الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

DEPARTEMENT GENIE MECANIQUE

SPECIALITE : MECATRONIQUE



Mémoire de Fin d'Études

Conceptionet Réalisation de l'horloge numérique intelligent

Proposé et dirigé par,

Dr kH. BERRAHAL

Préparé et présenté par,

- *DJEMAI DOUNIA ZED*
- *DERIADI SOMIA*

Remerciements

Au terme de ce travail . on remercie en premiere lieu allah qui a donnè la force pour achiever cette ètude.

On remercie nos très chèrs parents pour leurs soutiens et leur patiences .

On expreme notre profonde gratitude à notre promoteur : dr Kh benrahal , pour ses consiel et ses aides prècieuses .

Nos sincère remerciement s'adressent ègalement à tous les enseignants du dèpartement de genie mecanique de l'universitè de Mèdèa .

Notre gratitude à tous les membres de jury qui font l'honneur de juger notre travail

Un grand merci à tous ceux qui contribuè de près ou de loin à l'èlaboration de ce projet

.

Dédicace

Je dédie mon diplôme à celui qui a englouti la coupe vide pour me donner une goutte d'amour A celui qui a récolté les épines pour mon chemin Pour ouvrir le chemin de la connaissance, mon père et ma mère

Mon père, qui a lutté contre les difficultés et les obstacles de la vie afin de me rendre heureuse et de réaliser le rêve d'éducation et d'obtention du diplôme.

Et **ma mère** qui m'a porté pendant neuf mois et m'a donné naissance pour la vie, et elle n'était pas satisfaite de ça jusqu'à ce que j'atteigne ce que je suis

À la fontaine de la patience, de l'optimisme et de l'espoir, ma grand-mère et mon grand-père bien-aimés.

Aux cœurs purs et doux et aux âmes innocentes Aux vents de ma vie mes frères, surtout ma seconde moitié, mon frère, plutôt mon fils **Achraf** et son rire qui fait pleuvoir la paix dans mon cœur.

A la personne la plus précieuse et merveilleuse qui est entrée dans ma vie par amour dans tous ses sens, donc le lien et la tendresse étaient à mon amour et mon fiancé **Farouk**. Je ne dirai pas merci, mais je vivrai toujours grâce à toi.

Et à tous mes proches de la famille, amis et professeurs qui m'ont accompagné et m'ont encouragé à rester debout sur ce chemin....vous avez plus que moi, mon amour et ma gratitude

Surtout mon partenaire et ami dans ce projet **Dounia, Sabrina, Siham, Djehan, Samia, Fatiha**

Somia

Dédicace

Je didie ce modeste travail à :

Au premiere lieu , nous tenons à remercier « allah » qui nous à donnée le courage et la volontè pour terminer ce travail.

Mes chers parents que dieu les protège, pour leurs soutien durant la pèriode de ma formation , pour leurs bienveillance .

Ma sœur et mes frères (khaled , abde elhak , AFEF , Mohammed Badr) et tous ma famille sans exception .

Tout mes aimes surtôt :(Lamia, Hassna , Abir , Sabrina , Malek , ……..)

En fin , je remercié mon binôme Soumia qui a contribué à la réalisation de ce modeste travail .

Mes enseignant qui mon contribuer durant ma formation .

Dounia zed

Table des matières

Sommaire……..I

Liste des figures……II

Liste des tableaux…….III

Liste des acronymes et des abréviations……...V

Résume……VI

Introduction Générale……1

Chapitre I : Généralités sur NODE MCU(ESP8266) et Arduino

I.1Introduction……...3

I.2 Système embarque ……..3

I.2.1 LA définition de système embarque……..3

I.2.2Caractèrèstique d'un système embarque……3

I.2.3La composition de système embarque……...4

I.2.3 Spécifiés d'un système embarque ……4

I.3 Partie Hardware……....4

I.3.1 Le microcontrôleur……....4

I.3.1.1 Définition de microcontrôleur……5

I.3.1.2 Les avantage d'un microcontrôleur……5

I.3.1.3 contenue d'un microcontrôleur……...5

I.4 Les langage de programmation du microcontrôleur ……...6

I.4.1 Langage c……...6

I.4.2 Langage java……7

I.5 La carte électronique de l'Arduino……...7

I.5.1 définition de la carte Arduino……...7

I.5.2 les gammes de la carte Arduino……....7

I.6 La plate forme NODE MCU (ESP8266)……..8

I.6.1 Définition la plate forme Node MCU (esp8266)……..8

I.6.2 Description technique de l'ESP8266……9

I.6.3 les avantage de l'ESP8266……..10

I.6.4 Brochage de l'ESP8266……..10

I.6.5 les caractéristiques de l'ESP8266……11
I.6.6 Mise sous tension du module……...12
I.6.7 La mémoire……...13
I.6.8 Programmation……14
I.6.8.1 Programmation NODE MCU (ESP8266)……14
I.6.8.1 Téléchargement de votre premier programme……...14
I.6.8.2 Applications de NODE MCU (ESP8266)……15
I.6.19 Conclusion sur la carte ……...15
I.7 Partie Software……15
I.7.1 Arduino IDE……...15
I.7.1.1 L'environnement de la programmation……15
I.7.1.2 Structure générale du programme (IDE Arduino) ……...15
I.7.1.3 les étapes de téléchargement du programme……....16
I.7.1.4 Description du programme……17
I.7.1.5 Injection du programme……....17
I.8 conclusion……18

Chapitre II : Etude et simulation de l'horloge numérique intelligente

II.1 Introduction……20
II.2 Principe de fonctionnement……20
II.3 Déroulement du projet……20
II.4 Le module RTC ……....20
II.4.1 Définition de RTC…….....20

II.4.2Principe de fonctionnement du moduleRTC3231 ……..21
II.4.3Puce DS3231 RTC ou gros plan de la puce ……21
II.4.4 Les caractéristiques d'un module RTC3231……...22
II.4.5 Brochage le module RTC DS3231 avec Node MCU (esp8266)…….23
II.5 Capteur humidité DHT11……23
II.5.1 les caractéristiques de DHT11……..23
II.5.2 Brochage d'un capteur d'humidité DHT11 avec NodeMCU (esp8266)…….....24
II.6 Led Matrice Md7219……...24
II.6.1 Le type de Led matrice……24
II.6.2 Construction de la matrice de points……25

II.6.3 Fonctionnement de la Matrice des LEDs 8*8…….25
II.6.4 Le circuit intégré Max 8219(ou Max 7221)…….26
II.6.5 Connexions du module à L'esp12F ou ESP8266 ……...26
II.7 Schéma synoptique générale……...27
II.8 Le logiciel Fritzing……...27
II.8.1 Définition de programme de Fritzing……27
II.8.2 Présentation de logiciel……..28
II.8.2.1 L'interface de Fritzing……28
II.8.2.1.1 Breadboard sur Fritzing……...28
II.8.2.1.2 Schematic…….29
II.8.2.1.3 Le circuit imprimé Pcb……....29
II.8.2.1.4 Code……30
II.8.2.2 Les composant sur Fritzing……30
II.9 Démarche de la simulation……...31
II.10 Circuit globale de la simulation…….32
II.11 conclusion……..33

Chapitre III Réalisation pratique de l'horloge

III.1 Introduction……...35
III.2 Déroulement de la réalisation de l'horloge……35
III.3 Composants utilisés……....35
III.4 Schéma global du montage……35
III.5 Partie atelier……...37
III.5.1 Assemblage de l'horloge…….37
III.5.1.1 Conception de la boite à l'horloge……...37
III.5.1.1.1 Définition générale du solide Works……37
III.5.1.1.2 Industrie de la boite……..38
III.6 Partieprogrammation…….39
III.6.1 Organigramme de fonction fondamentale……..39
III.6.2 Organigramme de fonction secondaire……...41
III.6.2.1 Fonction lire temps……..41
III.6.2.2 Fonction lire Temps/Hum……41

III.6.2.3Fonction Beep Sonore…….42
III.7 L'application Blynk…….43
III.7.1 Commente fonctionne Blynk ……...43
II.7.2 La creation de compete Blynk……44
III.7.3 Crèèrun nouveau Projet…….45
III.7.4 La réalisation d'application……..46
III.8 Résultat de la réalisation……...49
III.9 Obstacle rencontré pendant la réalisation ……...49
III.10 Conclusion…….49
Conclusion Générale……50
Référence…….51

Liste des figures

FIG I.1 Schéma illustratif des composants d'un système embarque……..4
FIG I.2 Structure interne d'un système microcontrôleur……6
FIG I.3 Module ESP8266 sur NodeMCU……..9
FIG I.4 Composition d'une carte NodeMCU ESP8266……11
FIG I.5 Les pins d'alimentation électrique du NodeMCU ESP8266……13
FIG I.6 Interface IDE Arduino…….16
FIG I.7 Les étapes de téléchargement du code…….17
FIG I.8 paramétrage de code…….18
FIG II.1 Le module RTC……21
FIG II.2 Le module RTC ds3231……21
FIG II.3 Ds3231-RTC-Module-CR2032-Support de batterie……...22
FIG II.4 Capteur DHT11……...23
FIG II.5 Fonctionnement de la matrice……..25
FIG II.6 Les circuit intégré Max 7219……...26
FIG II.7 Connexions du module à L'ESP12Fou ESP8266……....26
FIG II.8SchémasynoptiqueL'horlogenumérique intelligent……...27
FIG II.9 Lafenêtre du logicielFritzing……28
FIG II.10 La plaque diesis (Breadboard)……...28
FIG II.11 Schéma de cablageune carte Arduinodans la catégorie Breadboard…….29
FIG II.12 Circuit imprimé PCB…….. 29
FIG II.13 Code du logiciel Fritzing……..30
FIG II.14 Les composantsurFritzing…….30
FIG II.15 Circuit global de simulation sous Fritzing……32
FIG III.1 Schéma global du montage électronique de l'horloge……..36
FIG III.2 Solide Works……..37
FIG III.3 Conception à Solid works……...38
FIG III. 4 Industries de la boite…….38
FIG III. 5 Organigramme explicatif du code final de l'horloge ……..40
FIG III.6 Fonctionnement de d'application Blynk……...44
FIG III.7 Création de compète Blynk……44
FIG III.8 Le type de connexion utilise dans l'application…….46
FIG III.9 Résultat de la réalisation……49

Liste des tableaux

Tableau I.1 : Les types de l'Arduino……8
Tableau II.1 : Les bibliothèques de tous les composants……31
Tableau II.2 Broche et connexion du circuit électronique……..32
Tableau III.1 : Les composants utilisés……...35
Tableau III.2 : Les brochages de la différente de composant…….36

Liste des acronymes et des abréviations

AREF : Alimentation de Référence

CS : Chips Select

CLK : Clock

DHT11 : Digital Humidité and température sensor

DIN : Dirctor Identification Number

DC : Direct Current

E/S : Entrée / Sortie

ESP8266 : Expressif modules 8266

EEPREOM : Electrically-Ecrasable Programmable Read-Onlay Memory

GND : Grounds

GPIO : General Prépose Input/Output

IDE:Integrated Développement Environnement

IP : Internet Protocole

LED : Light-Emitting Diode

MCU : Microcontrôleur unit

PWM : Pulse Width Modulation

Pcb : Printed circuit Board

SDK : Software Développement Kit

SCL : Serial Clock line

SQW : Sortie signal carré

SPI : Serial Puerpéral Interface

UART : Universelle Asynchrones ReceiverTransmitter

USB: Universelle Serial Bus.

RTC: Real Time Clock.

SOC : System on a Chip

SPIFFS:SPI Flash File System

SDA : Serial Clock line

VCC: Voltage Current Continu.

WIFI : Wireless fidélité

TR : Tempe Réel

Résume :

Ce manuscrit est une présentation du projet, où une horloge numérique intelligente a été conçu et réalisé. Une horloge qui nous permet reconnaître l'heure tout au long de la journée avec la date actuelle, ainsi que la température et l'humidité actuelles. On peut aussi transmettre des messages via l'application Blynk en appuyant sur un bouton poussoir. Ce mémo combine les différentes étapes de création d'une smart-watch numérique en utilisant la carte NodeMCU (ESP826) qui offre la possibilité d'utiliser le Wifi, en plus du module RTC , une Led max md7219, et le module DHT11 chargé de capturer la température et l'humidité, quant à la simulation, il a été implémenté dans le cadre de la programmation Fritzing et de la programmation sous Arduino IDE en plus du programme de dessin solide Works.

الملخص :

هذه المخطوطة عبارة عن عرض تقديمي لمشروع، حيث تم إنشاء ساعة رقمية ذكية تتيح لنا التعرف عل الوقت على مدار اليوم التاريخ الحالي و درجة الحرارة والرطوبة الحالية ,بالإضافة إلى توصيل رسالة عبر تطبيق BLYNKبضغط زر. تجمع هذه المذكرة بين المراحل المختلفة لإنشاء ساعة رقمية ذكية عبر(NodeMCUESP826) باستخدام, wifi بالإضافة إلى وحدة RTC وLed max md7219، و وحدةDHT11 مسؤولة عن التقاط درجة الحرارة والرطوبة، أما بالنسبة للمحاكاة ,فقدتم تنفيذها كجزء من برمج Fritzing والبرمجة تحت IDE Arduino بالإضافة إلى برنامج الرسمsolide Works

Abstract:

This manuscript is a presentation of the project, where a smart digital clock was created that allows us to recognize the time throughout the day with the current date, current temperature and humidity, as well as deliver messages via the Blynk app with the push of a button. This memo combines the different stages of creating a digital smart watch via NodeMCU (ESP826) using Wifi, in addition to the RTC module and Led max md7219, and the DHT11 module responsible for capturing temperature and humidity, as for the simulation, it was implemented as part of Fritzing programming and programming under Arduino IDE in addition to the drawing program solide works.

Introduction Générale :

Connaître le temps est important pour que les humains se connaissent. A notre époque, le temps est devenu une chose importante dans la vie, notamment en ce qui concerne l'organisation de la vie des sociétés. Les phénomènes périodiques de l'environnement dans lequel vit l'homme, tels que le déplacement quotidien de l'ombre, le retour des saisons, ou le cycle lunaire, ont servi de premières références. Mais peu à peu, l'homme s'est inspiré des phénomènes physiques, constatant leur caractère cyclique, pour concevoir et développer des appareils de mesure du temps de plus en plus précis, ainsi que des unités adaptées, l'horloge numérique en étant un exemple. Ce dispositif permet aux personnes de connaître l'heure à tout moment et en tout lieu et de connaître la température et l'humidité, et de programmer leurs journées de manière à rationner le temps, pour une meilleure organisation.

L'objectif de ce projet est de réaliser une montre numérique qui offre un support rapide avec différentes conceptions et fonctionnalités telles que la réception et le contrôle des messages, mais également l'affichage de la température et de l'humidité car elle fonctionne en parallèle. Ce projet se compose de trois chapitres dont le premier chapitre résume des Informations générales sur l'environnement Arduino et NODE MCU ESP8266, il couvre le chemin et les domaines d'utilisation, Ainsi que les différents composants qui composent les deux cartes et leurs rôles. Le chapitre sera deuxième dédié à l'étude et à la simulation d'une horloge numérique sous Fritzing. En plus de mentionner tout les composantes électroniques utilise dans le projet et comment les relier au NodeMCU esp8266. Quant au troisième et dernier chapitre, il est Nous sommes passés à la réalisation pratique de l’horloge numérique intelligente. Avec le code final pour exécuter l'élément, et à la fin nous nous retrouvons avec une conclusion générale suivie de points de vue sur la cognition.

Chapiter I :

Généralités sur NODE MCU(ESP8266) et Arduino

I.1 Introduction :

De nos jours (2021), l'électronique classique est de plus en plus remplacée par 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 :

Le terme "système embarqué "désigne un système électronique et informatique autonome dédie à une tâche précise, souvent en tempe réel, possèdent une taille limitée et ayant une consommation énergétique d'un restreinte [1].

I.2.2 Caractéristique d'un système embarque :

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 de système embarqué :

Quelle que soit sa nature et sa complexité, on décompose un système embarqué 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].

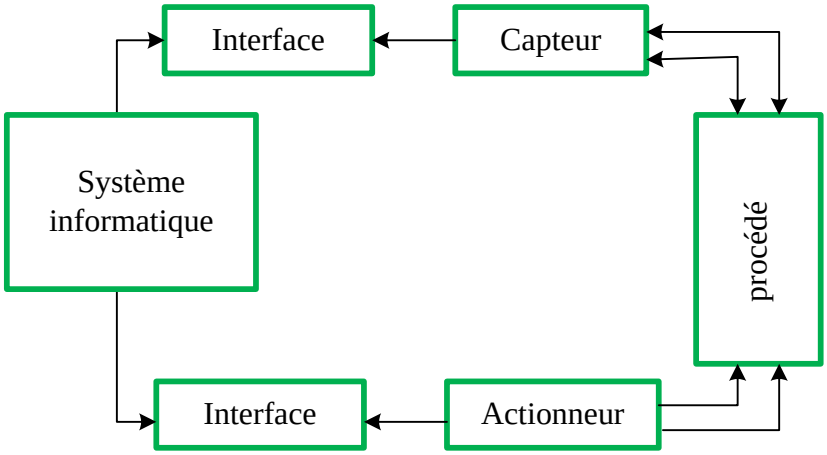


FIG I.1 Schéma illustratif des composants d'un système embarque

I.2.4 Spécificités d'un système embarqué :

Les caractéristiques principales d'un système électronique embarqué sont :

- **Autonomies** : Une fois 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'information extérieure non prévues [1].

  On abordera dans la prochaine section l'Arduino, un exemple des systèmes embarqués nécessaire pour la réalisation de notre horloge.

I.3 Partie matérielle hardware :

I.3.1 Le microcontrôleur :

I.3.1.1 Définition du microcontrôleur:

Un microcontrôleur est un composant électronique qui rassemble touts les éléments d'un mini–ordinateur et qui se présente sous la forme d'un circuit intégré. Un microcontrôleur permet de réaliser des systèmes et des montages électroniques programmés. Cela veut dire que l'on pourra, avec le même montage réaliser des fonctions très différentes qui dépendirent du programme qui aura état injecte dans le microprocesseur [2].

I.3.1.2 Les avantages d'un microcontrôleur:

L'utilisation des microcontrôleurs pour les circuits programmables a plusieurs points forts, il suffit s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants des circuits intégrés en ce domaine depuis quelque années [2].

Nous allons voire que le nombre d'entre eux découle du simple sens.

- Tout d'abord, un microcontrôleur intègre dans un seul et même boiter une dizaine d'éléments, il résulte donc une diminution évidente de l'encombrement du matériel et de circuit imprimé.
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisque 'il n'est pas plus nécessaire de véhiculer les, buse d'adresses et de donnée d'un composant à un autre.
- L'augmentation se la fiabilité du système puisque, le nombre des composant diminue.
- Le microcontrôleur contrebuter à réduire les couts à plusieurs nivaux [2].

  *moins cher que les autres composants qu'il remplace.

  *Démineur les coutes de main d'œuvre. Réalisation des applications non réalisable avec d'autre composant

I.3.1.3 constitution d'un microcontrôleur :

Un circuit microcontrôleur doit contenir dans un seul bottier les éléments de base qu'on verra par la suit, rêne effet, pour l'analyse des système réaliser avant l'avènement des microcontrôleurs

les fabricants des circuits intègre ont fané un peu définition de ce qu'il fallait intégrer pour arriver à un schéma type analogique à la figure (I.2) [2].

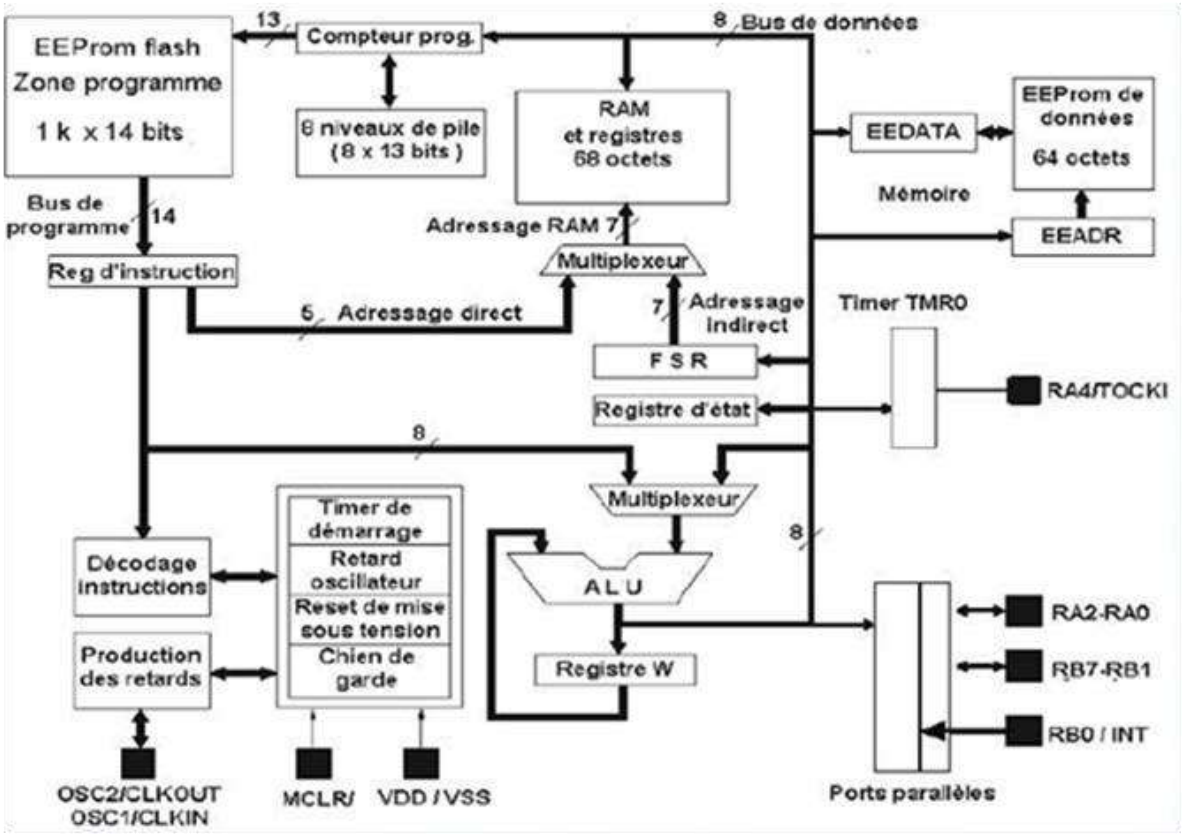


FIG I.2 : Structure interne d'un microcontrôleur.

I.4 Les langage de programmation du microcontrôleur :

I.4.1 Langage c :

Le langage de fonctionnement C est la référence pour programmer les microcontrôleurs avec des exigences temps réel. En effet, Le C présente l'avantage d'être un langage de programmation orienté matériel, le code machine généré par le compilateur est presque optimal en ce qui concerne d'une part la taille de la mémoire requise et d'autre part la vitesse d'exécution du programme. Par conséquent, environ 65% des applications embarqués sont programmées en C [2].

I.4.2 Langage java :

Est aujourd'hui rarement utilisé pour programmer l'application embarquée (moins de 3%). Java n'est pas adéquate pour les applications temps réels. En effet, ce langage est interprété (lent) et le comportement temporel du «récupérateur de place (anglais: Gerbage Collecter) n'est pas reproductible [2].

I.5 La carte électronique Arduino :

I.5.1a Définition de la carte Arduino :

Le module Arduino est une carte de programmation 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, le pilotage d'un robot, la commande des moteurs faire des jeux de lumières, communiquer avec l'ordinateur et commander des appareils mobiles (modélisme). Chaque module 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 [3].

I.4.2 Les gammes des cartes Arduino :

Actuellement, il existe plus de 20 versions du module Arduino, nous citons quelques uns afin d'évaluer de ce produit [3].

Tableau I.1 : les types de l'Arduino.

Version Arduino	Caractéristiques
Le NG d'Arduino	Une interface d'USB pour programme 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 petit carte programmée à l'aide d'un port USB cette version utilisant un microcontrôleur ATmega168. (ATmega328 pour les plus récentes).
Le livpad Arduino	Une conception de minimaliste pour l'application wear able en utilisant un microcontrôleur ATmega168.
Le NG d'Arduino plus	Avec une interface d'USB pour programmer et usage d'un microcontrôleur ATmega168.
L'Arduino Bluetooth	Avec une interface Bluetooth pour programme et usage d'un microcontrôleur ATmega168.
L'Arduino décimal	Avec une interface d'USB et utilise un microcontrôleur ATmega168.
L'Arduino demi la nove (2009)	Utilisant un microcontrôleur l'ATmega 168(ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC
L'Arduino méga	Utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
L'Arduino uno	Utilisation un microcontrôleur l'ATmega328.
I'Arduino Leonardo	Avec un morceau ATmega 3u4 qui éliminé le besoin de raccordement USB et peut être employé comme clavier
L'Arduino explora	Ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

II.6 Partie Hardware

I.6 La plate forme NODE MCU (ESP8266) :

I.6.1 Définition la plate forme NodeMCU (ESP8266) :

NodeMCU est une plate-forme open source, matérielle et logicielle basée sur un Soc Wifi ESP8266 fabriqué par Expressif Systems. Le terme "NodeMCU" se réfère par défaut au frimaire

plutôt qu'aux kits de développement. Le frimaire, permettant nativement l'exécution de scripts écrits en Lua, L'ESP 8266 est un circuit intégré à microcontrôleur avec connexion Wifi de taille réduite, permet de connecter un microcontrôleur à un réseau Wifi et d'établir des connexions TCP/IP avec des commandes AT. Il existe à ce jour plus de 12 versions de modules qui ont été construits à partir de ce composant. Chaque version est identifiée par une nomenclature sous la forme : ESP-01, ESP-02 ESP-12[4].



FIG I.3 : Module ESP8266 sur NodeMCU

L'ESP8266 peut se programmer de plusieurs façons :

- ✓ Avec des scripts Lua, interprétés ou compilés, avec le frimaire NodeMCU.
- ✓ En C++, avec l'IDE Arduino.
- ✓ En JavaScript, avec le frimaire Espriu.
- ✓ En Micro Python, avec le frimaire MicroPython5 ou Circuit Python,
- ✓ En C, avec le SDK d'Expressif.
- ✓ En C, avec le SDK esp-open-sdk7 basé sur la chaîne de compilation GCC.
- ✓ En Go, avec le Framework Goblot.

I.6.2 Description technique de L'ESP8266 :

Les spécifications principales de l'ESP8266 sont :

- ❖ Une pile de protocole TCP/IP intégrer.
- ❖ processeur 32 bits intégrer de faible puissance.
- ❖ Il fonctionne à 3,3 V et ne possède pas de régulateur de tension, il faudra donc bien veiller à toujours l'alimenter en 3,3 V et non en 5 V ou depuis un pack de piles.

- Le processeur est cadencé à 80 MHz (contre 16 MHz pour carte Arduino UNO) et possède 80 KB de RAM (contre 2,5 KB pour une carte Arduino UNO).
- WIFI direct 802.11 b/g/n.3.
- Il intègre une mémoire flash externe d'environ 400 KB (contre 32 KB pour une carte Arduino UNO) [5].

I.6.3 Les avantages de la carte ESP8266 :

- Pas cher.
- Environnement de programmation clair et simple (Arduino
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses librairies disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc.).
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires [4].

I.6.4 Brochage de l'ESP8266 :

Le NodeMCU ESP8266 possède un total de 30 broches (pin) nous permettant de le brancher à d'autres périphériques et supportant des fonctionnalités comme PWM, I2C, SPI et UART. Remarque : Pour une raison inconnue, les numéros de pins écrit sur la carte NodeMCU ne Correspond pas à celle de l'ESP8266 et donc pas à celle de l'IDE Arduino lors de la Programmation. La FIG I.4 montre la correspondance entre les noms des ports tels qu'indiqués sur La carte, les GPIO et les fonctions spécifiques associées à chacun [5].

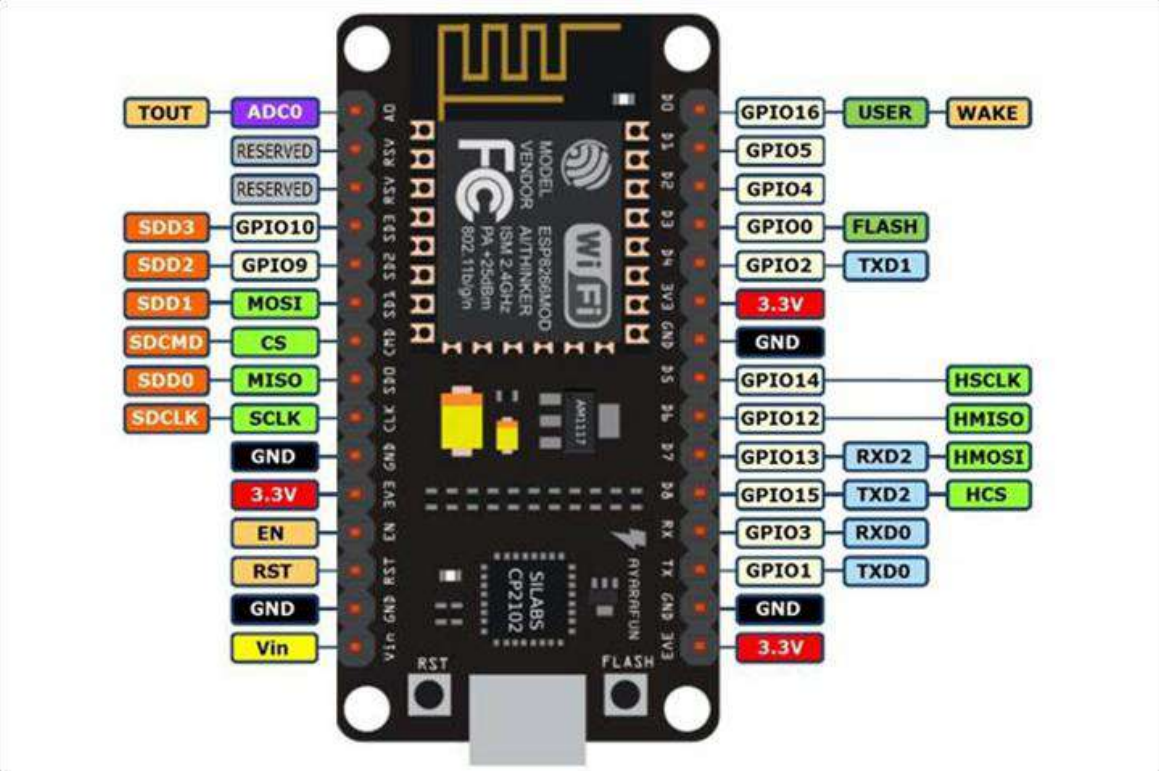


FIG I.4 : Composition d'une carte NodeMCU ESP8266.

I.6.5 Les caractéristiques du NodeMCU ESP8266 :

Ce module est base principalement sur la version ESP-12F de l'ESP8266 avec toute fois des spécificités qui lui sont propre [6].

L'interface du module est principalement devisée en deux parties, complément le micro logiciel et le matériel, ou l'exécution du frimaire s'effectue sur le Soc ESP8266.

Wifi et le matériel est sur la base du module ESP-12F.

- Le micro logiciel est base sur LUA, un langage de script facile à appende, offrant un environnement de programmation simple et un langage de script rapide qui rassemble une large comminutif de développer.
- Le frimaire étant en open source, offre la possibilité d'éditer, de modifier et de reconstruire le module existent et de modifier constamment l'interface jusqu'à ce que vous puissiez optimiser le module en fonction des besoins, du programmeur

- Le convertisseur de l'USB en UART est ajouté au module facilitât ainsi la conversion des données USB en données UART qui suivent principalement un langage de communication en série.
- Au lieu du port USB habitua, un port Micro USB est inclus dans le module pour le connecter à l'ordinateur et remplissant un double objectif. Programmation et mise sous tension de la carte.
- La carte intégré un voyant LED qui clignote et sentaient instantanément, indiquant d'état actuel du module s'il fonctionne correctement lorsqu'il est connecté à l'ordinateur. (Vous devrez peut-être installer certains pilots sur votre ordinateur s'il ne parvient pas à détecter la carte NodeMCU).
- La capacité du module à établir un parfait connexion Wifi entre deux canaux en fait un choix idéal pour l'intégrer à d'autres périphériques embarqués tells que le Rosebery Pi.

I.6.6 Mise sous tension du module:

La FIG I.5 montre, qu'il y a cinq broches de masse.GND et trois broches 3.3v sur la carte. Elle peut être mise sous tension de trois manière [7] .

- **Alimentation USB**: Il s'avère être un choix idéal pour charger des programmes, sauf si le projet à réaliser nécessite une interface séparée, c'est-à-dire déconnectée de l'ordinateur.
- **Sur pin 3,3V:** C'est une autre excellente option pour alimenter le module. Si vous avez votre propre régulateur externe, vous pouvez générer une source d'alimentation instantanée pour votre kit de développement.
- **Sur vin:** IL s'agit d'un régulateur de tension piquant supporter jusqu'à 800 mA. IL peut gérer entre 7 et 12 V. Vous ne pouvez pas alimenter les appareils fonctionnant à 3, 3 V car ce régulateur ne peut générer aussi bas que 3, 3 V et il ne faut pas dépasser les 12 V.

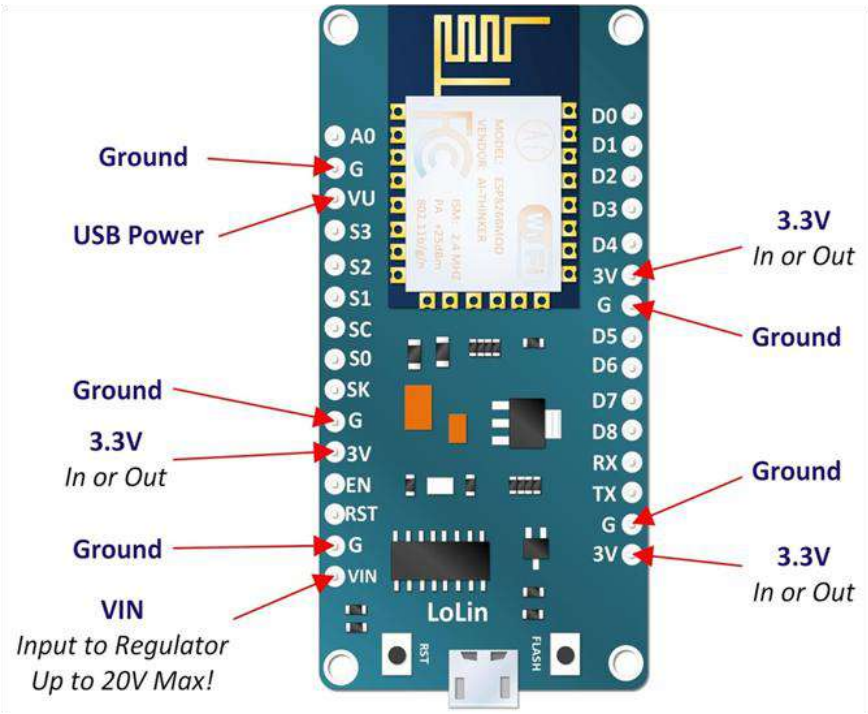


FIG I.5 : Les pins d'alimentation électrique du NodeMCU ESP8266

I.6.7 La mémoire:

Les ESP8266 embarquent une plus ou moins grosse mémoire flash accessible en SPI. Cette mémoire peut être intégré au processeur ou alors associée sur la carte NodeMCU comme mémoire flash externe. la mémoire flash est structure de la manière suivante [8].

•Un espace de stockage pour le frimaire.

•Un espace de stockage temporaire pour les mises à jour OTA (Over the Air) du frimaire.

•Un système de fichier SPIFFS.

•Un emplacement EEPROM pour la sauvegarde de données par les programmes.

•Un emplacement pour stocker la configuration du WIFI dans le cas de l'utilisation du SDK natif.

Dans la carte NodeMCU v3 il y a 4MO de mémoire, dont 3 peuvent être dédiés au système de fichier. Ce système de fichier peut être utilisé pour y stocker des données et des fichiers pour un serveur web par exemple. Cependant, il n'y voit pas l'équivalent d'un file system moderne Il n'y a pas de correction d'erreur et il n'y a pas d'arborescence de fichier (répertoires et sous répertoires), tout est au même niveau. Mais comme le caractère " est accepté dans un nom de fichier, vous pouvez stocker un fichier du nom de "/web/index.htm" si vous voulez avoir quelque chose de

structuré. Attention cependant les noms de fichiers sont limités à 32 caractères, y compris le '0' de fin de chaine (donc 31 caractères utiles). La mémoire EEPROM est particulièrement intéressante car c'est dans cette dernière que pourront être sauvegardées les données persistantes pour nos programmes. Par exemple, si une variable de notre programme sert à mémoriser un mot de passe et que ce mot de passe peut être changé, s'il est stocké dans la mémoire EEPROM nous pourrons retrouver ce changement en cas de reboot ou de coupure électrique.

I.6.8 Programmation:

I.6.8.1 Programmation NODE MCU (ESP8266*) :*

Le NodeMCU peut être programme avec l'Arduino ide, l faut installer l'ide et le driver windows pour que l'USB arrive à communiquer avec la carte, puis installer dans l'ide les modules et librairies permettant de compiler pour L'ESP8266.

Lorsque le programme est compilé, il faut l'injecter dans le microcontrôleur. On appelé cela "flasher le composant" car notre programme deviendrez le frimaire de ce dernier **[8].**

Normalement, la carte NODEMCU se met automatique en NodeMCU apprentissage, quand toutefois le <<flshage>>de la carte échoue, il faut remettre la carte dans le mode apprentissage en utilisant les boutons situés de part et d'autre du porte micro USB comme indiqué ci – dessous :

1. Appuyez sur le bouton Flash et maintenez le bouton appuyé.
2. Appuyez sur Reset.
3. Relâchez le bouton Reset.
4. Relâchez le bouton Flash.

I.6.9.1 Téléchargement de votre premier programme :

Une fois Arduino IDE installé sur l'ordinateur, connectez la carte à l'ordinateur à l'aide du câble USB. Ouvrez maintenant l'IDE Arduino et choisissez la bonne carte en sélectionnant Outils> Cartes> NodeMCU1.0 (Module ESP-12E), et choisissez le bon port en sélectionnant Outils> Port. Pour démarrer avec la carte NodeMCU et faire clignoter le voyant intégré, chargez l'exemple de code en sélectionnant Fichiers> Exemples> Bases> Clignotement. Une fois que l'exemple de code est chargé dans votre IDE, cliquez sur le bouton "Télécharger" situé dans la

barre supérieure. Une fois le téléchargement terminé, vous devriez voir le voyant intégré de la carte clignoter**.**

I.6.9.2 Application de NodeMCU (ESP8266) :

- Prototypage d'appareils Iota.
- Applications de batterie basse consommation Prototypage d'appareils Iota
- Projets de réseau.
- Les projets nécessitent plusieurs interfaces d'E/S avec les fonctionnalités Wifi et Bluetooth.

I.6.10 Conclusion sur le NodeMCU ESP8266:

Touts ces connaissances collectées à droite et à gauche nous ont permises de comprendre les bases du fonctionnement d'une carte NodeMCU ESP8266. Ces bases seront utiles pour comprendre les concepts que nous devront manipuler dans notre solution informatique.

I.7 Partie Software:

I.7.1 Arduino IDE :

I.7.1.1 L'environnement de la programmation:

Le logiciel de programmation de la carte Arduino sert à éditeur le code (langage proche du C). Une fois, le programme édité 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 les informations ce programme appelé IDE Arduino [10].

I.7.1.2 Structure général du programme (IDE Arduino) :

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 de la figure (I.6).

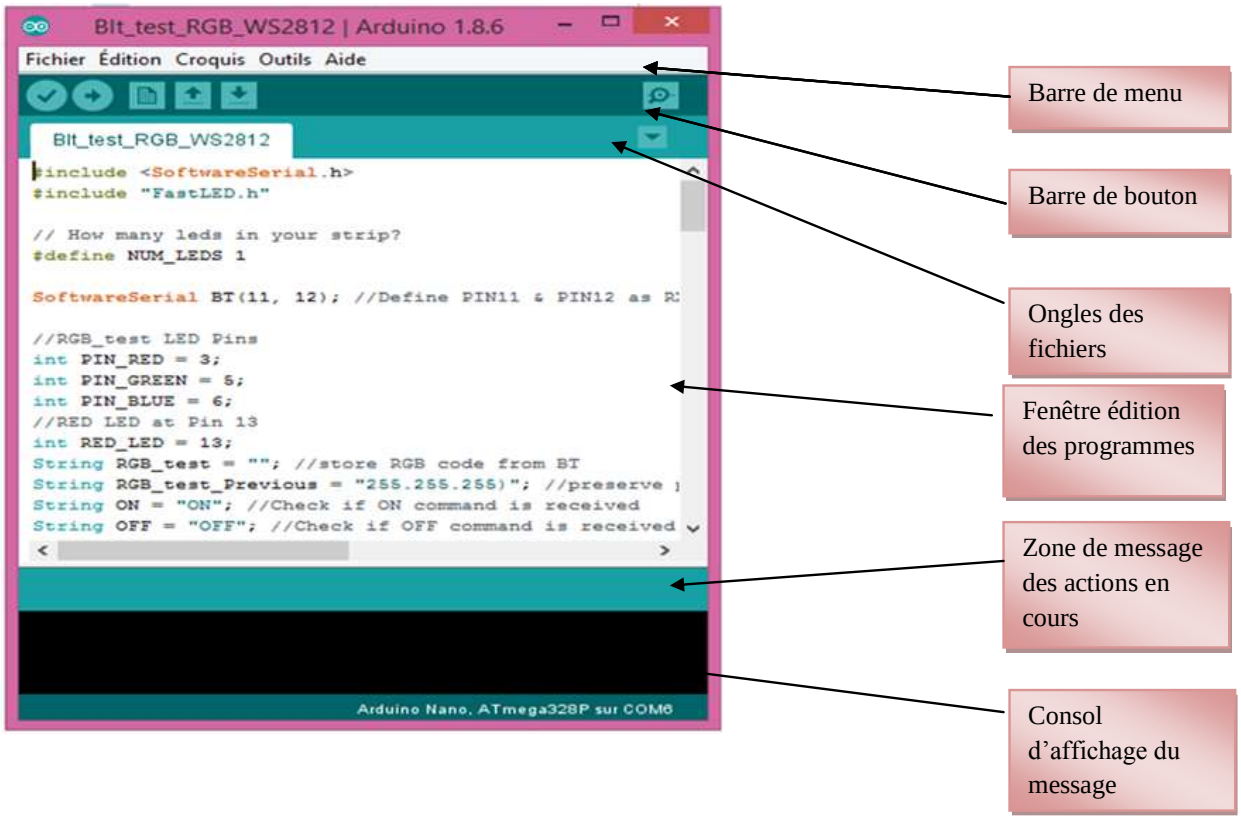


FIG I.6: Interface IDE Arduino

I.7.1.3 les étapes de téléchargement du programme :

Une simple manipulation enchainé doit être suive afin d'injecter un code vers la carte Arduino via le porte USB (FIG I.7):

- 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.

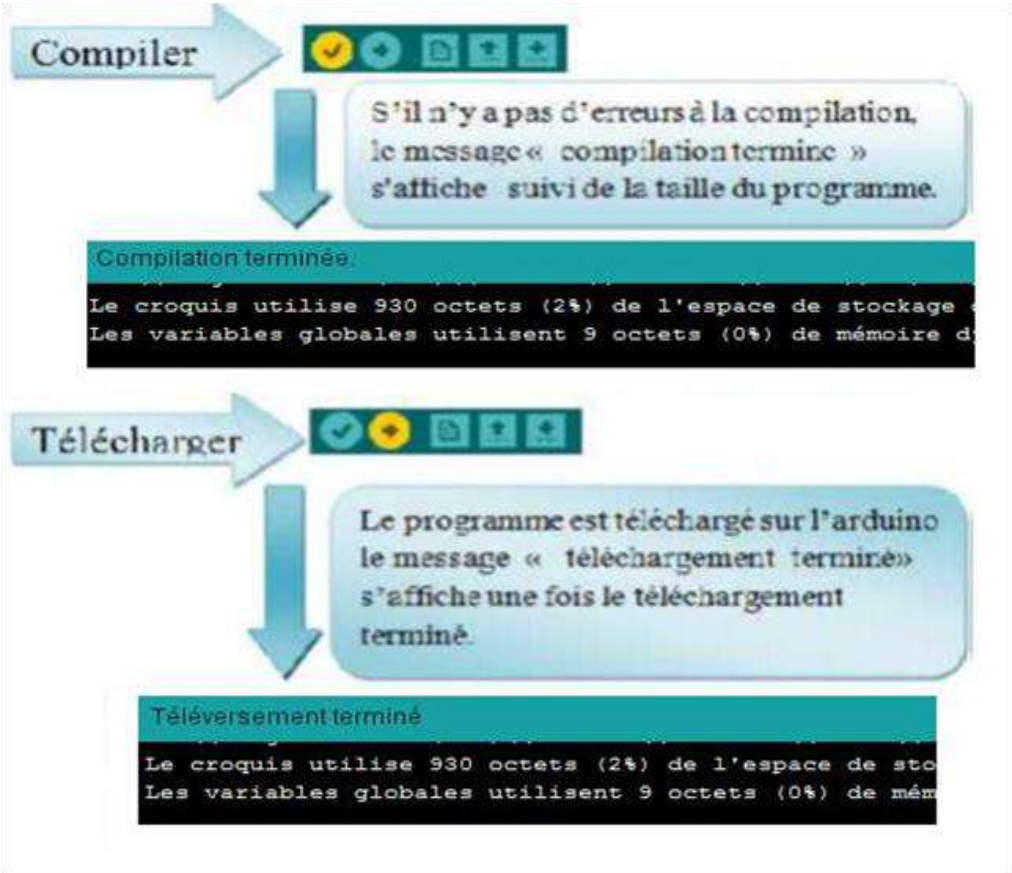


FIG I.7: Les étapes de téléchargement du code

I.7.1.4 Description du programme:

Un programme Arduino est un suit d'instruction élémentaire sous forme textuelle (ligne par ligne). La carte lies puis effectuée l'instruction les une après les autres dans l'ordre définit par les lignes de codes.

I.7.1.5 Injection du programme:

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte NodeMCU (ESP8266) et le numéro de port USB (COM) comme, à titre d'exemple la figur**e**(I.8)**.**

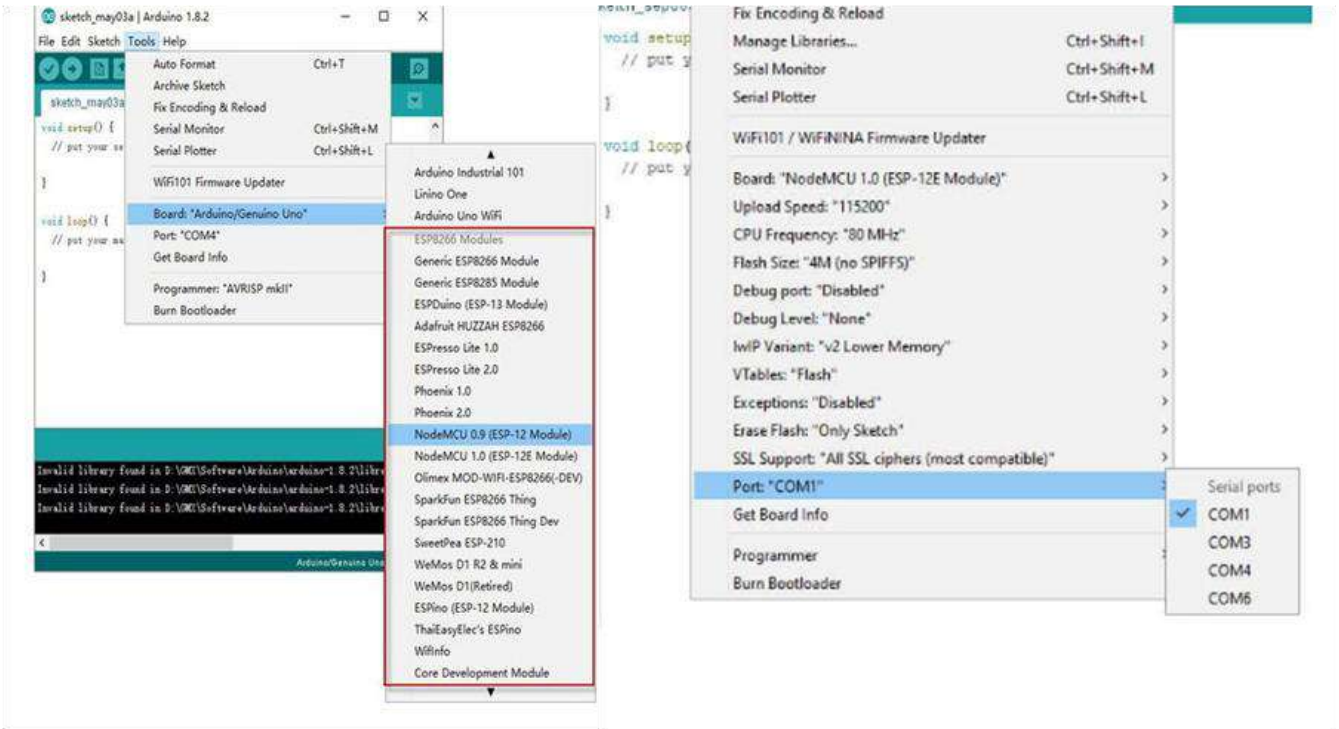


FIG I.8: Paramétrage de la carte

I.8 Conclusion:

Ce Chapitre est une introduction aux outils de notre projet pour réaliser une montre numérique intelligent en ligne avec les systèmes listés et plus prédiquâmes NodeMCU qui sera utilisé plus tard dans l'étude de notre système. Dans ce chapitre nous avons également parlé du fonctionnement de la carte NodeMCU et aussi (la partie hardware et la partie programmation) et nous avons également expliqué le principe de travail de la carte NodeMCU, une petite description des systèmes embarque et leurs principe de fonctionnement est présenté aussi dans ce premier chapitrer. Tous ces éléments sont nécessaires pour faire fonctionne la montre numérique intelligent.

Chapitre II :

Etude et simulation de l’horloge numérique intelligente

II.1 Introduction :

Dans ce chapitre, nous présenterons une simulation d'une horloge numérique intelligente. Après avoir fourni dans le chapitre précédent une description théorique et générale de l'architecture de NodeMCU (ESP8266), dans ce chapitre nous concentrerons sur l'étude la simulation et la réalisation de l'horloge numérique intelligente. Pour y parvenir, plusieurs blocs ont été nécessaires pour réaliser une telle commination avec le module NodeMCU (ESP8266) et le module d'environnement de développement IDE.

II.2 Principe de fonctionnement :

Dans notre projet, nous utiliserons l'horloge en temps réel RTC (Real Time Clock) et la carte NodeMCU (ESP8266) et un capteur d'humidité DHT11. L'avantage de cette montre est qu'elle garde la date en permanence pendant très longtemps et affiche la température et l'heure exactes même si l'alimentation de la NodeMCU (ESP8266) est coupée. IL vous suffit de planifier la configuration avant d'utiliser l'horloge RTC pour obtenir la date et l'heure correctes. Après avoir programmé le module RTC avec NodeMCU (ESP8266), nous obtenons une horloge qui s'affiche avec un écran.

II.3 Déroulement du projet :

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

- La première partie est de faire le schéma de brochage électronique.
- La deuxième partie est la simulation avec Fritzing de l'horloge.
- La troisième partie est la programmation de NodeMCU sous l'IDE.

Pour la partie programmation, nous passons à:

- Programmation de la carte NodeMCU (ESP8266).
- Perceptions des résultats.

II.4 Le module RTC :

II.4.1 Définition de l'RTC :

L'RTC (Real Time Clock). Les sont simplement des positives de mémorisation de la date et de l'heure qui ont un réglage de batterie qui maintient l'unité en marche en de 'absence l'alimentation externe. Cela actualise TIME ET DATE [10].



FIG II.1: Le module RTC.

II.4.2 Principe de fonctionnement du module RTC3231 :

Le RTC maintient son horloge en comptant les cycles de l'oscillateur - généralement un circuit oscillateur à cristal externe de 32,768kHz, un condensateur basé sur l'oscillation interne ou même un cristal de quartz intégré. Certains RTC conservent la configuration de l'oscillateur au dernier point connu avant qu'il ne se déverrouille avec l'entrée [11].

II.4.3 Puce DS3231 RTC ou gros plan de la puce :

Au cœur de l'unité se trouve une puce RTC à faible coût et très précise. Il gère toutes les fonctions de chronométrage et dispose d'une interface I2C simple à deux fils qui peut être facilement interfacée avec n'importe quel microcontrôleur de votre choix.

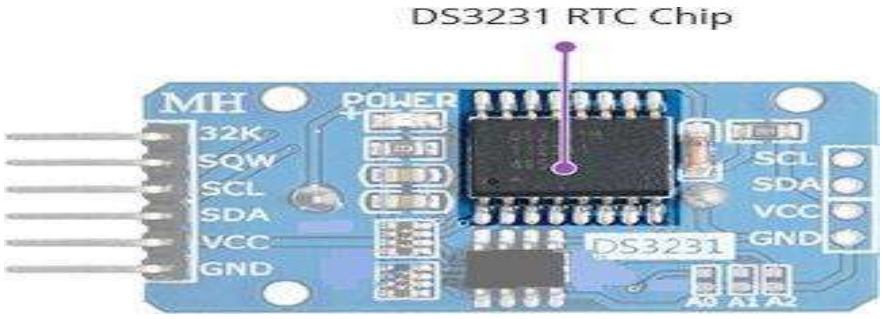


FIG II.2 Le module RTC ds3231

La puce conserve des informations sur les secondes, les minutes, les heures, le jour, la date, le mois et l'année. La date de fin de mois est ajustée automatiquement pour plusieurs mois à moins de 31 jours, y compris les corrections des années bissextiles (valable jusqu'en 2100).

L'horloge fonctionne au format 24 heures ou 12 heures avec un indicateur AM/PM.

Une autre fonctionnalité intéressante de cette carte est fournie avec la broche SQW, qui produit une onde carrée à 1Hz, 4Hz, 8Hz ou 32 Hz et peut être traitée par programmation.

Pour la batterie de secours Le DS3231 comprend un siège d'insertion de batterie et maintient un chronométrage précis lorsque l'alimentation principale de l'appareil est coupée. Circuit Le circuit d'alimentation est mis hors tension en continu et s'éteint de bas en bas, pas besoin de s'inquiéter de la mise hors tension, de la rotation du suivi du temps MCU.

DS3231-RTC-Module-CR2032- Support de batterie.



FIG II.3 DS3231-RTC-Module-CR2032-support de batterie.

II.4.4 Les caractéristiques d'un module RTC3231 :

Le module DS3231 RTC possède 6 broches qui le connectent au monde extérieur. Les communications sont les suivantes :

- Broche **32K** avec sortie stable (compensation de température) et indication des minutes et des heures.
- **SQW** produit une onde carrée à 1Hz, 4kHz, 8kHz ou 32kHz et peut être traitée par programmation.
- **SCL** est la broche d'horloge série de l'interface I2C.
- **SDA** est la broche de données série de l'interface I2C.
- **VCC** est la broche d'alimentation de l'unité. Il peut être de l'ordre de 3,3V à 5,5V.

- **GND** est la broche de masse.

II.4.5 Brochage le module RTC DS3231 avec NodeMCU (ESP8266) :

Pour des connexions assez simples, commencez par connecter la broche VCC à la sortie 3,3 V du NodeMCU (ESP8266) et connectez GND à la terre. Puis on continu avec les parties qui sont utilisées pour la communication. Notez que chaque carte NodeMCU (ESP8266) a des broches différentes qui doivent être connectées en conséquence. Sur les cartes de Nodemcu, le SDA (ligne de données) et le SCL (ligne d'horloge) sont situés sur les têtes de broche à proximité de la broche AREF. Il est également appelé d1 (SCL) et d2 (SDA).

II.5 Capteur humidité DHT11 :

Dans notre projet, nous avons utilisé le capteur DHT11, capable de mesurer des températures de 0 à + 50° C avec une précision de +/- 2° C et des niveaux d'humidité relative de 20 à 80% avec une précision de +/- 5 %. Chaque seconde une mesure peut être effectuée. Ce capteur est une alimentation avec 3.3V et 5V. Ce capteur n'pas cher mais il fonctionne de la même manière qu'un DHT22 ou tout autre capteur similaire (comme DHT21). La principale différence entre ces modèles est la précision, le DHT22 donné des résultats plus précises et dans une plage plus large. Le DHT11 semble être environ 5 degrés plus froid que la température réelle [12].

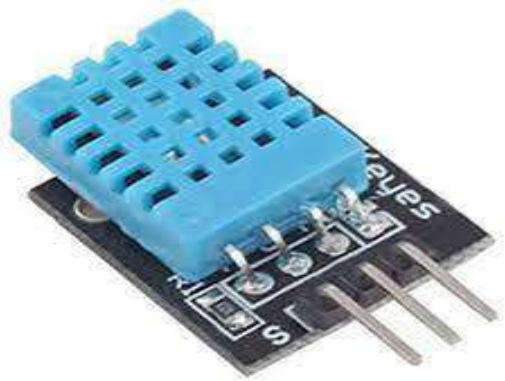


FIG II.4 : Capteur DHT11.

II.5.1 les caractéristiques du DHT11 :

• Faible coût.

• Alimentation 3,3v à 5v.

• Consommation de courant maximum de 2,5 mA lors de la commutation (lorsque les données sont demandées).

• Bon pour des lectures d'humidité de 20 à 80% avec une précision de 5%.

• Bon pour une température de lecture de 0 à 50 ° C Précision ± 2 ° C.

• Pas plus de 1 Hz pour un échantillonnage répété (1 fois par seconde).

• Taille du corps 15,5 mm x 12 mm x 5,5 mm.

• 4 broches avec un espacement de 0, 1 mm.

II.5.2 Brochage d'un capteur d'humidité DHT11 avec NodeMCU (ESP8266) :

Connexions sont suffisantes pour contrôler le module dht11 :

- **VCC** du capteur DHT11: sortie **3.3 V** de l'ESP8266.
- **GND** du capteur DHT11: broche **GND** de l'ESP8266.
- **DATA** du capteur DHT11: broche d3(Gpio0) de l'ESP8266.

II.6 Led matrix Md7219 :

En utilisant un Md 7219, on peut piloter 64 LEDs en utilisant seulement 3fils pour les interfacer avec un microcontrôleur (ceci exclut VCC et GND qui sont supposés être disponibles). De plus, on peut connecter en série plusieurs puces Md 7219 pour des écrans plus grands.
En de16 lignes de sortie entraînant 64 LEDs individuelles. Cela semble le Md7219 est composé complique, mais la méthode de conduite utilise la façon dont nos yeux fonctionnent. La persistance de la vision est exploitée pour donner l'impression que les LEDs sont allumées tout le temps alors qu'en fait elles ne le sont pas. En fait, les LEDs sont disposées comme un ensemble 8x8 de lignes et de colonnes. Chaque colonne est pulsée pendant un court instant tandis que les bits de ligne de cette colonne sont pilotés. Nos yeux se souviennent d'un flash de lumière pendant environ 20 ms, donc lorsque vous faites clignoter en continu une lumière (ou une LED) à une vitesse égale ou supérieure à 20 ms, il semble que la lumière ne s'éteigne jamais. C'est ainsi que fonctionne le Md7219. Toutes les LEDs sont allumées individuellement pendant une courte période, à une vitesse supérieure à 20 ms.

II.6.1 Les type des Led matrix :

Il existe tellement de types de matrices LEDs qui ont été utilisées à des fins industrielles et commerciales, en ce qui concerne la structure, on trouve :

- Matrice 8*8 points.
- Matrice 5*7 points.
- 128*16 deux doublés.

- 128*32 quatre lignes.
- 128*64 Huit doublé.

II.6.2 Conception de la matrice de points:

La construction de l'écran est très simple, les diodes sont simplement soudées sur une carte Véro ou une carte double ou simple face préfabriquée. Supposons que nous voulions construire un prototype d'affichage à matrice de points 8*8, alors nous avons besoin de 64 diodes électroluminescentes. Ces diodes sont soudées sur une carte à la mer ou sur une seule face. L'extrémité positive d'un diois est connectée à l'autre extrémité positive de la diode pour faire les rangées, de même les 8 rangées sont faites en connectant l'extrémité positive de chaque diode. L'extrémité négative d'une diode est connectée à l'extrémité négative d'une autre diode. Pour réaliser les colonnes de cet écran, de même les 8 colonnes sont réalisées en connectant l'extrémité négative de chaque diode. Toutes ces diodes sont connectées les unes aux autres par un fil en cuivre nu et tous doivent être correctement vendues, sinon il n'y aurait pas de connexion entre toutes.

II.6.3 Fonctionnement de la matrice des LEDs 8×8 :

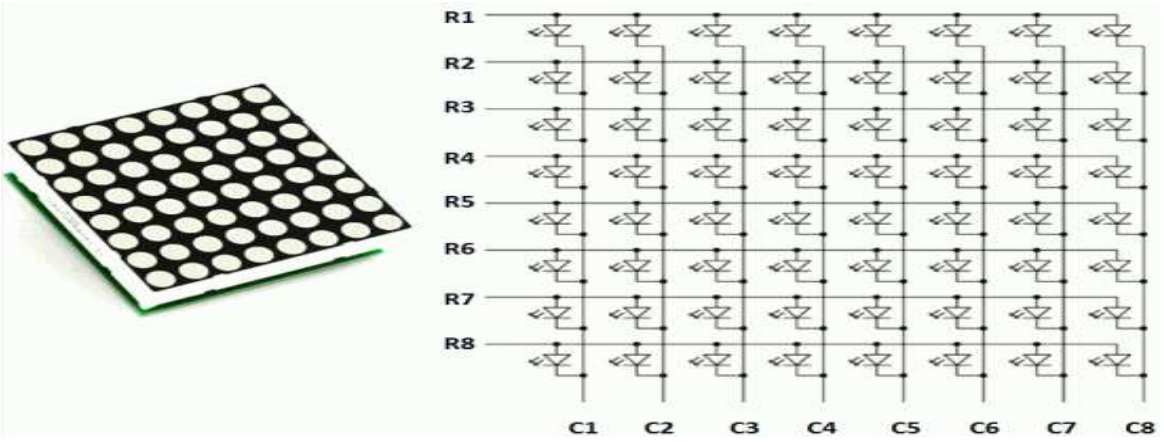


FIG II.5 : Connexion des LEDs dans la matrice.

La matrice des LEDs est constituée de 8 lignes et 8 colonnes. On peut assimiler la matrice à un tableau 1D de 8 éléments, chaque élément du tableau est codé sur 8 bits Ex B00001111 ou 0x0F, dans la figure II.5 illustre le passage d'une matrice 8×8 à un tableau sur 8 éléments de type Byte

ou Insignes char. On verra dans la suite la fonction qui permet d'afficher le code dans la matrice 8×8 [13].

Le module utilisé est basé sur MAX7219 avec une liaison série pour les données. Le bus SPI est constitué de trois fils :

- **CS** : Chip Select pour l'activation du modèle.
- **CLK** : l'horloge synchrone pour les données. L'horloge définie la vitesse de transmission des données série.
- **DIN** : La donnée série sur 8 bits.

II.6.4 Le circuit intégré Max7219 (ou Max7221):

Les circuits intégrés Max7219 et Max7221 sont pratiquement interchangeable. Ils sont spécialement conçus pour allumer et éteindre un grand nombre de LEDs en maintenant dans celles-ci un courant constant. Dans le schéma de la FIG II.6 les pins, "Digs" et "Seg" font référence aux digits et aux segments d'un afficheur à 7segments [14].

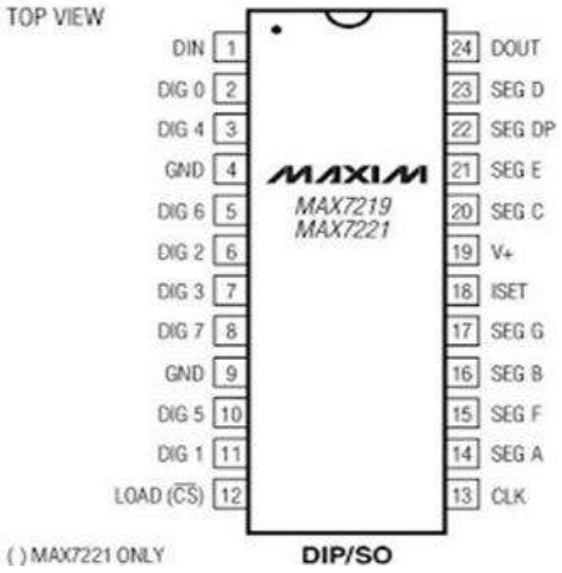


FIG II.6 : Circuit intégré Max 7219.

II.6.5 Connexions du module à ou ESP8266 :

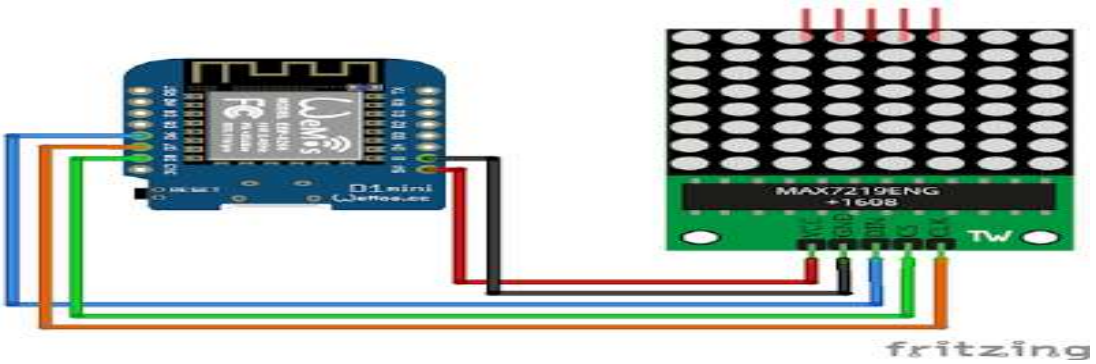


FIG II.7 Connexions du module à l'ESP8266.

La connexion nécessaire pour contrôler le module MAX7219:

- **VCC** du module MAX7219 le relier à la sortie **3.3 V** de l'ESP8266
- **GND** du module MAX7219 à la broche **GND** de l'ESP8266
- **DIN** du module MAX7219 le relie à la broche d7 de l'ESP8266
- **CS** (LOAD) du module MAX7219 le relie à la broche d de l'ESP8266
- **CLK** du module MAX7219 le relie à la broche d5 de l'ESP8266

II.7 Schéma synoptique générale :

Le programme global de l'horloge sera injecté dans la carte NodeMCU (ESP8266) par l'ordinateur. La carte envoie à son tour une demande à l'unité RTC pour lui envoyer les données l'heure et la date et à l'unité du capteur DHT 11 pour afficher la température, une fois que la carte NodeMCU (ESP8266) a le temps, elle convertit les informations reçues de l'RTC et du capteur DHT11 de manière à ce qu'il soit affiche en décimal à l'écran. Les boutons poussoirs servent à leur tour à régler l'heure.

Le schéma synoptique général de la simulation de l'heure est indiqué par la figure (II.8):

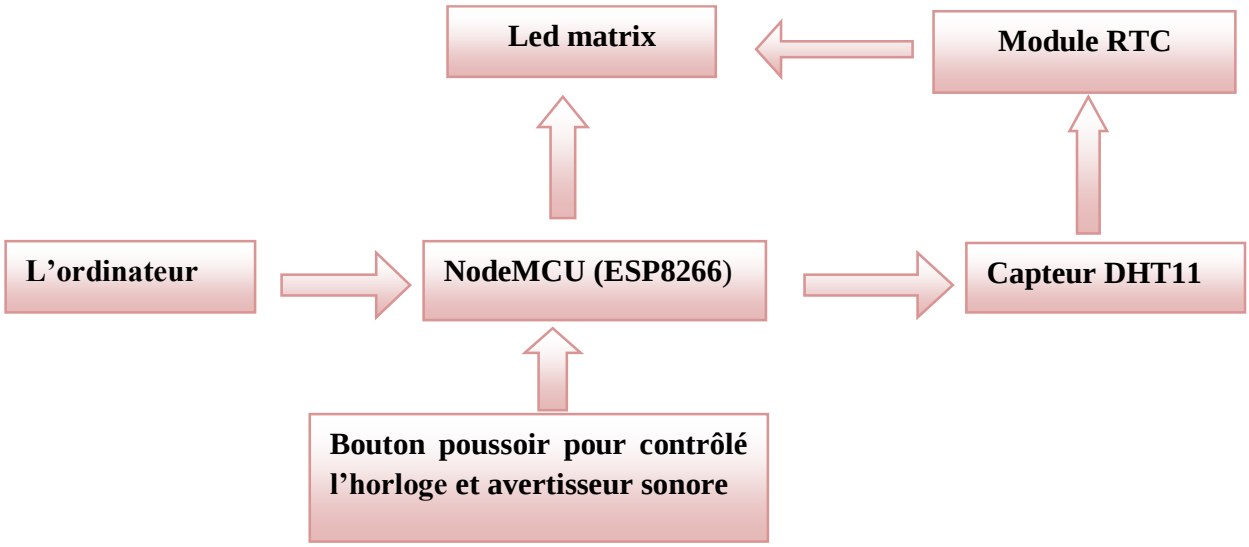


FIG II.8 Schéma synoptique l'horloge numérique intelligente.

II.8 Le logiciel Fritzing:

II.8.1 Définition de programme Fritzing :

Fritzing est un logiciel libre de conception de circuit imprimé qui va vous permet de concevoir des circuits électroniques pour des liquation avec Arduino. En plus on peut les exporter en image afin de les présenter à votre guise [15].

II.8.2 Presentation du logiciel :

II.8.2.1 L'intrface de Fritzing :

La FIG II.9 présente l'interface du logiciel Fritzing une fois ouvert.

- ✓ Il offre plusieurs menus, comme Breadboard, Schematic, PCB et Code qui permettrent de faire différents types de schémas. [15]

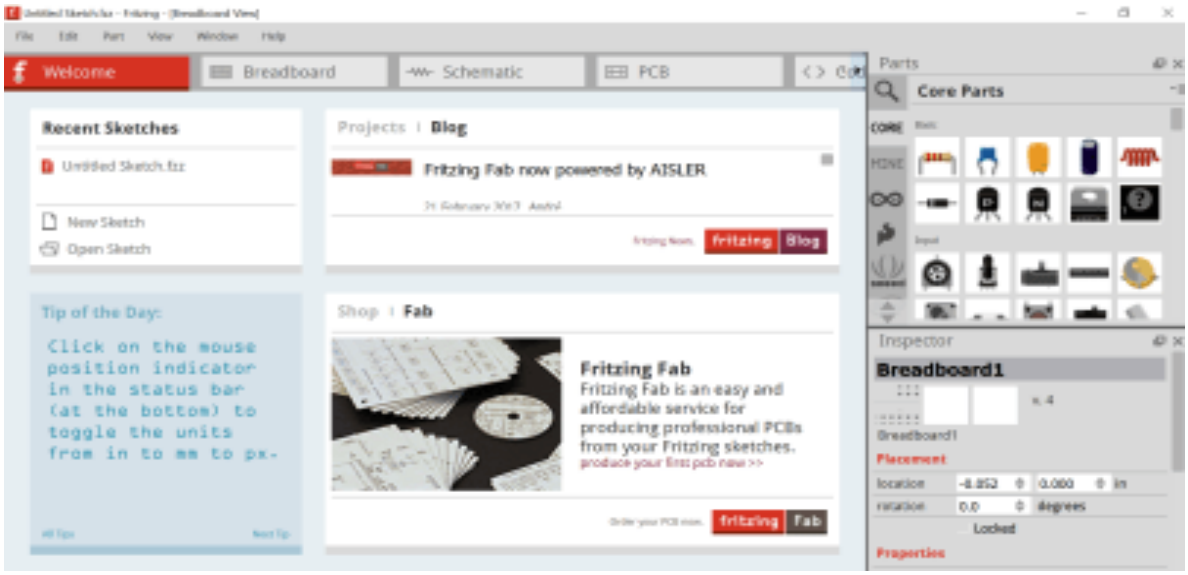


FIG II.9 La fenêtre du logiciel Fritzing**.**

II.8.2.1.1 Breadboard sur Fritzing :

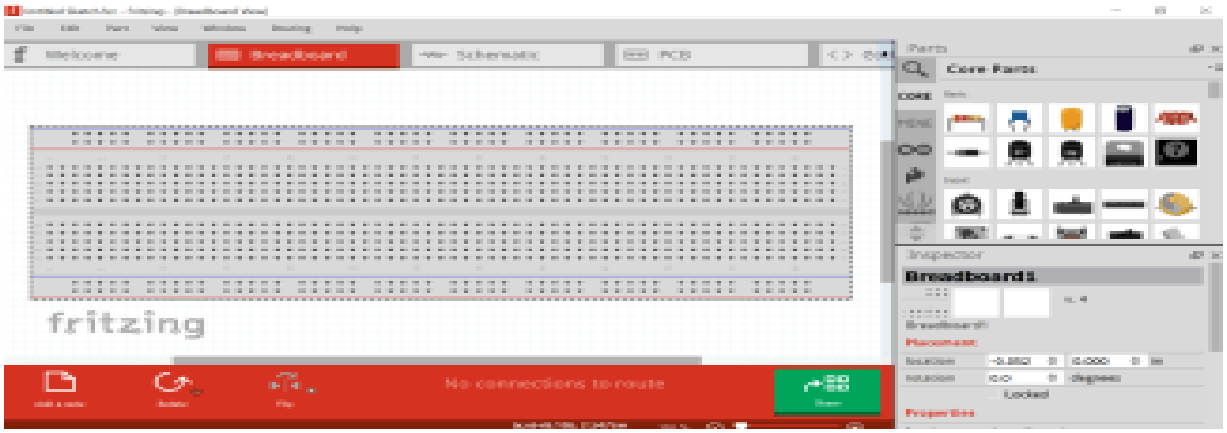


FIG II.10 La plaque d'essais (Breadboard).

La plaque Breadboard permet de consevoir des circuits.à droits on trouve les différents composants que l'on peut ajouter sur notre circuit. On pourra ensuite relier les différentes bornes des composants à la breadboard [15].

II.8.2.1.2 Schematic :

La catégorie schématic permet d'avoir une image des composants avec les différentes bornes de sortie. Ceci permet de voir quels composants sont branchés sur quels ports et d'avoir une vision plus claire du schéma électrique. Sur la FIG II.11 un exemple avec un branché sur une carte Arduino dans la catégorie Breadboard [15].

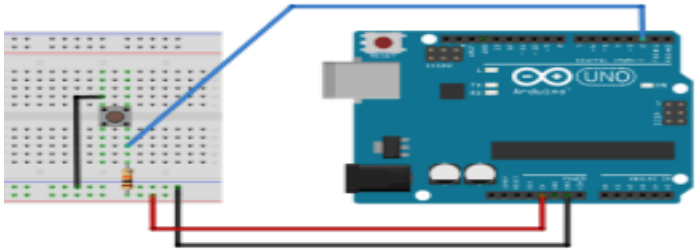


FIG II.11 Schéma de cablage une caret Arduino dans la catégorie breadboard.

II.8.2.1.3 Le circuit imprimé PCB :

La partie PCB de fritzing permet de construire une maquette après avoir concu le circuit dans breadboard. Dans le menu PCB, on peut voir notre carte



FIG II.12 : Circuit imprimé PCB.arduino avec la led qui est reliée dessus.

Il y a un bouton sur la barre d'outils en bas, view from above. Celui-ci permet de voir la maquette du dessus.

En cliquant dessus on peut avoir la vue de la maquette du dessous, view from below.

Le bouton both layer permet de cliquer soit sur les fils, soit sur les bornes de la carte arduino, sinon sur les deux à la fois.

Le bouton autoroute permet de préparer la maquette afin de l'exporter en PCB,nous n'avons pas besoins de connaitre le prix de realisation de la PCB car on va pas le faire [15].

II.8.2.1. 4 Code :

La partie code permet d'ecrire et téléverser les codes pour les circuit (projets) à réaliser. On doit choisir le port sur le quel la carte Arduino est connecté en cliquant upload [15].

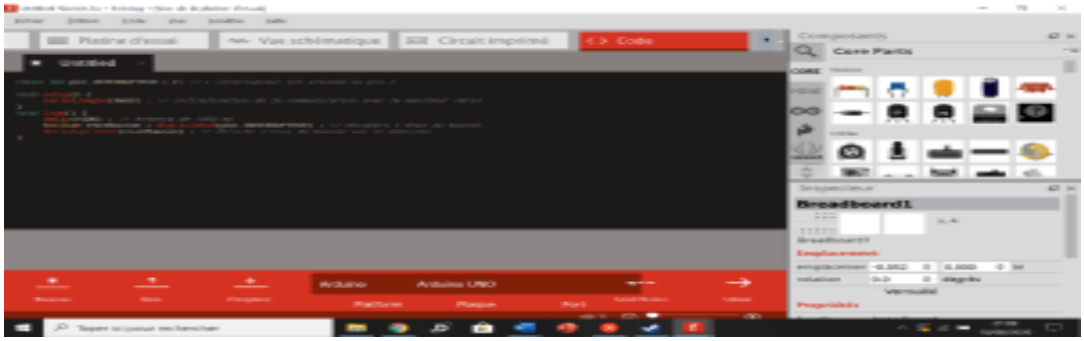


FIG II.13 : Code

II.8.2.2 Les composant sur Fritzing :

En haut à droite on peut voir les différents composants que l'on peut placer sur le schéma. On voit les différentes catégories de composants sur Fritzing. La loupe en haut permet de recherché un composant si on ne connait pas son nom. La partie core regroupe touts les composant utilisables pour concevoir un circuit tel qu'une simple résistance, ou bien un moteur, photo résistance, potentiomètre.[15]

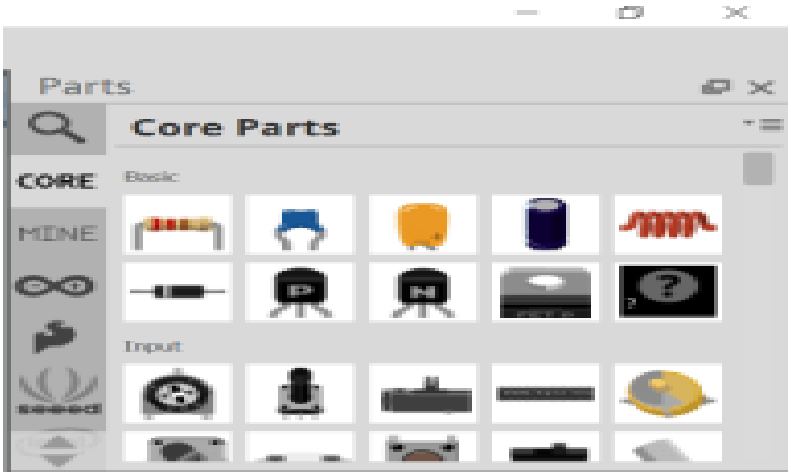
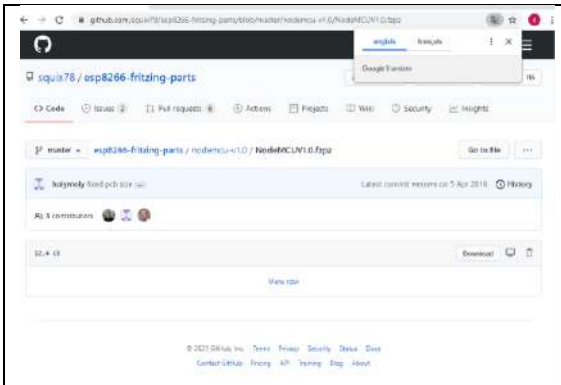
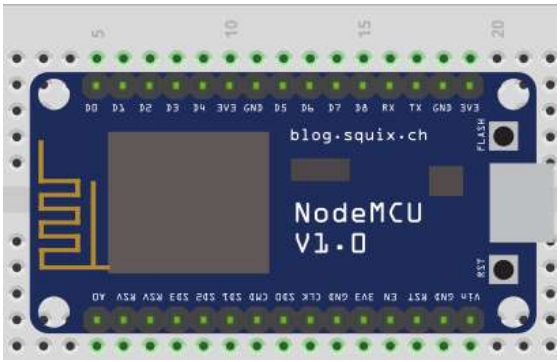
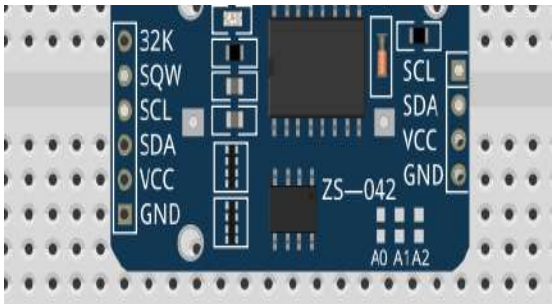
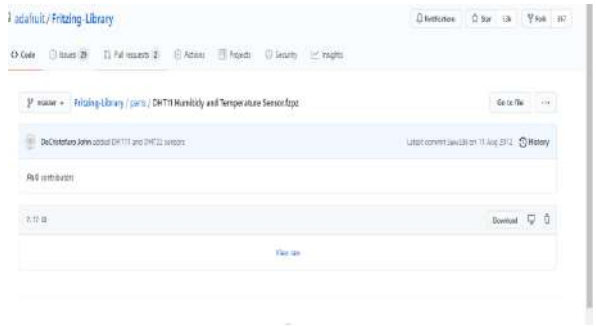
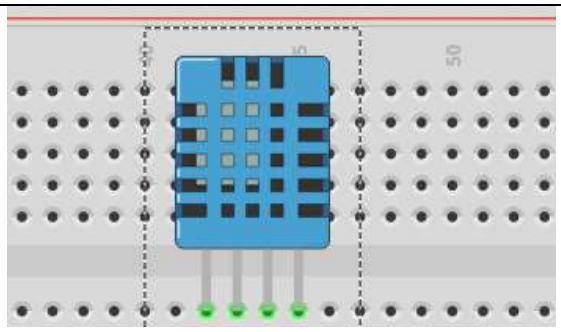


FIG II.14 : Les composants sur Fritzing.

II.9 Démarche de la simulation :

Comme les librairies des éléments utiliser dans notre projet ne sont pas des bibliothèques officielles de Fritzing, on doit les télécharger un par un, les installes et les ajouter à la bibliothèque de Fritzing comme explique par les figures II.15, II.16, II.17, II.18, II.18, II.19 et II.19.

Tableau II.1 : Les bibliothèques de tous les composants.

 Téléchargement du NodeMCU dans Fritzing	 NodeMCU dans Fritzing
 Téléchargement de NodeMCU dans Fritzing	 Module RTC3231 dans Fritzing
 Comment téléchargement du capteur DHT11 dans Fritzing	 Capteur DHT11

II.10 Circuit global de la simulation :

La figure II.13 représente le circuit global de notre simulation sous fritzing

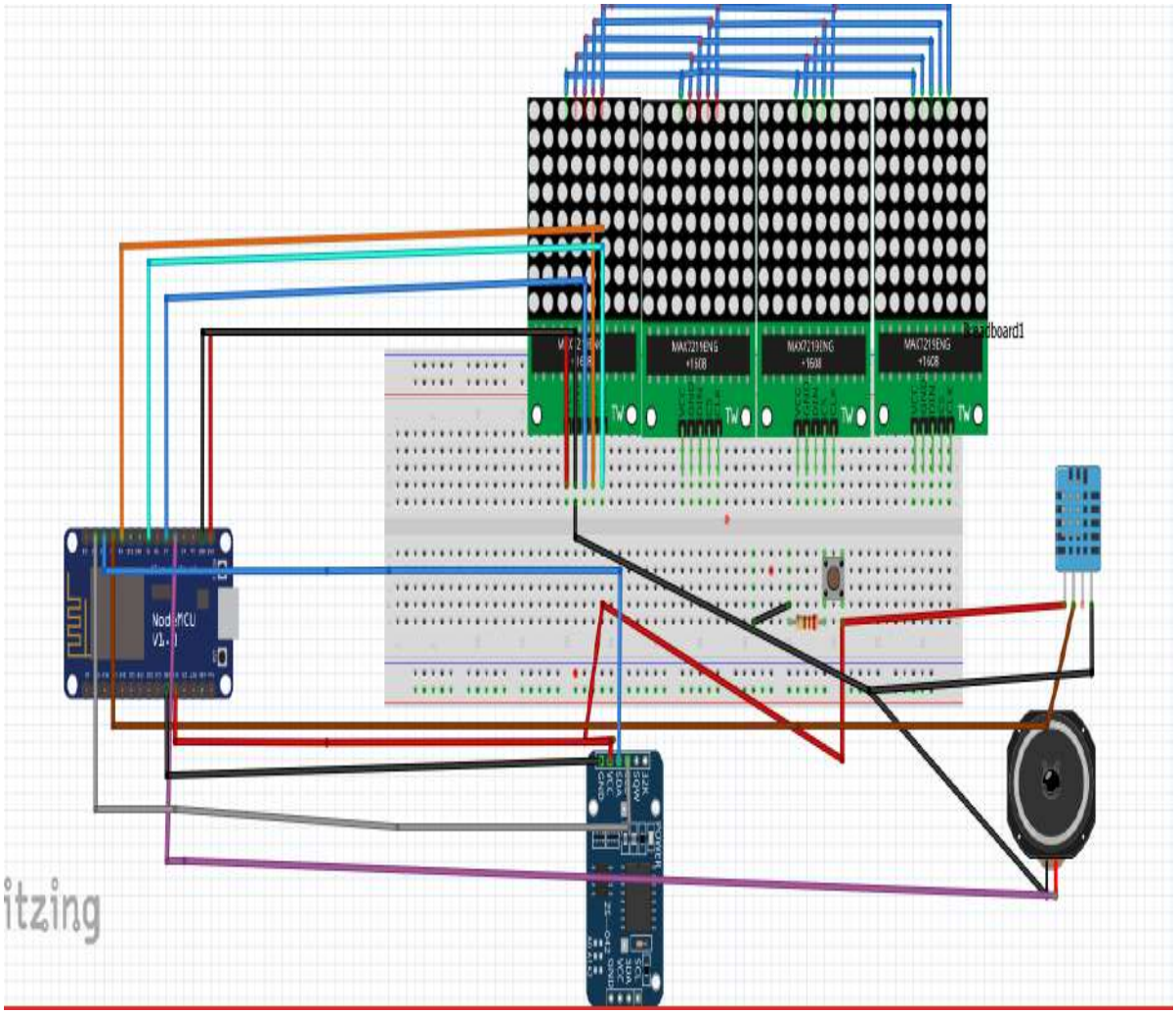


FIG II.13 Circuit globale de simulation sous fritzing

Les différentes connexions et numéro des pins de circuit sont présente donne le tableau.

Tableau II.2 : Broches et connexion du circuit électronique sur Fritzing.

Module	Port NodeMCU	Port de module	Alimentation
Led max 7219	D5(GPIO14) D4(GPIO02) D7(GPIO13)	Clk Cs DIN	3.3V Gnd
RTC (ds3231) 3231)	D2(GPIO04) D1(GPIO05)	SDA SCL	3.3V Gnd
DHT11	D3(GPIO0)	DATA	
Bouton poussoir			Gnd 3.3V
Resistance 10kΩ	D0(GPIO16)		Gnd
Avertisseur sonore	D8(GPIO15)	SP	Gnd

II.9 Conclusion :

Ce chapitre couvre les différentes étapes de la perception hypothétique nécessaires pour visualiser le résultat sous Fritzing, qui commence par ajouter les bibliothèques nécessaires pour simuler l'horloge et connaître les principaux composants utilisés. Ensuite, nous nous sommes concentrés sur le circuit général et les différentes prises reliant NodeMCU (ESP8266), le module RTC, la matrice Led md7219, les boutons poussoirs et l'écran du capteur DHT11. Finalement, nous avons fait quelques tests sur le circuit global sous Fritzing. Le prochain chapitre sera consacré à la réalisation pratique d'une horloge numérique intelligente pilotée avec une application Blynk

Chapiter III:

Conception et réalisation Pratique de l’horloge

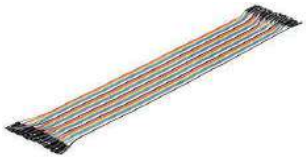
III.1 Introduction :

Après avoir expliqué dans le chapitre précédent la simulation avec le Node MCU ESP8266 via l'application Fritzing. Dans ce chapitre, nous allons présenter les différentes étapes de réalisation d'une horloge numérique intelligente à l'aide de la carte Node MCU ESP 8266. Ce travail, basé sur la carte ESP 8266, permet de générer l'heure en utilisant le RTC (Real Time Clock) ainsi que le DHT11 qui donne nous la température et l'humidité et l'application Blynk qui nous permet de contrôler la montre intelligente via le module wifi connecté à la carte ESP8266. Nous passerons par différentes étapes qui nous permettront de concevoir et de construire une horloge numérique : étude, sélection des composants nécessaires, tests sur la carte de test et, enfin, nous connectons les composants.

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

III.3 Composants utilisés :

Tableau III.1 : Composants utilisés dans la réalisation de l'horloge.

 Carte NODE MCU ESP8266.	 Un module RTC 3231 ou autre.
 Plaque d'éssai	 Des fils jumpers
 DHT11	 Une résistance
 Bouton Poussoir	 Buzzer sonore.

III.4 Schéma global du montage :

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

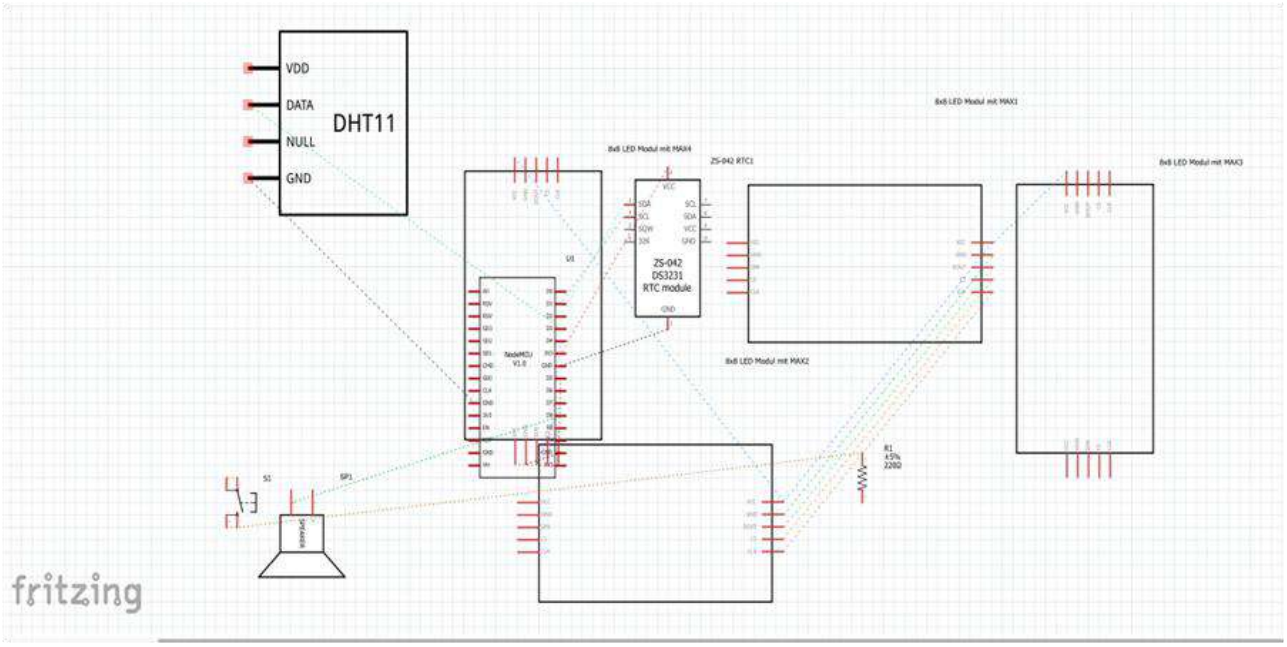


FIG III.1 Schéma global du montage électronique de l'horloge.

Le brochage des différents composants:

Tableau III.2 : Brochages des composants constituants le circuit.

Module	**Port de Node MCU**	**Port du module**	**Alimentation**
Led max 7219	D7	DIN	3.3VCC
	D4	CS	GND
	D4	CLK	
RTC	D2	SDA	VCC
	D1	SCL	GND
DHTT11	D3	DATA	VCC
			GND
BOUTON			VCC
			GND
avertisseur sonore	D8	SP	GND

III.5 Parti atelier :

Dans cette partie on représentera tous les matériaux utilisés et les étapes de la réalisation du la boite de l'horloge.

II.5.1 Assemblage de L'horloge

II.5.1.1 Conception de la boite à loger:

Pour une conception parfait et tridimensionnelle de la boite, nous avons au recoures à l'utilisation de solide Works

II.5.1.1.1 Définition générale du solide Works :

Solide Works est un logiciel de conception technique 3D. Ce logiciel fonctionne sous l'environnement Microsoft Windows et a été développe par Dassault Systèmes Solide Works Corp.
Groupe de sociétés du Groupe
Dassault Systèmes en France.
Solide-Works est actuellement utilisé par plus de deux millions d'ingénieurs et de concepteurs dans plus de 165000 entreprises à travers le monde [16].

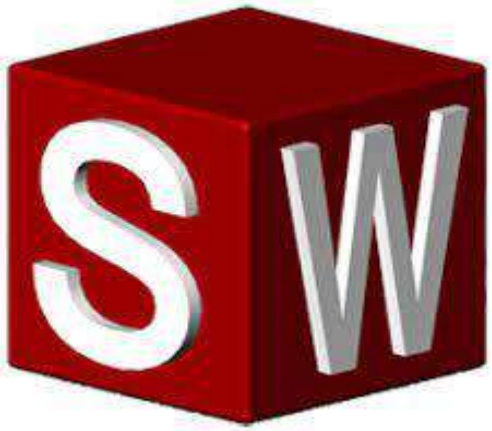


FIG III.2 : Solide Works

Notre conception sur ce programme est présentée sur la figue III.3.

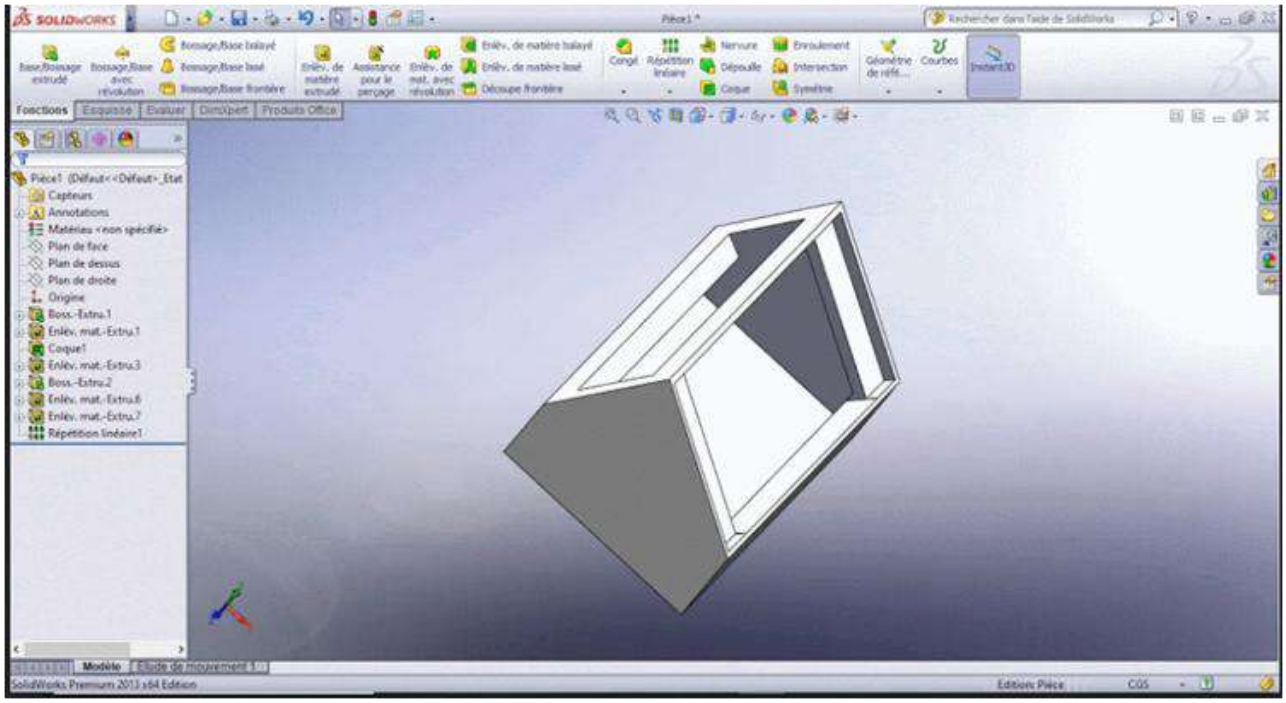


FIG III.3: Conception sous Solideworks

II.5.1.1.2 Fabrication de la boite :

Après avoir conçu la boîte sue le programme Solide Works, nous l'avons imprimé à l'aide d'une impriment 3D pour obtenir la forme illustré ci-dessous

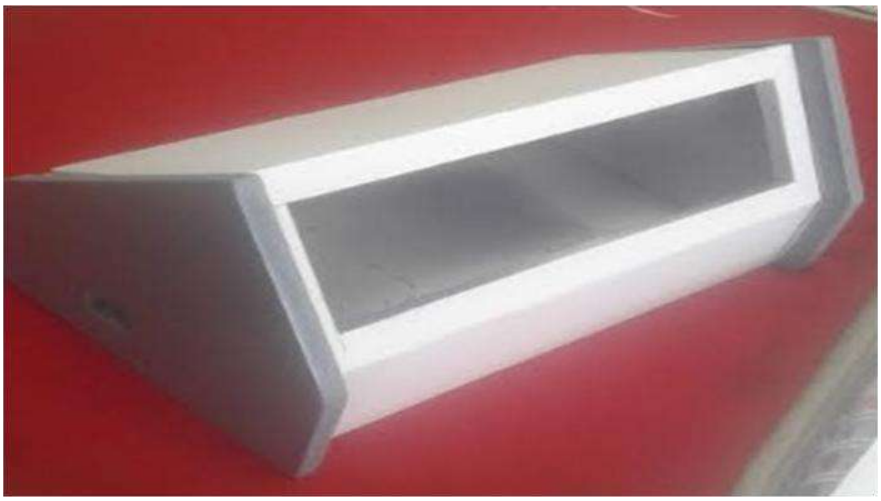


FIG III. 4: Montage finale de la boite.

Une fois la boîte imprimée, nous collectons toutes les pièces à l'intérieur pour apparaître comme suit :

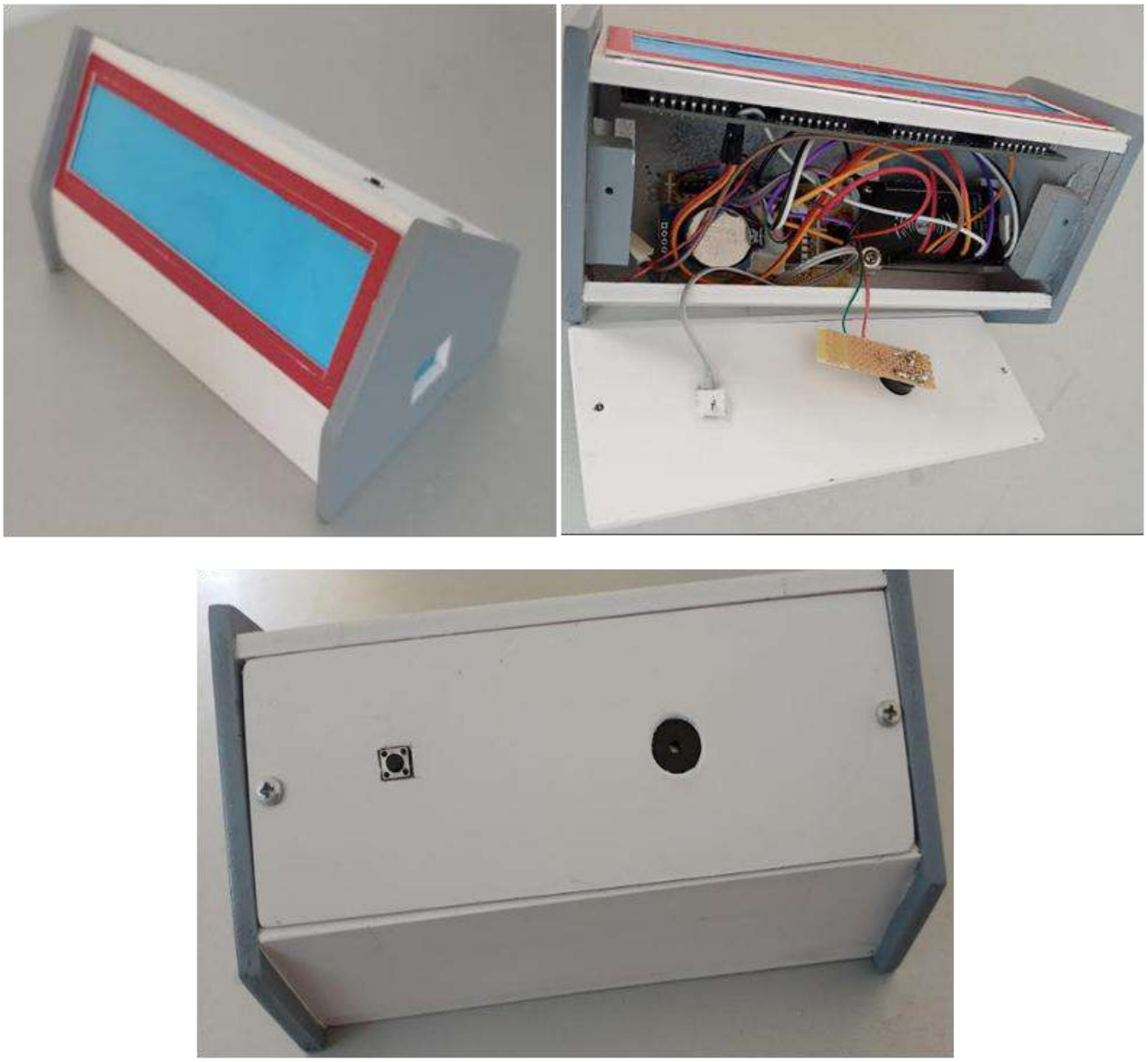


FIG III.5 : Montage finale des composants avec le boitier.

III.6 Partie programmation:

Dans cette partie on représente le code final de L'horloge.

III.6.1 Organigramme de fonction fondamental:

Le code global pour régir l'ensemble des composants constituants l'horloge est organisé comme l'indique l'organigramme de la FIG.III.6.

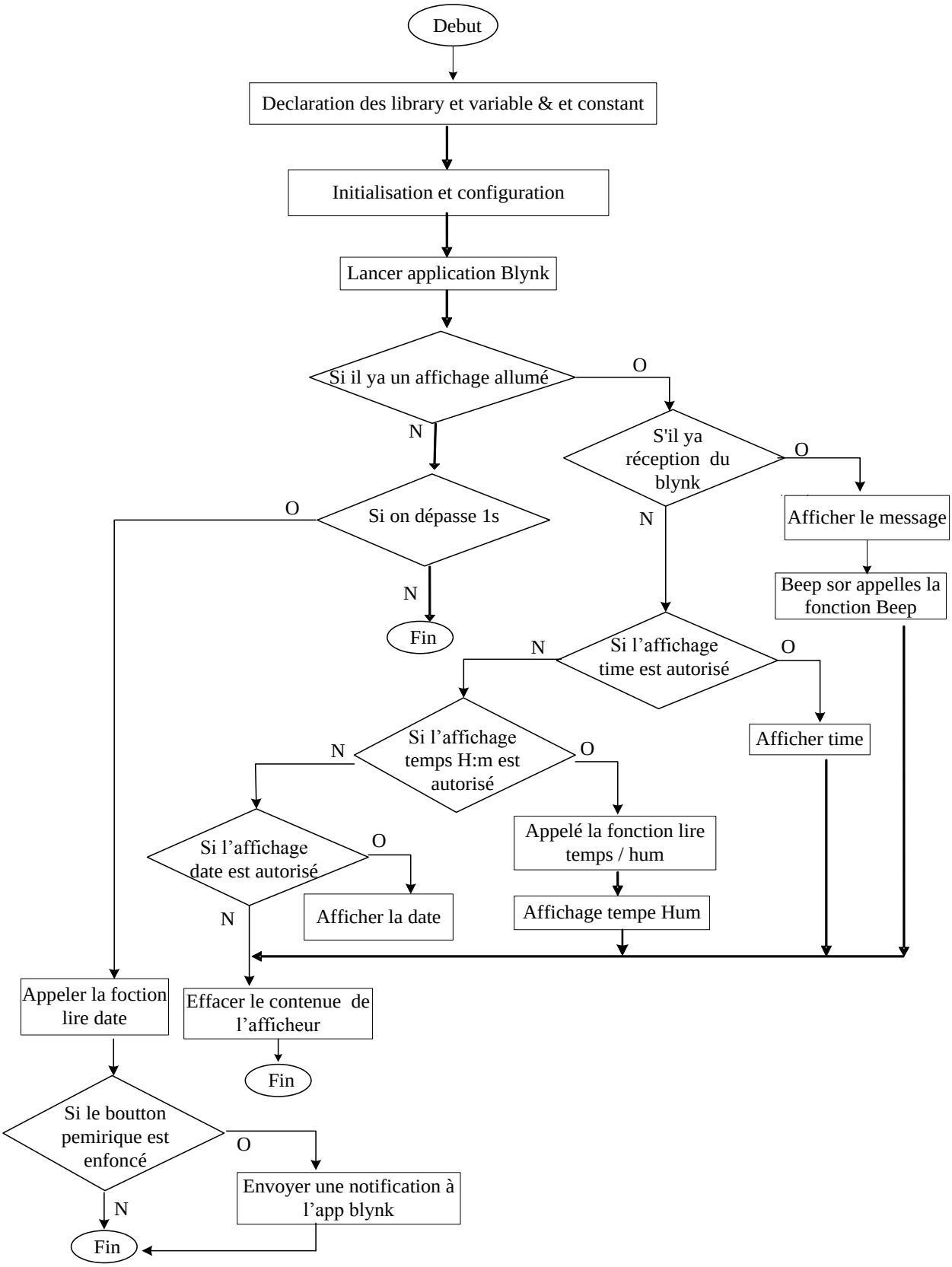


FIG III. 6 : Organigramme explicatif du code global de l'horloge

III.6.2 Organigramme de fonction secondaire :

III.6.2.1 Fonction lire temps:

La fonction permettant de lire le temps est programmé selon l'organigramme suivant :

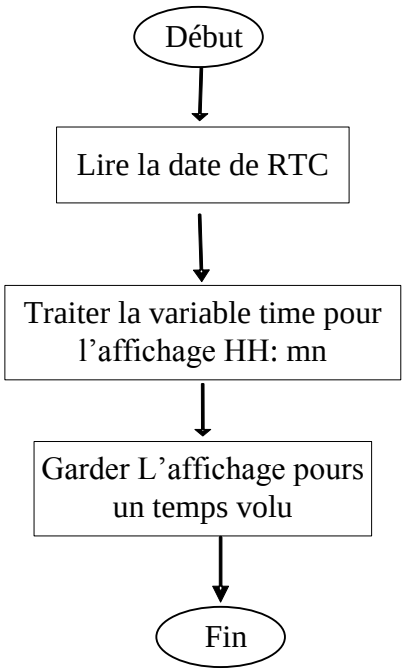


FIG.III.7 Organigramme pour lire le temps à partir de l'RTC.

III.6.2.2 Fonction lire Temps/Hum:

L'obtention de l'information à partir du capteur DHT11 se fait selon l'rganigramme suivant :

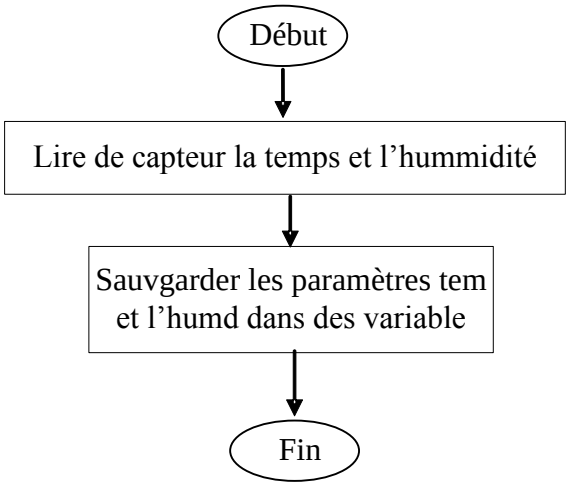


FIG.III.8 Organigramme pour lire la température et l'humidité à partir du DHT11.

III.6.2.3Fonction Beep Sonore :

L'émission d'un beep sonore est réalisée par une fonction dont la fréquence et la durée sont comme suit :

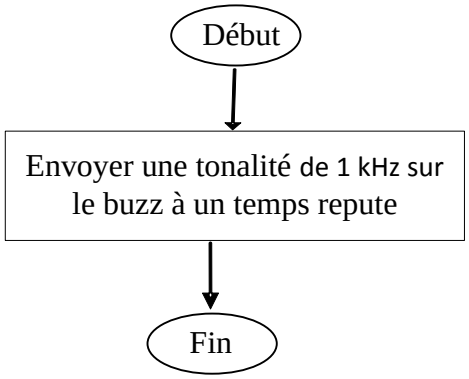


FIG.III.9 Organigramme pour générer un beep sonore.

Première partie du programme: 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.

- ✓ ESP8266 pour convertir l'information hexadécimale.
- ✓ Md parola : pour afficher.
- ✓ RTC : pour donner la date et l'heur.
- ✓ DHT : donner d'humidité température.
- ✓ Blynk – print serial

Après avoir déclaré les bibliothèques à utiliser, on commence la déclaration des variables et des ports de connexion des modules sur la carte

La deuxième Partie :

Fonction d'initialisation "Setup()" de l'IDE Arduino.

C'est la première fonction qui sera lancée au démarrage de l'ESP8266, dans lequel nous allons configurer nos différents objets, nos protocoles de communication, et notre configuration d'entrées/sorties, nous ne pouvons appeler cette fonction qu'une seule fois au démarrage du programme. Après la fonction setup(), nous avons ajouté quelques fonctions secondaires afin quelles soient utilisées dans la boucle de fonction last()

La troisième partie :

La fonction loop () est la deuxième fonction sur laquelle NODE s'exécutera, immédiatement après la fonction Setup (). En fait, cette fonction tourne à nouveau sur elle-même. Lorsque la dernière ligne de la boucle loop () sera exécutée par le programme, et la dernière reviendra à la première ligne. Dans notre programme, nous commençons cette partie en appelant toutes les fonctions que nous avons ajoutées après setup () correspondant à la fonction Read time, la fonction Read Temp/hum, la fonction alarme sonore. Les données seront envoyées à la carte Arduino pour les compiler et afficher.

III.7 L'application Blynk :

III.7.1 Commente fonctionne Blynk :

Pour pouvoir contrôler l'horloge à distance via le module WIFI, une application BLYNK doit être programmée pour pouvoir régler l'horloge. Blynk est une solution pour construire des applications lot disponible pour Androïde et iso basé sur des plateformes connues comme Arduino Raspberry pi. ESP8266, ente Galileo, particule périphérique série et wifi entre autre. Il peut contrôler un hardware à distance il peut afficher des données de capteur stocker des données les visualiser et faire beaucoup d'autre troue cools.

Il ya trois composant majeurs dans la plateforme

- Application Blynk : vous permet créer de fantastique interface pour le projet
- Serveur Blynk : responsable de toutes les communications entre le Smartphone et le hardware, vous pouvez utiliser le coude Blynk on faire tourner le serveur privé Blynk localement c'est open source ca peut facilement gérer des milliaire périphérique et peut même être démarré sur respire pi.
- Bibliothèques Blynk : pour toute la plateforme hardware populaire, active la communication avec le serveur et traite toutes les commandes entrantes et sortantes.
- Dans notre cas effectuerons le teste avec le module NodeMCU ESP8266 .

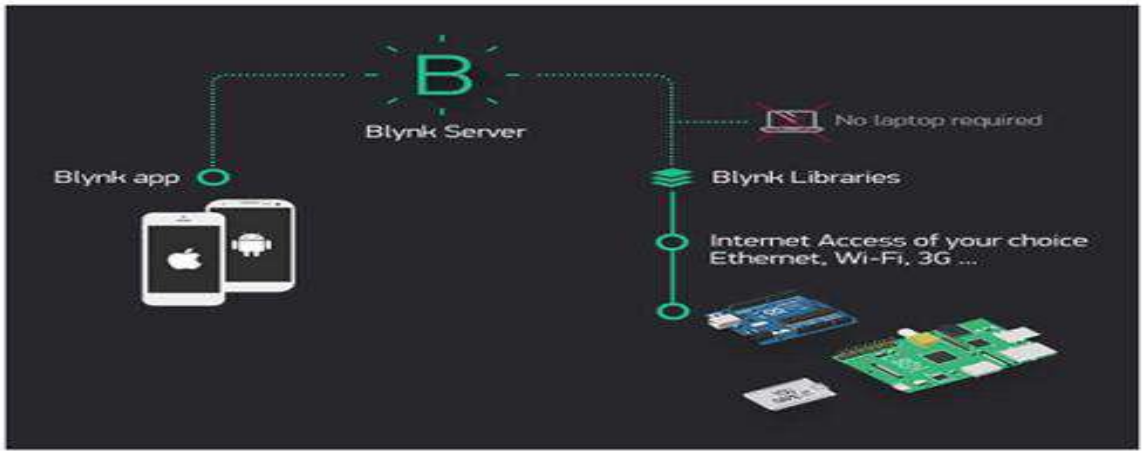


FIG III.10 : Fonctionnement de d'application Blynk.

Les déférent étapes suivre pore faire teste sur Blynk.

1. Introduction rapide à Blynk.
2. Installez Blynk sur Androïde.
3. Créer un compte sur Blynk.
4. Téléchargement des bibliothèques et des exemples
5. Révision du code IDE Arduino.

II.7.2 La creation de compete Blynk:

Une fois l'application Blynk téléchargé, nous avons besoin de créer un compte, qui est séparée des compotes utilisés pour les forums Blynk.



FIG III.11: Creation de compete Blynk.

III.7.3 Créer un nouveau Projet:

Les étapes à suivre pour créer un nouveau projet sont résumées par les figures suivantes :



FIG III.11: Création d'un nouveau projet.

Puis on remplit les champs correspondant au nom du projet et choisit le Hardware.

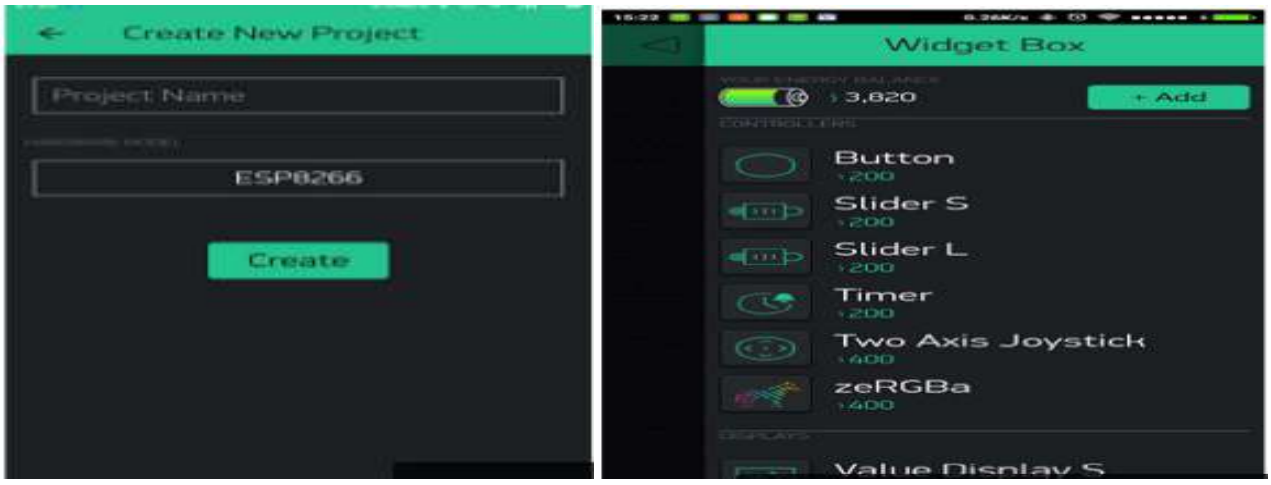


FIG III.13: nomination du projet et chois du hardware.

ET après on ajoute les windows qu'on a besoin.

III.7.4 Réalisation d'une application:

L'application de notre application dans le Blynk est réaliser en suivant les étapes résumés par la figue III.14.

Tout d'abord nous commencé par crée un nouveau projet et le nommé « Horloge intelligente ».

Après on précise le nome, les dispositifs et le type de connexion utilisé dans l'application.

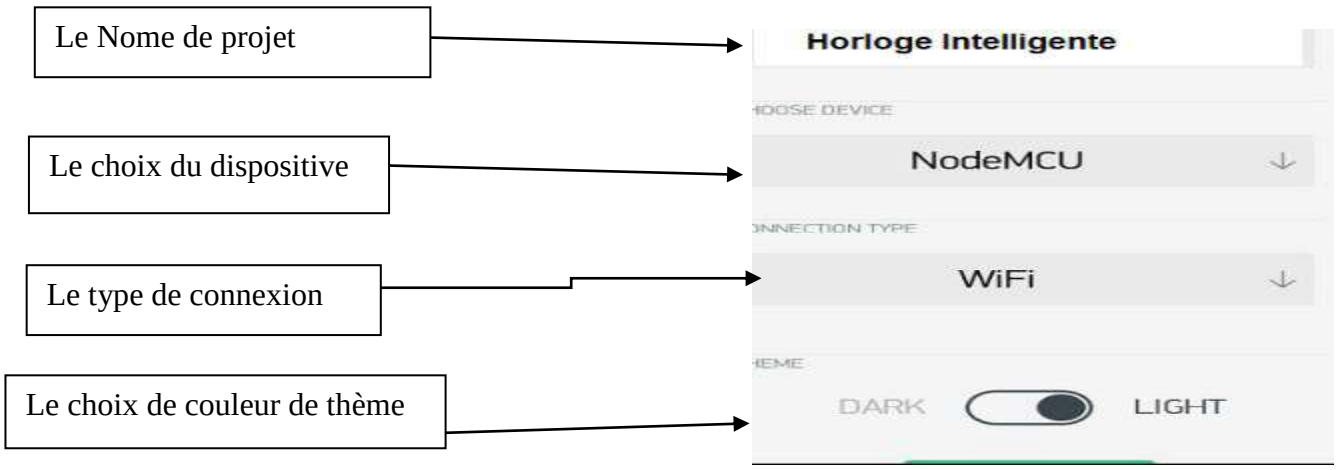


FIG III.14 : Le type de connexion utilise dans l'application

On reçoit par la suite un jeton TOKEN envoyé par email et qui sera copié sur notre programme.

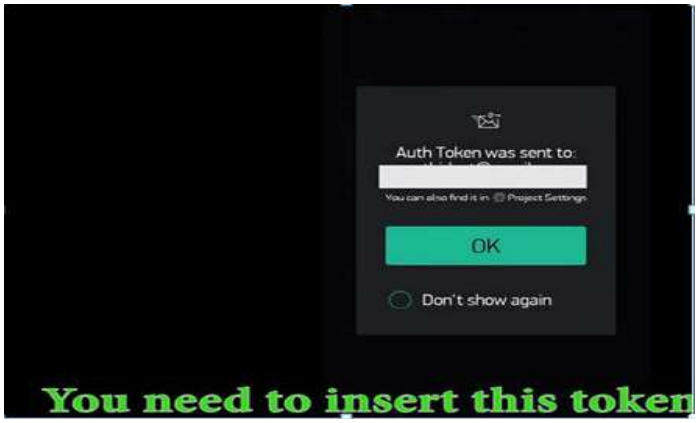


FIG III.15 : Confirmation par code.

Ensuite, nous choisissons l'outil terminal qui portera le message à transférer sur notre montre comme le montrent les figures suivantes.

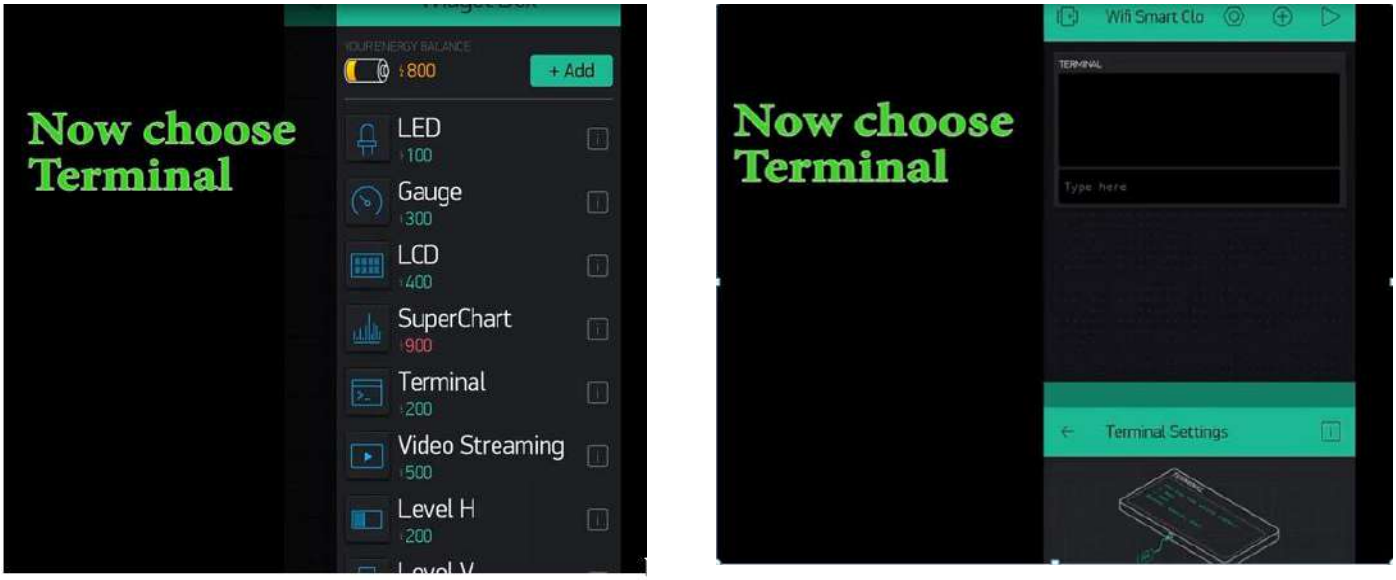


FIG III.16 : Choix du dispositif d'affichage.

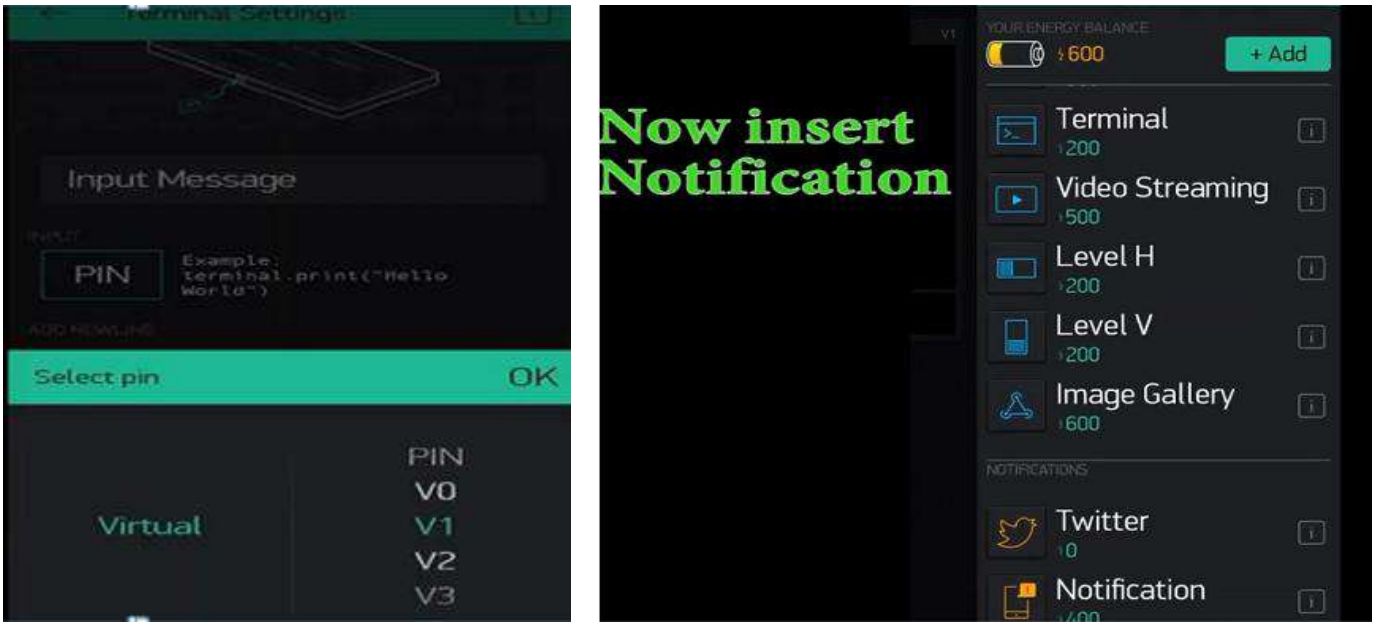


FIG III.17 : widget de notification.

III.8 Résultat de la réalisation :

Le résultat final de notre projet est présenté par la figure III.18, des exemples d'affichage sont montrés.

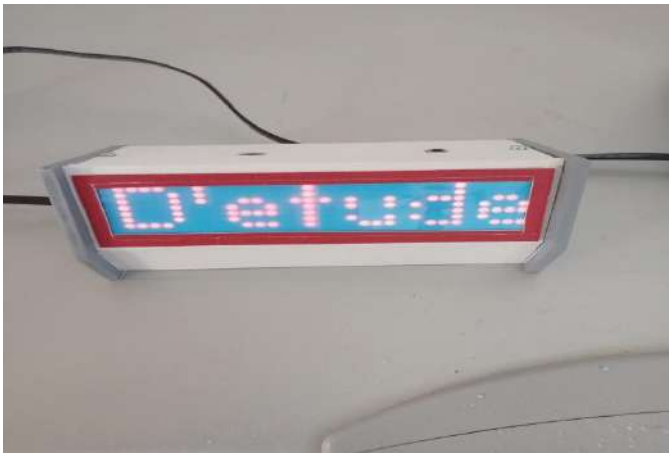




FIG III.18 : Résultat de la réalisation

III.9 Obstacle rencontré pendant la réalisation :

• Le premier obstacle auquel nous avons été confrontés a été le financement du projet sachant que le projet était des composants électroniques coûteux, précis et périssables autofinancé par nos soins.

• Manque de composantes nécessaires au niveau du laboratoire universitaires (Esp8266 NodeMCU, fils, capteur d'humidité...).

• Avec les matériaux nécessaires à la réalisation du projet que nous avons essayé de réaliser résultat avec les moyens disponibles

III.10 Conclusion:

Ce chapitre est la dernière partie de notre projet où nous avons testé notre partie théorie et simulation sous fritzing. Pour réaliser une montre numérique intelligente qui affiche l'heure, la date, la température et l'humidité correctement.

Afin d'obtenir le résultat souhaité, nous avons commencé par fabriquer une boîte, compléter la peinture du boîtier de l'horloge. Il était également important de travailler sur l'application Blynk, qui est compatible avec le Wifi pour contrôler l'horloge à l'aide d'un Smartphones.

Des images montrant le résultat final de notre projet sont présentés à la fin de ce chap

Conclusion Générale

Conclusion Générale

Notre sujet dans ce projet est de réaliser une horloge numérique basée sur la carte de contrôle numérique NodeMCU (ESP8266) qui appartient à la famille de cartes ARDUINO.

Ce projet est étudié en trois chapitres :

La première partie théorique consiste en une présentation de l'électronique programmée, et à travers elle nous avons introduit les différents types de cartes de programmation de la famille Arduino, leur principe de fonctionnement et leurs avantages. Nous avons également fourni des langages de programmation pour la carte Arduino et NodeMCU (ESP8266).

La deuxième partie est consacrée à la conception et à la simulation d'un circuit d'horloge numérique dans le cadre du programme Fritzing, chaque composant utilisé a été présenté, et nous nous concentrons sur les caractéristiques techniques de chacun.

Pour la dernière partie, il peut être en trois parties

Dans la première partie, nous avons parlé des différentes composantes du projet et de leurs relations les unes avec les autres

Quant à la deuxième partie, nous avons conçu une boîte complète pour tous les composants, en plus d'un affichage élégant de l'horloge à l'aide du programme de dessin SOLIDEWORKS

Dans la troisième et dernière partie, nous avons parlé de l'application Blynk et de son utilisation

Référence

[1] Marc Gatti,” Évolution des Architectures des Systèmes Avioniques Embarqués ”, thèse de doctorat Paris, de l’UPMC , France, Dec 2017.

[2]MihaSmolnikar, MihaelMohorcic, ” A Framework for Developing a Microchip PIC Microcontroller Based Applications”, WSEAS TRANSACTIONS on ADVANCES in ENGINEERING EDUCATION , Issue 2, Volume 5, February 2008, pp 83-91.

[3] A. H. G. Al-Dhaher, Integrating hardware and software for the development of microcontroller-based systems, Microprocessors and Microsystems, vol. 25, issu. 7, October 2001, pp. 317-328.

[4] https://arduino-esp8266.readthedocs.io/en/latest/

[5] M ABDALLAH Fethi,” Conception d’un système immotique, Smart building, pour la société CNAS, mémoire de Master Université Saad Dahleb de Blida, Algérie, 2018-2019

[6]Ali Al Dahoud, Mohamed Fezari“ Presentation, Programming and Applications of the NodeMCU-ESP-01 module”. Project WSN in solarPower Panel Monitoring and Water quality monitoring.June 2019. *DOI:10.13140/RG.2.2.18894.72007*.

7- Manoj R. Thakur, ” NodeMCU ESP8266 Communication Methods and Protocols : Programming with Arduino IDE”, 1 ere edition, Amazon Media EU S.àr.l. 2018.

[8] PradeekaSeneviratne ” Robotisc Projects ESP8266 : robots DIY contrôlés par Wi-Fi”, 1ereédition, Packt Publishing Limited, Birmingham, UK, 2017.

[9] Neil Cameron, ” Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices”, 1ere edition, Apress 2021.

[10] Marco Schwartz, “Internet of Things with ESP8266”, 1ere edition, Packt Publishing, 2016.

[11] Jeremy Blum, “Exploring Arduino: Tools and Techniques for Engineering Wizardry”, 2eme edition, Wiley, 2019.

[12] Tero Karvínen, Kimmo Karvínen, Ville Vaitokarí, “ les capteurs pou r Arduino et Raspberry Pi Tutoriels et projets, Dunod edition 2014.

[13] Mark Geddes , « Arduino Project Handbook 25 Practical Projects to Get You started”, 1ere edition William Pollock, 2016.

[14] André KnörigBrendan Howell Advanced Prototyping with Fritzing,Conference: Proceedings of the 4th International Conference on Tangible and Embedded Interaction 2010, Cambridge, MA, USA, January 24-27, 2010.

[15] AndréKnörig, RetoWettach, Jonathan Cohen, “Fritzing: a tool for advancing electronic prototyping for designers”, Conference: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction 2009, Cambridge, UK, February 16-18, 2009.

[15] Kuang-HuaChang, Design Theory and Methods Using CAD/CAE, The Computer Aided Engineering Design Series, Project S5 - Design with SolidWorks 2014, 2014 Elsevier Inc. Pages 407-447