الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministre de l'Enseignement Supérieure et de la Recherche Scientifique
جامعة 08 ماي 1945 قالمة
Université 8 Mai 1945 Guelma
Faculté des Sciences Et De La Technologie

Mémoire

Master Académique

Domaine : Science Et Technologie
Filière : Electronique
Spécialité/Option : Instrumentation
Département : D'électronique Et Télécommunication

Thème :

Testeur des circuits logiques à base d'Arduino

Présentée Par

Benyoub Tahar

Madi Azzeddine

Sous La Direction De

Dr. MENASRIA AZZEDDINE

Octobre 2020

Remerciements

Nous profitons de cette occasion pour remercier :

*Avant tout, **ALLAH**, le tout puissant, de nous avoir donné le courage et la volonté pour accomplir ce travail de recherche.*

Tous ceux qui ont de près ou de loin ont participés à notre éducation.

Nos parents, nos femmes, nos enfants et amis pour leur soutien indéfectible.

L'ensemble du personnel, Enseignants et Administratif de l'université de Guelma.

Un remerciement et une reconnaissance particulière à notre encadreur Mr MENASRIA AZZEDDINE pour ses conseils avérés, ses efforts et son écoute particulière.

Dédicaces

C'est tout plein de joie que nous dédions ce travail à ceux qui nous ont été une source d'inspiration et de volonté.

A nos parents, nos chères femmes, nos chères enfants, nos chères sœurs et nos chers frères.

A toute nos grandes familles Madi et Benyoub.

Une dédicace à tous nos amis et collègues de travail, et pour tous ceux qui nous connaissent.

ملخص:

في أطروحة التخرج قمنا بالتحقق من إمكانية إنشاء جهاز دقيق يعمل مع لوحة أرد وينو للكشف عن حالة الدوائر، المتكاملة، هل هي بحالة جيدة أم بها عيب؟ تم تقسيم العمل بأكمله إلى قسمين: جزء خصصناه للدراسة النظرية، حيث ذكرنا بعض هذه الأجهزة المتوفرة في السوق ووضحنا كيف أن لكل مصنع طريقته الخاصة في استخدام جهازه، وكذلك جميع العناصر الإلكترونية التي يمكن التحقق منها، بما في ذلك هذه الدوائر المتكاملة وفي نفس الجزء درسنا جميع العناصر التي ينطوي عليها التصميم الإلكتروني للجهاز المراد إنشاؤه، سواء من حيث سبب الاختيار أو الخصائص التكنولوجية.

أما الجزء الثاني فقد خصصناه للجانب التطبيقي واعتمدنا فقط على المحاكاة. كانت النتائج التي تم الحصول عليها من المحاكاة رائعة للغاية من حيث السرعة ومقدار الاكتشاف، فضلاً عن أهمية توفير الكثير من الوقت. من الناحية النظرية، يمكن القول إننا ربحنا جهازا فعالا، خفيفا وسريعا ويتكون من عدد قليل من المكونات الإلكترونية التي يسهل استخدامها في جميع الأوقات.

Résumé :

Dans notre mémoire de fin d'étude, nous avons étudié la possibilité de créer un appareil précis qui fonctionne avec la carte Arduino pour détecter l'état des circuits intégrés, sont-ils en bon état ou présentent-ils un défaut ?

Le travail dans son intégralité a été divisé en deux parties :

Une partie que nous avons consacrée à l'étude théorique, où nous avons évoqué certains de ces appareils disponibles sur le marché et montré comment chaque fabricant a sa propre façon d'utiliser son appareil, ainsi que l'ensemble des éléments électroniques qui peuvent être vérifiés, y compris ces circuits intégrés. Et dans la même partie, nous avons étudié tous les éléments impliqués dans la conception électronique de l'appareil à créer, tant au niveau de la raison du choix que des caractéristiques technologiques.

Quant à la deuxième partie, nous l'avons consacrée au côté applicatif, nous nous sommes appuyés uniquement sur la simulation. Les résultats obtenus grâce à la simulation étaient très impressionnants en termes de vitesse et de quantité de détection, ainsi que l'important que nous ayons gagné beaucoup de temps.

En théorie, on peut dire que nous avons gagné un appareil efficace, léger et rapide et composé de quelques composants électroniques faciles à utiliser à tout moment.

Abstract:

In our graduation thesis, we investigated the possibility of creating a precise device that works with the Arduino board to detect the condition of integrated circuits, are they in good condition or have a defect?

The entire work was divided into two parts:

A part that we have devoted to the theoretical study, where we have mentioned some of these devices available on the market and shown how each manufacturer has his own way of using his device, as well as all the electronic elements that can be verified, including these integrated circuits. And in the same part, we studied all the elements involved in the electronic design of the device to be created, both in terms of the reason for the choice and the technological characteristics.

As for the second part, we devoted it to the application side, we relied solely on simulation. The results we got from the simulation were very impressive in terms of speed and amount of detection, as well as the importance that we saved a lot of time.

In theory, it can be said that we have won an efficient device that is light and fast and consists of a few electronic components that are easy to use at all times.

LISTE DES FIGURES

LISTE DES TABLEAUX

ABREVIATION :

CIs : Circuits Intégrés.

CMOS : Complementary Metal Oxide Semi-conducator.

CNC : Commande Numérique par Calculateur.

FGT: Flottante Gate Transistor.

Kg: kilogramme.

LED: Light Emitting Diode.

PCB: Printed Circuit Board.

TTL: Transistor Transistor Logic.

SCL: Signal Clock.

SDA: Signal Data.

SSD: Solid-State Drive.

VSM: Virtual System Modelling.

ZIF : Zéro Insertion Force.

TABLE DES MATIERES

Introduction générale :

De nos jours l'utilisation d'appareils électroniques dans notre vie quotidienne est devenue un impératif, lavage, cuisine, communication ...etc.

L'utilisation excessive de ces appareils les a rendus susceptibles d'être endommagés à tout moment ce qui doit être interféré dès que possible pour les réparer, mais il y a des obstacles qui font perdre du temps à l'intervention jusqu'à découvrir la défaillance de ces appareils car les systèmes électroniques sont quelque peu compliqués.

Pour participer à la détection de ces problèmes de défaillance ou bien le mal fonctionnement de ces appareils, nous avons vu que le composant important présent toujours dans l'installation de circuits électroniques est le circuit intégré. A la base de cette vue nous allons répondre à la question :

Comment examiner le circuit intégré dans la période la plus courte, car c'est l'élément principal et important qui comprend la plupart des appareils électroniques et des industries.

Il existe de nombreux appareils présents sur le marché qui répondent à cet objectif, mais les opinions de leurs utilisateurs sont souvent affectées par des crampes lors de l'examen des circuits intégrés, ce qui a suscité notre désir de rechercher ce sujet pour trouver des solutions ou au moins découvrir le meilleur.

Parmi les solutions que nous avons trouvées ou découvertes est ; la réalisation d'un examinateur qui dépend de la carte électronique (Arduino), car cette dernière est devenue une solution urgente pour contrôler de nombreux appareils, une facilité d'utilisation et inclusion dans des solutions simples.

L'achèvement de ces projets simples (testeurs des CIs logiques) et la compréhension de leur fonctionnement nous évitent d'avoir à chercher sur le marché des appareils qui répondent à nos demandes, ce que nous trouvons rarement en raison de leurs utilisations, nous ne trouvons donc, par exemple, que des étudiants en laboratoires et des revendeurs de composants électroniques qui les utilisent pour vérifier l'état des circuits intégrés.

Le saut qualitatif créé par la carte électronique Arduino, qui se caractérise par sa large utilisation, son abondance, et la facilité de la traiter en termes d'installation ou de connexion, ainsi que sa programmation, nous ont motivés à l'utiliser dans le cadre de nos recherches.

Notre mémoire de fin d'étude (GRADUATION EN MASTERE 2) est basé sur trois chapitres :

1. Chapitre I : généralités sur les testeurs des circuits logiques.
2. Chapitre II : étude d'un testeur des circuits logiques.
3. Chapitre III : réalisation d'un testeur des circuits logique.

I.1. Introduction :

Ces dernières années, le monde a connu un développement rapide dans tous Les domaines, et parmi les plus importants de ces domaines figurent les industries électroniques de toutes sortes, qui sont devenues dépendantes dans leurs industries des circuits intégrés pour leurs performances élevées et leurs petites tailles.

L'existence des circuits intègres dans tous les systèmes électroniques nécessitait des solutions rapide pour l'intervention en cas d'endommage de ces derniers. A partir de cette exigence, les inventeurs ont créés des dispositifs permettant de détecter l'endommagement de ces circuits intégrés qui sont les testeurs des circuits logiques.

La plupart des entreprises qui fabriquent ces dispositifs cherchent à être en phase avec ce développement et y excellent en fournissant constamment des produits conformes aux exigences du marché en adoptant les meilleures technologies pour fabriquer ces dispositifs de haute qualité.

Dans ce chapitre nous allons citer des certains de ces appareils existent dans le marché, illustrerons leurs caractéristiques technologiques.

I.2.Les différents types de testeurs des CIs logiques :

De point de vue les testeurs des circuits intégrés sont limités par leur capacité de nombre des composants intégrés qu'ils peuvent tester ; ce qui indique la pauvreté et la gamme étroite de test.

Chacun de ces tests est mémorisé dans le microcontrôleur pour qu'il soit réalisé par le numéro du circuit intégré à tester, donc ces tests sont indépendants pour chaque circuit.

I.2.1. Testeur des circuits intégrés modèle 570A :

Le testeur 570A est un testeur de circuit logique numérique portable qui fournit des fonctionnalités avancées et il est facile à utiliser. La bibliothèque de tests contient une large gamme de TTL et CMOS, comprend tous les circuits intégrés analogiques courants, y comprises ; amplificateurs opérationnels, les comparateurs, les régulateurs de tension, les références de tension, les commutateurs et multiplexeurs analogiques, les isolateurs et coupleurs optiques et les circuits intégrés audio. [1]Ces caractéristiques sont illustrées au-dessous. (Voir Tab.I 1)

SOURCES D'ALIMENTATION	4 PILES X 1,5
ENTREE DC	6 V, 850 MA MAX
REGLEMENTE CONSOMMATION ELECTRIQUE ETEINT	10 MA
STANDBY	30 MA
SEUILS DE TEST	MAX 2,2 V MIN 0,8 V

Tab.I. 1. Caractéristiques testeur modèle 570A

Fig. I. 1.Testeur de circuits intégrés analogiques modèle 570A [2]

I.2.2. Testeur de circuits intégrés numériques de type YBD868 :

Le testeur de circuit intégré numérique de type YBD868 prend le microprocesseur avancé comme cœur, l'appareil principal sélectionne le circuit intégré produit par des sociétés d'appareils bien connues telles que INTEL, MOTOROLA, etc., coopère avec le logiciel et le système d'extension périphérique pour simuler complètement la fonction intégrée de l'appareil testé, et convient à presque tous les circuits intégrés numériques avec une sortie fixe. Il est utilisé pour la maintenance, teste tous les types d'ordinateurs, équipements d'automatisation industrielle, équipement médical à grande échelle, machines-outils CNC, périphériques informatiques, commutateurs à programme contrôlé, instrumentation électronique, équipement de communication numérique, dispositifs de protection de relais électroniques et divers types de produits électroniques, ces caractéristiques sont illustrées au-dessous. (Voir Tab.I 2)

Le testeur **de type YBD868** peut tester plus de 2000 types de circuits intégrés, famille TTL 54XXX ,55XXX, 74 XXX et la famille CMOS 14, 40,45. [3]

TENSION D'ALIMENTATION	220 V OU 110 V 50HZ
TEMPERATURE APPLICABLE	0 ° C A + 40 ° C
LA TEMPERATURE AMBIANTE	25 ° C
DIMENSIONS	285*95*290 CM
POIDS	2.0 KG

Tab.I. 2. Caractéristiques testeur type YBD868

Fig. I. 2. Testeur Modèle YBD868 [3]

I.2.3. Testeur des circuits intégrés 74 40(TES200) :

TES200 est un testeur numérique intégré, principalement teste les séries 74 et les séries 40 des circuits numériques intégrés, peut tester 200 types d'intégration.

L'utilisation d'une intégration logique simple et de test peut également déterminer lesquelles des portes logiques sont mauvaises. Sélectionnez l'intégration de test par opération clé. [4]

Ce testeur est caractérisé comme illustre au-dessous. (Voir Tab.I 3)

TENSION DE FONCTIONNEMENT	7 ~ 12 VDC
COURANT DE TRAVAIL/.L./	MA <30
TAILLE MM X LONG	LONG 70*45*40 LARGE X HAUT
TEMPERATURE DE TRAVAIL	-40 A 65 DEGRES CELSIUS
TEMPERATURE DE STOCKAGE	-40 A 65 DEGRES CELSIUS

Tab.I.3. Caractéristiques du testeur 74 40(TES200)

Fig. I. 3. Testeur type TES200 [4]

I.2.4. Testeur type LEAPER-1A :

LEAPER-1A est un testeur portable, spécialement conçu pour les circuits intégrés.

Il a une prise **ZIF** 24 broches pour s'adapter à différents circuits intégrés numériques. Aucun **PC** n'est requis pour fonctionner LEAPER-1A, il fonctionne en mode autonome grâce à un adaptateur secteur ou à des piles.

Le testeur LEAPER-1A léger et économe en énergie, peut tester la famille TTL 54/74 xxxx et la famille CMOS 40/45/14xxx. [5]

Les caractéristiques sont citées au-dessous dans le Tab.I 4

PLAGE DE TENSION	2.5 V/3.0 V/3.3 V/5 V
VITESSE D'ESSAI ELEVEE	0.6 SECONDE
TEMPERATURE	+ 5 ° C ~ + 45 ° C
ALTITUDE DE FONCTIONNEMENT	JUSQU'A 5000 M
POIDS	312G
BROCHES D'ESSAI	14 ~ 24
AFFICHAGE	16 CARACTERES EN 1 LIGNE LCD

Tab.I. 4 Caractéristiques du testeur type LEAPER-1A

Fig. I. 4. Tester type LEAPER-1A [5]

I.2.5. Testeur des circuits intégrés GUT-6000B :

Le testeur de circuits intégrés, GUT-6000B, est un nouveau lancement et la meilleure qualité possible et multi- produit équipé de fonctions. Installation conviviale de l'utilisateur en remplaçant un autre CI ; le GUT-6000B continue à entreprendre la tâche. La conception matérielle de la fonction «lumière noire» étend la commodité de l'utilisateur pour tester les circuits intégrés dans un environnement de lumière inadéquat. La touche Buzzer intégrée dans différentes tonalités peut facilement identifier le résultat du test. La capacité unique d'identifier plus de 1800 circuits intégrés numériques CMOS / TTL (jusqu'à 24 broches) surpasse les autres principaux testeurs de circuit La fonction GUT-6000B telle que les vitesses intégrées de `` recherche et test automatiques " pour identifier et tester les circuits intégrés. La conception en «boucle» pour la fonction de test en continu est appliquée de manière intelligente pour détecter les circuits intégrés défectueux et leur stabilité. Tous ces atouts offrent des avantages significatifs pour les testeurs de circuits intégrés numériques. Tous les avantages incalculables accumulés du GUT-6000B sont à découvrir lorsque les utilisateurs utilisent le testeur à valeur ajoutée et équipé de fonctions multiples.

C’est le meilleur choix pour les usines, les départements de maintenance, des laboratoires ainsi que des universitaires. [22]

Il caractérisé comme illustrer dans le Tab.I 5.

AFFICHAGE	16*1 A MATRICE A POINT DE CARECTERE
PRISE ZIF	UNE POSITION POUR PRISE IC A 28 BROCHES
CLE OPERATIONNELLE	BUZZER, VOLTAGE, LOOP, AUTO, GO, BACK SPACE (2) 10 TOUCHE (0 –9)
ALARM	DIFFERENTES TONALITES POUR LE RESULTAT DU TEST
TENSION D'ALIMENTATION	220/110V 50/60HZ
TEMPERATURE	10° DEGRES CELSIUS A 40 DEGRES CELSIUS

Tab.I. 5. Caractéristiques testeurGUT-6000B

Fig. I. 5. Testeur de circuits intégrés GUT-6000B [23]

I.2.6. Testeur de circuits intégrés HTS OO1 :

Le testeur des circuits intégrés HTS OO1, est un testeur existe souvent dans les laboratoires d'enseignement à l'école, il est utilisé dans certaines des fonctions de base d'un grand nombre de circuits intégrées. Par exemple, séries TTL 74/54, série CMOS 4000/4500 amplificateurs, comparateurs, transistors, optocoupleurs, etc.

Comme les étudiants commencent à apprendre le circuit, l'utilisation du processus, il y aura beaucoup d'erreurs stupides occasionnelles, causants des dommages des circuits intégrés, ainsi au cours de l'expérience, afin d'exclure de tels facteurs et économisant du temps d'enseignement ; l'utilisation de testeur HTS-001 est dédiée pour la vérification. En outre, le testeur des circuits intégrés prend en charge la fonction de recherche automatique de circuit intègre pour trouver le succès affichera. La conception du test est indépendante du matériel et

du logiciel, les bibliothèques de circuits intègres disponibles en ligne, mises à jour en temps réel, facile à utiliser. Ces caractéristiques sont illustrées au-dessous. Voir Tab.I 6 [7]

PLAGE DE TEMPERATURE DE FONCTIONNEMENT	-20° C-50° C
TAILLE	170*120*30MM
LE TEMPS DE TEST A PUCE UNIQUE	INFERIEUR A 3S
PLAGE D'HUMIDITE DE FONCTIONNEMENT	30% -95%
POIDS	250G
NOMBRE DE BROCHE	40 BROCHES

Tab.I. 6. Caractéristiques testeur HTS OO1

Fig. I. 6. Testeur de circuits intégrés HTS OO1 [7]

I.2.7. Testeur numérique IC DICT-01 :

Testeur numérique IC DICT-01 Peut tester une large gamme de circuits intégrés numériques ; plus de 300 circuits intégrés tels que la série 74, la série 40/45, possède une prise ZIF à 24 broches. [25]

La fonction de recherche automatique des CIs répertoriés, et 16 touches et un écran LCD 16 x 1 ligne.

Limité par le nombre de CI qui peuvent tester et ne peut pas tester aucun CI a plus de 24 broches.

Fig. I.7. Testeur numérique IC DICT-01 [25]

I.2.8. Testeur IC universel DICT-03 :

Le testeur IC universel DICT-03 peut tester une large gamme de CIs numériques, plus de 1500+ CIs ; tels que 74 séries, 40/45 séries, 8085, 8086, Z80, 8255, 8279, 8253, 8259, 8251, 8155, 6264, 62256, 8288 et 8284. Aussi les circuits intégrés analogiques tels que des amplificateurs opérationnels, des minuteries, des matrices de transistors, des commutateurs analogiques, des optocoupleurs, ADC, DAC, régulateur de tension, etc. l'affichage à 7 segments, recherche automatique de tous les CIs numériques, possède une prise ZIF à 40 broches, clavier de 50 touches et écran LCD 16 X 2. [25]

Fig. I. 8. Testeur IC universel DICT-03 [25]

I.2.9. Testeur IC numérique 575A :

Le testeur IC numérique 575A est capable de localiser les défauts intermittents liés à la température en utilisant ses modes de test de boucle inconditionnel ou conditionnel. Le 575A recherchera sa bibliothèque et identifiera le dispositif, affichant les équivalents fonctionnels possibles pour le remplacement. Dans le cadre du test IC, le numéro IC spécifique, la description fonctionnelle de l'appareil et l'état des broches défectueuses défilent sur l'écran intégré Caractéristiques La bibliothèque complète de périphériques couvre la plupart des périphériques TTL, CMOS, de mémoire et d'interface, une capacité de 40 broches (portes NAND ou CPU), identifie les périphériques non marqués et codés maison, détecte les défauts intermittents liés à la température, affiche des informations de diagnostic pour les broches individuelles et fonctionne sur batterie.

Très coûteux, difficile à utiliser, ne peut pas détecter les circuits intégrés, limité par le nombre de circuits intégrés pouvant être testés et codés en interne (non portables). [26]

Fig. I. 9. Testeur IC numérique 575A [26]

I.2.10. Testeur Instek Gut-6600 :

Le testeur numérique de CI Instek GUT-6600 est un testeur facile à utiliser, spécialement conçu pour le CI numérique. Il est petit, portable, léger et économe en énergie, et utilisable avec des piles. Son temps de recherche moyen est rapide : 0,8 seconde. Les séries : 74/40/45/41/44S sont pris en charge pour le test. L'affichage LCD est de 16 caractères sur 1 ligne, les broches de test de 14 à 24. [27]

Fig. I. 10. Testeur Instek Gut-6600 [27]

I.3. Domaines d'utilisations des testeurs des circuits intégrés logiques :

Ce n'est un secret pour personne, la vitesse de développement en électronique va très vite. En effet, tout le monde aura remarqué ces dernières années l'inéluctable course à la miniaturisation de tout appareil électronique. Plus petit, plus de fonctionnalités, plus rapide, plus d'autonomie... Ceci est dû essentiellement à la demande croissante de mobilités des usages de plus en plus ciblés qui sont devenus aujourd'hui les enjeux forts du marché de l'électronique.

D'autres parts, l'industrie des semi-conducteurs a continué à améliorer la vitesse des circuits tout en diminuant la consommation énergétique.

Le circuit intègre est l'un des composants les plus utilisés dans le domaine d'électronique, on le trouve pratiquement dans tous les systèmes électroniques, et par conséquent il va être toujours soumis à des dangers ou bien endommager carrément ; et pour cela l'action a été une création ou bien une invention d'un testeur des circuits intègres logiques pour l'intervention dans les brèves délais.

Habituellement les laboratoires électroniques, les télévisions, les ateliers d'entretien et les établissements d'enseignement ; sont toujours besoins de tel appareil car ils confrontent quotidiennement des énormes problèmes avec les lots des circuits défectueux.

Donc un testeur des circuits intégrés logiques permet dans ce cas la vérification de l'état du circuit intégré et en suite l'intervention pour résoudre le problème.

I.4. Conclusion :

Dans ce chapitre nous avons présenté certains types de dispositifs de testeurs des circuits intégrés logiques et nous avons connu que chaque fabricant d'un tel dispositif de test des circuits intégrés logiques avait sa propre idée sur la façon de mener le test, et nous nous sommes familiarisés avec certaines des caractéristiques de ces dispositifs, dont la plus importante est la vitesse et la précision du test.

Nous avons également présenté quelques domaines d'utilisation.

II.1. Introduction :

Notre mémoire de fin d'étude est considéré comme un simple montage électronique à réaliser en raison de nombre limité de composants électroniques qu'il contient, et il n'est pas habituel car ces éléments électroniques ne sont pas simples dans leur composition, ils remplissent de multiples fonctions en fonction de leur programmation.

Dans ce chapitre, nous parlerons de ces éléments électroniques qui entrent dans la mise en œuvre du montage à réaliser en termes de caractéristiques technologiques, ainsi que du côté programmation.

II.2. Schéma bloc du testeur :

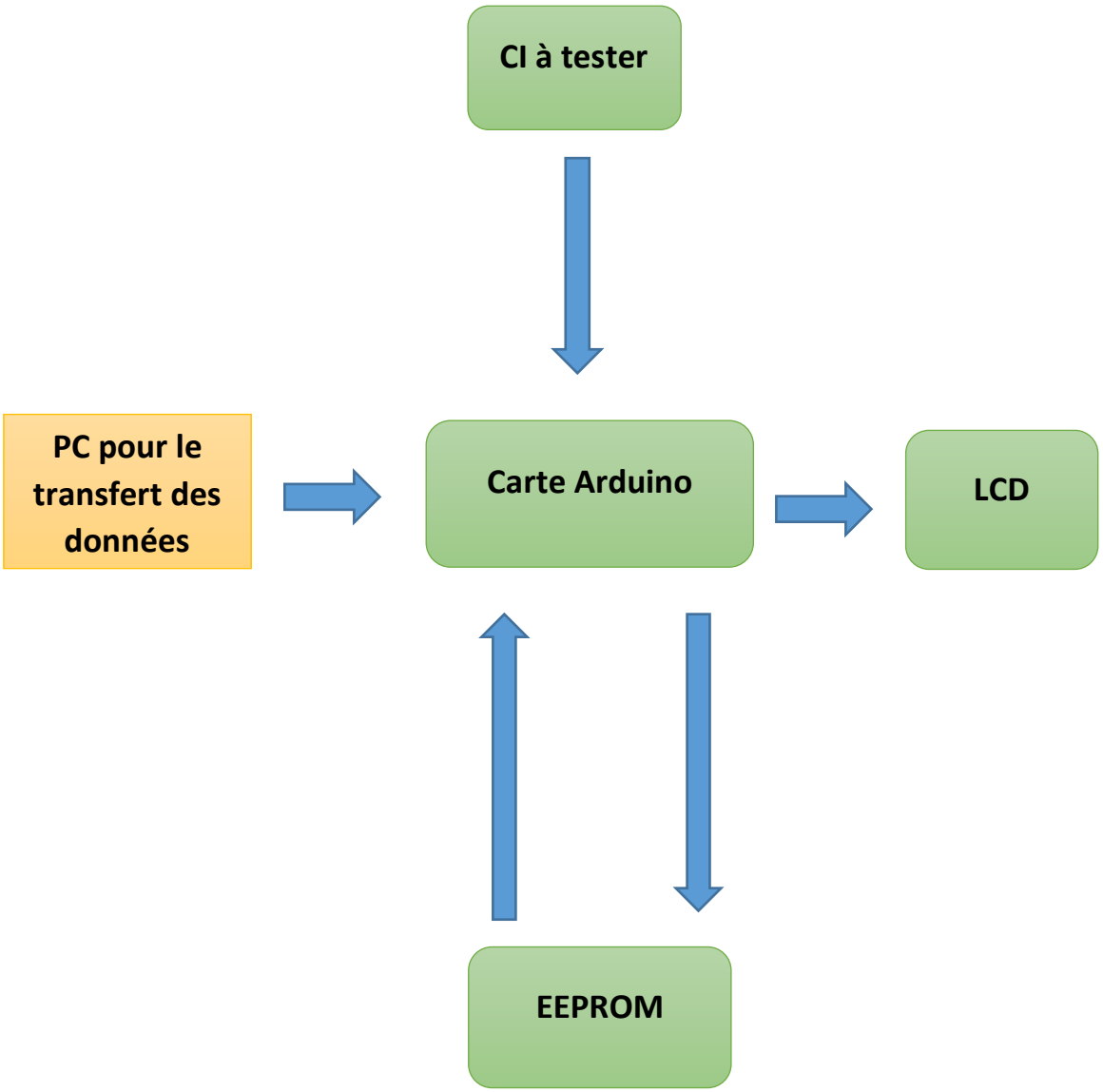

Fig. II. 1. Schéma bloc du testeur.

II.3. Arduino :

L'Arduino est une carte électronique programmable basée sur un microcontrôleur, conçu pour être accessible à tous par sa simplicité, aussi peut également être d'usage professionnel, tant les possibilités d'application sont nombreuses.

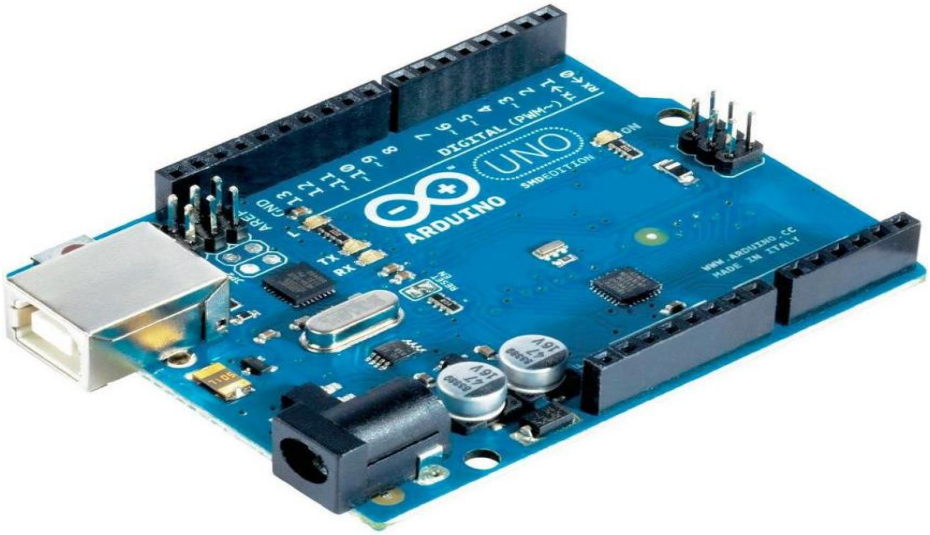

Fig. II. 2. Arduino Uno [8]

II.3.1. Historique de la carte Arduino :

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 (avant 2003 - 2004) : les outils nécessaires à la création de projets d'interactivité étaient complexes et onéreux. Ces coûts souvent trop élevés rendaient difficiles le développement par les étudiants de nombreux projets et ceci ralentissait la mise en œuvre concrète de leur apprentissage.

Jusqu'alors, les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Ils sont puissants mais leurs processus de développement sont longs et ils sont difficiles à apprendre et à utiliser pour les artistes, les designers d'interactions et, plus généralement, pour les débutants.

Leur préoccupation se concentra alors sur la réalisation d'un matériel moins cher et plus facile à utiliser. Ils souhaitaient créer un environnement proche de Processing, ce langage de programmation développé dès 2001 par Casey Reas et Ben Fry, deux anciens étudiants de John Maeda au M.I.T., lui-même initiateur du projet DBN .

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. Pour ce travail, Hernando Barragan réutilisait les sources du projet Processing. Basée sur un langage de programmation facile d'accès et adaptée aux développements de projets de designers, la carte Wiring a donc inspiré le projet Arduino (2005).

Comme pour Wiring, l'objectif était d'arriver à un dispositif simple à utiliser, dont les coûts seraient peu élevés, les codes et les plans « libres » (c'est-à-dire dont les sources sont ouvertes et peuvent être modifiées, améliorées, distribuées par les utilisateurs eux-mêmes) et, enfin, « multi-plates-formes » (indépendant du système d'exploitation utilisé).

Conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Cuartielles, Massimo Banzi ainsi que Nicholas Zambetti), l'environnement Arduino est particulièrement adapté à la production artistique ainsi qu'au développement de conceptions qui peuvent trouver leurs réalisations dans la production industrielle.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver. Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort ». [8]

II.3.2. Description de l'Arduino :

Arduino fait partie de la famille des cartes de développement ; il ne possède pas d'**OS**. Il reste par contre l'un des plus abordables et des plus répandus. Une carte de développement est en général un circuit imprimé équipé d'un microcontrôleur. Comme Arduino est open source, il existe un grand nombre de clones et de platines compatibles, tout comme il existe de nombreux modèles d'Arduino officiels, avec des fonctions particulières: **(voir Fig.12)** [9]

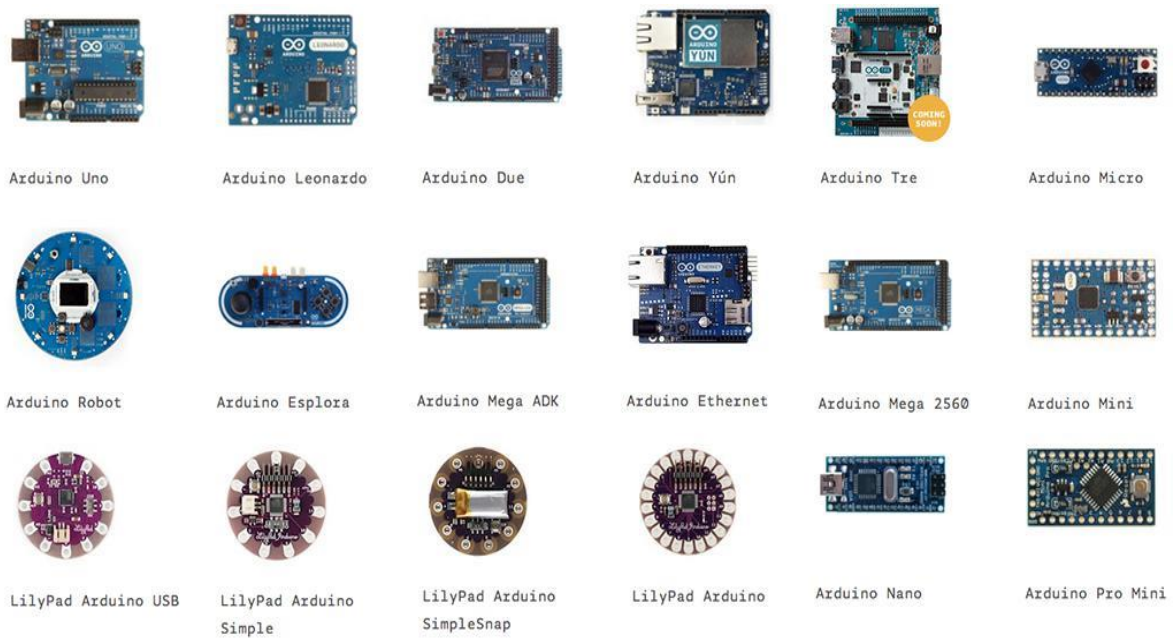

Fig. II. 3. Types d'Arduino [9]

II.3.3. Le microcontrôleur :

C'est l'unité centrale de la carte Arduino. Il communique à l'aide d'un programme crée et stocké dans sa mémoire avant de l'exécuter. Grâce à ce programme, il savoir faire des choses, qui peuvent être :

faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur, mettre en route ou arrêter un moteur… [9]

Ce large éventail d'applications a donné naissance à différents types classées en familles selon les fabricants qu'appartiennent.

- **Familles de microcontrôleurs :** [18]

- la famille Atmel AT91 .
- les familles ARM Cortex-M et ARM Cortex-R.
- la famille Atmel AVR (utilisée par des cartes Wiring et Arduino).
- le C167 de Siemens/Infineon .
- la famille des Infineon AURIX TC3x, Infineon AURIX TC2x, Infineon TriCore TC1x, Infineon XMC, XC2000 de Infineon Technologies .
- la famille Hitachi H8 .
- la famille Intel 8051, qui ne cesse de grandir ; de plus, certains processeurs récents utilisent un cœur 8051, qui est complété par divers périphériques (ports d'E/S, compteurs/temporisateurs, convertisseurs A/N et N/A, chien de garde, superviseur de tension, etc.) .
- l'Intel 8085, à l'origine conçu pour être un microprocesseur, a en pratique souvent été utilisé en tant que microcontrôleur.
- le Freescale 68HC11 .
- la famille Freescale 68HC08.
- la famille Freescale 68HC12.
- la famille Freescale Qorivva MPC5XXX .
- la famille des PIC de Microchip .
- la famille des dsPIC de Microchip .
- la famille ADuC d'Analog Devices .
- la famille PICBASIC de Comfile Technology.
- la famille MSP430 de Texas Instruments .
- la famille 8080, dont les héritiers sont le microprocesseur Zilog Z80 (désormais utilisé en tant que contrôleur dans l'embarqué) et le microcontrôleur Rabbit .
- la famille PSoC de Cypress Semiconductor .
- la famille LPC21xx ARM7-TDMI de Philips .
- la famille V800 de NEC .
- la famille K0 de NEC.
- la famille des ST6, ST7, ST10, STR7, STR9, de STMicroelectronics .
- la famille STM32 de STMicroelectronics.
- la famille STM8 de STMicroelectronics .

C'est la famille **Atmel AVR** d'où :

Atmel : est un fabricant connu dans le monde, spécialisé dans la fabrication des composants a semi-conducteur.

AVR : un terme utilisé par Atmel indiquant le cœur du microprocesseur et la famille de microcontrôleurs qui le mette en œuvre.

Qui nous intéresse dans la réalisation du testeur désiré, car la carte Arduino utilisée est basée sur le ATmega640 / 1280/1281 / **2560**/2561 qu'est un CMOS à faible puissance microcontrôleur 8 bits.

En exécutant des instructions puissantes dans un seul cycle d'horloge, atteint des débits approchant 1 MIPS par MHz permettant au concepteur du système pour optimiser la consommation d'énergie par rapport à la vitesse de traitement.

II.3.4. L'alimentation :

Pour fonctionner, une carte Arduino a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée directement en 5V par le port USB ou bien par une alimentation externe qui est comprise entre 7V et 12V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. [9]

II.3.5. Les connections des cartes d'extension :

A part une LED sur la broche 13, la carte Arduino ne possède pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme. Il est nécessaire de les rajouter. Mais pour cela, il faut les connecter à la carte. C'est là qu'interviennent les connecteurs, aussi appelés **broches** (*pins*, en anglais). [9]

Sur les Arduino et sur beaucoup de cartes compatibles Arduino, les broches se trouvent au même endroit.

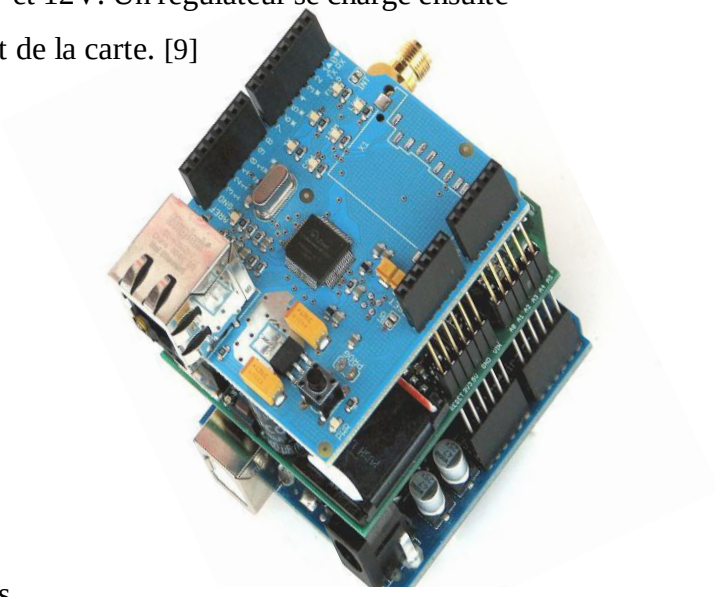

Fig. II. 4. Les connections des cartes d'extension [9]

Cela permet de fixer des cartes d'extension, appelée shields en les empilant.

II.3.6. Exploration des broches Arduino :

- 0 à 13 Entrées/sorties numériques.
- A0 à A8 Entrées/sorties analogiques.
- 5V Alimentation +5V.
- 3.3V Alimentation +3.3V.
- Vin Alimentation non stabilisée (= le même voltage que celui à l'entrée de la carte.

Les connexions entre les composants sont réalisées par des *jumpers*, sortes de petits câbles.

Fig. II. 5. Les Jumpers [9]

II.3.7. Le logiciel Arduino IDE :

Le langage de programmation d'Arduino est basé sur le langage C/C++, qui s'écrit sur le logiciel Arduino **IDE**. C'est grâce à ce logiciel que nous allons créer, tester et envoyer les programmes sur l'Arduino.

L'IDE est téléchargeable à l'adresse suivante: http://Arduino.cc.

- **Le déroulement du programme :**

Le programme se déroule de la façon suivante :

1. Prise en compte des instructions de la partie déclarative.
2. Exécution de la partie configuration (fonction *setup( )*).
3. Exécution de la boucle sans fin (fonction *loop ( )*) : le code compris dans la boucle sans fin est exécuté indéfiniment.

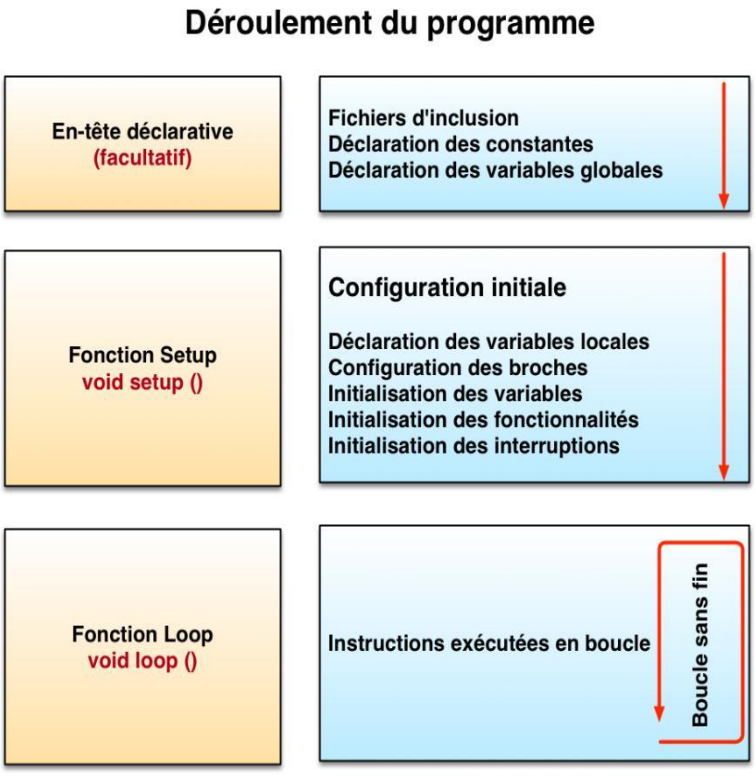

Fig. II. 6. Etapes de déroulement du programme [9]

II.4. La carte Arduino NANO :

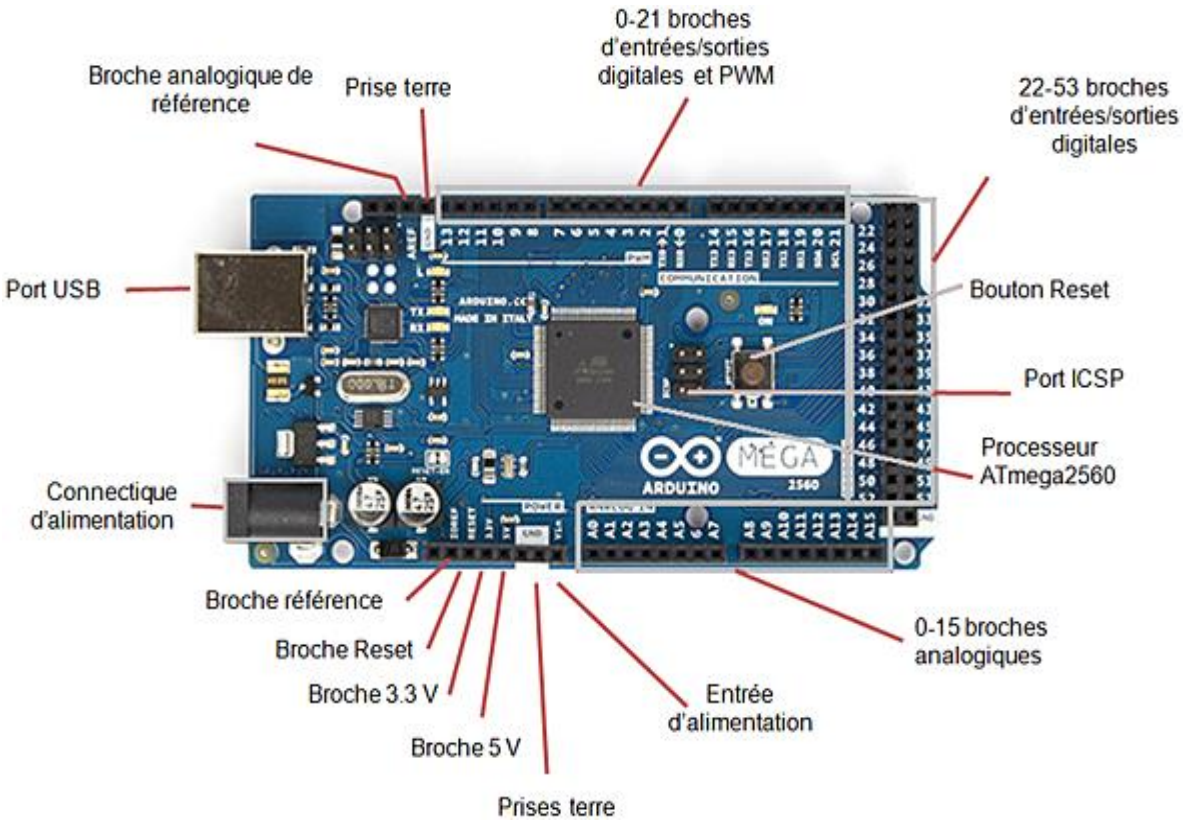

Fig. II. 7. La carte Arduino NANO [17]

II.4.1. Description ;

La carte Arduino Nano est basée sur un contrôleur ATMega328 contient un boot loader qui permet de modifier le programme sans passer par un programmateur, 16 MHz de cadence.
Sa mémoire de 32 KB et son grand nombre d'E/S font de ce circuit idéal pour les systèmes embarqués ou pour des applications robotiques nécessitant du multitâches.[24]

II.4.2. Caractéristiques :

- Alimentation :

-via port USB ou
-5 Vcc régulée sur broche 27 ou
- 6 à 20 V non régulée sur broche 30

- Microprocesseur : ATMega328
- Mémoire flash : 32 kB
- Mémoire SRAM : 2 kB
- Mémoire EEPROM : 1 kB
- Interfaces :
  - 14 broches d'E/S dont 6 PWM
  - 8 entrées analogiques 10 bits
  - bus série, I2C et SPI
- Intensité par E/S:40 mA
- Cadencement: 16 MHz
- Gestion des interruptions
- Fiche USB: mini-USB B
- Boîtier DIL30
- Dimensions: 45 x 18 x 18 mm

II.5. Mémoire MOS :

Une mémoire est par définition un dispositif capable d'emmagasiner puis de restituer l'information. Les trois fonctions essentielles qui définissent une mémoire sont l'écriture, l'effacement et la lecture de l'information, c'est-à-dire d'un ensemble de bits.
Le bit représente l'unité d'information élémentaire stockée dans une cellule ou point mémoire. Cette donnée binaire (1 ou 0) exploite des phénomènes physiques bistables à l'intérieur de chaque cellule mémoire, par exemples l'état d'un transistor (passant/bloqué), d'un fusible, du niveau logique d'une bascule, de la charge d'une capacité [12]

Le besoin progressive de la technologie a donné une naissance de divers types de mémoires. En effet chaque système, chaque application, nécessite un type de mémoire aux performances précises (densité, dimension, consommation, portabilité, durée de rétention de l'information, vitesse de lecture)

Il existe trois grandes catégories de dispositifs mémoires: les mémoires optiques (par exemple les CD et DVD), les mémoires magnétiques (par exemples les disques durs, les bandes magnétiques) et les mémoires à semi-conducteurs (**MOS**)

Les mémoires à semi-conducteur sont devisées en deux classes :

- ✓ Les mémoires vives volatiles aussi appelées RAM (Random Access Memory).
- ✓ Les mémoires non volatiles (NVM), peuvent être divisées en mémoires mortes ROM(Read Only Memory),mémoires réinscriptibles (EEPROM et Flash) et mémoires émergentes (FeRAM, MRAM,PCM…)

Dans notre mémoire de fin d'étude, on se limitera aux mémoires à semi-conducteurs, et précisément **l'EEPROM** appartienne aux mémoires non volatiles (NVM).

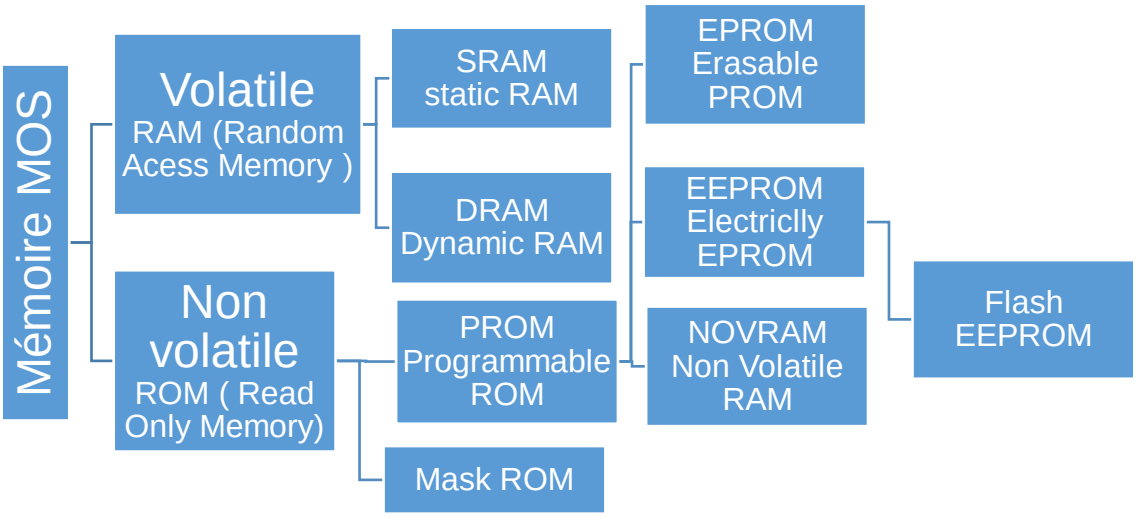

Fig. II. 8. Mémoires MOS [12]

Et pour mieux comprendre le motif du choix de l'**EEPROM**, nous allons parler brièvement sur les grandes générations des mémoires non volatiles apparues successivement sur le marché.

II.5.1. Les mémoires non volatiles :

L'avantage majeur qu'a apporté cette génération des mémoires, elle garde l'énergie ou bien l'information stocker même en l'absence de l'alimentation électrique. Cette énergie est magasinée dans un point mémoire ou cellule.

Les cellules sont attribuées suivant un schéma matriciel, d'où Chaque cellule est adressée dans cette matrice par des lignes horizontales appelées ***Word Lines (WL)*** et des lignes verticales appelées ***Bit Lines (BL)*.**

L'unité d'information exploitable la plus répandue est l'octet (le mot). Celui-ci est constitué de 8 cellules qui sont activées en même temps. Chaque octet de données possède donc une adresse qui lui est propre. [12]

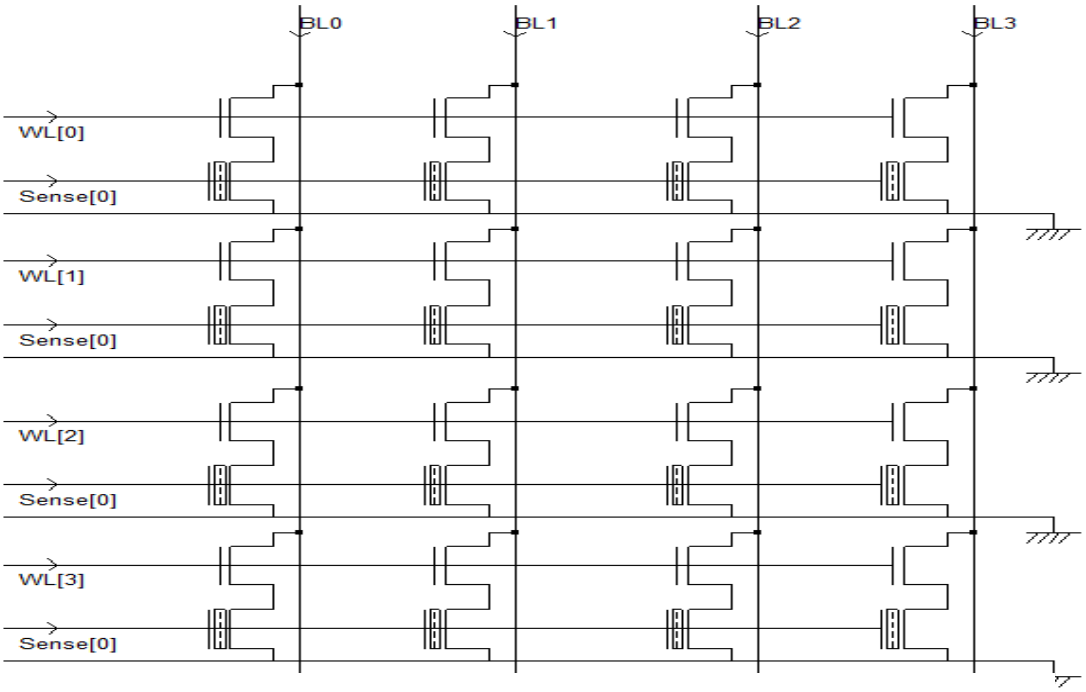

Fig. II. 9. Schéma matriciel des cellules [12]

II.5.1.1. Mémoire en lecture seule (ROM) :

Le contenu de cette mémoire ne peut être ni effacé ni modifié, uniquement à être lue, elle est programmée lors de fabrication, et par conséquence la mise à jour de son contenu nécessite un échange pur de composant. Une cellule ROM est constituée d'un transistor bloqué ou passant en permanence. [12]

II.5.1.2. Mémoire morte programmable (PROM):

Les PROM comme l'indique leur nom sont des ROM ou l'utilisateur peut les programmer. Cette programmation ni valable qu'une seule fois, et son contenu ne peut être ni effacé ni modifié, uniquement à être lue.

La cellule PROM est constituée d'un fusible (ou d'un fusible en série avec une diode ou d'un fusible en série avec un transistor) reliant la BL à la WL.

Ces technologies PROM sont maintenant abandonnées à cause de leur faible capacité d'intégration (taille importante des fusibles) et le manque de fiabilité dans la programmation. [12]

II.5.1.3. Mémoire morte programmable effaçable (EPROM) :

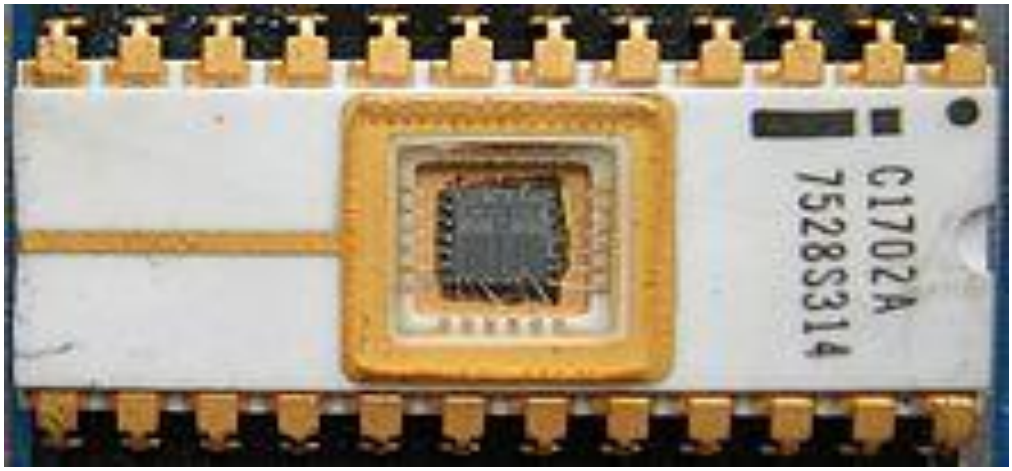

Fig. II. 10. La première EPROM fabriquée par INTEL, 256 × 8, ~1971. [11]

L'EPROM ou UV-ROM son programmation se faite électriquement et l'effacement en exposition aux rayons UV. Une cellule EPROM est constituée d'un transistor NMOS double grille :

- ✓ L'application des tensions de lecture et de programmation de la cellule sont dirigées par la grille de contrôle (CG poly2).
- ✓ Une qui stocke l'information mémoire est 'une concentration d'électrons stocke par la grille flottante (FG poly1). Cette grille est dite flottante car elle est totalement isolée de la structure par une couche **ONO** (Oxyde-Nitrure-Oxyde) qui la sépare de la grille de contrôle et par un oxyde de grille d'épaisseur >15nm qui la sépare du canal. Une cellule EPROM standard est réalisée en technologie Stacked Gate Avalanche Injection MOS présentée ci-dessus. [12]

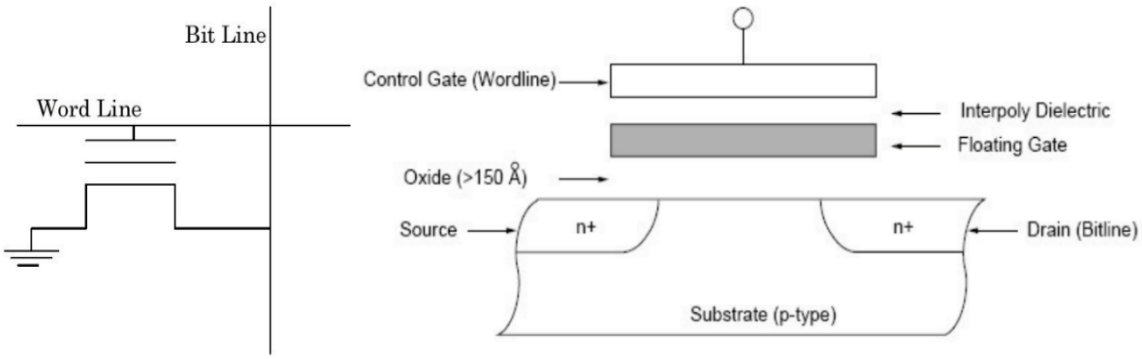

Fig. II. 11. Schéma représente la technologie Stacked Gate Avalanche injection MOS. [12]

- ✓ Une cellule dont la grille flottante est riche en électrons a un VThaut très positif généralement supérieur à 5V.Pour une tension de lecture$VTbas < Vread < VThaut$, elle est donc bloquée.
- ✓ Une cellule avec une grille flottante vide d'électrons possède une VTbas faiblement positif,

VTbas=1V a 3V.Elle est donc passante pour une tension de lecture $Vread > VTbas$ appliquée sur sa grille de contrôle.

II.5.1.4. Mémoire morte programmable effaçable électriquement (EEPROM) :

Pour éviter l'effacement de l'EPROM en utilisant les rayons **UV**,une invention a donné naissance d'une nouvelle génération de mémoire NVM programmable et effaçable électriquement. Il s'agit de l'**EEPROM** qui répond au cahier de charge suivant :

- Un effacement électrique sur place, c'est-à-dire composant inclus dans le système.
- Un effacement électrique rapide. Le cycle d'écriture complet d'un octet (effacement + programmation) dure $t_{WC} = 1ms$ à $10ms$.
- Un effacement partiel, c'est-à-dire la possibilité d'effacer un seul octet à la fois. Cela nécessité le rajout d'un transistor d'accès par cellule.
- Programmation en utilisant un courant électrique rapide et octet-sélective.
- Accessibilité en lecture de l'ordre de $t_{ACC} \approx 50\text{ns} - 200\text{ns}$.
- Une fiabilité de longue durée de stockage supérieure à 10 ans (jusqu'à 200 ans pour l'EEPROM) et une endurance de l'ordre 105 à 106 cycles d'écriture.
- Une faible consommation en veille (1µA-100µA) et en lecture/écriture (1mA-30mA) pour des applications embarquées. [12]

A. Description :

Une cellule EEPROM est constituée de deux transistors par cellule : un transistor de stockage (**FGT**) qui permet une long durée de stockage de l'information, et un transistor d'accès (**AT**) par lequel se déroule les opérations de programmation/effacement/lecture octet par octet (et même bit par bit en théorie, vu qu'il y a un transistor d'accès par cellule). La programmation et l'effacement sont symétriques, réalisés par effet tunnel Fowler-Nord Heim. [12]

Le transistor d'accès joue deux rôles majeurs, l'un est l'effacement bit par bit l'autre est la protection du transistor de stockage qui le suit. S'accomplir par l'isolation de ce dernier de la ligne de bits (BL). Théoriquement on peut dire qu'une mémoire EEPROM est plus fiable par rapport à une cellule qui ne présenterait qu'un seul transistor. Mais (2T/cellule) représente un désavantage majeur car sa taille de cellule importante limitant les possibilités de réalisation d'EEPROM de grande densité. La gamme disponible est de 64kbits à 10Mbits.

La technologie utilisée dans la fabrication de transistor de stockage de l'EEPROM est, la technologie a grille flottante de type **FLOTOX.**

Le FLOTOX est un transistor de MOS double grilles à canal N. Il présente un amincissement localisé de son oxyde de grille côté drain. Cette région possédant un oxyde tunnel fin typiquement de 8nm à 12nm permet l'injection des électrons en provenance du drain vers la

grille flottante en poly silicium et l'extraction des électrons en provenance de la grille flottante vers le drain. Ces deux opérations sont réversibles et réalisées par effet tunnel Fowler-Nord Heim. [12]

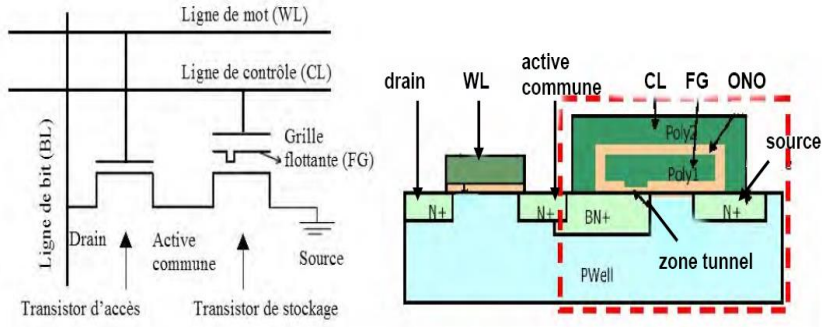

Fig. II. 12. Schéma électrique d'une cellule EEPROM. [12]

B. Principe de fonctionnement d'une cellule EEPROM :

Une variation de la valeur de la charge QFG stockée sur la grille flottante du transistor FGT agit directement sur sa tension de seuil **VT** sur laquelle repose le fonctionnement de ce transistor.
Ce principe est présenté par une équation valable pour tout transistor FGT donnant VT en fonction de QFG :

$$V_T = V_{Tnul} - \frac{Q_{FG}}{C_{ONO}} \text{ où } V_{Tnul} = \frac{Vt_{FS}}{\alpha_{ONO}} \qquad (1)$$

où V_T est la tension seuil du transistor mesurée au niveau de la grille de contrôle, Vt_{Fs} est la tension seuil au niveau de la grille flottante, CONO la capacité de la couche ONO située entre les deux grilles et $\alpha_{ONO} = C_{ONO}/C_T$ le coefficient de couplage inter grille qui est par définition le rapport de la capacité C_{ONO} de la couche *ONO* sur la capacité totale C_T de la grille flottante. Vt_{fs} est indépendante de QFG et ne dépend que de la technologie. Il s'agit de la tension qu'il faudrait appliquer sur la grille flottante pour atteindre le régime d'inversion de population (canal passant) lorsque aucune tension n'est appliquée entre le drain et la source $V_D=V_S=0V$.[12]
Cette équation fondamentale montre que la tension seuil VT d'un transistor FGT est la somme de deux termes. Le terme VTnul qui ne dépend que de la technologie et le terme QFG/CONO qui est fonction de la charge stockée sur la grille flottante. Cette équation explique pourquoi les transistors à grille flottante sont utilisés pour réaliser des NVM. En modifiant la charge de la grille flottante d'une variation (positive ou négative) de QFG/CONO, il est possible de définir trois états mémoires à partir de dernier équation :

- ✓ Un état haut VThaut>VTnul si QFG<<0, autrement dit si des électrons sont stockés sur la grille flottante. Le canal du transistor FGT est alors régime accumulation de trous, donc la cellule est bloquée.

- ✓ Un état vierge VT=VTnul vide de charge si QFG=0. Le transistor FGT est en régime de désertion si VCL=0V est appliquée sur sa grille de commande.
- ✓ Un état bas VTbas<VTnul si QFG>>0, autrement dit si des trous sont stockés sur sa grille flottante. Le canal du transistor FGT est en régime d'inversion donc passant à VCL=0V.

Ces trois états mémoires vérifient donc l'inéquation

$$V_{Tbas}=V_{Tnul}-\frac{Q_{FGbas}}{C_{ONO}}<V_{Tnul}<V_{Thaut}=V_{Tnul}-\frac{Q_{FGhaut}}{C_{ONO}} \qquad (2)$$

Dans le cas d'un FLOTOX, elle est toujours centrée sur 0V et vaut typiquement $\Delta VT \approx 4V$ pour une EEPROM alimentée en VCC=5V. La tension seuil haute d'un FLOTOX est positive et vaut typiquement VThaut>+2V. La tension basse est négative typiquement VTbas<-2V. La **Fig. II 13** présente une courbe d'endurance d'un transistor FLOTOX. Une augmentation du nombre de cycle de programmation/effacement au-delà de 100 provoque un début de fermeture de la fenêtre de programmation.

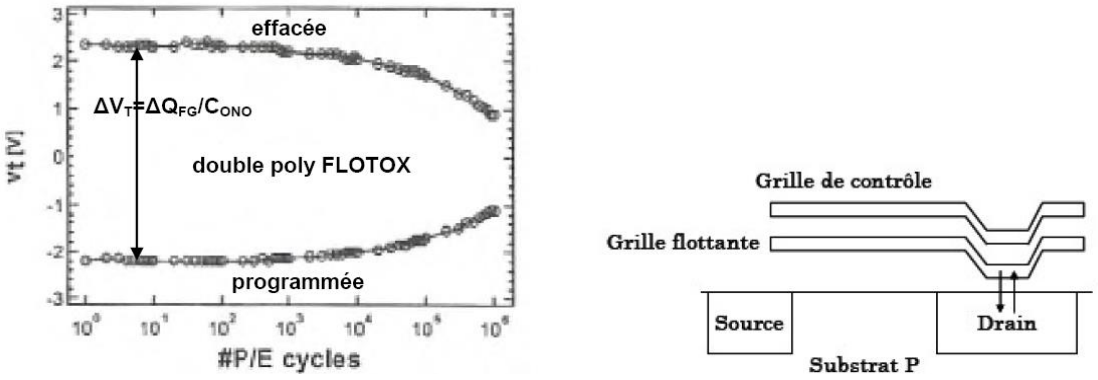

Fig. II. 13. Courbe d'endurance d'un transistor FLOTOX. [12]

- ✓ Endurance d'une EEPROM FLOTOX programmée et effacée par effet tunnel FN. L'état programmé a unVTbas $\approx -2V$, l'état effacé un VThaut $\approx +2V$ et doncΔVT $\approx 4V$. Le niveau de référence de détection de l'amplificateur de lecture est fixé au centre de la fenêtre de programmation, soit Vread=0V. Mouvements des électrons par effet tunnel FN entre la grille flottante et le drain du FLOTOX lors des étapes d'effacement (injection) et de programmation (extraction). Ces deux étapes sont parfaitement symétriques. [12]
- ✓ Lors d'une étape de lecture de la donnée stockée sur une cellule mémoire, une tension VCL=Vread=0V est appliquée sur la grille de contrôle, une tension $VD \approx 1V$ est appliquée sur le drain,

La source et le substrat étant à la masse VS=VB=0V. En conventions EEPROM :

- ✓ Une cellule effacée, dont la grille flottante est riche en électrons, possède une tension haute supérieure à la tension de lecture Vread<VThaut. Par conséquent, le canal du transistor est en régime d'accumulation de trous et la cellule est bloquée, ce qui correspond par convention à un « 1 » logique.
- ✓ Une cellule programmée, dont la grille flottante est riche en trous, possède une tension basse inférieure à la tension de lecture VTbas<Vread. Par conséquent, le canal du transistor atteint le régime d'inversion de population dans les conditions de lecture et la cellule est passante, ce qui correspond par convention à un « 0 » logique.

Pour programmer une cellule EEPROM, il faut injecter des charges positives dans la grille flottante (autrement dit extraire des électrons de la grille flottante). Ceci est réalisé en polarisant le drain de la cellule à un potentiel positif VPP=12V à 16V, la grille de contrôle et le substrat étant à la masse et la source étant flottante (haute impédance HZ).

La grille du transistor d'accès est polarisée à un potentiel fortement positif de VSEL=12V à 16V afin de rendre ce transistor passant. Les électrons transitent alors par effet tunnel de la grille flottante vers la zone active commune aux deux transistors, zone qui sert de drain au FLOTOX **(voir Fig.II.13).** Cette extraction d'électrons se poursuit jusqu'à accumulation de trous dans la grille flottante (QFG>>0) et décale la caractéristique IDS(VCS) du transistor de stockage du côté des tensions négatives par rapport à la caractéristique d'une cellule vierge : la tension seuil VTbas d'un transistor effacé est donc inférieure à la tension seuil d'un transistor vierge VTnul.

Pour effacer une cellule EEPROM, il faut injecter des électrons dans la grille flottante. Il faut inverser le sens du champ électrique appliqué aux bornes de l'oxyde tunnel (typiquement 10MV/cm) par rapport à une opération de programmation. On réalise cela en polarisant cette fois la grille de contrôle à un potentiel positif de VPP=12V à 16V, le drain et le substrat étant à la masse et la source étant flottante. La grille du transistor d'accès est polarisée à un potentiel positif de VSEL=12V à 16V pour le rendre passant. Les électrons transitent donc de la zone active commune vers la grille flottante à travers l'oxyde tunnel du FLOTOX par effet tunnel (voir **Fig. II 14**). Cette fois-ci, la caractéristique IDS(VCS) est décalée vers les tensions positives par rapport à une cellule vierge et sa tension seuil VThaut est supérieure à celle d'une cellule vierge VTnul. La **Fig. II 14** présente les polarisations des électrodes d'une cellule EEPROM en phase de programmation et d'effacement, ainsi que les caractéristiques IDS(VCS) correspondant aux trois états mémoires (programmé, vierge et effacé).

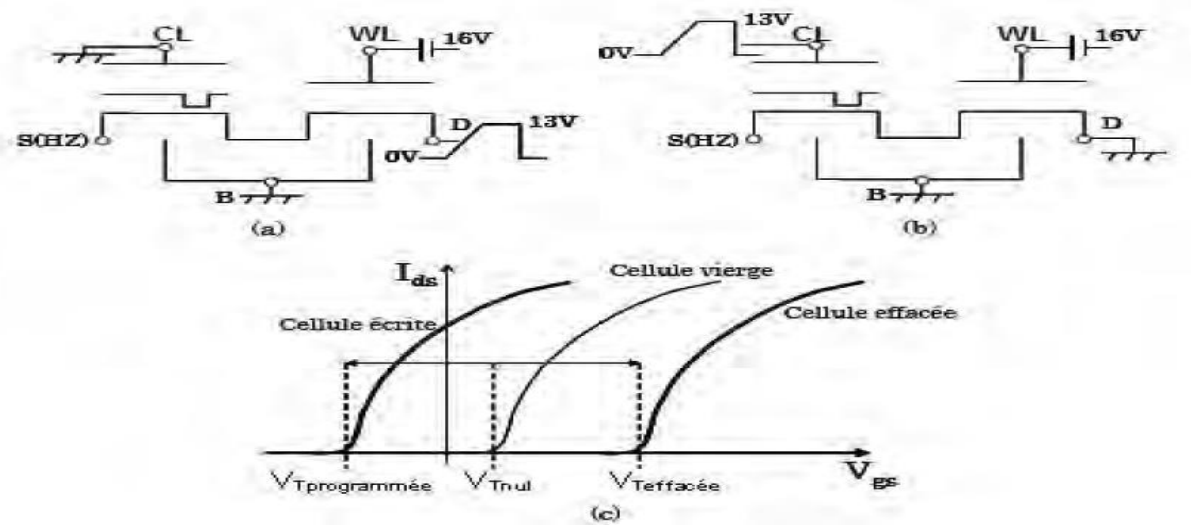

Fig. II. 14. Polarisation d'une cellule EEPROM [12]

(a) programmation et (b) effacement. (c) Caractéristiques IDS(VCS) des trois états mémoires (programmé, vierge et effacé). [12]

Lors d'une opération de lecture, une tension VCS=Vread=0V est appliquée sur la grille de contrôle, une tension de pré charge $VDS \approx 1V$ est appliquée sur le drain, la source et le substrat étant à la masse VS=VB=0V. L'état logique d'une cellule EEPROM est lu en détectant le courant circulant dans la ligne de bits puis en le comparant à un courant de référence IREF. Pour cela, le circuit de lecture est constitué d'un amplificateur et d'un comparateur :

- ✓ Si la cellule est dans l'état programmé, la tension de pré charge VDS chute de 1V vers 0V du fait de la présence d'un courant non nul circulant dans la BL. Le circuit de lecture détecte ce courant et le compare au courant de référence. S'il est supérieur à ce dernier, la cellule est considérée dans l'état passant « 0 ».
- ✓ Si la cellule est dans l'état effacé (bloqué), la tension de pré charge VDS reste à 1V du fait de l'absence de courant dans la BL. Le circuit de lecture détecte un courant très faible inférieur au courant de référence, et la cellule est considérée dans l'état bloqué « 1 ».

C. Architecture des matrices EEPROM :

Une cellule EEPROM, constituée de deux transistors MOS en série (transistor d'accès et transistor de stockage), est reliée au reste du plan mémoire par quatre électrodes : la ligne de mots (WL), la ligne de contrôle (CL), la ligne de bits (BL) et la ligne de source (SL). Le transistor d'accès agit comme un interrupteur placé entre le drain du transistor de stockage et la ligne de bits BL. Sa grille est commandée par la ligne de mots WL, alors que celle du transistor de stockage est commandée par la ligne de contrôle CL. Enfin, le substrat des deux transistors et la source du transistor de stockage sont reliés à la masse (dans certaines technologies la source du

transistor de stockage peut également rester flottante).

Une matrice EEPROM est organisée suivant n lignes (WL) et m colonnes (BL) de cellules mémoires. Il y a un point mémoire à chaque intersection d'une ligne et d'une colonne. Du fait de la présence du transistor d'accès dans chaque cellule, l'architecture d'une matrice EEPROM ne peut être que de type NOR (contact de BL spécifique à chaque cellule et source commune). Les points mémoires qui se trouvent sur une même ligne WL forment un mot de k bits, auquel s'ajoute un transistor de sélection du mot (ou octet si k=8). Sur la **Fig-24-** nous avons l'exemple d'une matrice EEPROM 4x4 : c'est une matrice de 4 mots de 4 bits. Ce plan mémoire comporte quatre lignes de mots (WL0, WL1, WL2, WL3), quatre lignes de contrôle (CL0, CL1, CL2, CL3), quatre lignes de bits (BL0, BL1, BL2, BL3) et une ligne de grille (GL). La brique de base du plan mémoire est le mot (i, k) : il est constitué de k cellules mémoires, d'un transistor de sélection de mot, d'une ligne de mots WLi, d'une ligne de contrôle CLi et de k lignes de bits BLj. Le transistor de sélection du mot permet, comme son nom l'indique, de sélectionner le mot qui suit. Il s'agit d'un transistor FLOTOX identique à ceux des points mémoires et ayant sa grille reliée à WLi, son drain relié à GL et sa source reliée à CLi

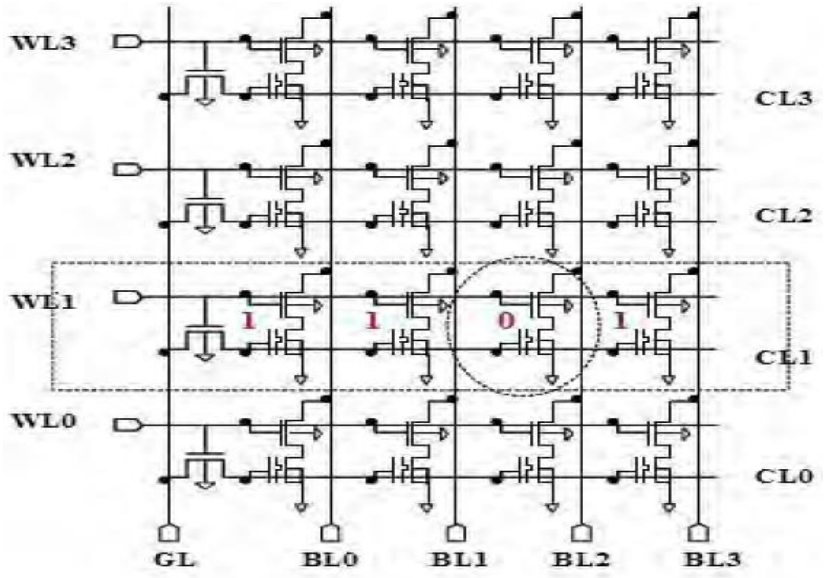

Fig. II. 15. Matrices EEPROM 4x4 [12]

D. Transfert des données à l'EEPROM :

Le transfert des données à l'EEPROM habituel se fait ailleurs à l'aide d'un programmateur qui supporte l'EEPROM et qui doit être connecté au PC. Cette opération se déroule par l'intermédiaire d'un logiciel installé sur le PC. Le plus connu est Le programmateur universel pour EOROM, EEPROM et mémoires FLASH [13]

27Cxxxx I 28Cxxxx I 28Fxxxx I 29Cxxxx I 29Fxxxx I 39Fxxxx I 49Fxxxx

zr »ou«‹zszz»«oMsis»i-F‹asH24cxxx i 2sxxx i2scxxx i 2s‹xxx i ssxxx

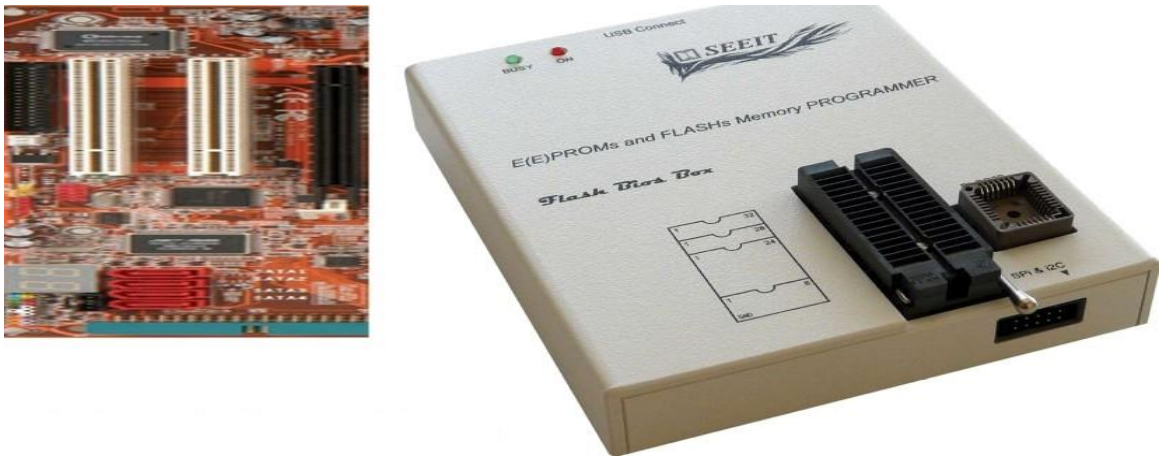

Fig. II. 16. Programmateur universel pour EEPROM SEEIT [13]

Mais dans notre mémoire de fin d'étude le transfert des données se fera à l'aide de l'Arduino, et ceci représente un avantage que nous avons gagné.

II.6. L'EEPROM 24C512B :

Dans un montage il est conseiller ou bien est une obligation de faire la détente de la mémoire du microcontrôleur ou de substituer l'utilisation de la mémoire du microcontrôleur vers une autre mémoire pour le préserver. [28]

La mémoire 24C512B est une EEPROM destiné pour le stockage d'une façon permanente des données mais avec la possibilité de les modifier simplement, fonctionnent selon la norme **I2C**, c-à-dire les interfacer avec seulement 2 fils.

II.6.1. Caractéristiques des 24C512B :

La mémoire 24C512B dispose de 512 kilobits d'espace de stockage 512 / 8 bits = 64 KO de mémoire, de plus le nombre de cycle d'effacement et d'écriture est limité dans la mémoire : pour un microcontrôleur Atméga elle est de 100000 cycles (cent mille), Pour une 24C512B elle est de 1000000 (1 million).la consommation est (400 µA en fonctionnement, 100 nA en standby), le temps d'accès en écriture est 5ms max, alimentation entre 2.5V et 5.5V.

La protection contre l'écriture s'établie via la mise à l'état haut d'une broche du circuit, sur un même montage peuvent être installées 8 mémoires de types 24C512B.

II.6.2. Le bus I2C :

L'interface **I2C** permet de commander le composant avec simplement deux fils.

• **SCL** : représente le signal d'horloge.

• **SDA** : représente le signal des données.

L'intégration d'un composant dans un montage est très facile en utilisant le bus I2C car il permet d'avoir seulement 8 broches. Ce bus supporte 127 composants, tous branchés sur le

même fil de données, ils reçoivent tous les informations qui circulent sur ce fil.
Chaque composant est doté d'une adresse unique sur le fil de données pour pouvoir le commander seul, si l'on souhaite accéder à un composant, en lecture ou en écriture, il faut impérativement faire précéder cette demande de l'adresse du composant qui est sollicité. Cette adresse est codée sur 7 bits (127 possibilités).
Une partie de la mémoire 24C512B est fixée par le constructeur par câblage interne au composant, elle a pour adresse la valeur : **1010.**
L'utilisateur peut choisir une adresse sur le bus I2C parmi 8, grâce à la position des 3 autres bits (**A0**, **A1**, **A2**) qui sont à son disposition. Cela explique pourquoi on ne peut brancher sur un même bus I2C que 8 mémoires 24C512B. [28]

II.6.3. Les broches de la 24C512B :

La mémoire 24C512B contient 8 broches à savoir :
• **VSS** : Masse.
• **VCC** : Alimentation.
• **WP** : Broche de protection à l'écriture.
• **SCL** : Signal d'horloge du bus I2C.
• **SDA** : Signal de données du bus I2C.
• **A0** permet de fixer l'adresse du composant sur le bus I2C.
• **A1** permet de fixer l'adresse du composant sur le bus I2C.
• **A2** permet de fixer l'adresse du composant sur le bus I2C.

II.6.4. Attribution de l'adresse de la mémoire 24C512B :

Chaque mémoire utilisant le bus I2C nécessite l'attribution de sa propre adresse pour pouvoir accéder dedans. Le bus 12C peut accepter 127 circuits pour les faire brancher, il faut donc leur attribuer une adresse sur 7 bits.
Comme nous avons mentionné précédemment qu'une partie de l'adresse de 24C512B est fixée par le constructeur qui sont les 4 premiers bits représentent (Le code de contrôle : 1010.) ; et les 3 suivants pour le choix de l'utilisateur représentent **:** Les bits de sélection du composant : A2, A, A0 représentant les bits les plus significatifs de l'adresse.et pour cette raison que 8 mémoires 24C512B peuvent être branché sur le même bus I2C et étendre la capacité mémoire à (512Kb * 8).

Adresse esclave						
Code de contrôle				Bits de sélection du composant		
1	0	1	0	A2	A1	A0

Tab.II. 1. Adressage de mémoire 27C512B [28]

L'utilisateur peut commander par les bits de sélection du composant. [28]

• **A0** branché au **VCC (+)** cela **fixe un 1**, branché au **VSS(-)** cela **fixe un 0.**

• **A1** branché au **VCC (+)** cela **fixe un 1**, branché au **VSS(-)** cela **fixe un 0.**

• **A2** branché au **VCC (+)** cela **fixe un 1**, branché au **VSS(-)** cela **fixe un 0.**

Donc les combinaisons possibles pour choisir une adresse pour chaque mémoire parmi les 8 qui peuvent être branchées sur le même bus 12C sont **:**

• 1010 000 en binaire : 80 en décimal ou 0x50 en hexadécimal.

• 1010 001 en binaire : 81 en décimal ou 0x51 en hexadécimal.

• 1010 010 en binaire : 82 en décimal ou 0x52 en hexadécimal.

• 1010 011 en binaire : 83 en décimal ou 0x53 en hexadécimal.

• 1010 100 en binaire : 84 en décimal ou 0x54 en hexadécimal.

• 1010 101 en binaire : 85 en décimal ou 0x55 en hexadécimal.

• 1010 110 en binaire : 86 en décimal ou 0x56 en hexadécimal.

• 1010 111 en binaire : 87 en décimal ou 0x57 en hexadécimal.

Exemple : Si les 3 broches A2 relie au **VSS (+)**, A1 relie au **VSS (+)** et A0 relie au **VSS (-)** ; l'adresse du composant sera : **1010 110 en binaire : 86 en décimal ou 0x56 en hexadécimal.**

Donc à l'aide de cette adresse qu'on peut accéder a cette mémoire.

II.6.5. Mise en œuvre la mémoire 24C512B utilisant le logiciel Arduino IDE :

Pour écrire ou lire dans cette mémoire elle doit être d'abord raccordée à la carte Arduino. Pour cela on faire appel à la bibliothèque "**Wire**" présente dans l'IDE Arduino ce qui facilite le travail.

• **# Include <Wire.h>**

ensuite on déclare l'adresse de la mémoire par l'instruction **« define » :**

• **# define** EEPROM_ADRESSE 0x50.

```
#include < Wire.h>
#define 24LC512_ADRESSE 0x50
void setup()
{
//Instructions exécutées une fois.
}
```

La carte Arduino gère par l'intermédiaire du sketch les circuits branchés sur le bus I2C, elle est donc : le **MAÎTRE,** et la mémoire branchée sur le bus I2C sera un **ESCLAVE**.La carte Arduino doit être déclarée en tant que **MAÎTRE** par l'instruction **« Wire.begin »** :

• **Wire.begin ()** ;

```
void setup()
{

Wire.begin () ;
```

A. Ecriture :

Le code doit être insérer dans une fonction pour éviter d'épuiser la mémoire à cause du nombre de cycles d'enregistrement possible. [28]

L'ordre successif d'instructions pour stocker une donnée dans la mémoire doit être respecté:

1). Indiquer à quelle périphérique l'on s'adresse :

• **Wire.beginTransmission** (EEPROM _ADRESSE);

2). La 24C512B à une capacité de 64 koctets il est donc nécessaire de coder cette adresse sur 2 octets, et indiquer à quelle adresse de la mémoire on veut stocker la donnée.

• **Wire.write (adresse &0xFF);** //Position adresse EEPROM : Octet le moins significatif.

• **Wire.write (adresse >> 8);** //Position adresse EEPROM : Octet le plus significatif.

3). A l'adresse sélectionnée ou la donnée est stocké :

• **Wire.write** (donnée);//La valeur de la donnée stockée ne doit pas dépasser 255 (8 bits).

4). fin de transmission :

• **Wire.endTransmission ();**

```
Wire.write (adresse >> 8);
Wire.write (adresse >> &0xFF);
Wire.write (donnee);
Wire.endTransmission ();
}
```

B. Lecture :

Comme nous avons dit précédemment il faut toujours respecter l'ordre successif du jeu d'instructions : [28]

1). Indiquer à quelle périphérique l'on s'adresse:

• **Wire.beginTransmission** (EEPROM _ADRESSE);

2). Indiquer quelle est l'adresse de la mémoire que l'on veut lire. La 24CB512 a une capacité de 64 koctets il est nécessaire de coder cette adresse sur 2 octets :

• **Wire.write** (adresse &0xFF); //Position adresse EEPROM : Octet le moins significatif.

• **Wire.write** (adresse >> 8); //Position adresse EEPROM : Octet le plus significatif.

3). Indiquer la fin de transmission:

• **Wire.endTransmission** ();

4). Interroger la mémoire:

• **Wire.requestFrom** (EEPROM _ADRESSE, 1); //Adresse et quantité 1 octet

5). Stocker la donnée dans une variable:

• adresse_stockage_donnee = **Wire.read** ();

6). Indiquer la fin de transmission:

• **Wire.endTransmission** ();

C. Appel à une donnée stocké :

" **Adresse_stockage_donnee** " est une variable ou la donnée est stockée (déclarée au préalable par un **int**), et peut ensuite être utilisée dans le sketch.

```
adresse_stockage_donnee = Wire.read ();
Wire.endTransmission ();
}
```

II.7. Les afficheurs LCD interface I2C :

Fig. II. 17. Ecran LCD avec un I2c en arrière [19]

II.7.1. Description :

Ces 16 caractères par 2 lignes d'affichage a un contraste texte blanc très clair et très haut sur un fond bleu / rétro-éclairage. Il comprend également une série I2C carte adaptateur pré soudé à l'arrière de l'écran LCD. Cela signifie qu'il peut être contrôlé avec seulement 2 I2C broches de données série (**SDA** et **SCL**) et nécessite donc beaucoup moins broches numériques lorsqu'il est commandé à partir d'un microcontrôleur. Au total, le module ne nécessite 4 fils dont la puissance 5V et GND. Le réglage du contraste est également fourni par la carte fille par l'intermédiaire d'un potentiomètre.

Note :

Ces modules sont actuellement fournis avec une adresse I2C par défaut soit 0x27 ou 0x3F. Pour déterminer quelle version, nous allons vérifier l'adaptateur I2C tableau noir sur la face inférieure du module. S'il y a 3 jeux de patins étiquetés A0, A1, A2 et puis l'adresse par défaut sera 0x3F. S'il n'y a pas l'adresse par défaut sera 0x27. Le module est équipé d'un potentiomètre de réglage de contraste sur la face inférieure de l'écran. Cela peut nécessiter de réglage pour l'écran pour afficher le texte correctement. Si la pression est appliquée à la carte fille I2C il est possible de se plier et de venir en contact avec le module LCD. [14]

Les pins de LCD (I2C) sont : GND, VCC (+5V), SDA et SCL

On trouve sur la toile des afficheurs LCD équipés d'adaptateur I2C types PCA8574

Ce sont des "expandeurs de bits", ils reçoivent un octet sur l'entrée I2C et le ventilent sur 8 sorties.

II.7.2. Bloc diagramme :

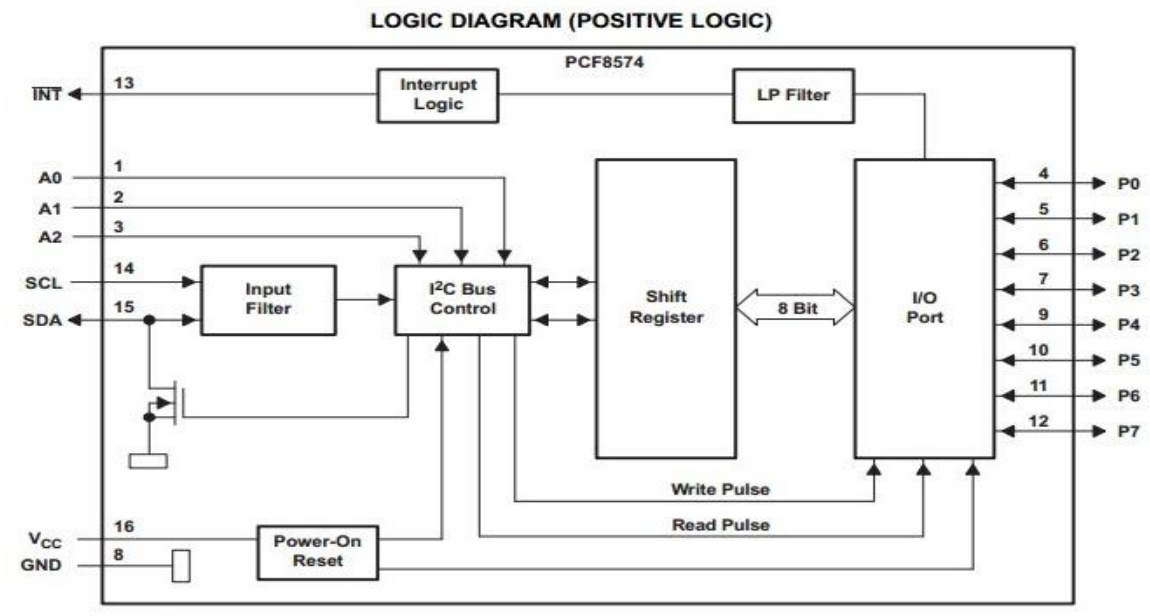

Fig. II. 18. Bloc diagramme adaptateur 12C types PCF8574 [14]

Ce bloc diagramme présente les 16 broches des CI PCF8574 implantés sur ces adaptateurs :
Les broches SDA, SCL, Vdd, Vss (GND) que l'on retrouve sur le connecteur.
Les broches A0, A1, A2, permettant d'adapter l'adresse I2C si elles sont accessibles et modifiables.

II.7.3. Les liaisons entre le PCF8574 et LCD relevées sur afficheur avec adaptateur intégré

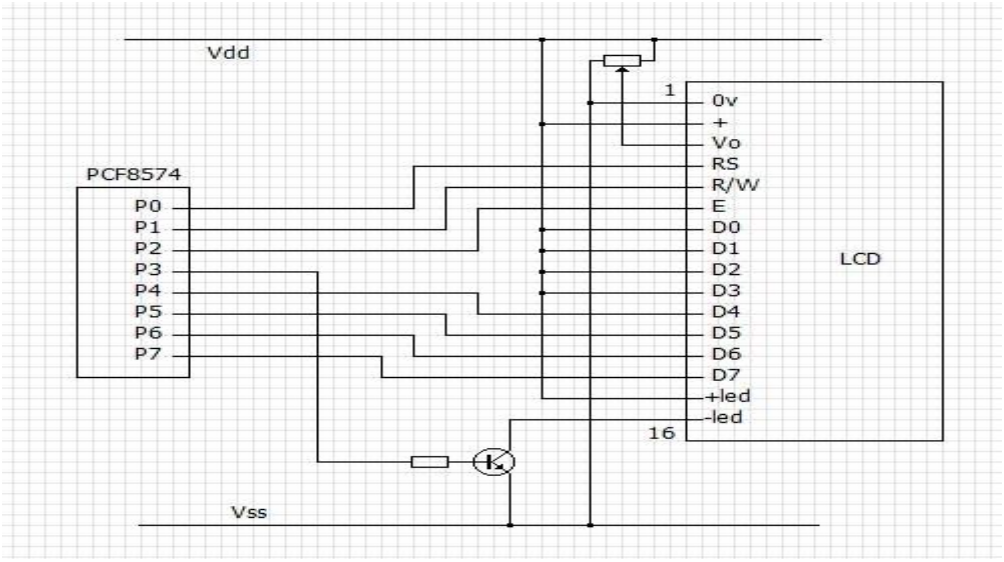

Fig. II. 19. Les liaisons entre adaptateur 12C types PCF8574 et LCD [14]

II.8. Conclusion :

Dans ce chapitre nous avons parlé sur les matériels et logiciels utilisées dans la réalisation du testeur des circuits logiques à base d'Arduino. Nous avons entamé une étude pour chaque élément en terme, cote caractéristiques technologiques ainsi que de cote programmation.
Nous avons illustré les avantages et les inconvénients de chaque élément, aussi le principe de fonctionnement. Et par conséquent notre choix de ces éléments a été basé sur une vue technique étudié remplissant nos besoins ou bien le cahier des charges désir

III.1. Introduction :

Comme nous l'avons dit précédemment que le testeur des **CIs** logiques est un circuit basé sur un microcontrôleur qui teste si le CI est en bon état de fonctionnement ou en mauvais état. L'existence d'un testeur des CIs dans les établissements d'enseignement est devenue inévitable, car il est capable de tester des circuits intégrés avant de faire des travaux pratiques pour éviter les erreurs et les résultats indésirables. De même, dans les industries, le test du produit est un processus important, coûteux et long, et avant de faire fonctionner le système il est nécessaire de vérifier si les CIs sont bons ou mauvais, cette vérification est la plus intéressante pour nous car elle facilite grandement la réparation des choses lorsque nous savons quelle pièce est défectueuse plutôt que de remplacer toutes les bonnes pièces par essais et erreurs, et tout ça va nous faire gagner du temps et de capacité.

III.2. Choix du testeur des CIs logiques à base d'Arduino :

D'après les opinions d'utilisateurs sur les sociaux media, les testeurs des CIs logiques disponibles sur marché cités précédemment ont créés des problèmes, il n'a pas été reconnu dans certains cas et le programme a gelé pendant les tests des CIs, aussi la nécessité d'existence d'un autre appareil ;qu'est le programmeur de l'EEPROM avec son logiciel nous a motivé pour chercher une solution assez simple. [15]

La solution a donc été un testeur des circuits intégrés basé sur Arduino avec une sortie série en option qui fait le travail dans la plupart des cas (il reste encore des améliorations à apporter), et donc de surmonter ces inconvénients en développant un testeur basé sur un Arduino Nano.

Le testeur des CIs logiques à base d'Arduino proposé est petit, portable et facile à manipuler.

III.3. Choix des composants électroniques :

Le choix d'Arduino NANO a était basé en générale sur une vue technique, ainsi que la disponibilité sur marché à savoir :

- Pour sa simplicité d'utilisation.
- Son prix bas par rapport aux autres modèles.
- Du fait du nombre de ports disponibles, le nombre maximum de ports à tester est de 16 (ce qui est suffisant pour la plupart des circuits intégrés).

L'écran LCD est un simple écran 16 x 2 standard comprenant un adaptateur I2C, qui ne nécessite que deus broches (SDA et SCL) d'Arduino.

Le choix de l'EEPROM (utilisée ici, la série AT24C512) a été détaillée dans le chapitre II, mais ça n'empêche pas de rappeler que les avantages majeurs sont :

- L'écriture, l'effacement et la programmation à l'aide d'un courant électrique dont le composant est surmonté sur le système.
- Une fiabilité de longue durée de stockage de l'information, supérieure à 10 ans.
- Accessibilité en lecture rapide.

III.4. Montage électronique du testeur fonctionnel à base d'Arduino :

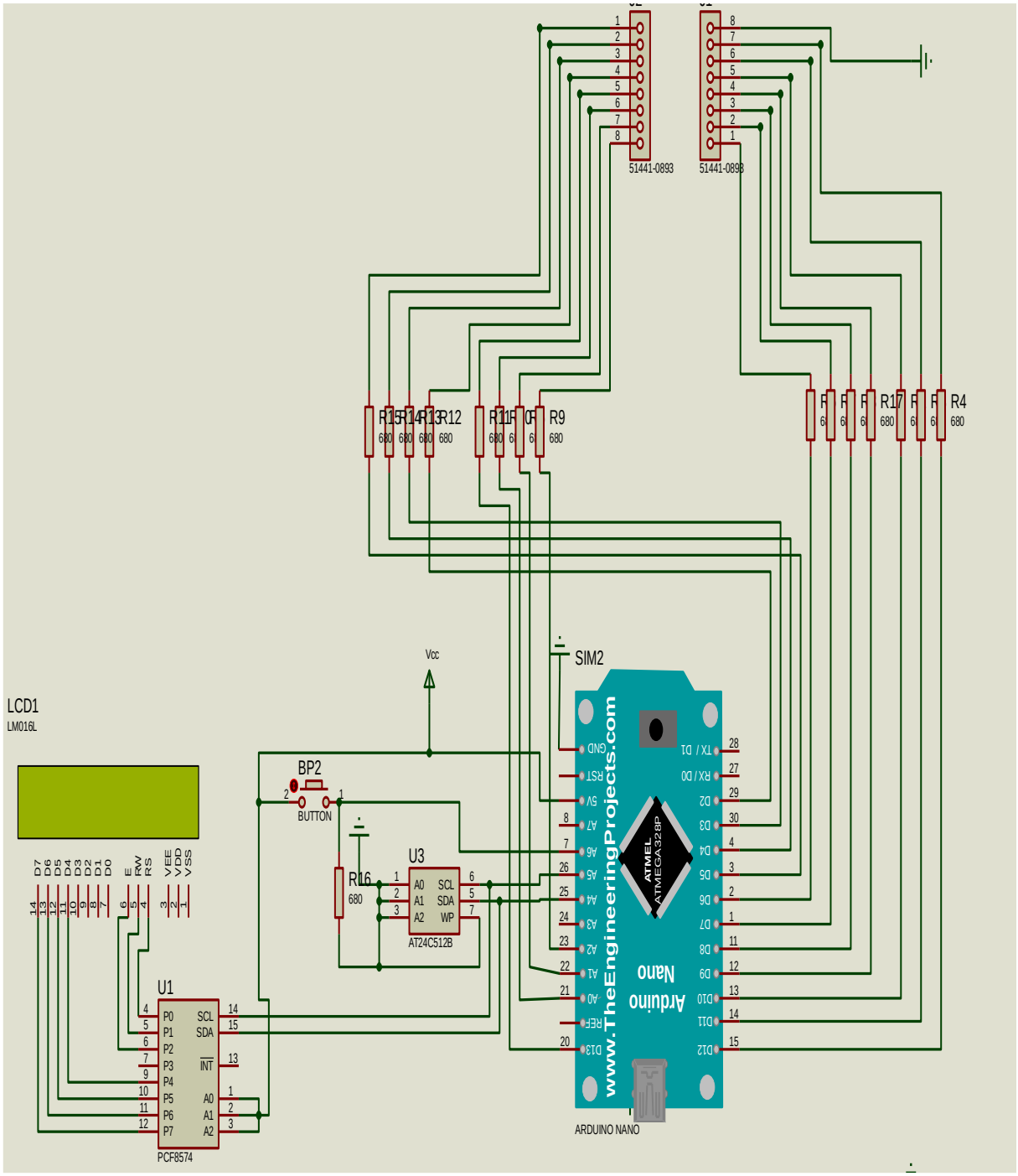

Fig. III. 1. Montage électronique du testeur fonctionnel à base d'Arduino. [15]

III.5. Description du testeur fonctionnel à base d'Arduino :

La connexion globale du circuit pour le testeur fonctionnel des circuits logiques à base d'Arduino est présentée sur **la Fig. III. 1**.

La source d'alimentation de ce testeur est facultative. En utilisant le PC en relation avec l'Arduino lors du transfert des données (programme et base des données des CIs à tester), aussi pour afficher les résultats de test détaillés ; ou bien une batterie de 9V ; des régulateurs de tension embarqués ont été utilisés pour réduire le niveau de tension de 5V à 2.5V utiliser pour ATmega2560 et d'autres périphériques. Le microcontrôleur ATmega2560 passe la condition de test pré-écrite pour tester IC, et l'envoie des résultats de test à l'écran LCD via le 12C. Le bouton poussoir est connecté sur une entrée analogique d'Arduino pour démarrer, arrêter et réinitialiser le test.

La prise ZIF est utilisée pour la mise en place des CIs à tester. L'EEPROM pour magasiner la base des données des CIs à tester, ici le script est une séquence d'entrées et de sorties logiques de définition à fournir pour chaque type de CI, si les résultats ne correspondent pas aux attentes, le script passera au prochain jeu possible.

Toutes les connexions sont effectuées par des résistances de 680 Ohms pour éviter la surcharge de l'Arduino ; plusieurs signaux sont définis sur la partie testée qui ne correspondent pas aux spécifications de la partie (par exemple, le niveau bas est défini comme entrée sur une broche qui agit comme une sortie élevée) car toutes les combinaisons possibles sont testées ou bien une création beaucoup de signaux "inférieurs aux spécifications" conduisant ainsi à des sorties aléatoires du CI testé. Pourtant, si le circuit intégré utilise des signaux testés, la sortie du circuit intégré testé est utilisable. [15]

Pour un essaye de démonstration nous avons près un exemple d'une porte IC 7432, c'est-à-dire **ET**. **(Voir la Fig. III.2).**Il a deux entrées et une sortie. Par convention les deux entrées sont élevées, seule la sortie est élevée. Nous donnons une entrée externe à deux broches et lisons la sortie de la broche de sortie à l'aide d'un microcontrôleur. Si la sortie souhaitée est obtenue, l'écran LCD affiche le test de réussite, sinon l'CI est défectueux ou introuvable dans la base des données. [20]

La façon de tester les CIs est, en utilisant leurs tables de vérité ou leurs tables fonctionnelles.

III.5.1. Eléments constitutifs de la conception logique :

La fonction de base du testeur des CIs logiques est de tester le fonctionnement logique des CIs comme décrit dans la table de vérité / table des fonctions de chaque porte. Le but est de vérifier les circuits intégrés sur place et d'afficher immédiatement les résultats des circuits intégrés bons

ou défectueux sur l'écran LCD. Le test est réalisé avec les CIs appartenant à la série de CI de porte logique de base. [20]

Les conditions de signal d'entrée nécessaires, sont appliquées aux entrées de la porte via le microcontrôleur, et la sortie de chaque porte est surveillée et comparée à la table de vérité, et en fonction de cette comparaison, le circuit intégré est testé s'il est bon ou défectueux c'est-à-dire le testeur IC détermine simplement quelles sont les portes utilisables et lesquelles sont défectueuses. Les tables de vérité sont stockées dans la base de données lors du codage du microcontrôleur.

Les portes logiques sont les fondamentaux éléments constitutifs de la conception logique. Ils sont utilisés en séquence pour développer des circuits séquentiels et combinatoires pour la solution de complexes problèmes.

La table de vérité des portes logiques fondamentales sont illustrées au-dessous :**(voir Tab.III. 1)**

La table de vérité logique pour chacune de ces autres portes logiques représente, la conception logique et le développement de circuits intégrés logiques peut être basique ou complexe ; et pour ça les portes logiques et l'algèbre booléenne sont très importantes.

Les tables de vérité de base sont référencées et mis en place dans l'environnement de programmation Arduino est effectuée selon les besoins.

Entrés		**Table de vérité de chaque sortie de porte logique fondamentale**					
A	**B**	**AND**	**NAND**	**OR**	**NOR**	**EX-OR**	**EX-NOR**
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Tab.III. 1. Table de vérité des portes logiques fondamentale.

III.5.2.Mise en œuvre du testeur fonctionnel par simulation virtuel :

Nous nous sommes contentés d'une simulation hypothétique, dans cette simulation nous avons utilisé le **PROTEUS**.

III.5.2.1. Le Proteus :

VSM Proteus Virtual System Modeling (VSM) est un logiciel connu dans le domaine de simulation répond à nos besoins, dont son libraire est riche. Grâce à la simulation, nous pouvons directement déboguer l'erreur dans le code source et corriger l'erreur immédiatement.

Le Proteus combine la simulation de circuits en mode mixte, des composants animés et des modèles de microprocesseurs pour faciliter la Co-simulation de conceptions complètes basées sur des microcontrôleurs.

Nous écrivons le code dans la fenêtre de code d'Arduino IDE. Après avoir écrit le code, nous faisons la sélection du type de carte, la vitesse de transmission, etc. ensuite la compilation du code et téléchargeons-le sur la carte. La compilation du programme dans l'Arduino IDE génère le fichier HEX du code source, qui doit être téléchargé dans le circuit Proteus VSM pour la simulation. [20]

Avec le Proteus nous pouvons faire une visualisation en **3D**.Cette option du Proteus nous donne une vision presque réel. **(Voir Fig. III. 3)**

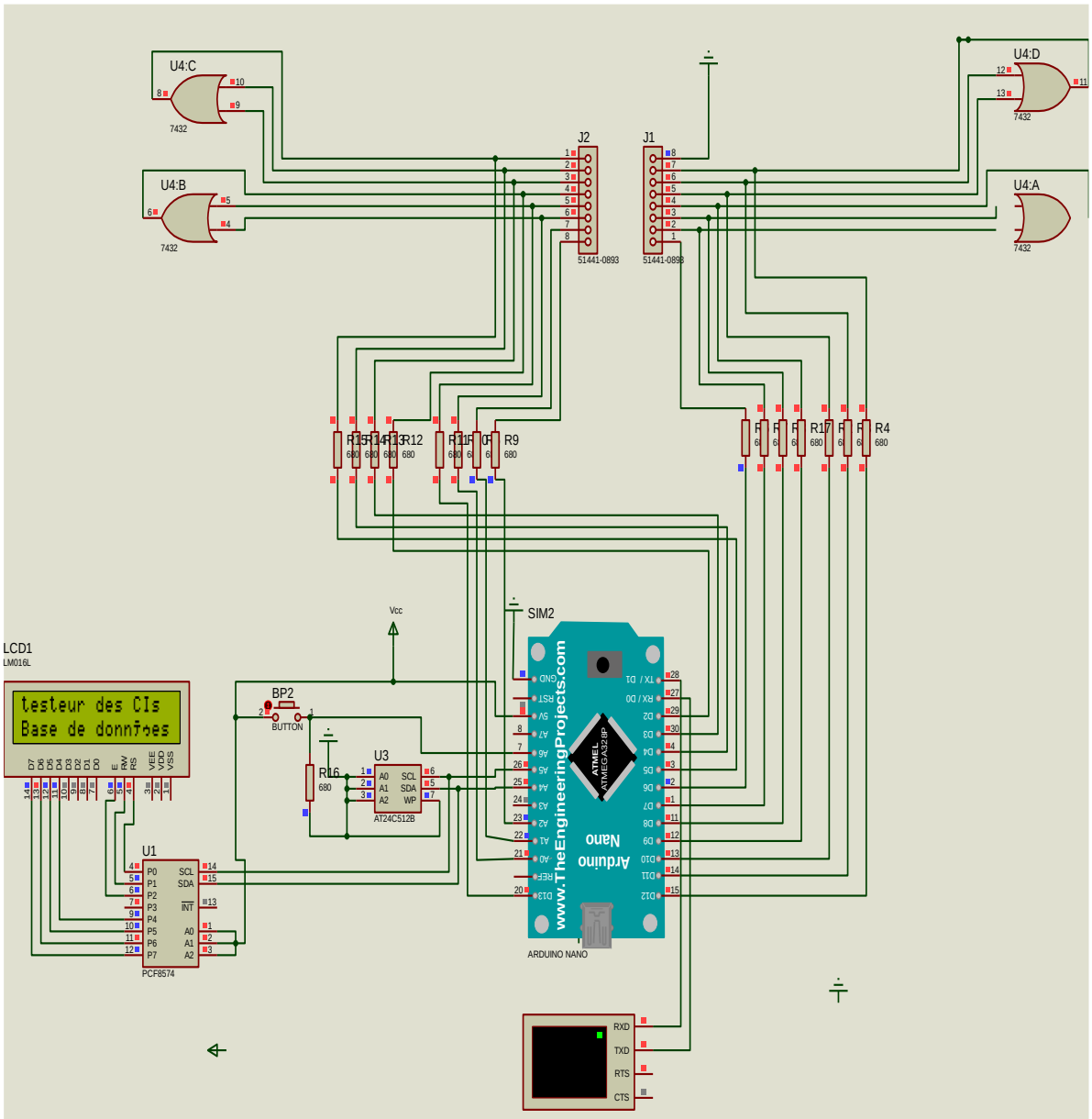

Fig. III. 2. Mise en œuvre le testeur en utilisant le CI 7432.

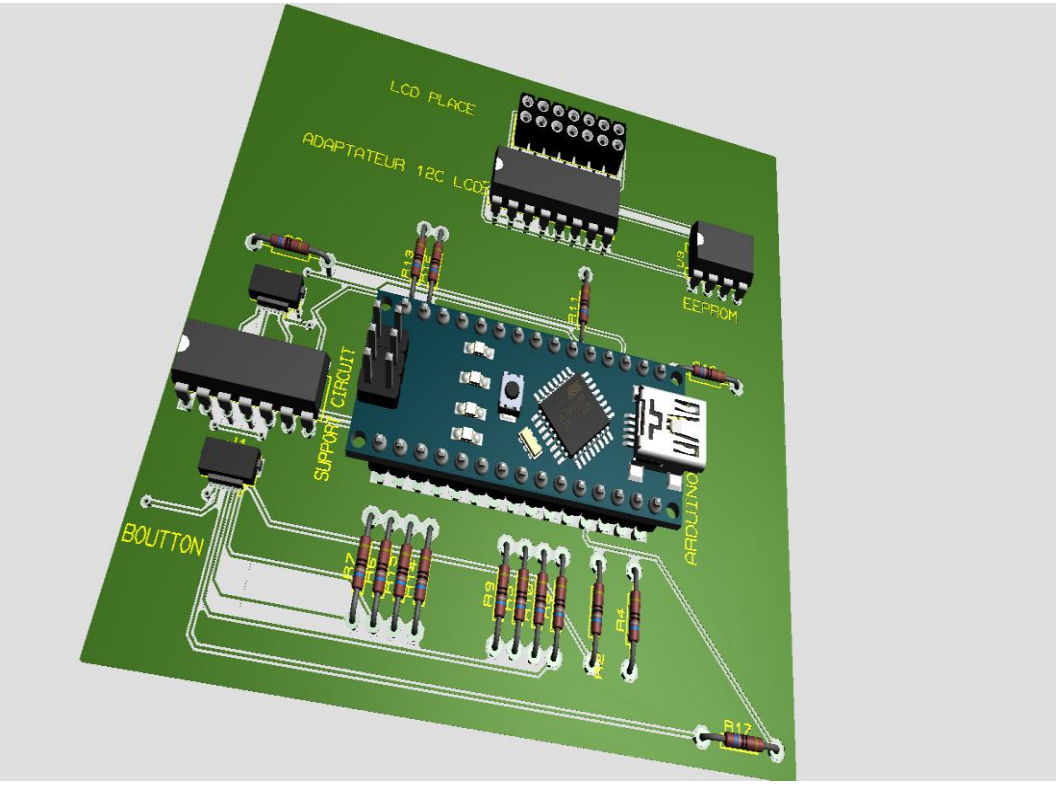

Fig. III. 3. Une visualisation en 3D du testeur fonctionnel.

III.5.2.2. Déroulement du programme :

Le script de test est reconfigurable parce qu'il est en texte clair. La syntaxe se trouve dans l'esquisse Arduino. Si d'autres modifications sont nécessaires en fonction des besoins de l'utilisateur, cela peut être fait très facilement.

III.5.2.3. Block diagramme du programme :

Le block diagramme du programme explique les étapes successives de déroulement du programme :

1. Initialisation de l'écran à l'aide du bouton poussoir.
2. La mise en place du CI sur le support ZIF.
3. Le transfert du programme écrit à l'Arduino a l'aide du PC.
4. La vérification dans la base des données de l'EEPROM l'existence du circuit intégré en question, et faire la comparaison avec le composant placé.
5. L'affichage du numéro référence du circuit intégré en question.
6. Le retour vers le début de la manipulation pour tester un autre circuit intégré.

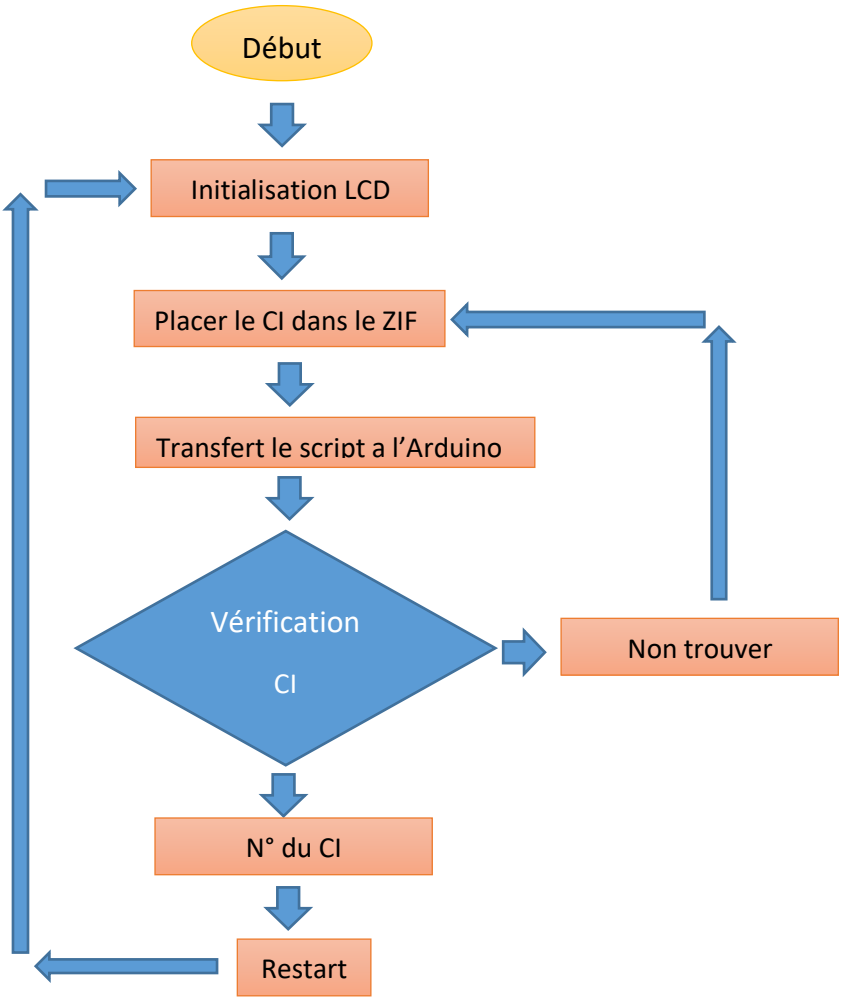

Fig. III. 4. Block diagramme du programme.

NB :

- Si nous avons mis en place un circuit intégré connu et existe dans la base des données de L'EEPROM et l'LCD affiche **Non trouver,** ça explique que le CI est d'effectuer ; sinon l'LCD affiche le numéro référence du circuit en question. **(voir Tab.III. 5)**
- Et si nous ignorons le numéro référence du CI et il existe dans la base des données de L'EEPROM, les résultats seront les mêmes que le cas précèdent. Mais ici nous avons connu le numéro référence du CI en question. **(voir Tab.III. 6)**
- Dans le cas ou, le circuit en question n'existe pas dans la base des données de L'EEPROM ; l'LCD affiche toujours **Non trouver. (Voir Tab.III. 5)**
- Ensuite nous redémarrons l'opération de test pour un autre CI en appuyant sur le bouton poussoir après que l'LCD affiche **restart. (Voir Tab.III. 7)**
  La procédure va être répéter jusqu'à la détection du circuit intégré ou non.

(Voir Tab.III. 8)

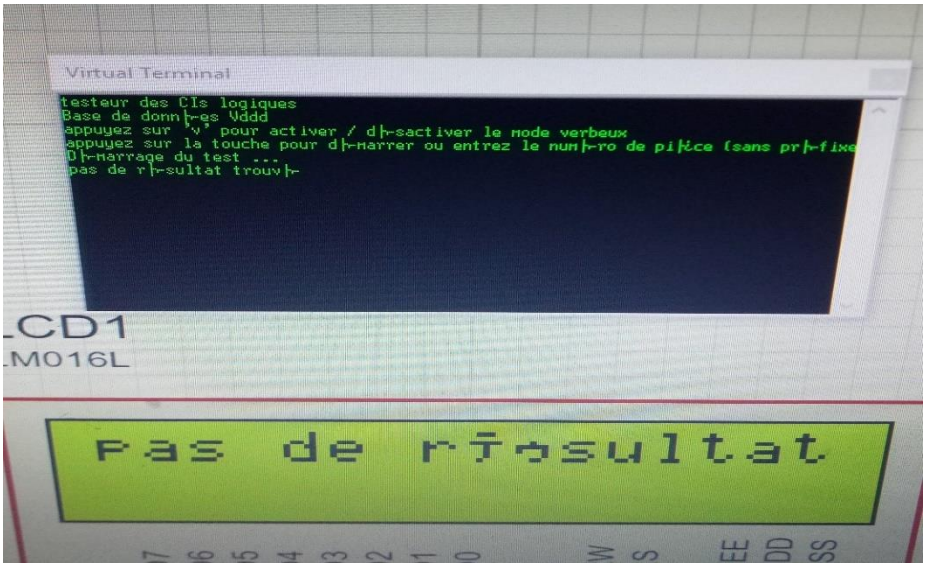

Fig. III. 5. Affichage pas de résultat

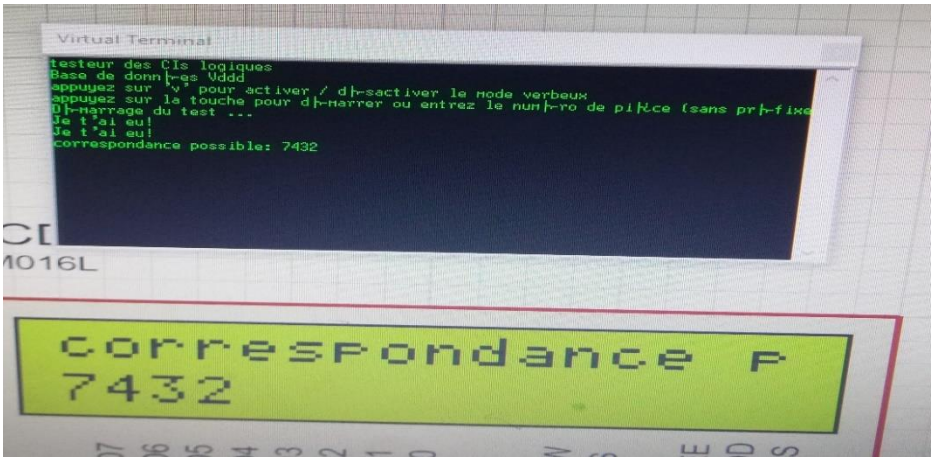

Fig. III. 6. Affichage correspondance possible 7432

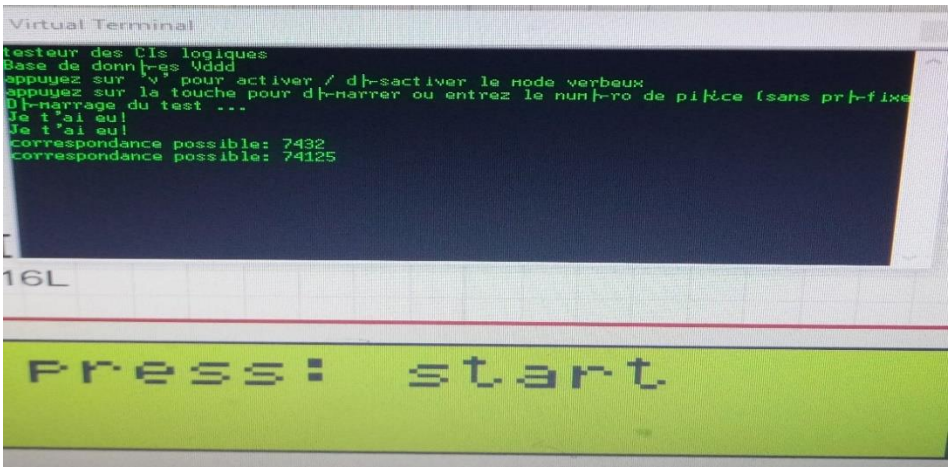

Fig. III. 7. Affichage presse : Start

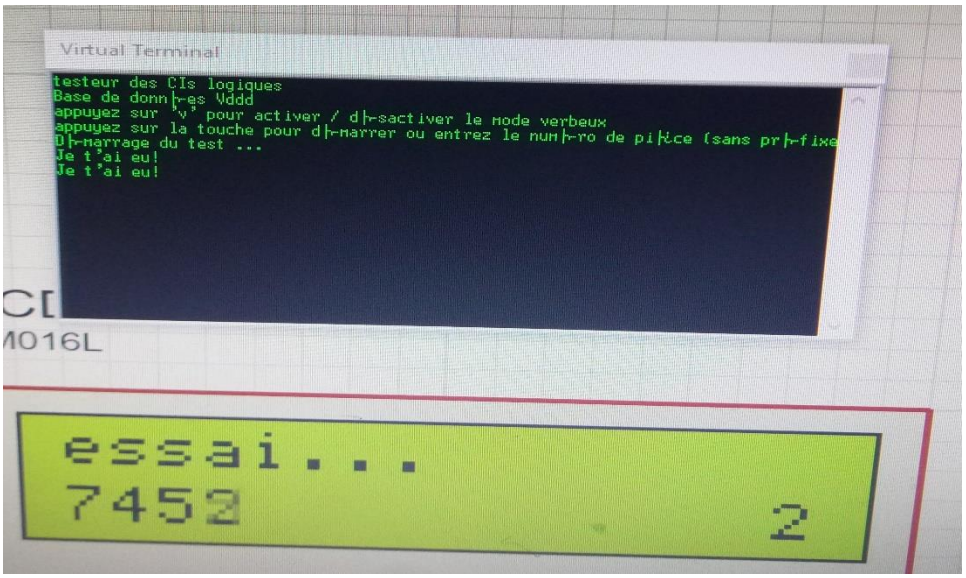

Fig. III. 8. Affichage test est en cours

III.5.2.4. Compréhension de programme :

En générale on peut dire que le programme est constitué de deux parties principales :

1. Représente la partie déclarative du début de programme, de tous les signaux entrants et sortants lors du test, ainsi que les instructions nécessaires qui dépendent aux résultats obtenues lors du test
2. . La deuxième partie représente une boucle par laquelle le programme lire le contenue de base des données de chaque circuit logique stockés dans L' EEPROM, et injecter les signaux d'entrées ensuite, faire les sorties comparés jusqu'à le balayage de tous les bases des données des circuits logiques mentionnées.

```
Testeur pour CMOS et TTl IC.
/ IC sera testé selon un fichier de test.
/ Syntaxe du fichier
/
$40161
16
0C10110G10XXXXXV
0C10110G11LLLLLV
1C10110G00HHLHLV
1000001G11HHLHLV
1C00001G11HHHLLV
1C00001G11HHHHHV
```

```
1C00001G11LLLLLV1
C00001G11LLLHLV1
100001G11LLHLLV
1000001G11LLHLLV
0000001G11LLLLLV
1C00001G11XXXXXV
/ $ Caractère de début avant le nom de la pièce (max 16 caractères après $)
/ Nombre de broches du CI
/ V = tension d'alimentation
/ G = GND (correction par le matériel)
/ 0/1 = 0 ou 1 logique comme entrée du CI
/ L / H = 0 ou 1 logique en sortie de l'IC
/ X = sortie IC non pertinente (le signal sera ignoré)
/ C = Horloge -> Déclenchement
/
/ routine de test générale :
/ 1. recherche le fichier jusqu'à ce qu'un $ soit trouvé. Stocker l'adresse (cette adresse est utilisée plus tard pour imprimer le nom de la pièce)
/ 2. définir le CI comme étant "OK"
/ 3. si la pièce est réglée OK "OK"
/ 4. régler tous les signaux d'entrée
/ 5. horloge de déclenchement, si nécessaire
/ 6. lire les sorties et comparer aux signaux attendus -> si différent : IC -> NOK
/ 7. si IC est toujours OK, incrémenter "où trouver l'adresse-description"
/ 8. trouver le prochain $
/ 9. si & -> fin
* /
```

III.6. Conclusion :

Dans ce chapitre nous avons entamé la réalisation du testeur des circuits logiques à base d'Arduino, mais malheureusement par simulation seulement à l'aide du logiciel ***Proteus*.** Cette simulation nous a donné une idée claire sur le fonctionnement du testeur. Les résultats ont était appréciant, c-à-dire l'objectif ou bien le but chercher a été atteint. Ensuite nous avons brièvement expliqué le déroulement du programme, et à la fin nous avons illustré les éléments constitutifs de la conception logique agitent dans les combinaisons et les comparaisons des données de base avec celle des circuits logiques testés.

Alors nous pouvons juger des maintenant que le testeur réalisé réduit le temps de test du circuit intégré car nous le testons automatiquement par rapport au test manuel où diverses connexions et applications d'entrées sont effectuées.

CONCLUSION GENERALE :

Lors de notre préparation de mémoire de fin d'étude, nous avons divisé le travail en trois chapitres, pour atteindre l'objectif de réaliser ou bien créer un appareil précis qui fonctionne avec la carte Arduino pour détecter l'état des circuits intégrés, sont-ils en bon état ou endommager ?

Dans le premier chapitre nous avons cités quelques différents types de testeurs des circuits intégrés disponibles sur le marché, et nous avons montré comment chaque fabricant a sa propre façon d'utiliser son appareil, ainsi que l'ensemble des éléments électroniques qui peuvent être vérifiés, y compris ces circuits intégrés.

Dans le deuxième chapitre, nous avons étudié tous les éléments impliqués dans la conception électronique de l'appareil à créer, tant au niveau des caractéristiques technologiques que de la raison du choix.

Quant au troisième chapitre, nous l'avons consacrée au côté applicatif, nous nous sommes appuyés uniquement sur la simulation

Les résultats obtenus grâce à la simulation étaient très impressionnants en termes de vitesse et de quantité, de détection, ainsi que l'important que nous ayons gagné beaucoup de temps.

En théorie, on peut dire que nous avons gagné un appareil efficace, léger et rapide et composé de quelques composants électroniques faciles à utiliser à tout moment, et par conséquent on peut dire que on a offert au département une solution rapide et efficace pour la vérification des états des lot des circuits intégrés avant leur utilisations dans les systèmes par les étudiants lors des travaux dirigés.

Bibliographie :

[1] https://www.directindustry.fr/prod/b-k-precision/product-18583-1500153.html

[2] https://img.directindustry.fr/images_di/photo-g/18583-7389679.jpg

[3] https://fr.aliexpress.com/item/32886776911.html)

[4] https://fr.aliexpress.com/item/32805317772.html

[5] https://french.alibaba.com/product-detail/digital-ic-tester-leaper-1a-141710695.html

[6]https://french.alibaba.com/product-detail/low-cost-digital-ic-tester-analog-ic-tester-universal-ic-teste-50029208536.html

[7] https://fr.aliexpress.com/item/32851616211.html

[8] http://fr.flossmanuals.net/Arduino/

[9] Simon. Monk, Arduino : les basses de la programmation. Pearson, (2013).P(192).

[10] https://www.gaudry.be/memoire-eeprom.html5

[11] https://fr.wikipedia.org/wiki/M%C3%A9moire_morte

[12] Fall. MIT. (2006). ROM, EPROM, AND EEPROM TECHNOLOGY

[13] http://kudelsko.free.fr/prog_pic_usb_V2/logiciel_microchip.htm

[14] http://electromag1.wifeo.com/lcd-i2c-expandeur-pcf8574.php

[15] https://trybotics.com/project/Arduino-IC-Tester-874

[16] https://www.instructables.com/id/Arduino-IC-Tester/

[17]https://www.google.com/search?source=univ&tbm=isch&q=schema+d%27une+carte+Arduino+nano&sa=X&ved=2ahUKEwitwO-5zOvrAhUjtnEKHWsQDeoQsAR6BAgKEAE

[18] https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur

[19] Projet de fin d'études, 2015, Réalisation d'une maison intelligente à base d'Arduino, El yahiaoui, Khadija, Boukoutaya, Abdel Adem, université Mohamed v

[20] International Journal of Trend in Research and Development, Digital IC Tester using Arduino, Volume 6(1).

[21]https://www.researchgate.net/publication/322159080_Digital_Logic_Gate_Simulation_using_Arduino_Microcontroller

[22] http://www.drviragopete.com/electronic-parts-testing-service.php

[23] https://www.drviragopete.com/resources/GUT%20IC%20Tester.jpg

[24]Yves, Mergy.ardouino programmation visuelle,Paris,Bod Books D emand .2016.P(257)**.**

[25]https://french.alibaba.com/product-detail/dict-01-ic-tester-digital-hand-held-50029311584.html

[26] https://www.bkprecision.com/products/component-testers/575A-digital-ic-tester.html

[27] https://www.gwinstek.com/en-global/products/detail/GUT-6600A

[28] Sid .katzen . (2007).the quintessential pic