Ordre...../F.S.S.A/UAMOB/2018

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE AKLI MOAND OULHADJE-BOUIRA

Faculté des Sciences et des Sciences Appliquées
Département : **Génie Electrique**

Mémoire de fin d'étude

Présenté par :
MANAMANI Yassine
En vue de l'obtention du diplôme de **Master** en **:**

Filière : Electronique
Spécialité : Electronique des Systèmes Embarqués

Thème :

Amélioration des fonctionnalités de l'horloge géante de la FSSA

Date de soutenance: 05/12/2019

Devant le jury composé de :

Président	Mr. KHERCHI Mohamed	MCB	UAMOB
Examinateur 1	Mr. ALI MOHAD Abdenour	MCB	UAMOB
Examinateur 2	Mr. FEKIK Arezki	MAA	UAMOB
Encadreur	Mr. NOURINE Mourad	MCA	UAMOB

Année Universitaire 2018/2019

Remerciements

J'offre ma grande gratitude à Dieu qui m'a aidé à faire ce travail.

J'exprime ma profonde gratitude à mes parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.

Je remercie mon promoteur ***Mr NOURINE Mourad*** *pour les efforts qu'il a déployé, pour m'aider, conseiller, encourager et corriger. Je voudrais remercier les membres de jury d'avoir accepté d'examiner mon travail.*

Je remercie aussi tout le corps enseignant dans le département de Génie Electrique qui a contribué à ma formation universitaire.

Sans oublier tous mes amis, Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici ma sincère reconnaissance.

Dédicaces

A mon père et ma mère

A mes Sœurs et mes Frères

A toute ma famille

A ma promo de ESE 2018/2019

A tous mes ami(e)s, proches ou lointains, je vous dédie ce travail.

M.yassine

Résumé

Mon projet est une horloge géante qui affiche l'heure et la date ainsi que la température et l'humidité de l'atmosphère. Cette horloge est de type numérique composée d'un cadre en bois et d'un couvercle en verre et un ensemble des LEDs (ws2812b) et d'éléments pour l'affichage. Pour contrôler ces LEDs, une carte Arduino Uno a été utilisée en plus d'un groupe d'éléments associés à la carte qui envoie les informations nécessaires à l'affichage (DHT11, RTC1307)

Cette horloge est contrôlée par une application sur le smartphone qui envoie les informations du contrôleur au Bluetooth connecté à la carte Arduino. Ce travail se compose de trois parties :

- La première partie parle généralement des différents systèmes d'horloge
- La deuxième partie parle des éléments utilisés dans ce projet
- La troisième partie explique comment programmer ce projet

Abstract

My project is a giant clock that displays the time and date as well as the temperature and humidity of the atmosphere. This clock is digital type composed of a wooden frame and a glass cover and a set of LEDs (ws2812b) and elements for the display. To control these LEDs, an Arduino Uno board was used in addition to a group of elements associated with the board which sends the information necessary for the display (DHT11, RTC1307)

This clock is controlled by an application on the smartphone which sends the information from the controller to the Bluetooth connected to the Arduino board. This work consists of three parts:

- The first part generally talks about the different clock systems
- The second part talks about the elements used in this project
- The third part explains how to program this project

الملخص

مشروعي عبارة عن ساعة عملاقة تقوم بعرض الوقت والتاريخ بالإضافة الي عرض درجة الحرارة والرطوبة للجو،
هده الساعة تنتمي الى النوع الرقمي تتكون من هيكل من الخشب وغلاف من الزجاج ومجموعة من اللدات والعناصر التي تقوم

للتحكم في هده اللدات تم استخدام لوحة أرد وينو اونو بالإضافة الي مجموعة من العناصر المرتبطة مع اللوحة التي تقوم بالعرض بإرسال المعلومات اللازمة من اجل العرض.

يتم التحكم في هده الساعة عن طريق تطبيق على الهاتف الذكي الدي يقوم بإرسال معلومات المتحكم الي البلوتوث المرتبط بلوحة أرد وينو، يتكون هدا العمل من ثلاث أجزاء

الجزء الأول يتحدث بصورة عامة عن مختلف أنظمة الساعات

الجزء الثاني يتحدث عن العناصر المستعملة في هدا المشروع

الجزء الثالث يتحدث عن كيفية برمجة هدا المشروع

Sommaire

Sommaire

Chapitre II : Etude théorique du circuit électronique

Sommaire

Sommaire

Sommaire

Liste des abréviations

FPGA : Fild Programmable Gate Array (matrice à porte logique programable).

PIC : Programmable Interface Controler

LCD : Liquid Crystal Display

LED : Lighting Electricaly Diode.

RTC : Real Time Clock.

MHz : Mega hertz

USB : Universal Serial Bus

PWM : Pulse Width Modulation.

E/S : Entrée/Sortie

I/O : Int put/Out put

UART / USART : Universal Asynchronous / Synchronous Receiving Transmitting

EEPROM : Electrically Erasible Programmable Read-Only Memory

SRAM : Static Random Acces Memory.

IDE : Integrated Development Environment.

BCD : Binaire codé en décimal

SDA : Serial Data

SCL : Serial Clock

GND : Ground (mass).

Vbat : tension batterie

Vcc : Voltage Current Continu

I2C : Inter Integrated Circuit Bus

R/W : lecture/ écriture

DIN : Data in (entrée)

DO : Data out (sortie)

AFH : Adaptive Frequency Hopping Feature

EDR : Enhanced Data Rate

PWM : Pulse Width Modulation

IDE : Environnement de Développement Intégré

Liste des figures

Liste des tableaux

Introduction Générale

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

Le projet dont fait l'objet ce PFE est la continuité d'une athématique déjà initiée en Master 2 de l'année précédent (2017/2018). En effet, il consiste à améliorer les options d'une horloge géante, réalisée pour la FSSA. Ces options cent l'affichage de la date (jour et mois), puis de l'affichage de la température ambiant, et de l'humidité environment, en tenait en compte eu particulier des moyens matériels disponibles du niveau de FSSA.

Ce rapport de mémoire est organisé en trois chapitres :

- Le premier est consacré aux généralités sur les systèmes des horloges.
- Le deuxième est dédié à l'étude théorique des composants du circuit à réaliser.
- Le troisième est consacré à la fois à simulation et à la réalisation pratique de l'horloge géante.

En fin, on termine par conclusion générale eu mettant en exergue les résultats obtenus.

Chapitre 1 : Généralités sur les systèmes des horloges

I.1 Introduction

Pour qu'une personne puisse organiser ses activités quotidiennes, elle a besoin de connaitre l'heure, pour cela, il utilise les montres. Que ce soit des horloges numériques ou des horloges analogiques, ils sont fabriqués dans le même but (connaitre l'heure), mais la différence réside dans les différentes technologies utilisées avec les avantages et les inconvénients qu'elles offrent.

Ce chapitre de l'étude fournira un bref historique de l'évolution des horloges, ainsi que les différents types et techniques qu'elles contiennent, ainsi que leurs avantages et inconvénients.

I.2 Bref historique des horloges en général

L'horloge est utilisée depuis l'aube de la civilisation, de sort que les horloges solaires étaient considérées comme la première horloge à être utilisée, mais leur utilisation tout au long de l'année était impossible, ce qui a conduit à l'utilisation de nouveaux concepts pour mesurer le temps, comme la durée d'une certaine quantité d'eau ou de sable pour s'écouler ou combien de temps il faut à la bougie pour bruler et d'autres manières…, mais cette méthode n'a pas donné la notion exacte de temps [1].

Les concepts précédents ont été utilisés pour calculer le temps jusqu'à l'apparition de l'horloge mécanique au moyen d'un moteur capable de déplacer le mécanisme. Le fonctionnement constant et continu de l'horloge est assuré par un régulateur qui représente la vitesse à laquelle le temps passe [1]. Après l'horloge mécanique, l'horloge atomique fait son apparition en 1994, Cette horloge a une plus grande précision que les horloges apparues auparavant, elle est la base de la seconde. L'horloge atomique n'est plus définie à partir de mouvements astronomiques, mais à partir des fréquences de transition atomiques : par définition, la fréquence de transition entre les niveaux hyperfins de l'état fondamental de Césium 133 est 9 192 631 770 Hz [2].

Puis, dans la renaissance scientifique, les horloges numériques ont été programmées par des composants électroniques, qui ont plus de possibilité de les fabriquer en différentes tailles et caractéristiques.

I.3 Le système horaire

D'une manière générale, les systèmes horaires sont des modules électroniques d'indication du temps. Il existe plusieurs catégories de modules horaires : le système numérique horaire dont l'affichage est de type digital ou numérique, et le système analogique à l'instar des montres à aiguilles. Ainsi, nous pouvons définir un système numérique horaire comme un système électronique constitué principalement d'une horloge digitale pouvant être associée à un autre module électronique, pour réaliser une fonction bien précise.

I.4 Les différents systèmes horaire

Dans le cas d'une montre, nous distinguons deux types d'affichage des systèmes horaires : le système horaire numérique et le système horaire analogique.

I.4.1 Système horaire numérique

C'est une technologie qui utilise l'affichage numérique ou alphanumérique, contrôlé par des cartes électroniques programmables (cartes Arduino, cartes à FPGA, PIC, etc.…). Cette technique est la dernière dans la grande industrie horlogère, elle offre de nombreux avantages qui ne sont pas offerts par les horloges analogiques, L'affichage d'une horloge numérique peut prendre plusieurs formes comme :

I.4.1.1 Afficheur sept segments

Les afficheurs 7 segments sont des afficheurs, ou chaque segment est composée de plusieurs LED, ils sont particulièrement présents dans les calculatrices et les montres à affichage numérique…. Les chiffres s'écrivent en allumant ou en éteignant des segments,

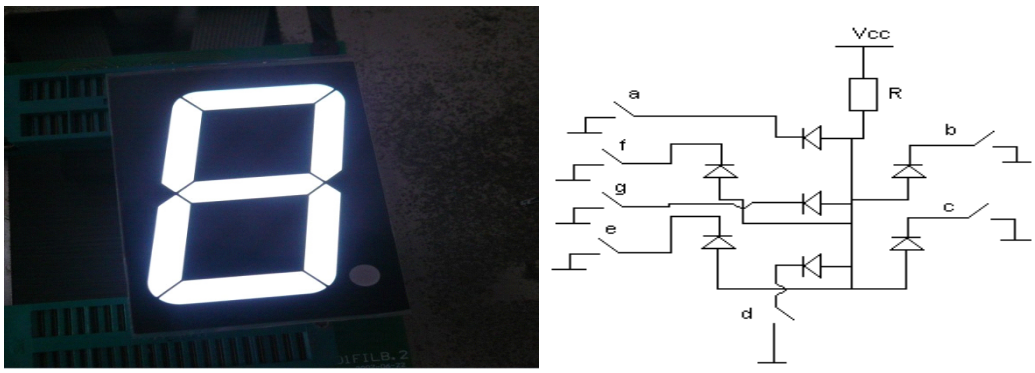


Figure I.1 Afficheur 7 segments a anode commune [3]

Dans un afficheur 7 segments, il y a deux types d'afficheur :

- **Afficheur à cathode commune :** dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas [4].
- **Afficheur à anode commune :** dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal. En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera. si la broche de signal est à l'état haut [4].

Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal.

I.4.1.2 Afficheur LCD

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement, il existé plusieurs afficheurs disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, mais aussi par leurs caractéristiques techniques et leur tension de service.

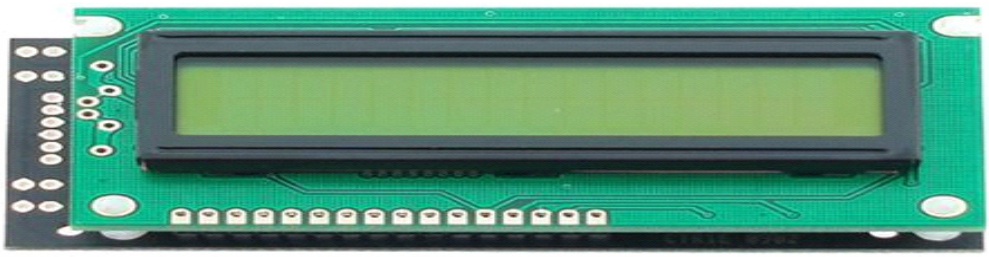


Figure I.2 : afficheur LCD [5]

L'afficheur est constitué de deux lames de verre, distantes de 20 µm en général, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réflectifs. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réflectif ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Le cristal liquide devient transparent lorsqu'il est excité pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière [7].

I.4.1.3 Afficheur Matrice à LED

Un afficheur matrice de LED est donc un ensemble de LED dont il est possible de choisir l'état de chacune d'elles indépendamment des autres, Les LED sont généralement disposées en lignes et en colonnes : on obtient un afficheur orthogonal. Cette matrice de LED est programmée pour l'affichage des lettres ou des chiffres.

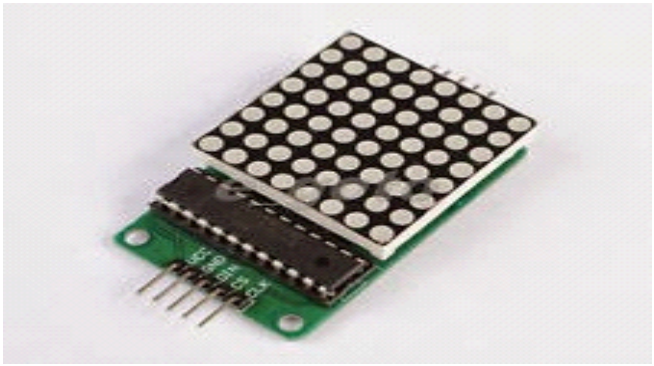


Figure I.3 : afficheur Matrice de LED 8x8 [6]

Cette matrice possède plusieurs performances (conception, dimensions et technologies) ce qui rend l'information plus claire, lisible, valorisante et facilement modulable. Généralement ils servent à effectuer plusieurs fonctionnalités telles que l'accueil, orientation, information, sensibilisation et la communication [9].

I.4.2 Système horaire analogique

Dans le monde de l'horlogerie, le terme "analogique" fait référence aux montres qui possèdent un système d'indication de temps analogique, c'est-à-dire que l'utilisateur connaît l'heure grâce au déplacement d'aiguilles qui indiquent, en règle générale, les heures, les minutes ainsi que les secondes, sur un cadran qui peut être divisé ou non. Ce type d'affichage peut s'accompagner d'un affichage alphanumérique ou numérique qui, quant à lui, donne des

indications de temps par le biais d'un guichet à chiffres ou à lettres [8].

Figure I.4 : Photos d'une horloge analogique [11]

I.4.3 Les avantages et inconvénients d'une montre numérique

Les avantages	Les inconvénients
• Plus récentes et plus résistantes. • Ce type d'affichage donne l'heure en une fraction de seconde. • La taille de l'écran n'affecte pas le cerveau. • Le cerveau n'effectue aucun effort supplémentaire d'interprétation pour associer l'aiguille à une heure, il lit simplement les données. • Visible dans la nuit.	• Le reproche le plus fréquent tient à leur look, traditionnellement moins élégant. • Dépendent de leur alimentation : pile, source.

Tableau I.1 Les avantages et inconvénients d'une montre numérique [10]

I.4.4 Les avantages et inconvénients d'une montre analogique

Les avantages	Les inconvénients
• L'apparence est élégante • Les infinies possibilités offertes par le cadran aux joailliers : incrustations, gravures, peintures, etc. • Les montres analogiques peuvent également utiliser un mouvement à quartz ou mécanique	• Une lisibilité moindre, parfois même • Très faible pour certains modèles conceptuels, l'impossibilité de les consulter dans la noire

Tableau I.2 Les avantages et inconvénients d'une montre analogique [10]

I.5 Le choix de la technologie pour notre PFE

Après avoir donné les divers systèmes d'affichage pour les horloges, en soulignant leurs avantages et inconvénients, nous avons choisi la technologie numérique pour réaliser notre projet, car de part son aspect moderne, celle-ci permet d'introduire de nouvelles fonctionnalités et d'options lu termes d'affichage et d'ergonomie.

I.6 Cahier de charges

Ce projet s'inscrit dans le cadre des PFE en master en électronique des systèmes embarqués. Il s'agit de l'amélioration des fonctionnalités de l'horloge géante de la FSSA, déjà initié dans le cadre d'un PFE l'année 2017/2018, ce PFE constitue donc une continuité de ce qui est déjà réalisé, en ajoutant bien sur de nouvelles fonctionnalités, en plus de l'affichage de l'heure eu temps réel, ces nouvelles fonctionnalités sont :

- L'affichage de la date complet (jour, mois, année).
- L'affichage de la température externe.
- L'affichage de l'humidité.
- Possibilité de contrôler la luminosité de l'horloge manuellement.

La commande et le control de cette horloge géante se fera à l'aide d'une application implémentée sur téléphone, via une liaison de communication à distance telle Bluetooth. Ce panneau d'affichage possède essentiellement une carte de contrôle à base de la carte Arduino Uno, avec un ensemble de composants électroniques.

I.7 Schéma synoptique

Afin de simplifier la conception du circuit électrique désiré, nous pro posons le schéma synoptique illustré sur la figure I.3. Les différents modules entrant dans la réalisation de ce système d'affichage sont :

- L'alimentation qui fournit l'énergie nécessaire au système pour son fonctionnement.
- La carte Arduino qui commande l'affichage du l'horloge.
- Le périphérique d'entré (modules Bluetooth) qui permet de recevoir les consignes de l'utilisateur.
- Le périphérique d'entré (RTC 1307) qui permet d'envoyer le temps et la date.
- Le périphérique d'entré (photorésistance) qui contrôle la luminosité.
- Le périphérique de sortie (ws2812b) qui permet l'affichage.

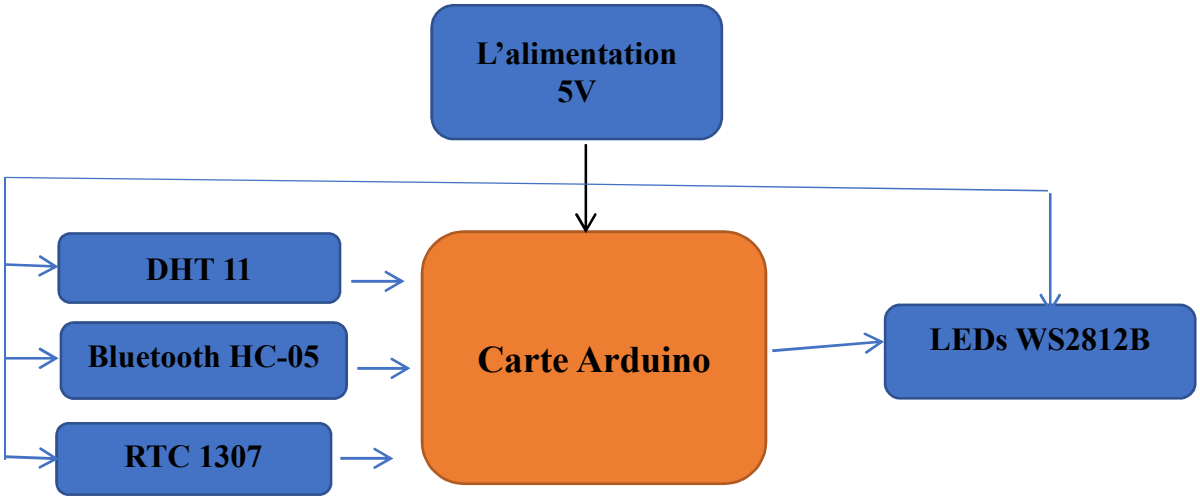


Figure I.5 Schéma synoptique

I.8 Langage de programmation

I.8.1 Programmation Arduino

Un programme permet de traduire le cahier des charges en une suite ordonnée d'actions que doit réaliser le processus de commande, il se base sur un algorithme qui est une procédure composée d'une séquence d'opérations qui sera traduite en instructions élémentaires. Par la suite, il suffit de transformer ces actions en un langage évolué tel que le langage java ou le langage C [9].

Un langage de programmation permet d'écrire un ensemble d'instructions (code source), qui seront directement converties en langage machine grâce à un compilateur [13].

Le logiciel Arduino, basé sur le langage C++, possède une bibliothèque de développement riche. L'exécution s'effectue d'une manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres [15].

I.8.2 Structure d'un programme Arduino

Le programme Arduino est composé de 3 parties principales :

- La partie optionnelle : déclaration des constantes et des variables.
- La partie initialisation et configuration des entrées/sorties dans la fonction Setup ().
- La partie principale qui s'exécute en boucle (la fonction Loop), elle est réservée aux instructions à effectuer par le programme [12].

I.9 Conclusion

Nous avons donnée dans ce chapitre un aperçu des différents systèmes d'affichage des montres. Il en ressort que les systèmes numériques de gestion horaire sont des systèmes électroniques constitué principalement d'une horloge digitale pouvant être associé à un autre module électronique pour réaliser une fonction bien précise. Ces systèmes présentent de nombreux avantages parmi lesquels : sa clarté et son type d'affichage qui donne l'heure en une fraction de seconde, quelle que soit la taille de l'écran. Le panneau d'affichage est construit autour de la carte Arduino et les pics. Le chapitre suivant va porter sur l'étude des composants de ce panneau d'affichage à réaliser.

Chapitre 2 : Etude théorique du circuit électronique

II. 1 Introduction

Au fils des années, l'Arduino est devenu un élément clé de milliers de projets, ces derniers varient du plus simple au plus complexe. De ce fait, une large communauté profite de cette plateforme à source libre. Il représente un pont tendu entre le monde réel et le monde numérique, et permet une manipulation facile de divers composants électroniques [14].

Dans ce chapitre, nous allons présenter la description théorique et générale sur la carte Arduino et divers composants utilisés dans ce projet.

II.2. Présentation de l'Arduino

II.2.1 Bref historique de l'Arduino

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie). Ils rencontraient un problème majeur à cette période (2003-2004) ; les outils nécessaires à la création de projet d'interactivité étaient complexes et onéreux (entre 80 et 100 euros).

Les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Leur préoccupation se concentre alors sur la réalisation d'un matériel moins cher et plus facile à utiliser [16].

En 2003, Heranado Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée « Wiring », accompagnée d'un environnement de programmation libre et ouvert. Cette carte a donc inspiré le projet Arduino (2005) et conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Caurtielles, Massimo Banzi et Nicholas Zambetti) [17].

II.2.2 Définition

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif, c'est un circuit imprimé en matériel libre (dont les plants sont publiés en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière d'effectuer des taches diverses. Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe [14].

II.2.3 Constitution d'une carte Arduino UNO

La carte Arduino Uno est basée sur un ATMega328 cadencé à 16 MHz. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires [15].

Pour pouvoir le programmer, il suffit de la connecter à un ordinateur à l'aide d'un câble USB, puis vérifiez le port, après cela, vous pouvez envoyer le code à l'Arduino.

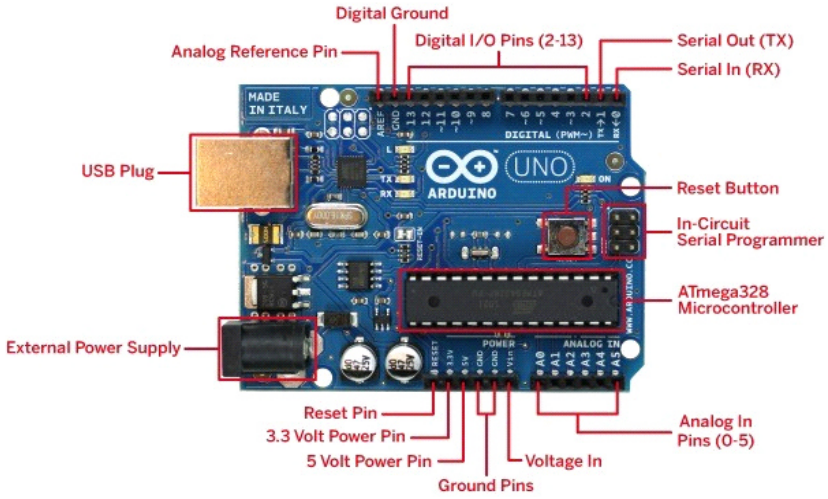


Figure II.1 Constitution de la carte Arduino UNO [18].

II.2.4 Caractéristiques de la carte Arduino UNO

Dans ce tableau II.1, on présentera les caractéristiques de la carte Arduino UNO

Microcontrôleur	ATmega328
Tension d'alimentation interne	5V
Tension d'alimentation (recommandée)	6-20V
Entrées/sorties numériques	14 dont 6 sorties PWM
Entrées analogiques	6
Courant max par broches E/S	40 mA
Courant max sur sortie 3,3V	50mA
Mémoire Flash	32 KB
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB

Fréquence horloge	16 MHz
Dimensions	68.6mm x 53.3mm

Tableau II.1 Caractéristiques de la carte Arduino UNO [19]

II.2.5 Brochage de la carte Uno

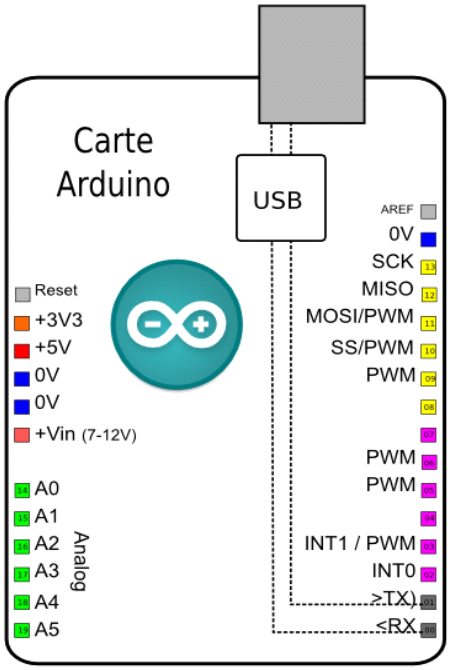


Figure II.2 Brochage de la carte Arduino Uno [19]

II.2.6 Microcontrôleur ATMEL ATMega328

Le microcontrôleur de la carte Arduino UNO est un ATMega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

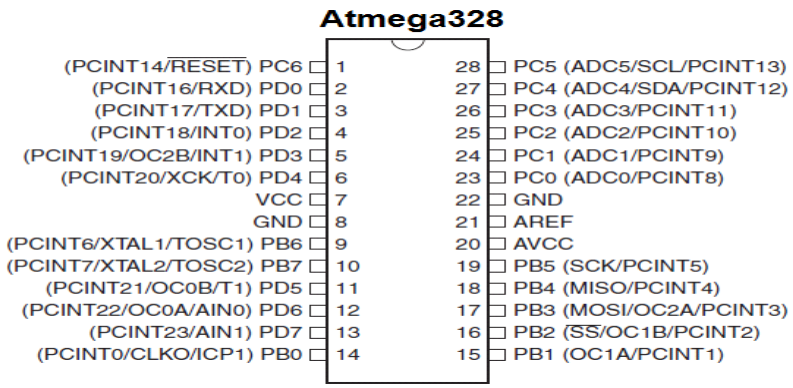


Figure II.3 Microcontrôleur ATMega328 [20]

II.2.6.1 Les principales caractéristiques d'atmega328

Mémoire FLASH	Mémoire programme de 32Ko
Mémoire SRAM	Données 2Ko
Digital I/O (entrées-sorties Tout Ou Rien)	3 ports Port B, Port C, Port D
Mémoire EEPROM	Données 1Ko
Timers	Timer0 et Timer1Timer2
PWM	6 broches OC0A(PD6), OC0B(PD5), 0C1A(PB1), OC1B(PB2), OC2A(PB3), OC2B(PD3)
Analog to Digital Converter	6 entrées multiplexées ADC0(PC0) à ADC5(PC5)
Gestion bus I2C	SDA(PC5) /SCL(PC4).
Port série (USART)	TXD(PD1) /RXD(PD0)
Comparateur Analogique	Broches AIN0(PD6) et AIN1 (PD7)

Tableau II.2 les caractéristiques d'atmega328 [20]

II.2.7 Le programmation avec l'IDE Arduino

II.2.7.1 Caractéristiques du développement Arduino [21]

- Arduino fournit un environnement de développement (IDE) avec un éditeur de source, les opérations de compilation et de chargement dans la mémoire du microcontrôleur étant ramenées à des clicks sur des boutons dans l'IDE (très simple). La communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par Arduino).
- Structure d'un projet Arduino L'outil impose de structurer l'application de façon spécifique. Le compilateur utilisé est AVR GCC (compilateur C/C++ pour processeur AVR).

Le programme principal (fonction main) est imposé, non modifiable, et décrit ci-dessous.

Les seules parties que l'on développe spécifiquement sont :

- **La fonction Setup () :** doit contenir les initialisations (times, interruptions...)

- **La fonction Loop () :** fonction répétée indéfiniment.

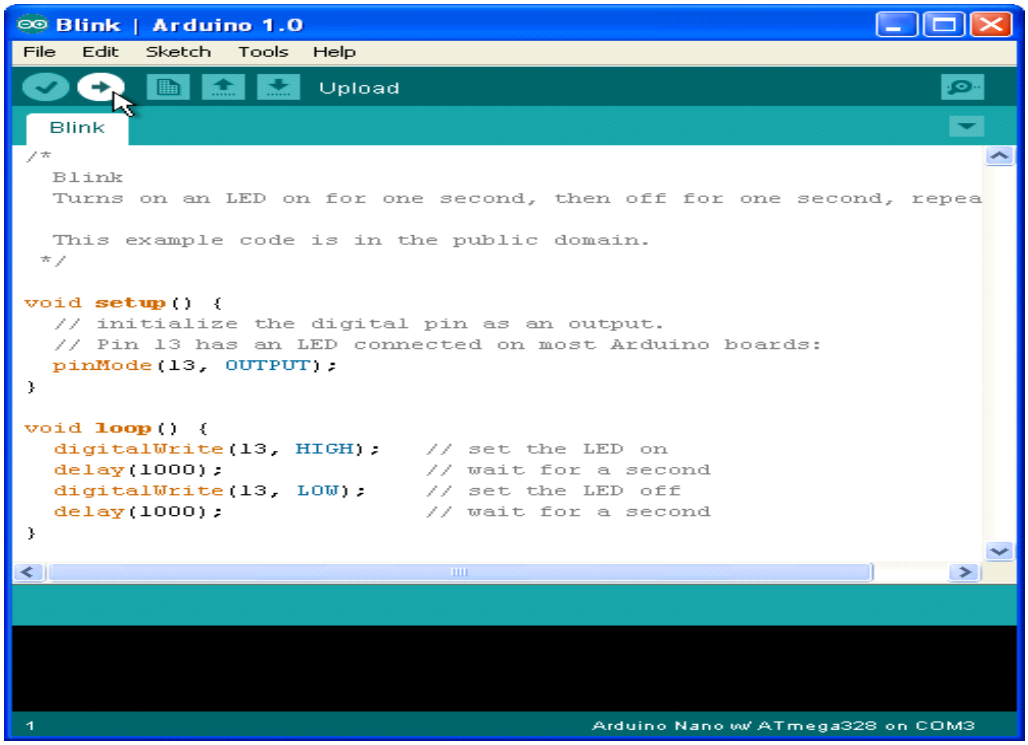


Figure II.4 Interface IDE Arduino [21]

II.2.7.2. Langage C pour Arduino UNO

Nom	**Description**
Boolean	Donnée logique (true ou false) sur 8 bits
Byte	Octet
Int	Entier signé sur 16 bits
Unsigned Int	Entier non signé sur 16 bits
Long	Entier signé sur 32 bits
Unsigned long	Entier signé sur 32 bits
Float	Nombre à virgule flottant sur 32 bits
Char	Caractère sur 8 bits. Seuls les caractères ayant pour code ASCII une valeur 0 à 127 sont définis
Unsigned char	Caractère sur 8 bits. Caractères du code ASCII étendu (valeur 0 à 255)

Tableau II.3 Les différents types utilisés dans la programmation Arduino [22]

L'exemple le plus simple, fourni par ARDUINO, consiste à faire clignoter la LED (sur la carte UNO) connectée à la broche PB5 du microcontrôleur, broche n° 13 sur les connecteurs de la carte. La fonction setup () configure la broche PB5 (connexion n°13 sur la carte) en sortie, à l'aide de la fonction Arduino pin Mode (). La fonction loop () décrit ensuite ce qui sera répété indéfiniment : mettre PB5 à 1 pendant 200ms puis mettre PB5 à 0 pendant 1s, et ainsi de suite [21].

```
// Exemple de programmations
 void setup () {
pin Mode (13, OUTPUT); //broche PB5 en sortie
 }
void loop () {
digitalWrite (13, HIGH); // set PB5 Delay (200); // délai 200 ms
digitalWrite (13, LOW); // clear PB5 Delay (1000); // délai 1 s       }
```

II.2.7.2.1 Quelques fonctions courantes fournies par arduino [21]

Constantes : **HIGH, LOW, INPUT, OUTPUT**

I/O (gestion entrées/sorties binaires) :

- **Pin Mode (pin, mode) :** pin = numéro, mode = INPUT/OUTPUT/INPUT_PULLUP
- **digitalWrite (pin, value):** pin = numéro, value= HIGH/LOW
- **int digitalRead (pin) :** pin = numéro sur connecteur, retourne la valeur

Temporisations :

- **Delay (unsigned long ms) :** délai de ms milli secondes avec ms sur 32 bits
- **DelayMicroseconds (unsigned int ms) :** délai de ms microsecondes avec ms sur 16 bits

- **unsigned long micros () :** nombre de micro secondes depuis démarrage. Revient à zéro tous les 70mn environ.
- **unsigned long milli () :** nombre de milli secondes depuis démarrage. Revient à zéro tous les 50 jours environ.

Liaison Série : ces fonctions permettent de lire/écrire sur la liaison série du microcontrôleur et donc de communiquer avec le PC auquel il est relié (via cordon USB).

- **Serial.begin (9600);** // configure la liaison à 9600 bits/s
- **Serial.println (v, f);** // envoie v au format f (DEC, HEX, BIN)

II.3 Etude du RTC DS1307 [23]

Le DS1307 est une horloge temps réel (RTC) série de une faible puissance, on retrouve une information binaire ?codé décimal (BCD) de l'horloge et du calendrier ainsi que 56 octets de secours NV SRAM.

La fonction horloge / calendrier fournit les informations secondes, minutes, heures, jour, date, mois, et l'année. La fin de la date de mois est ajustée automatiquement pour les mois de moins de 31 jours, y compris des corrections pour l'année bissextile. L'horloge fonctionne soit dans le format 24 heures ou 12 heures avec indicateur AM / PM.

Une pile de sauvegarde peut être mise en place pour conserver les données en cas de coupure de l'alimentation extérieure. Comme indique la figure II.5.

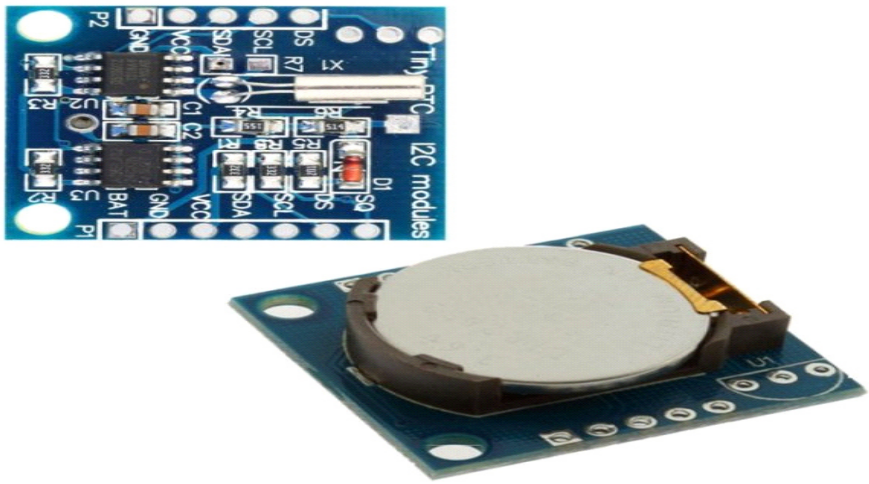


Figure II.5 Module du RTC DS1307 [24]

II.3.1. Caractéristiques du DS1307

Le DS1037 contient 8 comme la figure II.6.

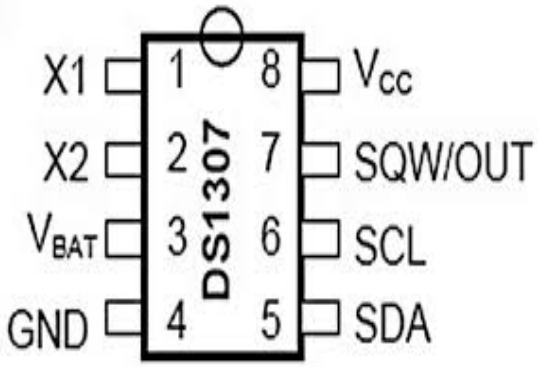


Figure II.6 DS1307 [25]

Les caractéristiques du DS1307 sont représentées dans le tableau II.4 suivants :

Pin	Rôle
X1	Cristal
X2	Cristal
Vbat	Tension batterie
GND	Masse
SDA	Ligne des données (data) I²C
SCL	Ligne d'horloge (clock) I²C
SQW/OUT	Sortie signal carré
Vcc	Tension logique 5V

Tableau II.4 Caractéristiques du RTC 1307 [23]

II.3.2 Câblage du DS1307

Dans le circuit simple, les deux entrées X1 et X2 sont connectées à un oscillateur à cristal de 32,768 kHz en tant que source pour la puce. VBAT est connecté à la culture positive d'une puce de batterie 3V. L'alimentation Vcc de l'interface I2C est de 5 V et peut être fournie

à l'aide de microcontrôleurs. Si l'alimentation Vcc n'est pas autorisée, les lectures et écritures sont inhibées [25]. Comme la figure II.7.

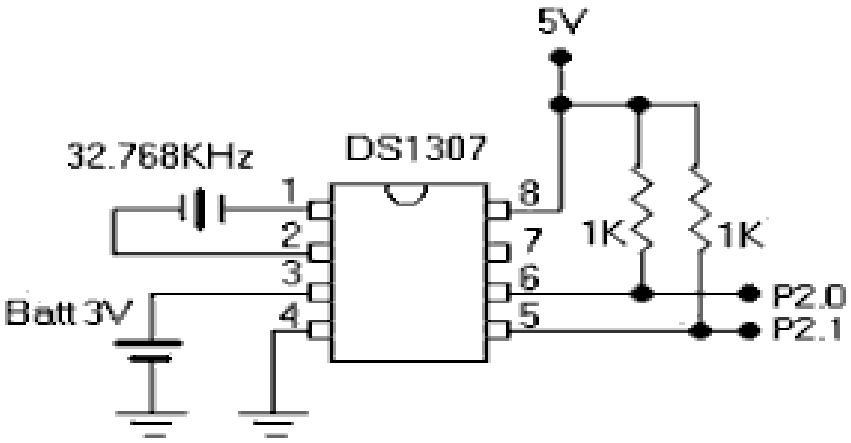


Figure II.7 Câblage du DS1307 [25]

II.4. Etude du bus I2C

II.4.1. Présentation de l'I2C [26]

L'I2C (Inter Integrated Circuit bus) a été créé par PHILIPS dans les années 80. Il a été adopté par un grand nombre de constructeurs.

L'I2C est un protocole de communication de niveau physique. Son fonctionnement s'appuie sur trois fils (SDA : Serial Data, SCL : Serial Clock et la référence : Masse).

Le fil nommé SCL transmet l'horloge pour la synchronisation de communication alors que la ligne SDA transmet les informations (adresses et données).

Un seul boîtier maître peut commander plusieurs boîtiers esclaves, (chacun d'entre eux répondant à une adresse unique).

Un esclave ne s'exprime que sur ordre d'un maître et plusieurs maîtres peuvent partager le même réseau.

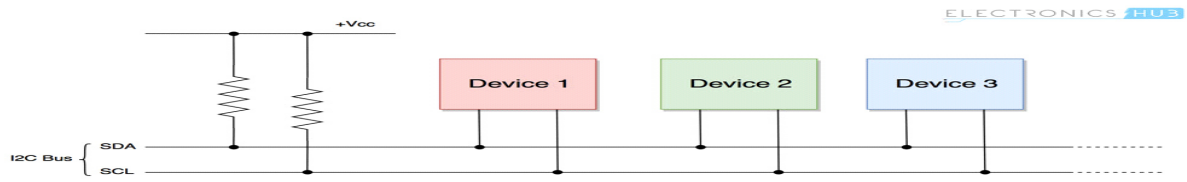


Figure II.8 Exemple d'un bus I2C [27]

II.4.2 Le protocole I2C

II.4.2.1 Configuration matérielle [26]

Les lignes SDA et SCL sont bidirectionnelles (entrée/sortie), de ce fait pour éviter tout conflit électrique elles sont de type « collecteur ou drain ouvert » et sont câblées à travers une résistance de tirage vers l'alimentation positive du circuit.

Le câblage réalise ainsi sur chaque ligne un "ET logique".

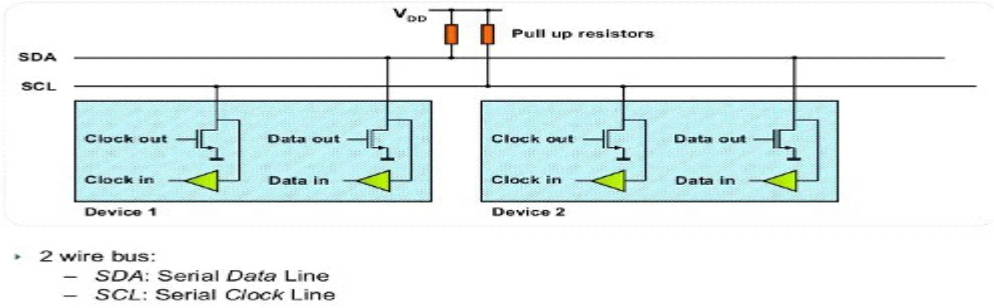


Figure II.9 Configuration matérielle de l'I2C [28]

II.4.2.2 La transmission des données

II.4.2.2.1 Validité des données [29]

Les paquets de données I2C sont organisés en octets de 8 bits comprenant l'adresse de l'esclave, le numéro de registre et les données à transférer. La transmission sur le bus est une opération de lecture ou d'écriture. Les protocoles de lecture et d'écriture reposent sur une série de sous-protocoles tels que les conditions de début et de fin, les bits de début répétés, l'octet d'adresse, les bits de transfert de données et les bits d'accusé / non acquitté

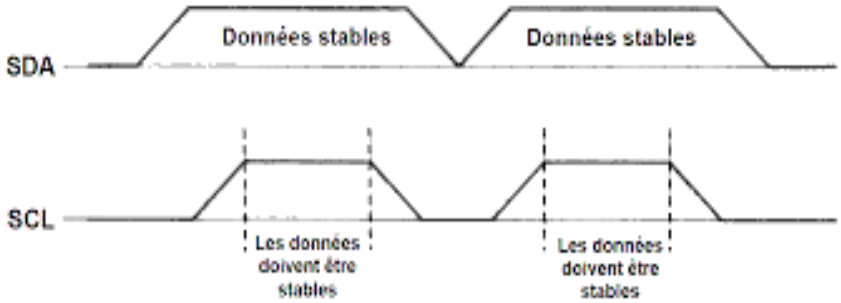


Figure II.10 validités des données [26]

II.4.2.2.2. Conditions de START et STOP [30]

Un message débute par une condition de START et se termine par une condition de STOP. Après une condition de START le bus est considéré comme occupé, il est de nouveau libre après la condition de STOP. Les conditions de START et de STOP sont émises par le maître.

La condition de START : Passage de l'état haut à l'état bas de SDA pendant que SCL est à l'état haut.

La condition de STOP : Passage de l'état bas à l'état haut de SDA pendant que SCL est à l'état haut.

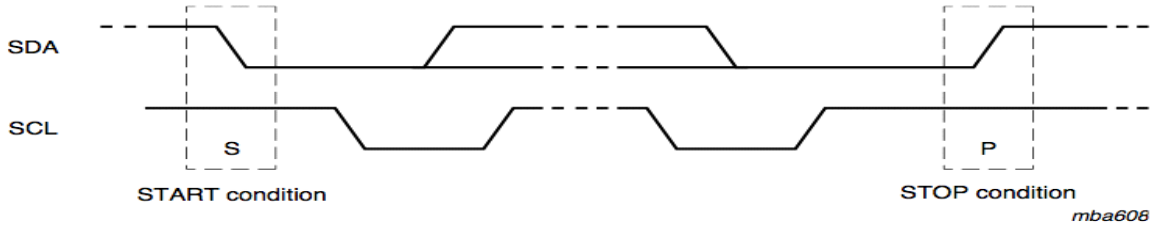


Figure II.11 Conditions de START et STOP [30]

II.4.2.2.3 Format de transmission

Il est possible de combiner les protocoles de lecture et d'écriture selon différentes variantes pour effectuer certaines transactions I2C complexes. Le maître peut écrire sur le même esclave, puis le lire ou donner une nouvelle adresse pour parler à un autre appareil slave au sein d'une même transaction I2C. Les données peuvent changer de sens, de sorte qu'un appareil en cours d'écriture est en train de lire les données. Tout cela est accompli en utilisant un bit de départ répété [29].

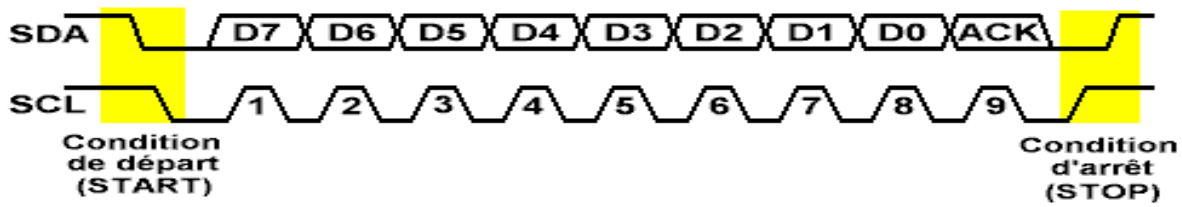


Figure II.12 Illustration de format de la transmission [26]

II.5 Etude du module BLUETOOTH HC-05

II.5.1. Le HC-05 [31]

Le module Bluetooth HC-05 est un module MASTER / SLAVE. Par défaut, le réglage par défaut est SLAVE. Le rôle du module (maître ou esclave) ne peut être configuré que par AT COMMANDS. Les modules esclaves ne peuvent pas établir de connexion à un autre périphérique Bluetooth., mais peut accepter les connexions. Le module maître peut établir une connexion avec d'autres périphériques.

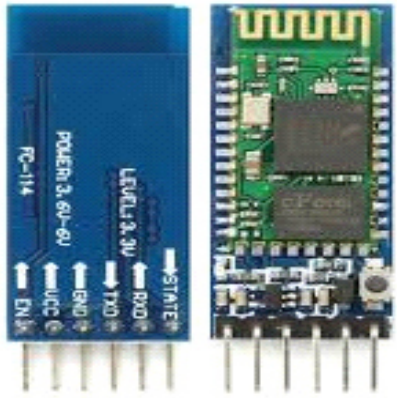


Figure II.13 Bluetooth HC-05 [31]

II.5.2 Caractéristiques matérielles [31]

- Sensibilité typique de -80dBm.
- Jusqu'à 4 dBm de puissance d'émission RF.
- 3.3 à 5 V E / S. - Contrôle (entrée / sortie programmable).
- Interface UART avec débit de transmission programmable.
- Avec antenne intégrée.
- Avec connecteur de bordure

II.5.3 Les caractéristiques logicielles [31]

- Esclave par défaut Vitesse de transmission : 9600, bits de données : 8, bit d'arrêt : 1, parité : sans parité.
- Connecté automatiquement au dernier périphérique sous tension par défaut.

- Autoriser l'appareil de couplage à se connecter par défaut.
- Emplacement automatique PINCODE : "1234" par défaut.

II.5.4. La description de la broche [31]

Comme nous savons que Vcc et GND du module vont à Vcc et à GND d'Arduino. La broche TXD va à la broche RXD d'Arduino et la broche RXD à la broche TXD d'Arduino (broches numériques 0 et 1).

II.6 Etude du capteur de température et d'humidité DHT11

II.6.1 Le capteur DHT11 [32]

Le capteur DHT11 est lui capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes.

Le DHT11 compatibles 3.3 volts et 5 volts (le fabricant recommande cependant de toujours alimenter le capteur en 5 volts pour avoir des mesures précises).

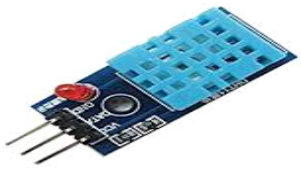


Figure II.14 Capteur DHT11 [33]

II.6.2 les caractéristiques du capteur DHT11

Humidité (relative %)	20 ~ 80%
Précision (humidité)	+/- 5%
Température	0 ~ +50°C
Précision (température)	+/- 2°C
Fréquence mesure max	1Hz (1 mesure par seconde)
Tension d'alimentation	3 ~ 5 volts
Stabilité à long terme	+/- 1% par an

Tableau II.5 caractéristiques du capteur DHT11 [32]

Note : Le DHT11 ne peut pas mesurer (et supporter) des températures négatives ou des températures supérieures à 50°C.

II.6.3. Câblage d'un capteur DHT11 [32]

Le brochage du capteur est le suivant :

- La broche n°1 (Vss) est la broche d'alimentation (5 volts ou 3.3 volts).
- La broche n°2 (Data) est la broche de communication
- La broche n°3 est la masse du capteur (GND).

II.6.4. Le protocole de communication

Les capteurs DHTxx ont la particularité de communiquer avec le microcontrôleur via une unique broche d'entrée / sortie.

Bien que cela soit marqué "One Wire" un peu partout sur le document constructeur du capteur, il ne s'agit pas d'un véritable bus de communication 1-Wire. Il s'agit simplement d'un protocole de communication propriétaire, utilisant un seul fil et nécessitant des timings très précis [32].

II.7 LED WS2812B

II.7. 1 Description générale [34]

WS2812B est une source lumineuse de contrôle à LED intelligente dans laquelle le circuit de contrôle et la puce RGB sont intégrés dans un ensemble de 5050 composants. Il intègre en interne un verrou intelligent de données de port numérique et un circuit de commande d'amplification de remodelage du signal. Inclut également un oscillateur interne de précision et un courant constant programmable avec une tension de 12V e-nt contrôle partie, assurant efficacement la hauteur de couleur de la lumière de point de pixel cohérente.

Le protocole de transfert de données utilise un mode de communication NZR unique. Après la réinitialisation à la mise sous tension du pixel, le port DIN reçoit les données du contrôleur, le premier pixel collecte les données initiales à 24 bits, puis envoyées au verrou de données interne, les autres données qui sont remodelées par le circuit d'amplification de

remodelage du signal interne envoyées à la cascade suivante.

Pixel à travers le port DO. Après transmission pour chaque pixel, le signal doit être réduit de 24 bits. Le pixel adopte la technologie de transmission de refonte automatique, rendant le nombre de pixels en cascade n'est pas limité à la transmission du signal, dépend uniquement de la vitesse de transmission du signal.

LED à basse tension de commande, protection de l'environnement et économie d'énergie, luminosité élevée, angle de diffusion est grand, bonne consistance, faible puissance, longue durée de vie et autres avantages. La puce de contrôle intégrée dans la LED ci-dessus devenir plus simple circuit, petit volume, pratique.

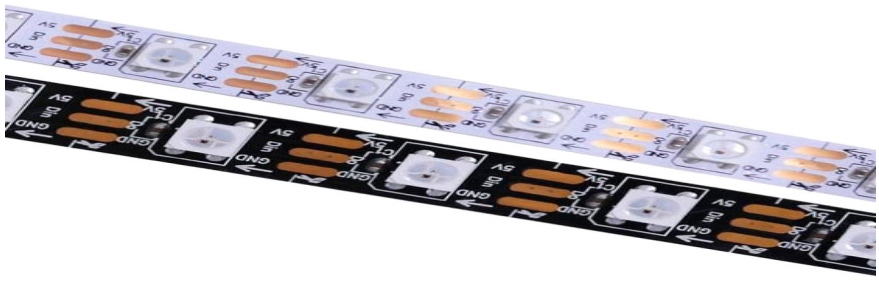


Figure II.15 LED WS2812B [35]

II.7. 2 Dimensions mécaniques

La Figure II.16 montre les dimensions mécaniques et les ports de LED ws2812b.

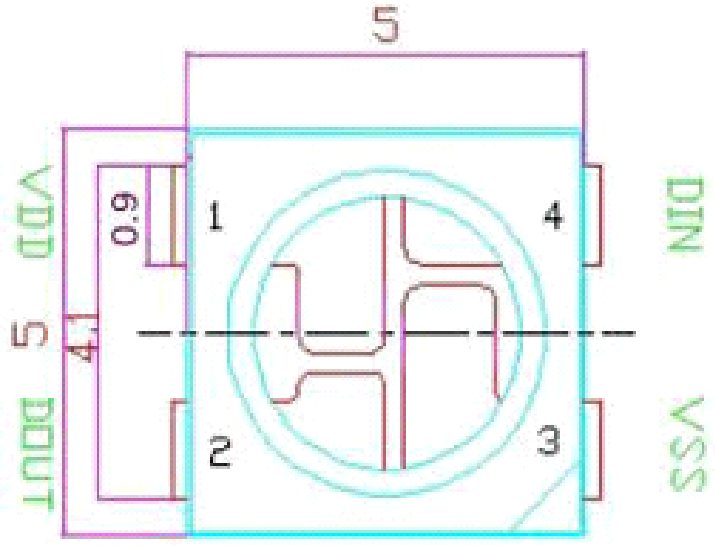


Figure II.16 : La taille mécanique et le dessin des broches ws2812b [34]

II.7. 3 Fonction PIN

No :	Symbole :	Description de la fonction :
1	VDD	LED d'alimentation
2	DOUT	Sortie du signal de données de contrôle
3	VSS	Ground
4	DIN	Entrée de signal de données de contrôle

Tableau II.6 : fonction PIN [34]

II.7. 4. Caractéristiques électriques

Paramètre :	Symbole :	Évaluations :	Unit :
Température	TA	-20 ～ +70	℃
Tension d'alimentation	VDD	4.5 ～ 5.5	V
Ground	VSS	0	V

Tableau II.7 Caractéristiques électriques du LED WS2812B [34]

II.7. 5. Paramètre du caractéristique RGB

Émission de la couleur :	Modèle :	Longueur d'onde (nm) :	Intensité lumineuse (mcd) :	Tension (V) :
Rouge	13CBAUP	620-630	550-700	1.8-2.2
Vert	13CGAUP	515-530	1100-1400	3.0-3.2
Bleu	10R1MUX	465-475	200-400	3.0-3.4

Tableau II.8 Paramètre du caractéristique RGB [34]

II.7.6. Câblage

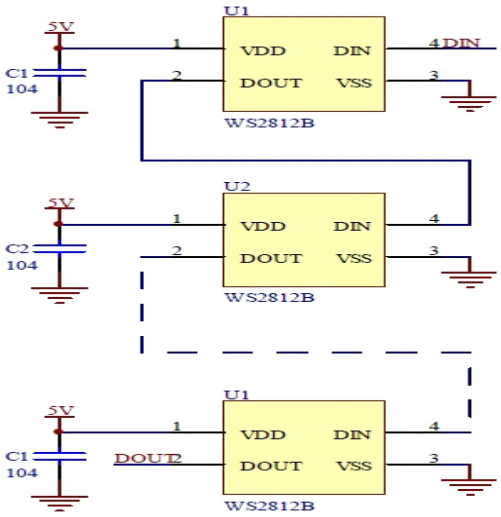


Figure II.17 Câblage d'une LED ws2812b [34]

II.8 Conclusion

En somme, nous avons fait un aperçu sur l'étude de la technologie des différents composants électroniques qui constituent notre module numérique du système horaire. Il en ressort que chacun joue un rôle spécifique pour le bon fonctionnement du module. L'étude des composants électroniques est très importante pour la compréhension du fonctionnement de tous appareils électroniques. Ces réflexions nous poussent à faire la simulation et la réalisation de notre maquette dans le chapitre suivant.

Chapitre 3 : Simulation et réalisation pratique de l'horloge

III.1 Introduction

Après avoir mené une étude générale sur le projet, nous avons conçu et programmé notre système numérique de gestion horaire selon les données fixées par le cahier des charges. Ce chapitre va porter d'une part sur la simulation et d'autre part sure la réalisation de cette maquette qui est marquée essentiellement par le devis et le cout estimatifs pour atteindre les objectifs du projet.

III.2 Partie simulation

III.2.1 Présentation du Proteus

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. De plus, nous pouvons également simuler ces schémas et ainsi permettre la détection de certaines erreurs de conception.

III.2.2 Présentation des outils de la simulation

Avant de passé à la réalisation, nous avons simulé Proteus pour la simulation de circuits et Arduino IDE pour la programmation de circuits.

III.2.3 Création d'un projet

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez new project (nouveau projet) de puis le menu project (projet), comme indiqué sur figure III.1.

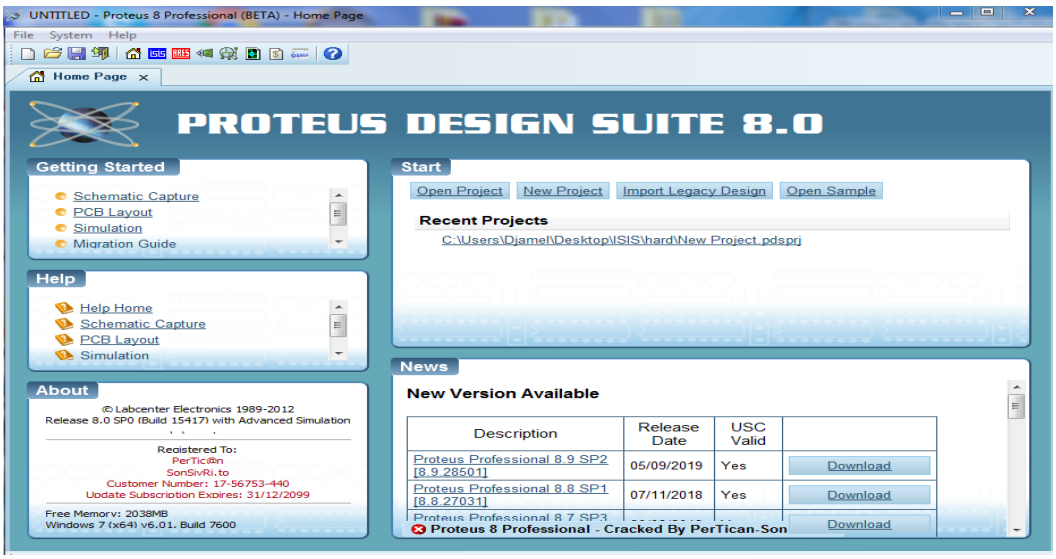


Figure III.1 Création d'un projet

Une nouvelle fenêtre apparaîtra comme indiqué sur figure III.2. Sélectionnez le nom et l'emplacement du projet, puis cliquez sur next pour ouvrir une nouvelle fenêtre vide pour écrire notre circuit.

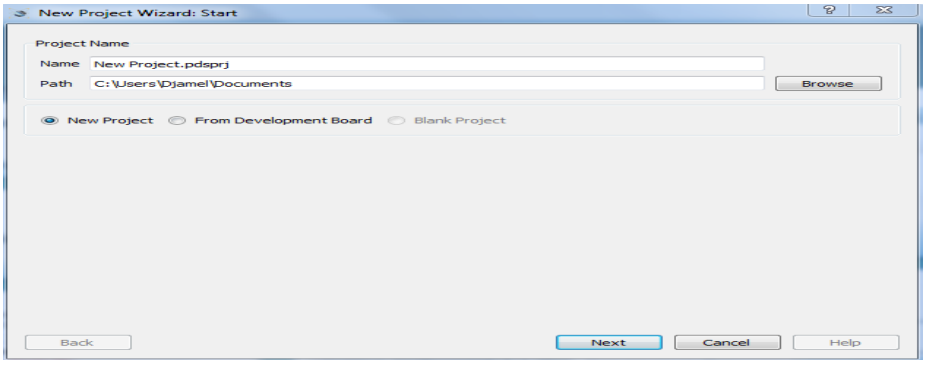


Figure III.2 Configurations de projet

III.2.4 Démarche de la simulation

La bibliothèque de RTC1307 et DHT11 ne sont pas des bibliothèques officielles de Proteus, donc il faut télécharger sur internet les bibliothèques et ajouter sur Library de Proteus.

III.2.5 Circuit de projet sur Proteus

La figure suivante représente le circuit de notre projet sous Proteus qui fonctionne selon le même concept que le circuit global du projet. Parce que certains composants ne peuvent pas être affiches dans la bibliothèque Proteus et seront utilisés dans la mise en œuvre.

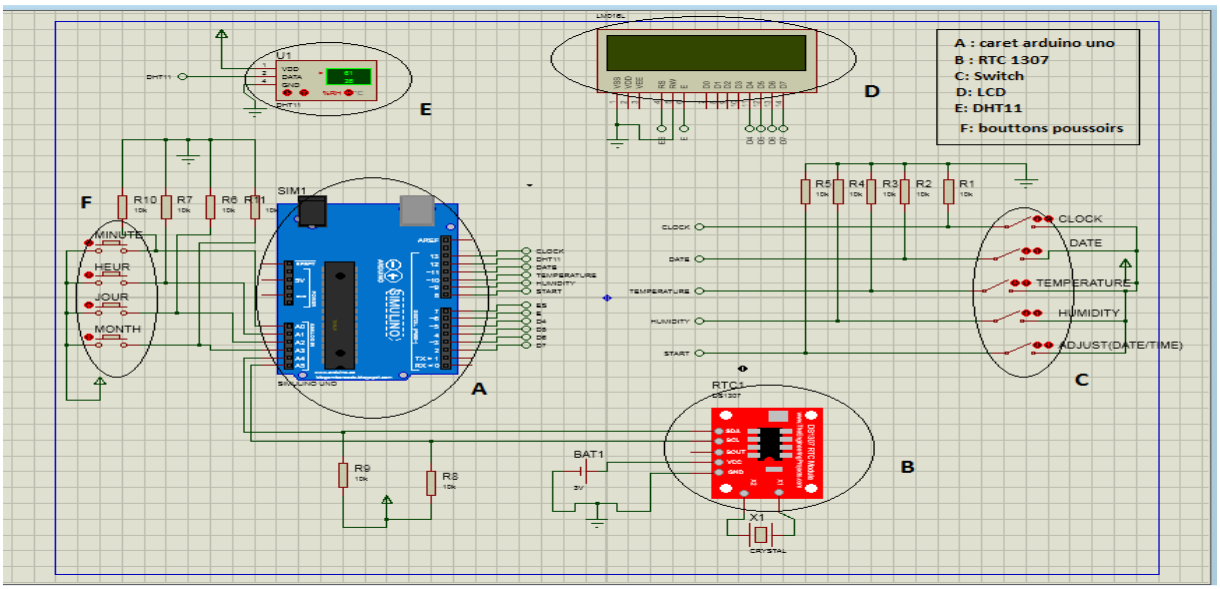


Figure III.3 Circuit de projet sur Proteus

Chaque module de ce circuit a une tâche spécifique qui est expliquée dans le tableau III.1, ainsi qu'une explication des différentes connexions.

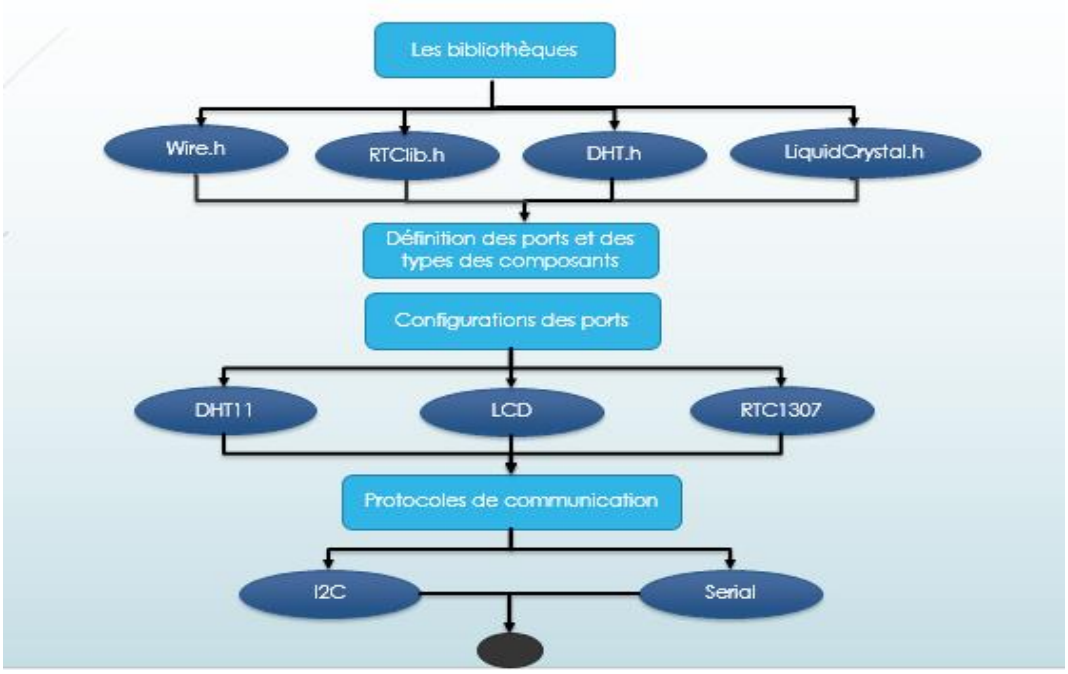
Module	**Rôle du module**	**Port du module**	**Port Arduino**	**Alimentation**
RTC 1307	Envoie l'heure et la date	SDA	A4	5V GND
		SCL	A5	
LCD	Pour l'affichage	D7	2	5V GND
		D6	3	
		D5	4	
		D4	5	
		E	6	
		ES	7	
DHT11	Envoie la température et l'humidité	Data	12	5V GND
Switch (clock)	L'affichage de temps	1	13	5V GND
		2	Vcc	
Switch (date)	L'affichage de la date	1	11	
		2	Vcc	
Switch (temp)	L'affichage de la température	1	10	
		2	Vcc	
Switch (hum)	L'affichage de l'humidité	1	9	
		2	Vcc	
Switch (adjust)	Changement	1	8	

	d'heure format	2	Vcc	
Bouton (minute)	Contrôler les minutes	1	A0	5V GND
		2	Vcc	
Bouton (heur)	Contrôler les heurs	1	A1	
		2	Vcc	
Bouton (jour)	Contrôler les jours	1	A2	
		2	Vcc	
Bouton (month)	Contrôler les mois	1	A3	
		2	Vcc	

Tableau III.1 Les différents rôles et connexions des modules

III.2.6 L'organigramme

Afin de simplifier le principe de fonctionnement, je vais créer un organigramme dressé en figure III.4.



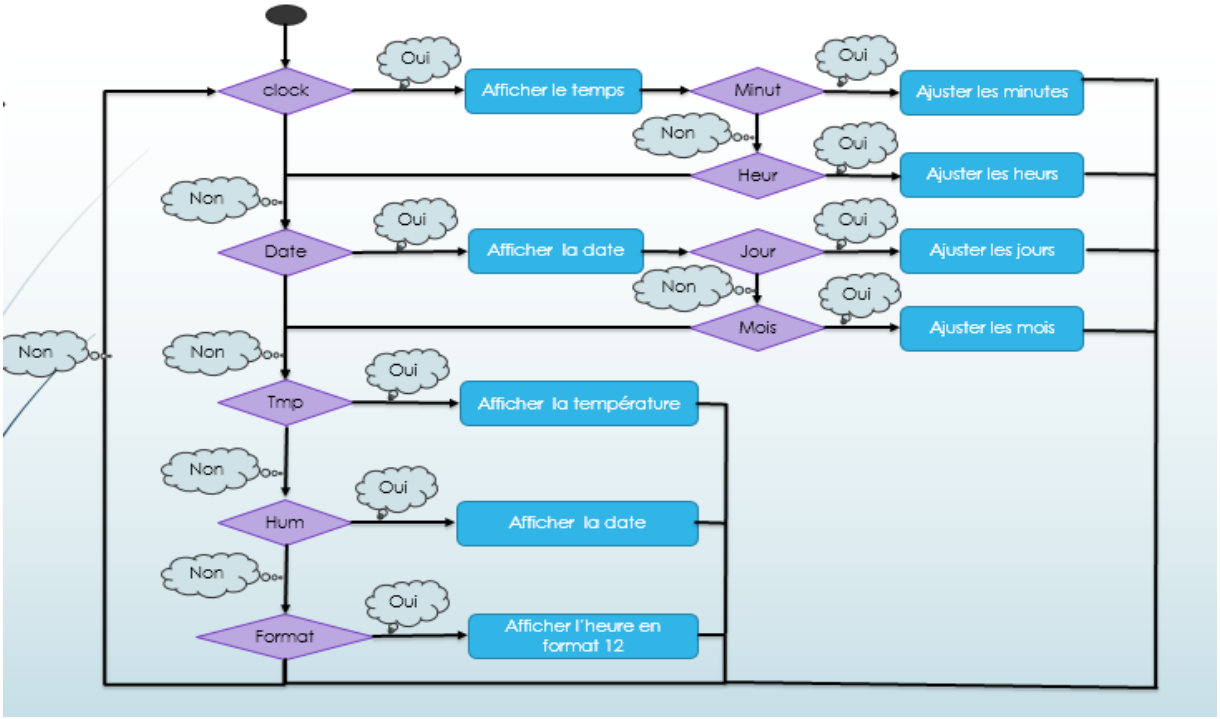


Figure III.4 L'organigramme du programme injecté à l'Arduino

III.2.7 Le programme de la simulation sur Proteus

Le programme de projet sur Proteus, affiché dans l'annexe de mémoire.

III.3 Partie pratique

Afin de réaliser le projet, vous devez écrire le programme qui sera chargé sur la carte Arduino pour obtenir les résultats nécessaires. Ce programme est imprimé sur un logiciel Arduino.

III.3.1 Création d'un programme de ce projet

III.3.1.1 Création d'un nouveau programme sur logiciel Arduino

Le processus de création d'un nouveau programme est très simple. Cliquer sur fichier comme la figure III.5, une petite fenêtre apparaitra, cliquez sur nouveau.

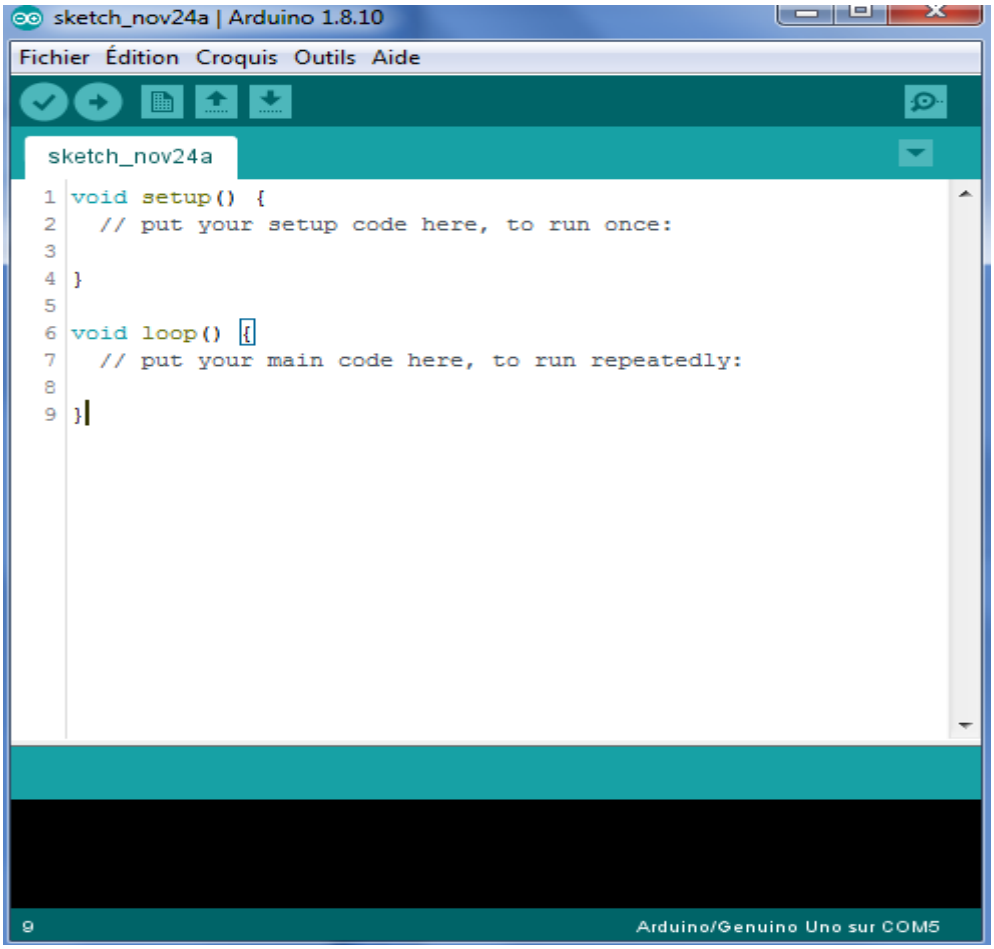


Figure III.5 Création d'un nouveau programme

III.3.1.2 Démarche de la programmation

Comme les bibliothèques de Proteus, les bibliothèques suivantes doivent être téléchargées et ajoutées dans la bibliothèque de logiciel Arduino :

- DHT.h
- FastLED.h
- RTClib.h
- SoftwareSerial.h
- Timer.h
- Wire.h

III.3.1.3 Transformation d'un programme

On part dans le menu <outils> puis dans <type de carte >. On vérifie que c'est bien le nom "Arduino Uno" qui est coché. Si ce n'est pas le cas, vous devez régler. Ensuite dans le menu outil, puis Serial port, on choisit le port COMX (X étant le numéro du port qui est affiché).

Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton <Téléverser>.

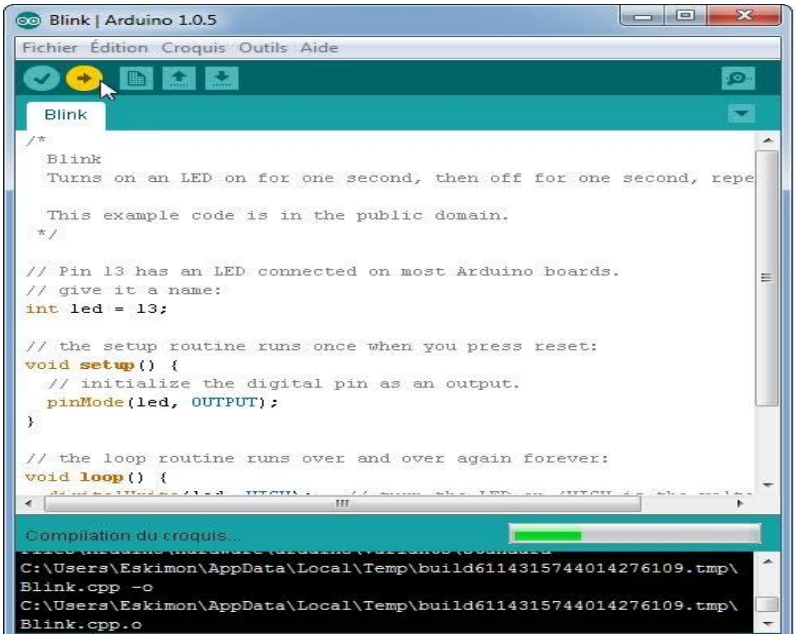


Figure III.6 Fenêtre correspondant à l'envoi du programme dans la carte Arduino

III.3.2 Les composants utilisés dans ce projet

Pour réaliser ce projet, vous devez obtenir cet ensemble des composants électronique :

- Carte Arduino uno.
- Module RTC DS1307.
- Capteur DHT11.
- Module Bluetooth HC-05.
- Alimentation 5V / 2A.
- LEDs WS2812b.
- Deux résistances (330 Ω – 1K Ω).
- Ensemble des fils.
- Plaque d'essai.

III.3.3 Schéma global du circuit

Ce schéma a été réalisé sur Fritzing, c'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

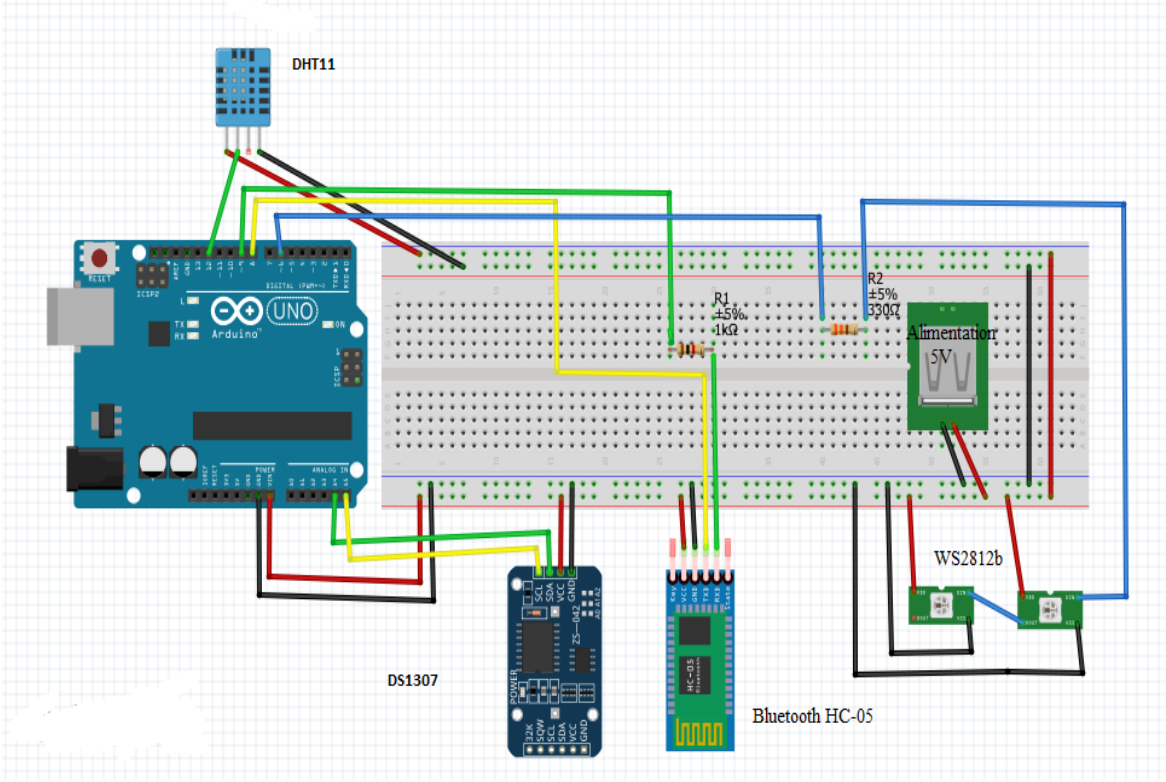


Figure III.7 Le schéma global du circuit

III.3.4 Brochage des composants

Module :	Port du module :	Port Arduino :	Alimentation :
DHT11	Data	12	5V GND
RTC DS1307	SDA	A4	5V GND
	SCL	A5	
Bluetooth HC-05	TXD	8	5V

	RXD	9	GND
LEDs Ws2812b	DIN	6	5V GND

Tableau III.2 Brochages des différents composants utilisant dans le circuit

III.3.5 Devis et cout estimatifs des composants

Composant	**Quantité**	**Prix unitaire (DA)**	**Prix total (DA)**
Arduino uno	1	2400	2400
Capteur DHT11	1	800	800
RTC DS1307	1	700	700
LEDs WS2812b	5 mètres	900	4500
Bluetooth HC-05	1	1200	1200
Résistance	2	20	40
Des fils jumpers	18	20	360
Plaque d'essai	1	500	500
Prix total (DA)	10500		

Tableau III.3 Devis et cout estimatifs

III.3.6 L'organigramme de code global du circuit

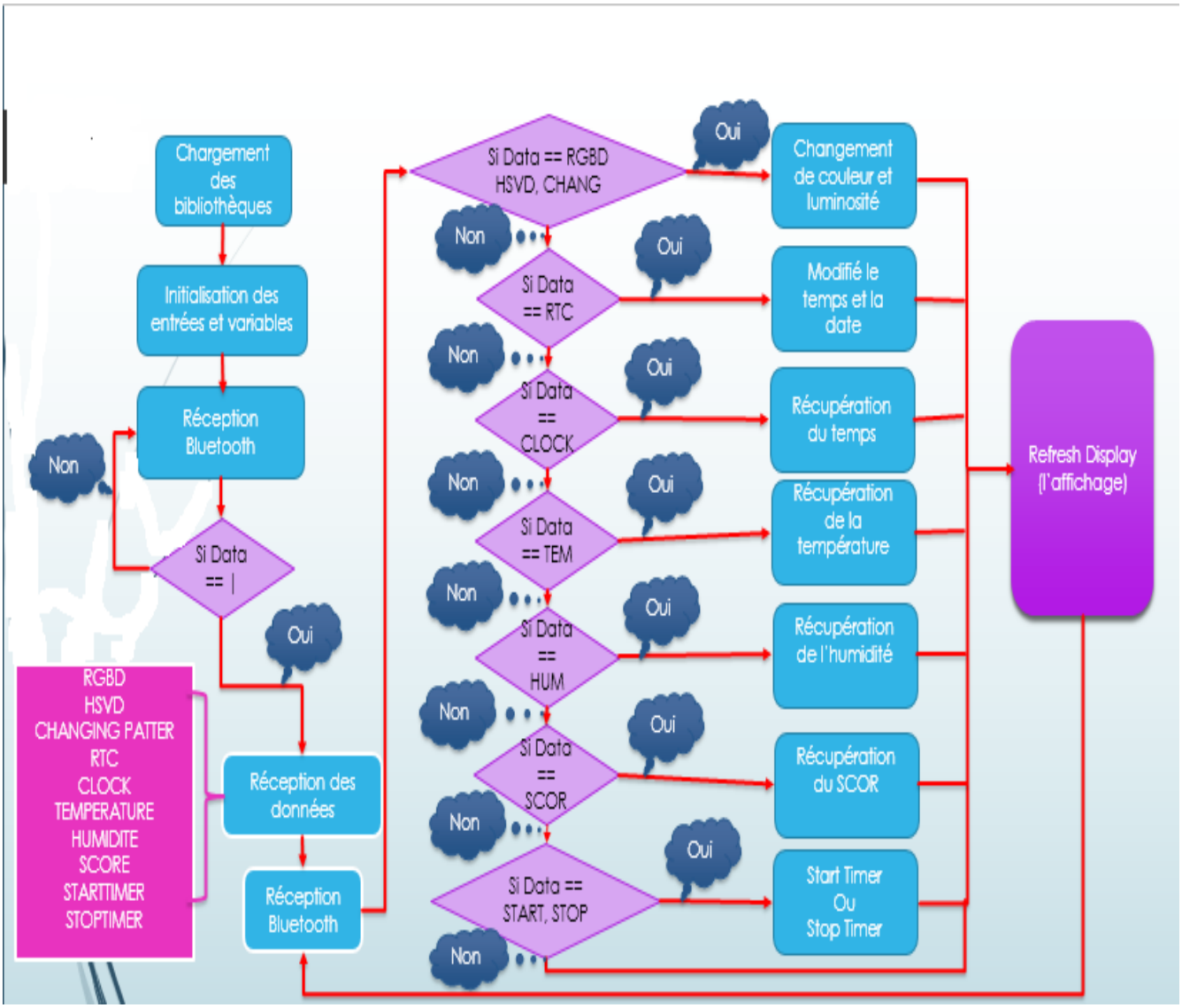


Figure III.8 L'organigramme de code global du circuit

III.3.7 Programme final de projet

Le programme final du projet d'horloge, affiché dans l'annexe de mémoire.

III.3.8 L'application Android qui contrôle l'horloge

III.3.8.1 Le rôle de l'application

L'application permet le contrôle de projet à distance via un module Bluetooth, vous devez télécharger une application Android permettant de contrôler Bluetooth s'ils rencontrent le désir, ou bien réaliser une application spéciale selon les besoins.

III.3.8.2 Présentation de l'application

J'ai téléchargé une application à partir de Play Store, après avoir pris connaissance des messages que l'application envoie à Bluetooth lorsque vous appuyez sur un bouton de l'application. Le visage complet de l'application présentée dans les figures suivantes

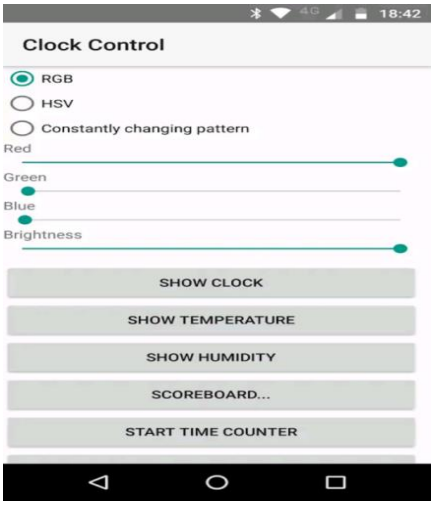


Figure III.9 Le premier aperçu de l'application

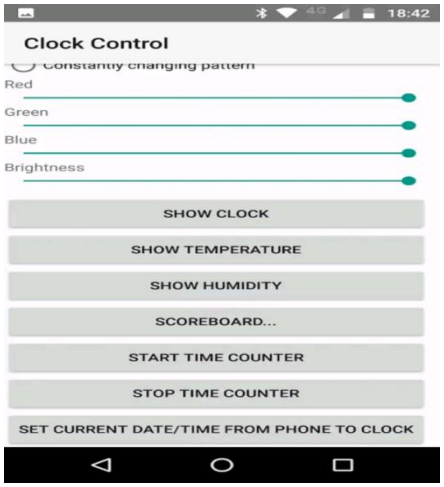


Figure III.10 Supplément du le premier aperçu de l'application

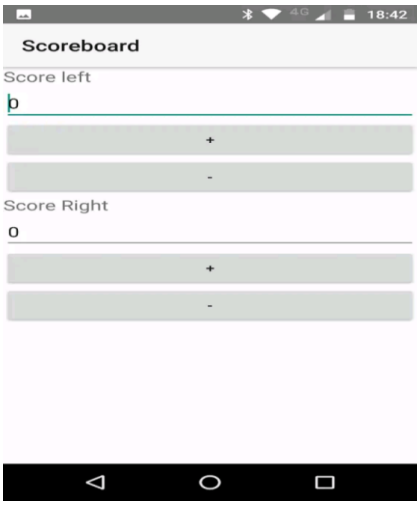


Figure III.11 La deuxième aperçu de l'application

III.3.8.3 Principe de fonctionnement de l'application

Lorsque vous appuyez sur n'importe quel bouton d'une application, il envoie un code à Bluetooth pour lui permettre de contrôler le projet. Tout cela est montré dans le Tableau III.4.

Bouton	**Code**
RGB	RGBD
HSV	HSVD
Constantly changing…	CHANGINGPATTERN
SHOW CLOCK	CLOCK
SHOW HUMIDITY	HUMIDITY
SHOW TEMPERATURE	TEMPERATURE
SCOREBOARD…	SCOREBOARD
START TIME COUNTER	STARTTIMER
STOP TIME COUNTER	STOPTIMER
SET CURRENT…	RTC

Tableau III.4 Le fonctionnement de l'application

III.4 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.

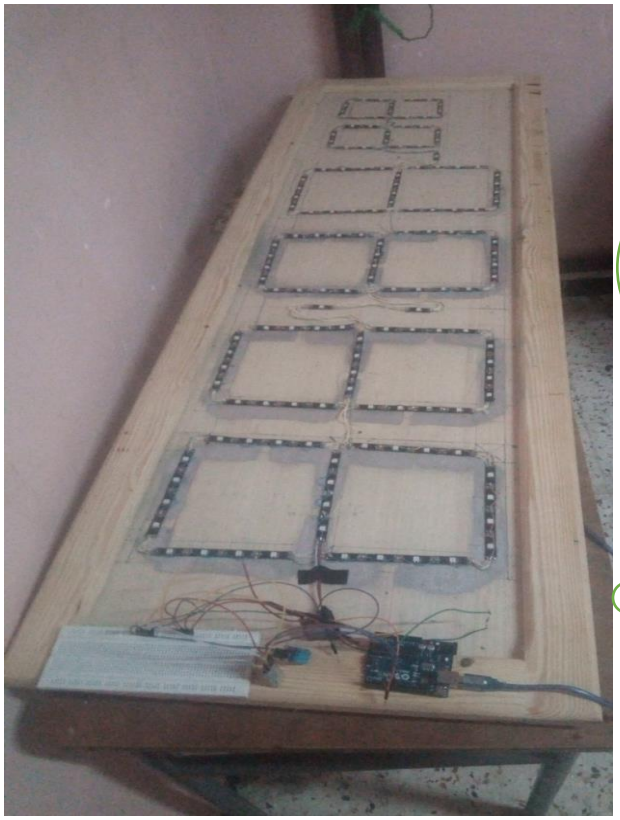
III.4.1 Matériaux utilisés

Pour réaliser le cadre il nous faut :

- Du bois.
- 160x40 cm du verre.
- De la colle.
- Du polystyrène expansé.
- De la peinture noire.
- Des clous

III.4.2 Fabrication du cadre

Le cadre en bois est une production locale, dont le résultant est donné sur la figure III.11



Les dimensions :

Largeur= 50 cm

Langueur= 150 cm

Poids= 7,20 kg

Figure III.12 Le cadre de l'horloge géante réalisé

III.5 Les différents résultants :

III.5.1 Résultat en mode horloge

Figure III.13 Résultats en mode horloge

III.5.2 Résultat en mode date

Figure III.14 Résultats en mode date

III.5.3 Résultat en mode humidité

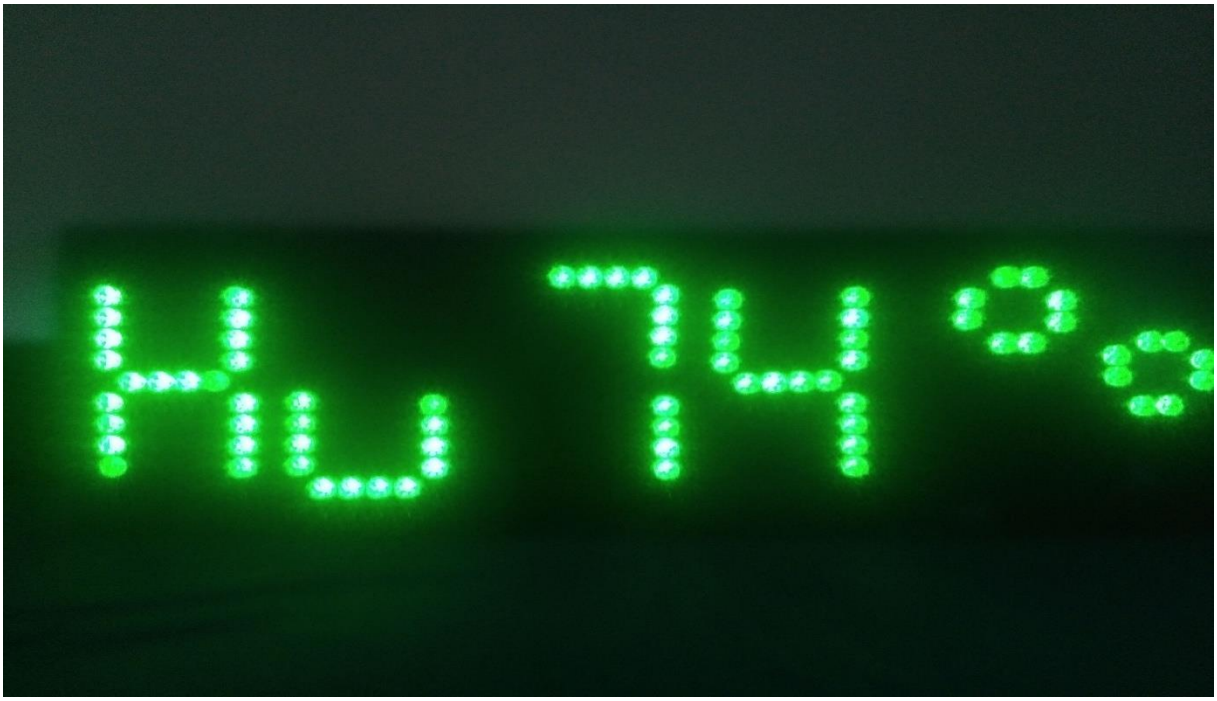


Figure III.15 Résultats en mode humidité

III.5.4 Résultat en mode température

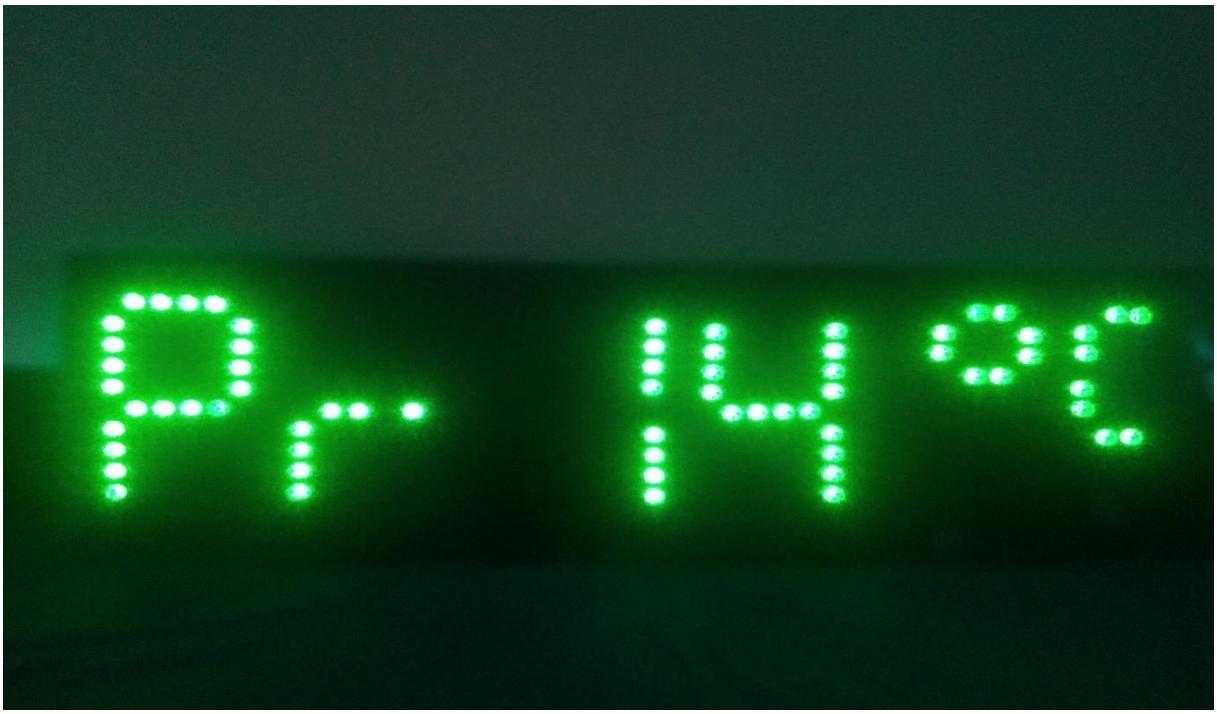


Figure III.16 Résultats en mode température

III.6 Conclusion

Ce chapitre, est la dernière partie de notre projet ou on a mis en épreuve notre partie théorique de la simulation sous Proteus. Pour avoir une horloge qui affiche l'heure et date correctement plus l’affichage de température et l’humidité, 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

Ce projet de mémoire de fin d'étude du Mestre 2, il était question pour l'amélioration des fonctionnalités de l'horloge géante de la FSSA. L'amélioration à réaliser dans ce travail est :

- L'affichage de la date complète (jour, mois, années) plus l'heure.
- L'affichage de la température et l'humidité à l'extérieur.
- L'affichage des annonces importantes telles que des lettres de l'administration aux étudiants ou des conférences.
- L'affichage de chronométré.
- Contrôlez l'affichage à distance, de manière meilleure et plus simple.
- Améliorer les éléments utilisés dans le projet en fonction de leur disponibilité sur le marché.

Dans mon projet, j'ai pu créer un modèle avec143 LED pour l'affichage, plus un ensemble d'éléments offrant les différentes options du projet, contrôlés par la carte arduino uno, commençant par la conception du circuit électrique, passant par la simulation sous ISIS m'a permis d'ajuster et de modifier le circuit, puis la réalisation des différents étages, par la suite, nous avons élaboré des organigrammes pour aboutir au programme arduino.

Au cours de ce travail, j'ai pu atteindre tous les objectifs souhaités dans cette amélioration a l'exception de la possibilité de publier des annonces, et cela est dû au manque de temps et de capacités matérielles, et en d'autres termes ce qui j'ai réalisé dans ce projet est considéré comme satisfaisant afin que tout le travail humain reste loin de la perfection et a besoin de développements et de reformes. Pour ce là j'ai proposée comme améliorations, d'ajouter une matrice LED afin de diffuser des annonces. Et utilisez le module wifi pour contrôler davantage la distance dans le projet.

Bibliographie

Bibliographie

[1] : http://ardavin.net/fr/content/15-histoire-des_horloges [16/09/2019]

[2] : http://au-fil-du-temps.e-monsite.com/pages/les-premieres-horloges/les-premieres-horloges.html. [16/09/2019]

[3] : https://www.jaycar.com.au/fnd507-lts542r-s50-rwb-common-anode-segment-display/p/ZD1857. [16/092019]

[4] : https://www.robot-maker.com/forum/tutorials/article/41-composant-les-afficheurs-7-segments/ [16/09/2019]

[5] : https://simple-duino.com/progressbar-arduino/. [18/09/2019]

[6] : https://www.gemaganga.com/index.php?main_page=product_info&products_id=110093 [18/09/2019]

[7] : https://www.aurel32.net/elec/lcd.php. [16/09/2019]

[8] : https://www.lepoint.fr/montres/Magazine/Dictionnaire-horlogerie/analogique-10-12-2012-2018339_2975.php [20/09/2019]

[9] : Boulouiz Youssouf *« Etude & réalisation d'un panneau lumineux »* Mémoire de master 2, Université, Aboubakr Blekaid- Tlemcen, 2017

[10] : https://www.grazia.fr/mode/analogique-ou-numerique-quelle-montre-choisir-838506 [20/09/2019]

[11] : https://fr.sport-thieme.ch/%C3%89quipement/Montres_et_horloges/art=1876505 [20/09/2019]

[12] : Livret Atelier Arduino en français par Jean-Noël Montagné, Centre de Ressources Art Sensitif, novembre 2006, page : 8

[13] : https://arduino.developpez.com/tutoriels/cours-complet-arduino/?page= programmer-arduino [2/10/2019]

[14] : Benaddi Harrage & Bououda Bouabdellah *« Conception et réalisation d'un journal lumineux à base d'un arduino »* Mémoire de master2, Université Abdelhamid Ibn Badis de Mostaganem, 2018.

[15] : DJAFRI Menad CHELOUCHE Djalal « Etude et réalisation d'une carte arduiono » Mémoire de master 2, Université A.MIRA DE BEJAIA, 2016.

[16] : https://fr.flossmanuals.net/arduino/historique-du-projet-arduino [2/10/2019]

[17] : https://arduino.developpez.com/tutoriels/cours-complet-arduino/?page=historique-du-projet-arduino [3/10/2019]

[18]:https://www.editionseni.fr/open/mediabook.aspx?idR=8457f073dc836b48a324c65a97bdc71b [3/10/2019]

[19]: http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno [10/10/2019]

[20] : https://www-lisic.univ-littoral.fr/~hebert/microcontroleur/atmel/ [10/10/2019]

[21] : https://www.academia.edu/12257228/Microcontr%C3%B4leurs_E13_Option_AGI [10/10/2019]

[22] : https://lewebpedagogique.com/inseiffel/files/2017/04/Carte-de-d%C3%A9veloppement-Arduino.pdf [19/10/2019]

[23] : https://datasheets.maximintegrated.com/en/ds/DS1307.pdf [02/11/2019]

[24] : https://www.netrozim.co.zw/product/real-time-clock-rtc-ds1307/ [02/11/2019]

[25] : https://www.elprocus.com/rtc-ds1307/ [02/11/2019]

[26] : http://bravo.univ-tln.fr/er/Cours%20divers/I2C_cours_2005.pdf [03/11/2019]

[27] : https://www.electronicshub.org/basics-i2c-communication/ [03/11/2019]

[28] : https://f-leb.developpez.com/traductions/arduino-i2c-FR/ [03/11/2019]

[29] : https://f-leb.developpez.com/tutoriels/arduino/bus-i2c/ [03/11/2019]

[30] : http://www.gecif.net/articles/genie_electrique/cours/terminal/cours/le_bus_I2C.pdf [03/11/2019]

[31] : https://wiki.eprolabs.com/index.php?title=Bluetooth_Module_HC-05 [10/11/2019]

[32] : https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/ [10/11/2019]

[33] : https://fotaks.com/produt/dht11-temperature-and-humidity-sensor-module-for-arduino/ [10/11/2019]

[34] : https://www.kitronik.co.uk/pdf/WS2812B-LED-datasheet.pdf [12/11/2019]

[35] : http://arabic.bendableledstrip.com/supplier-246560-ws2812b-led-strip [12/11/2019]

Annexe

Le programme de la simulation sur proteus

```
#include <Wire.h>
#include "RTClib.h"
#include <DHT.h>
#include <LiquidCrystal.h> // bibliothèque de LCD

//lcd module connections(RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // connections des ports de LCD
#define DHTPIN 12
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

RTC_DS1307 rtc;
#define hourFormat  24            // Set this to 12 or to 24 hour format

void setup() {
 lcd.begin(16, 2);
 Serial.begin(9600);
  while (!Serial) { /* Wait until serial is ready */ }
  dht.begin();
   if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
   }
     if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
     rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
      // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
     }
```

```
 pinMode(13, INPUT_PULLUP);
 pinMode(11, INPUT_PULLUP);
 pinMode(10, INPUT_PULLUP);
 pinMode(9, INPUT_PULLUP);
 pinMode(8, INPUT_PULLUP);
 pinMode(A0, INPUT_PULLUP);
 pinMode(A1, INPUT_PULLUP);
 pinMode(A2, INPUT_PULLUP);
 pinMode(A3, INPUT_PULLUP);
}

void loop() {

 int a1 = digitalRead(13);
 int a2 = digitalRead(11);
 int a3 = digitalRead(10);
 int a4 = digitalRead(9);
 int a5 = digitalRead(8);
 int a6 = digitalRead(A0);
 int a7 = digitalRead(A1);
 int a8 = digitalRead(A2);
 int a9 = digitalRead(A3);
 if(a1 == HIGH){
   lcd.clear();
   lcd.setCursor(2, 0);
   lcd.print(" CLOCK ");
   DateTime now = rtc.now();
   int h = now.hour();
   int mm= now.minute();
   int s = now.second();
```

```
  int y = now.year();
  int n = now.month();
  int dd = now.day();
  if (hourFormat == 12 && h > 12){
  h = h - 12;}
   lcd.setCursor(0, 1);
  lcd.print("Time: ");
  lcd.print(h);
  lcd.print(":");
  lcd.print(mm);
  lcd.print(":");
  lcd.print(s);
  lcd.print(" ");
   delay(50);
if(a6 == HIGH){
mm += 1;
if (mm == 30){
  h+=1;
  rtc.adjust(DateTime(y, n, dd, h, mm, s));
}
 rtc.adjust(DateTime(y, n, dd, h, mm, s));
}else if(a7 == HIGH){
 h += 1;
 if(h == 24){
  h = 00;
  rtc.adjust(DateTime(y, n, dd, h, mm, s));
 }
  rtc.adjust(DateTime(y, n, dd, h, mm, s));
}
}
```

```
if(a2 == HIGH){
  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("LA DATE: ");
  DateTime now = rtc.now();
  int y = now.year();
  int n = now.month();
  int dd = now.day();
  int h = now.hour();
  int mm= now.minute();
  int s = now.second();
  lcd.setCursor(0, 1);
  lcd.print(" ");
  lcd.print(y);
  lcd.print("/");
  lcd.print(n);
  lcd.print("/");
  lcd.print(dd);
  lcd.print(" ");
  delay(50);
  if(a8 == HIGH){
 dd += 1;
 if (dd == 31){
  n+=1;
  dd=0;
  rtc.adjust(DateTime(y, n, dd, h, mm, s));
 }
 rtc.adjust(DateTime(y, n, dd, h, mm, s));
 }else if(a9 == HIGH){
  n += 1;
```

```
  if(n == 31){
   y+=1;
   n=0;
   rtc.adjust(DateTime(y, n, dd, h, mm, s));
  }
  rtc.adjust(DateTime(y, n, dd, h, mm, s));
 }
}
if(a3 == HIGH){
   lcd.clear();
   lcd.setCursor(2, 0);
   lcd.print("TEMPERATURE:");
  int tmp = dht.readTemperature();
  if (isnan(tmp)) {
  lcd.clear();
  lcd.setCursor(5, 0);
  lcd.print("Error");
  delay(50);
  return;
 }
  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.print(tmp);
  lcd.print("C,");
   delay(50);
}
if(a4 == HIGH){
lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("HUMIDITY:");
```

```
  int hum = dht.readHumidity();
  if (isnan(hum)) {
  lcd.clear();
  lcd.setCursor(5, 0);
  lcd.print("Error");
  delay(50);
  return;
 }
  lcd.setCursor(0, 1);
  lcd.print("Hum: ");
  lcd.print(hum);
  lcd.print("% ");
   delay(50);
}
if(a5 == HIGH){
#define hourFormat  12
 }

}
```

Programme final de projet

```
#include <DHT.h>
#include <FastLED.h>
#include <Wire.h>
#include "RTClib.h"
#include <SoftwareSerial.h>
#include "Timer.h"

#define DHTPIN 12
```

```
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define NUM_LEDS 45
#define DATA_PIN 6
CRGB LEDs[NUM_LEDS];

SoftwareSerial BTserial(8, 9);
RTC_DS1307 rtc;
Timer t1;
Timer t2;
Timer t3;

String btBuffer;
CRGB colorCRGB = CRGB::Red;          // Change this if you want another default color, for example CRGB::Blue
CHSV colorCHSV = CHSV(95, 255, 255);  // Green
CRGB colorOFF  = CRGB(0,0,0);      // Color of the segments that are 'disabled'. You can also set it to CRGB::Black
volatile int colorMODE = 1;          // 0=CRGB, 1=CHSV, 2=Constant Color Changing pattern
volatile int mode = 04;         // 0=Clock, 1=Temperature, 2=Humidity, 3=Scoreboard, 4=Time counter
volatile int scoreLeft = 0;
volatile int scoreRight = 0;
volatile long timerValue = 0;
volatile int timerRunning = 0;

#define blinkDots   1               //   0 to disable , 1 to blink in clock mode,
#define hourFormat  24              // Set this to 12 or to 24 hour format
#define temperatureMode 'C'         // Set this to 'C' for Celcius or 'F' for Fahrenheit
```

```
void setup () {

  // Initialize LED strip
  FastLED.delay(60);

  FastLED.addLeds<WS2812B, DATA_PIN, GRB>(LEDs, NUM_LEDS);

  Serial.begin(9600);
  while (!Serial) { /* Wait until serial is ready */ }

  BTserial.begin(9600);
  Serial.println("BTserial started at 9600");

  dht.begin();

  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
     // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }

  t1.every(1000 * 29, refreshDisplay);
  t2.every(1000, refreshTimer);
  t3.every(50, updateHue);
  refreshDisplay();
}
```

```
void loop () {

  t1.update();
  t2.update();
  t3.update();

  if (BTserial.available())
  {
    char received = BTserial.read();
    btBuffer += received;

    if (received == '|')
    {
       processCommand();
       btBuffer = "";
    }
  }
}
void processCommand(){

  if (btBuffer.startsWith("RGBD")) {
    long R = getValue(btBuffer, ',', 1).toInt();
    long G = getValue(btBuffer, ',', 2).toInt();
    long B = getValue(btBuffer, ',', 3).toInt();
    long D = getValue(btBuffer, ',', 4).toInt();
    colorCRGB.red = R;
    colorCRGB.green = G;
    colorCRGB.blue = B;
    colorMODE = 0;
```

```
  if (D > 0) FastLED.setBrightness(D);
 } else if (btBuffer.startsWith("HSVD")) {
  long H = getValue(btBuffer, ',', 1).toInt();
  long S = getValue(btBuffer, ',', 2).toInt();
  long V = getValue(btBuffer, ',', 3).toInt();
  long D = getValue(btBuffer, ',', 4).toInt();
  colorCHSV.hue = H;
  colorCHSV.sat = S;
  colorCHSV.val = V;
  colorMODE = 1;
  if (D > 0) FastLED.setBrightness(D);
 } else if (btBuffer.startsWith("RTC")) {
  long y = getValue(btBuffer, ',', 1).toInt();
  long m = getValue(btBuffer, ',', 2).toInt();
  long d = getValue(btBuffer, ',', 3).toInt();
  long h = getValue(btBuffer, ',', 4).toInt();
  long mm = getValue(btBuffer, ',', 5).toInt();
  long s = getValue(btBuffer, ',', 6).toInt();
  rtc.adjust(DateTime(y, m, d, h, mm, s));
  Serial.println("DateTime set");
   mode = 5;
 } else if (btBuffer.startsWith("CLOCK")) {
  mode = 0;
 } else if (btBuffer.startsWith("TEMPERATURE")) {
  mode = 1;
 } else if (btBuffer.startsWith("HUMIDITY")) {
  mode = 2;
 } else if (btBuffer.startsWith("SCOREBOARD")) {
   scoreLeft = getValue(btBuffer, ',', 1).toInt();
  scoreRight = getValue(btBuffer, ',', 2).toInt();
```

```
    mode = 3;
  } else if (btBuffer.startsWith("STARTTIMER")) {
    timerValue = 0;
    timerRunning = 1;
    mode = 4;
  } else if (btBuffer.startsWith("STOPTIMER")) {
    timerRunning = 0;
    mode = 4;
  } else if (btBuffer.startsWith("CHANGINGPATTERN")) {
    colorMODE = 2;
  }

  refreshDisplay();
}

void updateHue() {
  if (colorMODE != 2)
    return;

  colorCHSV.sat = 255;
  colorCHSV.val = 255;
  if (colorCHSV.hue >= 255){
    colorCHSV.hue = 0;
  } else {
    colorCHSV.hue++;
  }
  refreshDisplay();
}

void refreshTimer() {
```

```
  if (mode == 0 && blinkDots == 1) {
    displayDots(3);
  } else if (mode == 4 && timerRunning == 1 && timerValue < 6000) {
    timerValue++;

    int m1 = (timerValue / 60) / 10 ;
    int m2 = (timerValue / 60) % 10 ;
    int s1 = (timerValue % 60) / 10;
    int s2 = (timerValue % 60) % 10;
    displaySegments(0, m1);
    displaySegments(7, m2);
    displaySegments(16, s1);
    displaySegments(23, s2);
    displaySegments(31, 13);
    displaySegments(38, 13);
     displayDots(0);
    FastLED.show();
  }
}
void refreshDisplay() {
  switch (mode) {
    case 0:
      displayClock();
      break;
    case 1:
      displayTemperature();
      break;
    case 2:
      displayHumidity();
```

```
      break;
    case 3:
      displayScoreboard();
      break;
    case 4:
      // Timer
      break;
      case 5:
      displayDate();
      break;
    default:
      break;
  }
}
void displayClock() {
  DateTime now = rtc.now();

  int h  = now.hour();
  if (hourFormat == 12 && h > 12){
    h = h - 12;}

  int hl = (h / 10) == 0 ? 13 : (h / 10);
  int hr = h % 10;
  int ml = now.minute() / 10;
  int mr = now.minute() % 10;
  displaySegments(0, hl);
  displaySegments(7, hr);
  displaySegments(16, ml);
  displaySegments(23, mr);
  displaySegments(31, 13);
```

```
  displaySegments(38, 13);
  displayDots(1);
  FastLED.show();
}

void displayTemperature() {
  float tmp = dht.readTemperature(temperatureMode == 'F' ? true : false);

  if (isnan(tmp)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    int tmp1 = tmp / 10;
    int tmp2 = ((int)tmp) % 10;
    displaySegments(0, 15);
    displaySegments(7, 16);
    displaySegments(16, tmp1);
    displaySegments(23, tmp2);
    displaySegments(31,  10);
    displaySegments(38, (temperatureMode == 'F' ? 14 : 11));
    displayDots(1);
    FastLED.show();
  }
}

void displayHumidity() {
  float hum = dht.readHumidity();

  if (isnan(hum)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
```

```
    int hum1 = hum / 10;
    int hum2 = ((int)hum) % 10;
    displaySegments(0, 17);
    displaySegments(7, 18 );
    displaySegments(16, hum1);
    displaySegments(23, hum2);
    displaySegments(31,  10);
    displaySegments(38,  12);
    displayDots(1);
    FastLED.show();
  }
}

void displayScoreboard() {

  int s1 = scoreLeft % 10;
  int s2 = scoreLeft / 10;
  int s3 = scoreRight % 10;
  int s4 = scoreRight / 10;
  displaySegments(0, s4);
  displaySegments(7, s3);
  displaySegments(16, s2);
  displaySegments(23, s1);
  displayDots(2);

  FastLED.show();
}
void displayDate(){
  DateTime now = rtc.now();
  int y = now.year() % 100;
```

```
  int yl = y / 10;
  int yr = y % 10;
  int nl = now.month() / 10;
  int nr = now.month() % 10;
  int dl = now.day() / 10;
  int dr = now.day() % 10;
  displaySegments(0, dl);
  displaySegments(7, dr);
  displaySegments(16, nl);
  displaySegments(23, nr);
  displaySegments(31, yl);
  displaySegments(38, yr);
  displayDots(2);
   FastLED.show();
}

void displayDots(int dotMode) {
  switch (dotMode) {
    case 0:
      LEDs[14] = colorMODE == 1 ? colorCRGB : colorCHSV;
      LEDs[15] = colorMODE == 1 ? colorCRGB : colorCHSV;
      break;
    case 1:
      LEDs[14] = colorOFF;
      LEDs[15] = colorOFF;
      LEDs[30] = colorOFF;
      break;
    case 2:
      LEDs[14] = colorOFF;
      LEDs[15] = colorMODE == 1 ? colorCRGB : colorCHSV;
```

```
      LEDs[30] = colorMODE == 1 ? colorCRGB : colorCHSV;
      break;
    case 3:
      LEDs[14] = (LEDs[14] == colorOFF) ? (colorMODE == 1 ? colorCRGB : colorCHSV) :
colorOFF;
      LEDs[15] = (LEDs[15] == colorOFF) ? (colorMODE == 1 ? colorCRGB : colorCHSV) :
colorOFF;
      FastLED.show();
      break;
    default:
      break;
  }
}

void displaySegments(int startindex, int number) {

 byte numbers[] = {
    0b01111110, // 0
    0b01000010, // 1
    0b00110111, // 2
    0b01100111, // 3
    0b01001011, // 4
    0b01101101, // 5
    0b01111101, // 6
    0b01000110, // 7
    0b01111111, // 8
    0b01101111, // 9
    0b00001111, // º              10
    0b00111100, // C(elcius)      11
    0b01110001, // º lower        12
    0b00000000, // Empty          13
```

```
    0b00011101, // F(ahrenheit)   14
    0b00011111, //p              15
    0b00010001, //r              16
    0b01011011, //h              17
    0b01110000, //u              18
  };


  for (int i = 0; i < 7; i++) {
    LEDs[i + startindex] = ((numbers[number] & 1 << i) == 1 << i) ? (colorMODE == 0 ? colorCRGB : colorCHSV) : colorOFF;
  }
}


String getValue(String data, char separator, int index) {
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length()-1;


  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==separator || i==maxIndex){
        found++;
        strIndex[0] = strIndex[1]+1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }


  return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```