



وزارة البحث العلمي والتعليم العالي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis Mostaganem
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie
DEPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE

Présenté pour obtenir le diplôme de

MASTER EN GENIE ELECTRIQUE

Option : Electronique des Systèmes Embarqués

Par :

DIANE SEKOU

Intitulé:

Mesure de Longueur par Arduino et Souris

| | | | |
|---------------|-------------|-----|--------------------------|
| Président : | B.BOUADJEMI | MCA | Université de Mostaganem |
| Examineur 1 : | A.ABBAD | MCA | Université de Mostaganem |
| Encadreur : | W.BENSTAALI | MCA | Université de Mostaganem |

Année Universitaire 2019/2020

REMERCIEMENTS

Je tiens à exprimer mes sentiments les plus nobles et sincères et ma profonde gratitude et reconnaissance à l'égard de tous ceux qui ont œuvré d'une façon ou d'une autre pour la réalisation de ce travail.

Je ne peux que témoigner et remercier mon encadreur Mr W.Benstaali pour son soutien, sa disponibilité, sa patience, ses qualités humaines, ses suggestions et conseils.

Mes remerciements vont aux membres du jury pour l'effort qu'ils fournissent en examinant mon modeste travail.

Sans oublier tous mes proches qui m'ont accompagné et encouragé durant ce travail.

Dédicace

Je remercie le bon Dieu de m'avoir permis de réaliser ce travail.

Je dédie ce mémoire de fin d'étude à :

Mon père Balla Diané.

Ma mère Fatoumata Sidibé

Mes frères et sœurs

Toute ma famille

Mes amis et camarades d'études

Ma modeste personne

Mon Encadreur

Résumé :

Les erreurs de mesures de longueur peuvent influencer sur certains processus de dimensionnement, la conception des micros objets qui demande une certaine précision et les performances techniques.

Pour répondre aux besoins des marchands de tissus, de l'industrie, et pallier aux problèmes qu'on peut retrouver chez les appareils classiques de mesures de longueurs d'un objet, ce projet propose un prototype électronique.

Pour réaliser ce travail, le choix s'est porté sur les différents composants électroniques selon leurs spécificités, caractéristiques propres : un bouton poussoir pour démarrer et arrêter le processus, une souris à bille servant de capteur de mouvement, une carte Arduino pour l'unité centrale et un afficheur LCD pour afficher le résultat.

Pour voir le résultat de notre travail nous avons fait la simulation du circuit à l'aide du logiciel Isis proteus.

Abstract :

Errors in length measurements can influence some sizing processes, the design of micro objects which requires a certain precision and technical performance. To meet the needs of fabric merchants and industry, and to overcome the problems that can be found in conventional devices for measuring the length of an object, this project offers an electronic prototype. To carry out this work, the choice fell on electronic component disputes according to their specificities, specific characteristics: a push button to start and stop the process, a bile mouse serving as a motion sensor, an Arduino card for the unit control unit and a lcd display to display the result. To see the result of our work we decided to do the circuit simulation using Isis proteus.

ملخص

يمكن أن تؤثر أخطاء قياس الطول على عمليات تحجيم معينة ، وتصميم الكائنات الدقيقة التي تتطلب دقة معينة وأداء تقني. لتلبية احتياجات تجار النسيج والصناعة، وللتغلب على المشاكل التي يمكن العثور عليها في الأجهزة التقليدية لقياس طول الشيء، يقدم هذا المشروع نموذجًا إلكترونيًا أوليًا.

لتنفيذ هذا لعمل، وقع الاختيار على نواعات المكونات الإلكترونية وفقًا لخصائصها وخصائصها المحددة: زر ضغط لبدء العملية وإيقافها، وفأرة كروية تعمل كمستشعر للحركة، وبطاقة arduino للوحدة وحدة تحكم وشاشة LCD لعرض النتيجة. لمشاهدة نتيجة عملنا قمنا بمحاكاة الدائرة باستخدام برنامج ISIS PROTEUS

Table des matières

| | |
|--|-----------|
| Chapitre I : Souris ps2 | 1 |
| I. Présentation : | 1 |
| 1. Fonctionnement | 1 |
| 2. Protocole de communication | 2 |
| II. Circuit interne d'une souris ps2 (Circuit d'émission et de détection) | 3 |
| 1. Circuit d'émission (Emetteurs LED rouge et LED infrarouge) | 3 |
| 1.1. Principe de fonctionnement de la LED | 4 |
| 1.1.1. Longueur d'onde du pic d'émission | 5 |
| 1.1.2. Spectre ou largeur spectrale à mi- intensité | 5 |
| 1.1.3. Correspondance couleurs, longueurs d'onde et énergie des photons | 6 |
| 1.1.4 Diagramme de rayonnement | 6 |
| 1.1.5. Angle d'émission à mi- intensité | 7 |
| 1.1.6. Intensité lumineuse | 7 |
| 1.2. Caractéristiques électriques | 7 |
| 1.2.1. Point de fonctionnement et tension directe | 7 |
| 1.2.2. Point de fonctionnement et Tension inverse VR | 9 |
| 2. Circuit de détection | 9 |
| 2.1. La photodiode | 10 |
| 2.1.1. Principe de fonctionnement | 10 |
| 2.1.2. Association émetteur – récepteur I.R | 11 |
| 2.2. Le phototransistor | 12 |
| 2.2.1. Principe de fonctionnement | 12 |
| Chapitre II : Carte Arduino | 14 |
| 1. Présentation | 14 |
| 2. Microcontrôleur ATMEMega328 | 20 |
| 2.1 Caractéristiques de L'Atmega 328 | 20 |
| 2.2. Structure interne de l'ATMega328 | 22 |

| | |
|--|----|
| 2.2.1. Status Register(<i>SREG</i>) | 22 |
| 2.2.2. Digital I/O Entrées Sorties Binaires/Tout Ou Rien(<i>TOR</i>) | 23 |
| 4. Timers/Counters de <i>ATMega328</i> | 24 |
| Chapitre III : L'afficheur LCD | |
| 1. Généralités | 27 |
| 2. Les Caractéristiques | 31 |
| 3. Le brochage | 33 |
| 4. La transmission de données | 34 |
| Chapitre IV : Simulation | |
| 1. Schéma synoptique..... | 38 |
| 2. Le logiciel de simulation <i>PROTEUS</i> | 38 |
| 3. Schéma électrique | 40 |
| 4. Programmation..... | 43 |
| | |
| Conclusion générale..... | 50 |
| Annexes | 51 |
| Indexe des figures..... | 52 |

Objectif

Objectif

Ce travail consiste à simuler un instrument de mesure de la distance (décamètre électronique) avec une souris ps2 et une carte arduino qui fonctionne selon un cahier de charge précis et répond à un besoin et une demande qui sont :

-Afficher la valeur de la longueur de l'objet mesuré avec une unité d'incrément de 1cm.

- Réduire les erreurs de mesure.

- Le démarrage et l'arrêt de la mesure à l'aide d'un bouton poussoir.

- Utilisation pour la mesure des tissus et applications industrielles.

Introduction

Introduction

Les instruments usuels de mesure de longueur sont par définition des instruments d'usage courant, ne nécessitant ni une formation trop longue ni un haut niveau de connaissance, et donc utilisant des principes de mesure simples. Les instruments dits usuels d'aujourd'hui sont généralement des instruments qui étaient déjà utilisés dans le courant du siècle dernier, voire avant pour le pied à coulisse ou le micromètre, dont les premiers instruments datent de la fin du XVIII^e siècle. Le principe du **vernier** consiste à placer une réglette gravée, proche de la règle, le plus souvent sur une partie biseautée au-dessus de celle-ci. Le vernier le plus courant est le vernier au 1/50 mm, la règle est gravée tous les millimètres et le vernier est constitué de 50 traits sur 49 mm ; pour le vernier au 1/20 mm, il comporte 20 traits sur 19 mm, et enfin pour le vernier au 1/10 mm (qui n'est pratiquement plus utilisé aujourd'hui), il comporte 10 traits sur 9 mm.

Le **micromètre**, souvent encore appelé **palmer** du nom de son inventeur J.L. Palmer en 1848, est constitué d'une vis de précision, la lecture étant effectuée à l'aide d'un tambour gradué pour les instruments au 1/100 mm et parfois à l'aide d'un vernier pour les instruments au 1/1 000 mm.

Les **comparateurs** ont permis la mise au point de toute une **métrologie**. Il est vrai que toute mesure est une comparaison, mais les habitudes sont telles que l'appellation comparaison est plus réservée aux méthodes de mesure dans lesquelles la pièce à mesurer est « comparée » à un étalon de même nature, c'est-à-dire même forme de surface et dimension proche de la dimension de la pièce. Cette dernière affirmation reste valable, dans son principe, pour les comparateurs à faible course,

pour lesquels le principe de la mesure (dite **méthode de Borda**) consiste à régler le comparateur à « zéro » sur l'étalon et à venir mesurer la pièce pour déterminer l'écart par rapport à l'étalon. Pour les comparateurs ayant une course de mesure de plusieurs millimètres, par exemple le comparateur le plus courant ayant une course de 10 mm, celui-ci ne mesure donc pas un petit écart et constitue ainsi un mesureur de déplacement ; c'est pour cette raison que, pour ces instruments, l'appellation comparateur est parfois controversée.

Le développement de l'électronique a permis l'essor de nouveaux instruments, les capteurs électroniques de déplacements ou les interféromètres à comptage de franges. Mais les instruments usuels ont aussi évolué avec ces technologies, en particulier avec la génération des instruments dits à affichage numérique. Parmi les instruments usuels, tous ont aujourd'hui un type à **affichage numérique**.

Ces instruments usuels à affichage numérique facilitent la lecture des mesures, mais leur **principal intérêt** est la possibilité de transmission des mesures vers un ordinateur pour traitements des résultats. Pour les comparateurs à affichage numérique, outre la possibilité de transmission des données, l'intérêt d'une faible résolution avec une grande course permet une utilisation plus universelle et évite la rigueur du principe de Borda, pièce et étalon de dimension proche. En effet, ces instruments, sur des multicotes ou outillages de contrôle spécifiques, évitent l'utilisation d'un étalon de forme complexe ; souvent une cale, un tampon ou une bague seront utilisés au moindre coût et avec des incertitudes de mesure plus faibles, compte tenu de la faible résolution du comparateur et d'une meilleure connaissance de l'étalon.

Chapitre I : Souris PS2

I. Présentation :

1. Fonctionnement :

La souris est un organe de pointage fournissant les coordonnées X et Y d'un déplacement ainsi que l'état de deux boutons (Droite et Gauche) ; la transmission de ces informations se fait par une liaison série synchrone à 9600 bauds (environ) au format 8 (8 bits de données, 1 bit de parité (impair (*Odd*)), 1 bit de stop).

La souris fonctionne par défaut avec une résolution de 200 CPI (*Counts Per Inch*), ce qui correspond à une résolution de 8 coups/mm (le pas est de 125 μm). Les déplacements sont codés sur 8 bits, soit une plage de (0 à 255), soit une distance d'environ 3,2 cm.

La souris semble pouvoir fournir jusqu'à 40 déplacements à la seconde, ce qui permet de transmettre des déplacements allant jusqu'à 2 m/s.

La souris est alimentée en 5V par l'intermédiaire du cordon ; le connecteur PS/2 comporte 6 broches.

La souris mécanique comporte une bille sur laquelle tournent deux rouleaux. Ces rouleaux comportent chacun un disque cranté qui tourne entre une photodiode et une LED (Diode électroluminescente) laissant passer la lumière par séquence. Lorsque la lumière passe, la photodiode renvoie un bit (1), lorsqu'elle rencontre un obstacle, la photodiode renvoie un bit nul (0). A l'aide de ces informations, l'ordinateur peut connaître la position du curseur, voire sa vitesse.

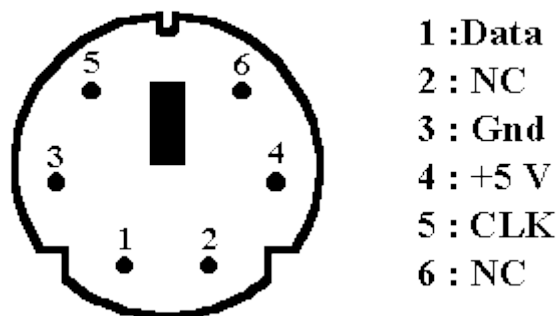


Figure 1 : Connecteur SOURIS PS/2 (mâle, vue en bout de cordon)

Chapitre I : Souris PS2

2. Protocole de communication :

La communication est de type série synchrone, il y a donc 2 lignes, l'une pour l'horloge (CLK) et l'autre pour les données (Data) ; l'interface électrique est de type " Collecteur Ouvert ", c'est à dire que les 2 éléments connectés peuvent prendre la main sur les lignes sans conflit électrique.

La transmission se fait par mots de 8 bits , précédés d'un bit de START (à 0) et suivis d'un bit de parité (impair) et d'un bit de STOP (à 1) ; le message est alors composé de 11 bits ; l'horloge de transmission a une période de 100 μ s.

En mode normal, c'est la souris qui émet ces 2 signaux.

En mode de configuration, la souris émet l'horloge et l'hôte place les données, la souris acquitte la transmission en forçant la ligne Data à 0 pendant le stop.

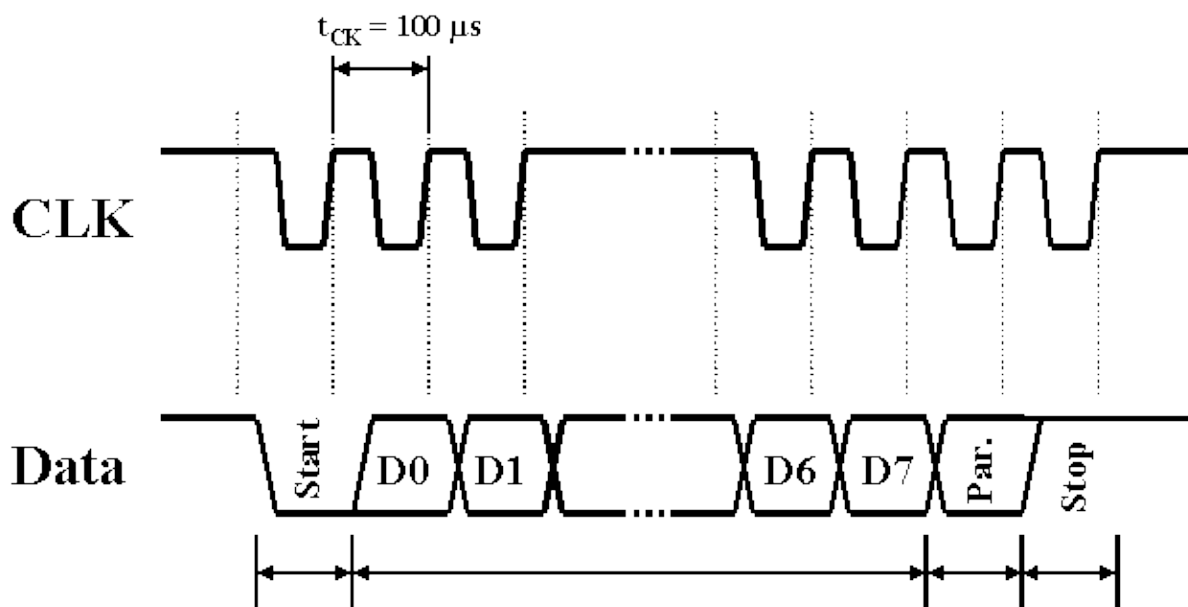


Figure 2. : Protocole de communication

Chapitre I : Souris PS2

La souris fournit ses informations sous forme de triplets espacés d'environ 350 μs ; les trames ainsi constituées durent alors 3,6 ms et sont espacées d'au moins 6,4 ms. Ainsi, la souris pourra envoyer jusqu'à 100 trames par seconde.

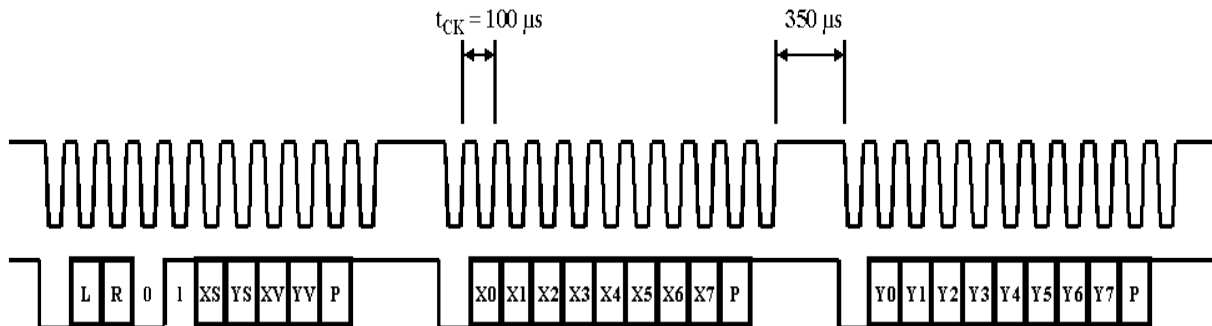


Figure .3 : Trame

Le premier mot de la trame fournit les informations suivantes :

- L : Bouton Gauche (actif à 1)
- R : Bouton Droit (actif à 1)
- XS : Direction du déplacement horizontal (1 pour Gauche)
- YS : Direction du déplacement horizontal (1 pour Bas)
- XV : Débordement suivant X
- YV : Débordement suivant Y

Les 2 mots suivants transportent la valeur du déplacement :

- X[7..0] : Déplacement Horizontal (Entier relatif, positif à droite)
- Y[7..0] : Déplacement Horizontal (Entier relatif, positif en bas)

II. Circuit interne d'une souris PS2 (Circuit d'émission et de détection) :

1. Circuit d'émission (Emetteurs LED rouge et LED infrarouge) :

L'émetteurs LED rouge et LED infrarouge :

L'émission de la lumière par une jonction a été observée en 1923 sur le carbure de silicium, mais il a fallu attendre 1953 pour avoir l'explication du phénomène avec la théorie des semi-conducteurs. C'est enfin vers 1963 que les premières diodes électroluminescentes LED ont été commercialisées.

Chapitre I : Souris PS2

Pour la réalisation du circuit d'émission, des diodes électroluminescentes sont utilisées pour l'émission des lumières rouge et infrarouge qui traverseront le site de mesure (doigt ou orteil) afin de caractériser l'onde de pouls sanguine. Ces lumières seront captées par une photodiode ou un phototransistor qui constitue l'élément principal du circuit de réception.

1.1 . Principe de fonctionnement de la LED:

Le mot LED est l'acronyme de Light Emitting Diode (Diode Electroluminescente en français). Le symbole de la LED ressemble à celui de la diode mais on y a ajouté deux flèches sortantes pour représenter le rayonnement lumineux émis. (figure4 ci-dessous)



Figure 4: Symbole de la LED.

Ce phénomène d'électroluminescence sera obtenu à la condition de créer une forte quantité d'électrons dans la bande de conduction. On l'obtient par injection de porteurs polarisant dans le sens direct, une jonction PN à semi- conducteur.

Le même résultat aurait pu être obtenu en irradiant le cristal avec une source lumineuse d'énergie importante (photoluminescence) ou par bombardement électronique (cathodoluminescence).

Selon la fabrication, la lumière peut être émise soit latéralement, soit perpendiculairement à travers la mince couche N ou P. (figure 5)

Chapitre I : Souris PS2

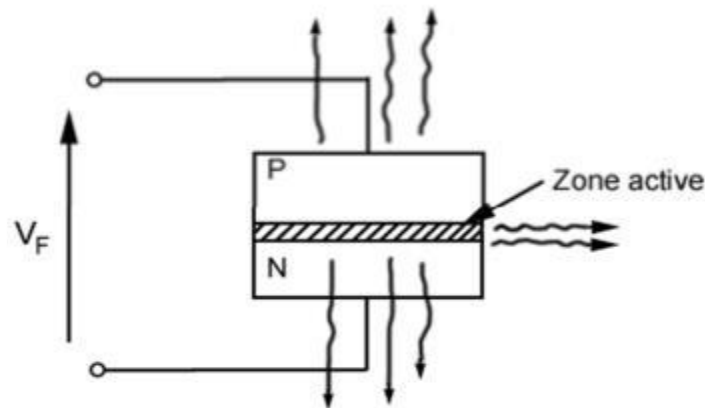


Figure 5: Fonctionnement de la LED

Caractéristiques optiques :

Longueur d'onde du pic d'émission :

Cette valeur nous indique la longueur d'onde (λ_p), en nanomètre, à laquelle est émis la plus importante partie du rayonnement (wavelength). La valeur est donnée pour une intensité de courant (IF).

Spectre ou largeur spectrale à mi- intensité :

Le spectre d'émission d'une diode LED est relativement étroit. Exemple : pour une longueur d'onde à intensité maximale égale à 520 nm, la longueur d'onde à intensité moitié pourra être comprise de 505 nm à 535 nm (soit une largeur spectrale de 30 nanomètres).

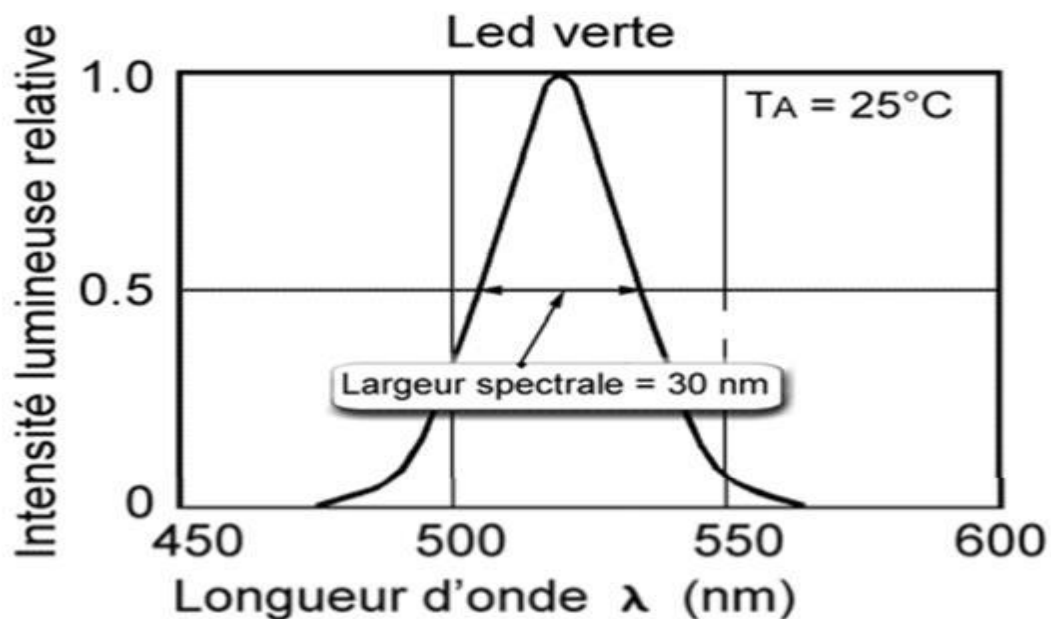


Figure 6 : Spectre d'émission de la LED

Chapitre I : Souris PS2

Il existe actuellement plusieurs types de LED donnant chacun des spectres différents. Cela est obtenu par la variété des semi-conducteurs utilisés pour fabriquer les jonctions PN. Des exemples sont donnés dans le tableau suivant pour l'obtention de certaines longueurs d'ondes :

| Matériaux | Rayonnement | Longueur d'onde |
|-----------|-------------|------------------------------|
| InAs | UV | 315 nm ou 3,15 μm |
| InP | infra-rouge | 910 nm |
| GaAsP4 | Rouge | 660 nm |
| GaAsP82 | Jaune | 590 nm |
| GaP | Vert | 560 nm |

Correspondance couleurs, longueurs d'onde et énergie des photons :

| Couleur | Longueur d'onde (nm) | Energie des photons (ev) |
|-------------|----------------------|--------------------------|
| Ultraviolet | < 390 | > 3.18 |
| Violet | 390-455 | 2.72-3.18 |
| Bleu | 455-490 | 2.53-2.72 |
| Cyan | 490-515 | 2,41-2,53 |
| Vert | 515-570 | 2,18-2,41 |
| Jaune | 570-600 | 2,06-2,18 |
| Orange | 600-625 | 1,98-2,06 |
| Rouge | 625-720 | 1,72-1,98 |
| Infrarouge | > 720 | < 1,72 |

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 :

- forme de la partie émissive (point, trait...),
- avec lentille de concentration ou sans,
- diffusante ou non.

Cette répartition est définie par le diagramme de rayonnement qui représente la répartition angulaire de l'intensité relative émise.

Chapitre I : Souris PS2

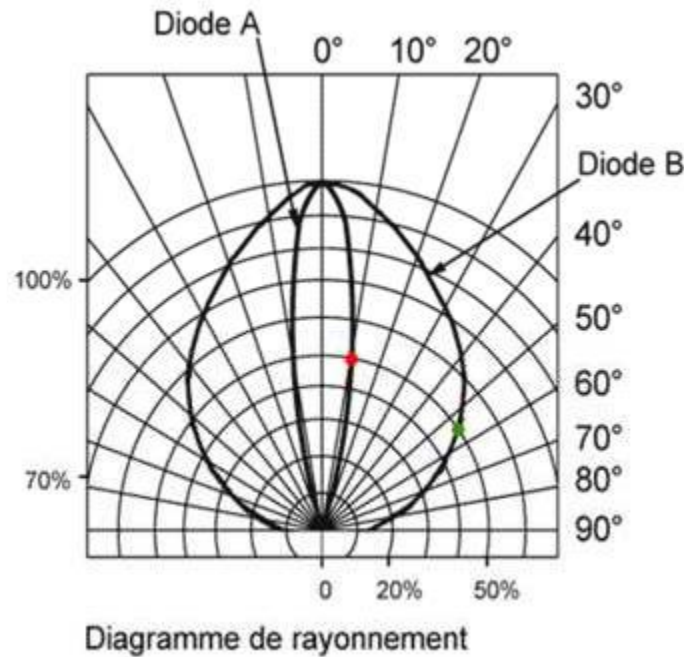


Figure 7 : Diagramme de rayonnement

Angle d'émission à mi- intensité :

Les fabricants précisent souvent l'angle pour lequel l'intensité lumineuse a été réduite de moitié. Sur le diagramme figure 7 ci-dessus, le point rouge indique un angle de 10 degrés et le point vert un angle de 50° pour une intensité relative émise de 50%.

Intensité lumineuse :

L'intensité lumineuse (mesurée en candelas) est la quantité de lumière émise dans une certaine direction à 1 mètre de distance. Dans les caractéristiques optiques des LEDs nous l'exprimons aussi en micro-candela (mcd) et se note IV.

Caractéristiques électriques :

Point de fonctionnement et tension directe :

Une LED se comporte électriquement comme une diode. Pour émettre elle doit être polarisée en direct.

Chapitre I : Souris PS2

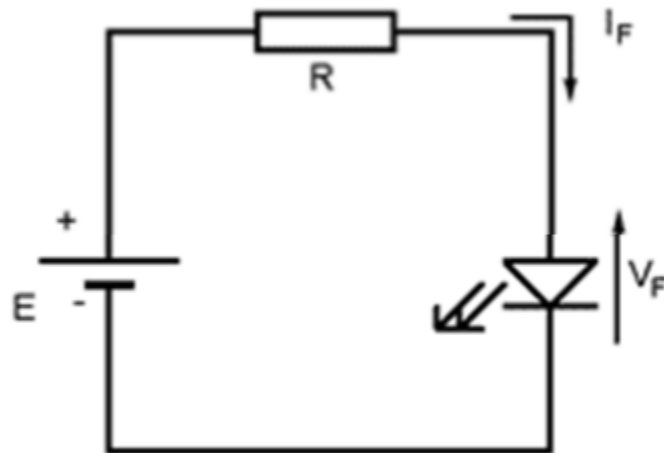


Figure 8: Polarisation de la LED

La caractéristique $I = f(V)$, (figure 9) montre que la tension de conduction de la diode LED (forward voltage) est environ 1,5 Volts à 2 V.

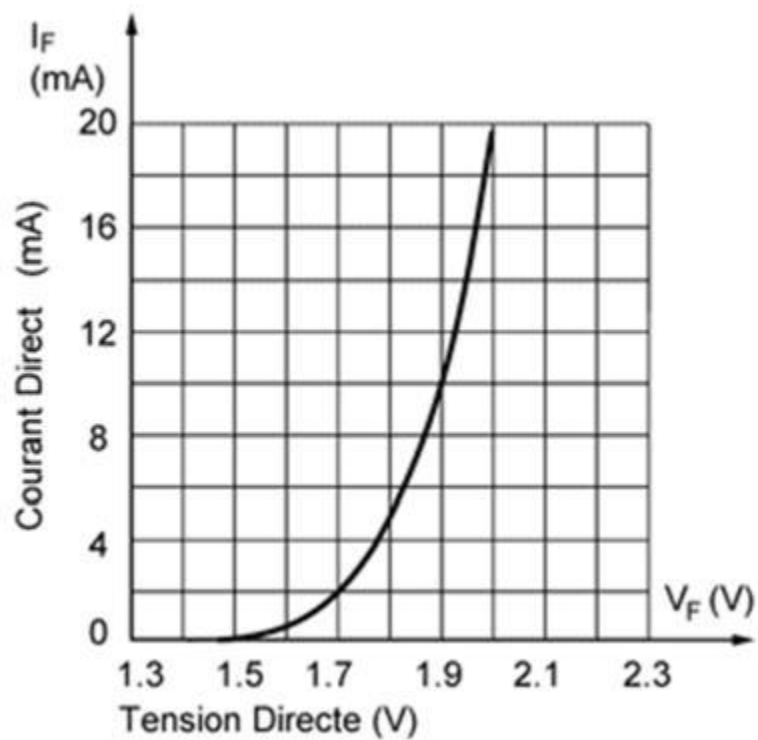


Figure 9 : Caractéristique $I_F - V_F$

Le courant I_F vaut environ $E - 2V/R$. En pratique, le constructeur préconise 10 à 20 mA. Le courant traversant la LED détermine l'intensité lumineuse émise.

Chapitre I : Souris PS2

1. Point de fonctionnement et Tension inverse VR:

Dans certains cas, on peut avoir besoin de polariser en inverse la LED. La diode est alors éteinte : elle n'émet plus d'intensité lumineuse. Mais attention, la diode LED ne peut pas supporter des tensions inverses trop importantes comme une diode de redressement par exemple. Les valeurs courantes se situent telles que $VR_{max} = \pm 3V$ à $5V$ (reverse voltage) ; au-delà de ces valeurs il y a endommagement ou destruction du composant. En cas de besoin nous plaçons une diode normale en série avec la LED.

Il est alors extrêmement intéressant d'alimenter la LED en courant pulsé au lieu du courant continu. La valeur crête du courant permet alors d'obtenir des intensités lumineuses importantes. De ce fait nous pouvons :

- augmenter l'intensité lumineuse émise à consommation électrique moyenne égale
- Diminuer la consommation électrique tout en obtenant une intensité lumineuse égale
- réduire l'échauffement de la jonction.

2. Circuit de détection :

Les photos détectrices transforment les radiations lumineuses (visible ou non) en signaux électriques. On les nomme aussi détecteur optique, photo coupleurs ou capteur optique. Le symbole d'une photodiode est celui d'une diode, auquel on a ajouté deux flèches pour symboliser l'action du rayonnement (figure 10.a).

Le phototransistor quant à lui est représenté en (figure 10.b)



Figure 10.a : symbole d'une photodiode



Figure 10 b: Phototransistor

Chapitre I : Souris PS2

La photodiode :

Une photodiode est un composant semi-conducteur ayant la capacité de détecter un rayonnement du domaine optique et de le transformer en signal électrique.

Principe de fonctionnement :

Quand un semi-conducteur est exposé à un flux lumineux, les photons sont absorbés à condition que l'énergie du photon (E_{ph}) soit supérieure à la largeur de la bande interdite (E_g). Ceci correspond à l'énergie nécessaire que doit absorber l'électron afin qu'il puisse quitter la bande de valence (où il sert à assurer la cohésion de la structure) vers la bande de conduction, le rendant ainsi mobile et capable de générer un courant électrique. L'existence de la bande interdite entraîne l'existence d'un seuil d'absorption tel que $E_{ph} = E_g$. Lors de l'absorption d'un photon, deux phénomènes peuvent se produire :

La photoémission : C'est la sortie de l'électron hors du matériau photosensible. L'électron ne peut sortir que s'il est excité près de la surface.

La photoconductivité : l'électron est libéré à l'intérieur du matériau. Les électrons ainsi libérés contribuent à la conductivité électrique du matériau.

Lorsque les photons pénètrent dans le semi-conducteur munis d'une énergie suffisante, ils peuvent créer des photos porteuses (électrons et trous d'électrons) en excès dans le matériau. On observe alors une augmentation du courant. Deux mécanismes interviennent simultanément :

Il y a création de porteurs minoritaires, c'est-à-dire des électrons dans la région P et des trous dans la région N. Ceux-ci sont susceptibles d'atteindre la ZCE par diffusion et d'être ensuite propulsés vers des zones où ils sont majoritaires. En effet, une fois dans la ZCE, la polarisation étant inverse, on favorise le passage des minoritaires vers leur zone de prédilection. Ces porteurs contribuent ainsi à créer le courant de diffusion. Il y a génération de paires électrons trou dans la ZCE, qui se dissocient sous l'action du champ électrique ; l'électron rejoignant la zone N, le trou la zone P.

Ce courant s'appelle le courant de transit ou photo courant de génération.

Ces deux contributions s'ajoutent pour créer le photo-courant I_{ph} qui s'additionne au courant inverse de la jonction.

Chapitre I : Souris PS2

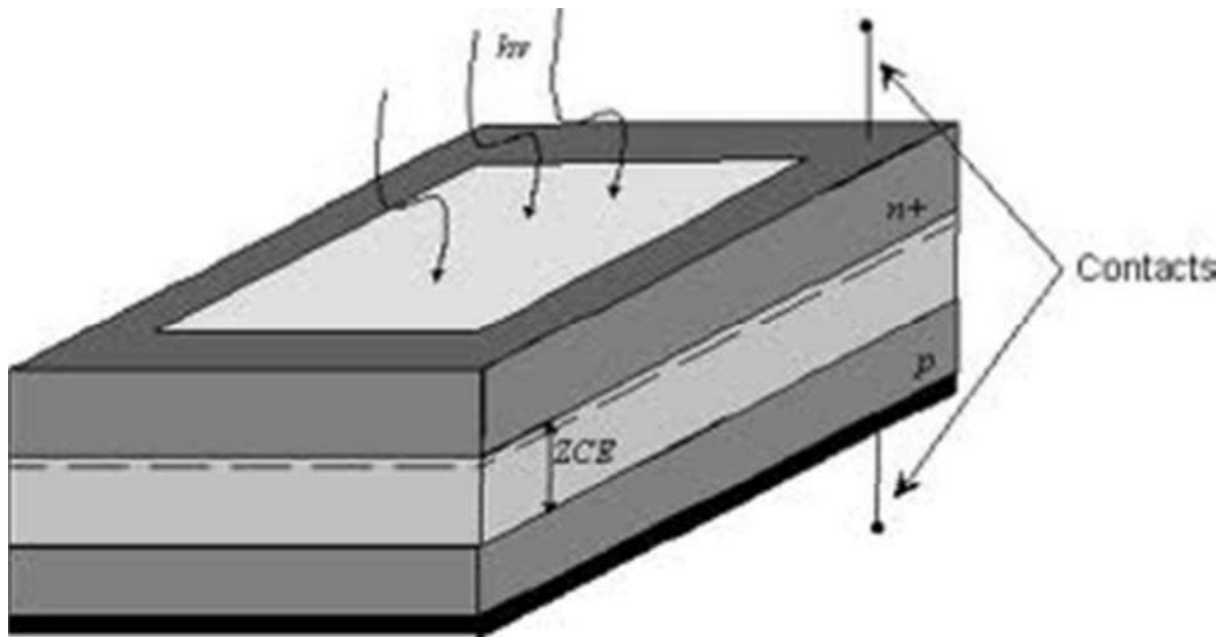


Figure 11 : Structure d'une photodiode

Association émetteur – récepteur I.R :

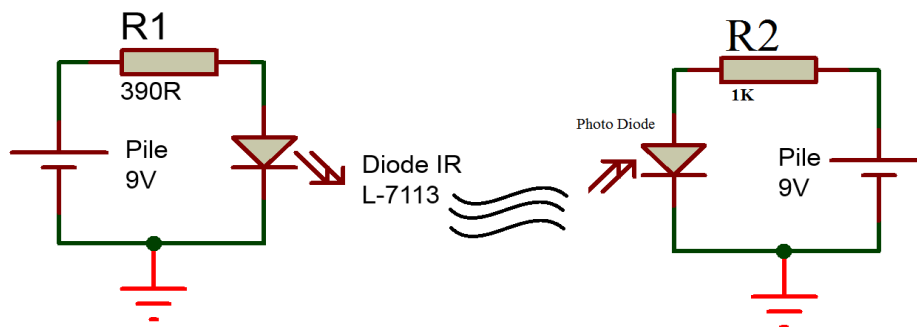


Figure 12 : émetteur-récepteur

On place les composants D.I.R. et P.D. très proches l'un de l'autre

D.I.R. : Diode émettrice I.R. c'est une L.E.D. mais qui émet un rayonnement dans l'infrarouge.

Chapitre I : Souris PS2

Photodiode P.D. : c'est une diode sensible à un rayonnement, dans notre cas, un rayonnement infrarouge. Elle a pour propriété de conduire le courant en inverse lorsqu'elle est éclairée. Ce courant est très faible (quelques μA)

Le phototransistor :

Principe de fonctionnement :

Un phototransistor est un transistor bipolaire dont la base est sensible au rayonnement lumineux ; la base est alors dite flottante puisqu'elle est dépourvue de connexion. Lorsque la base n'est pas éclairée, le transistor est parcouru par le courant de fuite I_{CE0} . L'éclairement de la base conduit à un photo-courant I_{ph} que l'on peut appeler courant de commande du transistor.

Celui-ci apparaît dans la jonction collecteur-base sous la forme :

$$I_C = \beta I_{ph} + I_{CE0}$$

Pour simplifier, lorsque la base est éclairée le phototransistor est équivalent à un interrupteur fermé entre l'émetteur et le collecteur et lorsque la base n'est pas éclairée, c'est équivalent à un interrupteur ouvert.

Le courant d'éclairement du phototransistor est le photo-courant de la photodiode collecteur-base multiplié par l'amplification β du transistor. Sa réaction photosensible est donc nettement plus élevée que celle d'une photodiode (de 100 à 400 fois plus). Par contre le courant d'obscurité est plus important.

On observe une autre différence entre le phototransistor et la photodiode : la base du phototransistor est plus épaisse, ce qui entraîne une constante de temps plus importante et, donc une fréquence de coupure plus basse que celle des photodiodes.

On peut éventuellement augmenter la fréquence de coupure en diminuant la photosensibilité en connectant la base de l'émetteur.

Chapitre II : La Carte Arduino

1-Présentation :

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. L'ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C/C++.

L'intérêt principal des cartes ARDUINO (d'autres modèles existent : Méga, Nano...) est leur facilité de mise en oeuvre. Un environnement de développement (IDE), s'appuyant sur des outils open-source, est fourni. En outre, charger le programme compilé dans la mémoire du microcontrôleur se fait très simplement (via par port USB) dans cet IDE. Enfin, beaucoup de bibliothèques de fonctions sont également fournies pour l'exploitation des entrées-sorties courantes : E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I²C, exploitation de servomoteurs, d'afficheurs LCD ...

❖ Ça sert à quoi



Fig13 : Contexte d'utilisation

❖ Arduino : C'est pour qui ?

1

- Le projet « Arduino » a été initié par un groupe d'enseignants et d'étudiants d'une école de design italienne en 2004 –2005.

Chapitre II : La Carte Arduino

❖ Arduino : Qu'est ce que s'est ?

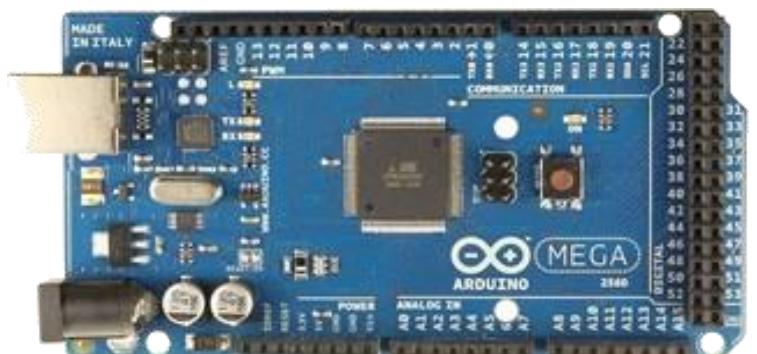
- Une plate-forme de développement et de prototypage Open Source.

Le rôle de la carte Arduino est de stocker un programme et de le faire fonctionner des Shields (cartes d'extension) avec des fonctions diverses qui s'enfichent sur la carte Arduino:

- Relais, commande de moteurs, lecteur carte SD,...
- Ethernet, WIFI, GSM, GPS,...
- Afficheurs LCD, Écran TFT,...



Uno : 5,33 x 6,86 cm



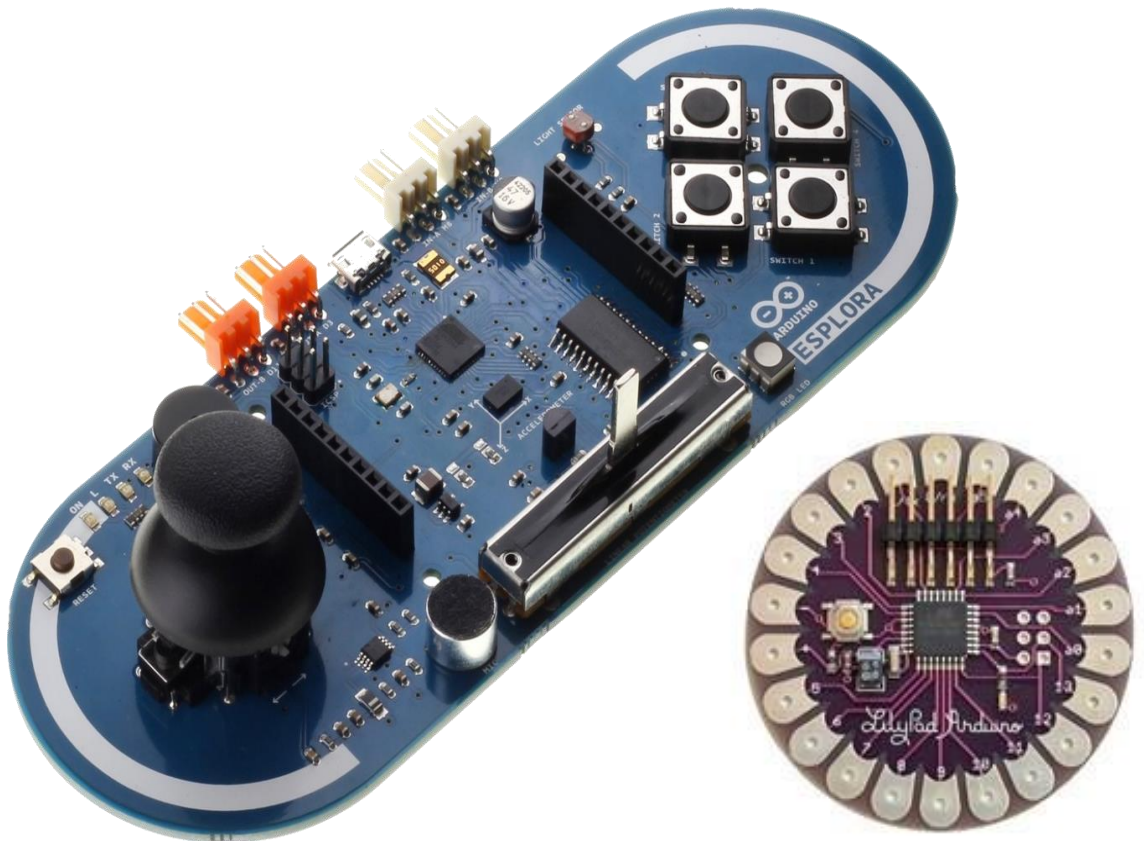
Mega : 5,33 x 10,16 cm



Nano : 1,85 x 4,32 cm

Figure 14a : Différentes cartes Arduino

Chapitre II : La Carte Arduino



Esplora : 6,1 x 16,51 cm

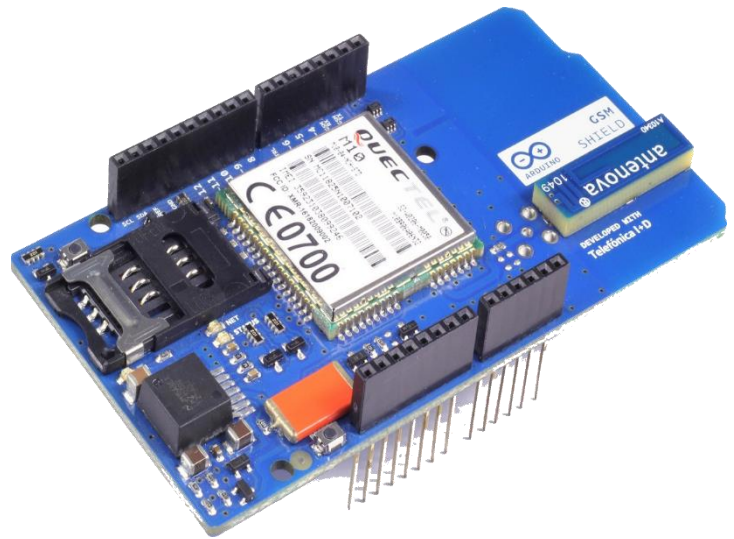
LilyPad : Ø 5 cm

Fig14b : Les cartes d'extension arduino

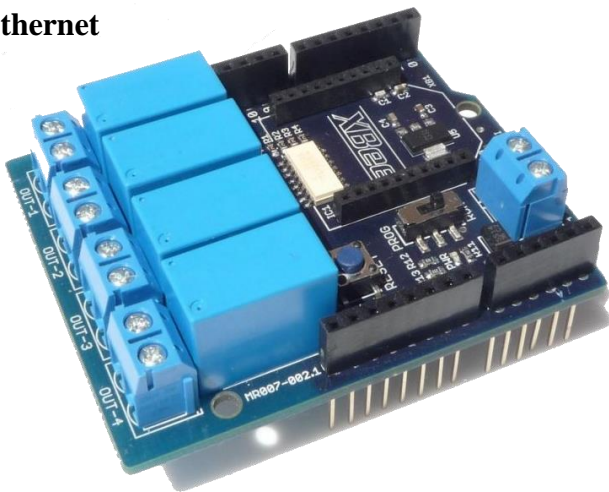
Chapitre II : La Carte Arduino



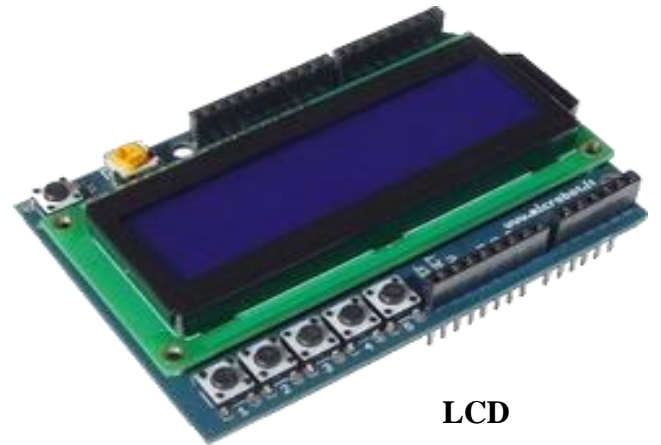
Ethernet



GSM



Relais



LCD

Fig14c : Shields Arduino (cartes d'extension)

Chapitre II : La Carte Arduino

❖ Schéma simplifié de la carte ArduinoUNO

Les broches du microcontrôleur sont reliées à des connecteurs selon le schéma ci-dessous

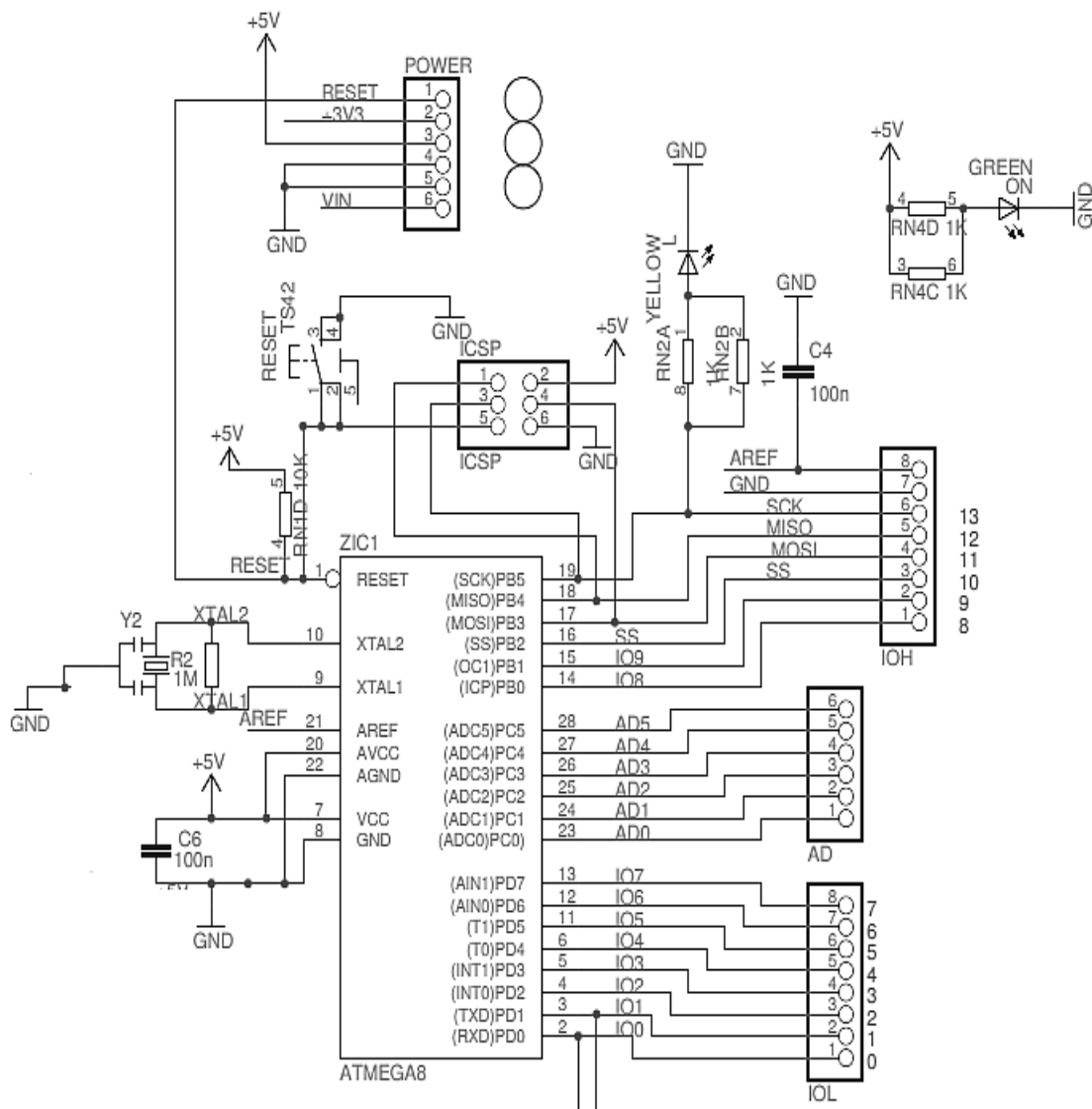


Figure 15 : Schéma simplifié (Arduino)

Chapitre II : La Carte Arduino

2. Microcontrôleur ATmega328 :

L'ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C/C++ afin de transmettre ou lire une valeur 0 ou 1 sur ses ports GPIO.

2.1. Caractéristiques du L'Atmega 328 :

Le microcontrôleur de la carte Arduino UNO est un **ATmega328**. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Les principales caractéristiques sont :

FLASH = mémoire programme de 32Ko
SRAM = données (volatiles) 2Ko
EEPROM = données (non volatiles) 1Ko

Digital I/O (entrées-sorties Tout Ou Rien) = 3 ports PortB, PortC, PortD

(soit 23 broches en tout I/O)

Timers/Counters: Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits). Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)

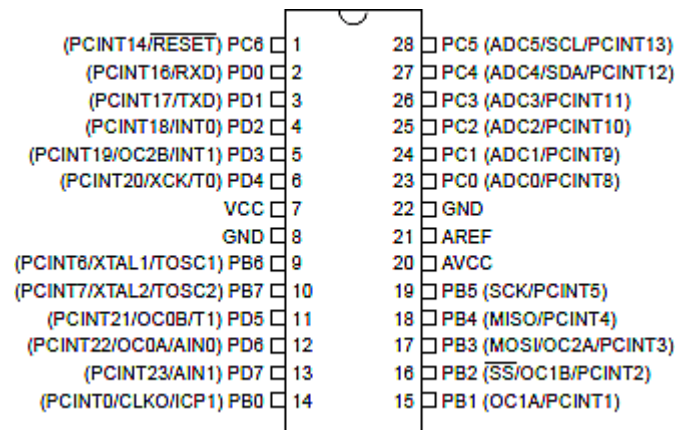


Fig16 : ATmega328

Plusieurs broches multi-fonctions : certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage.

Par exemple, les broches PB1, PB2, PB3, PD3, PD5, PD6 peuvent servir de sortie PWM (Pulse WidthModulation), c'est-à-dire des sorties qui joueront le rôle de sorties analogiques. Elles correspondent aux broches des connecteurs 3,5,6,9,10 et 11. Cet autre rôle possible est lié aux timers et ces broches sont alors appelées OCxA ou OCxB

Chapitre II : La Carte Arduino

dans la documentation. Ce sont les mêmes broches, mais pour une autre fonction.

PWM = 6 broches **OC0A(PD6)**, **OC0B(PD5)**, **OC1A(PB1)**, **OC1B(PB2)**,
OC2A(PB3), **OC2B(PD3)**

Les broches du PORT C peuvent être converties par un convertisseur Analog to Digital.

Analog to Digital Converter (résolution 10bits) = 6 entrées multiplexées **ADC0(PC0)** à
ADC5(PC5)

Gestion bus I2C (TWI Two Wire Interface) = le bus est exploité via les broches
SDA(PC5)/SCL(PC4).

Port série (USART) = émission/réception série via les broches
TXD(PD1)/RXD(PD0)

Comparateur Analogique = broches **AIN0(PD6)** et **AIN1 (PD7)** peut déclencher
interruption **Watchdog Timer programmable**.

Gestion d'interruptions (24 sources possibles (cfinterruptvectors)) : en résumé

- Interruptions liées aux entrées **INT0 (PD2)** et **INT1(PD3)**
- Interruptions sur changement d'état des broches **PCINT0 àPCINT23**
- Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
- Interruption liée au comparateur analogique
- Interruption de fin de conversion **ADC**
- Interruptions du port série **USART**
- Interruption du bus **TWI(I2C)**

Chapitre II : La Carte Arduino

2.2. Structure interne de l'ATMega328 :

L'utilisation des périphériques intégrés (Entrées Sorties TOR, Timers, ...) repose sur l'exploitation (lecture/écriture) de registres internes. Ces registres, essentiellement 8 bits, sont décrits par un nom y compris dans la programmation en C. Cette section fournit quelques détails importants sur les registres internes du microcontrôleur ATMega328.

Notation : par la suite, pour un registre nommé **R**, la notation **R.n** désigne le bit de rang n du registre R. Le bit R.0 est le bit de poids faible de R.

2.2.1. Status Register (SREG) :

Le registre **SREG** contient des indicateurs liés aux opérations et le bit d'autorisation générale des interruptions. Les bits de ce registre sont : **Z** (Zero), **C** (Carry), **S** (Sign) ...

Le bit d'activation général du système d'interruptions est le bit **I** (**SREG.7**)

SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

| | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0x3F (0x5F) | I | T | H | S | V | N | Z | C | SREG |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

Figure 17 : Le registre SREG

Note : en langage C, ce bit I est modifiable via les appels **sei()** (set IT) **cli()** (Clear IT)

Chapitre II : La Carte Arduino

2.2.2 . Digital I/O Entrées Sorties Binaires/Tout Ou Rien(TOR) :

Les microcontrôleurs disposent de broches d'entrée/sortie TOR, comme sur un automate programmable industriel. Pour placer l'état d'une sortie à 0 ou 1, ou lire l'état d'une entrée, il faut exploiter des registres internes décrits ci-dessous.

Les entrées-sorties sont réparties dans 3 groupes de broches appelés *ports*. Le port B regroupe les broches notées PBx, le port C les broches PCx et le port D les broches PDx (voir brochage). Chaque port est exploité grâce à 3 registres.

Ex: PORTB, DDRB et PINB les registres pour la gestion des broches PB0 à PB7

PORTx= pour l'*ECRITURE* de valeurs en sortie

DDRx= détermine la *DIRECTION* de chaque broche du port (1-sortie 0-entrée)

PINx= pour la *LECTURE* de la valeur en entrée

2.2.3 Timers/Counters de ATmega328

Le microcontrôleur ATmega328 dispose de plusieurs modules de temporisation/comptage internes (Timers), fonctionnant pour certains avec des registres de comptage sur 8 bits, et pour d'autres sur 16 bits. Dans tous les cas, chaque événement de comptage conduit à une modification du registre de comptage (+1). L'événement de comptage peut être un "tick" de l'horloge du microcontrôleur, ce qui revient à mesurer l'écoulement du temps. L'événement de comptage peut aussi être un front sur une broche d'entrée du microcontrôleur (les broches T0 et T1 peuvent servir d'entrée de comptage).

- ✓ **Fonction Temporisateur** : lorsque l'on compte des "ticks" de l'horloge qui cadence le microcontrôleur, on mesure du temps. Les modules Timers/Counters permettent de compter les ticks du signal d'horloge, ou un signal de fréquence plus faible obtenu par un diviseur appelé **prescaler**. C'est la fonction temporisateur qui va être détaillée par la suite.

Chapitre II : La Carte Arduino

Note : Sur la carte Arduino UNO, l'horloge est à **16MHz**, soit 16 000 000 de cycles horloge par seconde, ou 16 cycles horloge par micro seconde. Ce sont ces cycles là qui sont comptés en fonction temporisateur. Il faut compter 16000000 cycles pour faire une seconde.

- ✓ **Fonction Compteur :** lorsque l'on compte des fronts sur une entrée de comptage (broches T0 ou T1), on utilise alors la fonction "compteur" du module.

Le choix entre fonction de temporisation (avec prédiviseur de fréquence ou non) et fonction de comptage se fait par paramétrage de registres dédiés à la gestion des modules Timers/Counters.

- ✓ **Génération de signaux périodiques :** les modules Timers/Counters sont assez complexes et chacun de ces modules peut générer deux signaux PWM (Pulse Width Modulation) dont le rapport cyclique est facilement modifiable. Dans ce cas, utiliser la fonction Arduino `analogWrite` qui génère un signal PWM. Ce signal PWM n'est géré que sur les sorties liées à des Timers intégrés c'est-à-dire PD6, PD5, PD3, PB1, PB2 et PB3.

- **Timer/Counter 0 (comptage 8bits) :**

C'est un module Timer/Counter avec registre de comptage 8 bits. En utilisant l'IDE Arduino, le timer 0 (et l'interruption associée) est implicitement utilisé par les fonctions de gestion du temps (`delay()`, `millis()` ...). Ce module Timer/Counter n'est donc pas utilisable directement avec la carte ARDUINO Uno. Sauf si l'on accepte de se passer des fonctions de gestion de temps Arduino.

- **Timer/Counter 1 (comptage 16bits) :**

Le registre de comptage TCNT1, ainsi que les registres de comparaison OCR1A et OCR1B, sont cette fois-ci sur 16 bits.

Chapitre II : La Carte Arduino

Note: en langage d'assemblage, il faut deux accès 8 bits pour lire/écrire ces registres 16 bits. En langage C, on peut manipuler symboliquement des données 16 bits via les symboles TCNT1, OCR1A et OCR1B sans se soucier de la façon dont le code sera généré.

- **Timer/Counter 2 (comptage 8bits) :**

C'est un module Timer/Counter avec registre de comptage 8 bits. Le registre de comptage est TCNT2 (registre 8 bits).

Chapitre III : l'afficheur LCD

1. Généralité:

▪ Historique :

Les écrans LCD, qui correspondent à « Liquid cristal display » qui signifie en français « affichage à cristaux liquides » ont révolutionnés le monde du numérique. C'est en 1964 qu'ils ont été inventés par George Harry Heilmeyer, un ingénieur et homme d'affaire américain. Il découvre de nouvelles propriétés électro-optiques dans les cristaux liquides permettant la création d'écran plusieurs années plus tard. En effet en 1971 les premiers écrans LCD sont présentés ; en 1984 le laboratoire Thomson développe le premier LCD en couleur. Mais c'est réellement en 1985 que le LCD commence à être utilisé sur des micro-ordinateurs.

L'afficheur LCD est en particulier une **interface visuelle entre un système** (projet) **et l'homme** (utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il **affichera donc des données** susceptibles d'être exploiter par l'utilisateur d'un système.

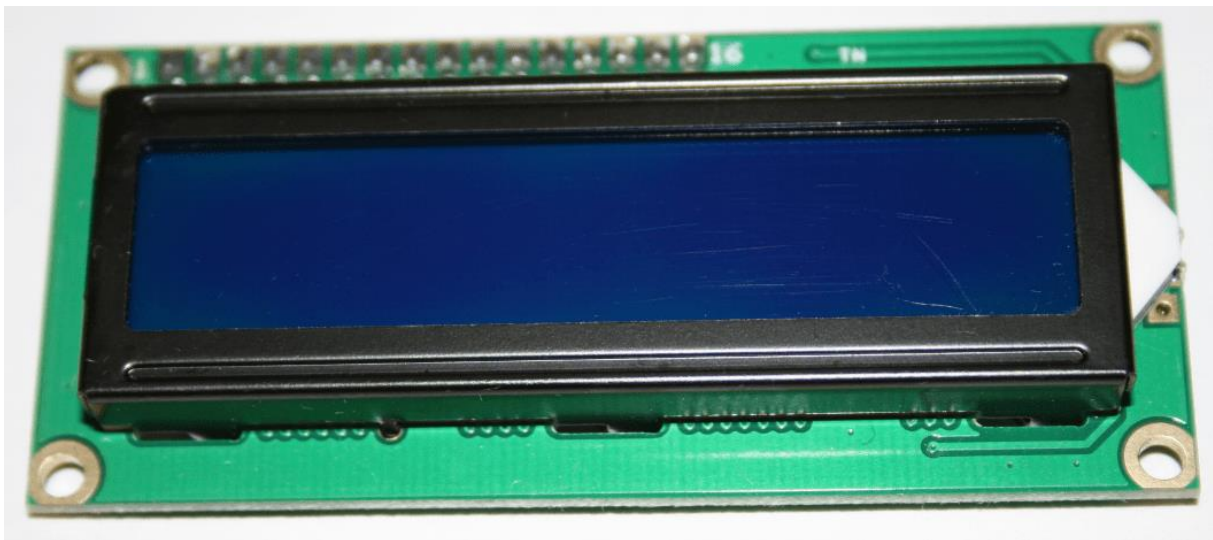


Fig18 : Afficheur LCD(16,2)

Chapitre III : l'afficheur LCD

- **Principe de fonctionnement d'un écran LCD :**

Pour comprendre comment fonctionne un écran LCD, il faut d'abord savoir que les cristaux liquides ont des propriétés polarisantes (ce qui signifie qu'ils peuvent donner une direction spécifique à la lumière). Lorsqu'ils sont sous l'action d'un champ électrique, ils s'orientent selon les lignes de ce champ. Il est donc possible de les diriger en changeant la direction du champ électrique qu'on leur applique.

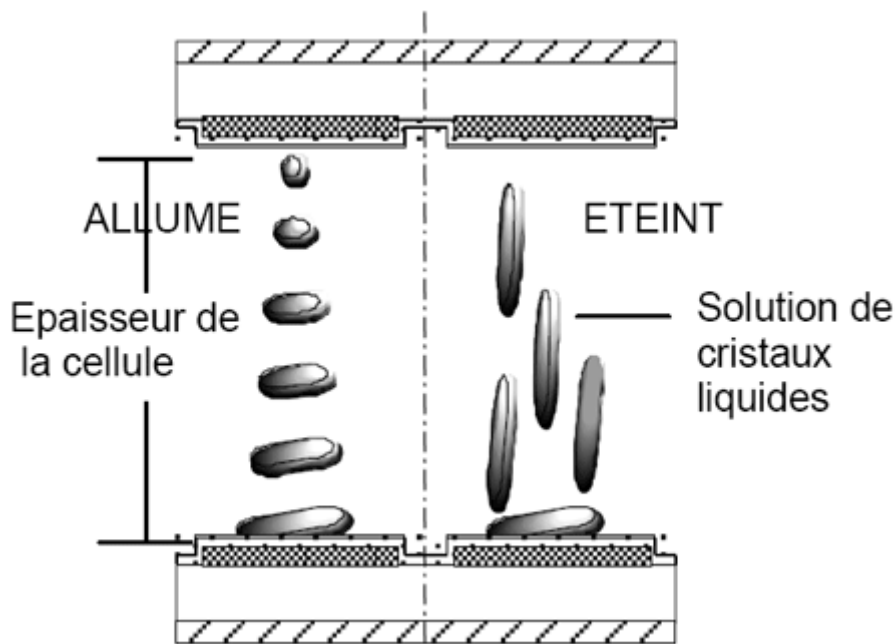


Fig19a : Principe de fonctionnement

Chapitre III : l'afficheur LCD

Comme nous le savons un écran est divisé en petites unités lumineuses : les pixels. Chaque pixel est subdivisé en trois rectangles appelés sous-pixels. La seule différence entre ces sous-pixels provient de la couleur du filtre qui le recouvre : rouge, vert ou bleu.

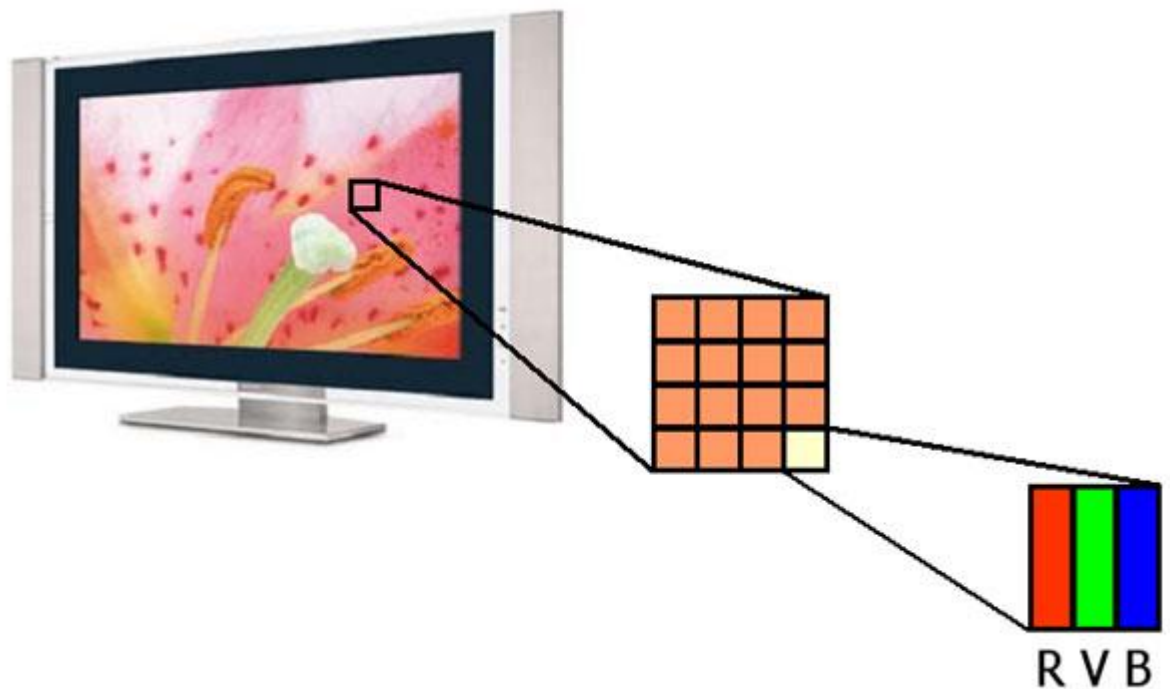


Fig19b : principe de fonctionnement

Un tube fluorescent ou un panneau de DEL, placé à l'arrière de l'écran produit de la lumière blanche.

Pour chaque sous-pixel, une couche de cristaux liquides est placée entre 2 polariseurs. L'ensemble laisse passer plus ou moins de lumière suivant la tension électrique appliquée aux cristaux liquides. L'ensemble joue le rôle de store : la quantité de lumière qu'il laisse passer varie en fonction de la tension électrique appliquée aux cristaux liquides. Cette lumière est ensuite colorée en traversant un filtre correspondant à la couleur du sous-pixel : rouge, vert ou bleu.

Chapitre III : l'afficheur LCD

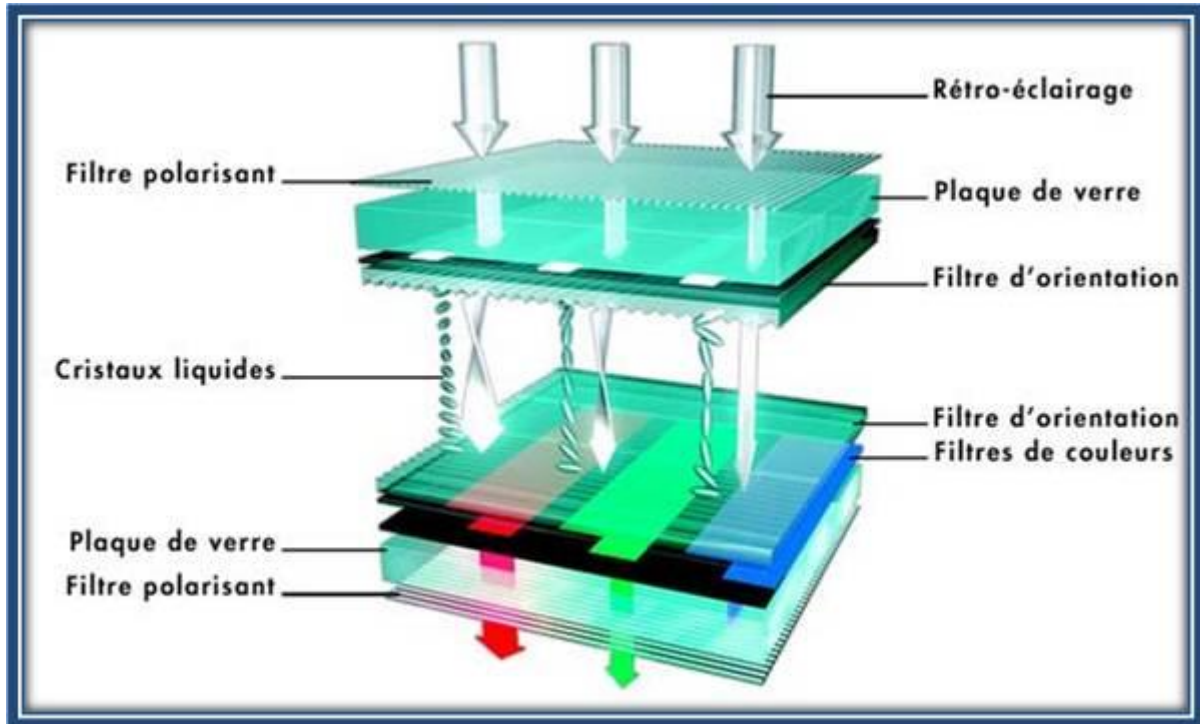


Fig19c : principe de fonctionnement

Nos yeux ne distinguent pas les différents sous-pixels, ils mélangent les lumières colorées primaires (rouge, verte et bleu) provenant de chacun d'eux. Cela suffit pour reproduire pratiquement toutes les couleurs perçues par l'œil humain : c'est ce qu'on appelle la synthèse additive des couleurs.

- **Avantages et inconvénients :**

Chapitre III : l'afficheur LCD

Les écrans LCD possèdent de nombreux avantages comme la possibilité d'avoir une grande luminosité ainsi qu'une faible consommation d'énergie, en effet celle-ci est essentiellement due à l'éclairage. Ils ont aussi des prix très abordables car ils sont moins chers que le plasma et sont légers et facilement manipulables. Ils possèdent aussi malheureusement des défauts : ils sont difficiles à fabriquer et leur taille est limitée car plus l'écran est grand, plus il nécessite de pixels et plus les chances de dysfonctionnement sont grandes. Enfin l'affichage du noir tend vers le gris car les cristaux laissent toujours échapper un peu de lumière.

2. Les caractéristiques :

La première information à connaître est le **nombre de caractères affichables par ligne**. Pour ce modèle, c'est 16 caractères sur deux lignes soit au total 32 caractères. De toute évidence, on retrouve cette information **dans le datasheet sous la forme 16 X 02**. Mais aussi dans la référence **1602 A**.

On recueille également l'information du mode de **transmission de données sur quatre (4)**

1. 0 FEATURES

- Display Mode: STN, BLU
- **Display Format: 16 Character x 2 Line**
- Viewing Direction: 6 O'Clock
- **Input Data: 4-Bits or 8-Bits interface available**
- Display Font : 5 x 8 Dots
- Power Supply : Single Power Supply (5V±10%)
- Driving Scheme : 1/16Duty,1/5Bias
- BACKLIGHT (SIDE) : LED (WHITE)

ou huit (8) bits.

Fig20 : caractéristiques

En outre, les informations suivantes à connaître sont les tensions d'exploitation, les caractéristiques électriques et mécaniques de l'afficheur LCD.

Chapitre III : l'afficheur LCD

2.0 ABSOLUTE MAXIMUM

| Item | Symbol | Min. | Max. | Unit |
|----------------------------|--------|----------|---------|------|
| Power Supply for logic | Vdd | -0.3 | +7.0 | V |
| Power supply for LCD Drive | Vlcd | Vdd-10.0 | Vdd+0.3 | V |
| Input Voltage | Vi | -0.3 | Vdd+0.3 | V |
| Operating Temperature | Ta | 0 | +50 | °C |
| Storage Temperature | Tstg | -10 | +60 | °C |

3.0 ELECTRICAL CHARACTERISTICS

(Ta=25°C; Vdd=3.0V±10%, otherwise specified)

| Item | Symbol | Test Condition | Min. | Typ. | Max. | Unit |
|---------------------------|--------|----------------|------|------|------|------|
| Power Supply for Logic | Vdd | -- | 4.7 | 5.0 | 5.5 | V |
| Operating Voltage for LCD | Vdd-Vo | -- | -- | 5.0 | -- | V |
| Input High voltage | Vih | -- | 2.2 | -- | Vdd | V |
| Input Low voltage | Vil | -- | -0.3 | -- | 0.6 | V |
| Output High voltage | Voh | -Ioh=0.2mA | 2.4 | -- | -- | V |
| Output Low voltage | Vol | Iol=1.2mA | -- | -- | 0.4 | V |
| Power supply current | Idd | Vdd=3.0v | -- | 1.1 | -- | mA |

4.0 MECHANICAL PARAMETERS

| Item | Description | Unit |
|----------------|---------------|------|
| PCB Dimension | 80.0*36.0*1.6 | mm |
| View Dimension | 69.5*14.5 | mm |

Fig21 :Tableau des tensions d'exploitations, caractéristiques électriques et mécaniques

Chapitre III : l'afficheur LCD

3. Le brochage :

Le tableau ci-dessous représente bien le rôle de chaque broche et définit le brochage de l'afficheur LCD .

5. 0 PIN ASSIGNMENT

| No. | Symbol | Level | Function |
|-----|--------|-------|---|
| 1 | Vss | -- | Power Supply |
| 2 | Vdd | -- | |
| 3 | V0 | -- | |
| 4 | RS | H/L | Register Select: H:Data Input L:Instruction Input |
| 5 | R/W | H/L | H--Read L--Write |
| 6 | E | H,H-L | Enable Signal |
| 7 | DB0 | H/L | Data bus used in 8 bit transfer |
| 8 | DB1 | H/L | |
| 9 | DB2 | H/L | |
| 10 | DB3 | H/L | Data bus for both 4 and 8 bit transfer |
| 11 | DB4 | H/L | |
| 12 | DB5 | H/L | |
| 13 | DB6 | H/L | |
| 14 | DB7 | H/L | |
| 15 | BLA | -- | BLACKLIGHT +5V |
| 16 | BLK | -- | BLACKLIGHT 0V- |

Fig22 : Tableau descriptif du brochage.

- Premièrement, les **broches 1, 2 et 3** sont dédiées à **l'alimentation**. La broche 1 et 2 pour l'alimentation générale. La broche 3 pour le branchement d'un potentiomètre qui **contrôle le contraste de l'afficheur**.
- Deuxièmement, les **broches 4, 5 et 6** pour le **pilotage de la transmission des données**. Elles pilotent l'écriture ou la lecture de données.
- Troisièmement, les **broches de 7 à 14** pour le **transfert des données elles-mêmes**.
- Et pour finir, les **broches 15 et 16** pour **l'alimentation du rétro-éclairage**.

Grâce à ces informations **nous connaissons le brochage de l'afficheur**. De plus nous avons en appui **un schéma de raccordement**.

Chapitre III : l'afficheur LCD

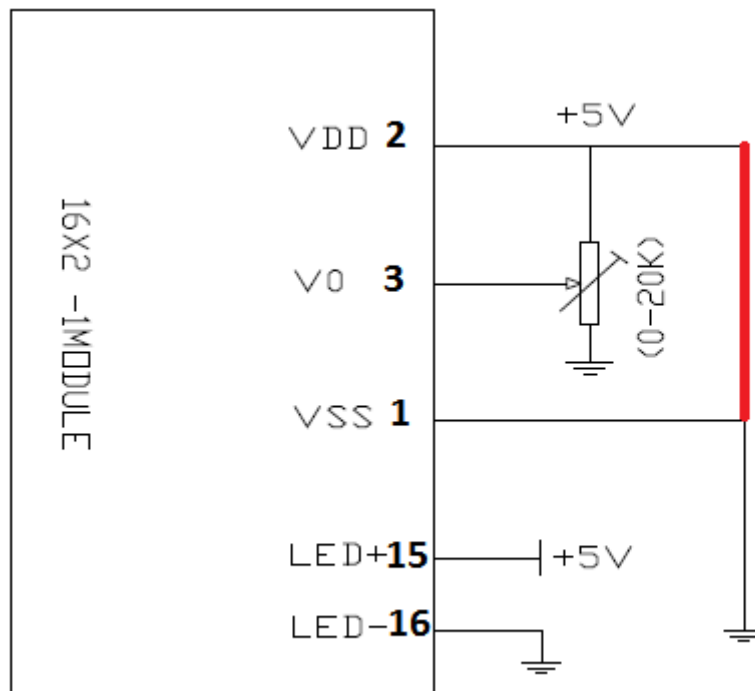


Fig23 : Schéma de câblage.

4. La transmissions de données :

Observez le schéma ci-dessous. Nous avons ajouté le brochage suivant les repères du tableau. De plus, vous constaterez bien-sûr, la représentation de deux "modules". Le **ST7066U** et le **ST7065C**.

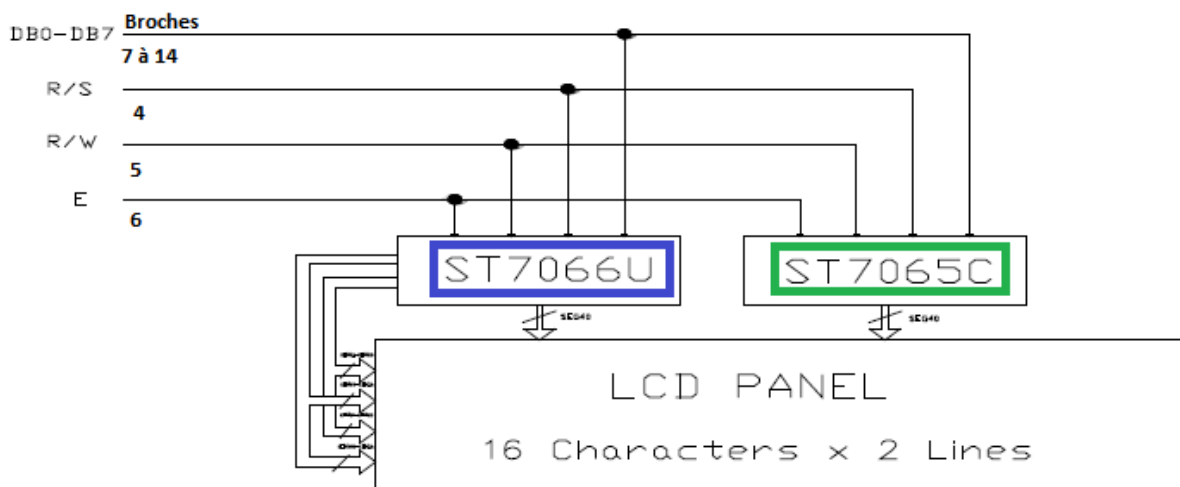


Fig 24:Schéma de transmission des données.

Chapitre III : l'afficheur LCD

Le ST7066U et le ST7065C sont des puces électronique utilisées en particulier pour piloter des matrices d'afficheurs LCD et capables de piloter 40 segments respectivement. Comme, par exemple, un MAX7219 avec une matrice à LED 8X8.

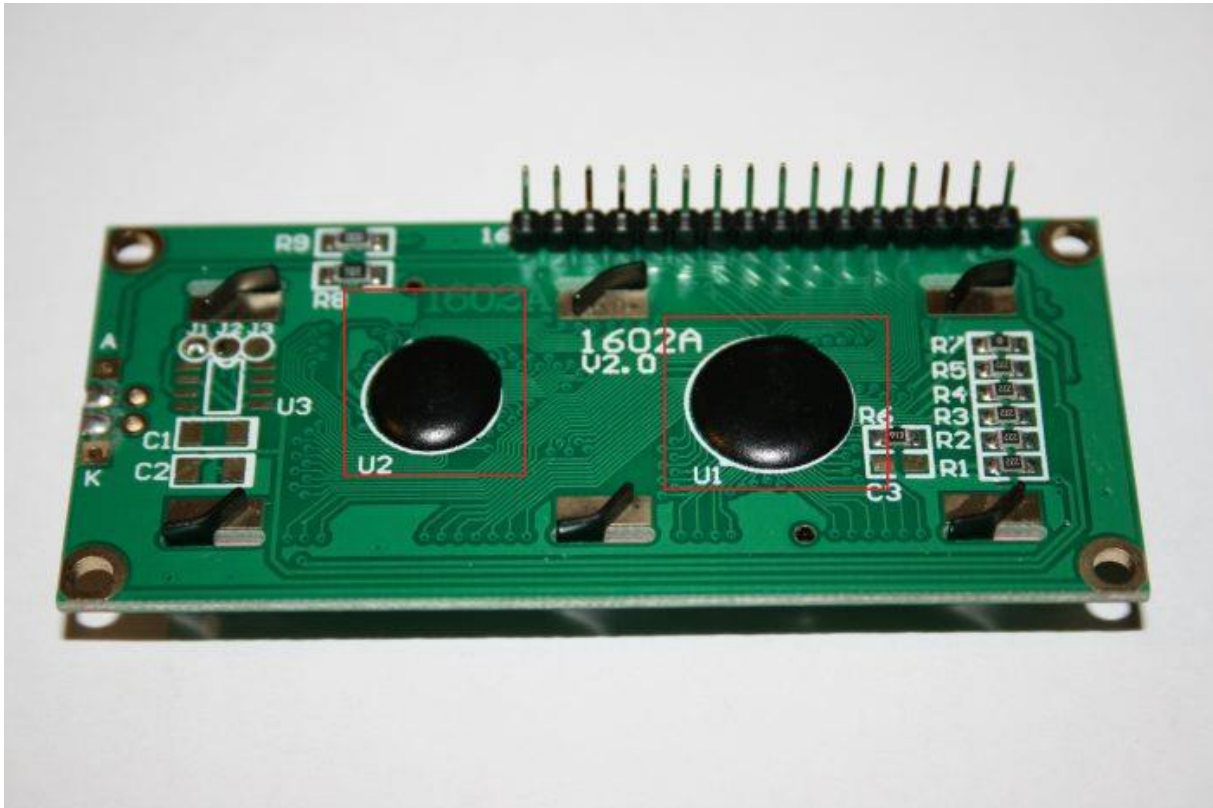


Fig25 : Localisation des deux puces

En effet l'afficheur est composé de plusieurs matrices chacune d'elle indépendante. En définitive, il y a 32 matrices de cinq points sur huit (5X8) disposés sur deux lignes qui composent l'afficheur représenté ci-dessous.

Chapitre III : l'afficheur LCD

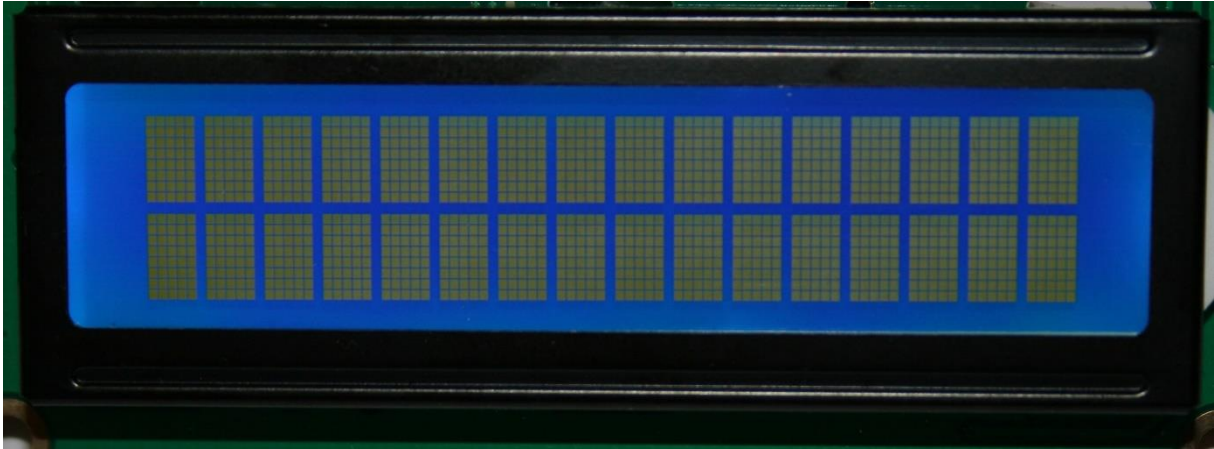


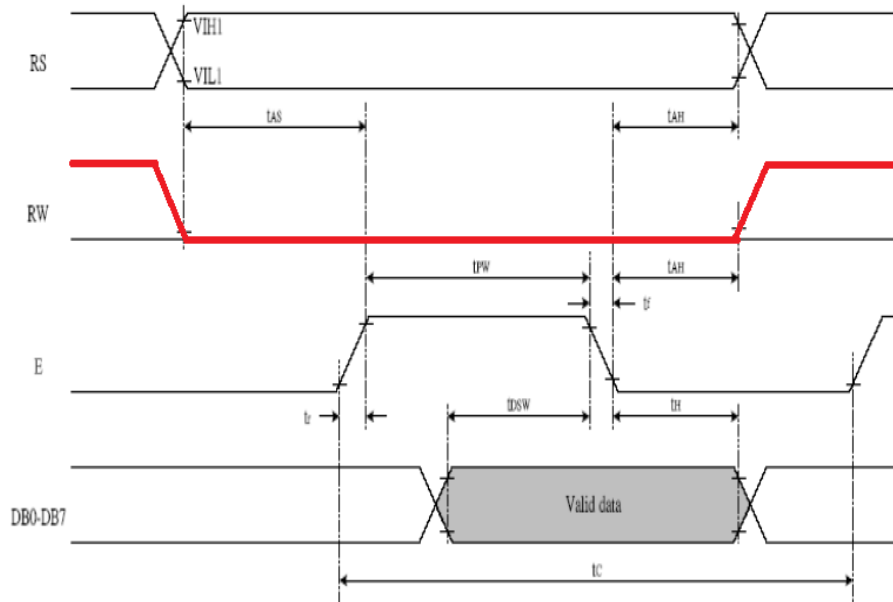
Fig 26 :Affichage des matrices 5X8

En fait, les pilotes reçoivent les données, les interprètent et affichent les caractères voulus aux positions souhaitées. Mais pour cela, il faut leur transmettre des données.

D'ailleurs, les deux logigrammes de transmission de données ci-dessous. L'un Writing data pour la réception de données et l'autre Reading data pour la transmission de données. Cependant, la seule chose qui diffère entre ces deux diagrammes, c'est l'état à la broche "RW" en rouge.

Chapitre III : l'afficheur LCD

- Writing data from MPU to ST7066U



- Reading data from ST7066U to MPU

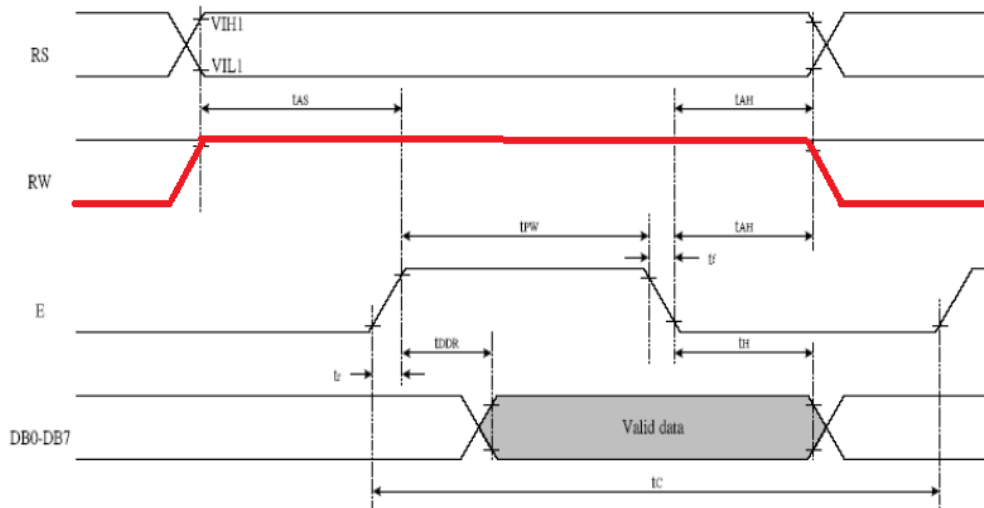


Fig 27 : Diagramme de réception de données et Diagramme d'émission de données.

Ainsi, on comprend qu'en fonction de l'état "Haut" ou "Bas" à la broche "RW", le pilote de la matrice ST7066U est en mode de réception ou émission de données.

Chapitre IV : la simulation

1. schéma synoptique :

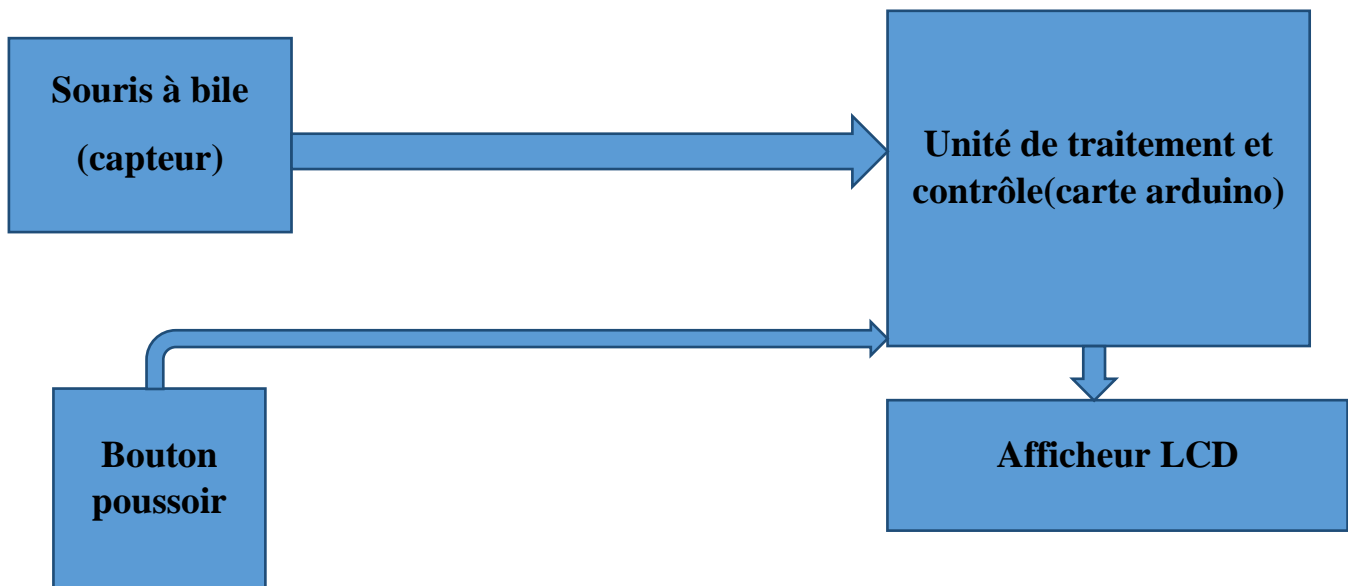


Fig28 : Schéma Synoptique

On a :

-l'unité de traitement et de contrôle des données qui est la carte arduino

- Un afficheur LCD pour afficher les données.
- 1 Bouton poussoir pour démarrer la mesure et l'arrêter.
- Une souris a bille pour faire le contact direct avec la distance à mesurer (partie capteur du circuit).

2. le logiciel de simulation proteus:

Le simulateur ISIS de Proteus utilisé dans ce projet est un logiciel puissant, développé par la compagnie électronique Labcenter, qui s'est positionné pour plus de 10 ans comme l'un des outils les plus utiles pour simuler les microcontrôleurs .

L'ISIS permet la simulation des familles des PICmicro les plus populaires tels que: 12F, 16F, 18F. En plus des PIC, ISIS peut simuler une variété de dispositifs numériques et analogiques, y compris les afficheurs sept segments, les LCD des caractères et graphiques. ISIS permet de simuler des capteurs de température, l'humidité, la pression et la lumière, entre autres.

Chapitre IV : la simulation

Le simulateur peut simuler les actionneurs tels que des moteurs CC, les servomoteurs, les lampes à incandescence, entre autres. Il est possible de simuler des périphériques d'entrée et de sortie comme les claviers, ordinateur et les ports physiques tels que RS232 et USB. Ce simulateur a une grande variété d'instruments de mesure tels que voltmètres, ampèremètres, oscilloscopes et analyseurs de signaux.

ces caractéristiques et d'autres font d'ISIS Proteus, un outil idéal pour la conception et l'étude des PICmicro.

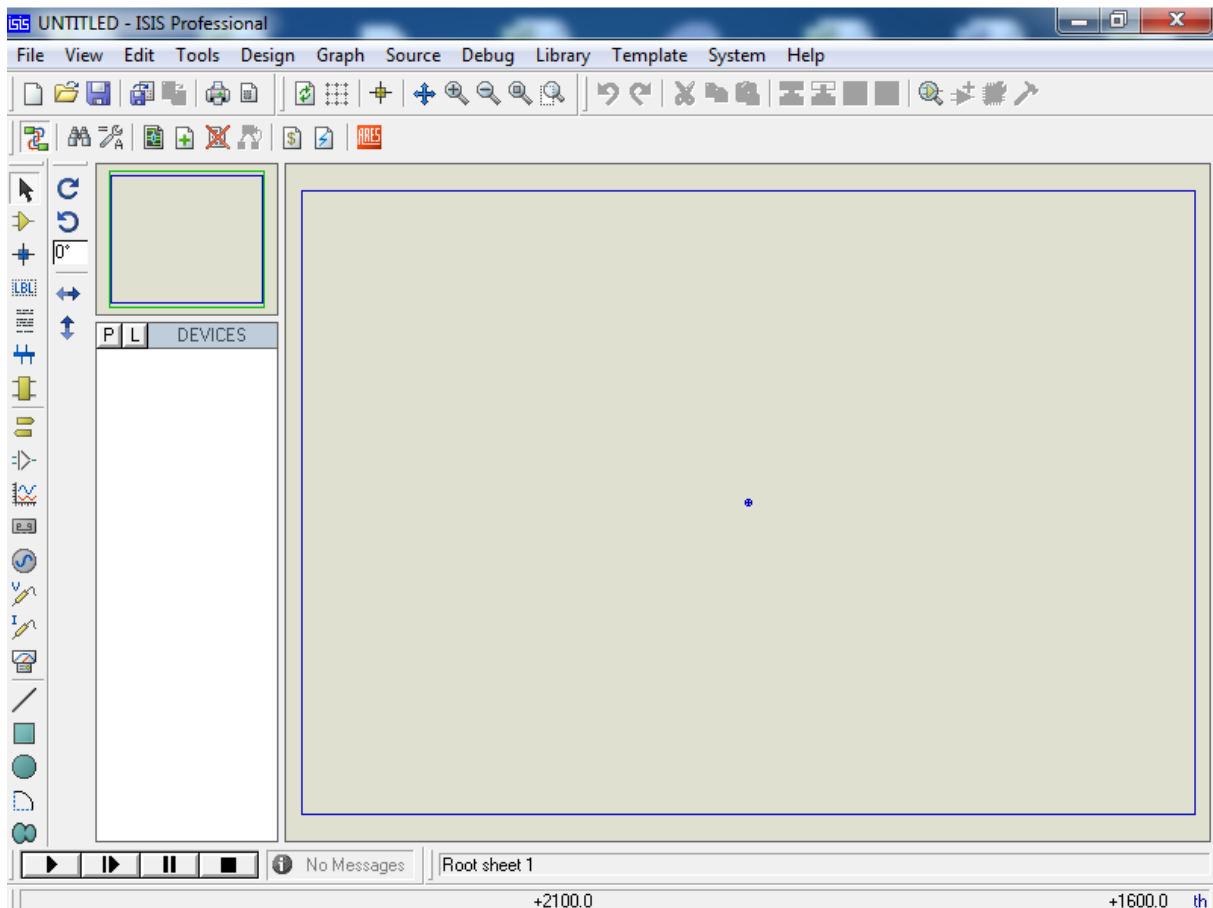


Fig 29:Interface Isis Proteus

Chapitre IV : la simulation

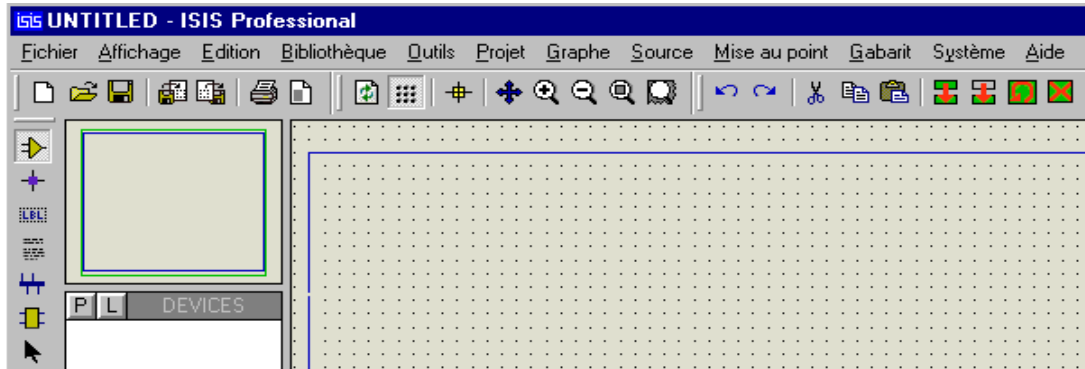


Fig 30 : Interface Isis

3. Schéma Electrique :

Dans le schéma les signaux carrés servent respectivement la fonction horloge et la donnée de la souris ps2.

La carte arduino traite les deux signaux (horloge , donnée) pour mesurer la distance de l'objet sur lequel glisse la bille de la souris Suivant le protocole de communication de la souris implémenté dans le programme informatique(voir page) .

Chapitre IV : la simulation

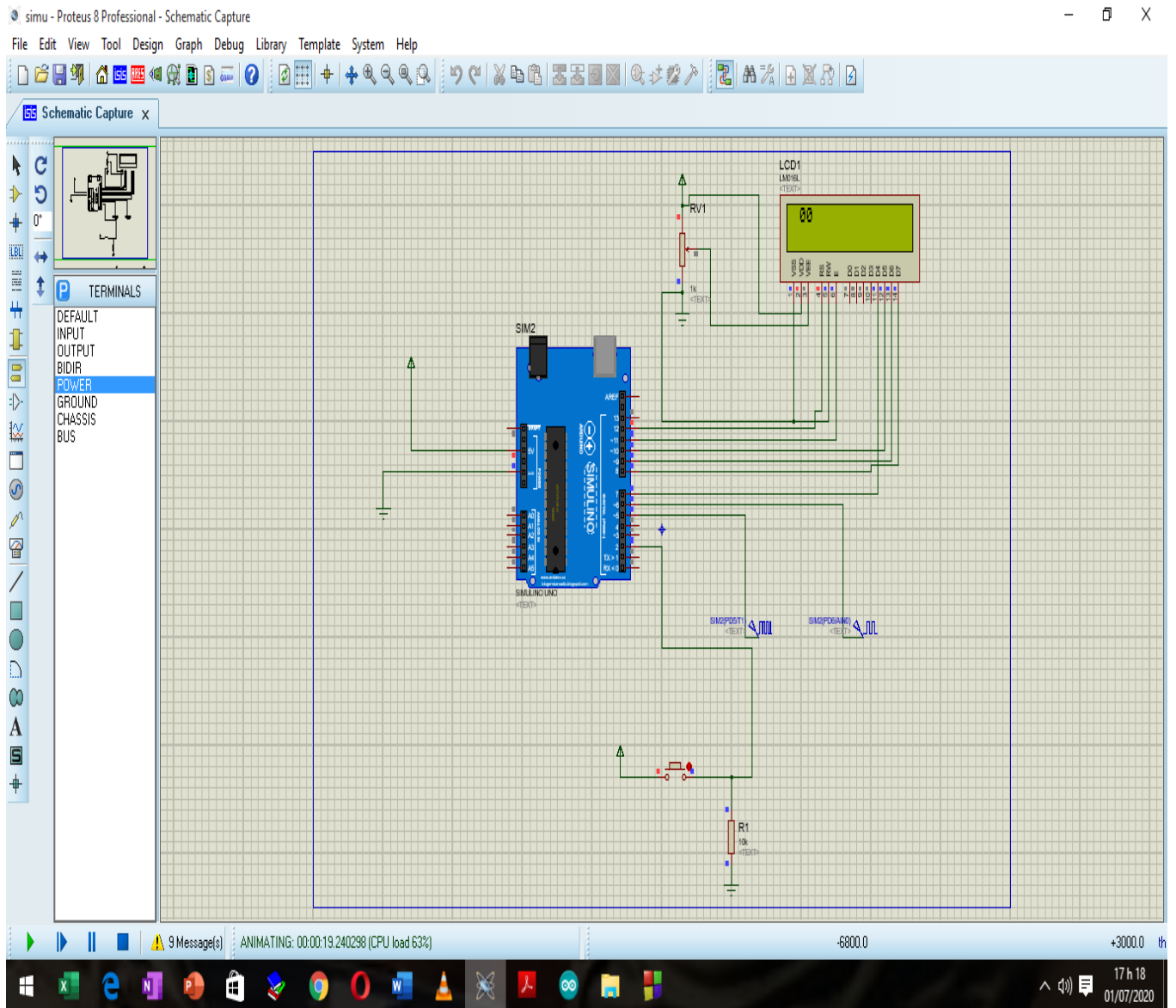


Fig31 : Le circuit

Chapitre IV : la simulation

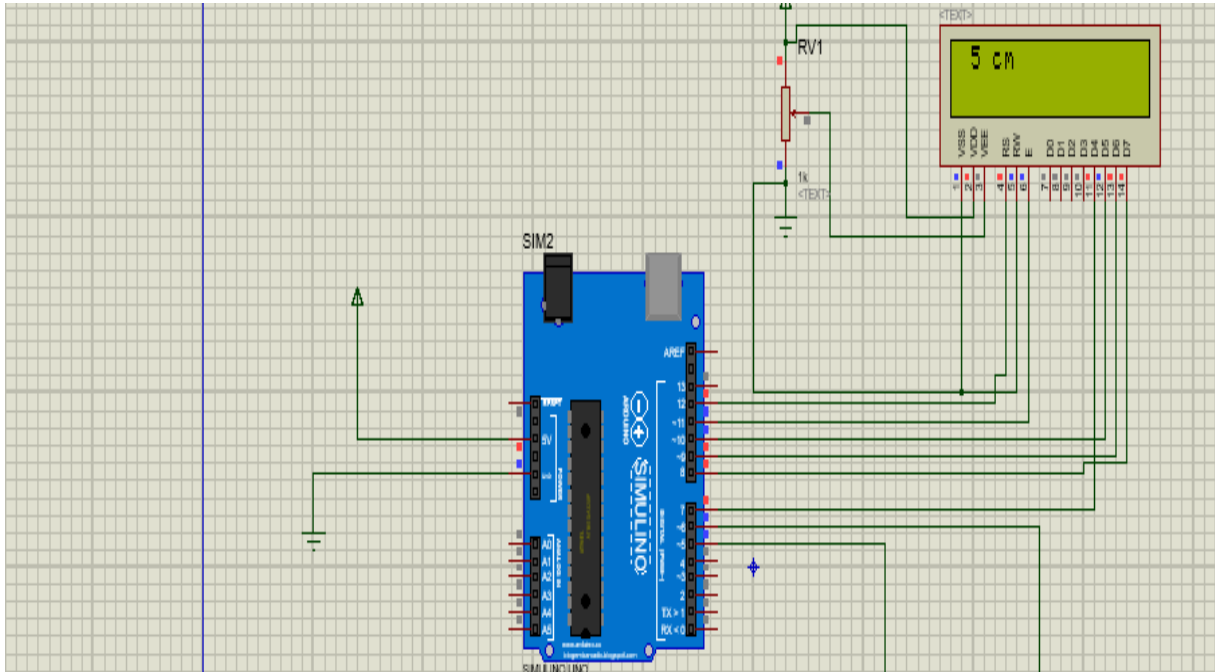


Fig 32 : la Simulation du Circuit

Chapitre IV : la simulation

Le logiciel Arduino

Le logiciel est un environnement de développement intégré (IDE) écrit en langage Java, totalement gratuit et facilement téléchargeable sur le site officiel de l'Arduino à savoir www.arduino.cc. Le logiciel étant multiplateforme, télécharger la version correspondant à votre système d'exploitation.

Une fois, le logiciel et ses pilotes, correctement installés, on peut donc programmer nos cartes Arduino. L'interface utilisateur du logiciel est présentée ci-dessous :

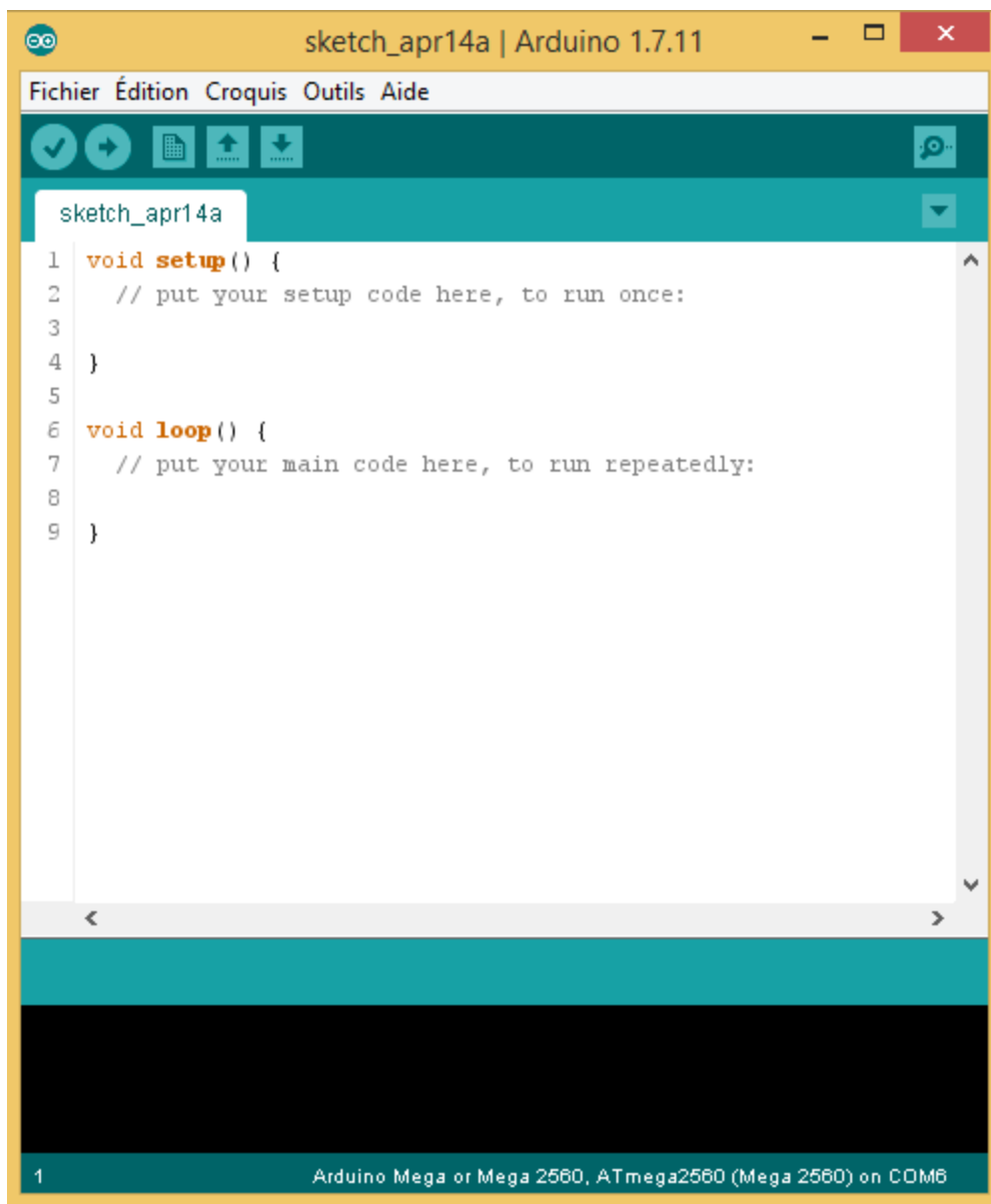


Figure 33 : L'interface utilisateur du logiciel Arduino

Chapitre IV : la simulation

On peut voir deux fonctions qui se présentent automatiquement dès qu'on ouvre le logiciel : il s'agit de la fonction *setup* () et de la fonction *loop* (). En effet, tout programme Arduino, appelé sketch, doit contenir obligatoirement ces deux fonctions spécifiques. Même si, elles sont vides, le programme ne marche pas si au moins l'une d'entre elles est absente.

La fonction *setup* () n'est exécutée qu'une seule fois juste après le lancement du programme. Elle contient généralement les instructions d'initialisation de certaines ressources de la carte telles que, par exemple, définition des lignes des ports parallèles soit en entrées ou en sorties, définition de la vitesse de fonctionnement du port série, etc...

Quant à la fonction *loop*(), elle est la boucle principale et se répète indéfiniment tant que l'Arduino restera sous tension. En d'autres termes, suite à un reset au moyen de son poussoir ou suite à une mise sous tension qui a pour effet de provoquer un reset automatique, l'Arduino exécute une seule fois les instructions contenues dans la fonction *setup* () puis exécute ensuite indéfiniment les instructions contenues dans la fonction *loop* () de ce même programme. Ces fonctions ne retournent pas de résultat, elles sont donc déclarées avec le mot clé « *void* ».

Selon le cas, une troisième partie peut ou non être présente dans un programme Arduino mais ne contient pas d'instructions exécutables. Il s'agit de la zone de définition de constantes au moyen du mot clé « *define* » ou de « *const* », d'inclusion d'éventuelles bibliothèques utilisées par le programme au moyen du mot clé « *include* », définition de variables, etc... Cette partie, si elle est présente, se place avant la fonction *setup* (), qui elle-même se place avant la fonction *loop* ().

Tout comme le langage C, le langage arduino est composé de fonctions arithmétiques et mathématiques, des opérateurs de comparaison et opérateurs logiques, les structures de contrôle (prise de décision, boucles et sauts), des fonctions de gestion du temps et des entrées/sorties et beaucoup d'autres fonctions spécifiques à l'Arduino (gestion du port série, gestion d'interruption, générateur de nombre aléatoire, jouer des notes de musique, etc...).

Chapitre IV : la simulation

❖ Programme :

```
#define MDATA 5
#define MCLK 6

void gohi(int pin)
{
    pinMode(pin, INPUT);
    digitalWrite(pin, HIGH);
}

void golo(int pin)
{
    pinMode(pin, OUTPUT);
    digitalWrite(pin, LOW);
}

void mouse_write(char data)
{
    char i;
    char parity = 1;

    // Serial.print("Sending ");
    // Serial.print(data, HEX);
    // Serial.print(" to mouse\n");
    // Serial.print("RTS");

    gohi(MDATA);
    gohi(MCLK);
    delayMicroseconds(300);
    golo(MCLK);
    delayMicroseconds(300);
    golo(MDATA);
    delayMicroseconds(10);
    /* bit de start */
    gohi(MCLK);
    /* attendre our que la souris prend le control; */
    while (digitalRead(MCLK) == HIGH)
        ;

    for (i=0; i < 8; i++) {
        if (data & 0x01) {
            gohi(MDATA);
        }
    }
}
```

Chapitre IV : la simulation

```
else {
golo(MDATA);
}
/* attendre pour le cycle de la souris*/
while (digitalRead(MCLK) == LOW)
;
while (digitalRead(MCLK) == HIGH)
;
parity = parity ^ (data & 0x01);
data = data >> 1;
}
/* parité */
if (parity) {
gohi(MDATA);
}
else {
golo(MDATA);
}
while (digitalRead(MCLK) == LOW)
;
while (digitalRead(MCLK) == HIGH)
;
/* bit de stop */
gohi(MDATA);
delayMicroseconds(50);
while (digitalRead(MCLK) == HIGH)
;
/* wait for mouse to switch modes */
while ((digitalRead(MCLK) == LOW) || (digitalRead(MDATA) ==
LOW))
;

golo(MCLK);
// Serial.print("done.\n");
}

/*
* obtenir le byte de donnée de la souris
*/
charmouse_read(void)
{
char data = 0x00;
int i;
char bit = 0x01;

// Serial.print("reading byte from mouse\n");
/* start the clock */
```

Chapitre IV : la simulation

```
gohi(MCLK);
gohi(MDATA);
delayMicroseconds(50);
while (digitalRead(MCLK) == HIGH)
;
delayMicroseconds(5);
while (digitalRead(MCLK) == LOW)
;
for (i=0; i < 8; i++) {
while (digitalRead(MCLK) == HIGH)
;
if (digitalRead(MDATA) == HIGH) {
data = data | bit;
}
while (digitalRead(MCLK) == LOW)
;
bit = bit << 1;
}

while (digitalRead(MCLK) == HIGH)
;
while (digitalRead(MCLK) == LOW)
;
/* bit de stop */
while (digitalRead(MCLK) == HIGH)
;
while (digitalRead(MCLK) == LOW)
;

golo(MCLK);
// Serial.print("Recvd data ");
// Serial.print(data, HEX);
// Serial.print(" from mouse\n");
return data;
}

voidmouse_init()
{
gohi(MCLK);
gohi(MDATA);
// Serial.print("Sending reset to mouse\n");
mouse_write(0xff);
mouse_read(); /* ack byte */
// Serial.print("Read ack byte1\n");
mouse_read(); /* blank */
mouse_read(); /* blank */
```

Chapitre IV : la simulation

```
    // Serial.print("Sending remote mode code\n");
    mouse_write(0xf0); /* remote mode */
    mouse_read(); /* ack */
    // Serial.print("Read ack byte2\n");
    delayMicroseconds(100);
}
constint Button = 2;
constint rs = 12, en = 11, d4 = 7, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
    // put your setup code here, to run once:

    pinMode(Button, INPUT);
    Serial.begin(9600);
    mouse_init();

    lcd.begin(16, 2);

}

void loop() {
    // put your main code here, to run repeatedly:
    int etatbutton = digitalRead(Button);
    char mstat;
    char mx;
    char my;
    if (etatbutton == HIGH) {
        /* obtenir la donnée de la souris */
        mouse_write(0xeb); /* donner data ! */
        mouse_read(); /* ignore ack */
        mstat = mouse_read();
        mx = mouse_read();
        my = mouse_read();

        lcd.print(mx, "cm");

    }
    else { }

}
```

Conclusion générale :

Ce projet nous montre qu'on peut donner une autre utilité à une souris à bile et nous offre des informations utiles sur comment l'élaborer.

J'ai acquis en travaillant sur ce projet une certaine connaissance en électronique notamment sur la souris à bile :

- Le principe de fonctionnement.
- Le protocole de communication
- Le circuit interne.
- Comment le microcontrôleur traite ses données ?

Le codage du code très enrichissant, peut aider à apprendre le codage d'autre protocole de communication.

J'aurais voulu ajouter d'autres fonctionnalités, mais le contexte et l'inaccessibilité de certains composants électroniques rendent la tâche difficile.

2. Annexes :

❖ Composants utilisés pour la simulation :

- Bouton poussoir.
- Résistance R1 (1 kilo ohm).
- Carte arduino uno.
- Afficheur LCD.

3. Table des Figures

| | |
|---|----|
| Figure 1 : Connecteur SOURIS PS/2 (mâle, vue en bout de cordon) | 1 |
| Figure 2 : Protocole de communication | 2 |
| Figure 3 : Trame | 2 |
| Figure 4: Symbole de la LED..... | 4 |
| Figure 5: fonctionnement de la LED..... | 4 |
| Figure 6 : spectre d'émission de la LED | 5 |
| Figure 7 : diagramme de rayonnement..... | 7 |
| Figure 8: polarisation de la LED | 8 |
| Figure 9 : Caractéristique $I_F - V_F$ | 8 |
| Figure 10.a : symbole d'une photodiode | 9 |
| Figure 10.b : Phototransistor..... | 9 |
| Figure 11 : structure d'une photodiode | 11 |
| Figure 22 : émetteur-récepteur | 11 |
| Fig13 :contexte d' utilisation d'une carte arduinouno | 14 |
| Figure 14a : les cartes arduino..... | 16 |
| Fig14b :Les cartes d'extension arduino | 17 |
| Fig14bc:shieldsarduino(cartes d'extension) | 18 |
| Figure 15 : schéma simplifié (arduino) | 19 |
| Fig16 : ATmega328 | 20 |
| fig 17 : SREG | 23 |

| | |
|--|-----------|
| Fig18 :afficheur LCD(16,2)..... | 27 |
| Fig19a : principe de fonctionnement | 28 |
| Fig19b : principe de fonctionnement | 29 |
| Fig19c : principe de fonctionnement | 30 |
| Fig20 : caractéristiques..... | 31 |
| Fig21 :Tableau des tensions d'exploitations, caractéristiques électriques et mécaniques..... | 32 |
| Fig22 :Tableau descriptif du brochage | 33 |
| Fig23 : Schéma de câblage..... | 34 |
| Fig 24:Schéma de transmission des données. | 34 |
| Fig25 : Localisation des deux puces | 35 |
| Fig 26 :Affichage des matrices 5X8..... | 36 |
| Fig 27 :Diagramme de réception de donné et Diagramme d'émission de données..... | 37 |
| | 37 |
| Fig28 : schéma synoptique | 38 |
| Fig 29: interface isisproteus | 39 |
| Fig 30 : interface isis | 40 |
| Fig31 :le circuit | 41 |
| Fig 32 : la simulation du circuit | 42 |
| Figure 33 : L'interface utilisateur du logiciel Arduino | 43 |

Bibliographie :

[1]http://www.rennes.supelec.fr/ren/perso/jweiss/docs/mouse_ps2/souris_ps2.php/#:~:text=2%20SOURIS%20PS%2F2&text=La%20souris%20semble%20pouvoir%20fournir,montre%20la%20figure%20ci%2Dapr%C3%A8s.

[2]<https://plaisirarduino.fr/afficheur-lcd-comment-lexploiter/>

[3]<https://www.arduino-france.com/review/arduino-uno/>

[4][L'afficheur%20LCD.%20Qu'est%20ce%20que%20c'est%20_%20%20PlaisirArduino.html](#)