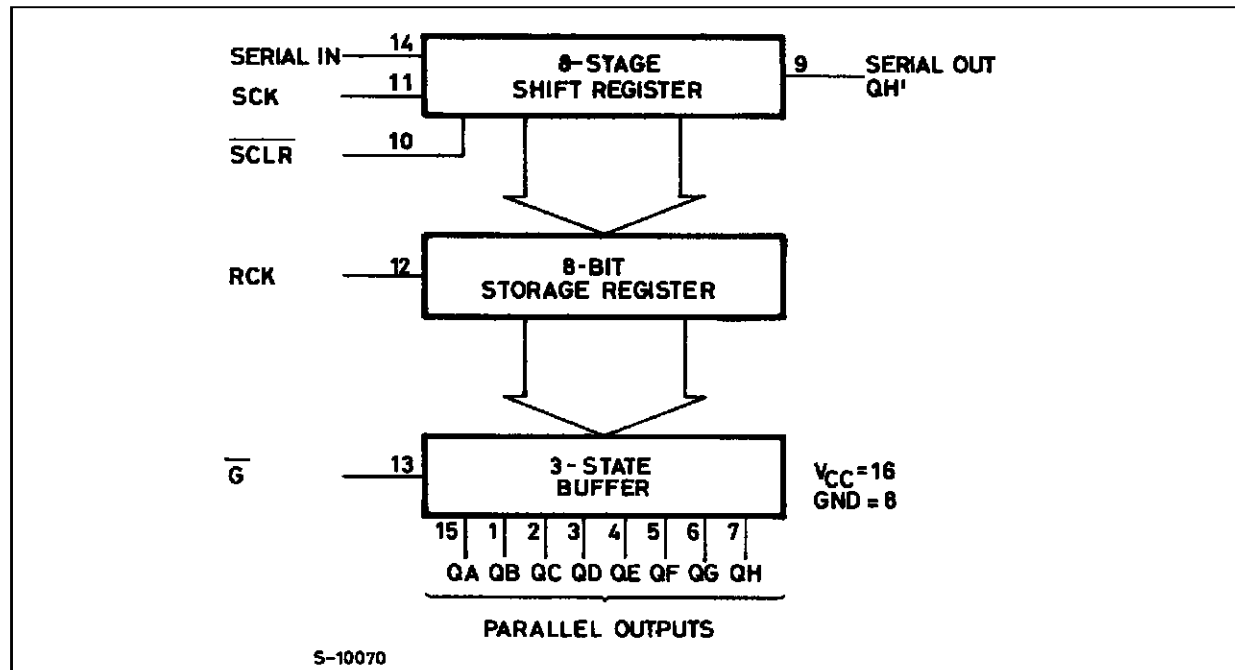


TRUTH TABLE

INPUTS					OUTPUT
SI	SCK	$\overline{\text{SCLR}}$	RCK	$\overline{\text{G}}$	
X	X	X	X	H	QA THRU QH OUTPUTS DISABLE
X	X	X	X	L	QA THRU QH OUTPUTS ENABLE
X	X	L	X	X	SHIFT REGISTER IS CLEARED
L		H	X	X	FIRST STAGE OF S.R. BECOMES "L" OTHER STAGES STORE THE DATA OF PREVIOUS STAGE, RESPECTIVELY
H		H	X	X	FIRST STAGE OF S.R. BECOMES "H" OTHER STAGES STORE THE DATA OF PREVIOUS STAGE, RESPECTIVELY
X		H	X	X	STATE OF S.R IS NOT CHANGED
X	X	X		X	S.R. DATA IS STORED INTO STORAGE REGISTER
X	X	X		X	STORAGE REGISTER STATE IS NOT CHANGED

X: DON'T CARE

LOGIC DIAGRAM

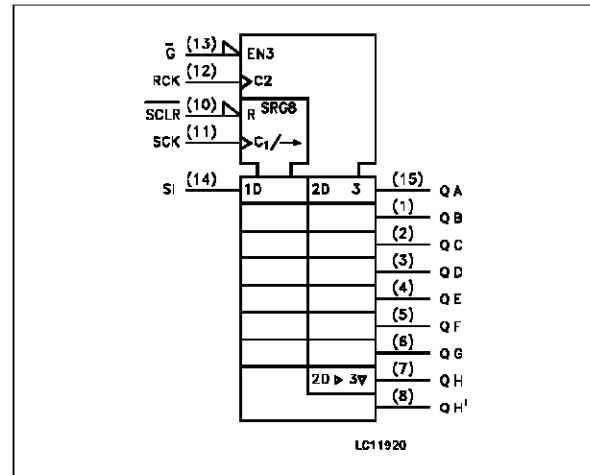


M54/M74HC595

PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 2, 3, 4, 5, 6, 7, 15	QA to QH	Data Outputs
9	QH'	Serial Data Outputs
10	SCLR	Shift Register Clear Input
11	SCK	Shift Register Clock Input
13	\overline{G}	Output Enable Input
14	SI	Serial Data Input
12	RCK	Storage Register Clock Input
8	GND	Ground (0V)
16	V _{CC}	Positive Supply Voltage

IEC LOGIC SYMBOL



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _{CC}	Supply Voltage	-0.5 to +7	V
V _I	DC Input Voltage	-0.5 to V _{CC} + 0.5	V
V _O	DC Output Voltage	-0.5 to V _{CC} + 0.5	V
I _{IK}	DC Input Diode Current	± 20	mA
I _{OK}	DC Output Diode Current	± 20	mA
I _O	DC Output Current Per Output Pin QA-QH	± 35	mA
I _O	DC Output Current Per Output Pin QH'	± 25	mA
I _{CC} or I _{GND}	DC V _{CC} or Ground Current	± 70	mA
P _D	Power Dissipation	500 (*)	mW
T _{stg}	Storage Temperature	-65 to +150	°C
T _L	Lead Temperature (10 sec)	300	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

(*) 500 mW: ± 65 °C derate to 300 mW by 10mW/°C: 65 °C to 85 °C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value	Unit
V _{CC}	Supply Voltage	2 to 6	V
V _I	Input Voltage	0 to V _{CC}	V
V _O	Output Voltage	0 to V _{CC}	V
T _{op}	Operating Temperature: M54HC Series M74HC Series	-55 to +125 -40 to +85	°C °C
t _r , t _f	Input Rise and Fall Time	V _{CC} = 2 V 0 to 1000 V _{CC} = 4.5 V 0 to 500 V _{CC} = 6 V 0 to 400	ns

DC SPECIFICATIONS

Symbol	Parameter	Test Conditions		Value						Unit		
		V _{CC} (V)		T _A = 25 °C 54HC and 74HC			-40 to 85 °C 74HC		-55 to 125 °C 54HC			
				Min.	Typ.	Max.	Min.	Max.	Min.		Max.	
V _{IH}	High Level Input Voltage	2.0		1.5			1.5		1.5		V	
		4.5		3.15			3.15		3.15			
		6.0		4.2			4.2		4.2			
V _{IL}	Low Level Input Voltage	2.0				0.5		0.5		0.5	V	
		4.5				1.35		1.35		1.35		
		6.0				1.8		1.8		1.8		
V _{OH}	High Level Output Voltage (for QH' output)	2.0	V _I = V _{IH} or V _{IL}	I _O =-20 μA	1.9	2.0		1.9		1.9		V
		4.5			4.4	4.5		4.4		4.4		
		6.0			5.9	6.0		5.9		5.9		
		4.5		I _O =-4.0 mA	4.18	4.31		4.13		4.10		
		6.0			I _O =-5.2 mA	5.68	5.8		5.63		5.60	
V _{OH}	High Level Output Voltage (for QA to QH outputs)	2.0	V _I = V _{IH} or V _{IL}	I _O =-20 μA		1.9	2.0		1.9		1.9	
		4.5			4.4	4.5		4.4		4.4		
		6.0			5.9	6.0		5.9		5.9		
		4.5		I _O =-6.0 mA	4.18	4.31		4.13		4.10		
		6.0			I _O =-7.8 mA	5.68	5.8		5.63		5.60	
V _{OL}	Low Level Output Voltage (for QH' output)	2.0	V _I = V _{IH} or V _{IL}	I _O = 20 μA			0.0	0.1		0.1		0.1
		4.5				0.0	0.1		0.1		0.1	
		6.0				0.0	0.1		0.1		0.1	
		4.5		I _O = 4.0 mA		0.17	0.26		0.33		0.40	
		6.0			I _O = 5.2 mA		0.18	0.26		0.33		0.40
V _{OL}	Low Level Output Voltage (for QA to QH outputs)	2.0	V _I = V _{IH} or V _{IL}	I _O = 20 μA			0.0	0.1		0.1		0.1
		4.5				0.0	0.1		0.1		0.1	
		6.0				0.0	0.1		0.1		0.1	
		4.5		I _O = 6.0 mA		0.17	0.26		0.33		0.40	
		6.0			I _O = 7.8 mA		0.18	0.26		0.33		0.40
I _I	Input Leakage Current	6.0	V _I = V _{CC} or GND				±0.1		±1		±1	μA
I _{OZ}	3 State Output Off State Current	6.0	V _I = V _{IH} or V _{IL} V _O = V _{CC} or GND			±0.5		±5		±10	μA	
I _{CC}	Quiescent Supply Current	6.0	V _I = V _{CC} or GND			4		40		80	μA	

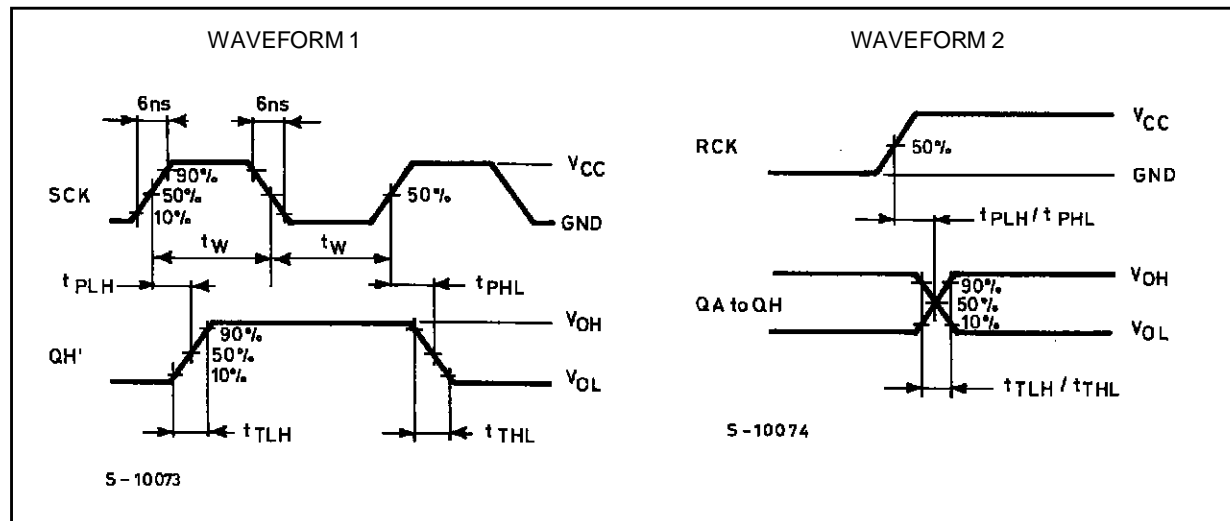
AC ELECTRICAL CHARACTERISTICS ($C_L = 50 \text{ pF}$, Input $t_r = t_f = 6 \text{ ns}$)

Symbol	Parameter	Test Conditions			Value						Unit	
		V _{CC} (V)	C _L (pF)		T _A = 25 °C 54HC and 74HC			-40 to 85 °C 74HC		-55 to 125 °C 54HC		
					Min.	Typ.	Max.	Min.	Max.	Min.		Max.
t _{TLH} t _{THL}	Output Transition Time (Qn)	2.0	50			25	60		75		90	ns
		4.5				7	12		15		18	
		6.0				6	10		13		15	
t _{TLH} t _{THL}	Output Transition Time (QH')	2.0	50			30	75		95		115	ns
		4.5				8	15		19		23	
		6.0				7	13		16		20	
t _{PLH} t _{PHL}	Propagation Delay Time (SCK - QH')	2.0	50			45	125		155		190	ns
		4.5				15	25		31		38	
		6.0				13	21		26		32	
t _{PLH} t _{PHL}	Propagation Delay Time (SCLR - QH')	2.0	50			60	175		220		265	ns
		4.5				18	35		44		53	
		6.0				15	30		37		45	
t _{PLH} t _{PHL}	Propagation Delay Time (RCK - Qn)	2.0	50			60	150		190		225	ns
		4.5				20	30		38		45	
		6.0				17	26		32		38	
		2.0	150			75	190		240		285	ns
		4.5				25	38		48		57	
		6.0				22	32		41		48	
t _{PZL} t _{PZH}	3 State Output Enable Time	2.0	50	R _L = 1 KΩ		45	135		170		205	ns
		4.5				15	27		34		41	
		6.0				13	23		29		35	
		2.0	150	R _L = 1 KΩ		60	175		220		265	ns
		4.5				20	35		44		53	
		6.0				17	30		37		45	
t _{PLZ} t _{PHZ}	3 State Output Disable Time	2.0	50	R _L = 1 KΩ		30	150		190		225	ns
		4.5				15	30		38		45	
		6.0				14	26		32		38	
f _{MAX}	Maximum Clock Frequency	2.0	50		6.0	17		4.8		4		ns
		4.5			30	50		24		20		
		6.0			35	59		28		24		
		2.0	150		5.2	14		4.2		3.4		ns
		4.5			26	40		21		17		
		6.0			31	45		25		20		
t _{W(H)}	Minimum Pulse Width (SCK, RCK)	2.0	50			17	75		95		110	ns
		4.5				6	15		19		22	
		6.0				6	13		16		19	
t _{W(L)}	Minimum Pulse Width (SCLR)	2.0	50			20	75		95		110	ns
		4.5				6	15		19		22	
		6.0				6	13		16		19	
t _s	Minimum Set-up Time (SI - CCK)	2.0	50			25	50		65		75	ns
		4.5				5	10		13		15	
		6.0				4	9		11		13	

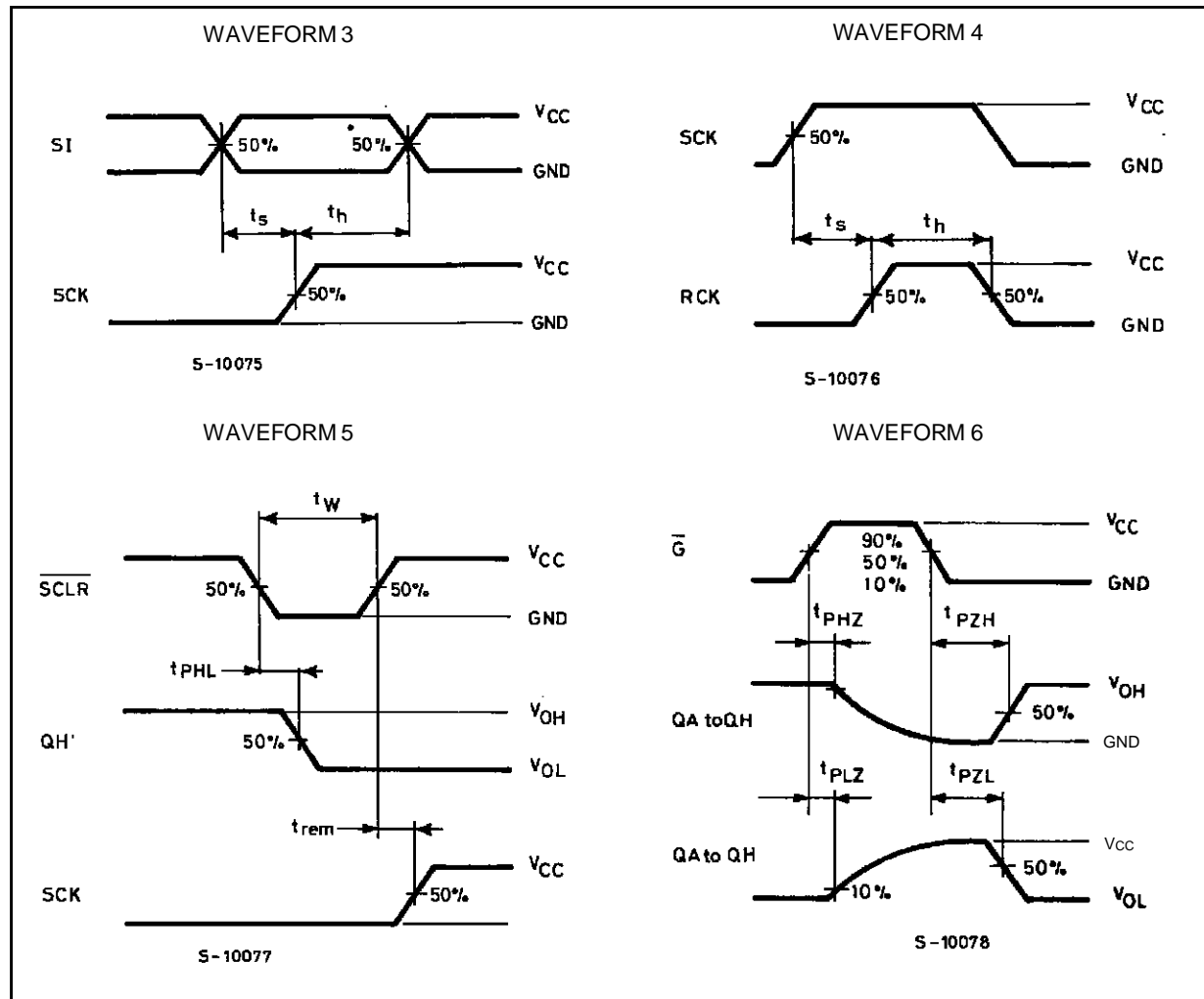
AC ELECTRICAL CHARACTERISTICS ($C_L = 50 \text{ pF}$, Input $t_r = t_f = 6 \text{ ns}$)

Symbol	Parameter	Test Conditions			Value						Unit	
		V _{CC} (V)	C _L (pF)		T _A = 25 °C 54HC and 74HC			-40 to 85 °C 74HC		-55 to 125 °C 54HC		
					Min.	Typ.	Max.	Min.	Max.	Min.		Max.
t _s	Minimum Set-up Time (SCK - RCK)	2.0	50		35	75		95		110	ns	
		4.5			8	15		19		22		
		6.0			6	13		16		19		
t _s	Minimum Set-up Time (SCRL - RCK)	2.0	50		40	100		125		145	ns	
		4.5			10	20		25		29		
		6.0			7	17		21		25		
t _h	Minimum Hold Time	2.0	50			0		0		0	ns	
		4.5				0		0		0		
		6.0				0		0		0		
t _{REM}	Minimum Clear Removal Time	2.0	50		15	50		65		75	ns	
		4.5			3	10		13		15		
		6.0			3	9		11		13		
C _{IN}	Input Capacitance				5	10		10		10	pF	
C _{PD} (*)	Power Dissipation Capacitance				184						pF	

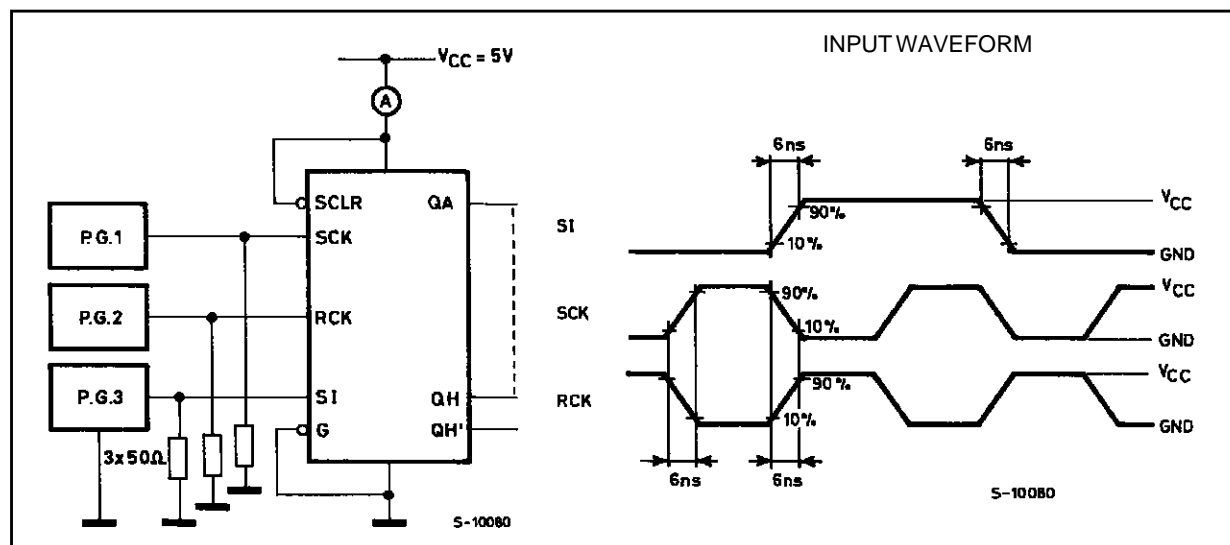
(*) C_{PD} is defined as the value of the IC's internal equivalent capacitance which is calculated from the operating current consumption without load. (Refer to Test Circuit). Average operating current can be obtained by the following equation. $I_{CC(opr)} = C_{PD} \cdot V_{CC} \cdot f_{IN} + I_{CC}$

SWITCHING CHARACTERISTICS TEST WAVEFORM

SWITCHING CHARACTERISTICS TEST WAVEFORM (continued)

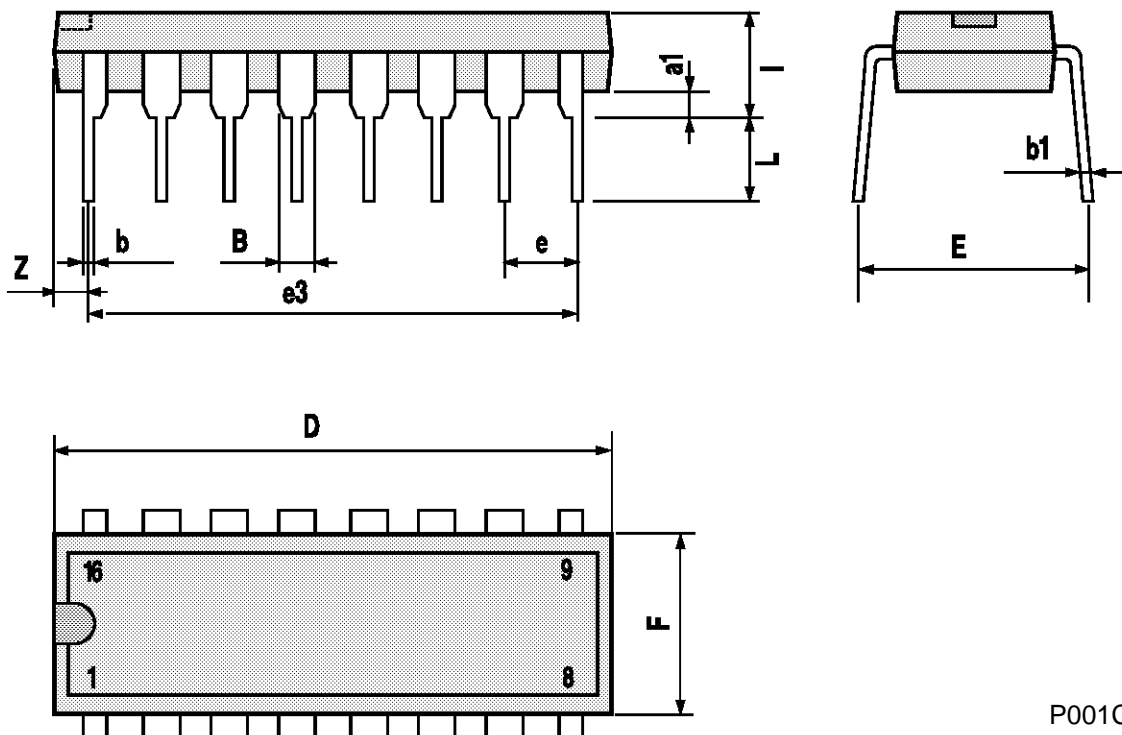


TEST CIRCUIT I_{CC} (Opr.)



Plastic DIP16 (0.25) MECHANICAL DATA

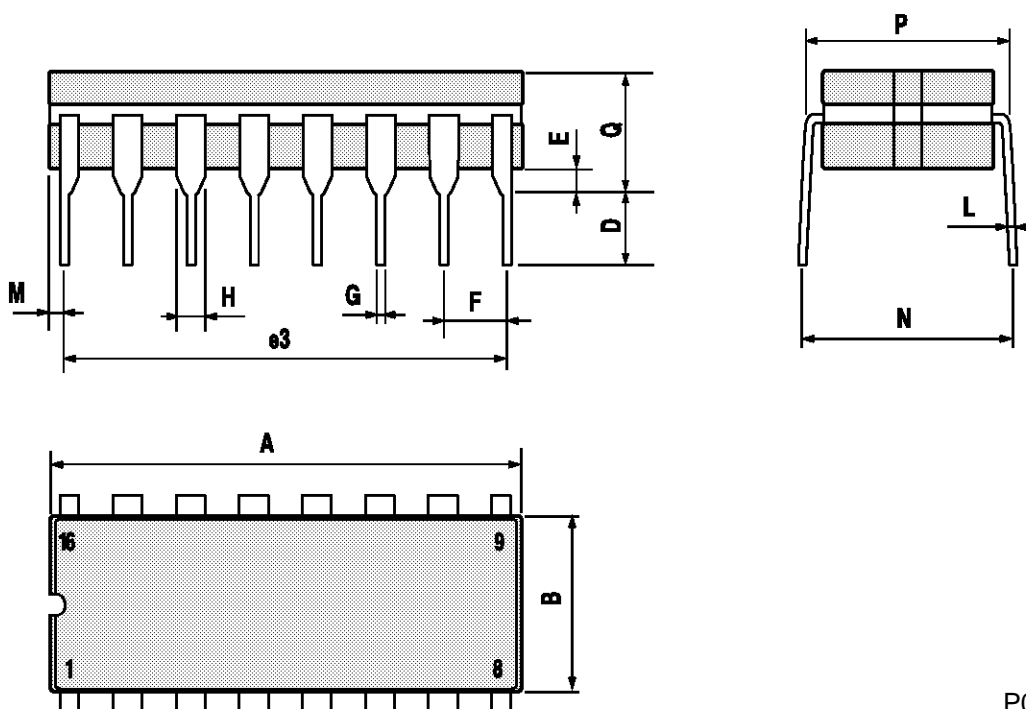
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050



P001C

Ceramic DIP16/1 MECHANICAL DATA

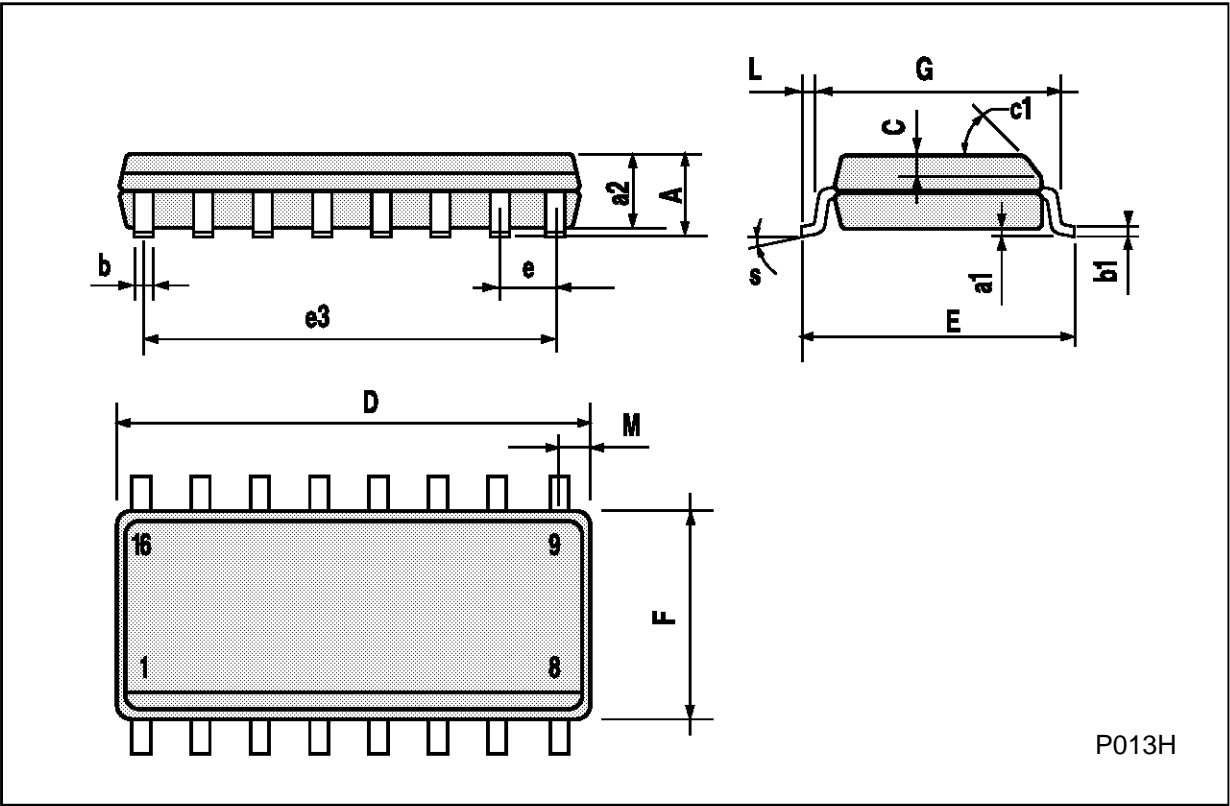
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			20			0.787
B			7			0.276
D		3.3			0.130	
E	0.38			0.015		
e3		17.78			0.700	
F	2.29		2.79	0.090		0.110
G	0.4		0.55	0.016		0.022
H	1.17		1.52	0.046		0.060
L	0.22		0.31	0.009		0.012
M	0.51		1.27	0.020		0.050
N			10.3			0.406
P	7.8		8.05	0.307		0.317
Q			5.08			0.200



P053D

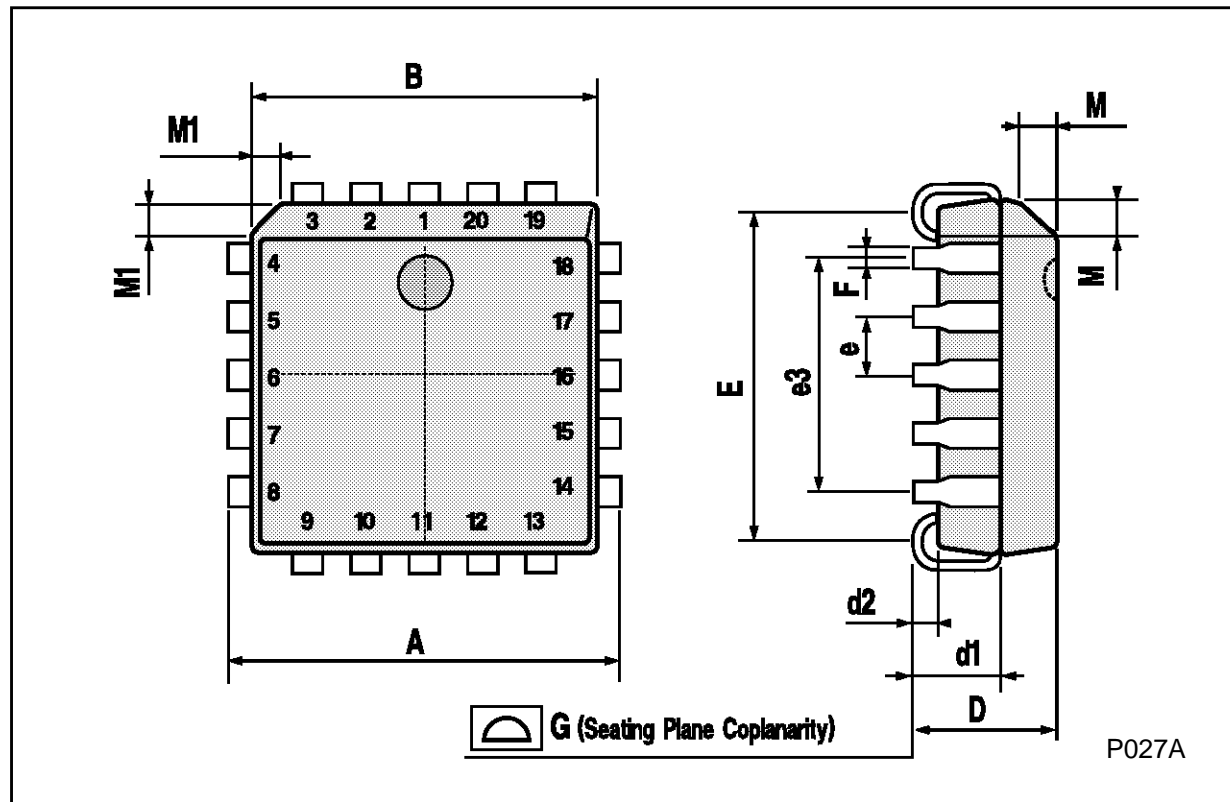
SO16 (Narrow) MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.75			0.068
a1	0.1		0.2	0.004		0.007
a2			1.65			0.064
b	0.35		0.46	0.013		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.019	
c1	45° (typ.)					
D	9.8		10	0.385		0.393
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F	3.8		4.0	0.149		0.157
G	4.6		5.3	0.181		0.208
L	0.5		1.27	0.019		0.050
M			0.62			0.024
S	8° (max.)					



PLCC20 MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	9.78		10.03	0.385		0.395
B	8.89		9.04	0.350		0.356
D	4.2		4.57	0.165		0.180
d1		2.54			0.100	
d2		0.56			0.022	
E	7.37		8.38	0.290		0.330
e		1.27			0.050	
e3		5.08			0.200	
F		0.38			0.015	
G			0.101			0.004
M		1.27			0.050	
M1		1.14			0.045	



Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1994 SGS-THOMSON Microelectronics - All Rights Reserved

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands -
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A

This datasheet has been downloaded from:

www.DatasheetCatalog.com

Datasheets for electronic components.

Résumé

Ce projet de fin d'étude porte sur la conception et la réalisation d'un journal lumineux à base de matrices LEDs, en utilisant des registres à décalage sous la référence 74HC595 avec un mode d'affichage multiplexé.

Le contrôle de l'affichage est assuré par la carte Arduino UNO et l'ensemble des programmes sont développés sur le logiciel Arduino.

Mot clés : Journal lumineux, multiplexage, registre à décalage, Arduino.

Abstract

The subject of this final project is the conception and the realization of LED board basically made of a LEDs matrix, using the shift registers 74HC595.

In order to limit the consumption and the number of wires, the multiplexing is fundamental.

The displaying control is assured by the programmable card « Arduino UNO » and the programmes are developed using Arduino software.

Keywords: LED board, multiplexing, shift register, Arduino.

ملخص

من اجل تسهيل عملية بث المعلومات و الإعلانات في مختلف الأماكن وفي إطار تحضيرنا لمشروع نهاية الدراسة، قمنا بتصميم و إنجاز لوح صمامات **LED**.

التحكم في نوعية الرسائل (عربي، فرنسي...) يكون من طرف لوحة الأردوينو بواسطة سجلات الإزاحة **74HC595**.

كلمات مفتاحية: أردوينو، لوحة صمامات، سجلات الإزاحة.