

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat
Département: Electronique

MEMOIRE

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

Intitulé :
**Virtual Lab design for électronic course using
appropriate language**

Domaine : Sciences et Technologie
Filière : Electronique
Spécialité: Réseaux et télécommunication

Par :
BENNECIB AHLEM

DEVANT Le JURY

Président : LARBIA Prof UBM Annaba

Directeur de mémoire: FEZARLM Prof UBM Annaba

Examineurs: TOUMLS Prof UBM Annaba

Examineurs: BENSAOULA.S MCA UBM Annaba

Remerciements

Je remercie Dieu qui nous a donné naissance, santé et plénitude de nos sens.

Je remercie Dieu qui nous a guidées dans le chemin du savoir et de la science.

Je remercie Dieu qui nous a armées durant notre travail de la soif du savoir et de patience

Je remercie très vivement mon encadreur :

Mr.FEZARI Mohamed

Pour son encadrement, ces précieux conseils et orientations.

Je retiens à remercier également les membres du jury d'avoir accepté l'évaluation de ce travail.

Je remercie toute ma famille, surtout mes parents mes frères, ma sœurs et mon époux pour leurs soutiens et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Un grand merci à mes chères amies qui ont toujours été présents et fidèles.

...MERCI



DEDICACES

Je dédie ce modeste travail à :

L'Esprit de mon père

Ma chère mère

Mes frères : Seïfou, Hichem et Imed

Ma sœur : Ouïded

Mon époux : Walid

Toute ma famille et mes fidèles amies.

A tous ceux qui me connaissent de près ou de
loin.

Résumé

Le Virtual Lab est un laboratoire situé sur un espace virtuel accessible via Internet, permet aux étudiants d'explorer un sujet en comparant et en contrastant différents scénarios, de suspendre et de relancer l'application pour prendre des notes afin de bénéficier d'une expérience d'expérimentation pratique sur Internet dans lequel c'est un meilleur solution pour le grand nombre d'étudiants et le manque de matériel dans les universités. Pour ce but, dans ce travail nous avons réalisé une interface graphique pour un laboratoire virtuelle en électronique numérique qui vérifier le fonctionnement des portes logiques combinatoire de base selon leur table de vérité. C'est un système a été testé par une simulation sous Matlab et une appareille Arduino. Comme nous avons utilisé aussi le logiciel Processing dans lequel le système a été fait dans un environnement virtuel.

Mots clés : Virtual Lab, Arduino, Matlab, Processing, environnement virtuel, les portes logiques.

Abstract

The Virtual Lab is a lab located on a virtual space accessible via the Internet, allows students to explore a subject by comparing and contrasting different scenarios, to pause and restart the application to take notes in order to benefit from an experience practical experimentation on the Internet in which it is a better solution for the large number of students and the lack of equipment in the universities. For this purpose, in this work we have created a graphical interface for a virtual laboratory in digital electronics that checks the operation of basic combinatorial logic gates according to their truth table. This system has been tested by a simulation under Matlab and an Arduino device. As we also used the Processing software in which the system was done in a virtual environment.

Key words: Virtual Lab, Arduino, Matlab, Processing, virtual environment, logic gates

ملخص

المختبر الافتراضي هو مختبر موجود على مساحة افتراضية يمكن الوصول إليها عبر الإنترنت ، ويسمح للطلاب باستكشاف موضوع من خلال مقارنة السيناريوهات المختلفة، وإيقاف التطبيق مؤقتاً وإعادة تشغيله لتدوين ملاحظات من أجل الاستفادة من تجربة ما تجربة عملية على الإنترنت والتي تعد الحل الأنسب للعدد الكبير من الطلاب ونقص المعدات في الجامعات. لهذا الغرض، قمنا في هذا العمل بإنشاء واجهة رسومية لمختبر افتراضي في الإلكترونيات الرقمية يتحقق من تشغيل البوابات المنطقية التوافقية الأساسية وفقاً لجدول الحقيقة الخاص بهم. تم اختبار هذا النظام من خلال محاكاة برنامج ماتلاب وجهاز أروينو. كما استخدمنا أيضاً برنامج المعالجة الذي تم فيه تنفيذ النظام في بيئة افتراضية

الكلمات المفتاحية: المختبر الافتراضي ، اردوينو ، ماتلاب ، المعالجة ، البيئة الافتراضية ، البوابات المنطقية

Liste des abréviations

ETH	Eidgenössische Technische Hochschule
LMS	Learning Management System
3D	3 Dimensions
LED	Light-Emitting Diode
DOM	Document Object Model
RAM	Random Access Memory
GND	Ground
GPS	Global Positioning System
IDE	Integrated Development Environment
USB	Universal Serial Bus

Liste des figures

Chapitre I

Figure I.1 : oscilloscope virtuel.....	7
Figure I.2 : Compteur décimal basé sur 74LS160.....	8
Figure I.3 : Circuit de test diode dc.....	8

Chapitre II

Figure II.1 : les deux fenêtres principales de Processing.....	12
Figure II.2 : L'appareil Arduino.....	14
Figure II.3 : La carte Arduino Mega et le logiciel affiché sur l'ordinateur	15
Figure II.4 : une résistance.....	16
Figure II.5 : La LED.....	17

Chapitre III

Figure III.1 : Table de vérité et symbole du porte AND.....	19
Figure III.2 : symbole et table de vérité de la porte OR.....	20
Figure III.3 : symbole et table de vérité de la porte XOR.....	20
Figure III.4 : symbole et table de vérité de la porte NOT.....	21
Figure III.5 : symbole et table de vérité de la porte NAND.....	21
Figure III.6 : symbole et table de vérité de la porte NOR.....	22
Figure III.7 : Le déroulement du programme.....	23
Figure III.8 : l'organigramme principal	25
Figure III.9 : La principale fenêtre de menu.....	26
Figure III.10 : résultat de simulation d'exécution les portes logique.....	29
Figure III.11 : Le schéma du montage à réaliser avec Arduino.....	30
Figure III.12 : Le déroulement du programme IDE Arduino.....	30

Liste des tableaux

Chapitre 1

Tableau I.1 : Avantages pédagogiques prévus et les inconvénients des laboratoires virtuels.....	9
---	---

Table des matières

Introduction général	1
Chapitre I : généralités sur le Virtual Lab	
I.1 Introduction.....	4
I.2 Qu'est-ce qu'un Virtual Lab ?.....	4
I.2.1 L'historique du Virtual Lab.....	4
I.2.2 Objectifs.....	5
I.2.3 Conception d'un laboratoire virtuel.....	5
I.2.4 Les exigences	5
I.2.5 Organisation du laboratoire.....	6
I.3 Grandes applications du Virtual Labs	6
I.3.1 Virtual Labs pour l'électronique	7
I.3.2 Exemples des Virtual Labs on domaine d'électronique	7
I.3.3 Avantages pédagogiques prévus et les inconvénients des laboratoires virtuels	9
I.4 Problématique.....	9
Chapitre II : étude de la partie matérielle et logicielle du projet	
II.1 Introduction.....	11
II.2 C'est quoi un Processing ?	11
II.2.1 Proce55ing	11
II.2.2 Extensions et projets alternatifs	11
II.2.3 L'environnement	12
II.3 L'Arduino.....	13
II.3.1 Choix sur Arduino	14
II.3.2 Arduino UNO.....	15
II.3.3 Arduino.cc.....	16
II.3.4 Conception d'un programme Arduino	16

II.4 Les LEDs et Les résistances	16
II.5 Conclusion	17
Chapitre III : conception et programmation du système	
III.1 Introduction	19
III.2 Présentation des Portes logique à Simuler.....	19
III.2.1 Définition des Porte Logique	19
III.2.2 La porte logique AND	19
III.2.3 La porte logique OR	20
III.2.4 La porte logique XOR	20
III.2.5 La porte logique NOT.....	21
III.2.6 la porte logique NAND	21
III.2.7 La porte logique NOR	21
III.3 Le principe de fonctionnement du projet.....	22
III.3.1 La fonctionnement du programme.....	22
III.3.1.1 Structure globale du programme en Processing.....	22
III.3.1.2 La déroulement du programme.....	22
III.3.2.1 Présentation de l'organigramme IDE.....	23
III.3.2.2 Organigramme Processing.....	23
III.3.2.3 Explication du programme.....	25
III.3.3.1 Présentation du programme IDE	26
III.3.3.2 les résultats de la simulation.....	26
III.4.1 Présentation du programme IDE d'Arduino	30
III.4.2 Le déroulement du programme IDE Arduino.....	30
III.4.3 Explication du programme.....	31
III.5 Discussion des résultats.....	31
Conclusion générale.....	32

Introduction générale

Les distances physiques et le manque de ressources nous empêchent d'effectuer des expériences, en particulier lorsqu'elles impliquent des instruments sophistiqués. En outre, les bons enseignants sont toujours une ressource rare. Les cours basés sur le Web et sur la vidéo abordent le problème de l'enseignement dans une certaine mesure. Mener des expériences conjointes entre deux institutions participantes et partager des ressources coûteuses a toujours été un défi. Avec les technologies Internet et informatiques actuelles, les limitations ci-dessus ne peuvent plus empêcher les étudiants et les chercheurs d'améliorer leurs compétences et leurs connaissances.

De plus, dans un pays comme le nôtre, des instruments et des équipements coûteux doivent être partagés avec d'autres chercheurs dans la mesure du possible. Les expériences activées sur le Web peuvent être conçues pour une utilisation et une visualisation à distance, afin de stimuler la curiosité et l'innovation des étudiants. Cela faciliterait l'apprentissage des concepts de base et avancés par des expériences à distance.

Aujourd'hui, la plupart des équipements disposent d'une interface informatique pour le contrôle et le stockage des données. Il est possible de concevoir de bonnes expériences autour de certains de ces équipements afin d'améliorer l'apprentissage d'un élève. L'expérimentation sur Internet permet en outre d'utiliser des ressources, des connaissances, des logiciels et des données disponibles sur le Web, en plus d'encourager des expériences habiles simultanément effectuées à des points séparés dans l'espace (et éventuellement dans le temps).

Pour ce but, nous avons engagé à ce projet expérimental destiné à réaliser un système dans lequel c'est une interface graphique qui nous permet de commander et de contrôler les opérations sur les portes logiques de base d'une façon virtuelle.

Dans ce projet, trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur le Virtual Lab (laboratoire virtuel) ainsi que ses caractéristiques et son principe de fonctionnement.
- Le deuxième consiste des informations sur une grande catégorie de cartes d'interfaçage (Arduino) et un très bon logiciel de simulation (Processing).
- Le troisième consiste à réaliser une interface graphique programmée sous l'environnement (Processing) afin de simplifier la commande sur les portes logiques de base virtuellement lors que réel.

Introduction générale

Ce mémoire est organisé en quatre chapitres :

Dans le premier chapitre nous développons des informations générales concernant le Virtual Lab (laboratoire virtuel), leurs domaines d'applications, conceptions et leurs exigences.

Le deuxième chapitre consiste à une explication du logiciel Processing et l'appareil Arduino qu'ils sont des éléments principal du notre projet.

Le troisième chapitre est de regrouper suffisamment quelques concepts théoriques sur les portes logiques de base et nous avons expliqué chaque porte et ses caractéristiques.

Le dernier chapitre se compose de l'information sur le logiciel de programmation IDE Arduino, notre simulation dans l'environnement virtuel fourni par le logiciel Processing et l'interface graphique conçus de notre système.

Nous terminons avec une Conclusion Générale.

Chapitre I :
généralités sur le
Virtual Lab

Chapitre I : Généralités sur le Virtual lab

I.1 Introduction

Dans ce chapitre, nous allons décrire les caractéristiques générales du Virtual Lab et nous allons expliquer ses applications dans des différents domaines.

I.2 Qu'est-ce qu'un Virtual Lab ?

L'un des principaux problèmes rencontrés par les étudiants de toutes les universités qui s'occupent de la tâche liée au matériel est le suivant: le matériel n'est pas disponible ou le rapport étudiants / kits d'expériences par lot n'est pas approprié.

La solution est donc la virtualisation. De nos jours, de nombreux étudiants utilisent actuellement **les laboratoires en ligne** dans le cadre du projet de **laboratoires virtuels**. Beaucoup de ces laboratoires sont accessibles en dehors des heures normales de laboratoire. Par conséquent, en effectuant un laboratoire virtuel avant un laboratoire physique, il aide les étudiants à tirer les leçons de leurs erreurs et fait enfin leurs expériences correctement dans leurs laboratoires physiques. Ainsi, nous nous concentrons sur les expériences effectuées conformément au programme universitaire en électronique de toutes les années.

Définition : Virtual Lab est un laboratoire situé sur un espace virtuel accessible via Internet.

C'est un site Web ou logiciel d'apprentissage interactif basé sur la simulation de phénomènes réels. Il permet aux étudiants d'explorer un sujet en comparant et en contrastant différents scénarios, de suspendre et de relancer l'application pour prendre des notes et de prendre des notes, afin de bénéficier d'une expérience d'expérimentation pratique sur Internet.

Une expérience est mise en place dans le laboratoire distant pour que les utilisateurs puissent y accéder via Internet à tout moment et en tout lieu. Comparé au laboratoire traditionnel, le laboratoire virtuel est particulièrement utile lorsque certaines expériences nécessitent des équipements pouvant avoir des effets nocifs sur l'homme. Une autre signification du laboratoire virtuel est d'implémenter le laboratoire au moyen d'une simulation logicielle.

I.2.1 L'historique du Virtual Lab

En 1977, la première version du laboratoire virtuel, intitulée **Laboratoire virtuel de physiologie**, a été présentée. À cette époque, l'accent était mis sur la mise au point de conditions technologiques préalables à la recherche en physiologie au XIXe siècle. Par conséquent, une base de données avec des textes et des images pertinents a été créée.

En 1998, le concept encore utilisé aujourd'hui a été créé après une série de modifications, suivi de la publication d'un CD-ROM en 1999. À cette époque, l'accent était mis sur la physiologie aux sciences de la vie en général, ainsi arts et littérature . Le projet ayant été étendu d'une base de données unique à une plate-forme de recherche historiographique, il a été présenté à la conférence. Utilisation du *World Wide Web* pour la recherche historique en science et technologie organisée par la Fondation Alfred P. Sloan de l'Université de Stanford . En 2000, le projet a été intégré au projet de recherche *The Experimentalization of*

Chapitre I : Généralités sur le Virtual lab

Life, financé par la Fondation Volkswagen. Cela a été suivi d'une autre présentation à la conférence *Virtual Research? L'impact des nouvelles technologies sur les pratiques scientifiques* à l'ETH de Zurich. En 2002, la première version du laboratoire virtuel a été mise en ligne. Depuis 2008, le laboratoire virtuel est répertorié comme une revue sous le numéro ISSN 1866-4784.

I.2.2 Objectifs

Le projet Virtual Labs a été initié pour résoudre le problème fondamental du manque de bonnes installations de laboratoire dans de nombreux collèges d'ingénierie. Les principaux objectifs du projet Virtual Labs sont les suivants:

- Fournir accès à distance aux laboratoires dans diverses disciplines de la science et de l'ingénierie.
- Pour répondre aux grand nombre d'étudiants au niveau de premier cycle licence électronique, physique et chimie ainsi que de la recherche universitaire.
- Pour permettre aux étudiants d'apprendre à leur propre rythme, et d'éveiller leur curiosité.
- Pour fournir un système complet de gestion de l'apprentissage (LMS) qui comprend en ligne laboratoire-manuels, ressources web, des conférences vidéo, des démonstrations animées, des procédures étape par étape, la sortie de l'échantillon, l'auto-évaluation et une banque de questions

I.2.3 Conception d'un laboratoire virtuel

Généralement, un laboratoire virtuel peut être divisé en deux composants: les programmes d'application, les fichiers de données et les descriptions textuelles constituant les expériences; et le support système sous-jacent et les utilitaires fournissant le cadre sur lequel sont construites ces expériences dépendantes du domaine.

I.2.4 Les exigences

La liste suivante spécifie les fonctionnalités considérées essentielles au fonctionnement du laboratoire.

- Organisation cohérente du laboratoire. Dans l'environnement de laboratoire, des expériences sont exécutées en appliquant des outils aux objets. Un objet est constitué de fichiers regroupés afin de pouvoir les récupérer facilement. Le format des objets doit être suffisamment normalisé pour permettre la mise en œuvre directe d'opérations courantes telles que la sauvegarde et la suppression d'objets.
- Héritage de fonctionnalités. Il arrive souvent que plusieurs objets ne diffèrent que par les détails. Le mécanisme de l'héritage devrait être utilisé pour manipuler et stocker efficacement de tels objets.

Chapitre I : Généralités sur le Virtual lab

- Manipulation interactive d'objets. Le laboratoire devrait fournir un ensemble d'outils d'utilisation générale permettant de manipuler les paramètres des objets. Par exemple, les objets peuvent être modifiés à l'aide de panneaux de contrôle ou en modifiant des champs spécifiques dans une description textuelle d'une expérience.
- Flexibilité dans la conduite des expériences. L'utilisateur doit pouvoir appliquer des outils (programmes) à des objets (fichiers de données) de manière dynamique pendant le déroulement d'une expérience. Cela peut être comparé à une association statique établie lorsque l'objet est incorporé au système.
- Guidage par le laboratoire. Le laboratoire devrait inclure un système hypertexte qui impose une organisation logique à l'ensemble d'objets, fournit une description textuelle des expériences et permet de parcourir les expériences de nombreuses manières. Des expériences spécifiques doivent être invoquées automatiquement lorsque le texte correspondant est sélectionné, afin de faciliter les démonstrations et d'assister un utilisateur novice.

I.2.5 Organisation du laboratoire

Le cadre du laboratoire virtuel comprend:

- une structure de fichier permettant de représenter des unités expérimentales (objets).
- des utilitaires pour manipuler ces objets.
- un système hypertexte qui guide à travers le laboratoire.
- Les fichiers de données comprennent notre connaissance d'un modèle particulier.
- Un fichier de spécification définit les fichiers de données qui constituent l'objet et les outils qui leur sont applicables.
- Un répertoire d'extensions répertorie les objets qui héritent de certaines fonctionnalités de l'objet actuel.

I.3 Grandes applications du Virtual Labs

Les projets de laboratoires virtuels couvrent presque tous les domaines de la science et de l'ingénierie. Les grandes régions du Virtual Labs sont:

- Génie électronique et des communications
- Génie informatique
- électrique appliquée
- Génie mécanique
- physique
- Génie chimique
- Génie biomédical et biotechnologique

Plus d'une centaine de laboratoires virtuels sont actuellement utilisés par des étudiants et des enseignants de diverses disciplines dans le monde.

I.3.1 Virtual Labs pour l'électronique

Les laboratoires virtuels électriques et électroniques sont principalement utilisés par les débutants en spécialités électriques dans les universités. Par conséquent, l'interface de l'application et le contenu expérimental sont simples et faciles à utiliser. il y a beaucoup d'instructions expérimentales.

Pour cette raison, il est nécessaire d'appliquer des laboratoires éloignés dans les cours et les expériences afin d'éviter une diminution de la motivation des élèves, un sentiment de solitude, etc.

I.3.2 Exemples des Virtual Labs on domaine d'électronique

- Interface oscilloscope virtuel produite par LabVIEW8.6. L'auteur sans spécialité en beaux-arts n'a eu recours qu'à un contrôle en face avant de LabVIEW et l'a achevé en moins d'une heure.

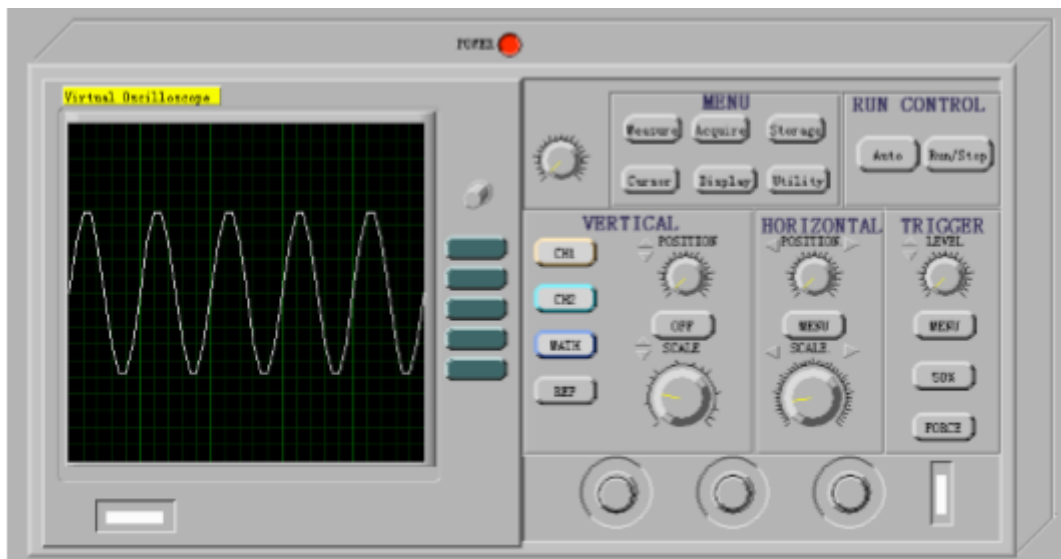


Figure I.1 : oscilloscope virtuel

- Sur la base du compteur décimal 74LS160, l'interface d'opération est illustrée à la (figure1.5). Par le biais de compteurs décimaux virtuels, les étudiants peuvent saisir des boutons de commande de fonctionnement et comprendre des concepts tels que la suppression et le réglage du zéro, qui sont difficiles pour les débutants.

Chapitre I : Généralités sur le Virtual lab

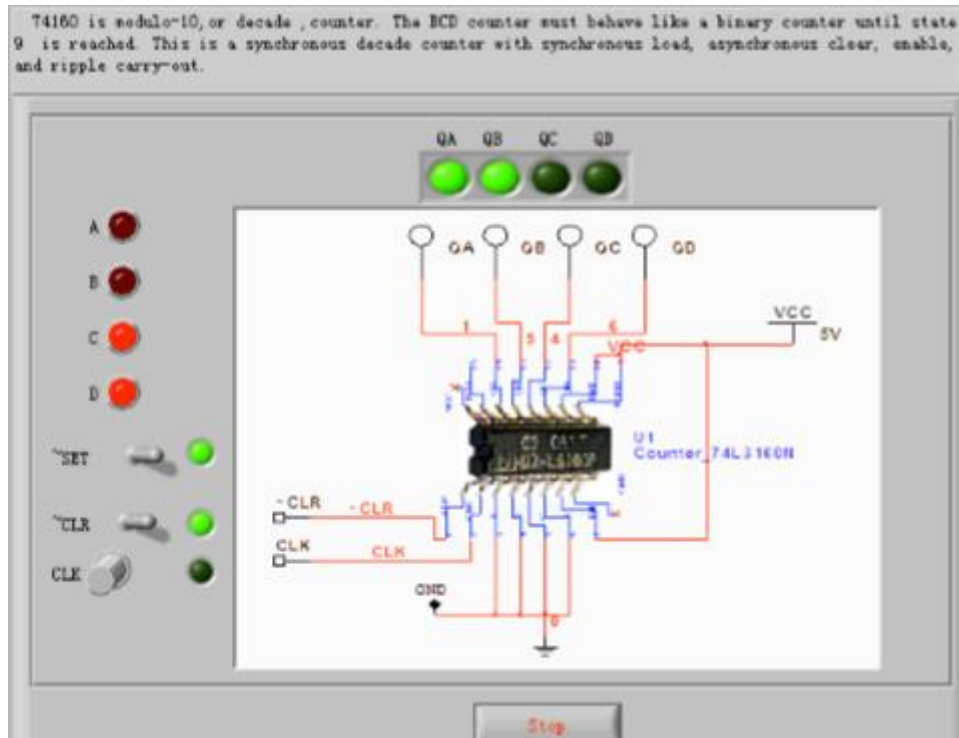


Figure I.2 : Compteur décimal basé sur 74LS160

- L'interface d'exploitation de la proposition de conception d'expérience virtuelle du circuit d'essai à diode DC est illustrée à la figure 1.6.

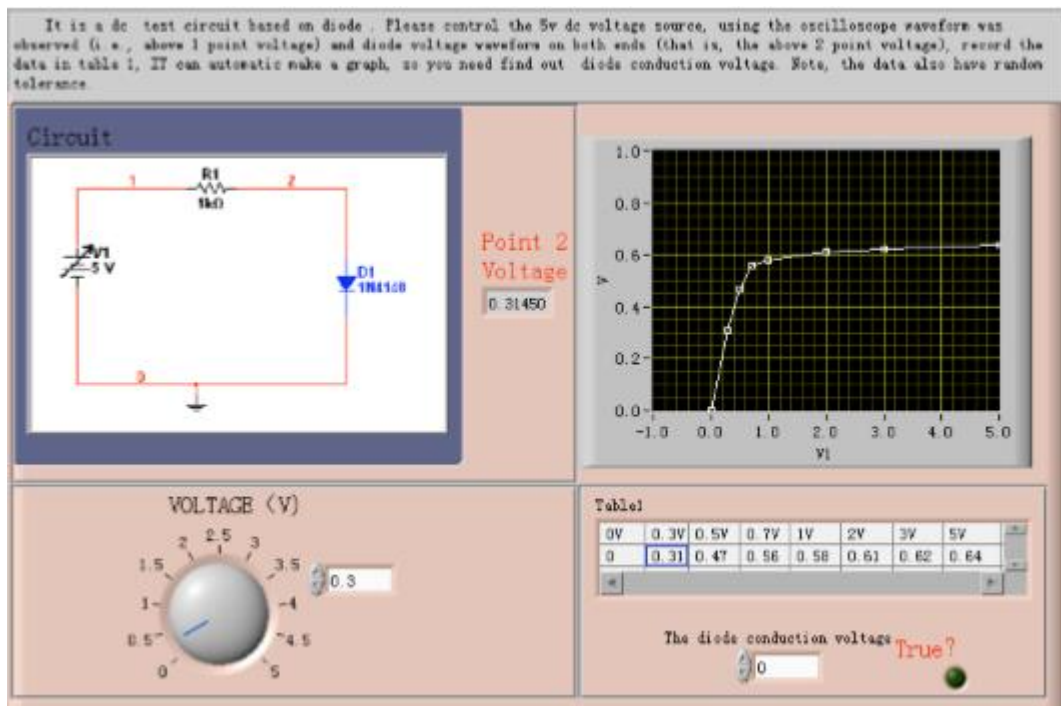


Figure I.3 : Circuit de test diode dc

Chapitre I : Généralités sur le Virtual lab

I.3.3 Avantages pédagogiques prévus et les inconvénients des laboratoires virtuels

avantages des laboratoires virtuels	Les inconvénients des laboratoires virtuels
Permet une expérience de laboratoire à tout moment et en tout lieu.	Dissuadent élèves d'apprendre les instruments physiques et dispositifs réels
Permet aux étudiants plus de possibilités de pratiquer des expériences, en particulier pour ceux qui ne peuvent pas être faciles à reproduire en raison de ressources, de temps et les questions de sécurité	L'accès à distance décourage la collaboration directe et l'interaction entre les élèves et les enseignants
Fournit un environnement d'atelier en toute sécurité, sans la nécessité d'une surveillance	Risque accru de plagiat dans l'évaluation
Améliore l'enthousiasme des élèves pour l'apprentissage grâce à l'interactivité	Physiques, les compétences pratiques qui sont attendues d'un ingénieur ne sont pas aiguisées
Accroître l'alphabétisation informatique des étudiants	
De nombreuses industries utilisent des logiciels de simulation pour les tests et le développement	
Contactez-heures de laboratoire sont rares. Les élèves peuvent utiliser des laboratoires virtuels pour renforcer les concepts théoriques qu'ils apprennent en classe	
Plus rentable, en particulier pour les circuits complexes qui peuvent nécessiter un certain nombre d'essais et d'erreurs	
Peut fournir assistance et d'autres renseignements sur les élèves	
commentaires en ligne et de l'évaluation peuvent être facilement disponibles	

Tableau I.1 : Avantages pédagogiques prévus et les inconvénients des laboratoires virtuels

I.4 Problématique

Vue le grand nombre d'étudiants inscrit en licence dans les différents spécialités science et technique, et vue le manque du matériel hardware et la disponibilité des laboratoires de TP pour réaliser des travaux pratiques sur les composants logique combinatoires et séquentiel. La solution consiste à utiliser les laboratoires virtuels et des remotes- labs.

Chapitre II :
Étude de la partie
matérielle et logicielle
du projet

II.1 Introduction

Dans ce chapitre nous allons décrire les matériels et les logiciels utilisés et leur fonctionnement pour concevoir une interface graphique pour réaliser les TP de logique combinatoire et séquentiel. Grâce à cette partie de notre projet nous avons augmenté nos connaissances dans les différents aspects de l'électronique.

Nous avons vu l'utilité du logiciel Processing son environnement, ses fenêtres et comment ça marche. Comme nous allons connaître l'appareil Arduino ses composants, les différents modèles de cet appareil.

II.2 C'est quoi un Processing ?

Processing est un environnement de programmation open-source. Il est employé par des étudiants, des artistes, des créateurs, des chercheurs, et des amateurs. Ce langage basique offre de très nombreuses possibilités grâce à sa riche librairie disponible. Le logiciel fonctionne sur Macintosh, sous Windows et sous Linux, en effet il est basé sur la plate-forme Java. Processing offre la possibilité de faire de la 3D, des jeux en réseau, des effets de son et lumière mais aussi bien d'autres possibilités.

II.2.1 Proce55ing

L'ancien nom de Processing, « Proce55ing », a été forgé ainsi car le nom de domaine Processing.org était pris et le site officiel a d'abord dû exister à l'adresse Proce55ing.org. À présent, les auteurs de Processing disposent du nom de domaine Processing.org, ils n'utilisent plus l'ancienne forme du nom et en déconseillent l'usage. Cependant, de nombreuses personnes abrègent le nom en p53.

II.2.2 Extensions et projets alternatifs

Le projet Arduino, qui se base sur Processing, comme son « ancêtre », Wiring4, permet la manipulation de circuits électroniques extérieurs pour interfacier des capteurs ou des appareils électroniques divers (servomoteurs, leds...). Un autre environnement logiciel, Fritzing, a été conçu pour aider les utilisateurs au prototypage de matériel destiné à être utilisé avec Arduino et Processing. Le projet Mobile Processing, comme son nom l'indique, a l'ambition de porter Processing sur téléphones portables. Enfin, Processing.js est une implémentation JavaScript de Processing.

Depuis sa version 1.5.1, Processing peut produire des applications pour plate-forme Android et accéder aux capteurs spécifiques aux dispositifs mobiles (accéléromètre, GPS, etc.). Depuis sa version 2.0, on peut aussi exporter directement du contenu au format HTML5/Javascript. Un projet voisin, P5.js, permet d'implémenter le langage directement à des pages web et de maîtriser le DOM.

De nombreuses bibliothèques externes peuvent être ajoutées à Processing pour accéder à des ressources telles que la vidéo, le son, la reconnaissance faciale, la caméra Kinect, les moteurs physiques, etc.

II.2.3 L'environnement

L'interface de Processing peut se découper en trois zones :

- La zone d'outils permettant d'exécuter ou stopper le programme et donnant accès au menu.
- L'éditeur de texte où sera écrit le code.
- La zone de console affichant les messages et les erreurs.



Figure II.1 : les deux fenêtres principales de Processing

Trois autres fenêtres vont également nous être utiles régulièrement à savoir :

- La fenêtre **Exemple** regroupant les exemples fournis avec Processing ou importés avec les bibliothèques
- La fenêtre **Sketchbook** regroupant l'ensemble de nos créations et pouvant être paramétrée dans la fenêtre **Préférences**
- La fenêtre **Préférences** regroupant l'ensemble des préférences du logiciel à savoir

Chapitre II : Étude de la partie matérielle et logicielle du projet

- L'emplacement du dossier sketchbook
- La langue
- Police utilisée dans l'éditeur et la console ainsi que leurs tailles respectives
- La couleur d'arrière-plan du mode présentation
- Le lissage des polices
- La saisie des caractères non latins
- La détection des erreurs et affichage des avertissements
- Utilisation de l'auto-complétion
- Suggestion des imports
- L'augmentation de la mémoire vive allouée (par défaut 256Mb)
- Effacer le dossier précédent lors de l'export (évitant ainsi les nombreux fichiers issus de multiples exports)
- Vérification des mises à jour au démarrage
- Taille par défaut du fullscreen
- Ouvrir tous les fichiers .pde avec processing

Dans la fenêtre préférences nous vérifierons que les éléments suivant sont actif :

- Détecter les erreurs en continu
- Afficher les avertissements
- Auto complétion du code ctrl+espace
- Augmenter la mémoire vive disponible (augmenter la mémoire en fonction de la quantité de RAM dont vous disposez)

II.3 L'Arduino

Arduino est une plate-forme libre de création d'objets électroniques à des fins artistiques, éducatives ou de recherche via la conception de prototypes. Elle repose sur l'utilisation d'un circuit électronique (un mini-ordinateur, appelé également microcontrôleur) comportant des entrées et sorties (des ports) sur lesquelles on peut brancher différents appareils :

- Côté entrées, des capteurs, appareils qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence, le contact via un bouton poussoir, etc.,

- Côté sorti, des acteurs, des appareils qui agissent sur le monde physique, telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

Arduino comporte également un environnement de développement logiciel calqué sur celui-ci de Processing, qui permet de programmer le circuit électronique. Arduino étant un projet dérivé de Processing, il apparaît donc comme une ouverture logique à la création interactive. Une fois que vous maîtrisez Processing, Arduino vous sera déjà familier.



Figure II.2 L'appareil Arduino

II.3.1 Choix sur Arduino

Il y a de nombreux microcontrôleurs et de nombreuses plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, et beaucoup d'autres qui offrent des fonctionnalités comparables. Tous ces outils prennent en charge les détails compliqués de la programmation des microcontrôleurs et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs intéressés par les autres systèmes :

- Pas cher
- Multiplateforme
- Un environnement de programmation clair et simple
- Logiciel Open Source et extensible « Arduino.cc »
- Matériel Open source et extensible



Figure II.3 : La carte Arduino Mega et le logiciel affiché sur l'ordinateur

II.3.2 Arduino UNO :

Dans notre projet nous utilisons la carte Arduino Uno. Le schéma qui suit montre les différents composants de cette carte:

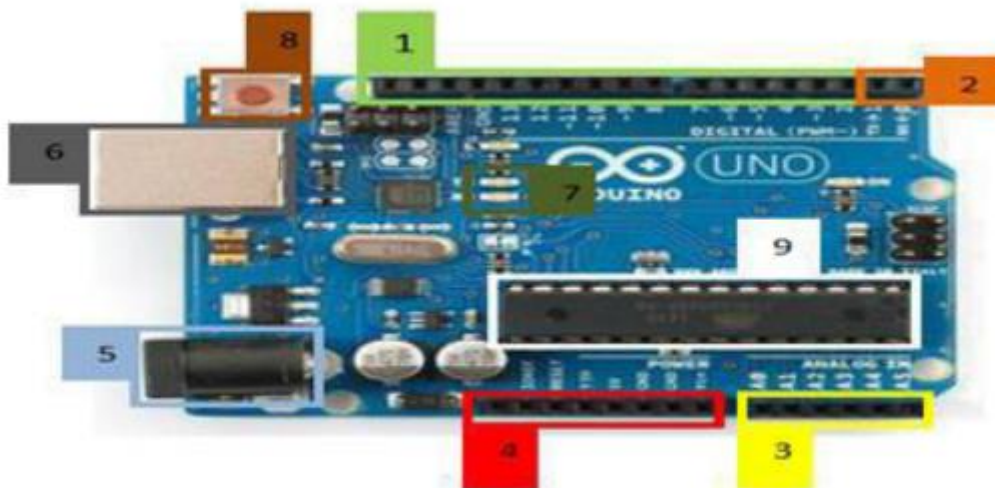


Figure II.3 : Les composants d'Arduino UNO

- 1 : Entrées/Sorties digitales (14 pin)
- 2 : Pin de communication
- 3 : Entrées/Sorties Analogique (peuvent aussi servir comme E/S digitales, 6 pin)
- 4 : Broches d'alimentation pour le montage (5V, 3,3V, GND...)

Chapitre II : Étude de la partie matérielle et logicielle du projet

5 : Entrée d'alimentation externe pour la carte

6 : Port USB pour la communication entre le PC et la carte

7 : LED indiquant la communication avec l'ordinateur (Tx, Rx)

8 : Botton RESET

9 : Microcontrôleur ATmega328

II.3.3 Arduino.cc

Arduino.cc est un logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information du ce programme appelé IDE Arduino.

II.3.4 Conception d'un programme Arduino

- **Configuration des entrées et des sorties void setup ()**

La fonction « Setup » contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).

- **Programmation des interactions void loop ()**

La fonction « Loop », elle est exécutée en boucle après l'exécution de la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset). Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation. En effet, si l'on omettrait cette boucle, à la fin du code produit, il sera impossible de reprendre la main sur la carte Arduino qui exécuterait alors du code aléatoire.

II.4 Les LEDs et Les résistances

Une résistance est un composant électrique ou électronique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (mesurée en ohms) à la circulation du courant électrique.



Figure II.4 : une résistance

Chapitre II : Étude de la partie matérielle et logicielle du projet

La LED est une diode électroluminescente, abrégée sous les sigles DEL ou LED, est un composant électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.



Figure II.5 : La LED

Dans ce projet on a besoin de 6 LED et 6 résistances de 270 Ohms pour représenter les sorties des 6 portes logique simulées.

Calcule des valeurs de résistances:

$$V_{led} = 2 \text{ Volts}$$

$$V_e = 5 \text{ volts} = R \cdot i + V_{led}; i = 10 \text{ mA}$$

$$R = (V_e - V_{led}) / i = (5 - 2) / 10 = 300 \text{ Ohm}$$

D'où nous utilisons une valeur proche **R=270Ohm**

II.6 Conclusion

Dans ce chapitre nous avons donné quelques concepts théoriques et nous avons expliqué le logiciel Processing et l'appareil Arduino qu'ils sont des éléments principal du notre projet. La bonne configuration des programmes dans le Processing est essentielle pour le fonctionnement global de notre travail.

La partie de programmation sera expliquée dans le prochain chapitre.

Chapitre III : conception et programmations du système

III.1 Introduction

Le système que nous allons réaliser consiste à deux parties essentielles hardware et software ; La partie hardware est constituée seulement d'un Arduino pour le tester le travail mais dans la partie software nous allons développer les programmes impliqués lors de la réalisation du système.

Nous allons créer une interface graphique pour laboratoire virtuelle en électronique numérique pour vérifier le fonctionnement des portes logiques. C'est pour ça dans ce chapitre on va aussi faire définir les portes logiques de base, leur principe, ses fonctions mathématique et ses table de vérité.

III.2 Présentation des Portes logique à Simuler

III.2.1 Définition des Porte Logique

Une porte logique est un composant élémentaire d'un circuit numérique. Il existe sept portes logiques de base (AND, OR, XOR, NOT, NAND, NOR et XNOR).

La majorité des portes logiques disposent de deux entrées et d'une sortie. A tout moment, chaque borne (ou « broche ») affiche un des deux états binaires *bas* (0) ou *haut*(1), représentés par des niveaux de tension électrique distincts.

III.2.2 La porte logique AND

La fonction ET (*AND* en anglais) est un opérateur logique de l'algèbre de Boole. À deux opérands, qui peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérands ont la valeur VRAI.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Table de vérité du porte AND



Figure III.1 : Table de vérité et symbole du porte AND

III.2.3 La porte logique OR

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



Figure III.2 : symbole et table de vérité de la porte OR

III.2.4 La porte logique XOR

La fonction OU exclusif, souvent appelée XOR (eXclusive OR) ou disjonction exclusive, ou encore \vee en algèbre relationnelle, est un opérateur logique de l'algèbre de Boole. À deux opérandes, qui peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont des valeurs distinctes.

Cet opérateur est très utilisé en électronique, en informatique, et aussi en cryptographie du fait de ses propriétés intéressantes. Son symbole est traditionnellement un signe plus dans un cercle : « \oplus ».

Définition

Appelons A et B les deux opérandes considérés. Convenons de représenter leur valeur ainsi :

- 1 = VRAI
- 0 = FAUX

L'opérateur XOR est défini par sa table de vérité, qui indique pour toutes les valeurs possibles de A et B la valeur du résultat R :

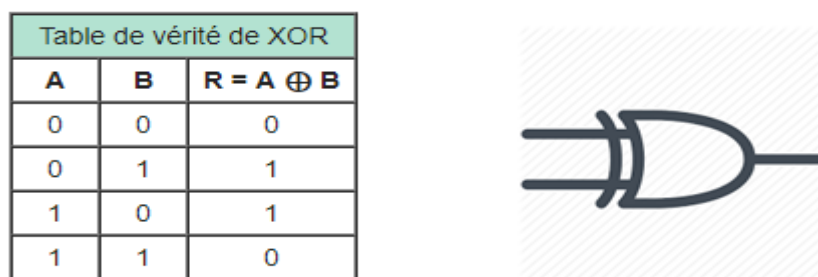


Figure III.3 : symbole et table de vérité de la porte XOR

Chapitre III : conception et programmation du système

Comme on peut le voir, l'opérateur logique XOR, ou OU exclusif, peut se définir par la phrase suivante :

*Le résultat est VRAI si un et un seul des opérandes A et B est VRAI

III.2.5 La porte logique NOT

La fonction NON (*NOT* en anglais) est un opérateur logique de l'algèbre de Boole et exprime un « état » en fonction de conditions. À un opérande, qui peut avoir la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur inverse de celle de l'opérande. Le circuit intégré 7404 intègre six portes logiques inverseuses du type NON.

Equation : $L = \bar{a}$

Entrée	Sortie
a	L
0	1
1	0

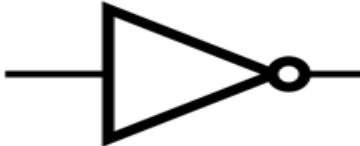


Figure III.4 : symbole et table de vérité de la porte NOT

III.2.6 la porte logique NAND

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

Entrées		Sortie
a	b	L
0	0	1
0	1	1
1	0	1
1	1	0




Figure III.5 : symbole et table de vérité de la porte NAND

III.2.7 La porte logique NOR

La fonction NON-OU (*NOR* en anglais) est un opérateur logique de l'algèbre de Boole. À deux opérandes, qui peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont la valeur FAUX.

Cette fonction logique correspond aux mots français *ni... ni*, car la phrase *ni A ni B* est vraie si et seulement si les phrases A et B sont toutes les deux fausses !

Entrées		Sortie
a	b	L
0	0	1
0	1	0
1	0	0
1	1	0


Le symbole de la porte NOR est représenté par une forme en D inversée avec une petite bulle à l'extrémité de sa sortie unique. Deux entrées sont indiquées par des lignes horizontales à gauche.

Figure III.6 : symbole et table de vérité de la porte NOR

III.3 Le principe de fonctionnement du projet

Le laboratoire virtuel c'est un système complet d'apprentissage interactif pour implémenter le laboratoire réel basé sur la simulation des phénomènes réels situé dans un espace virtuel accessible via internet ou un logiciel (comme Processing) par des interfaces graphiques que nous permettront de réaliser plusieurs projets d'après une simulation des programmes.

III.3.1 Le fonctionnement du programme

III.3.1.1 Structure globale du programme en Processing

Ce programme comprend :

- Un entête déclaratif
- Une partie « configuration » qui ne sera exécutée qu'une fois (fonction `setup()`) nous avons l'utilisée pour définir les propriétés initiales de l'environnement de notre programme telles que la taille de l'écran et pour charger des supports tels que les images et les polices.
- Une partie constituée d'une boucle sans fin que le programme répètera à l'infini (fonction `draw()`) : c'est le cœur du programme.
- Une partie pour finir dans laquelle la fonction `mousePressed()` est appelée une fois après chaque pression sur un bouton de la souris. La variable `mouseButton` peut être utilisée pour déterminer le bouton sur lequel nous avons appuyé.

III.3.1.2 Le déroulement du programme

Le programme que nous avons réalisé se déroule de la façon suivante:

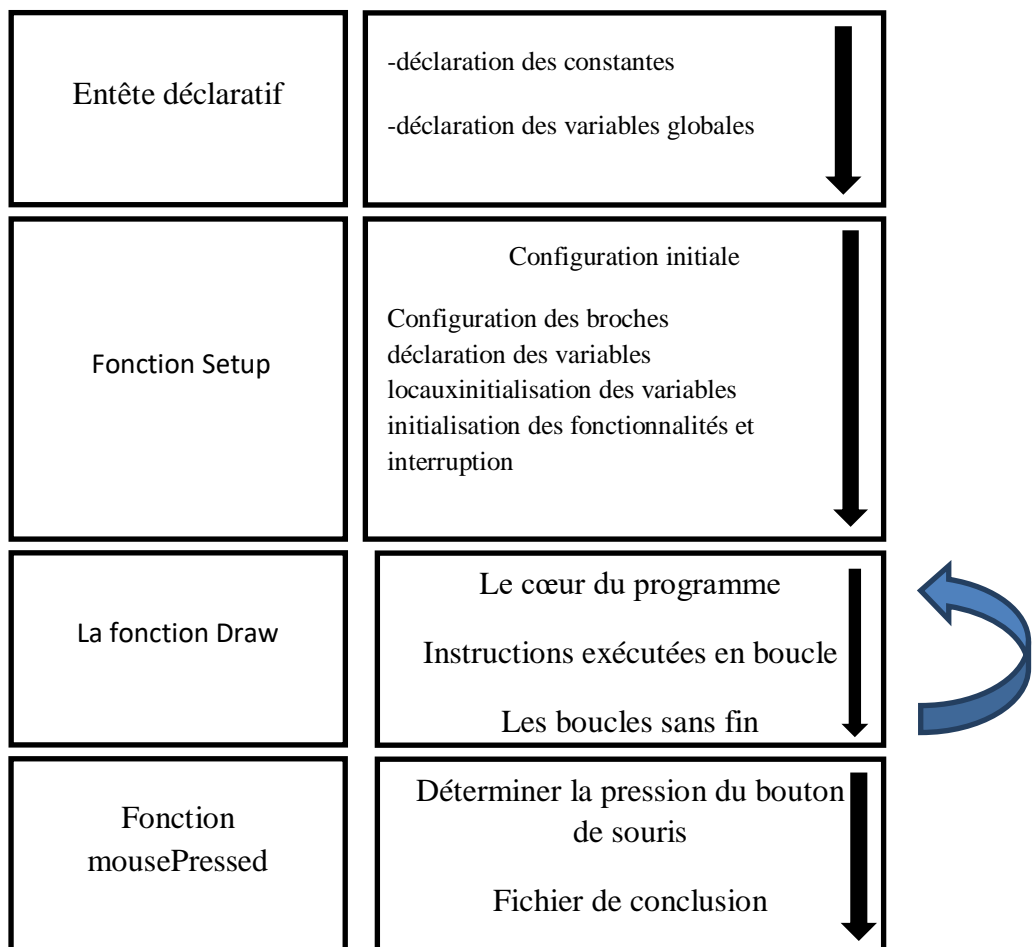


Figure III.7 : Le déroulement du programme

III.3.2.1 Présentation de l'organigramme IDE

L'organigramme est une représentation schématique ou graphique des liens fonctionnels, d'organisation et hiérarchiques d'un organisme, d'un programme, etc. Il peut être réalisé sous différentes formes.

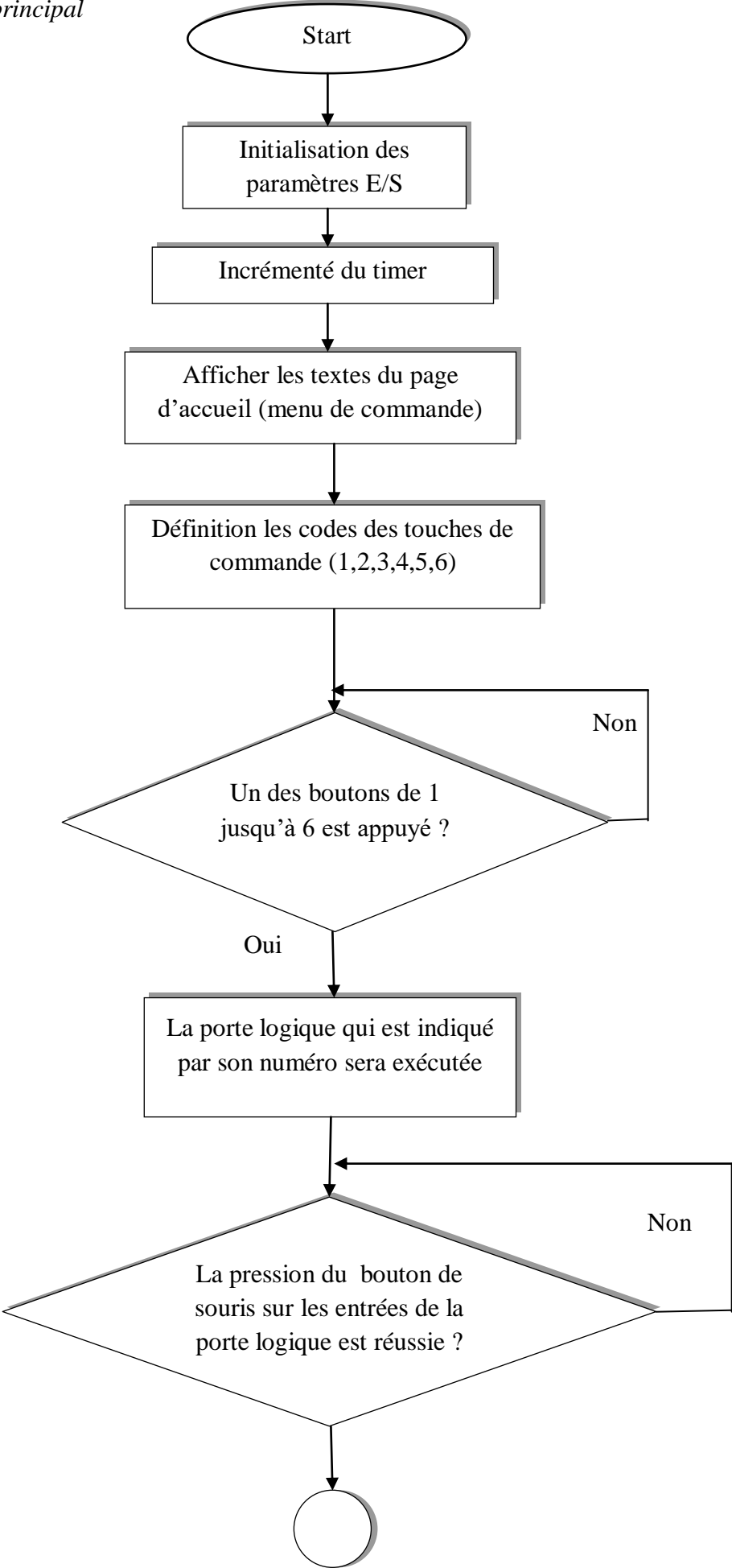
L'organigramme possède une référence documentaire.

III.3.2.2 Organigramme Processing

Avant de passer à la programmation, il est recommandé d'avoir une visualisation du programme qu'on va faire : nous devons réaliser un organigramme qui nous permet de visualiser beaucoup mieux les différents blocs du programme, les boucles, les tests...et expliquer le déroulement des différentes séquences, tant intérieures qu'extérieures.

Chapitre III : conception et programmation du système

Le programme principal



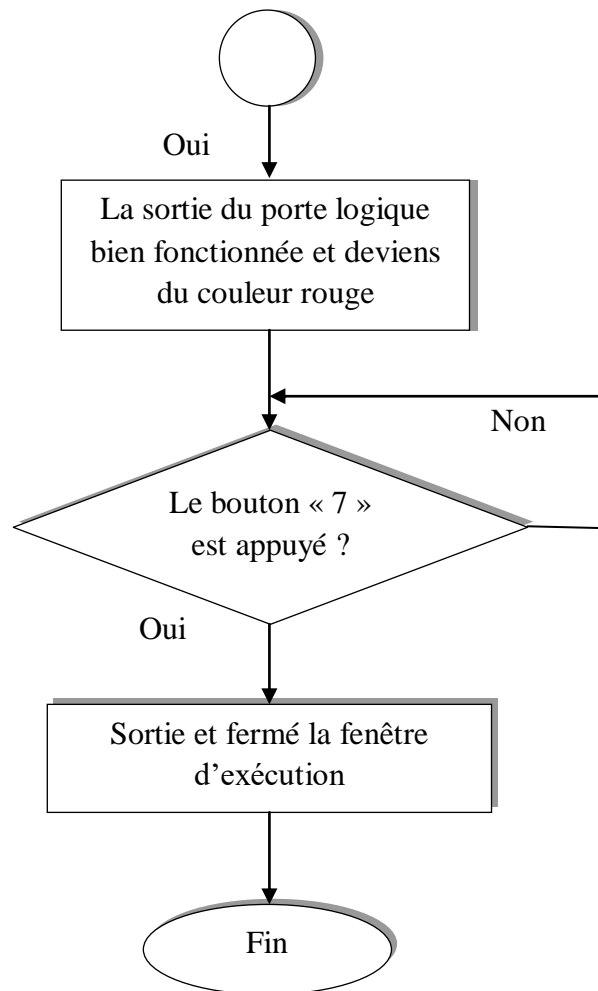


Figure III.8 : l'organigramme principal

Chaque action est symbolisée par un rectangle et chaque test est symbolisé par un losange.

III.3.2.3 Explication du programme

Notre programme nous permet de faire une définition des constantes et des variables dans un premier temps, comme il initialise les entrées/sorties des portes logique ainsi que l'initialisation des temporisateurs. Ce programme réalise un menu de commande affiché en première exécution comme une page d'accueil. Ensuite nous avons des sous-programmes qui contiennent des codes sous formes des boucles qui permettent le fonctionnement de chaque porte logique où il sera activé ou désactivé par la commande d'être humain selon le code proposé dans le programme.

III.3.3.1 Présentation du programme IDE

Le but de ce programme est de réaliser une interface graphique codée par des lettres ou par des chiffres permet d'accéder des opérations sur les portes logique de base (AND ,NAND ,OR, NOR, XOR, NOT).

III.3.3.2 les résultats de la simulation

Après l'exécution du programme nous avons rencontré une fenêtre (figure III.9) contient un menu de commande tel que :

- ✓ Les portes logiques seront choisies quand on appuie sur un des boutons (1-2-3-4-5-6).
- ✓ Le jeu sur une porte logique sera par la pression sur un bouton de la souris tel que :
 - Les deux rectangles présentent des Switch électrique (les deux entrées de la portelologique) et l'ellipse présente la LED (la sortie).
 - Quand les rectangles prennent la couleur « bleu » c'est un indice que le Switch est fermé.
 - l'ellipse prend la couleur « rouge » donc la LED est allumé.

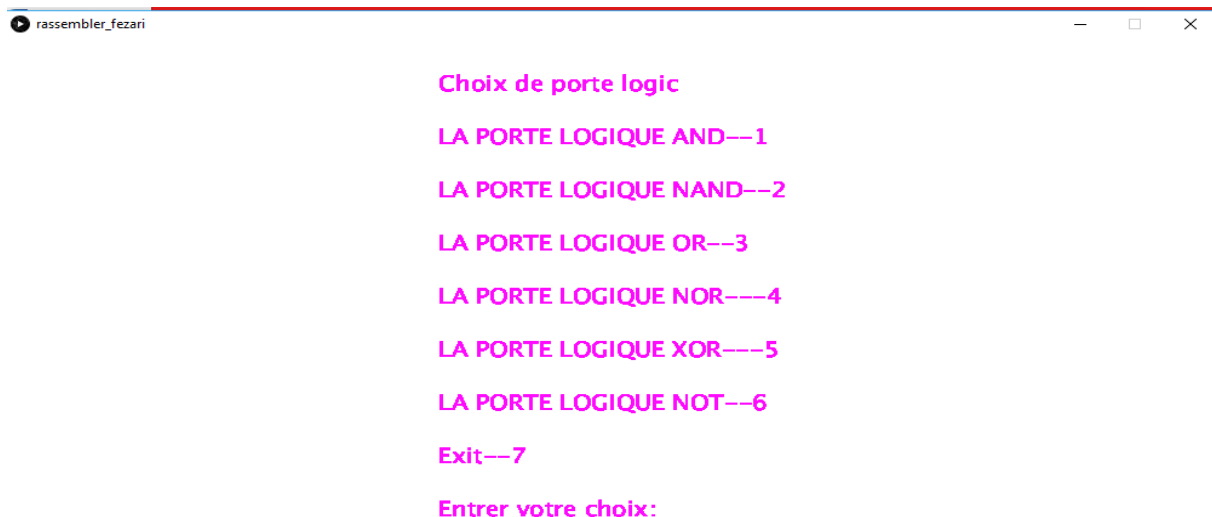


Figure III.9 : La principale fenêtre de menu

Chapitre III : conception et programmation du système

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3



LA PORTE LOGIQUE NOR---4

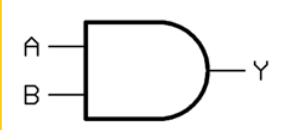
LA PORTE LOGIQUE XOR---5


LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse=AND





A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Table de vérité du porte AND

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3



LA PORTE LOGIQUE NOR---4

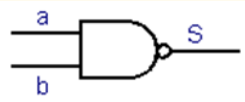
LA PORTE LOGIQUE XOR---5


LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse=NAND





a	b	S
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité du porte NAND

Chapitre III : conception et programmation du système

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3

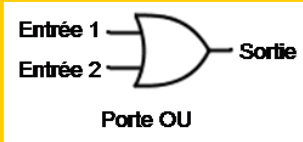

LA PORTE LOGIQUE NOR---4

LA PORTE LOGIQUE XOR---5

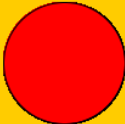
LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse= OR



Porte OU



Entrée1	Entrée2	sortie
0	0	0
0	1	1
1	0	1
1	1	1

Table de vérité du porte OU

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3

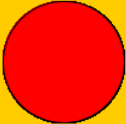
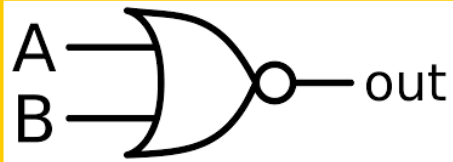

LA PORTE LOGIQUE NOR---4

LA PORTE LOGIQUE XOR---5

LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse= NOR



A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

Table de vérité du porte NOR

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3

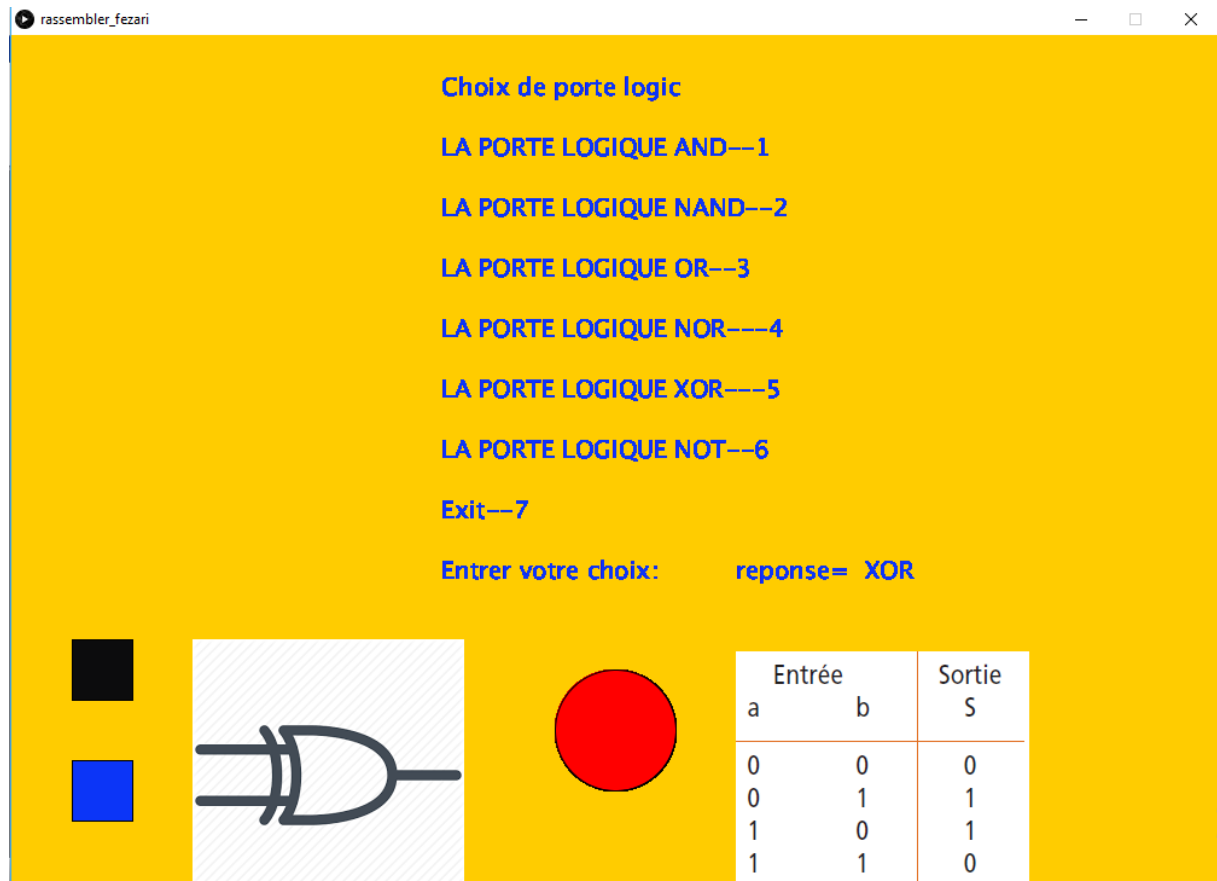
LA PORTE LOGIQUE NOR----4

LA PORTE LOGIQUE XOR---5

LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse= XOR



Entrée		Sortie S
a	b	
0	0	0
0	1	1
1	0	1
1	1	0

rassembler_fezari

Choix de porte logic

LA PORTE LOGIQUE AND--1

LA PORTE LOGIQUE NAND--2

LA PORTE LOGIQUE OR--3

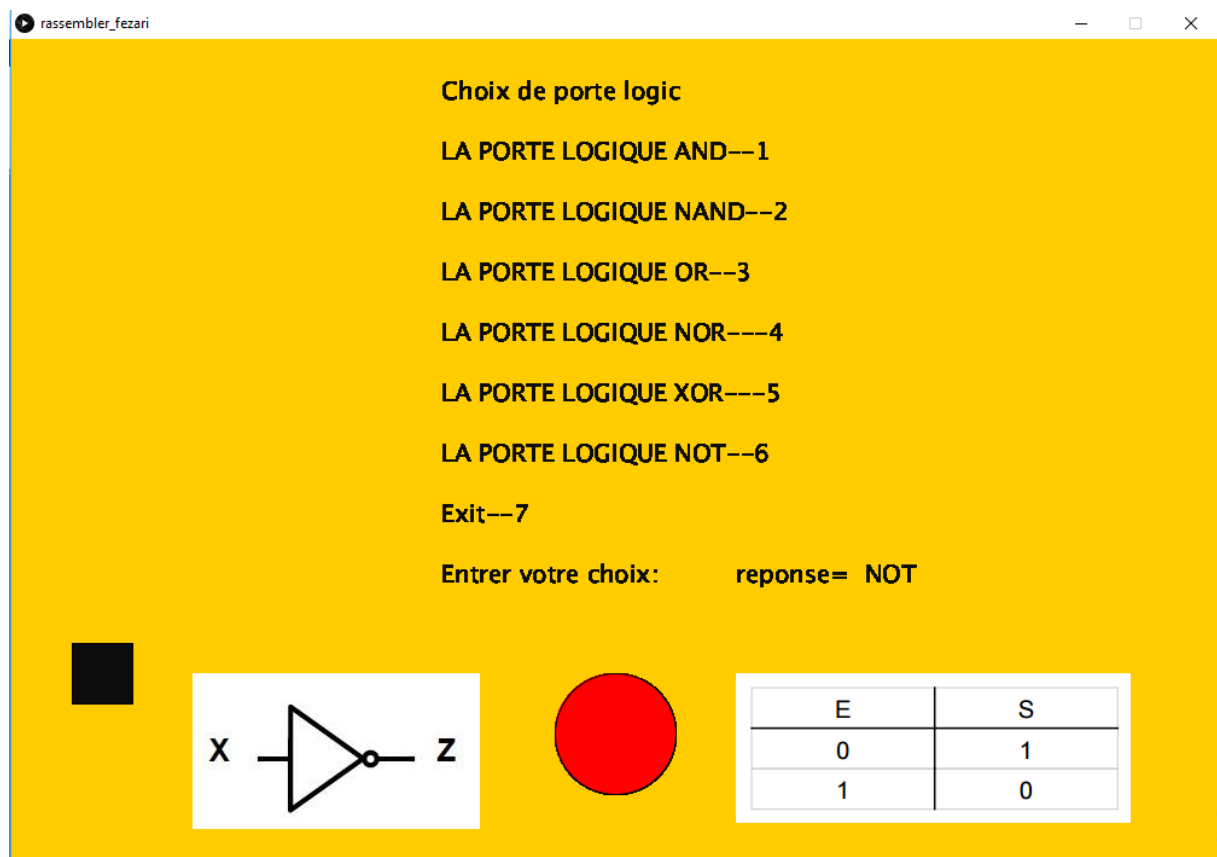
LA PORTE LOGIQUE NOR----4

LA PORTE LOGIQUE XOR---5

LA PORTE LOGIQUE NOT--6

Exit--7

Entrer votre choix: reponse= NOT



E	S
0	1
1	0

Figure III.10 : résultat de simulation d'exécution les portes logique

III.4.1 Présentation du programme IDE d'Arduino

Nous avons commencé par réaliser le montage indiqué sur la (figure III.11) avec Arduino présenté dans le schéma ci-dessous qui travaille simultanément avec le programme Processing précédent. Ensuite, nous avons programmé la carte Arduino (Annexes B) selon la procédure habituelle

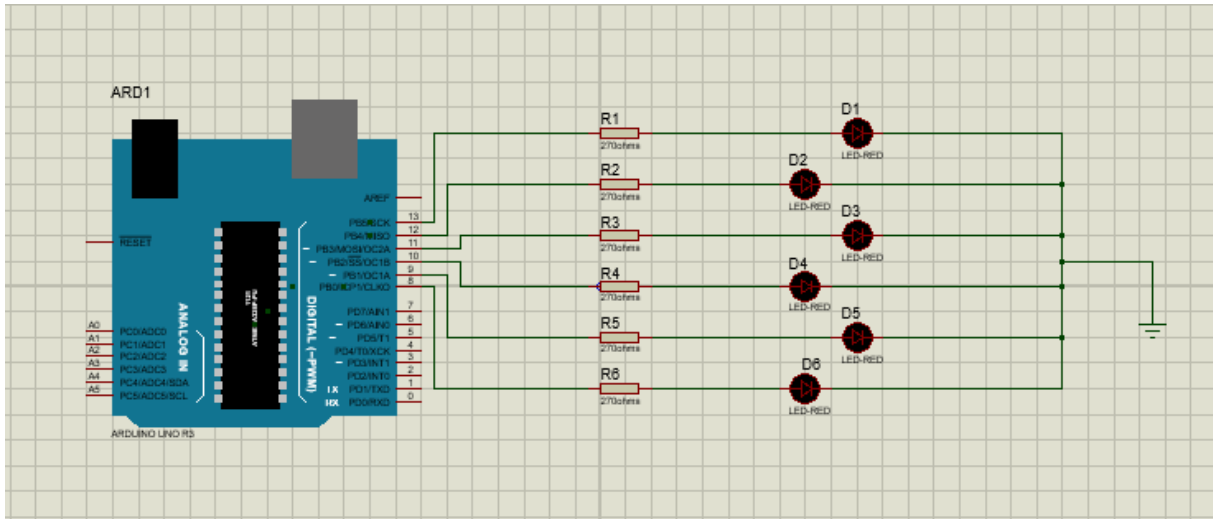


Figure III.11 : Le schéma du montage à réaliser avec Arduino

III.4.2 Le déroulement du programme IDE Arduino

Le programme est simple se déroule de la façon suivante :

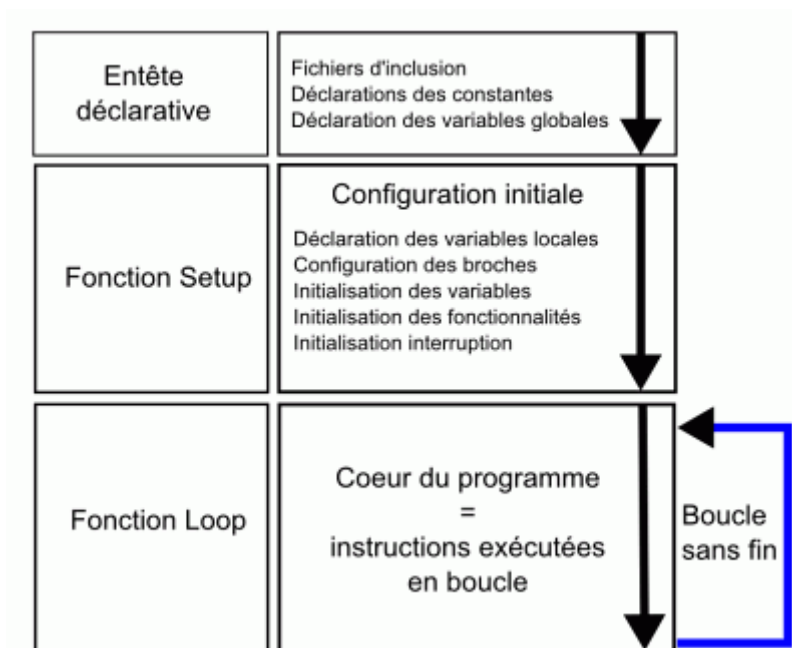


Figure III.12 : Le déroulement du programme IDE Arduino

III.4.3 Explication du programme

a. Au niveau de la partie déclarative :

On déclare les broches des LEDs (6 Leds) de pin 8 jusqu'à pin 13.

b. Au niveau de la fonction d'initialisation `setup()` :

On met les broches en sortie

c. Au niveau de la boucle principale, la fonction `loop()` :

On teste si un caractère est disponible sur le port série :

- Si ce caractère est '1,2,3,4,5,6', on allume la LED
- Si ce caractère est '10,20,30,40,50,60', on éteint la LED

Ce programme permet d'allumer des LEDs connectée sur des broches de la carte Arduino à partir des boutons inverseur graphique dans l'interface Processing côté PC.

En premier temps, on lance le programme Processing coté pc; L'interface graphique virtuelle du Processing est commandée par l'utilisateur tel qu'il joue sur les six portes logique dans lequel il change les valeurs des entrées entre (0et1) par des click sur les boutons rectangulaires donc la sortie va changer sa valeur aussi sa dépend la porte utilisé et sa sera entre 0 et 1.

Si la sortie prend la valeur 1 l'ellipse porte la couleur rouge et la LED branché sur la carte Arduino devient allumé (chaque broche est connecté avec une LED tel qu'il est en série avec sa résistance, et chaque LED présente une des sorties des portes logique)

III.5 Discussion des résultats

Nous avons testé notre program de simulation de Virtual Lab réalisé à base de logiciel Processing, l'étudiant a dans l'interface une table de vérité qu'il doit tester et vérifier les résultats. Les commandes sont envoyés sur la carte Arduino ou la LED correspondante sera allumé ou atteinte selon l'état de la porte logique choisie.

Notre conception est présente par les images de la (figure III.10), on constate que le Processing permet à faire des interfaces graphiques pour plusieurs expériences et notre projet c'est un meilleur exemple. On observe aussi qu'avec cette interface on peut faire des opérations de façon virtuelle.

Conclusion général

Conclusion général

Ce travail d'étude et de recherche concernant le Virtual Lab a été une excellente occasion pour nous d'augmenter nos connaissances sur ce grand projet. Nous avons vu l'importance de ce système comme une plateforme ou interface virtuelle conçu aux étudiants comme une solution après avoir leur grand nombre inscrit dans les universités et une solution aussi pour le manque de matériel utilisé dans les tp ; nous avons vu aussi ses diverses applications dans le monde réel comme mentionné dans le premier chapitre.

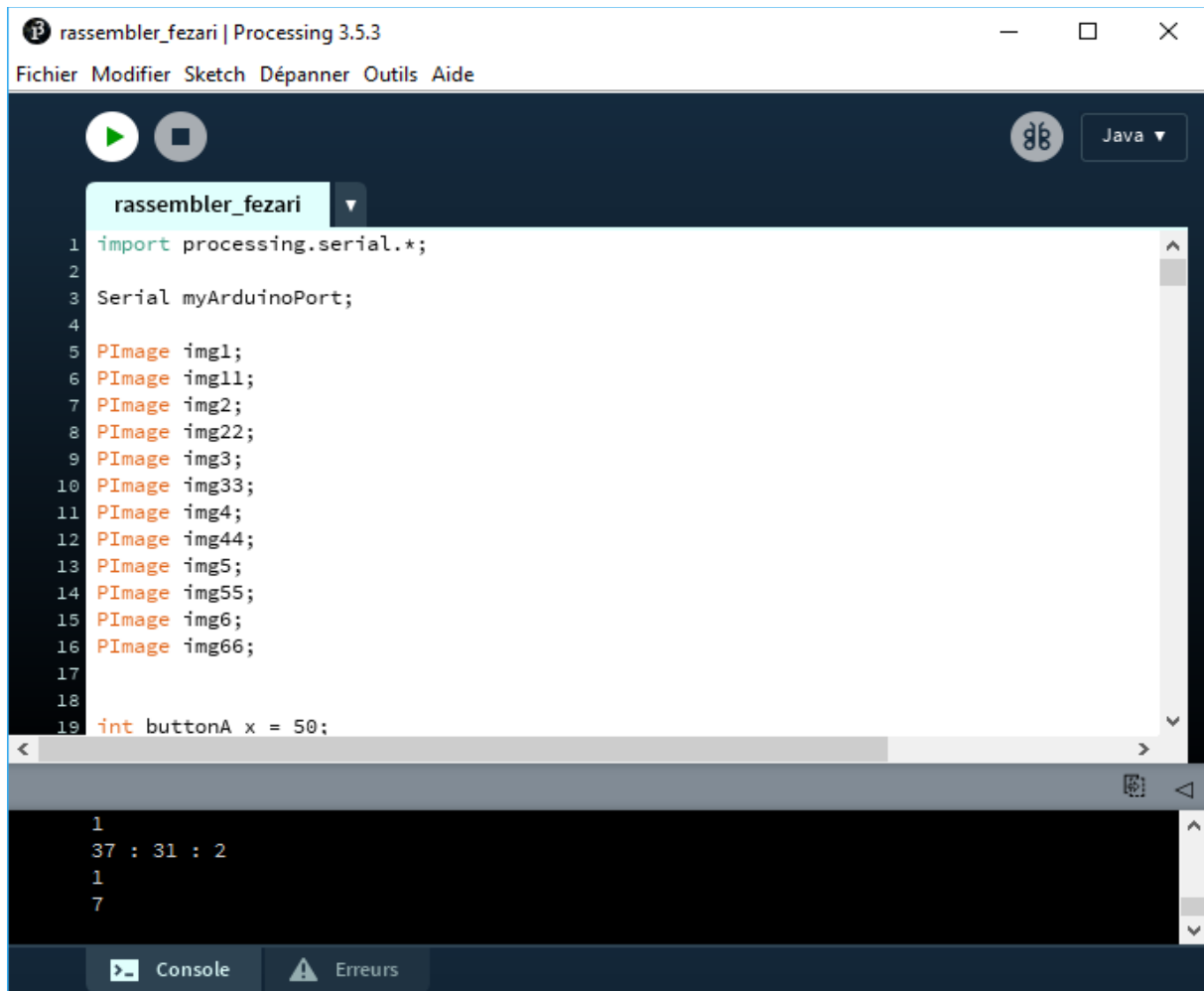
Le chapitre deux nous a permis de reconnaître le logiciel Processing et son environnement qu'il est très riche pour créer des interfaces graphiques virtuel, et comprendre encore l'utilisation d'appareil Arduino et ses caractéristiques, comme nous avons utilisé des composants électroniques de base (led et résistances).

D'un autre coté dans ce travail, nous avons combiné deux parties software et hardware pour la conception de ce virtuel Lab. La partie software consiste à développer les programmes de simulation et de réalisation du système basé sur un interface graphique dans le domaine d'électronique numérique. Nous avons choisie les portes logiques de base afin de vérifier le fonctionnement des circuits combinatoire. L'outil de développement qui a été proposé est le logiciel Processing , très riche en mode graphique mais lourd dans l'interaction homme machine. Les étudiants peuvent vérifier et tester les résultats dans l'interface selon leur table de vérité. Les commandes sont envoyés sur la carte Arduino ou la LED correspondante sera allumé ou atteinte selon l'état de la porte logique choisie (la partie hardware).

En perspective, nous souhaitons amélioré l'interface graphique par l'insertion des interrupteurs « images », l'ajout d'autres circuits logique tel que : compteurs, multiplexeurs et décodeurs, et l'ébergement de l'application dans un site web.

Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire des interfaces graphiques virtuel dans le développement des Virtual Lab.

Annexes : Annexes A



rassembler_fezari | Processing 3.5.3

Fichier Modifier Sketch Dépanner Outils Aide

```
1 import processing.serial.*;
2
3 Serial myArduinoPort;
4
5 PImage img1;
6 PImage img11;
7 PImage img2;
8 PImage img22;
9 PImage img3;
10 PImage img33;
11 PImage img4;
12 PImage img44;
13 PImage img5;
14 PImage img55;
15 PImage img6;
16 PImage img66;
17
18
19 int buttonA x = 50;
```

Console

```
1
2 37 : 31 : 2
3 1
4 7
```

```
19 int buttonA_x = 50;
20 int buttonA_y = 500;
21 int buttonA_width = 50;
22 int buttonA_height = 50;
23 int buttonA_colorR = 0xff;
24 int buttonA_colorG = 0x00;
25 int buttonA_colorB = 0x00;
26
27 int dure=20; // duré du TP7 en minutes
28
29
30 int buttonB_x = 50;
31 int buttonB_y = 600;
32 int buttonB_width = 50;
33 int buttonB_height = 50;
34 int buttonB_r = 00;
35 int buttonB_color = #FF0F17 ;
36
37 int defaultColor = #0C0C0D;
```

Annexes : Annexes A

```
39 boolean buttonA = false;
40 boolean buttonB = false;
41
42 // gestion temp
43 int s = second(); // Values from 0 - 59
44 int m = minute(); // Values from 0 - 59
45 int h = hour(); // Values from 0 - 23
46 int tim;
47 int t0=h*60+m;
48
49
50 void setup() { // setup() runs once
51 size(1000, 800);
52 background(255);
53 fill(#FE0AFF);
54
55 img1 = loadImage("PORTE AND.png");
56 img11 = loadImage("table de vérité AND.PNG");
57
```

```
58 img2 = loadImage("PORTE NAND.PNG");
59 img22 = loadImage("table de vérité NAND.PNG");
60
61 img3 = loadImage("images.png");
62 img33 = loadImage("table de vérité.PNG");
63
64 img4 = loadImage("PORTE NOR.png");
65 img44 = loadImage("table de vérité NOR.PNG");
66
67 img5 = loadImage("imagexor.png");
68 img55 = loadImage("tablexor.png");
69
70 img6 = loadImage("imagenot.jpg");
71 img66 = loadImage("tablenot.png");
72 }
73
74
75 void draw() {
76 // Menu *****
```

```
78 textSize(20);
79 text(" Choix de porte logic",350,50);
80 text(" LA PORTE LOGIQUE AND--1",350,100); text(" LA PORTE LOGIQUE NAND--2",350,150);
81 text(" LA PORTE LOGIQUE OR--3",350,200); text(" LA PORTE LOGIQUE NOR---4",350,250);
82 text(" LA PORTE LOGIQUE XOR---5",350,300); text(" LA PORTE LOGIQUE NOT--6",350,350);
83 text(" Entrer votre choix:",350,450);text(" ",450,450);
84
85 //gestion temps*****
86 int s = second(); // Values from 0 - 59
87 int m = minute(); // Values from 0 - 59
88 int h = hour(); // Values from 0 - 23
89 println(h,":",m,":",s);
90 int t1=h*60+m;
91 tim=t1-t0;
92 println(tim);
93 if (tim>=dure) {text(" TIME OUT !!!!",100,480); delay(5000); exit(); }
94 //*****
95
96 if (keyPressed) {background(255, 204, 0);
```

Annexes : Annexes A

```
97 // <<<<<<<<<<and *****
98   if (key == '1') {
99
100     print(key); text("reponse=AND",600,450);
101
102
103     image(img1, 150,525);
104     image(img11, 600,540);
105
106     if(buttonA && buttonB){
107       fill(#FF0F17);}
108     // fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
109     //}else if(buttonB){
110     //fill(buttonB_color);
111     //}
112     else{
113       fill(defaultColor);
114     }
115     ellipse(500,575, 100, 100);
```

```
97 // <<<<<<<<<<and *****
98   if (key == '1') {
99
100     print(key); text("reponse=AND",600,450);
101
102
103     image(img1, 150,525);
104     image(img11, 600,540);
105
106     if(buttonA && buttonB){
107       fill(#FF0F17);}
108     // fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
109     //}else if(buttonB){
110     //fill(buttonB_color);
111     //}
112     else{
113       fill(defaultColor);
114     }
115     ellipse(500,575, 100, 100);
```

```
135     image(img2, 150,525);
136     image(img22, 600,540);
137
138     if(!(buttonA && buttonB)){
139       fill(#FF0F17);}
140     // fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
141     //}else if(buttonB){
142     //fill(buttonB_color);
143     //}
144     else{
145       fill(defaultColor);
146     }
147     ellipse(500,575, 100, 100);
148     if(buttonA){
149       fill(#0C35F7);}
150     else{
151       fill(#0C0C0D);}
152     rect(buttonA_x, buttonA_y, buttonA_width, buttonA_height);|
153
```

Annexes : Annexes A

```
154 if(buttonB){
155     fill(#0C35F7);}
156 else{
157     fill(#0C0C0D);}
158 rect(buttonB_x, buttonB_y, buttonB_width, buttonB_height, buttonB_r);
159 }
160 // Or *****
161 if (key == '3') {
162
163     print(key); text("reponse= OR",600,450);
164
165     image(img3, 150,525);
166     image(img33, 600,540);
167
168     if(buttonA){
169         fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
170     }else if(buttonB){
171         fill(buttonB_color);|
172     }elseif
```

```
172 }elseif
173     fill(defaultColor);
174 ]
175 ellipse(500,575, 100, 100);
176 if(buttonA){
177     fill(#0C35F7);}
178 else{
179     fill(#0C0C0D);}
180 rect(buttonA_x, buttonA_y, buttonA_width, buttonA_height);
181
182 if(buttonB){
183     fill(#0C35F7);}
184 else{
185     fill(#0C0C0D);}
186 rect(buttonB_x, buttonB_y, buttonB_width, buttonB_height, buttonB_r);
187 }
188 // nor*****
189 if (key == '4') {
190
```

```
191     print(key); text("reponse= NOR",600,450);
192
193     image(img4, 150,525);
194     image(img44, 750,540);
195
196 if((!buttonA) && (!buttonB)){
197     fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
198 } else{
199     fill(defaultColor);
200 }
201 ellipse(650,575, 100, 100);
202 if(buttonA){
203     fill(#0C35F7);}
204 else{
205     fill(#0C0C0D);}
206 rect(buttonA_x, buttonA_y, buttonA_width, buttonA_height);
207 if(buttonB){
208     fill(#0C35F7);}
209     elseif
```

Annexes : Annexes A

```
210 fill(#0C0C0D);}
211 rect(buttonB_x, buttonB_y, buttonB_width, buttonB_height, buttonB_r);
212 }
213 // xor *****
214 if (key == '5') {
215     print(key); text("reponse= XOR",600,450);
216
217     image(img5, 150,500);
218     image(img55, 600,510);
219
220 if((buttonA && !buttonB)||(!buttonA && buttonB)){
221     fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
222 } else{
223     fill(defaultColor);
224 }
225 ellipse(500,575, 100, 100);
226 if(buttonA){
227 fill(#0C35F7);}
228 else{
```

```
229 fill(#0C0C0D);}
230 rect(buttonA_x, buttonA_y, buttonA_width, buttonA_height);
231 if(buttonB){
232 fill(#0C35F7);}
233 else{
234 fill(#0C0C0D);}
235 rect(buttonB_x, buttonB_y, buttonB_width, buttonB_height, buttonB_r);
236 }
237 // not *****
238 if (key == '6') {
239     print(key); text("reponse= NOT",600,450);
240
241     image(img6, 150,525);
242     image(img66, 600,525);
243
244 if(!buttonA) {
245     fill(buttonA_colorR, buttonA_colorG, buttonA_colorB);
246 }else{
247     fill(defaultColor);
```

```
248 }
249 ellipse(500, 575, 100, 100);
250 if(buttonA){
251 fill(#0C35F7);}
252 else{
253 fill(#0C0C0D);}
254 rect(buttonA_x, buttonA_y, buttonA_width, buttonA_height);}
255 // Exit *****
256 if (key == '7') {
257     print(key); text("reponse=EXIT",650,450); exit();
258 }
259 }
260 void mouseClicked(){
261 // mouseX = x of mouse click position
262 // mouseY = y of mouse click position
263 if (mouseX >= buttonA_x && mouseX <= buttonA_x + buttonA_width &&
264     mouseY >= buttonA_y && mouseY <= buttonA_y + buttonA_height) {
265 | buttonA = !buttonA;
266 }
```

Annexes : Annexes A

```
255 // Exit *****
256 if (key == '7') {
257     print(key); text("reponse=EXIT",650,450); exit();
258 }
259 }
260 void mouseClicked(){
261     // mouseX = x of mouse click position
262     // mouseY = y of mouse click position
263     if (mouseX >= buttonA_x && mouseX <= buttonA_x + buttonA_width &&
264         mouseY >= buttonA_y && mouseY <= buttonA_y + buttonA_height) {
265         buttonA = !buttonA;
266     }
267     if (mouseX >= buttonB_x && mouseX <= buttonB_x + buttonB_width &&
268         mouseY >= buttonB_y && mouseY <= buttonB_y + buttonB_height) {
269         buttonB = !buttonB;
270     }
271 }
272
273
```

Présentation graphique du programme Processing

Annexes : Annexe B

```
/ This is Ar// This is Arduino Code six LEDs on pins 8-13

int ledPin = 13;    // LED connected to digital pin 13

int val = 0;

void setup(){

pinMode(ledPin, OUTPUT); // sets the digital pin as output

pinMode(12, OUTPUT);pinMode(11, OUTPUT);pinMode(10, OUTPUT);

pinMode(9, OUTPUT);pinMode(8, OUTPUT);

Serial.begin(9600);

}

void loop(){

val = Serial.read();

// LED 13 → état de porte AND

if (val==10){digitalWrite(13,LOW);}

if (val==1){digitalWrite(13,HIGH);}

// LED 12 → état de porte NAND

if (val==20){digitalWrite(12,LOW);}

if (val==2){digitalWrite(12,HIGH);}

// LED 11 → état de porte OR

if (val==30){digitalWrite(11,LOW);}

if (val==3){digitalWrite(11,HIGH);}

// LED 10 → état de porte NOR

if (val==40){digitalWrite(10,LOW);}

if (val==4){digitalWrite(10,HIGH);}

// LED 09 → état de porte XOR

if (val==50){digitalWrite(9,LOW);}

if (val==5){digitalWrite(9,HIGH);}

// LED 08 → état de porte NOT
```

Annexes : Annexe B

```
if (val==60){digitalWrite(8,LOW);}
if (val==6){digitalWrite(8,HIGH);}
}
```


Bibliographie

- [1] https://www.researchgate.net/publication/325386735_Virtual_Laboratory_for_Basic_Electronics
- [2] International Research Journal of Engineering and Technology (IRJET). Volume: 04 Issue: 06 | June -2017
- [3] Lynn Mercer Przemyslaw Prusinkiewicz James Ranan Department of Computer Science University of Regina Regina, Saskatchewan Canada S4S 0A2
- [4] Cheng K.W.E. Chan C.L., Cheung N.C. and Sutanto D Power Electronics Research Centre Department of Electrical Engineering The Hong Kong Polytechnic University Hung Hom, Hong Kong
- [5] Engineering Education a Journal of the Higher Education Academy
- [6] Send Orders for Reprints to reprints@benthamsience.ae [7] Turkish Online Journal of Distance Education-TOJDE April 2017 ISSN 1302-6488 Volume: 18 Number: 2 Article 13
- [8] Boulares Ahmed et Mntambo Marjorie. Etude de la commande et simulation des circuits d'un pendule inversé. Master en Génie Electrique. UNIVERSITE M'HAMED BOUGARA-BOUMERDES. 2017.
- [9] Andreas Risberg. VIRTUAL LAB ENVIRONNEMENT design and imlementation of interactive, web based applications for increased learning. Lulea Tniversity of Technology. MASTER'S THESIS. 2009
- [10] Construire un laboratoire électronique interactif. Kin Cheong Chu (Hong Kong Institute of Vocational Education (Tsing Yi), Hong Kong). Titre de la source: Encyclopédie de l'enseignement à distance, deuxième édition Copyright: 2009 | Pages: 7
- [11] Conception d'une interface de mesure d'instrument de laboratoire virtuel basée sur le Web. Janvier 2001 avec 65 lectures
- [12] <http://ixd.education/2015/10/introduction-a-processing-3-0/> (disponible Avril 2019)
- [13] https://fr.wikipedia.org/wiki/Fonction_logique (disponible Mars 2019)
- [14] <https://whatis.techtarget.com/fr/definition/Porte-Logique> (disponible Mai 2019)