République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj de Bouira

Faculté des Sciences et des Sciences Appliquées

Département de Génie Electrique

Projet de fin d'études en vue de l'obtention du diplôme de Master

Option : Systèmes des Télécommunications

Thème :

Etude et réalisation d'une Horloge Géante pour la FSSA

Laboratoire des Matériaux et du Développement Durable (LM2D)

Réalisé Par :

- AIT MEZIANI Aghiles

-DECHOUN Daou

Encadré par :

-Dr. NOURINE Mourad

Membres de jury:

Président:
Examinateur 1:

Soutenance le: / /

Examinateur 2:
Raporteur:

2017/2018

Remerciements

Avant tout, nous remercions le dieu tout puissant de nous donner la force afin de mener à bien ce travail.

Il nous est offert ici, par ces quelques lignes, la possibilité de remercier les personnes qui ont contribué à faire ce mémoire.

Nous tenons tout particulièrement à remercier l'encadreur Mr M .NOURINE pour le savoir qu'il nous a inculqué, pour ses précieux conseils et orientations, et surtout pour le soutient qu'il nous a témoigné ; merci. Nous tenons également à remercier toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail, notamment LES ENSEIGNANTS du département de Génie Électrique de l'Université de Bouira pour leur soutien et leur aide pendant la rédaction.

Nous vous exprimons notre gratitude et notre profond respect, pour tous cela nous vous disons « Merci ».

Nous remercions aussi nos familles et amis qui nous ont aidé, soutenue et qui nous ont offerts tous les moyens possibles à nos dispositions et pour toutes connaissances nécessaires pour établir ce travail.

Ce travail est celui de tous ceux qui m'ont permis d'en arriver là. Celui de mes parents en premier lieu, qui ont toujours cru en moi, même dans les pires moments de ma scolarité. Eux qui m'ont toujours assuré le confort avec leurs modestes moyens pour uniquement m'encourager pour qu'arrive cet instant. Mon premier enseignant à l'école primaire, Mr Guellaz Boujemaa qui m'a tant donné confiance en moi. Une confiance parfois excessive qui m'a laissé lever le pied par moments mais qui m'a toujours obligé d'honorer ma présence où que je sois. Mes enseignants au CEM Mes enseignants au lycée. Ceux de l'Université particulièrement pour leur attitude envers nous, étudiants, une attitude professionnel qui responsabilise l'étudiant et le force à faire des efforts sans qu'il y ait contrainte, ce qui était nouveau pour nous une fois arrivé à cet espace estudiantin. C'est pour cela que je me dois de dédier ce travail à toutes ces personnes que je n'oublierai pas car ils ont construit la personne que je suis aujourd'hui.

Je dédie ce travail à mes frères et mes sœurs. Mon oncle Abdenour qui était et demeure mon repère dans la vie, pour qui son départ vers l'étranger a laissé comme une désorientation, un repère éloigné, des conseils et des orientations quotidiennes devenus occasionnels qui m'avait laissé prendre parfois des choix qui ne seraient pas les siens mais chez qui j'ai beaucoup appris. Je le dédie à mon neveu Aymen qui a fait de moi un tonton. Je dédie ce travail aussi à tous mes camarades et amis que j'ai côtoyé durant toutes ces années de scolarités. Ceux avec qui j'ai partagé d'excellents moment et dans tous les paliers, au primaire, au CEM et au lycée et en particulier mes amis et camarades à l'Université, spécialement ceux qui m'ont aidé dans ce mémoire, mon binôme Ghilas qui malgré ses difficultés n'a cessé de faire des efforts pour ce travail, tous mes camarades de classe qui sont donné à cœur joie de nous aider et durant toutes les étapes de ce mémoire.

Je le dédie aussi à tous mes amis et camarades de luttes, Aziz, Zinedine, Amine pour m'avoir dédié son mémoire, Smail Sasuki, Fawzi, Hamza, Mazigh, Tina, Amel², Imane, Leila …. C'est grâce à vous tous.

Le courage de dire, la force d'agir et la volonté de construire

DAOU DECHOUN

Table des matières

Chapitre II : Etude et Simulation de l'horloge

Chapitre III : Réalisation pratique de l'horloge

Liste des figures

Liste des tableaux

Liste des acronymes et des abréviations

3V3 : 3.3 Volts.

AREF : Alimentation de Reference.

CAN: Concerter Analogic Numeric

CMOS: Complimentary Metal Oxyde Semiconductor.

CMS : Composant Monté en Surface.

DTR: Data Terminal Ready.

EEPROM: Electrically Erasible Programmable Read-Only Memory.

E/S : Entrée/Sortie.

FTDI : Future Technology Device International.

GND: Ground (mass).

IDE: Integrated Development Environment.

LCD: Liquid Crystal Display.

LED: Lighting Electricaly Diode.

MIT: Massachusetts Institue of Tehnology.

PWM: Pulse Width Modulation.

RISC: Reduced Instruction Set Computer.

SPI: Serial Peripheral Interface.

SRAM: Static Random Acces Memory.

Vin: Volt in.

UART/USART: Universal Asynchronous/Synchronous Receiving Transmitting.

USB: Universal Serial Bus.

VCC: Voltage Current Continu.

FPGA: Fild Programmable Gate Array.

ASIC: Application Specific Integrated Circuits.

DC: Direct Current.

FTDI: Future Technology Devices International.

RTC: Real Time Clock.

SDA: Serial Data line.

SCL: Serial Clock line.

HEX: HEXADECIMAL.

Input/Output: Entrée/sortie.

Kbs : Kilo bit par seconde.

OS : Operating System.

BP : Boutons Poussoirs.

Introduction Générale

Introduction Générale

Connaitre le temps a toujours été important pour l'être humain afin de se répertorier. La mesure du temps a rapidement été une préoccupation importante, notamment pour organiser la vie sociale, religieuse et économique des sociétés. Les phénomènes périodiques du milieu où l'homme vivait comme le déplacement quotidien de l'ombre, le retour des saisons ou le cycle lunaire ont servi de premières références. Mais progressivement, l'humain s'est inspiré de phénomènes physiques, dont il avait remarqué le caractère périodique, pour concevoir et mettre au point des dispositifs de mesure du temps de plus en plus précis, ainsi que des unités adaptées , l'horloge numérique en est un exemple. Cet appareil permet aux hommes de connaître l'heure à tout moment et en tout lieu, et de programmer leurs journées de façon à rationner le temps, pour les mieux organisés.

L'objectif de ce projet PFE est de réaliser une horloge numérique qui permet à toute personne se trouvant à l'enceinte de la faculté FSSA de pouvoir consulter l'heure sans trop de difficultés. Cette horloge sera accrochée à un endroit visible, de distance et hauteur raisonnables. Son utilité et d'avoir à toute personne une référence temporelle, un accès facile à l'information du temps qu'il est, et d'unifier à l'ensemble de la communauté universitaire (étudiant, employés, visiteurs) le temps de référence.

Le présent mémoire s'articule sur trois chapitres, le premier chapitre aborde des généralités sur l'environnement Arduino, il englobe le parcours, les domaines d'utilisations, ainsi que les différents composants qui constituent la carte et leurs rôles. Le deuxième sera consacré à l'étude et à la simulation d'une horloge numérique sous Proteus. Quant au dernier on passe à la réalisation pratique de l'horloge numérique géante. A la fin nous terminons par une conclusion générale suivie par des perspectives envisagées.

Chapitre I

Généralités sur l'Arduino

Généralités sur l'Arduino

I.1 Introduction

De nos jours (2018), l'électronique classique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques, par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit (smartphone, ordinateur, Arduino …).

I.2 Système embarqué :

I.2.1 Définition d'un système embarqué :

Un système embarqué est un système complexe qui intègre du software et du hardware conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (Personal Computer).

Un système embarqué est autonome et ne possède pas des E/S standards tels qu'un clavier ou un écran d'ordinateur. Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables FPGA (Fild Programmable Gate Array), des circuits dédiés à des applications spécifiques ASIC (Application Specific Integrated Circuits) ou des modules analogiques sont en plus utilisés.

Le logiciel a une fonctionnalité fixe à exécuter qui est spécifique à une application. L'utilisateur n'a pas la possibilité de modifier les programmes.

Les systèmes embarqués sont désormais utilisés dans des applications diverses, tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation ... [1]

I.2.2 Caractéristiques d'un système embarqué :

Les systèmes embarqués fonctionnent généralement en Temps Réel (TR) : les opérations de calcul sont alors faites en réponse à un événement extérieur (interruption matérielle). La validité et la pertinence d'un résultat dépendent du moment où il est délivré. Une échéance manquée induit une erreur de fonctionnement qui peut entraîner soit une panne du système (plantage), soit une dégradation non dramatique de ses performances.

Lorsque les systèmes embarqués sont utilisés dans les produits de grande consommation, ils sont fabriqués en grande série. Les exigences de coût se traduisent alors en contraintes sur les différentes composantes du système : utilisation de faibles capacités mémoires et de petits processeurs (4 bits ou 8 bits), mais en grand nombre. Ainsi, les systèmes embarqués sont particulièrement sensibles au coût de production. Il existe des applications dans lesquelles les contraintes de coût de production et de maintenance ont une importance de même niveau que les performances envisagées.

Dans les systèmes embarqués autonomes, la consommation d'énergie est un point critique pour le coût. En effet, une consommation excessive augmente le prix de revient du système embarqué, car il faut alors des batteries de forte capacité. [1]

I.2.3 La composition d'un système embarqué :

Quelle que soit la nature et la complexité du système, on décompose un système embarqués en :

- **Le système contrôlé :** Environnement (procédé) équipé d'une instrumentation qui réalise l'interface avec le système de contrôle.

- **Le système de contrôle :** Eléments matériels (microprocesseurs) et logiciels dont la mission est d'agir sur le procédé via les actionneurs en fonction de l'état de ce procédé indiqué par les capteurs de manière à maintenir ou conduire le procédé dans un état donné. [1]

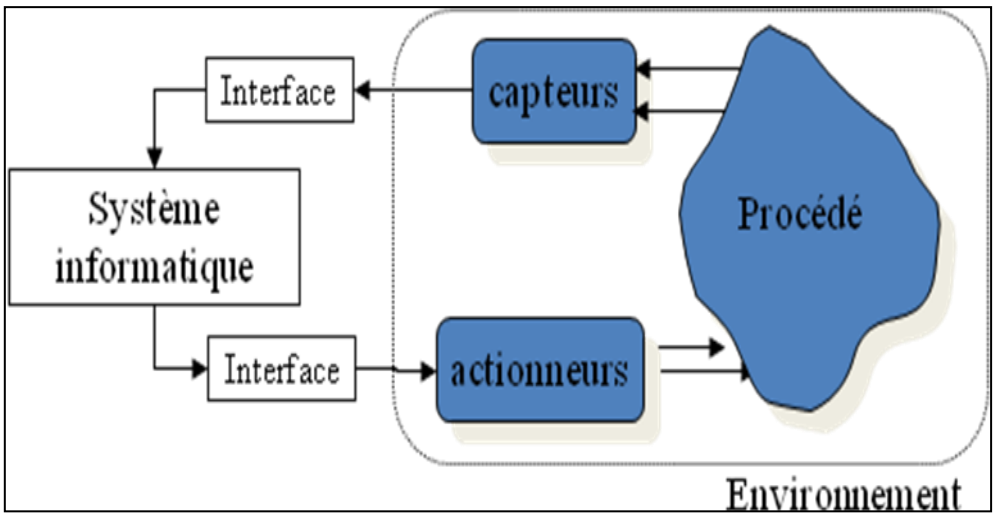


Figure I.1 Schéma illustratif des composants d'un système embarqué

I.2.4 Spécificités d'un système embarqué :

Les caractéristiques principales d'un système électronique embarqué sont :

- **Autonomies :** Une fois enfouis dans l'application ils ne sont (le plus souvent) plus accessibles.
- **Temps réel :** Les temps de réponses de ces systèmes sont aussi importants que l'exactitude des résultats.
- **Réactifs :** Il doit réagir à l'arrivée d'informations extérieures non prévues. [1]

On abordera dans la prochaine section l'Arduino, un exemple d'un système embarqué nécessaire pour la réalisation de notre horloge numérique.

I.3 Présentation de l'Arduino :

Arduino est un projet créé par une équipe de développeurs, composée de six individus : Massimo Banzi, David Cuartielles, TomIgoe, Gianluca Martino, David Mellis et Nicholas Zambetti. Cette équipe a créé le "système Arduino", C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. [8]

I.4 Le but de la création de l'Arduino :

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique. [8]

I.5 Définition du module Arduino :

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, l'électronique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise le logiciel IDE Arduino. [2]

I.6 Les gammes de la carte Arduino :

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques-uns afin d'éclaircir l'évaluation de ce produit :

Version d'arduino	Caractéristique
Le NG d'Arduino	Une interface d'USB pour programmer et usage d'un ATmega8.
L'Arduino Mini	Une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
L'Arduino Nano	L'Arduino Nano, une petite carte programmée à l'aide d'un port USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus récente).
Le LilyPad Arduino	Une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
Le NG d'Arduino plus	Avec une interface d'USB pour programmer et usage d'un ATmega168.
L'Arduino Bluetooth	Avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
L'Arduino Diecimila	Avec une interface d'USB et utilise un microcontrôleur ATmega168.
L'Arduino Duemilanove (2009)	Utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
L'Arduino Mega	Utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
L'Arduino UNO	Utilisations microcontrôleur ATmega328.
L'Arduino Mega2560	utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
L'Arduino Leonardo	Avec un morceau ATmega3Ù4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
L'Arduino Esplora	Ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Tableau I.1 Les gammes de l'Arduino [3]

Parmi ces types, nous avons choisi une carte Arduino NANO. L'intérêt principal de cette carte est de faciliter la mise en œuvre de notre projet.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega328. Il dispose de14 broches numériques d'E/S et 8 entrées analogiques. [4]

I.7 Arduino NANO :

I.7.1 La fiche technique de l'Arduino NANO :

Caractéristiques	**Détails**
Alimentation	- Via port USB. - Tension de fonctionnement 5V. - Tension d'entrée (recommandé) 7-12V. - Tension d'entrée (limite) 6-20V.
Microprocesseur	- ATMega328.
Mémoire	- Mémoire flash 32 ko. - Mémoire SRAM 2 ko. - Mémoire EEPROM 1ko.
Entrées et sorties	- 14 broches d'E/S dont 6 PWM. - 8 entrées analogiques.
Intensité par E/S	- 40 mA.
Cadencement	- 16 Mhz.
Bus	- Série, I2C et SPI.
Boitier	- DIL30.
Dimensions	- 45 x 18 x 18 mm.
Version d'origine	- Fabriquée en Italie.

Tableau I.2 Fiche technique de l'Arduino NANO [4]

La carte est présentée sur la figure (I.2)

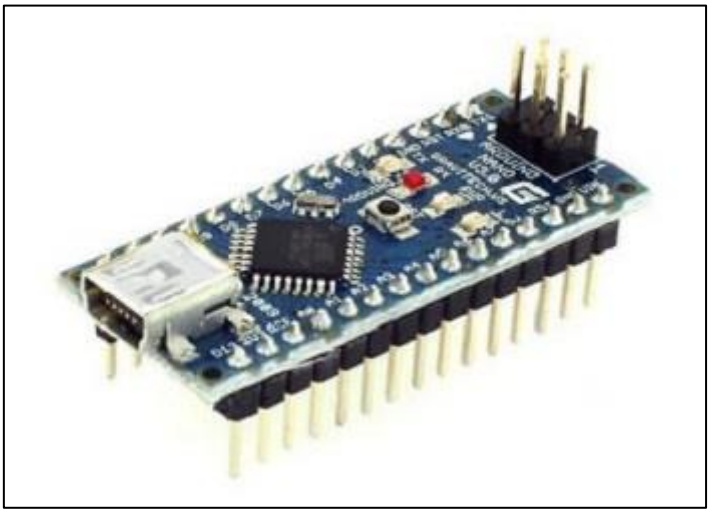


Figure I.2 La carte Arduino NANO [4]

I.7.2 Pourquoi l'Arduino NANO :

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation 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 tels que :

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions peut être assemblée à la main, (ces cartes préassemblées coûtent moins de 1500 Dinars Algérien).

- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

- **Un environnement de programmation clair et simple** : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer

profit également.

- **Logiciel Open Source et extensible :** le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

- **Matériel Open source et extensible** : les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût. [5]

- **La communauté arduino :** La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète des codes sur le site arduino.cc. [8]

I.8 La constitution de la carte Arduino NANO :

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmateur dédié ne soit pas nécessaire.

I.8.1 Partie matérielle hardware :

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

I.8.1.1 Le Microcontrôleur ATMega328 :

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors, les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C.

La figure I.3 montre un microcontrôleur ATmega328, qu'on trouve sur la carte Arduino.

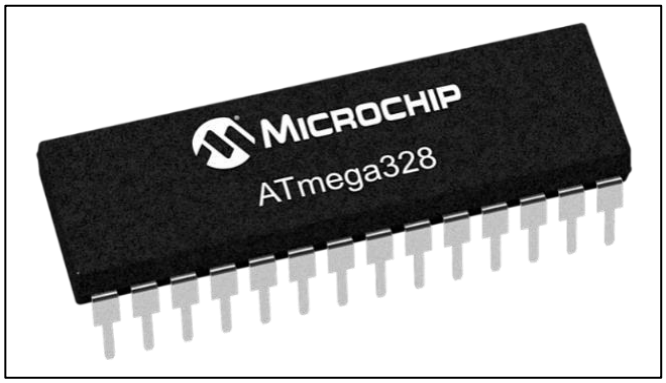


Figure I.3 Le microcontrôleur ATmega328 [17]

Le microcontrôleur ATMega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko.

- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface en coupant l'alimentation du microcontrôleur.

- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [4]

I.8.1.2 Les sources de l'alimentation de la carte :

On peut distinguer deux genres de sources d'alimentation (Entrée/Sortie) et cela comme suit :

- **VIN :** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe. On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le câble USB, accéder à la tension d'alimentation sur cette broche.

- **5V :** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

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

- **GND :** les épingles de la terre.

I.8.1.3 Les entrées & sorties de la carte Arduino NANO :

Chacune des 14 broches numériques sur la NANO peut être utilisé comme une entrée ou une sortie, en utilisant pinMode (), digitalWrite () et les fonctions digitalRead (). Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir un maximum de 40 mA et a une résistance de pull-up interne (déconnecté par défaut) de 20-50 kOhms. En outre, certaines broches ont des fonctions spécialisées :

Broche	Fonction
Série	0 (RX) et 1 (TX). Utilisé pour recevoir (RX) et transmettre (TX) des données série TTL. Ces broches sont connectées aux broches correspondantes de la puce série FTDI USB-TTL.
Interruptions externes	2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, un front montant ou descendant ou une modification de valeur.
PWM	3, 5, 6, 9, 10 et 11. Sortie PWM 8 bits avec la fonction analogWrite ().
SPI	10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches prennent en charge la communication SPI.
LED 13	Il y a une LED intégrée connectée à la broche 13 numérique.
Reset	passez cette ligne à LOW pour redémarrer le microcontrôleur. habituellement utilisé pour rajouter une fonction ou un bouton de reset directement sur les shields/modules.
AREF	Tension de référence pour les entrées analogiques. Utilisé avec analogReference
Entrées analogiques	La carte dispose pour cela de 8 entrées, repérées de A0 à A7, qui peuvent admettre toute tension analogique comprise entre 0 et 5 V.

Tableau I.3 Les broches spéciales de la carte Arduino [4]

La carte Arduino Nano intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [2]

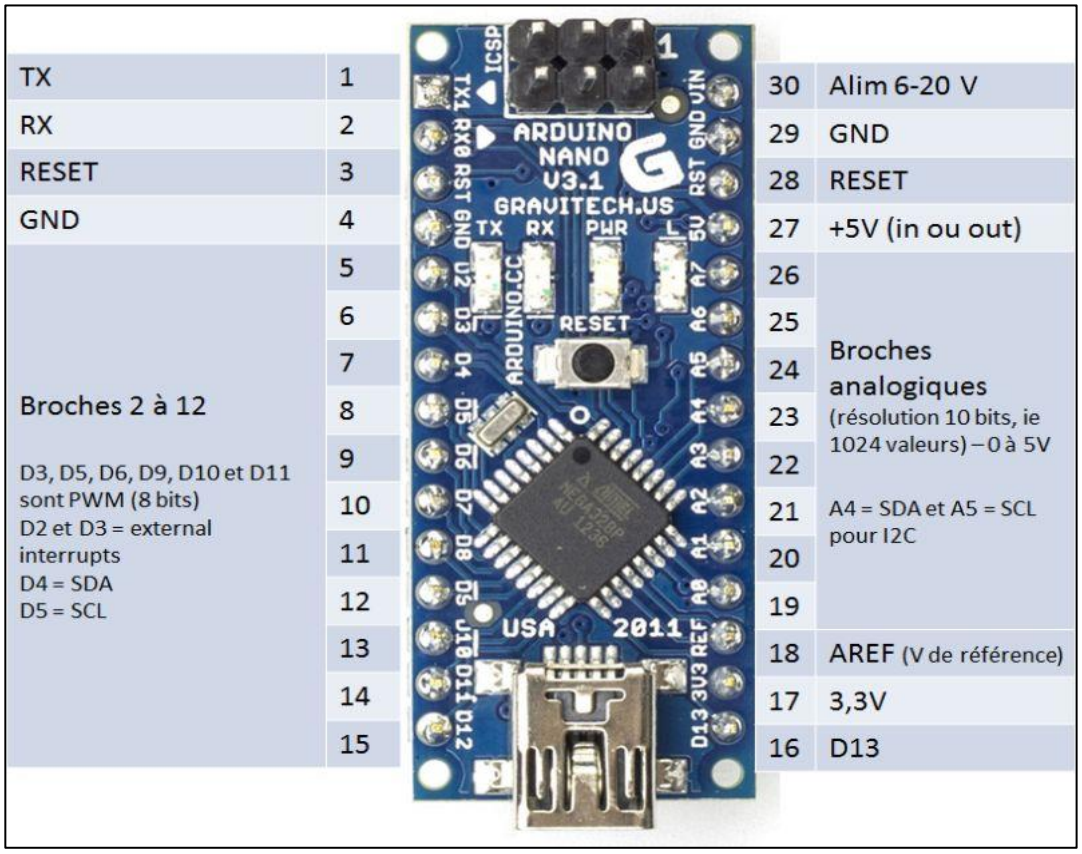


Figure I.4 Les broches de l'Arduino NANO [10]

I.8.2 Partie programme software :

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

I.8.2.1 L'environnement de la programmation :

Le 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 ce programme appelé IDE Arduino. [7]

I.8.2.2 Structure générale du programme (IDE Arduino) :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

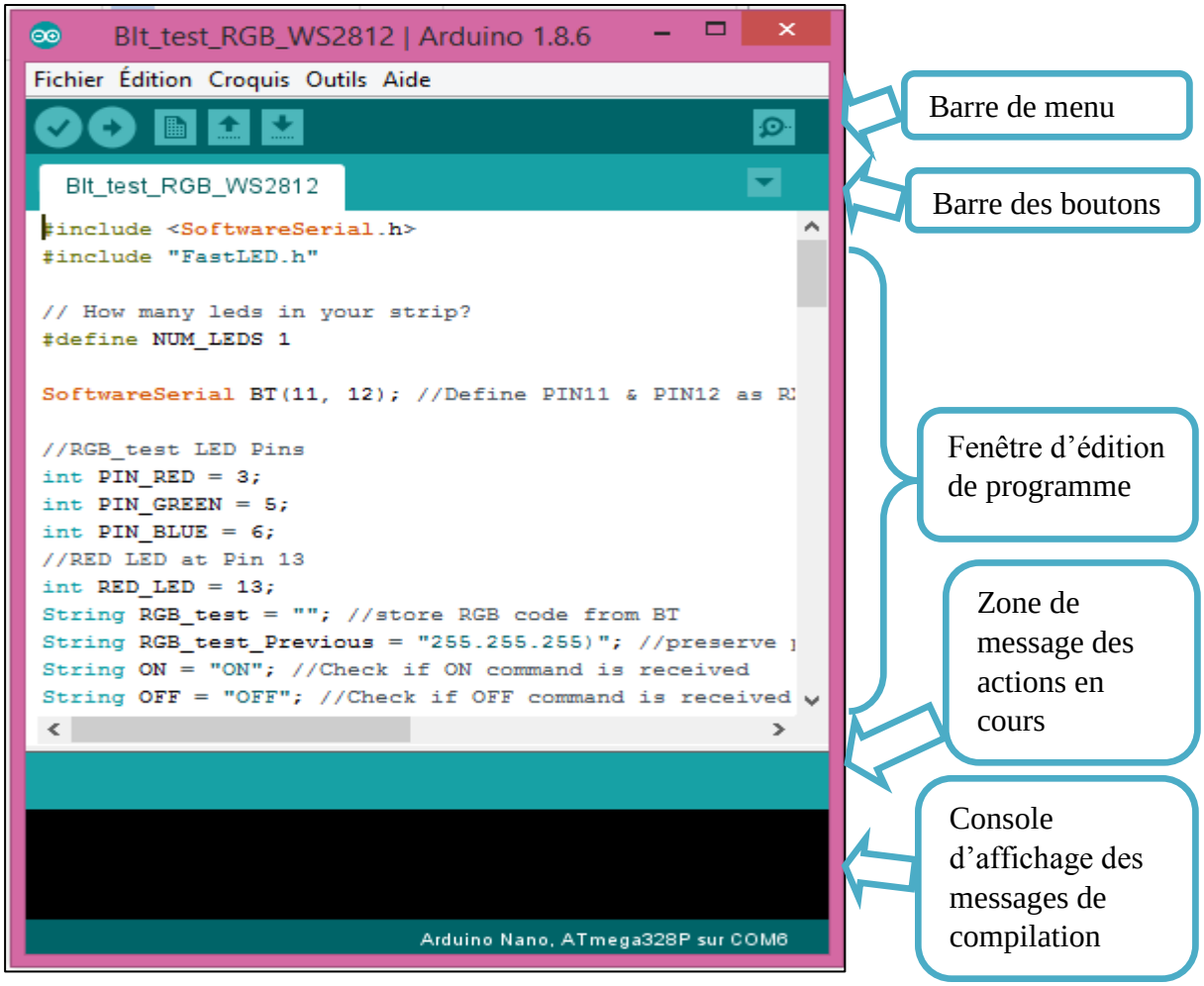
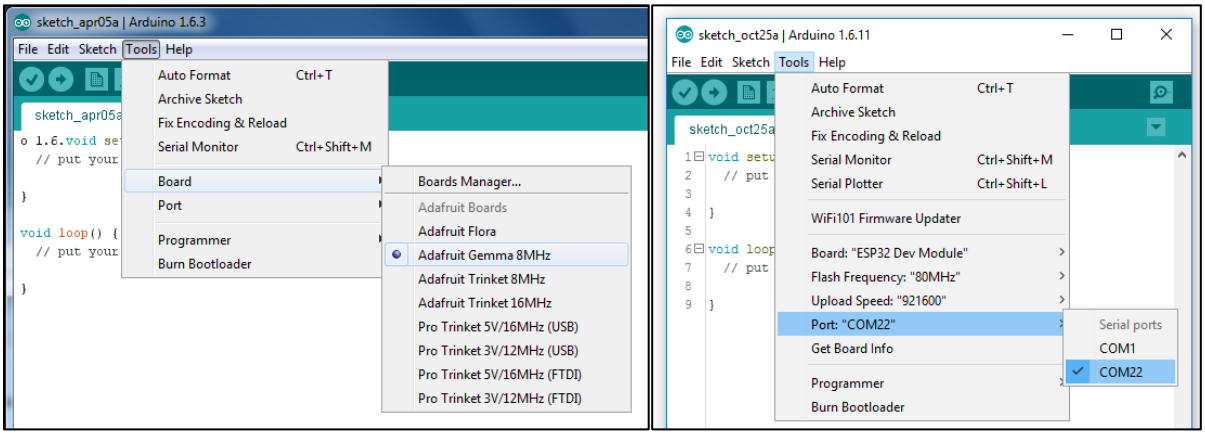


Figure I.5 Interface IDE Arduino

I.8.2.3 Injection du programme :

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte Arduino et le numéro de port USB (COM) comme, à titre d'exemple la figure 1.6 le

montre :



Phase 1 Phase 2

Figure I.6 Paramétrage de la carte

I.8.2.4 Description du programme :

Un programme arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

I.8.2.5 Les étapes de téléchargement du programme :

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

- On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
- On vérifie ce programme avec le logiciel Arduino (compilation).
- Si des erreurs sont signalées, on modifie le programme.
- On charge le programme sur la carte.
- On câble le montage électronique.
- L'exécution du programme est automatique après quelques secondes.
- On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (Pile 9v)
- On vérifie que notre montage fonctionne.

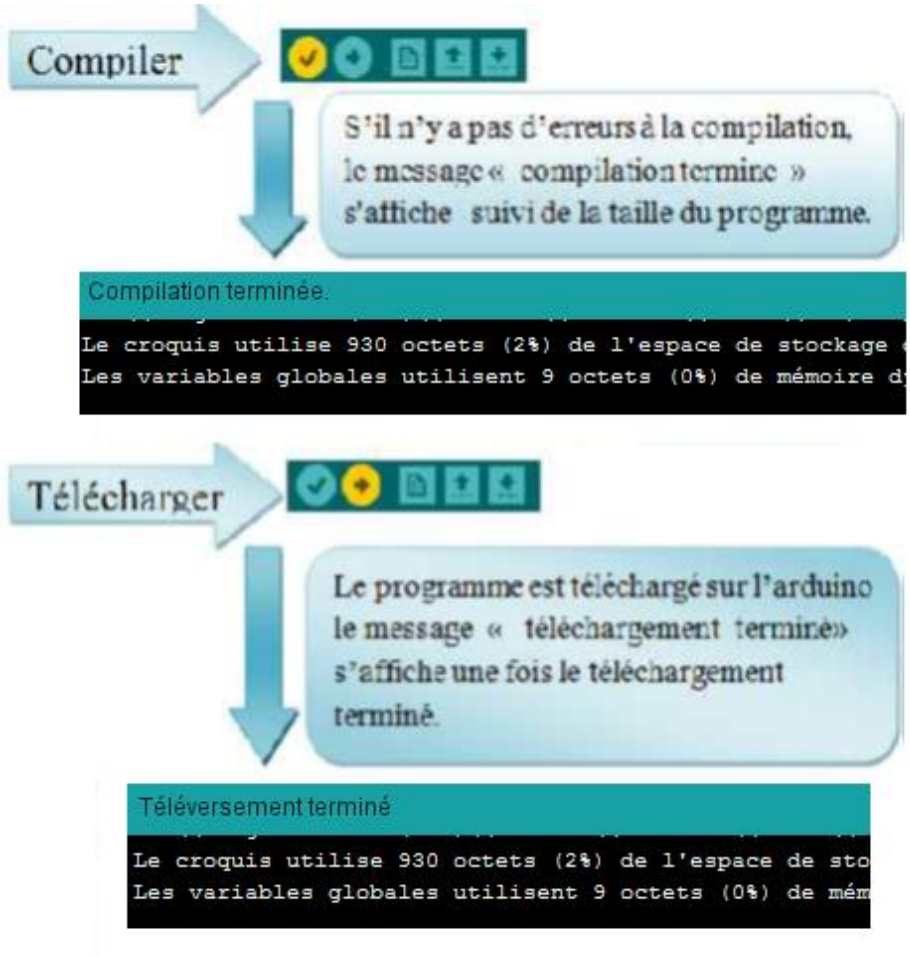


Figure I.7 Les étapes de téléchargement du code

I.9 Les Accessoires de la carte Arduino :

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

I.9.1 Communication :

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications, voilà quelques types :

I.9.2 Le module Arduino Bluetooth :

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion sérielle Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.

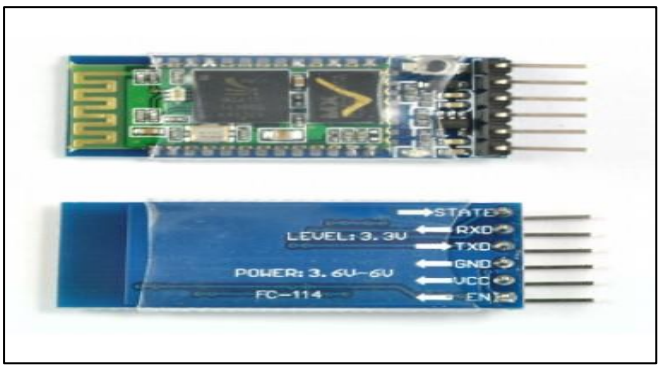


Figure I.8 Module Bluetooth pour Arduino [11]

I.9.3 Le module Arduino Wifi :

Le module Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.

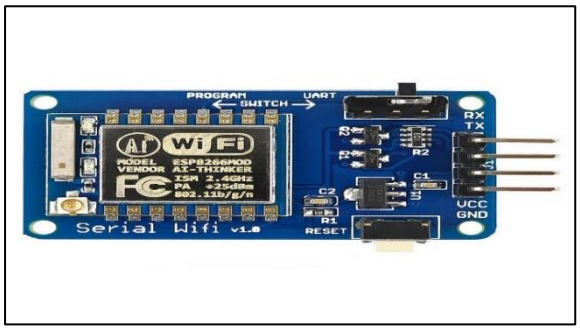


Figure I.9 Module WIFI [12]

1.10 Comparaison entre les supports de transmissions :

Afin que les informations circulent au sein d'un réseau, il est nécessaire de relier les différentes unités de communications à l'aide d'un support de transmission. Un support de transmission est un canal physique qui permet de relier des ordinateurs et des périphériques. Les supports de transmission les plus utilisés sont : les câbles, la fibre optique et les systèmes sans fil. Le câble est le type de support de transmission le plus ancien, mais aussi le plus utilisé et le moins cher. La fibre optique est un support de transmission très utilisé dans les réseaux de grandes tailles. Le principe de la fibre optique est d'acheminer des informations en envoyant des signaux lumineux dans un conducteur central en verre ou en plastique. Cette solution permet de transmettre très rapidement des informations, mais coûte encore cher. Sur les réseaux où les ordinateurs sont distants ou ne peuvent être connectés physiquement, la solution consiste à utiliser un support de transmission sans fil. Ces réseaux sans fil utilisent généralement : les rayons infrarouges, les micro-ondes, les ondes radio ou encore un satellite. [14] Dans notre projet on s'intéresse aux supports de transmissions sans fil. On se contentera de citer les deux types les plus utilisé et de faire une comparaison à la fin.

1.10.1 Infrarouge :

L'infrarouge est un faisceau de lumière. Les transmissions en infrarouge doivent être très intenses afin qu'il n'y ait pas de confusion avec les nombreuses sources de lumière qui existent dans une pièce (fenêtres, néons, télévision). La lumière infrarouge a une large bande passante, les débits sont relativement importants, mais la portée est faible (10 Mb/s, 30 m) Un réseau infrarouge est commode, rapide, mais sensible aux interférences lumineuses. [15]

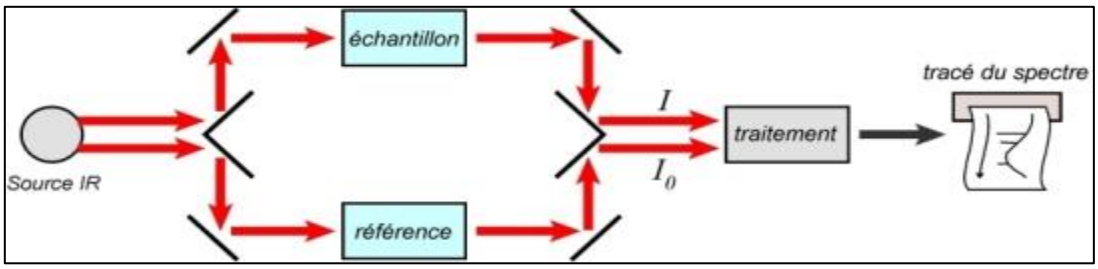


Figure I.10 Principe de fonctionnement de l'Infrarouge.

1.10.2 Bluetooth :

Bluetooth (la dent bleue), la technologie sans fil Bluetooth permet de partager des fichiers. Le standard Bluetooth est basé sur un mode de fonctionnement maître/esclave. Ainsi, on appelle « **picoréseau** » (en anglais **piconet**) le réseau formé par un périphérique et tous les périphériques présents dans son rayon de portée. Il peut coexister jusqu'à 10 picoréseaux dans une même zone de couverture. Un maître peut être connecté simultanément à un maximum de 7 périphériques esclaves actifs. En effet, les périphériques d'un picoréseau possèdent une adresse logique de 3 bits, ce qui permet un maximum de 8 appareils. Les appareils sont synchronisés mais ne possèdent pas d'adresse physique dans le picoréseau.

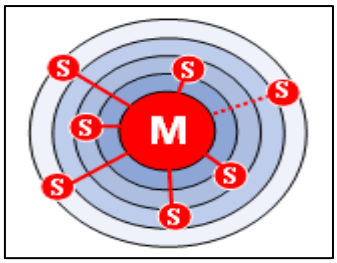


Figure I.11 Principe de fonctionnement du Bluetooth. [18]

En réalité, à un instant donné, le périphérique maître ne peut se connecter qu'à un seul esclave à la fois. Il commute donc très rapidement d'un esclave à un autre afin de donner l'illusion d'une connexion simultanée à l'ensemble des périphériques esclaves.

Le standard Bluetooth prévoit la possibilité de relier deux piconets entre eux afin de former un réseau élargi, appelé « **réseau chaîné** » (en anglais *scatternet*), grâce à certains périphériques faisant office de pont entre les deux piconets.[15]

1.10.3 Comparaison entre le Bluetooth et l'infrarouge :

	Avantages	Inconvénients
Bluetooth	- Fiabilité et facilité de l'utilisation. - Deux appareils équipés Bluetooth peuvent communiquer et échanger des données dans un rayon de dix mètres. - Promu par plus de 1000 industriels. - Faible encombrement. - Faible coût. - Une importante zone de couverture.	- Problème d'identification entre les périphériques de différentes marques. - Consommation importante de l'énergie. - Nombre de périphérique connectés limité.
Infrarouge	- Faible prix par rapport au Bluetooth. - Efficacité et facilité de mise en œuvre et de maintenance. - La technique infrarouge permet également la confidentialité des transmissions.	- Faible zone couverture. - Sensibilité aux interférences lumineuses. - Nombre de périphérique connecté (2 périphérique).

Tableau I.4 Comparaison entre les supports

I.11 Conclusion :

Ce premier chapitre, est l'introduction des outils de notre projet pour la réalisation d'une horloge numérique. Les systèmes embarqués et plus précisément l'Arduino seront utilisés ultérieurement dans l'étude de notre système. Dans ce chapitre nous avons donné les différents types de carte, ensuite, nous avons expliqué les deux parties essentielles de l'Arduino (la partie matérielle et la partie programmation). Nous avons également expliqué le principe de fonctionnement de la carte Arduino, sans oublier ses caractéristiques ainsi que quelques descriptions théoriques des accessoires de communication tels que le Wifi, Bluetooth et l'infrarouge. Le chapitre suivant sera consacré à la simulation et à la programmation d'une horloge numérique à l'aide de l'arduino.

Chapitre II

Etude et Simulation de l’horloge

Etude et Simulation de l'horloge

II.1 Introduction :

Dans ce chapitre, on présentera la simulation d'une horloge numérique sous Proteus. Après avoir donné dans le chapitre précédent une description théorique et générale sur l'environnement Arduino. Dans ce chapitre on se consacrera à l'étude et à la simulation expérimentale d'une horloge numérique. Pour parvenir à cela plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison avec le module Arduino et son environnement de développement IDE.

II.2 Principe de fonctionnement :

On utilisera dans notre projet une horloge temps réel RTC (Real Time Clock) et un Arduino. L'avantage de cette horloge est de conserver en permanence très longtemps la date et l'heure exactes même en cas de coupure de l'alimentation de l'Arduino. Il faut juste prévoir une initialisation avant l'utilisation de l'horloge RTC pour avoir la bonne date et la bonne heure. Apres avoir programmé le module RTC avec l'Arduino nous obtenons une horloge qui s'affiche grâce à un afficheur.

II.3 Déroulement de la simulation de l'horloge :

La simulation de l'horloge se déroulera en trois parties :

- La première partie est de faire le schéma de brochages électronique.
- La deuxième partie est la simulation avec PROTEUS de l'horloge.
- La troisième partie est la programmation de l'Arduino sous l'IDE.

Pour la partie programmation, on passe par :

- Programmation de la carte Arduino.
- Visualisations des résultats

II.4 Schéma synoptique général :

Le programme de l'horloge sera injecté à la carte Arduino par un ordinateur. La carte à son tour envoie une demande au module RTC pour qu'il lui transmet l'heure, une fois la carte Arduino a l'heure, elle convertie les informations reçus du RTC d'une manière à ce qu'il s'affiche en décimal sur l'afficheur. Les boutons poussoirs à leurs tours servent à ajuster l'heure.

Le schéma synoptique général de la simulation de l'heure est indiqué par la figure (II.1) :

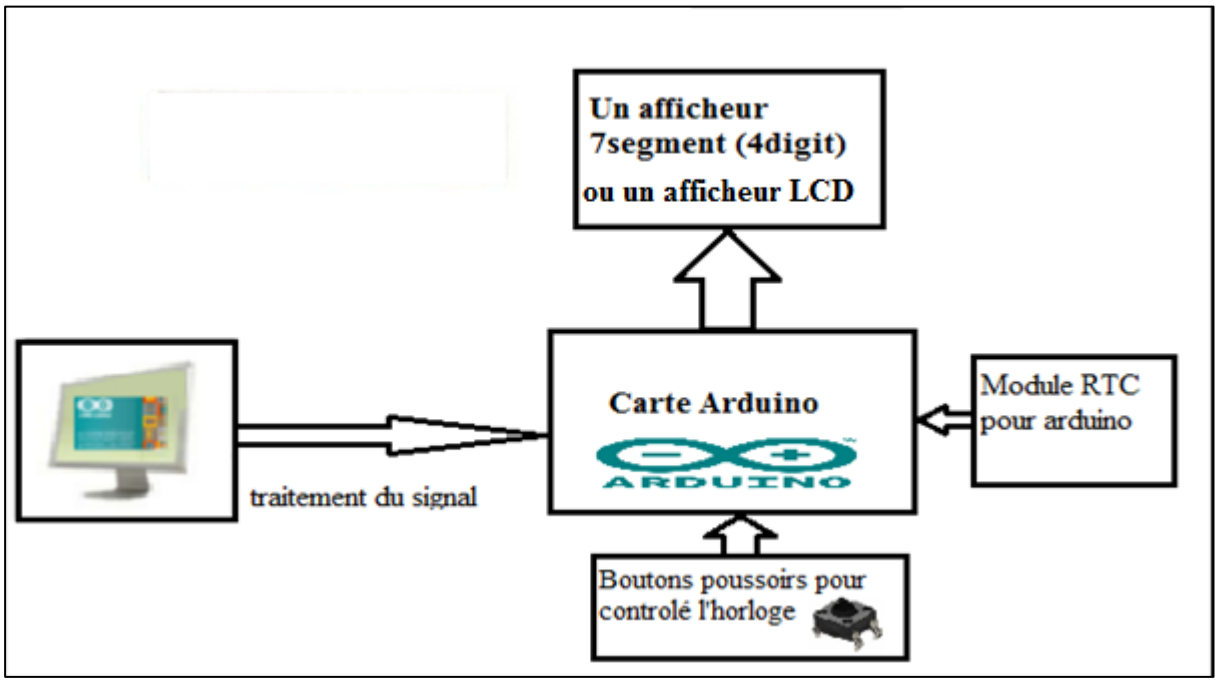


Figure II.1 Schéma synoptique d'horloge numérique

II.5 Simulation du projet sous Proteus :

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception.

II.5.1 Présentation des outils de la simulation :

Avant de passé à la réalisation, nous avons procédé à une simulation, d'où nous avons travaillé avec plusieurs logiciels, principalement le logiciel Proteus pour simuler les circuits électroniques et le logiciel Arduino IDE pour programmer le circuit.

II.5.2 Démarche de la simulation :

Comme les librairies Arduino et RTC ne sont pas des bibliothèques officielles de Proteus, donc il faut les télécharger sur internet et de les ajouter à la bibliothèque de Proteus.

II.5.2.1 Bibliothèque Arduino pour Proteus :

Lorsque la librairie Arduino est télécharger d'internet, on ajoute les deux fichiers nommés « ArduinoTEP.LIB et ArduinoTEP.IDX » et on les place dans le dossier des bibliothèques de notre logiciel Proteus, une fois cette étape terminé on redémarre le logiciel.

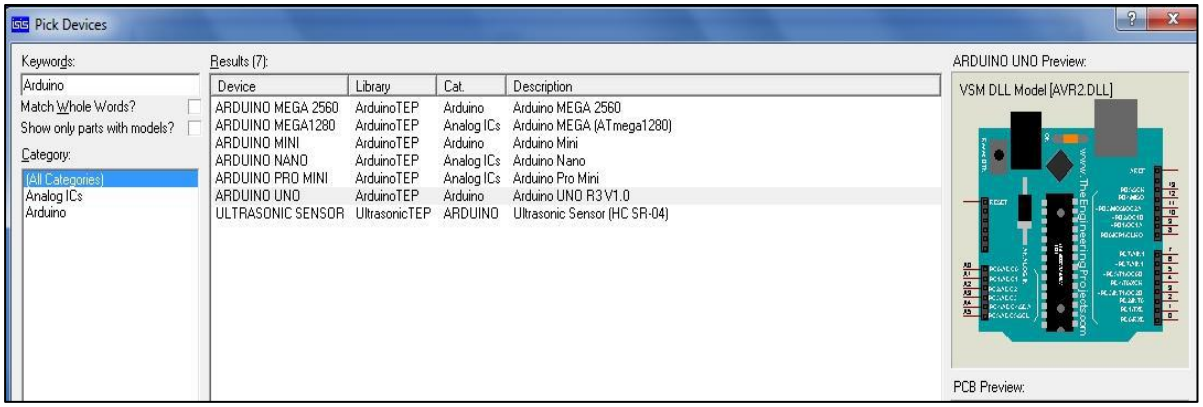


Figure II.2 Intégrations de la bibliothèque Arduino dans Proteus

II.5.2.2 Bibliothèque du module RTC pour Proteus :

Pour pouvoir utilisé le module RTC on télécharge une bibliothèque du module pour Proteus, on intègre les deux fichiers nommés « RTCModuleTEP.IDX, RTCModuleTEP.LIB » dans la bibliothèque du logiciel.

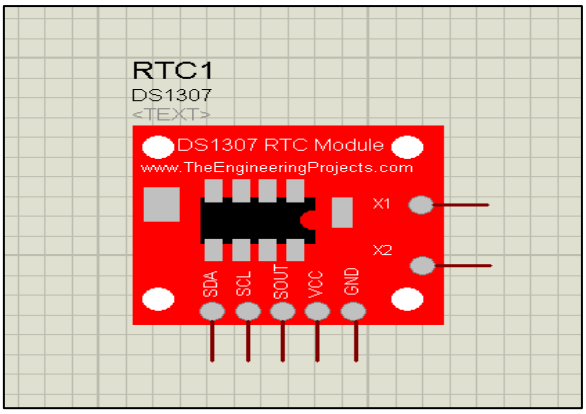


Figure II.3 Module RTC sous Proteus

II.5.2.3 Circuit global de la simulation :

La figure suivante et le circuit global de notre simulation sous Proteus.

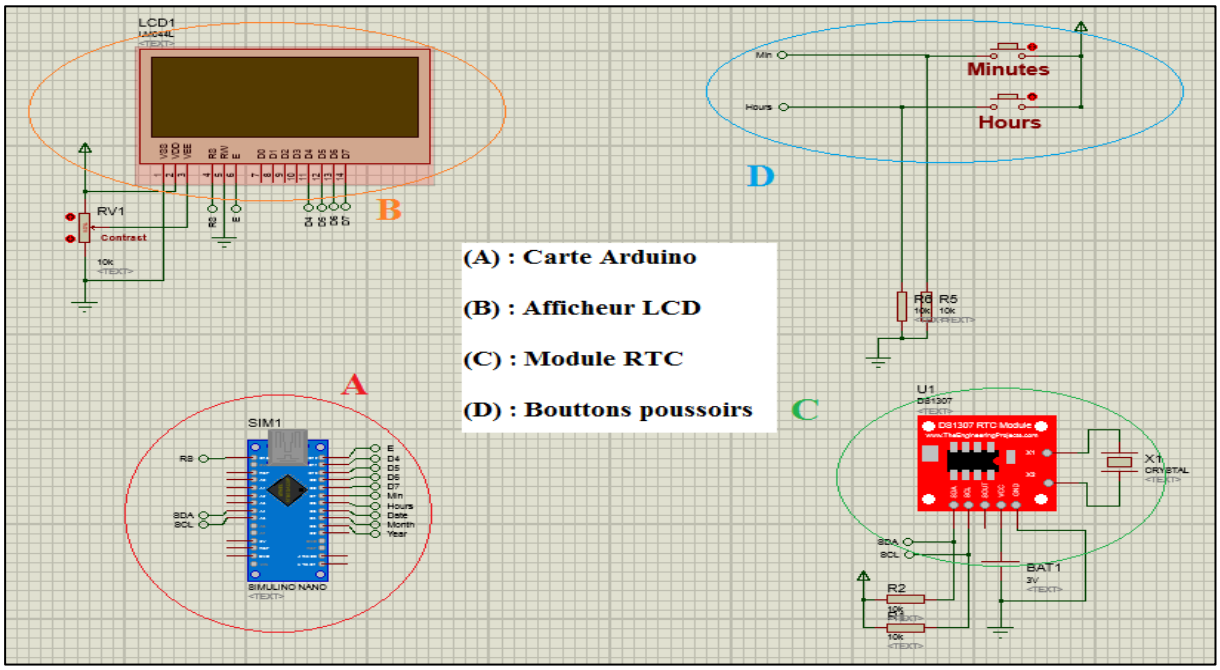


Figure II.4 Circuit global sous PROTEUS

Les différentes connexions et numéro des pins de circuit sont présentés dans le tableau (II.1) :

Module	Port arduino	Port du module	Alimentation
RTC	A4	SDA	5V GND
	A5	SCL	
Afficheur LCD	D2	1	5V GND
	D3	2	
	D4	3	
	D5	4	
	D12	5	
	D7	6	
	D8	11	
	D13	12	
	D11	13	
	D10	14	
BP(Minutes)	A2	1	GND
	GND	2	
BP2(Heures)	A1	1	
	GND	2	

Tableau II.1 Broche et connexion du circuit électronique

Remarque :

La figure II.4 ne représente pas le circuit global du projet car y'a des composants qui ne figure pas dans la bibliothèque de Proteus et qui seront utilisés dans la réalisation. Cette dernière sert à expliquer d'une manière globale le fonctionnement d'une horloge avec le module RTC Arduino.

II.6 Programmation de la carte Arduino :

Avant de procéder à la programmation de l'horloge sous le logiciel Arduino IDE, on doit tout d'abord ajouter les bibliothèques nécessaires pour les composants utilisées.

II.6.1 L'organigramme du code a injecté dans la carte :

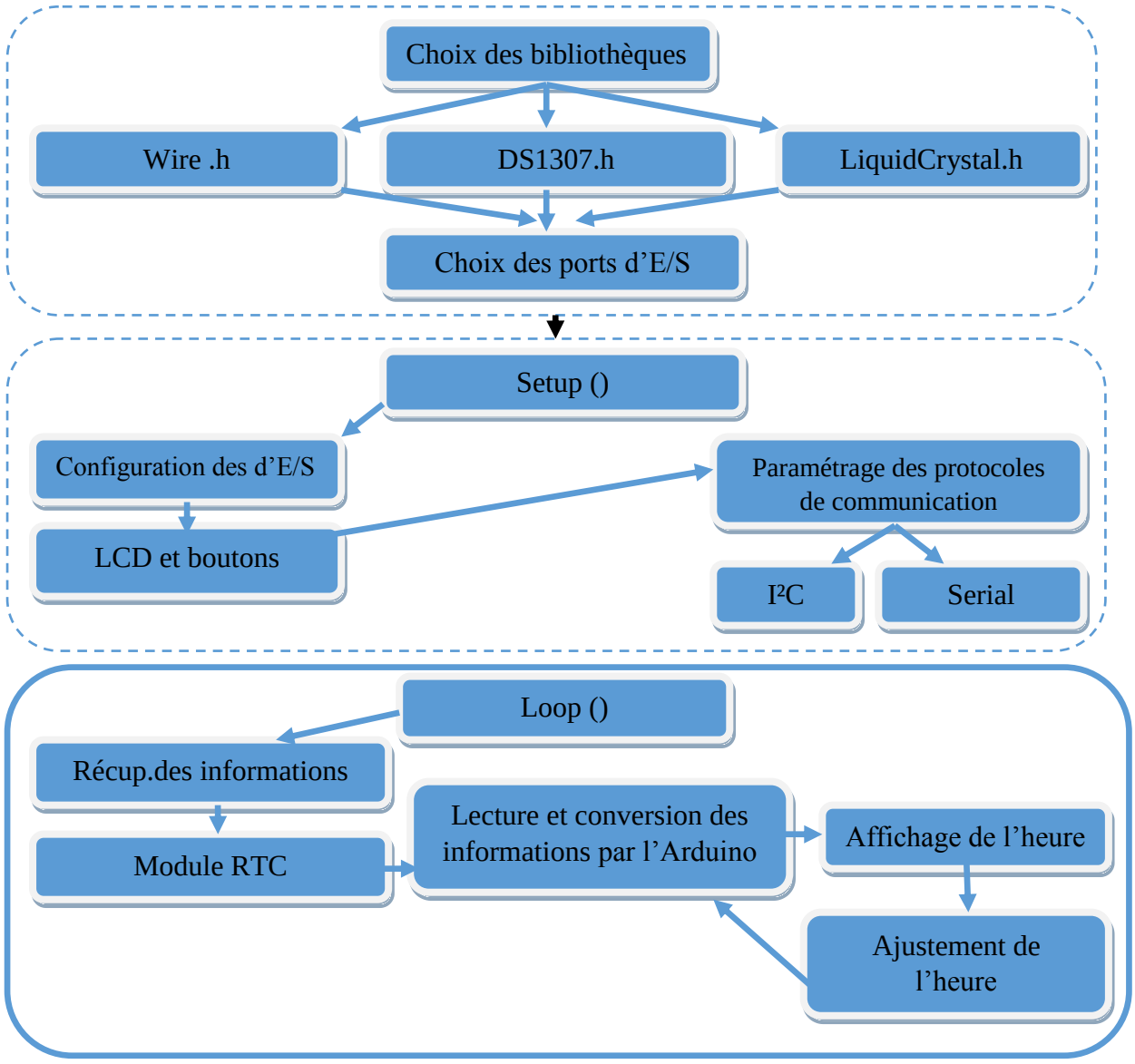


Figure II.5 Organigramme explicatif du code injecté à la carte

Comme tout programme (code) Arduino ce dernier est divisé en trois parties :

- **La première partie :** dans cette partie on déclare toutes les bibliothèques et les variables qui seront utilisées, dans notre programme nous avons fait appelle à plusieurs librairies dont on cite ;
  - ✓ Liquidcristal : pour contrôler l'écran LCD.
  - ✓ DS1307 : pour la lecture et l'écriture de l'heure.

  Après avoir déclaré les bibliothèques à utiliser, on commence la configuration des ports de connexion des modules sur la carte.

- **La deuxième partie :** Il s'agit de la fonction d'initialisation de l'Arduino 'Setup ()'. C'est ici qu'on va paramétrer nos protocoles de communications, la configuration de nos E/S.

- **La troisième partie :** La fonction loop () est la 2ème fonction à être lancée sur l'Arduino. En effet, cette fonction reboucle sur elle-même. Lorsque la dernière ligne de la fonction loop () sera atteinte par le programme, ce dernier remontera à la première ligne. Dans notre programme on commence cette partie par la récupération des données de tous les modules, ensuite ces données serons traduites en décimal pour l'affichage de l'heure sur l'afficheur. Puis commence la déclaration des instructions qui vont permettre aux boutons poussoirs de régler l'heure, enfin les données seront transmises à la carte Arduino et à son tour la carte traduit les informations reçus en décimal et les afficher sur l'écran.

Remarque :

L'organigramme sert à expliquer d'une manière générale comment programmer une carte Arduino avec un module RTC, un afficheur et les buttons poussoirs pour afficher l'heure.

II.7 Visualisation des résultats sous Proteus :

II.7.1 Injection du programme dans la carte Arduino :

Avant de faire la simulation avec Proteus il faut suivre les étapes suivantes :

- Copier l'adresse de fichier de compilation (voir le premier chapitre).
- Ouvrir le fichier de simulation.

- En Appuyant deux fois sur la carte arduino avec le bouton gauche de la souris, une fenêtre apparait comme le démontre la figure (II.6) :

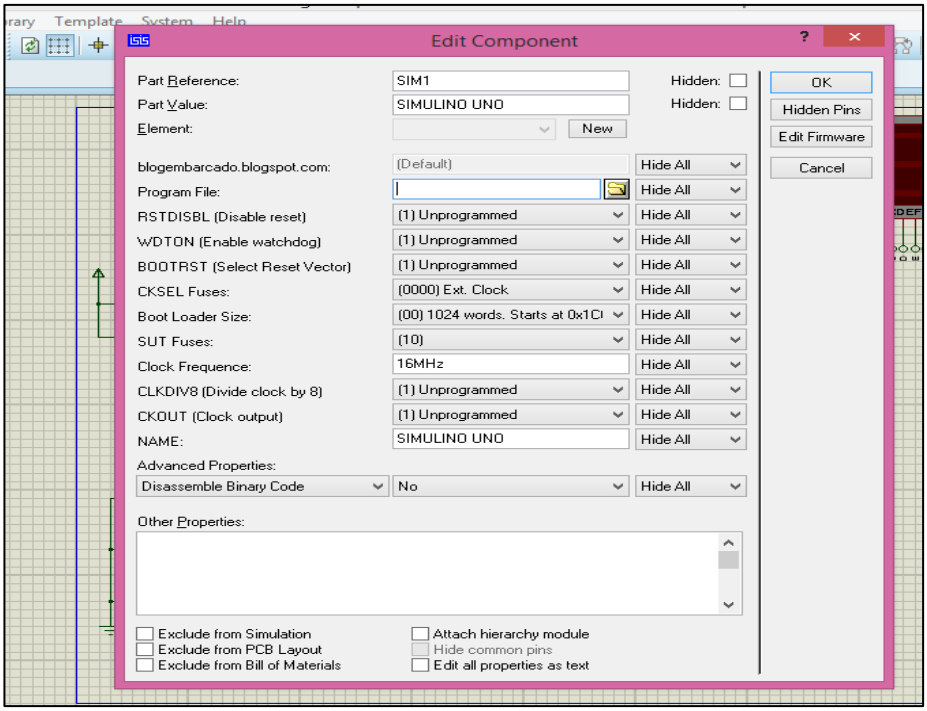


Figure II.6 Fenêtres de configuration de la carte Arduino

- Dans la case Program File on colle l'adresse de fichier de compilation comme le démontre la figure II.7

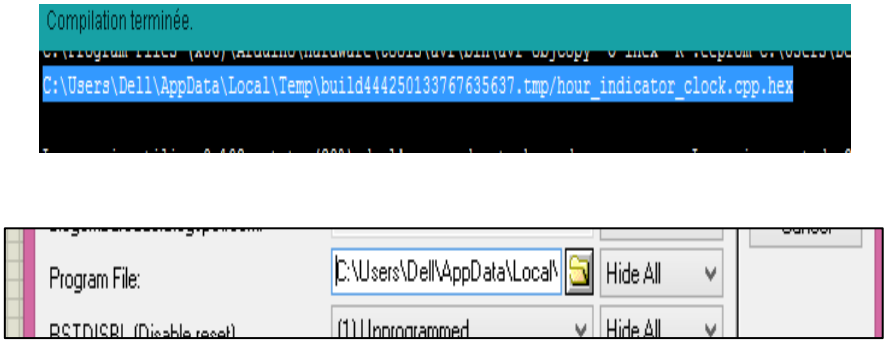


Figure II.7 Injection du programme dans la carte Arduino

- A la fin on peut visualiser la simulation on appuyant sur l'icône ''run the simulation''.

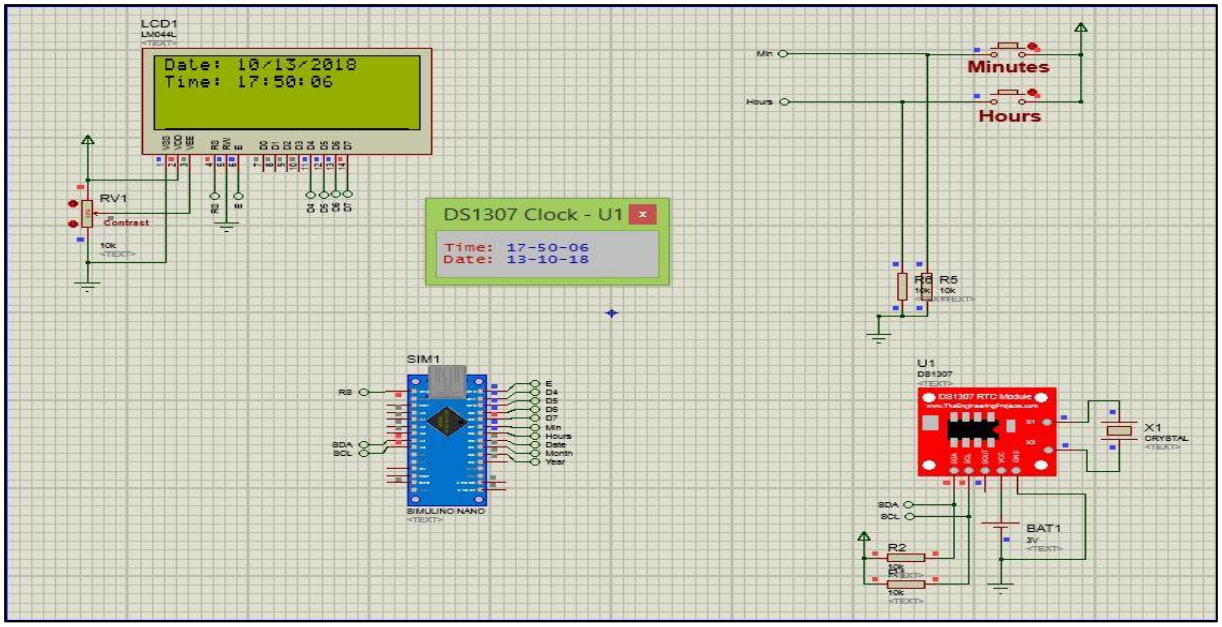


Figure II.8 La simulation de l'horloge

Après plusieurs essais de simulation, nous avons obtenu le résultat voulu, comme le démontre la figure (II.8).

II.8 Conclusion :

Ce chapitre englobe les différentes étapes de la réalisation virtuelle, nécessaire à la visualisation du résultat sous Proteus, qui commence par l'ajout des bibliothèques nécessaires pour la simulation d'une horloge et la connaissance des composants majeurs a utilisé. Ensuite on s'est consacré au circuit global et les différents brochages reliant l'Arduino, le module RTC, les boutons poussoirs et l'afficheur. Puis, nous avons donné un organigramme explicatif du code injecté à l'Arduino. À la fin nous avons fait des essais sur le circuit global sous Proteus. Le chapitre suivant sera consacré à la réalisation pratique d'une horloge numérique contrôlé avec une application Android.

Chapitre III

Réalisation Pratique de l’horloge

Réalisation Pratique de l'Horloge

III.1 Introduction :

Après avoir montré dans le chapitre précédent le fonctionnement et la simulation d'une horloge à base d'Arduino. Dans ce chapitre, on présentera les différentes étapes de la réalisation de l'horloge numérique géante avec une carte arduino.

Ce travail à base d'une carte Arduino NANO permet de générer l'heure à l'aide d'un module RTC (horloge à temps réel) et d'une application Android qui gère l'heure et les couleurs d'affichage via un module Bluetooth connecter à la carte. On va procéder à différentes étapes qui nous permettrons la conception et la réalisation d'une horloge numérique : l'étude, le choix des composants nécessaires, les tests sur la plaquette d'essai, et enfin la soudure des composants.

III.2 Déroulement de la réalisation de l'horloge :

III.3 Composants utilisés :

Pour réaliser une horloge numérique il nous faut :

Carte arduino NANO V3.0 ou autres	
Un module RTC DS1307 ou autres	
Un module Bluetooth arduino	

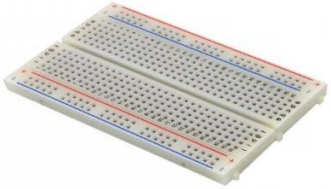
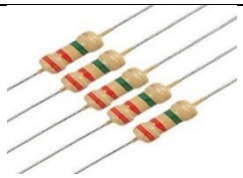
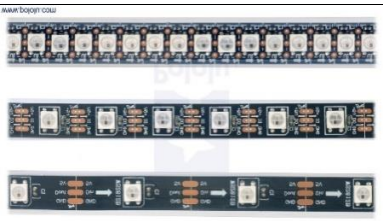
Des fils jumpers	
Boutons poussoirs	
Photorésistance	
Plaque d'essai	
Des résistances (100-500 Ω)	
Une alimentation 3.3V-5V 2A	
Ruban leds WS2812B ou WS2812	

Tableau III.1 Liste des composants utilisés

III.4 Schéma global du montage :

Ce schéma a été réalisé sous Fritzing .Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon développé par Interaction Design Lab Potsdam.

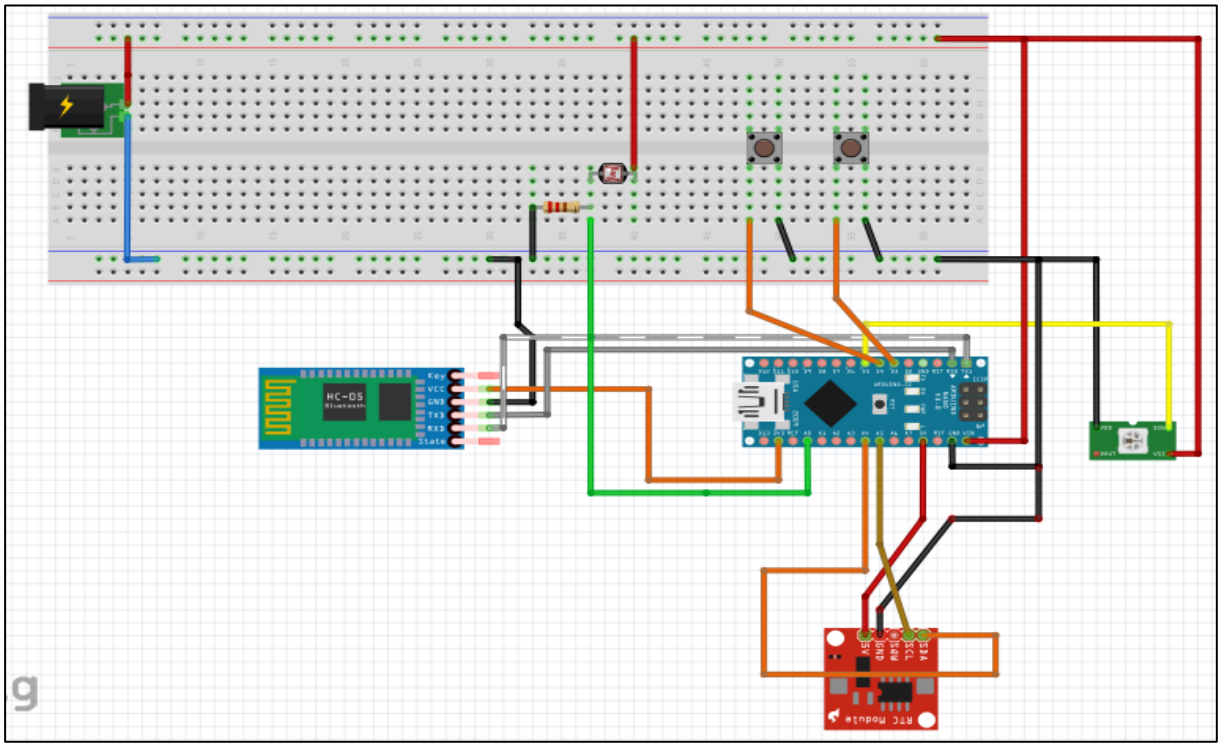


Figure III.1 Schéma global du montage électronique de l'horloge

Le brochage des différents composants :

Module	Port Arduino	Port du module	Alimentation
Leds WS2812B	D5	Din	5v GND
RTC (DS1302)	A4	SDA	5V GND
	A5	SCL	
Bluetooth	RX0	TXD	3V3 GND
	TX1	RXD	
BP1	D4	1	GND
	GND	2	
BP2	D3	1	GND
	GND	2	

Tableau III.2 Brochages des composants constituants le circuit

III.5 Partie atelier :

Dans cette partie on représentera tous les matériaux utilisés et les étapes de la réalisation du cadre de l'horloge ou le support.

III.5.1 Matériaux utilisés :

Pour réaliser le cadre il nous faut :

1. Du bois.
2. 100x50 cm du verre.
3. Papier collant utilisé dans la sérigraphie.
4. De la colle.
5. Du polystyrène expansé.
6. De la peinture noire.
7. Des clous

III.5.2 Fabrication du cadre :

Le cadre en bois a été fabriqué chez un menuisier le résultat peut être visualisé ci-dessous :

Figure III.2 Le cadre de l'horloge

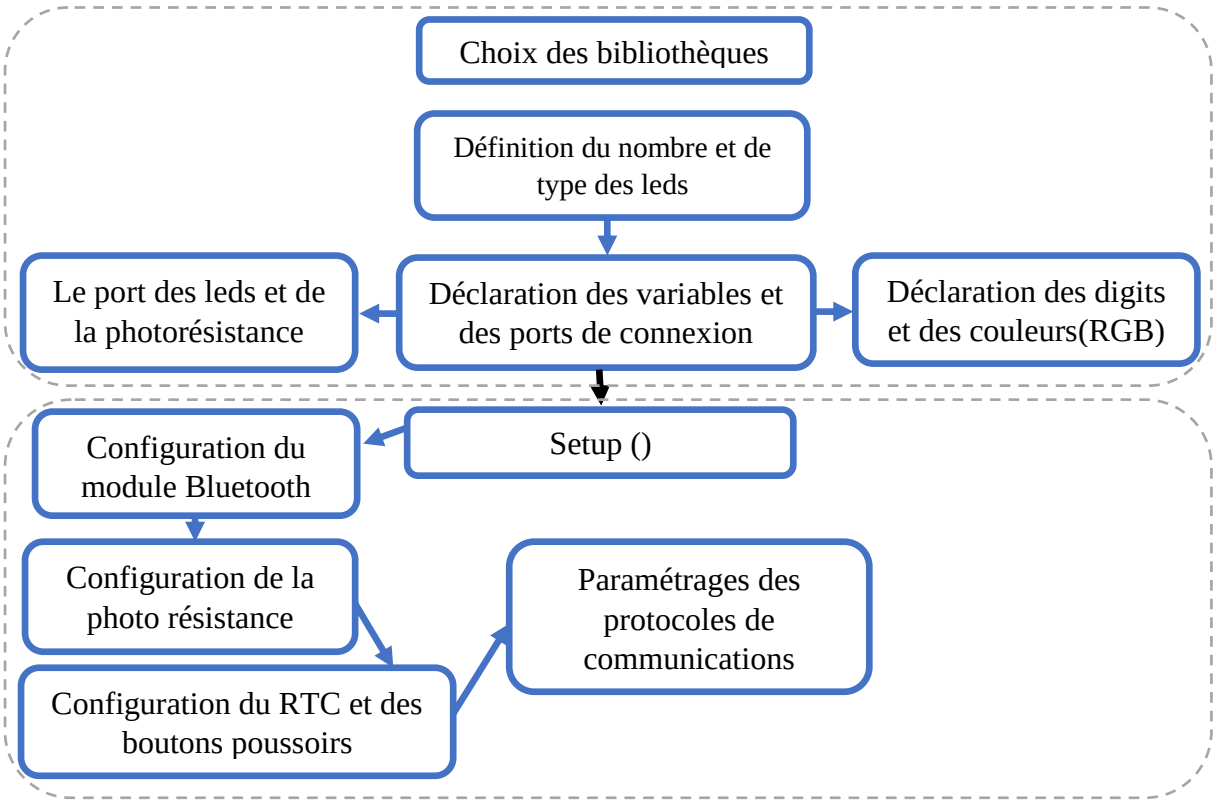
Une fois le cadre assemblé on l'a peint, mis le verre et découpé le polystyrène

Figure III.3 Montage finale du cadre

III.6 Partie programmation :

Dans cette partie on représentera le code final de l'horloge

III.6.1 Programmation de l'horloge :



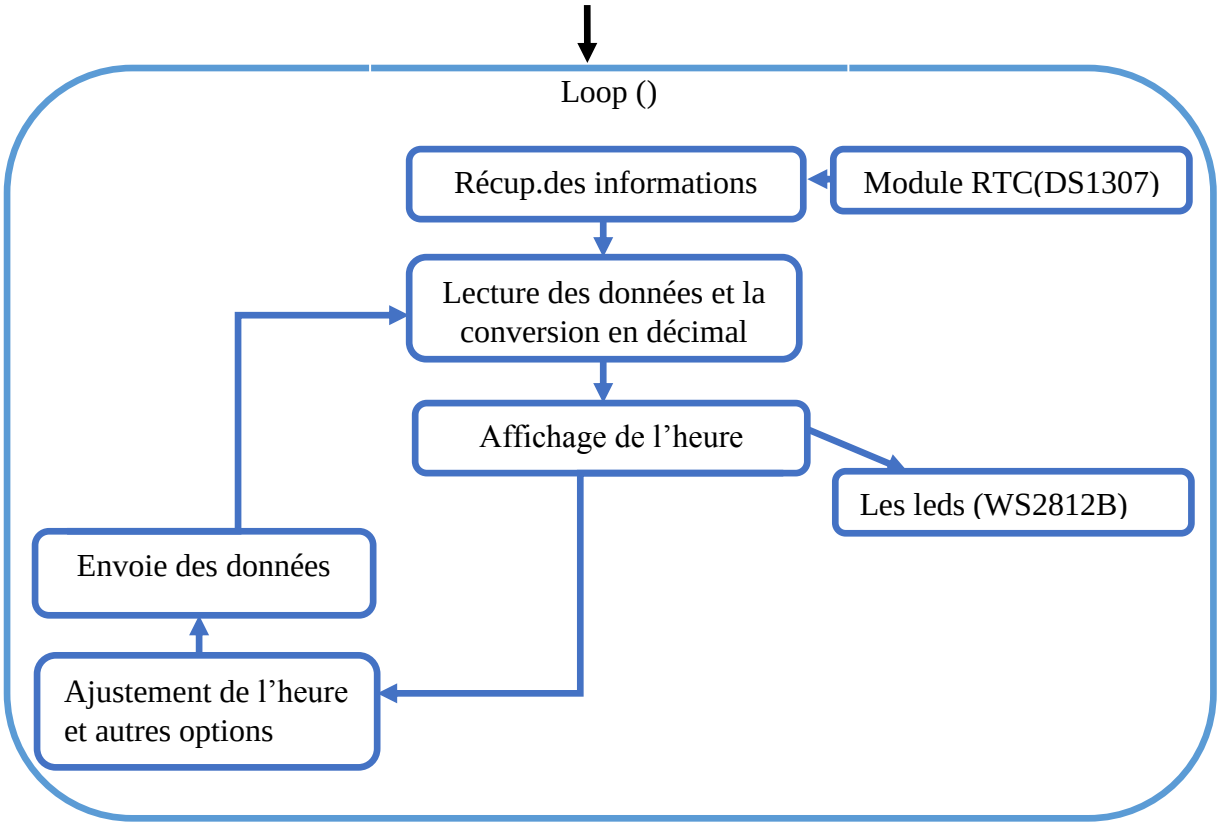


Figure III.4 Organigramme explicatif du code final de l'horloge

Comme tout programme (code) Arduino ce dernier est divisé en trois parties :

- **La première partie :** dans cette partie on déclare toutes les bibliothèques et les variables qui seront utilisées, dans notre programme nous avons fait appelle à plusieurs librairies dont on cite ;
  - ✓ Fastled : pour contrôler les leds WS2812B.
  - ✓ DS3232RTC : pour la lecture et l'écriture de l'heure.
  - ✓ Time : contrôler le temps.
  - ✓ SoftwareSerial : protocole de communication du module Bluetooth.

  Après avoir déclaré les bibliothèques à utiliser, on commence la déclaration des variables et des ports de connexion des modules sur la carte, par exemple :
  - ✓ Leds WS2812B pin (5).
  - ✓ Le nombre de leds pour chaque digit.
  - ✓ La photorésistance pin A0.

- **La deuxième partie :** Il s'agit de la fonction d'initialisation de l'Arduino 'Setup()'. C'est la première fonction qui va être lancée au démarrage de l'Arduino. C'est ici qu'on va paramétrer nos différents objets, nos protocoles de communications, la

configuration de nos entrées/sorties, on ne peut faire appel à cette fonction qu'une seule fois au début du programme.

- **La troisième partie :** La fonction loop () est la 2ème fonction à être lancée sur l'Arduino, juste après la fonction Setup (). Cette fonction a la particularité, d'être une boucle. En effet, cette fonction reboucle sur elle-même. Lorsque la dernière ligne de la fonction loop () sera atteinte par le programme, ce dernier remontera à la première ligne. Dans notre programme on commence cette partie par la récupération des données de tous les modules, ensuite ces données serons traduites en décimal pour l'affichage de l'heure sur les leds. Puis commence la déclaration des instructions qui vont permettre au module Bluetooth et aux boutons poussoirs de régler l'heure, enfin les données seront transmises à la carte Arduino et à son la carte traduit les informations reçus en décimal et les afficher.

III.6.2 Création de l'application Android :

Pour pouvoir contrôler l'horloge à distance via un module Bluetooth, une application Android doit être programmée afin de pouvoir régler l'heure.

III.6.2.1 Présentation de l'outil de création :

App Inventor est un outil de développement en ligne pour les téléphones et les tablettes sous Android. App Inventor est un OS créé par Google. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés des années 80 et s'inspire des travaux d'une étudiante en thèse au MIT, Ricarose Roque. Le projet a été dirigé par le professeur Hal Abelson. [16]

III.6.2.2 Réalisation de l'application :

Pour réaliser l'application Android, on a utilisé le site MIT APP INVENTOR 2Elle est très simple d'utilisation, aucune ligne de code, que des blocks logiques, voici les étapes de la réalisation :

- pour la gestion du temps :

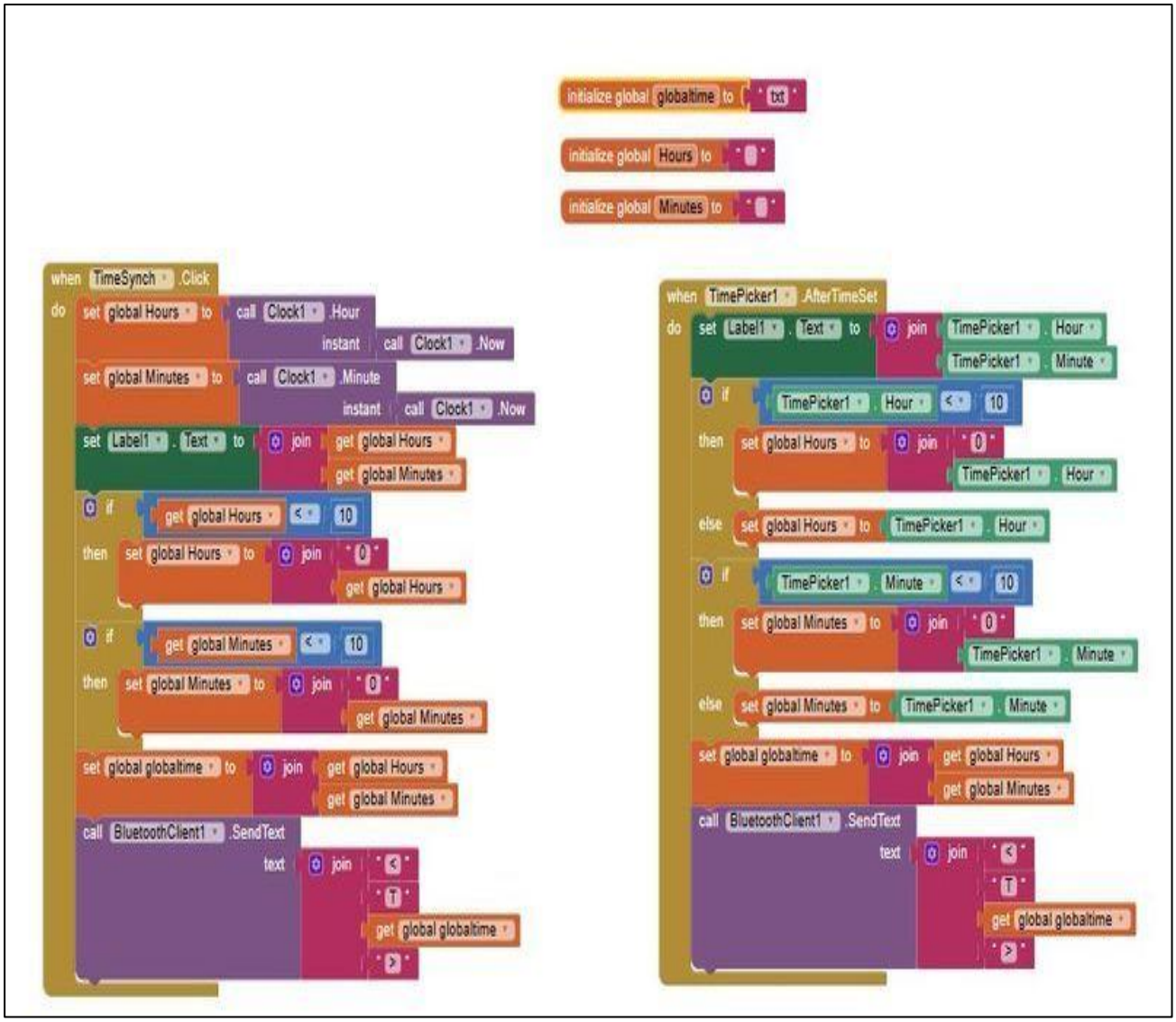


Figure III.5 Bloc de gestion du temps

Ce bloc de gestion de temps nous permet d'avoir le contrôle sur l'heure, par des fonctions bien clair tel que :

- ✓ Synchroniser l'horloge avec l'heure du portable.
- ✓ Régler l'horloge manuellement grâce à une liste des chiffres qui s'affichera sur l'écran du portable.

- Pour la gestion du Bluetooth :

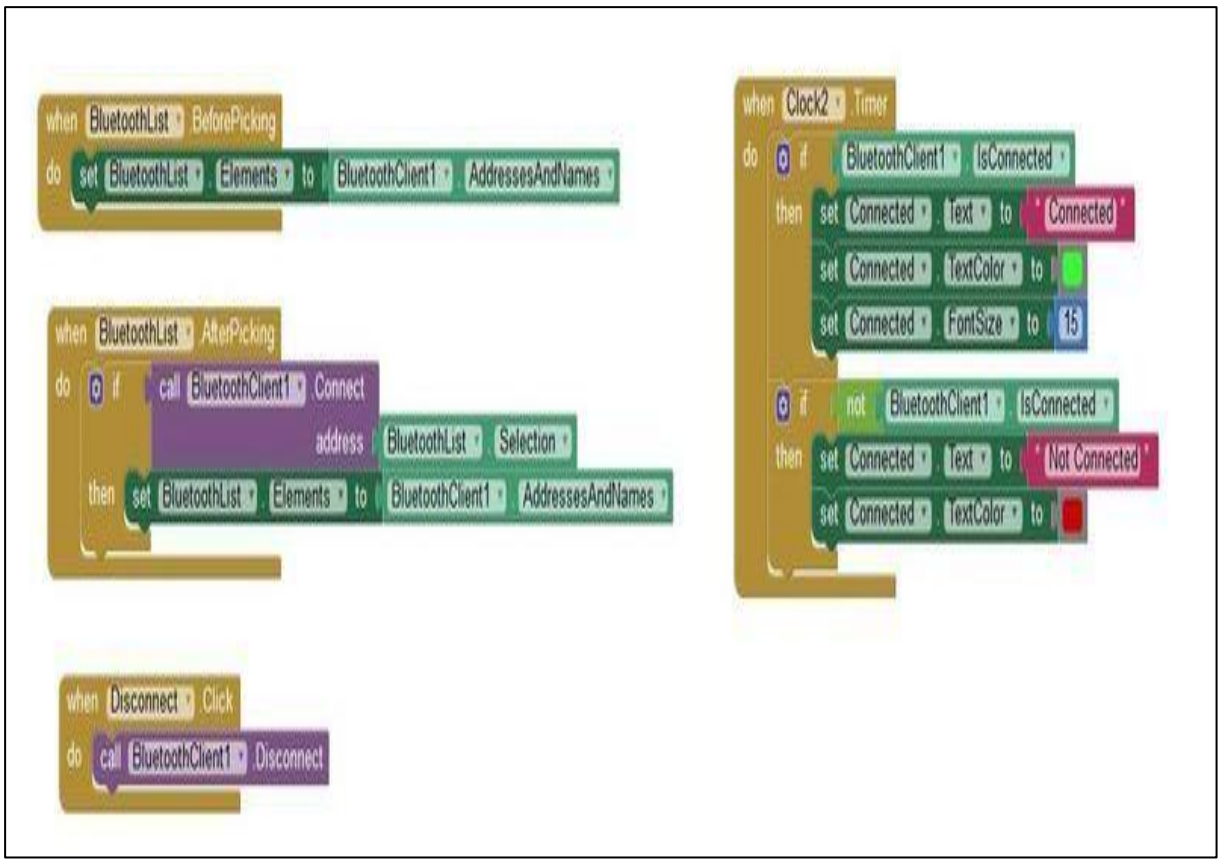


Figure III.6 Bloc de gestion du Bluetooth

Ce bloc permet au smartphone d'avoir le contrôle de l'horloge, et cela grâce un au module Bluetooth HC-05 connecter directement sur la carte Arduino.

- pour la gestion des couleurs :

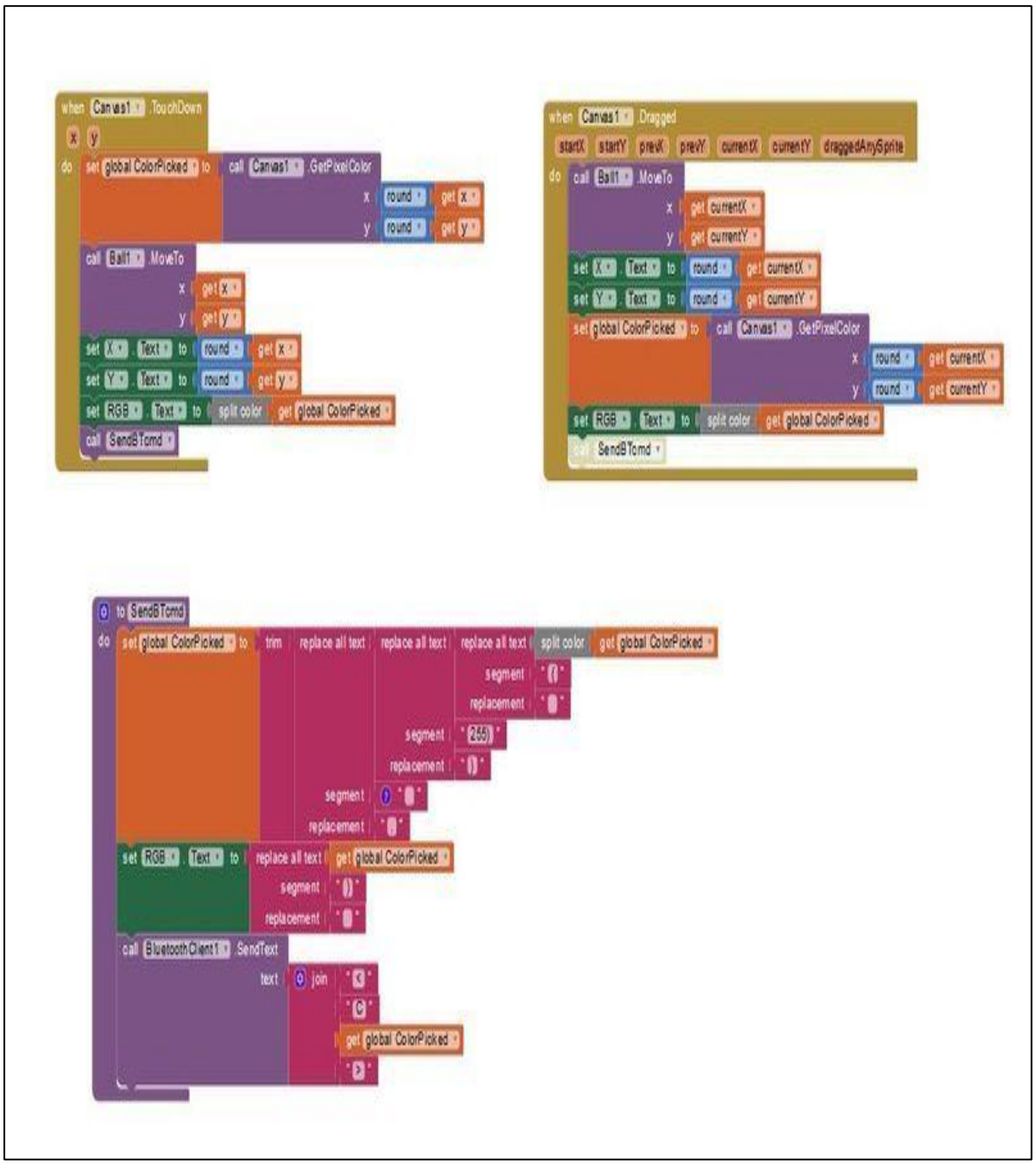


Figure III.7 Bloc de gestion des couleurs

Ce bloc nous permet de choisir la couleur de l'affichage de l'heure avec une variété importante de couleur.

- ✓ L'application finale :

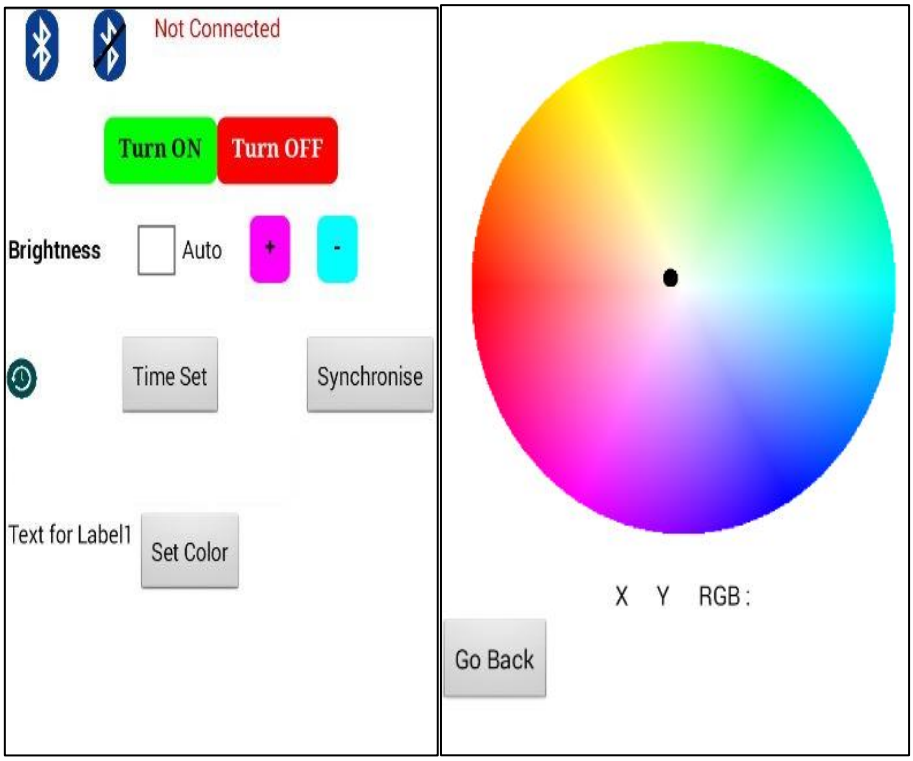


Figure III.8 L'application Android

III.7 Résultat de la réalisation :

Figure III.9 Résultat de la réalisation

III.8 Obstacle rencontré pendant la réalisation :

- le premier obstacle qu'on a rencontré était le financement du projet, sachant que les composants électronique sont cher est de disponibilité minoritaire, le projet a été autofinancé par le binôme créateur.
- la non disponibilité des composants nécessaire au niveau du laboratoire de la faculté (les résistances, les fils, les tins...).
- le matériel requis pour l'aboutissement du projet, le binôme a essayé d'aboutir au résultat escompté avec les moyens du bord.

III.9 Conclusion :

Ce chapitre, est la dernière partie de notre projet ou on a mis en épreuve notre partie théorique et la simulation sous Proteus. Pour avoir une horloge fini qui affiche l'heure correctement, afin d'avoir le résultat escompté on a commencé par fabriqué un cadre compléter par une plaque de verre floqué en 7segments qui compose la coqué de l'horloge, le polystyrène le circuit électronique le module Bluetooth ainsi que les LED ont été mis dans la partie intérieur de l'horloge, il était important de créé une application Bluetooth afin de contrôlé l'horloge à l'aide d'un Smartphone. On dernier lieu on a fait différents tests avec le matériel qu'on possède et les résultats sont plutôt encourageons.

Conclusion Générale

Conclusion Générale

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation de l'Arduino et de l'Android (programmation embarqué). Nous avons aussi appris à mieux gérer notre temps et nos équipes. Ce projet constitue en réalité la première étape d'un ensemble de projet. En effet l'objectif final est de faire disposer à la faculté FSSA un tableau d'affichage numérique géant. Ce tableau d'affichage permet, dans une première étape, d'afficher l'heure, puis la date, la température instantanée du jour, dans une deuxième étape. L'affichage des annonces importantes, tels des conférences, des colloques, des manifestations scientifiques en général, sera visé dans une troisième étape. L'affichage des périodes des examens, les notes obtenues des étudiants, sera l'ultime objectif de ce projet. Il servira à économiser les supports papiers, a gagner d'efficacité dans la gestion du temps et d'organisation de la faculté.

Ces points de repères peuvent aussi être les objets d'autres sujets de PFE dans un avenir à court terme

Ce travail reste, comme toute œuvre humaine, incomplet et perfectible, nous recommandons d'améliorer la conception, et pour cela nous proposons des améliorations a apporté qui commence par la réalisation d'un cadre avec une imprimante 3D, ensuite l'utilisation d'un module wifi pour permettre à l'horloge de se connecter au réseau de la faculté afin de pouvoir synchroniser plusieurs horloge, enfin l'amélioration de l'affichage grâce aux nouvelle LED WS2813.

Bibliographie

Bibliographie

[1] : https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm . Consulter le 26/03/2018 à 19H.

[2] : http://www.generationrobots.com/fr/152-arduino. Consulter le 26/03/2018 à 19H 30.

[3] : S.V.D REYVANTH, G.SHIRISH, D.SIVA VARMA, M. RAMU,2009-2013,'' Pid controller using arduino''.

[4] : C. Tavernier, « Arduino applications avancées ». Version Dunod, 2012.

[5] : www.mon-club-elec.fr. Consulter le 26/03/2018 à 20H.

[6] : http://www.acm.uiuc.edu/sigbot/tutorials/ 2009-11-17-arduino-basicsconsulter le 26/03/2018 à 23H 10.

[7] : Jean- Noël, «Initiation à la mise en oeuvre matérielle et logicielle de l'Arduino», novembre 2006. Centre de ressources art sensitif.

[8] : Eskimon, Olyte « Arduino pour bien commencer en électronique et en programmation », siteduzero, 2012.

[9] : A. Grimault, J. Querard « Article Procédé et dispositif de commutation d'un relais électromagnétique ».07/10/2009. Atlantic Industrie SAS

[10] : https://knowledge.parcours-performance.com/utiliser-arduino-nano/ consulter le 21/08/2018 à 22H 26.

[11] : https://www.openhacks.com/page/productos/id/1979/title/Bluetooth-module-HC-05#.W6Ebyfk6_IU consulter le 21/08/2018 à 22H 40.

[12] : https://www.dx.com/fr/p/esp-07-esp8266-serial-wi-f-wireless-module-w-built-in-antenna-compatible-with-3-3v-5v-for-arduino-400559#.W6EdiPk6_IU consulter le 21/08/2018 à 22H 42.

[13] : https://www.sparkfun.com/products/12571 consulter le 21/08/2018 à 22H 45.

[14] : http://www.hotosting.com/cresite/support-transmission.html consulter le 25/08/2018 à 9H 30.

[15] : http://jaures-col.spip.ac-rouen.fr/IMG/pdf/Transmissions_sans-fil.pdf consulter le 10/11/2018 à 9H 40.

[16] : http://sig.fgranotier.info/IMG/pdf/debuter_app_inventor.pdf consulter le 01/10/2018 à 23H.

[17] : www.atmel.com consulter le 20/10/2018 à 15H 50.

[18] : https://www.commentcamarche.net/contents/107-fonctionnement-du-bluetooth consulter le 10/11/2018 à 9H 50.

Résumé

Ce manuscrit est une présentation du projet, qui consiste à réaliser une horloge numérique géante au profit de la FSSA, qui permet à la communauté universitaire de consulter l'heure à longueur de journée. Ce PFE regroupe les différentes étapes qu'on a suivies pour réaliser une horloge numérique géante géré par une application Bluetooth à l'aide d'un Smartphone. L'Arduino, le module RTC, le module Bluetooth et les LEDS WS2812B sont utilisés dans la partie matérielle, quant à la simulation elle a été faite sous Proteus et la programmation sous IDE Arduino

Mots clés: Real Time Clock, Arduino, Bluetooth, LEDS WS2812B, Proteus, horloge.

Abstract

This manuscript is a presentation of the project, which consists in realizing a huge digital clock for the benefit of the FSSA, which allows the university community to consult the hour all day long. ARDUINO This PFE gathers the different steps followed to create a giant digital clock managed by a Bluetooth application has the help of a Smartphone. Arduino, RTC module, Bluetooth module and LEDS WS2812B are used in the part material, as for the simulation she was made under Proteus and the programming under IDE Arduino

Keywords: Real Time Clock, Arduino, Bluetooth, LEDS WS2812B, Proteus, Clock.

ملخص

هذه المخطوطة هي عرض تقديمي للمشروع، الذي يتكون من إنشاء ساعة رقمية عملاقة لصالح FSSA، مما يسمح للمجتمع الجامعي بالتشاور مع الوقت طوال اليوم. يجمع هذا المذكرة الخطوات المختلفة التي تم اتخاذها لتحقيق ساعة رقمية عملاقة يديرها تطبيق بلوتوث باستخدام هاتف ذكي. يتم استخدام اردوينو، وحدة الوقت الحقيقي على مدار الساعة، وحدة بلوتوث وLEDS WS2812B في جزء الأجهزة، أما بالنسبة للمحاكاة فقد تم إجراؤها تحت برمجة بروتيس وارد وينو IDE.

الكلمات الرئيسية: الوقت الحقيقي على مدار الساعة، اردوينو، بلوتوث، LEDS WS2812B، Proteus، الساعة .