



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Amar Thelidji- Laghouat

FACULTE: DE TECHNOLOGIE
DEPARTEMENT : D'ELECTRONIQUE

MEMOIRE DE MASTER

Réalisé par : BEYAZ Kheira et BENBEHAZ Amina

DOMAINE : Science et technologie

FILIERE : ELECTRONIQUE

OPTION : Electronique Des Système Embarqués

Thème

**Conception et réalisation d'un système
de télésurveillance des patients en temps
réel**

Jury de soutenance :

| Nom et Prénom | Grade | Qualité |
|------------------|-------|-----------|
| GUEFFAF Hamza | MCA | Encadrant |
| SEGHIER Tahar | Pr | Président |
| DELLACI Mustapha | MAA | Examineur |

Promotion : 2020/2021

Résumé

ملخص:

الهدف من هذا المشروع هو تصميم جهاز مراقبة عن بعد يسمح بمراقبة الحالة الصحية للمريض بحيث يعمل على قياس درجة الحرارة وقياس نبضات القلب ونسبة الأوكسجين لدى المريض ويمكن التحكم فيها عن بعد بواسطة أجهزة استشعار وإرسالها عبر وبي في وهذا عن طريق **Nodemcu** لتحقيق هذا الهدف وبعد دراسة لوحات الأردوينو استقرينا على لوحة نود امسيو وتعلمنا كيفية البرمجة الخاصة بها باعتبارها اللوحة الأكثر مناسبة خاصة على احتوائها على وبي في الذي يسهل عملية نقل المعلومات وعلى هذا الأساس تم اختيار باقي معدات جهازنا المختلفة و الضرورية لمشروعنا.

Résumé

Le but de ce projet est de concevoir un dispositif de surveillance à distance qui permet de surveiller l'état de santé du patient afin qu'il mesure la température, la fréquence cardiaque et le taux d'oxygène du patient et puisse être contrôlé à distance par des capteurs et envoyé via Wifi et ce via nodemcu

Pour atteindre cet objectif, et après avoir étudié la plupart des cartes Arduino, nous avons opté pour, la carte Nod Amcio et avons appris à la programmer, car c'est la carte la plus adaptée, notamment parce qu'elle contient le Wifi, ce qui facilite le processus de transfert informations.

Mots clés : carte wifi, arduino, capteurs

Abstract

The goal of this project is to design a remote monitoring device that can monitor the patient's health status so that it measures the patient's temperature, heart rate and oxygen level and can be controlled remotely. by sensors and sent via Wifi and this via nodemcu

To achieve this goal, and after studying most Arduino boards, we opted for the Nod Amcio board and learned to program it, because it is the most suitable board, especially because it contains Wifi, which facilitates the information transfer process.

Keywords: wifi card, arduino, sensors

Dédicace

Sont nombreux à qui je dédie ce mémoire A

mon très cher PAPA

En signe d'amour, de reconnaissance et de gratitude pour le dévouement et les sacrifices dont vous avez

fait toujours preuve à mon égard

A ma très chère MAMAN

C'est un moment de plaisir de vous dédier cette œuvre, vous qui raviver dans mon esprit un sentiment profond d'une vie sûre et correcte, suivi tant par vos chaleureuses bénédictions.

A mes chères sœurs.

FATIMA ZOHRA, HASSNA, NESSRINE, NOUR ALHOUDA.

Qui m'ont soutenu dans les moments les plus durs, je ne les remercierais jamais assez.....

A mon frère ALLAL

Pour son présence et son soutien Ames

tantes et oncles, et à toute la famille

A ma chère copine BEYAZ KHEIRA, avec qui j'ai partagé des moments inoubliables voyant naître cette œuvre.

Que tous mes amis, KHENIFER FAIROUZ, SABKHAOUI ZINEB, DINAR FIRIEL, ALLAOUI KHADIDJA, BELARBI RIHAB, LAGHA ADEL, MARFOUAA OUSSAMA, HAMDI YACINE, CHATTA ABDALLAH.

Trouvent en ce travail l'expression de ma profonde gratitude pour leur agréable compagnie, et leur chaude amitié.

A tous ceux qui m'ont assisté et encouragé A

tous ceux qui me sont chers

Puisse le seigneur DIEU tout puissant vous combler de toutes ses bénédictions

AMINA

Dédicace

A mes chers parents "MOSTAPHA" et "HADDA" Les mots me manquent pour exprimer toute la reconnaissance, la fierté et profond amour que je vous porte pour les sacrifices, et les prières que vous n'avez jamais cessé de consentir pour mon instruction et ma réussite. Que Dieu tout puissant vous garde et vous procure santé, bonheur et longue vie pour que vous demeuriez le flambeau illuminant mon chemin.

A mon frères "LAID" et mes sœurs "IKHLAS" "AMINA" pour leurs encouragements permanents, et leur soutien moral

A mon cher oncle NADIR, qui était toujours comme un grand frère pour moi qui m'a assisté dans tous les moments difficiles.

Toute ma famille, pour leur soutien tout au long de ces longues années d'études.

A ma binôme «AMNA» ainsi à toute sa famille ;

À mes chers ami(e)s "MERIEM OTHMANI", "HOUDA, KHADRA, ZINEB, FATIMA, REHAB" Pour tous les beaux moments que nous avons passés ensemble

A tous ceux qui me sont chers.

KHEIRA

Remerciements

**Nous remercions DIEU le miséricordieux de nous avoir données la sante, la
volonté et la force d'atteindre notre but.**

**Nous tenons à exprimer nos vifs remerciements à notre encadreur Mr. GUEFFAF
HAMZA de nous avoir proposé ce sujet, ainsi que pour ses précieux conseils tout
au long de notre travail, pour son aide et pour sa confiance.**

**Nous souhaitons aussi remercier tous les enseignants de l'université Amar
Thelidji Laghouat, et en particulier nos enseignants d'électronique la pour le
savoir qu'ils nous ont transmis, pour leur disponibilité et leur gentillesse.**

**Nous remercions aussi très chaleureusement les membres du jury pour l'honneur
nous qu'ils ont fait en acceptant d'être rapporteurs de notre mémoire.**

AMINA et KHEIRA

APL: Arduino programation language

GND: GrouND.

Http: Hypertext Transfer Protocol

IDE: Integrated Development Environment.

I2C: Inter-Integrated Circuit.

MISO: Master Input, Slave Output.

MOSI: Master Output, Slave Input.

PWM: Pulse Width Modulation

RST: Reset Input

RAM: Random Access Memory.

SPI: Serial Peripheral Interface.

SCL: Serial Clock Line.

SCK : Serial Clock.

SD: Secure Digital.

SDA: Serial Data Line.

SS: Slave Select.

SCLK: Serial Clock, Horloge.

SRAM: Static Random Access Memory.

USB: Universel Serial Bus.

Sommaire

Dédicaces Remerciement Liste des figures Liste des tableaux

| | |
|--|----|
| Introduction générale..... | 1 |
| Chapitre I: les systèmes embarqués et la télésurveillance | |
| I Systèmes embarqués..... | 3 |
| I .1 Introduction..... | 3 |
| I.2 l'historique:..... | 4 |
| I.3 Définition d'un système..... | 5 |
| I.3.1 Définition d'un système embarqué..... | 5 |
| I.3.2 L'utilisation d'un système embarqué..... | 6 |
| I.3.3 Fonctionnement d'un système embarqué..... | 7 |
| I.3.4 Caractéristiques d'un système embarqué..... | 7 |
| I.3.5 Domaines d'applications..... | 8 |
| I.3.6 Les Avantages d'un système embarqué..... | 10 |
| I.3.7 Les inconvénients d'un système embarqué..... | 10 |
| I .4 Préambule..... | 11 |
| I .4.1 Définition La télémédecine..... | 11 |
| I .4.2 Définition de la télésurveillance..... | 12 |
| I .4.2.1 domaine d'application..... | 12 |
| I.2.2.3 avantage et inconvénients de la télésurveillance..... | 13 |
| I .5 Conclusion..... | 13 |
| Chapitre II: Arduino et la carte wifi Nodemcu | |
| II.1 Introduction..... | 14 |
| II.1.1 Historique..... | 15 |
| II.1.2 Définition de la carte arduino..... | 15 |
| II.1.3 Domaines d'application des cartes arduino. | 16 |
| II.1.4 Avantages d'Arduino..... | 16 |
| II.1.5 Inconvénients de la carteArduino..... | 16 |
| II.1.6 Différentes cartes Arduino | 17 |
| ARDUINO UNO | 17 |
| ARDUINO NANO..... | 18 |
| ARDUINO DUE..... | 18 |
| ARDUINO MEGA..... | 19 |

| | |
|---|----|
| ARDUINO LEONARDO..... | 19 |
| ARDUINO NODEMCU | 20 |
| II.2.1 Cartes nodemcu (esp8266)..... | 22 |
| II.2.2 Caractéristiques du module ESP8266..... | 22 |
| II.2.3 Principe de fonctionnement..... | 23 |
| II.2.4 Arduino et NodeMCU..... | 23 |
| II.2.5 Architecture interne..... | 24 |
| II.2.6 Définitions des pins..... | 25 |
| II.2.7 Etapes d'installation du module sur l'IDE..... | 28 |
| II.2.8 Applications de la carte Nodemcu..... | 31 |
| II.2.9 Avantage de Nodemcu..... | 32 |
| II.3 Définition d'un capteur | 32 |
| II.3.1 Capteur max30100..... | 32 |
| II.3.2 Caractéristiques..... | 33 |
| II.3.3 Applications..... | 33 |
| II.3.4 Capteur MAX30205..... | 33 |
| II.3.5 Caractéristiques du capteur MAX30205..... | 34 |
| II.3.6 Les application du capteur MAX30205..... | 35 |
| II.4 conclusion..... | 35 |

Chapitre III : conception et réalisation

| | |
|---|----|
| III Introduction..... | 36 |
| III.1 Le principe de fonctionnement de notre système..... | 36 |
| III.1.1 Branchement avec Nodemcu..... | 37 |
| III.1.2 Connexion de capteur MAX30100 (montage)..... | 38 |
| III.1.3 code de programme..... | 39 |
| III.1.4 Branchement avec Nodemcu:..... | 45 |
| III.1.5 connexion de capteur MAX30205(montage)..... | 45 |
| III.1.6 Code de programme..... | 46 |
| III.2.1 OLED (organic light emitting Diode)..... | 47 |
| III.2.2 Connexion avec nodemcu..... | 48 |
| III.2.3 Connexion des pins de la carte avec OLED | 48 |
| III.2.4 Code de programmation avec l'IDE..... | 49 |
| III.2.5 montage final..... | 59 |

| | |
|--------------------------------|----|
| III.2.6 code finale..... | 60 |
| III.3.1 schéma synoptique..... | 66 |
| III.3.2. l'organigramme..... | 67 |
| III.4 Conclusion..... | 68 |

LISTE DES FIGURES

Chapitre I: Les systèmes embarqués et la télésurveillance Figure I.1

| | |
|---|----|
| système embarqué | 5 |
| Figure I.2 système embarqué | 6 |
| Figure I.3 champs de système embarqué..... | 10 |
| Figure I.4 la télémédecine..... | 11 |

Chapitre II: Arduino et carte wifi

| | |
|--|----|
| Figure II.1 Logod' une carte Arduino..... | 14 |
| Figure II.2 Carte Arduino..... | 15 |
| Figure II.3 Carte Arduino UNO..... | 17 |
| Figure II.4 Carte Arduino NANO | 18 |
| Figure II.5 Carte Arduino DUE | 18 |
| Figure II.6 Carte Arduino MEGA | 19 |
| Figure II.7 Carte Arduino LEONARDO..... | 19 |
| Figure II.8 Carte NODEMCU (esp8266) | 20 |
| Figure II.9 NodeMCU ESP8266..... | 22 |
| Figure II.10 L'architecture interne du module ESP8266 | 24 |
| Figure II.11 Les pins de l'ESP8266 | 26 |

Chapitre III: conception et réalisation

| | |
|---|----|
| Figure III .1 Branchement du capteur max30100 avec Nodemcu..... | 35 |
| Figure III.2 Capteur max30100 avec la carte Nodemcu (esp8266)..... | 36 |
| Figure III.3 Résultat du MAX30100 sur le moniteur série..... | 42 |
| Figure III.4 Branchement du capteur max30205 avec nodemcu..... | 44 |
| Figure III.5 Connexion du capteur MAX30205 avec Nodemcu | 44 |
| Figure III.6 Résultat de température sur moniteur série | 46 |
| Figure III.7 Module OLED | 46 |
| Figure III.8 Branchement OLED avec Nodemcu(esp8266)..... | 47 |
| Figure III.9 Montage finale..... | 58 |

Liste des tableaux

Chapitre II: Arduino et carte wifi

| | |
|---|----|
| Tableau II.1: Caractéristiques des différentes cartes Arduino..... | 21 |
| Tableau II.2: comparaison entre un Arduino et un NodeMCU | 23 |
| Tableau II.3 : Les pins de l'ESP8266..... | 25 |

Chapitre III: conception et réalisation

| | |
|---|----|
| Tableau III.1: Les pins de la configuration NODEMCU (esp8266) et OLED..... | 47 |
|---|----|

Introduction générale

Introduction Générale

Les progrès prodigieux, de ces dernières années, dans l'électronique embarquée, ont permis le déploiement à grande échelle des technologies touchant plusieurs domaines d'applications tels que la télésurveillance l'avionique, la télécommunication, le transport, etc.. [1].

Un système embarqué présente une intégration entre deux parties, logicielle et matérielle, qui sont conçues conjointement pour répondre à des fonctionnalités spécifiques. Ces fonctionnalités sont, pour la plus part des cas, des applications critiques.

En raison du manque d'infrastructures d'accueil et du manque de professionnels capables de diagnostiquer correctement les personnes à risques insalubres, le domaine médical s'est développé, notamment en recherchant un traitement médical à distance à domicile.

L'objectif principal est de réaliser un système de control d'un patient à distance en temps réel. Toutes ces études s'inscrivent dans le thème de « conception et réalisation d'un système de télésurveillance des patients en temps réel ».

Nos objectifs de recherche tournent autour de la conception et du développement de services de veille sanitaire, notamment dans le contexte de la pandémie de COVID-19.

Pour cela notre mémoire est structuré comme suit :

Le premier chapitre, nous présenterons le contexte général du projet. Ce dernier sera divisé en 3 parties : la 1ère partie présentera le système embarqué en générale, et la 2ème partie représente la télémédecine la 3ème partie fait nous voir le concept de télésurveillances, qui représente l'épine dorsale du domaine du suivi médical. Une autre entité a déclenché avec succès une énorme révolution dans le monde médical.

Après dans le deuxième chapitre, nous ferons une description générale de la carte Arduino et la carte Nodemcu (esp8266)

Avant de conclure notre travail dans troisième chapitre, nous définissons les composants

Introduction Générale

et les matériels nécessaires pour la réalisation de ce projet ; la simulation et la programmation en générale

Finalemment nous clôturons ce mémoire par une conclusion générale, qui reprendra les points essentiels de tous les travaux et les perspectives envisagées.

Chapitre I : Les

Systemes Embarqués et la

Télesurveillance

I Systèmes embarqués

I.1 Introduction

De nos jours, les systèmes embarqués nous entourent et nous sommes littéralement envahis par eux. Il suffit de regarder autour de soi au quotidien pour s'en rendre compte. On retire de l'argent dans un distributeur, c'est un système embarqué.

On prend sa voiture, le régulateur de vitesse, la direction assistée électrique, le système de contrôle de trajectoire c'est des systèmes embarqués. Le pilote automatique dans un avion est un système embarqué, et les exemples ne manquent pas. Dans divers domaines, au cours de ces dernières années, le progrès technologique a fait que beaucoup de fonctions réalisées manuellement, mécaniquement et hydrauliquement se dématérialisent au profit de l'électronique et de l'informatique embarquée.

Un confort considérable auquel il est de plus en plus difficile de renoncer. Les systèmes électroniques sont de plus en plus présents dans la vie courante. Les ordinateurs et micro-ordinateurs sont des systèmes électroniques bien connus. Mais l'électronique se trouve maintenant embarquée dans de très nombreux objets usuels : les téléphones, les agendas électroniques, les voitures. Ce sont ces systèmes électroniques enfouis dans les objets usuels qui sont appelés systèmes embarqués.

Les systèmes embarqués font intervenir des technologies différentes, en plus de comporter une partie logicielle et une partie matérielle.

I.2 Historique

L'un des premiers appareils informatiques électroniques à avoir reçu le terme «système embarqué» et plus proche de notre conception actuelle de tel était l'ordinateur de guidage Apollo (AGC), développé au laboratoire d'instrumentation du MIT par un groupe de concepteurs dirigé par Charles Stark Draper au début des années 1960, l'AGC faisait partie du système de guidage et de navigation utilisé par la NASA dans le programme Apollo pour divers vaisseaux spatiaux.

À ses débuts, il était considéré comme l'un des éléments les plus risqués du programme Apollo en raison de l'utilisation des circuits intégrés monolithiques nouvellement développés.

Le début de la décennie 1970 a vu le développement du premier microprocesseur dessinés. À la fin de 1971, presque simultanément et indépendamment, les équipes de conception pour Texas Instruments, Intel et l'US Navy avaient développé des implémentations des premiers microprocesseurs.

Gary Boone de Texas Instruments a obtenu en 1973 le brevet de la première architecture de microprocesseur mono puce pour sa conception de 1971 du TMS1000 (Fig.1.3). Cette puce était un processeur 4 bits qui incorporait dans la même puce 1K de ROM et 256 bits de RAM pour offrir une fonctionnalité informatique complète dans une seule puce, ce qui en faisait le premier micro-ordinateur sur puce (alias microcontrôleur). Le TMS1000 a été lancé en septembre 1971 en tant que puce de calcul avec numéro d'article TMS1802N

Les applications embarquées basées sur des microprocesseurs ont atteint des centaines de millions de dollars. La liste des premiers acteurs est passée à plus d'une douzaine de fabricants de puces qui, en plus du Texas Instruments et Intel, inclus Motorola, Zilog, Intersil, National Instruments, MOS Technologie et Signétique, pour ne citer que quelques-uns des plus renommés. Parmi les pièces remarquables, citons l'Intel 8080 qui a finalement évolué pour devenir la célèbre série 80 × 86 / Pentium, le Zilog Z-80, le Motorola 6800 et le MOS 6502.

L'évolution des tailles des processeurs s'est poursuivie dans les années 1980 et 1990 jusqu'à 16-, 32- et Des conceptions 64 bits et, de nos jours, même des processeurs spécialisés qui traitent les données à des largeurs de 128 bits. En termes de fabricants et de disponibilité des processeurs, la liste s'est allongée au point qu'il est possible de trouver plusieurs dizaines de choix différents pour les tailles de processeur 32 bits et plus, et des centaines de processeurs 16 et 8 bits.

Les exemples de fabricants disponibles aujourd'hui incluent Texas Instruments, Intel, Micro chip, Freescale (anciennement Motorola), Zilog, Advanced Micro Devices, MIPS Technologies, ARM Limited, et la liste est longue. De nos jours, les applications des microprocesseurs sont devenues plus complexes; exigeant des applications pour être divisé en plusieurs systèmes embarqués interactifs.



Figure I .1. Système embarqué

I.3 Définition d'un système

Un système est un arrangement dans lequel toutes ses unités assemblées travaillent ensemble selon un ensemble de règles. Elle peut également être définie comme une manière de travailler, d'organiser ou d'effectuer une ou plusieurs tâches selon un plan fixe. On peut donc dire, dans un système, que toutes ses sous-composantes dépendent les unes des autres.

I.3.1 Définition d'un système embarqué

Un système embarqué ou « Embedded system » en anglais, tout système conçu d'une architecture d'ordinateur pour résoudre un problème ou une tâche spécifique mais n'est pas un ordinateur d'usage général.

Il travaille avec une précision d'horloger et rassurez-vous, il n'est jamais mal embarqué. Vu de près, il ressemble souvent à une carte rectangulaire avec fond vert, truffé de composants de toutes les formes et de petits fils. Le système embarqué est à la fois un matériel et un logiciel utilisé en informatique et en électronique, avec des applications dans de nombreux domaines.

Logé à l'intérieur d'une machine, d'un appareil ou d'un véhicule, il se voit assigner une tâche précise qu'il se doit d'exécuter. Il s'agit d'un système autonome qui doit réaliser son travail en tenant compte de plusieurs contraintes : sa taille (il dispose d'un espace mémoire très limité), le temps d'exécution dont il dispose et l'énergie (on cherchera toujours le meilleur ratio fonctionnement / consommation d'énergie) .[2]



Figure I.2.Système embarqué

I.3.2 L'utilisation d'un système embarqué

A exécuter des tâches au sein d'une machine. Quelles tâches, exactement ? Cela dépend de la machine dans laquelle il se trouve. Les systèmes embarqués électriques sont très utilisés en aéronautique et dans l'industrie automobile (avions, fusées, voitures).

On les trouve également dans le domaine militaire à l'intérieur des missiles, dans les distributeurs automatiques de billets, dans les consoles de jeux, dans certains matériels médicaux, dans les appareils électroménagers (lave-vaisselle, télévision, four à micro-ondes), dans les imprimantes... Et bien entendu, dans les ordinateurs, les téléphones portables, les disques durs et les lecteurs de disquettes.

Ces accessoires qui vous semblaient inutiles à l'époque sont indispensables pour confectionner des systèmes embarqués. La fabrication des systèmes embarqués nécessite aussi de nombreux composants comme un microprocesseur, un compilateur croisé, des résistances (celles que vous brisiez par erreur pendant les cours) et un programmeur in-situ [2].

I.3.3 Fonctionnement d'un système embarqué

Un système embarqué doit être capable d'exécuter des tâches en temps réel. Pour ce faire, il est équipé de capteurs, d'actionneurs et d'une interface. Imaginez par exemple le système embarqué du portail électrique d'une maison. Le capteur intégré détecte les contacts, un signal qui entraîne l'ouverture ou la fermeture du portail par un moteur.

Les systèmes embarqués appartiennent à la grande famille de l'intelligence artificielle faible. Cette intelligence est dite « faible » car le système n'est pas capable de ressentir des émotions ni de communiquer avec les êtres humains, contrairement à l'intelligence artificielle forte. Le système embarqué se contente de réaliser des tâches prédéfinies (ce qui n'est déjà pas mal, reconnaissons-le).

Les robots aspirateurs fonctionnent de la même façon que les portails automatiques. Les capteurs détectent les objets et les différents obstacles qui altèrent son bon fonctionnement. Le système embarqué perçoit l'anomalie et ordonne alors au moteur de changer de direction, tandis que la machine continue d'aspirer le sol. Vous l'aurez compris : nous sommes constamment entourés de systèmes embarqués [2].

I.3.4 Caractéristiques d'un système embarqué

Plutôt que des systèmes universels effectuant plusieurs tâches, les systèmes embarqués sont étudiés pour effectuer des tâches précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de rentabilité. D'autres ayant peu de contraintes au niveau performances permettent de simplifier le système et de réduire les coûts de fabrication.

Les systèmes embarqués ne sont pas toujours des modules indépendants. Le plus souvent ils sont intégrés dans le dispositif qu'ils contrôlent.

- ✓ Utilise généralement un microprocesseur combiné avec d'autres matériels et logiciel pour résoudre un problème de calcul spécifique.
- ✓ Système électronique et informatique autonome ne possédant pas des entrées
- ✓ sorties standards.
- ✓ Le système matériel et l'application sont intimement liés et noyés dans le matériel et ne sont discernables comme dans un environnement de travail classique de type PC.

- ✓ N'est pas visible en tant que tel, mais est intégré dans un équipement doté d'une autre fonction ; ou dit aussi que le système est enfoui, ce qui traduit plus fidèlement le terme anglais « Embedded ».
- ✓ Une faible barrière existe entre les systèmes embarqués et les systèmes temps réel (un logiciel embarqué n'a pas forcément de contraintes temps réel).
- ✓ La conception de ces systèmes est fiable (avions, système de freinage ABS) à cause de leur utilisations dans des domaines à fortes contraintes mais également parce que l'accès au logiciel est souvent difficile une fois le système fabriqué
- ✓ Les microprocesseurs utilisés s'étendent depuis de simples microcontrôleurs 8bits aux 64bit les plus performants et les plus sophistiqués.
- ✓ Le logiciel système inclus s'étend d'un petit programme à un grand logiciel d'exploitation en temps réel (RTOS) avec une interface utilisateur graphique (GUI). Typiquement, le logiciel système inclus doit répondre aux événements d'une manière déterministe et devrait toujours être opérationnel. [2]

🚦 Fonctionnement en Temps Réel

Le temps Réel est un concept un peu vague. On pourrait le définir comme : "Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente". Cela veut dire que dans le cas d'une information arrivant de façon périodique (sous forme d'une interruption périodique du système), les temps d'acquisition et de traitement doivent rester inférieurs à la période de rafraîchissement de cette information. Pour cela, il faut que le noyau ou le système Temps Réel soit déterministe et préemptif pour toujours donner la main durant le prochain tick à la tâche de plus forte priorité prête [3].

🚦 Généralement, un système embarqué doit respecter :

- des contraintes temporelles fortes (Hard Real Time).
- on y trouve enfoui un système d'exploitation ou un noyau Temps Réel (Real Time Operating System, RTOS).
- Réactivité : des opérations de calcul doivent être faites en réponse à un événement extérieur (interruption matérielle).
- La validité d'un résultat (et sa pertinence) dépend du moment où il est délivré.
- Rater une échéance va causer une erreur de fonctionnement :

- Temps Réel dur : plantage.
- Temps Réel mou : dégradation non dramatique des performances du système.
- Beaucoup de systèmes sont « multi rate » : traitement d'informations à différents rythmes.

I.3.5 Domaines d'applications

Les domaines dans lesquels on trouve des systèmes embarqués sont de plus en plus nombreux :

Transport : Automobile, Aéronautique (avionique), etc.

Astronautique : fusée, satellite artificiel, sonde spatiale, etc.

Militaire : missile

Télécommunication : Set-top box, téléphonie, routeur, pare-feu, serveur de temps, téléphone portable, etc.

Électroménager : télévision, four à micro-ondes

Impression : imprimante multifonctions, photocopieur, etc.

Informatique : disque dur, Lecteur de disquette, etc.

Multimédia : console de jeux vidéo, assistant personnel guichet automatique bancaire (GAB)

Automate programmable industriel, contrôle-commande

Jeux : consoles

Équipement médical

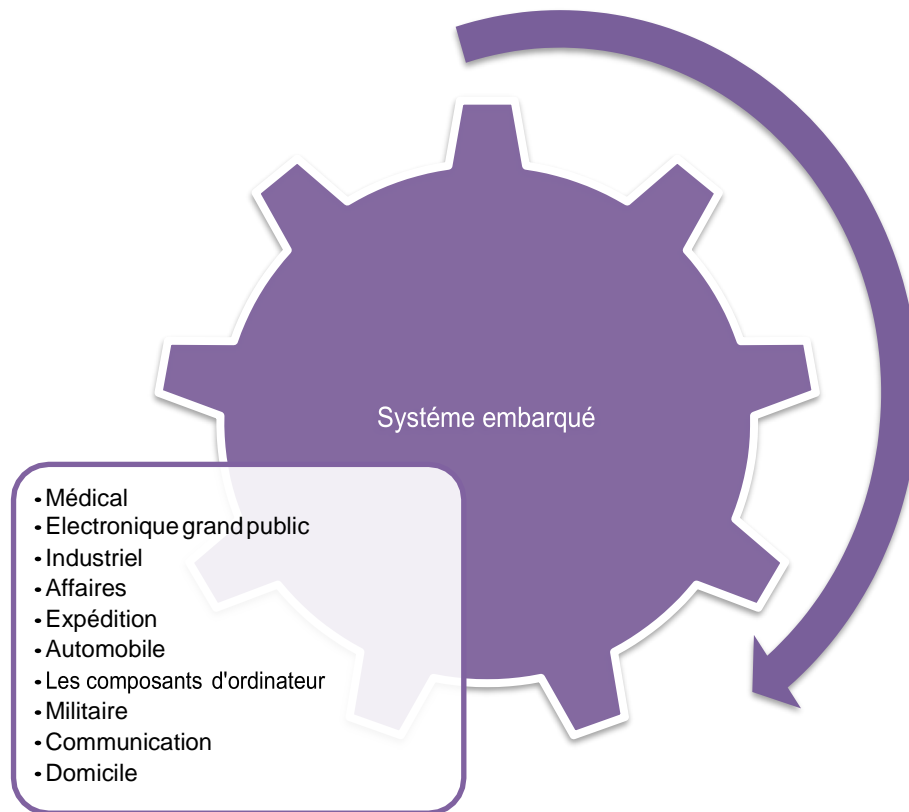


Figure I.3. Champs de système embarqué

I.3.6 Les Avantages d'un système embarqué

- ✓ Facilement personnalisable.
- ✓ Faible consommation d'énergie a bas prix.
- ✓ Performance améliorée.

I.3.7 Les inconvénients d'un système embarqué

- ✓ Effort de développement élevé.
- ✓ Délai de mise sur le marché plus long [3]
- ✓ Maintenance délicate
- ✓ Cout de réalisation des fois élevé

I. 4 Préambule

Le besoin de renforcer les niveaux de sécurité se fait de plus en plus ressentir dans différents domaines d'activités. Un des moyens utilisé est la télésurveillance.

La télésurveillance est omniprésente et on la retrouve dans de nombreux secteurs d'activité (banque, transport, industrie, etc.) ou lieux de vie (ville, immeubles de bureaux,... etc.) dans le but de surveiller et de protéger des personnes et des biens. Dans ce chapitre nous allons étudier les composants essentiels de notre système de télésurveillance : MAX30100 est une solution intégrée de capteur d'oxymétrie de pouls et de moniteur de fréquence cardiaque, le capteur de température MAX30205 mesure avec précision la température et on termine par un module wifi. (ESP8266).

I.4.1 Définition La télémédecine

La télémédecine est une forme de pratique médicale à distance utilisant les technologies de l'information et de la communication. Concrètement, c'est se servir de la technologie pour mettre en relation un patient avec des médecins, pour surveiller l'état de santé d'une personne, pour prescrire des médicaments, pour faire un suivi post-thérapeutique et qui a pour objectif de diminuer le nombre d'hospitalisations et améliorer la qualité de vie; dans ce chapitre nous allons voir tous les détails qui concerne la maladie surveillé et citer tous les périphériques utilisé dans notre projet. [4]



Figure I.4.1a télémédecine

I.4.2 Définition de la télésurveillance

La télésurveillance est la surveillance à distance d'un lieu , public ou privé, de machines ou d'individus. Elle est employée dans de nombreuses situations, généralement pour des raisons de sécurité. Son but est de surveiller des installations de systèmes de sécurité alarmes technique, alarmes vidéo, alarmes médicale (personnes âgées), alarmes agression et même de la vidéosurveillance [5].

I.4.2.1 Domaine d'applications

La télésurveillance est employée dans de nombreuses situations, généralement pour des raisons de sécurité :

- Dans le cadre de la sécurité routière, au moyen de caméras spécialisées ou des capteurs à proximité voire même noyés dans la chaussée permettent d'évaluer la densité du trafic, les ralentissements qui peuvent en découler, la présence de personnes sur les bandes d'arrêt d'urgence, etc.
- Pour la surveillance des machines : divers capteurs permettent d'évaluer l'état de la machine, ces informations peuvent alors être envoyées à un poste de surveillance.

L'épuisement de consommables, une anomalie de fonctionnement ou même un acte de malveillance serait alors détecté à distance.

Dans le cadre de la prévention de la délinquance (avec notamment la vidéosurveillance)

- Pour la surveillance de lieux sensibles (banques, centrales nucléaires, etc.) et d'habitations, afin de prévenir les intrusions, les cambriolages et les actes de vandalisme.
- Dans le cadre de la télémédecine, et en particulier pour la surveillance des patients à distance.
- Pour la surveillance à distance des enfants et des personnes vulnérables [5].

I.4.2.3 Avantages et inconvénients de la télésurveillance

La télésurveillance présente les avantages suivants :

- ✓ La dissuasion des voleurs par l'affichage des caméras de télésurveillance sur écran de télésurveillance.
- ✓ l'analyse visuelle des intrusions et vols sur les enregistrements vidéo des caméras.
- ✓ les images capturées peuvent être visionnées en temps réel ou visionner à distance à partir d'un ordinateur connecté au réseau (Internet, Intranet...).
- ✓ il peut être paramétré en fonction des besoins : enregistrement vidéo 24h/24 et 7j/7 ou bien enregistrement uniquement lorsque l'alarme se déclenche, prise de photos, alerte par e-mail ou SMS (sur PC ou PDA), association avec un système d'alarme ou de détection de mouvement, d'intrusion, de chaleur
- ✓ l'amélioration de la sécurité et la diminution des actes illicites dans les rues.
- ✓ Parmi ces avantages ce système nous assurent la sécurité des patients aussi par exemple les gents peuvent voir ses proches malades

Toutefois la télésurveillance présente l'inconvénient de risque d'atteinte à la vie privée et aux libertés d'où la nécessité d'une démarche éthique [6].

La réalisation de ce système est très compliquée, le coût est très élevé

I.5 Conclusion

Dans ce chapitre, on est intéressés à l'utilisation des systèmes embarqués dans le domaine de la télémédecine en premier temps on a présenté la télésurveillance et ça commence par sa définition, ses domaines d'applications, ces avantages et ses inconvénients ainsi que les différents équipements nécessaires pour sa mise en place.

Chapitre II : Arduino et la carte wifi Nodemcu

II .1 Introduction

Le système à base d'arduino nous donne le droit de rejoindre les performances de programmation à ceux de l'électronique. L'un des avantages de l'électronique programmée est qu'elle simplifie grandement les schémas électroniques et donc le coût de réalisation et moins onéreux, mais aussi la charge de travail dans la conception d'une carte électronique.

Où les choses commencent à devenir un peu délicates. En d'autres termes, le NodeMCU est presque identique à l'Arduino, sauf qu'il dispose du WiFi intégré. Il est basé sur le SoC ESP8266, qui est bien plus puissant que l'Arduino. Nous pouvons installer une carte d'extension WiFi sur l'Arduino et elle a les mêmes fonctionnalités, mais le NodeMCU fait un meilleur travail. Il nous donne également un choix de langage de programmation. Nous pouvons utiliser Lua ou APL "Arduino Programming Language" pour le programmer. Veuillez noter qu'il est recommandé aux débutants de commencer avec APL, à moins que l'implémentation n'exige le contraire. NodeMCU fournit plus de puissance de traitement, nous permettant de construire n'importe quoi, des appareils intelligents aux robots autonomes. Il est à peu près compatible avec Arduino et vendu dans la même gamme de prix.

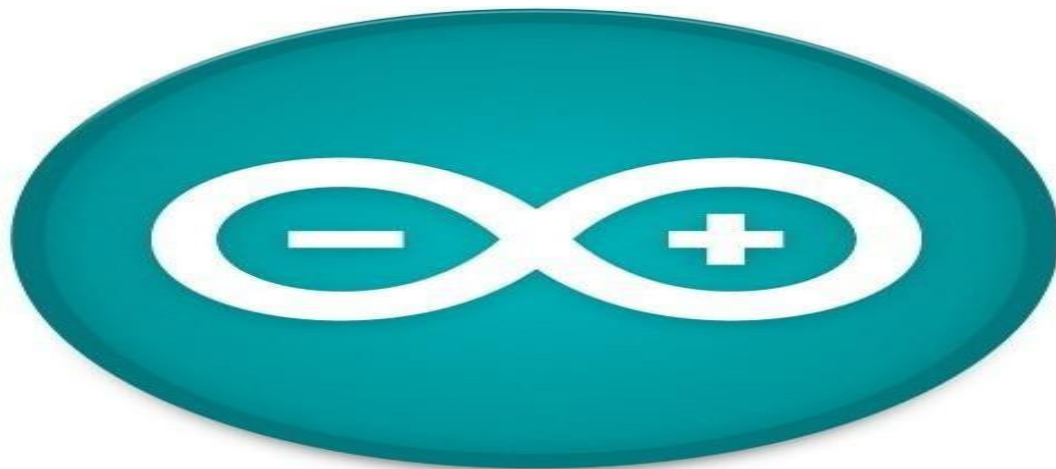


Figure.II.1. Logo d'une carte Arduino

II .1 .1 Historique

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. 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, MassimiBanzi et Nicholas Zambetti).

II .1 .2 Définition de la carte arduino

La carte Arduino est une carte de circuit imprimé en matériau libre sur laquelle se trouve un microcontrôleur programmable pour étudier et créer des signaux électriques, de façon à effectuer des tâches très varié telles que la domotique, le gouvernance d'un robot, contrôler les moteurs et les lumières de jeu, communiquer avec l'ordinateur, contrôler les appareils mobiles etc Chaque module Arduino possède un régulateur de tension +5V et un oscillateur à quartz. Pour programmer cette carte, on utilise le logiciel IDE Arduino[6]

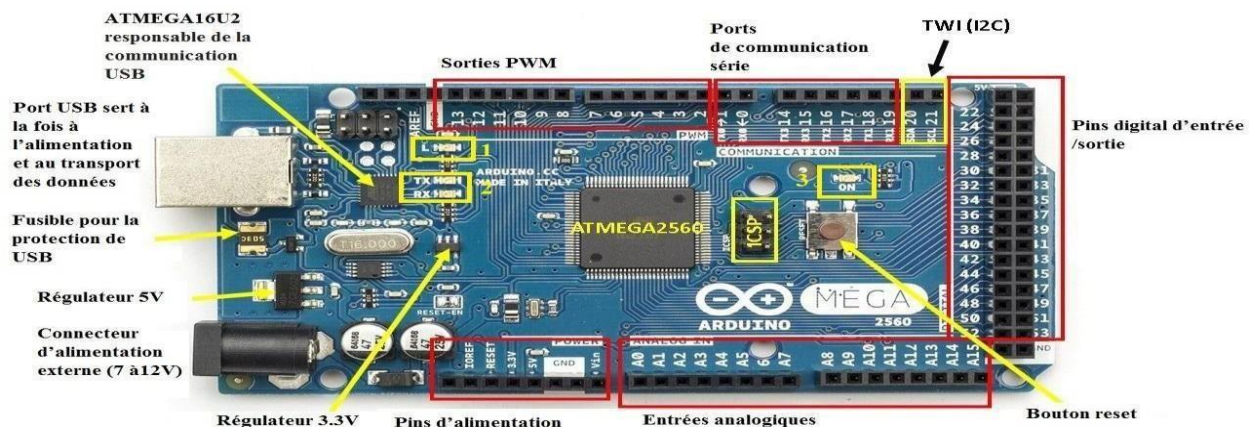


Figure II .2. Carte Arduino

II.1.3 Domaines d'application des cartes arduino

Les applications possibles réalisées grâce à Arduino sont:

- ✓ Électronique industrielle et embarquée
- ✓ contrôler des appareils domestiques
- ✓ donner une "intelligence" à un robot
- ✓ réaliser des jeux de lumières
- ✓ permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- ✓ télécommander un appareil mobile (modélisme) etc ...[7]

II.1.4 Avantages d'Arduino

Elle est largement utilisable car elle est:

- ✚ Recueillir Des offres de la programmation
- ✚ Simplifie grandement les schémas électroniques.
- ✚ Diminué le coût de la réalisation.
- ✚ La charge de travail à la conception d'une carte électronique.
- ✚ Environnement de programmation clair et simple.
- ✚ Multiplateforme : tourne sous Windows, Macintosh et Linux.
- ✚ Nombreuses bibliothèques 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)...[8]

II.1.5 Inconvénients de la carte Arduino

- ✚ Basique : pas de composants Ethernet, Bluetooth ou WIFI.
- ✚ Processeur ATMEGA lent.
- ✚ Mémoire de stockage limité.

II .1 .6 Différentes cartes Arduino

Choisir la bonne option pour votre projet n'est pas une tâche facile et vous devez réfléchir à celle que vous utiliserez. Dans cet article, nous en apprendrons davantage sur les différents types de cartes Arduino et leurs utilisations...

- ✓ Arduino UNO
- ✓ Arduino Nano
- ✓ Arduino Due
- ✓ Arduino Mega
- ✓ Arduino Leonardo
- ✓ Arduino Nodemcu [9]

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.[10]

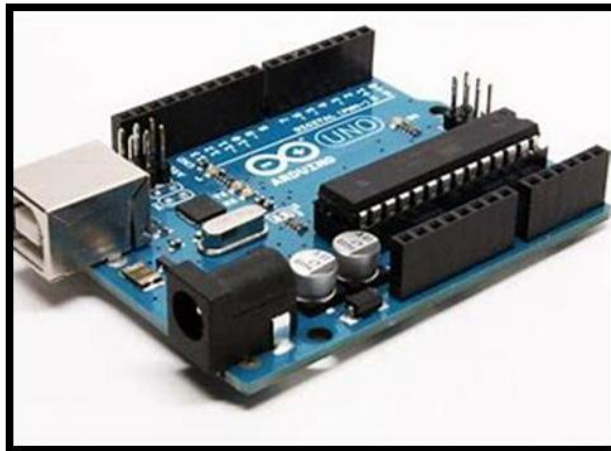


Figure. II.3. Carte Arduino UNO

Arduino Nano

La carte Arduino Nano est basée sur un ATmega328. Sa mémoire de 32 kB et ses E/S font de ce circuit un élément idéal pour les systèmes embarqués ou pour des applications nécessitant du multitâches [11]

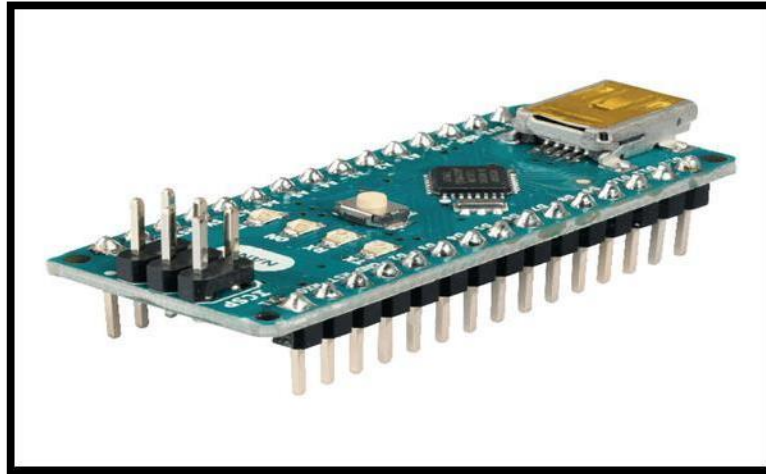


Figure II.4. Carte Arduino NANO

Arduino Due

L'Arduino Due est la première carte Arduino basée sur un microcontrôleur ARM Cortex-M3, ce qui en fait la carte Arduino la plus puissante comparativement aux autres cartes Arduino[12]

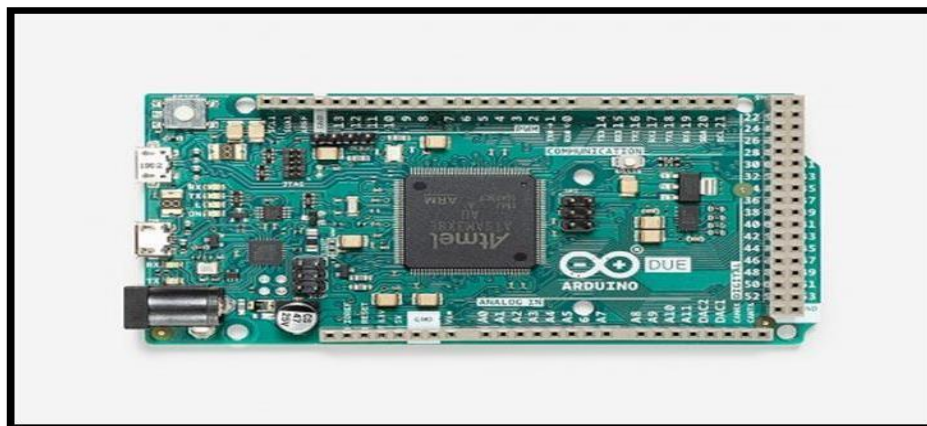


Figure II.5. Carte Arduino DUE

Arduino Mega

La carte Arduino Mega 2560 est basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs.[13]

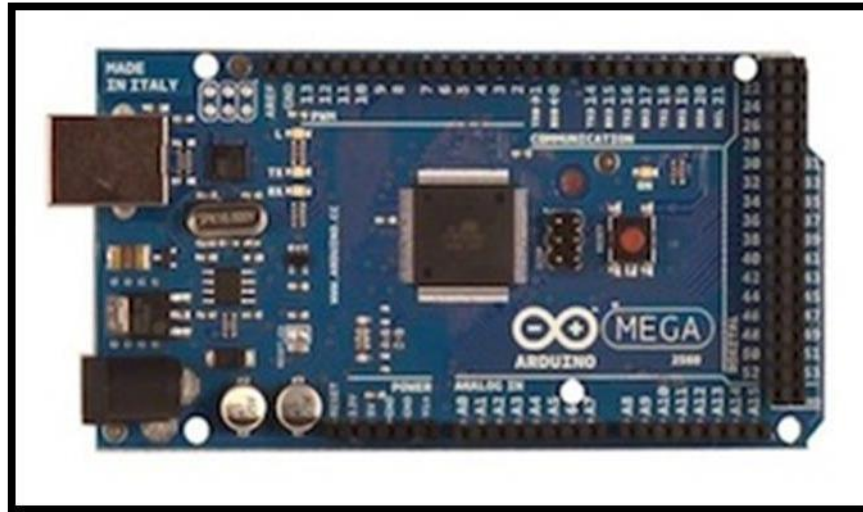


Figure II.6. Carte Arduino MEGA

Arduino Leonardo

La carte Arduino Leonardo est la dernière née de la gamme Arduino. Elle est basée sur le microcontrôleur ATmega32u4. PRODUIT EN FIN DE VIE [14]

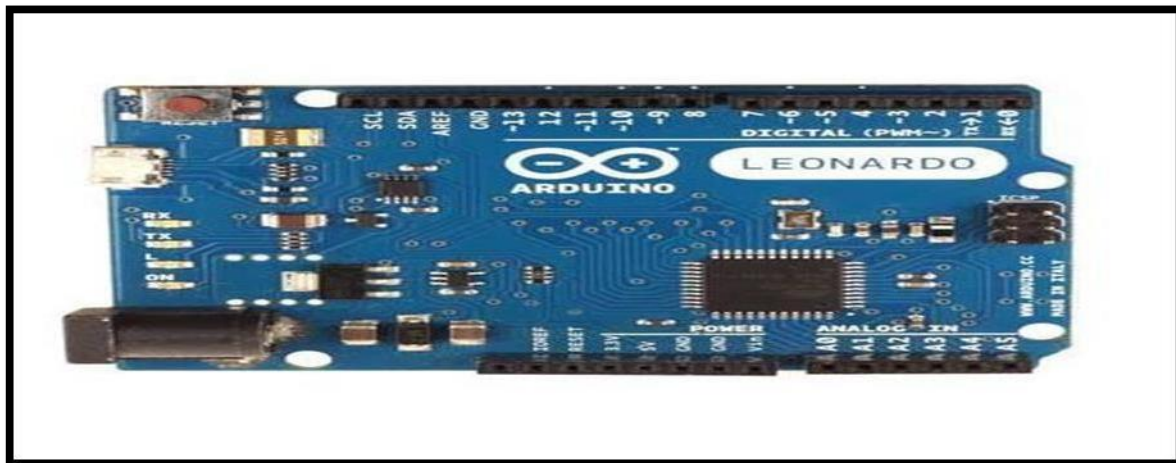


Figure II.7. Carte Arduino LEONARDO

Arduino Nodemcu

Module basé sur un ESP8266 cadencé à 80 MHz et exécutant le firmware open source NodeMCU. Cette carte se programme via l'IDE Arduino est compatible avec les scripts LUA.[15]

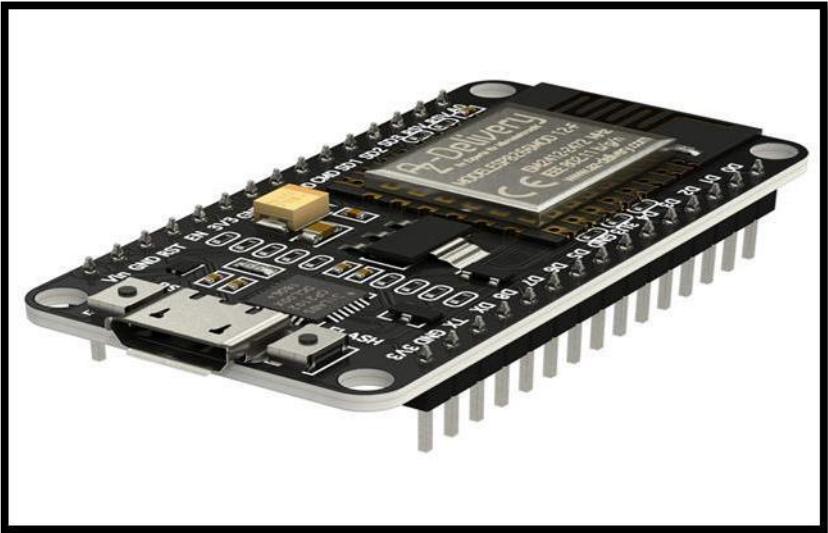


Figure II.8. Carte NODEMCU(esp8266)

Tableau comparatif des différentes cartes Arduino

| Cartes Arduino Caractéristiques | UNO R3 (classique & CMS) | UNO R3 Ethernet (classique & POE) | Leonardo | Mega 2560 | Mega ADK | DUE | Esplora | Mini | Nano | Yun (classique & POE) | Zero PRO |
|------------------------------------|--------------------------------|--|------------|------------|---------------------------------------|--|------------|------------|------------|--------------------------|--|
| Microcontrôleur | ATmega328P | ATmega328P | ATmega32u4 | ATmega2560 | ATmega2560 | AT91SAM3X8E | ATmega32u4 | ATmega328P | ATmega328P | ATmega32u4 | ATSAMD21G18 |
| Cadencement Horloge | 16 MHz | 16 MHz | 16 MHz | 16 MHz | 16 MHz | 84 MHz | 16 MHz | 16 MHz | 16 MHz | 16 MHz | 48 MHz |
| Tension d'entrée | 7 - 12V | 7 - 12V | 7 - 12V | 7 - 12V | 7 - 12V | 7 - 12V | 7 - 12V | 7 - 9V | 7 - 9V | 5V | 5V |
| Tension de fonctionnement | 5V | 5V | 5V | 5V | 5V | 3,3V | 5V | 5V | 5V | 5V | 3,3V |
| Entrée/Sortie Numérique | 14/6 | 14/4 | 20/7 | 54/15 | 54/15 | 54/12 | | 14/6 | 14/6 | 20/7 | 14/12 |
| Entrée-Sortie (PWM) Analogique | 6/0 | 6/0 | 12/0 | 16/0 | 16/0 | 12/2 (DAC) | | 8/0 | 8/0 | 12/0 | 6/1 (DAC) |
| Mémoire vive (Flash) | 32 Ko | 32 Ko | 32 Ko | 256 Ko | 256 Ko | 512 Ko | 32 Ko | 32 Ko | 32 Ko | 32 Ko | 256 Ko |
| Mémoire vive (SRAM) | 2 Ko | 2 Ko | 2,5 Ko | 8 Ko | 8 Ko | 96 Ko | 2,5 Ko | 2 Ko | 2 Ko | 2,5 Ko | 32 Ko |
| Mémoire morte (EEPROM) | 1 Ko | 1 Ko | 1 Ko | 4 Ko | 4 Ko | | 1 Ko | 1 Ko | 1 Ko | 1 Ko | 16 Ko |
| Interface USB | USB-B mâle | USB-B mâle | Micro-USB | USB-B mâle | USB-B mâle & USB-A pour Android | 2 ports micro- USB (Native et programming) | Micro-USB | | Mini-USB | Micro-USB | 2 ports micro- USB (Native et programming) |
| Port UART | 1 | 1 | 1 | 4 | 4 | 4 | | | 1 | 1 | 2 |
| Carte SD | | | | | | | | | | | |
| Ethernet | | | | | | | | | | | |
| Wi-Fi | | | | | | | | | | | |
| Dimensions | 68x53mm | 68x53mm | 68x53mm | 101x53mm | 101x53mm | 101x53mm | 165x60mm | 30x18mm | 45x18mm | 68x53mm | 68x53mm |

Tableau II.1. Caractéristiques des différentes cartes Arduino

II .2.1 Cartes Nodemcu (esp8266)

L'ESP8266 est un module Wifi permettant à une carte Arduino équipé d'une liaison série, de disposer une connectivité Wifi. Les cartes wifi basées sur le microcontrôleur ESP8266 sont programmables comme les cartes Arduino et peuvent communiquer par wifi avec d'autres appareils (ordinateurs, Smartphones, etc.).

Il existe plusieurs modèles : l'ESP-D1, l'ESP-03, l'ESP-12 etc... L'ESP8266-12E (NODEMCU) serait utilisé dans ce travail est présenté dans la figure (I.1) :

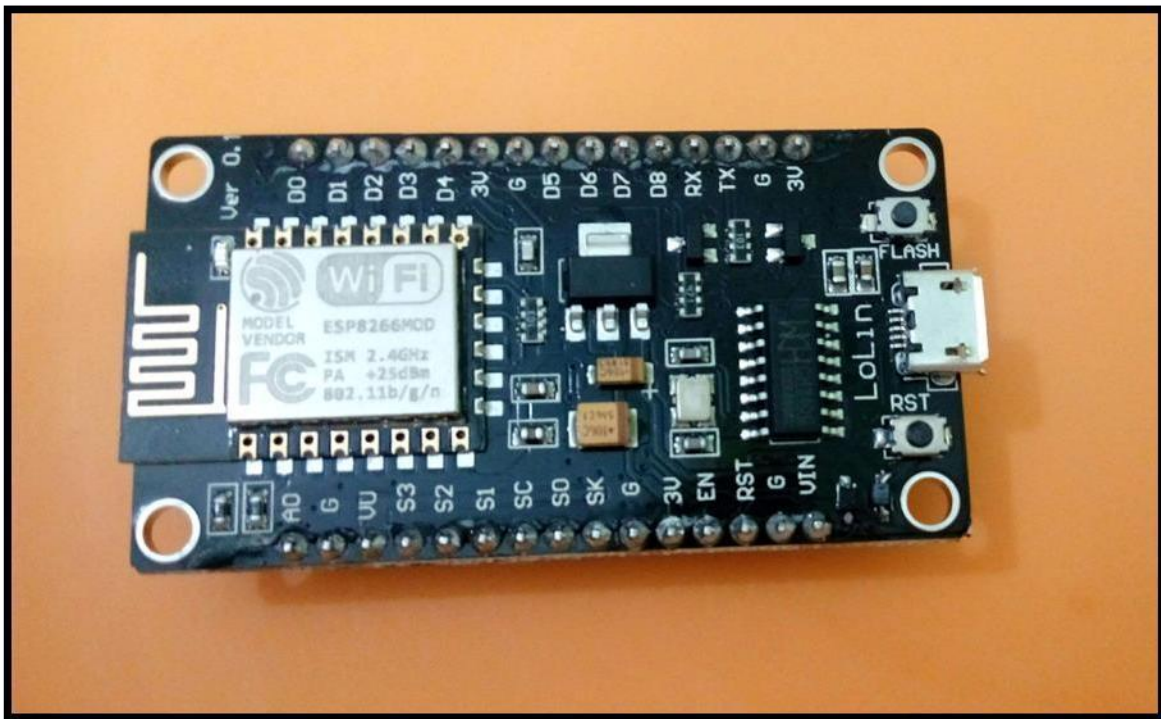


Figure II.9.NodeMCU ESP8266

II .2.2 Caractéristiques du module ESP8266

- ❖ Fonctionne dans la bande des fréquences 2,4 GHz
- ❖ Possède plusieurs options des microcontrôleurs: GPIO, I2C, ADC, SPI, PWM...
- ❖ Une mémoire RAM: 160Ko (64Ko pour le code, 96Ko pour les données)
- ❖ Une mémoire ROM: 64Ko
- ❖ Taille: 11.5mm x 11.5mm x 0.8mm.
- ❖ Alimentation: 1.9V ~ 3.6V (Non 5V!).

- ❖ Courant nécessaire de travail: 13.5mA
- ❖ Température de service: -40 à 125 °C
- ❖ Les pins I/O utilisable:8[16]

II .2.3 Principe de fonctionnement

La carte NodeMCU ESP8266 contient un module ESP-12E qui contient un microprocesseur 32 bits. Il intègre un émetteur-récepteur WiFi qui lui permet de se connecter à des réseaux existants ou de mettre en place le sien. Le WiFi est un réseau radio qui fonctionne sur les fréquences 2,45 GHz et 5 GHz.

II .2.4 Arduino et NodeMCU

| | Atmega328 | ESP8266 |
|---------------------|------------------|----------------|
| Circuit | Arduino Uno | NodeMCU |
| Tension | 5 V | 3.3 V |
| Adressage | 8 bits | 32 bits |
| Fréquence d'horloge | 16 MHz | 80 MHz |
| Mémoire flash | 32 kiB | 0 |
| Mémoire RAM | 2 kiB | 96 kiB |
| E/S numérique | 14x dont 6 PWM | 16x toutes PWM |
| Entrées analogiques | 6 (10 bits) | 1 (10 bits) |
| WiFi | non | oui |

Tableau II.2-comparaison entre un Arduino et un NodeMCU

II .2.5 Architecture interne

L'architecture interne du module, présentée sur la figure I.2 , contient un microprocesseur principal ESP8266, et des broches d'entrée / sortie pour l'utilisation avec un microcontrôleur externe.

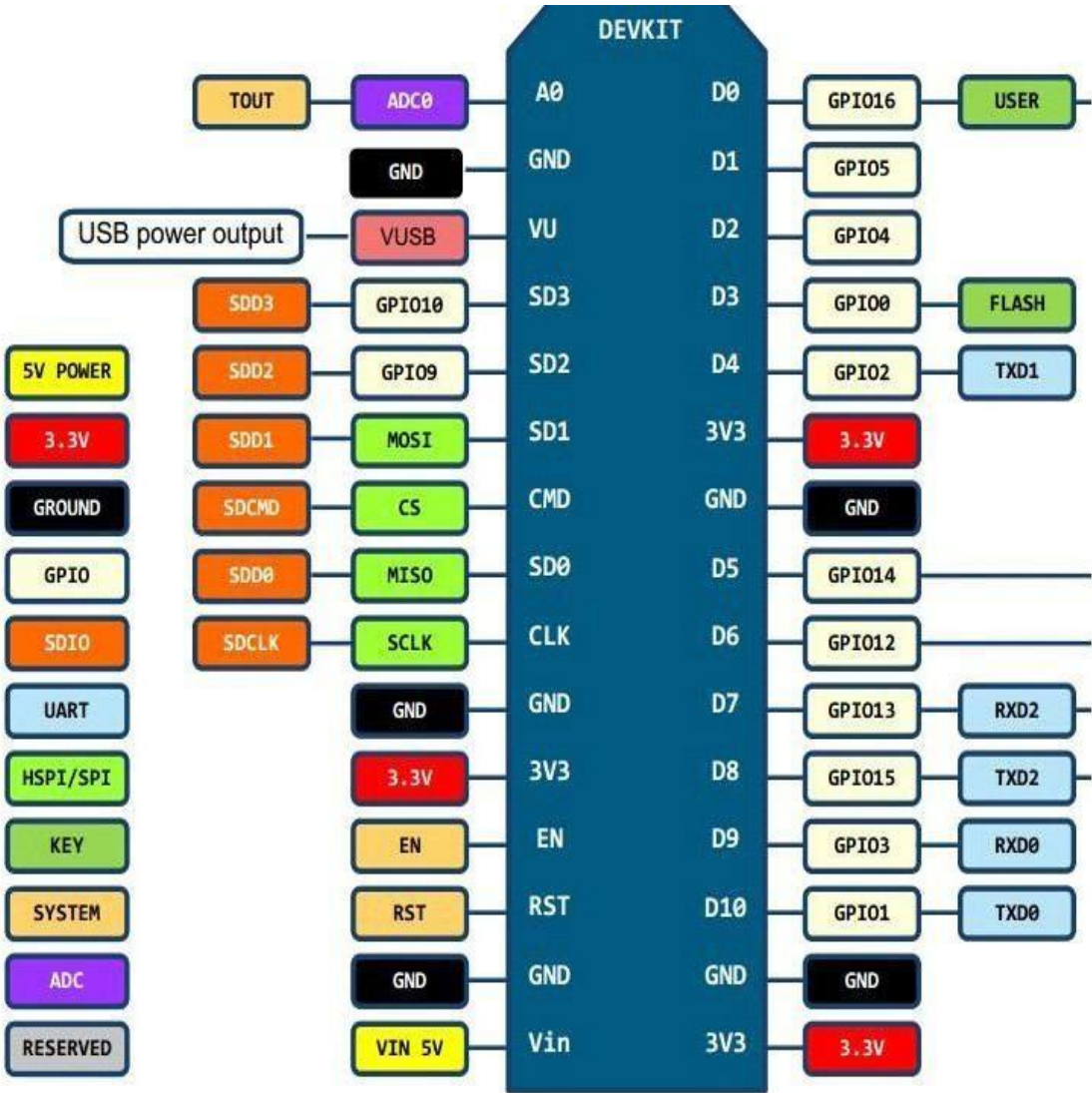


Figure II.10. L'architecture interne du module ESP8266

II .2.6 Définitions des pins

ESP8266 utilise 7 Pins parmi les 32 pins de l’ESP8266.Ces 7 pins suffisent pour que le module travaille en mode Wi-Fi (tableau 3).

| Pin | Non | Fonction | Description |
|------------|------------|-----------------|---|
| 1 | UTXD | Power | Transféré les données du module vers un microcontrôleur Externe |
| 2 | URXD | Input/output | Transfère les données d’un microcontrôleur externe vers le Module |
| 3 | CE | Input | Chip enable : il active le module de transmission RX/TX |
| 4 | GPIO 0 | Input/Output | Pin de contrôle |
| 5 | GIPO 2 | Input/output | Pin de contrôle |
| 6 | GIPO16I | Input/output | Pin de contrôle |
| 7 | VCC | Power | Alimentation 3.0V~3.6V |

Tableau II.3 Les pins de l’ESP8266

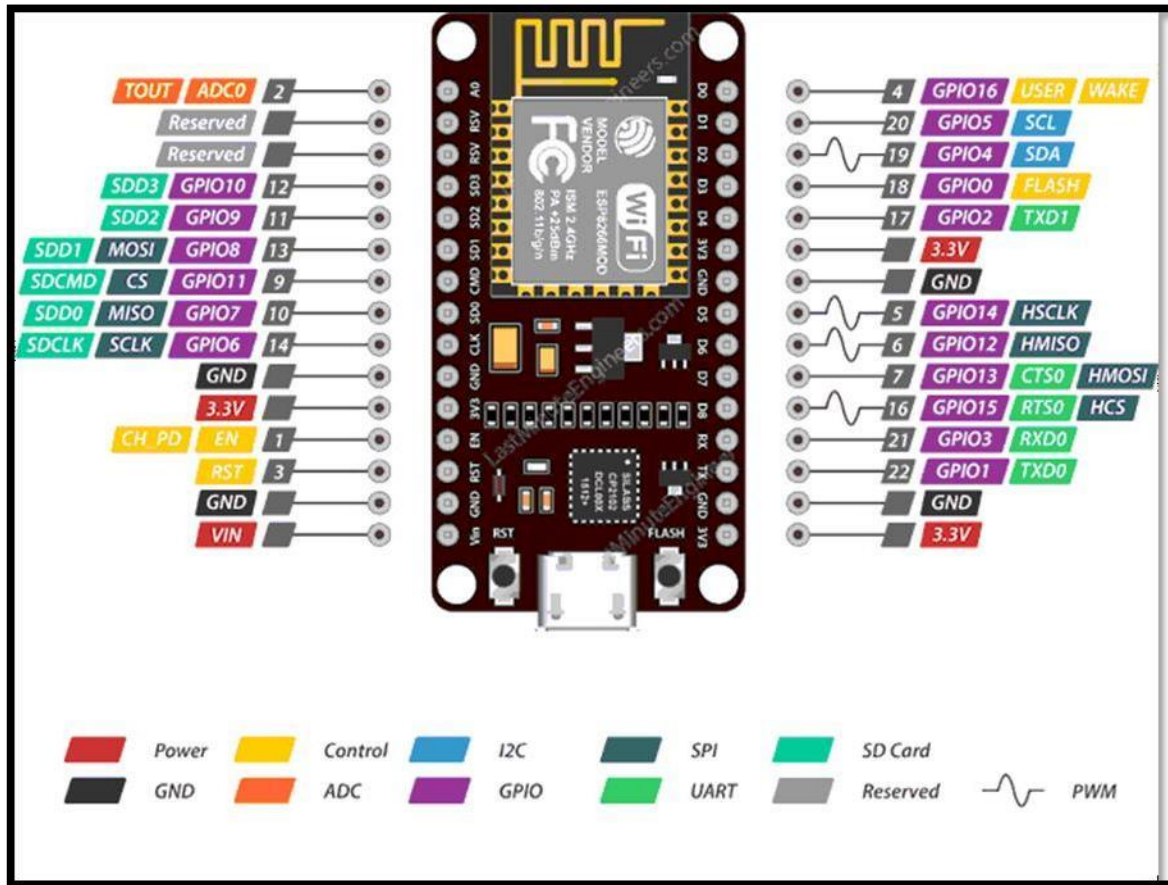


Figure II.11. Les pins de l'ESP8266

Power Pins : Il y a quatre broches d'alimentation à savoir. Une broche VIN et trois broches 3,3 V. La broche VIN peut être utilisée pour alimenter directement l'ESP8266 et ses périphériques, si vous disposez d'une source de tension régulée 5V. Les broches 3,3 V sont la sortie d'un régulateur de tension embarqué. Ces broches peuvent être utilisées pour alimenter des composants externes.

GND: est une broche de masse de la carte de développement ESP8266 NodeMCU

I2C Pins: sont utilisées pour connecter toutes sortes de capteurs et de périphériques I2C dans votre projet. Le maître I2C et l'esclave I2C sont pris en charge. La fonctionnalité d'interface I2C peut être réalisée par programmation et la fréquence d'horloge est de 100 kHz au maximum. Il convient de noter que la fréquence d'horloge I2C doit être supérieure à la fréquence d'horloge la plus lente de l'appareil esclave

GPIO Pins: ESP8266 NodeMCU dispose de 17 broches GPIO qui peuvent être affectées à diverses fonctions telles que I2C, I2S, UART, PWM, télécommande infrarouge, lumière LED et bouton par programmation. Chaque GPIO à activation numérique peut être configuré en mode pull-up ou pull-down interne, ou réglé sur haute impédance. Lorsqu'il est configuré en tant qu'entrée, il peut également être réglé sur front-trigger ou level-trigger pour générer des interruptions CPU.

ADC Channel : Le NodeMCU est intégré à un ADC SAR de précision 10 bits. Les deux fonctions peuvent être implémentées en utilisant ADC viz. Test de la tension d'alimentation de la broche VDD3P3 et test de la tension d'entrée de la broche TOUT. Cependant, ils ne peuvent pas être mis en œuvre en même temps.

UART Pins: ESP8266 NodeMCU dispose de 2 interfaces UART, à savoir UART0 et UART1, qui fournissent une communication asynchrone (RS232 et RS485), et peuvent communiquer jusqu'à 4,5 Mbps. UART0 (broches TXD0, RXD0, RST0 et CTS0) peut être utilisé pour la communication. Il prend en charge le contrôle des fluides. Cependant, UART1 (broche TXD1) ne comporte que le signal de transmission de données, il est donc généralement utilisé pour l'impression du journal.

SPI Pins: L'ESP8266 dispose de deux SPI (SPI et HSPI) en modes esclave et maître. Ces SPI prennent également en charge les fonctionnalités SPI générales suivantes:

- 4 modes de synchronisation du transfert au format SPI
- Jusqu'à 80 MHz et les horloges divisées de 80 MHz
- Jusqu'à 64 octets FIFO

SDIO Pins: L'ESP8266 dispose d'une interface d'entrée / sortie numérique sécurisée (SDIO) qui est utilisée pour interfacier directement les cartes SD. SDIO v1.1 4 bits 25 MHz et SDIO v2.0 4 bits 50 MHz sont pris en charge.

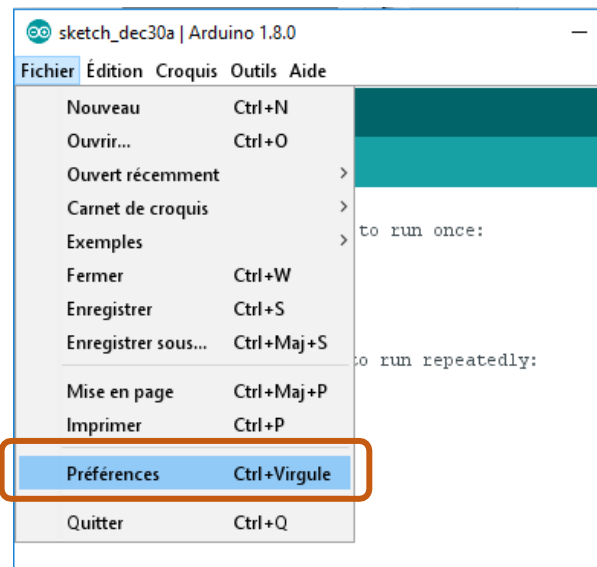
PWM Pins: La carte dispose de 4 canaux de modulation de largeur d'impulsion (PWM). La sortie PWM peut être implémentée par programme et utilisée pour piloter des moteurs numériques et des LED. La gamme de fréquences PWM est réglable de 1000 μ s à 10000 μ s, c'est-à-dire entre 100 Hz et 1 kHz.

Control Pins: sont utilisés pour contrôler l'ESP8266. Ces broches comprennent la broche d'activation de la puce (EN), la broche de réinitialisation (RST) et la broche WAKE.

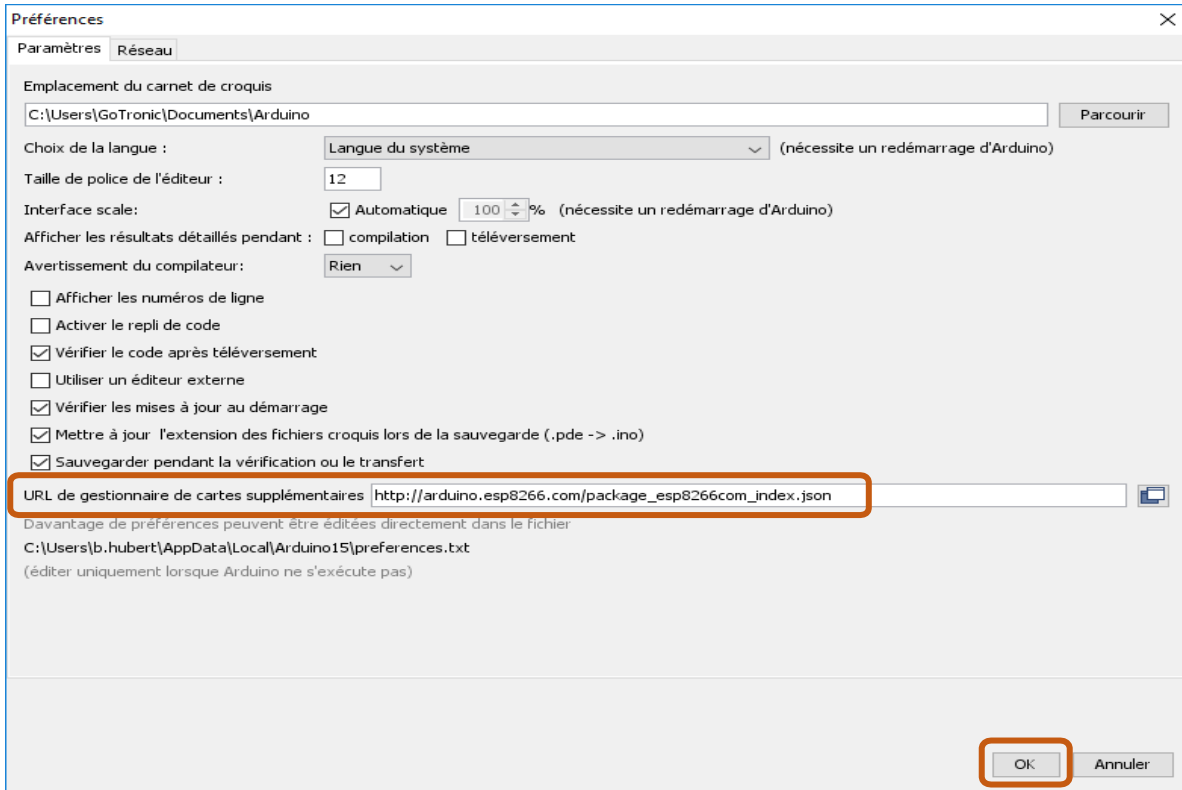
- Broche EN - La puce ESP8266 est activée lorsque la broche en est tirée HAUT. Lorsqu'elle est tirée BAS, la puce fonctionne à une puissance minimale.
- Broche RST - La broche RST est utilisée pour réinitialiser la puce ESP8266.
- Broche WAKE - La broche Wake est utilisée pour réveiller la puce du sommeil profond. :[13]

II .2.7 Etapes d'installation du module sur l'IDE

Ouvrez l'IDE Arduino, allez dans *Fichier* → *Préférences*

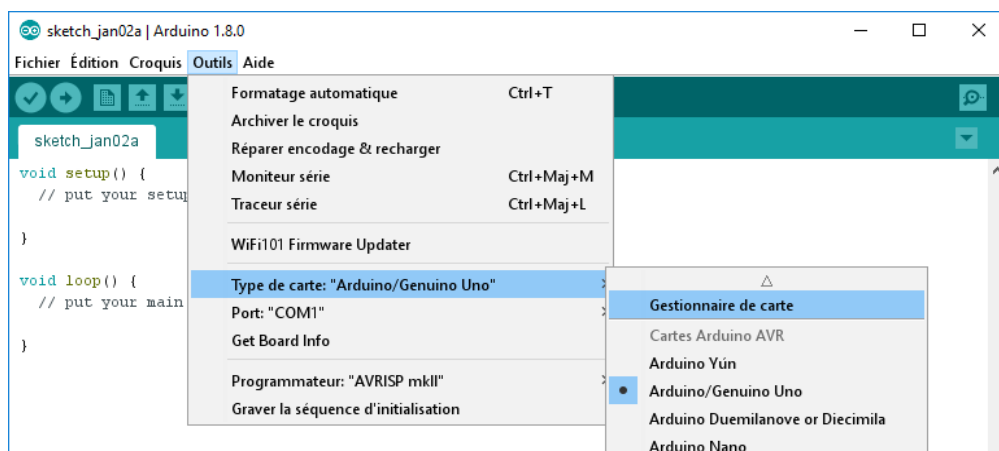


et entrez l'URL « http://arduino.esp8266.com/stable/package_esp8266com_index.json » dans le champ *URL de gestionnaire de cartes supplémentaires*



Pour installer la carte, procédez comme ci-dessous :

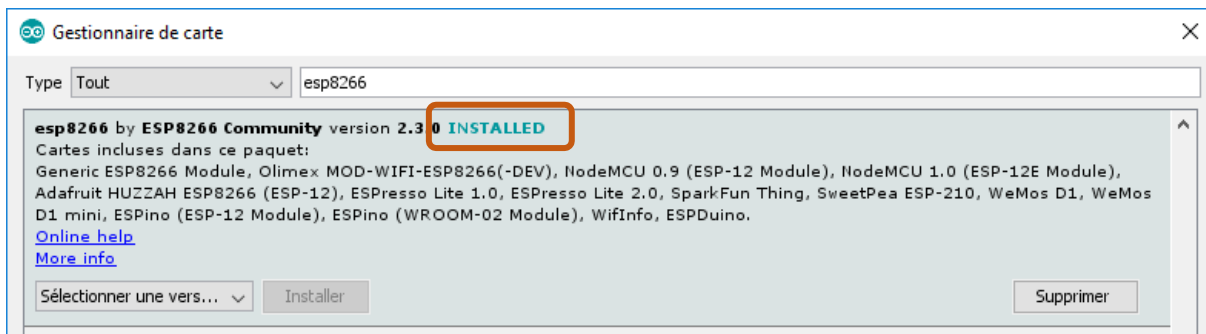
Outils → Type de carte → Gestionnaire de carte



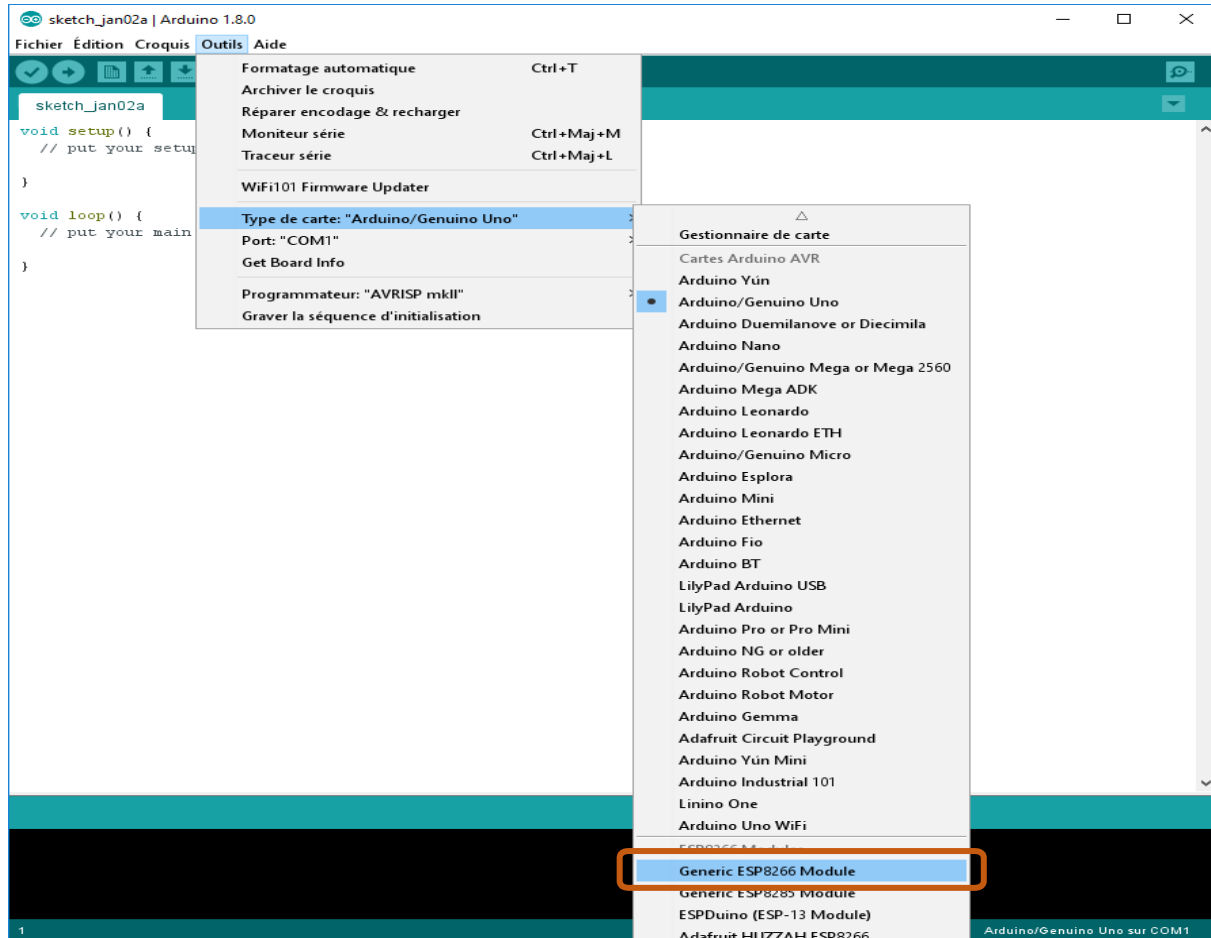
Dans le gestionnaire de cartes, recherchez « ESP8266 » et installez `esp8266` :



La carte a bien été installée :

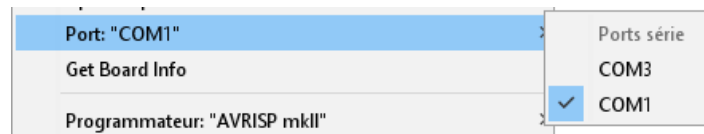


Le programme Arduino doit être paramétré pour le module ESP8266 :



Le module ESP8266 est maintenant prêt à être utilisé.

Attention : veillez à sélectionner le bon port (menu *Outils* → *Port*).



Voilà c'est fini pour cette partie, maintenant vous utiliserez votre module NodeMCU « comme un Arduino », branchement sur le port USB et programmation depuis l'IDE Arduino.

II .2.8 Applications de la carte Nodemcu

- ❖ Utilisez une interface web pour piloter des moteurs ou gérer des capteurs branchés à votre microcontrôleur ESP8266

- ❖ Prototypage d'appareils IoT
- ❖ Applications fonctionnant sur batterie à faible puissance
- ❖ Projets de réseau
- ❖ Projets nécessitant plusieurs interfaces E / S avec fonctionnalités Wi-Fi et Bluetooth

II .2.9 Avantage de Nodemcu

✓ Matériel Open source et extensible: Les NodeMCU les plus trouvés sur le marché viennent de chez Amica, DOIT ou Lolin & D1 mini /Wemos parmi bien d'autres et ils ne ressemblent pas obligatoirement à l'original.

✓ Bas prix : Les coûts diffèrent d'un modèle à l'autre dû à leurs spécifications techniques mais ils restent néanmoins assez bas.

✓ Facilité de programmation : Programmer en Lua via l'IDE Arduino avec un grand nombre de bibliothèques déjà disponibles pour créer vos firmwares.

✓ Wifi inclus.

✓ Port micro USB inclus.

II .3 Définition d'un capteur

Un **capteur** est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable exemple : une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille.

II .3.1 Capteur max30100

Le MAX30100 est une solution intégrée de capteur d'oxymétrie de pouls et de moniteur de fréquence cardiaque. Il combine deux DEL, photodétecteur, des optiques optimisées et un traitement de signal analogique à faible bruit pour détecter les signaux d'oxymétrie de pouls et de fréquence cardiaque.

Le MAX30100 fonctionne à partir d'alimentations 1.8V et 3.3V et peut être éteint grâce à un logiciel courant, permettant à l'alimentation d'être connectée à tout moment. Elle est bien appliquée aux dispositifs portables, aux dispositifs d'aide à la remise en forme et aux dispositifs de surveillance médicale.

II .3.2Caractéristiques

Les principaux caractéristiques de ce capteur sont :

- MAX30100 intégré oxymétrie de pouls et solution de capteur de fréquence cardiaque.
- LED intégrées, capteur photo et frontal analogique haute performance.
- Le fonctionnement à ultra basse consommation augmente la durée de vie de la batterie pour les appareils portables.
- La fonctionnalité avancée améliore les performances de mesure.
- Le SNR élevé fournit une résilience robuste aux artefacts de mouvement.
- Taux d'échantillonnage programmable et courant LED pour des économies d'énergie.
- Courant d'arrêt ultra-faible (0,7 μ A, typ).
- Annulation de la lumière ambiante intégrée.
- Haute capacité de taux d'échantillonnage.
- Capacité de sortie de données rapide.
- Il s'agit d'une solution intégrée de capteur d'oxymétrie de pouls et de moniteur de fréquence cardiaque.

II .3.3 Applications

- Appareils portables
- Appareils pour assistant de fitness
- Appareils de surveillance médicale

II .3.4 Capteur MAX30205

Le capteur de température MAX30205 mesure avec précision la température et fournit une sortie d'alarme/interruption/arrêt de surchauffe.

Cet appareil convertit les mesures de température en forme numérique à l'aide d'un convertisseur analogique-numérique (ADC) haute résolution sigma-delta. La précision est conforme aux spécifications de thermométrie clinique de l'astm E1112 une fois soudé sur le PCB final.

La Communication se fait par une interface série I2C-compatible, 2 fils. L'interface série I2C accepte les commandes d'octets d'écriture standard, d'octets de

lecture, d'envoi d'octets et de réception d'octets pour lire les données de température et configurer le comportement de la sortie d'arrêt de surchauffe opendrain. Le MAX30205 dispose de trois lignes de sélection d'adresse avec un total de 32 adresses disponibles. Le capteur a une plage de tension d'alimentation de 2.7V à 3.3V, faible 600 & micro; Un courant d'alimentation, et une interface I2C-compatible protégée par verrouillage qui le rend idéal pour le fitness portable et les applications médicales. Cet appareil est disponible dans un paquet TDFN à 8 broches et fonctionne sur la plage de température 0NC à + 50NC.

II .3.5 Caractéristiques du capteur MAX30205

- Le fonctionnement de haute précision et de basse tension aide les designers à répondre aux erreurs et aux économies en énergie
- Précision de 0.1 °C (37 °C à 39 °C)
- Résolution de température de 16 bits (0.00390625 °C)
- Plage de tension d'alimentation de 2.7V à 3.3V
- Les Modes One-Shot et stop aident à réduire la consommation d'énergie
- 600 µa (typ) courant d'alimentation en fonctionnement
- Les fonctions numériques facilitent l'intégration dans n'importe quel système
- Le délai sélectionnable empêche le verrouillage du Bus
- La sortie OS séparée à Drain ouvert fonctionne comme sortie d'interruption ou de comparateur/Thermostat [14]

II .3.6 Les application du capteur MAX30205

- Forme physique
- Médical
- Ressources connexes

II .4 Conclusion

Dans cette partie on a pris en considération la présentation d'une carte Arduino et aussi on a parlé de la carte Nodemcu qui est la base de notre projet, et le résultat sera présenté dans le chapitre suivant qui nous montre clairement notre propre module réalisé.

Chapitre III : Conception et Réalisation

III Introduction

Dans ce chapitre nous allons voir d'une manière détaillée les étapes correspondantes de notre réalisation et aussi schématiser notre solution en expliquant sa conception et son principe de fonctionnement.

III .1.Le principe de fonctionnement de notre système

Notre projet a pour but de créer un système de télésurveillance à base de nodemcu qui regroupe plusieurs capteurs pour collecter et transmettre Pour une bonne surveillance médicale d'une personne dans le but de prévenir une insuffisance cardiaque, autrement dit, réaliser une plateforme connectée grâce à des périphériques moins chers par rapport au prix existant afin d'obtenir un système avec un budget convenable et assez intelligent.

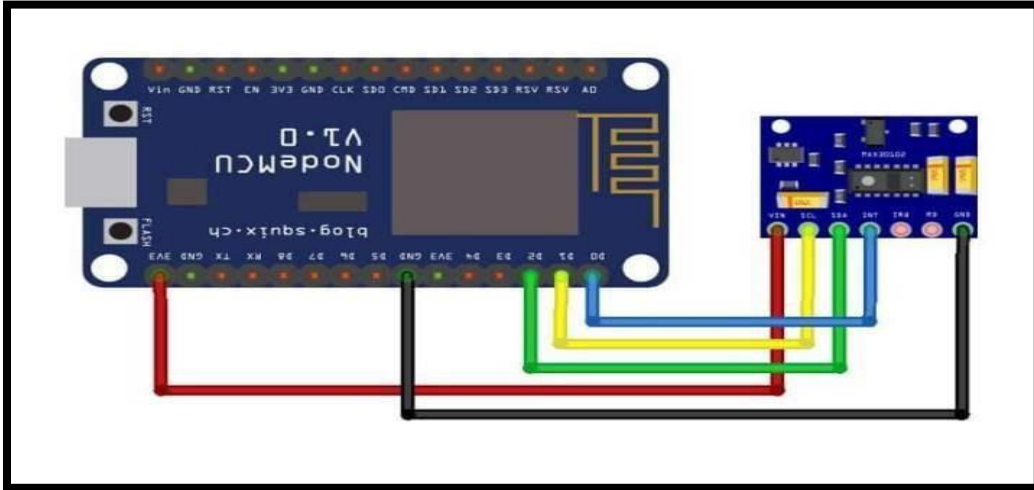
Le Nodemcu va périodiquement prendre les mesures des différents capteurs, les traite et les exécute a l'aide du programme stuqué dans le microcontrôleur.

Ce dernier va transmettre toutes les mesures collectées vers le site web via un réseau internet en temps réel; ces valeurs seront affichées sur l'interface de site web , le médecin aura toute la possibilité d'interpréter et diagnostiquer les résultats a distance.

notre système est constitué de :

- ✚ D'un capteur de rythme cardiaque et saturation on oxygène MAX30100 pour la mesure de la fréquence cardiaque et aussi pour simuler l'essoufflement et la saturation d'oxygène
- ✚ D'un capteur de température MAX30205 qui mesure à la fois la température corporelle et la température de l'environnement ou se trouve la personne concernée
- ✚ D'une carte wifi nodemcu ESP8266
- ✚ D'un OLED « Organic Light-Emitting Diode »

III .1.1 Branchement avec Nodemcu



**Figure III.1. Branchement du capteur max30100
avec Nodemcu**

III .1.2 Connexion de capteur MAX30100 (montage)

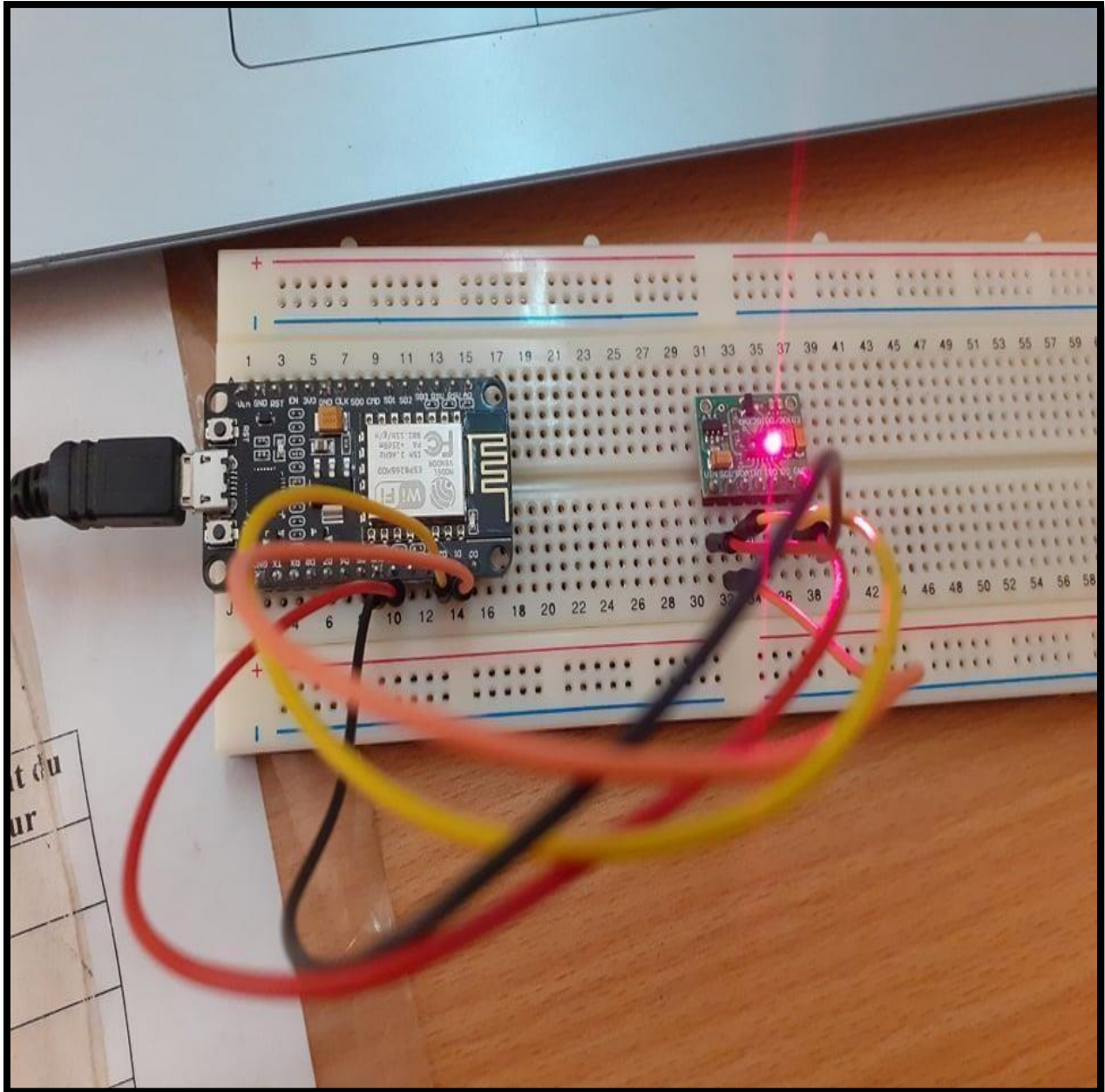


Figure III.2. Capteur max30100 avec la carte Nodemcu (esp8266)

III .1.3 CODE DE PROGRAMME

```
#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <Wire.h>

#include "MAX30100_PulseOximeter.h"

#define REPORTING_PERIOD_MS 1000

/*Put your SSID & Password*/

const char* ssid = "AndroidGH"; // Enter SSID here

const char* password = "hamza1986"; //Enter Password here

// PulseOximeter is the higher level interface to the sensor

// it offers:

// * beat detection reporting

// * heart rate calculation

// * SpO2 (oxidation level) calculation

PulseOximeter pox;

uint32_t tsLastReport = 0;

// Callback (registered below) fired when a pulse is detected

void onBeatDetected()

{

    Serial.println("Beat!");

}
```

```
ESP8266WebServer server(80);

float HeartRate;

float SpO2;

void setup()
{
  Serial.begin(115200);
  delay(100);
  Serial.println("Connecting to ");
  Serial.println(ssid);

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());
  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
```

```
server.begin();

Serial.println("HTTP server started");

Serial.print("Initializing pulse oximeter..");

// Initialize the PulseOximeter instance
// Failures are generally due to an improper I2C wiring, missing power supply
//orwrongtargetchip
if (!pox.begin()) {
    Serial.println("FAILED");
    for(;;);
} else {
    Serial.println("SUCCESS");
}

// The default current for the IR LED is 50mA and it could be changed
// by uncommenting the following line. Check MAX30100_Registers.h for all the
// available options.
// pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

// Register a callback for the beat detection
pox.setOnBeatDetectedCallback(onBeatDetected);
}
```

```
void loop()
{
  // Make sure to call update as fast as possible

  pox.update();

  HeartRate =pox.getHeartRate();// Gets the values of the Heart rate
  SpO2 = pox.getSpO2();// Gets the values of the Spo2

  // Asynchronously dump heart rate and oxidation levels to the serial
  // For both, a value of 0 means "invalid"
  if(millis()-tsLastReport>REPORTING_PERIOD_MS){
    Serial.print("Heart rate:");
    //Serial.print(pox.getHeartRate());
    Serial.print(HeartRate);
    Serial.print("bpm / SpO2:");
    //Serial.print(pox.getSpO2());
    Serial.print(SpO2);
    Serial.println("%");

    tsLastReport = millis();
  }
  server.handleClient();
}
```

```
void handle_OnConnect() {
    pox.update();
    HeartRate = pox.getHeartRate(); // Gets the values of the Heart rate
    SpO2 = pox.getSpO2(); // Gets the values of the Spo2
    server.send(200, "text/html", SendHTML(HeartRate, SpO2));
}

void handle_NotFound(){
    server.send(404, "text/plain", "Not found");
}

String SendHTML(float HearRateStat, float So2Stat){
    String ptr = "<!DOCTYPE html> <html>\n";

    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";

    ptr += "<title>ESP8266 Weather Report</title>\n";

    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";

    ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";

    ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";

    ptr += "</style>\n";

    ptr += "</head>\n";

    ptr += "<body>\n";

    ptr += "<div id=\"webpage\">\n";

    ptr += "<h1>ESP8266 NodeMCU Health Report</h1>\n";
```

```

ptr +="<p>Heart Rate: ";

ptr +=(int)HearRateStat;

ptr += "BPM</p>";

ptr +="<p>So2 : ";

ptr +=(int)So2Stat;

ptr +="%"</p>";

ptr +="</div>\n";

ptr +="</body>\n";

ptr +="</html>\n";

return ptr;

}

```

Après la compilation sur IDE, le moniteur série affiche les résultats des battements de cœur de la personne

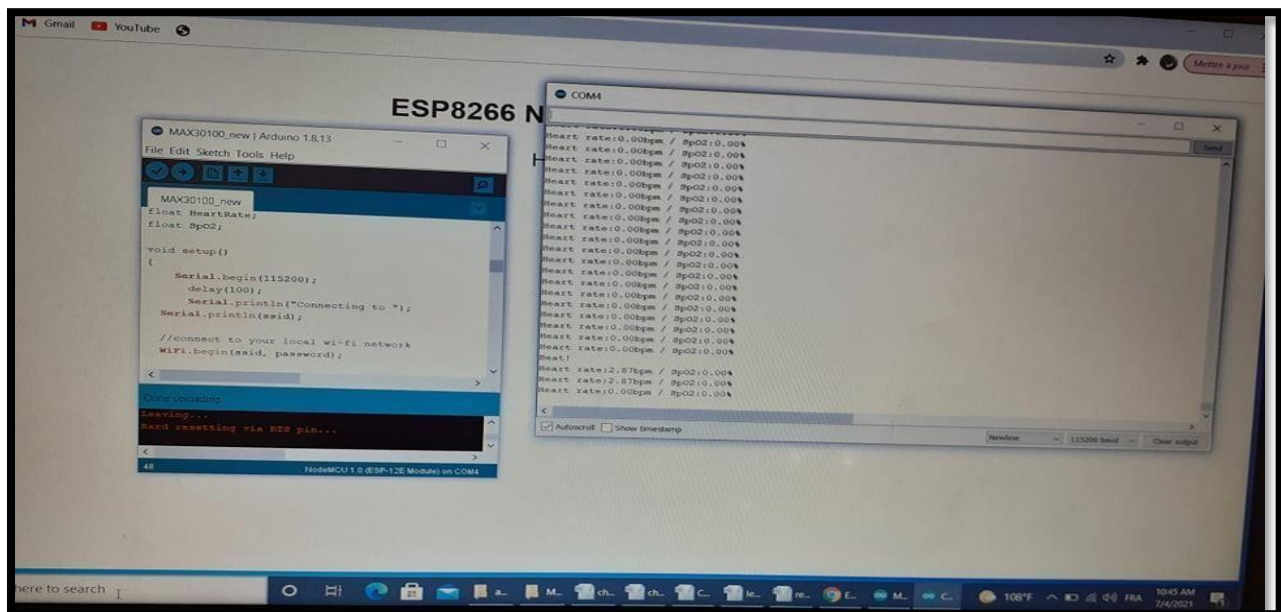


Figure III.3.Résultat du MAX30100 sur le moniteur série

III.1.4 Branchement avec Nodemcu

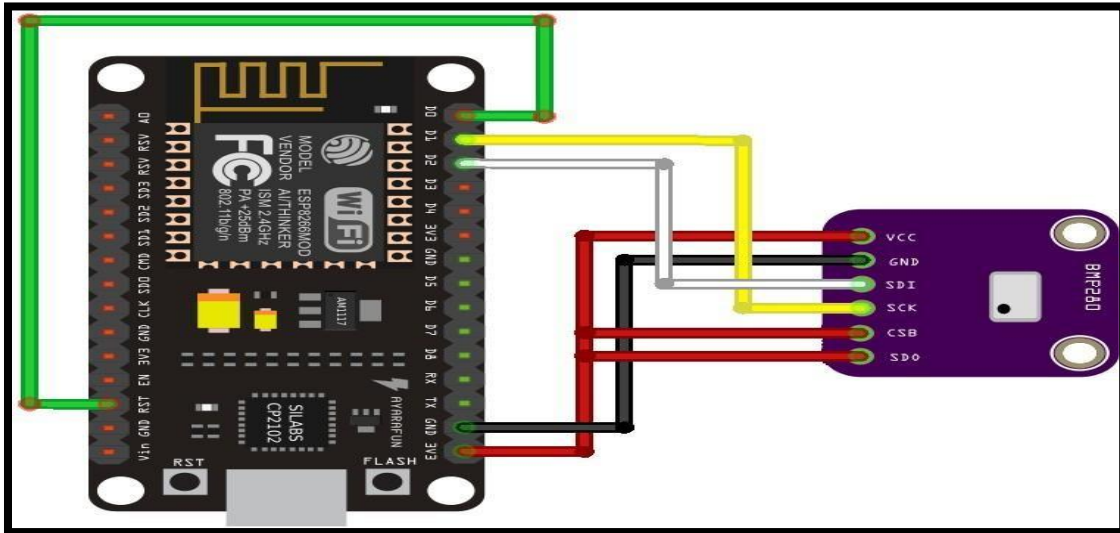


Figure III.4. Branchement du capteur max30205 avec nodemcu

III.1.5 connexion de capteur MAX30205 (montage)

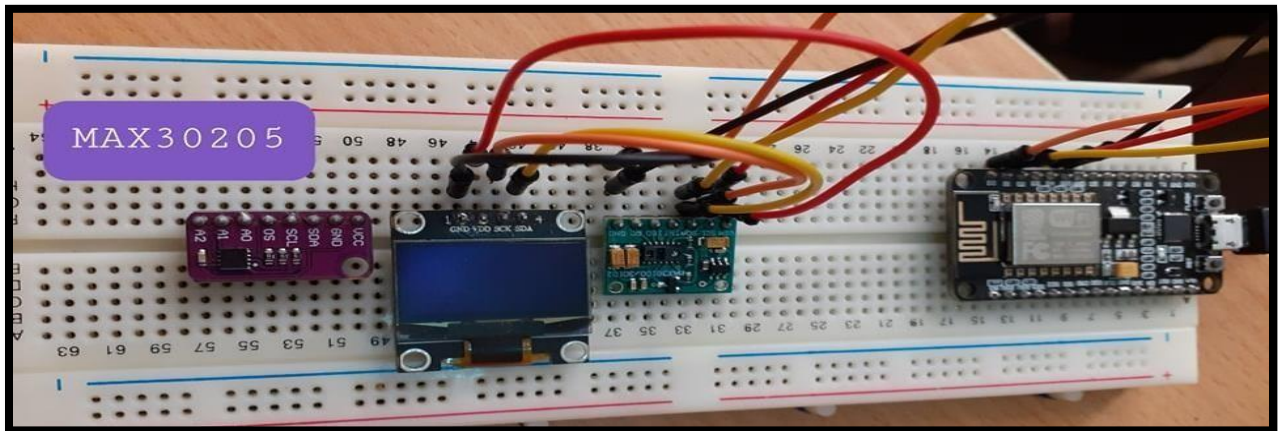


Figure III.5. Connexion du capteur MAX30205 avec Nodemcu

III .1.6 Code de programme

```
#include <Wire.h>

#include "ClosedCube_MAX30205.h"

ClosedCube_MAX30205 max30205;

void setup()
{
    Serial.begin(9600);
    Serial.println("ClosedCube MAX30205 Arduino Demo");

    max30205.begin(0x48);
}

void loop()
{
    Serial.print("T=");
    Serial.print(max30205.readTemperature());
    Serial.println("C");
    delay(300);
}
```

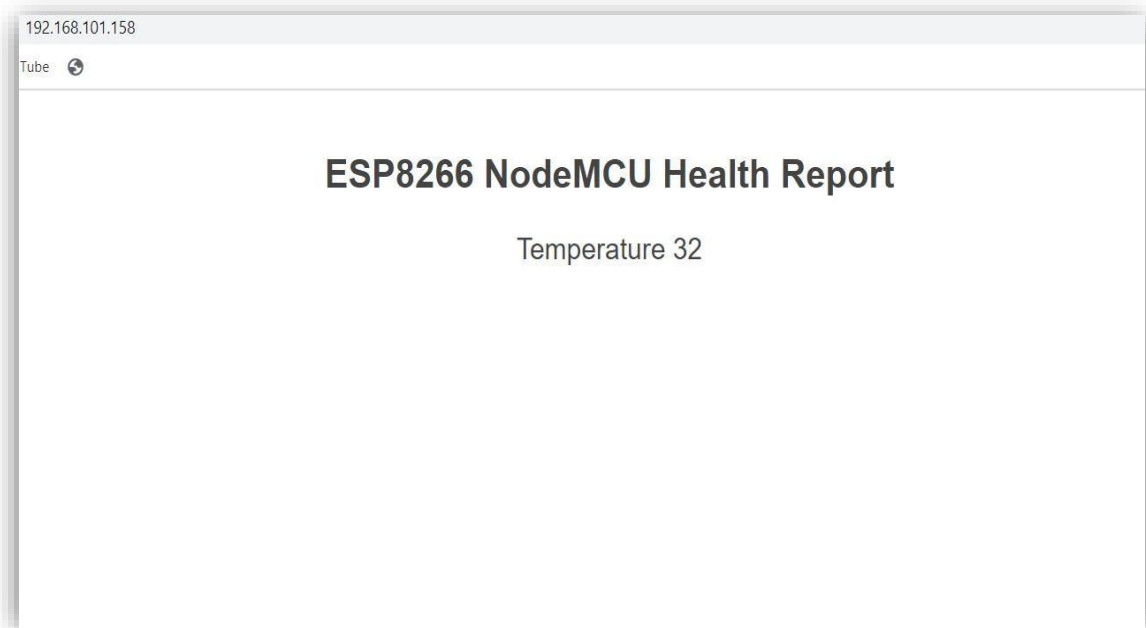


Figure III.6. Résultat de température sur moniteur série.

III .2.1 OLED (organic light emitting Diode)

Le terme OLED signifie "Diode électroluminescente organique" il utilise la même technologie que celle utilisée dans la plupart de nos téléviseurs mais a moins de pixels par rapport à eux. C'est vraiment amusant d'avoir ces modules d'affichage cool à interfacier avec les microcontrôleurs car cela rendra nos projets cool.

Il existe de nombreux modules d'affichage OLED disponibles sur le marché, chacun avec sa propre classification. Donc, avant d'en acheter un, assurez-vous que celui-ci conviendrait le mieux à notre projet. Les types les plus couramment utilisés sont classés ci-dessous :

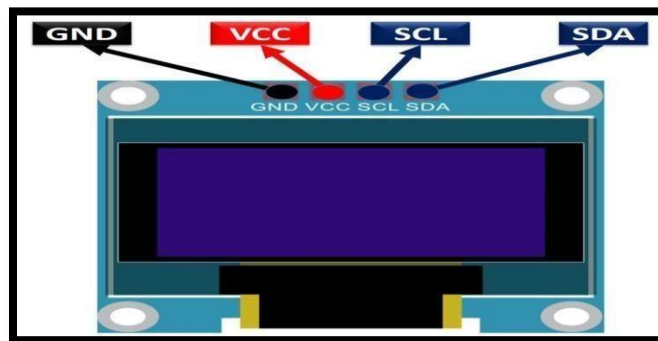


Figure III.7.Module OLED

Dans notre projet, nous avons utilisé l'écran OLED monochrome à 4 broches SSD1306. Cet écran peut fonctionner qu'avec le mode I2C.

III .2.2 Connexion avec nodemcu

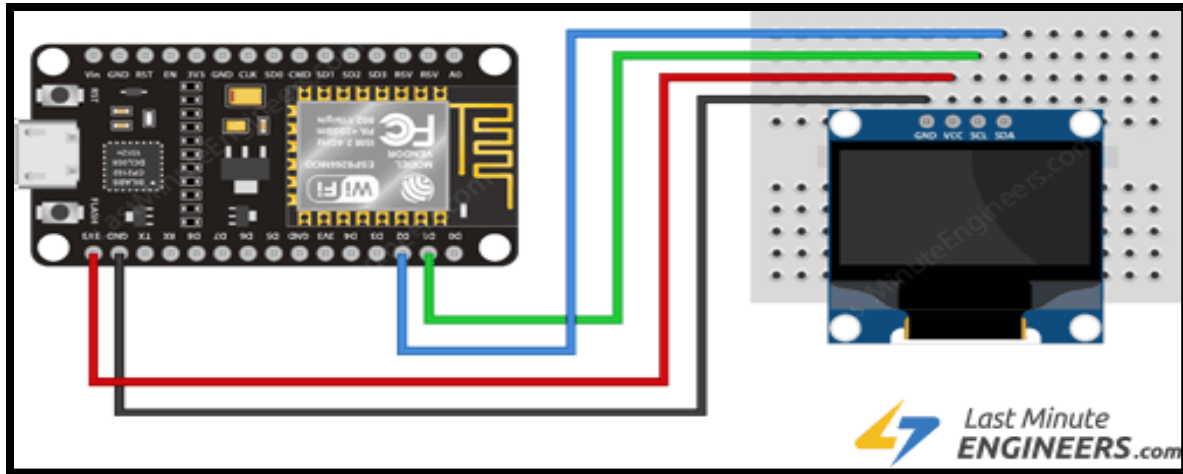


Figure III.8. Branchement OLED avec Nodemcu(esp8266)

III .2.3 Connexion des pins de la carte Nodemcu avec OLED

| SSD1306 OLED Pins | NODEMCU(esp8266) Pins |
|-------------------|-----------------------|
| VCC | 3.3V |
| GND | GND |
| D0 (SCL) | D5 |
| D1 (SDA) | D7 |

Tableau III.1. Les pins de la configuration

NODEMCU (esp8266) avec OLED

III .2.4 Code de programmation avec l'IDE

```
#include <Wire.h>

#include "MAX30100_PulseOximeter.h"

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#define ENABLE_MAX30100 1

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 //64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:   A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO: 2(SDA), 3(SCL), ...

#define OLED_RESET  -1 // 4 // Reset pin # (or -1 if sharing Arduino reset pin)

#define SCREEN_ADDRESS 0x3c ///< See datasheet for Address; 0x3D for 128x64, 0x3C for
128x32

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#if ENABLE_MAX30100

#define REPORTING_PERIOD_MS  5000

// PulseOximeter is the higher level interface to the sensor
// it offers:
// * beat detection reporting
// * heart rate calculation
// * SpO2 (oxidation level) calculation

PulseOximeter pox;

#endif
```

```
uint32_t tsLastReport = 0;

int xPos = 0;

// Callback (registered below) fired when a pulse is detected

void onBeatDetected()

{

  Serial.println("Beat!");

  heart_beat(&xPos);

}

void setup()

{

  Serial.begin(115200);

  Serial.println("SSD1306 128x64 OLED TEST");

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally

  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {

    Serial.println(F("SSD1306 allocation failed"));

    for (;;) // Don't proceed, loop forever

  }

  // Show initial display buffer contents on the screen --

  // the library initializes this with an Adafruit splash screen.

  //display.display();

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.setCursor(20, 18);

  // Display static text
```

```
display.print("Pulse OxiMeter");

int temp1 = 0;

int temp2 = 40;

int temp3 = 80;

heart_beat(&temp1);

heart_beat(&temp2);

heart_beat(&temp3);

xPos = 0;

display.display();

delay(2000); // Pause for 2 seconds

display.cp437(true);

display.clearDisplay();

Serial.print("Initializing pulse oximeter..");

#if ENABLE_MAX30100

// Initialize the PulseOximeter instance

// Failures are generally due to an improper I2C wiring, missing power supply

// or wrong target chip

if (!pox.begin()){

    Serial.println("FAILED");

    for (;;)

} else {

    Serial.println("SUCCESS");

}

// The default current for the IR LED is 50mA and it could be changed

// by uncommenting the following line. Check MAX30100_Registers.h for all the

// available options.
```

```
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

// Register a callback for the beat detection
pox.setOnBeatDetectedCallback(onBeatDetected);

display_data(0, 0);

#endif
}

void loop()
{
#if ENABLE_MAX30100

//Make sure to call update as fast as possible
pox.update();

int bpm = 0;
int spo2 = 0;

// Asynchronously dump heart rate and oxidation levels to the serial
// For both, a value of 0 means "invalid"
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

//Serial.print("Heart rate:");

bpm = pox.getHeartRate();

spo2 = pox.getSpO2();

Serial.println(bpm);

//Serial.print("bpm / SpO2:");

Serial.println(spo2);

//Serial.println("%");

tsLastReport = millis();

display_data(bpm, spo2);

}
```

```
#endif

drawLine(&xPos);
}

void display_data(int bpm, int spo2) {
    display.fillRect(0, 18, 127, 15, SSD1306_BLACK);

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 18);
    // Display static text
    display.print("BPM ");
    display.print(bpm);
    display.display();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(64, 18);
    // Display static text
    display.print("Spo2% ");
    display.println(spo2);
    display.display();
}

void drawLine(int *x_pos) {
    // Draw a single pixel in white
    display.drawPixel(*x_pos, 8, SSD1306_WHITE);
    display.drawPixel((*x_pos)++, 8, SSD1306_WHITE);
    display.drawPixel((*x_pos)++, 8, SSD1306_WHITE);
}
```

```
display.drawPixel((*x_pos)++, 8, SSD1306_WHITE);

display.drawPixel((*x_pos), 8, BLACK); // -----

//Serial.println(*x_pos);

display.fillRect(*x_pos, 0, 31, 16, SSD1306_BLACK);

display.display();

delay(1);

if (*x_pos >= SCREEN_WIDTH) {

    *x_pos = 0;

}

}

void heart_beat(int *x_pos) {

    /*****/

    //display.clearDisplay();

    display.fillRect(*x_pos, 0, 30, 15, SSD1306_BLACK);

    // Draw a single pixel in white

    display.drawPixel(*x_pos+0, 8, SSD1306_WHITE);

    display.drawPixel(*x_pos+1, 8, SSD1306_WHITE);

    display.drawPixel(*x_pos+2, 8, SSD1306_WHITE);

    display.drawPixel(*x_pos+3, 8, SSD1306_WHITE);

    display.drawPixel(*x_pos+4, 8, BLACK);//-----

    //display.display();

    //delay(1);

    display.drawPixel(*x_pos + 5, 7, SSD1306_WHITE);

    display.drawPixel(*x_pos + 6, 6, SSD1306_WHITE);

    display.drawPixel(*x_pos + 7, 7, SSD1306_WHITE); // .~.

    //display.display();
```

```
//delay(1);

display.drawPixel(*x_pos + 8, 8, SSD1306_WHITE);
display.drawPixel(*x_pos + 9, 8, SSD1306_WHITE); // --

//display.display();

//delay(1);

/*****/

display.drawPixel(*x_pos + 10, 8, SSD1306_WHITE);
display.drawPixel(*x_pos + 10, 9, SSD1306_WHITE);
display.drawPixel(*x_pos + 11, 10, SSD1306_WHITE);
display.drawPixel(*x_pos + 11, 11, SSD1306_WHITE);

//display.display();

//delay(1);

/*****/

display.drawPixel(*x_pos + 12, 10, SSD1306_WHITE);
display.drawPixel(*x_pos + 12, 9, SSD1306_WHITE);
display.drawPixel(*x_pos + 12, 8, SSD1306_WHITE);
display.drawPixel(*x_pos + 12, 7, SSD1306_WHITE);

//display.display();

//delay(1);

display.drawPixel(*x_pos + 13, 6, SSD1306_WHITE);
display.drawPixel(*x_pos + 13, 5, SSD1306_WHITE);
display.drawPixel(*x_pos + 13, 4, SSD1306_WHITE);
display.drawPixel(*x_pos + 13, 3, SSD1306_WHITE);

//display.display();

//delay(1);
```

```
display.drawPixel(*x_pos + 14, 2, SSD1306_WHITE);

display.drawPixel(*x_pos + 14, 1, SSD1306_WHITE);

display.drawPixel(*x_pos + 14, 0, SSD1306_WHITE);

display.drawPixel(*x_pos + 14, 0, SSD1306_WHITE);

//display.display();

//delay(1);

/*****/

display.drawPixel(*x_pos + 15, 0, SSD1306_WHITE);

display.drawPixel(*x_pos + 15, 1, SSD1306_WHITE);

display.drawPixel(*x_pos + 15, 2, SSD1306_WHITE);

display.drawPixel(*x_pos + 15, 3, SSD1306_WHITE);

//display.display();

//delay(1);

display.drawPixel(*x_pos + 15, 4, SSD1306_WHITE);

display.drawPixel(*x_pos + 15, 5, SSD1306_WHITE);

display.drawPixel(*x_pos + 16, 6, SSD1306_WHITE);

display.drawPixel(*x_pos + 16, 7, SSD1306_WHITE);

//display.display();

//delay(1);

display.drawPixel(*x_pos + 16, 8, SSD1306_WHITE);

display.drawPixel(*x_pos + 16, 9, SSD1306_WHITE);

display.drawPixel(*x_pos + 16, 10, SSD1306_WHITE);

display.drawPixel(*x_pos + 16, 11, SSD1306_WHITE);

//display.display();

//delay(1);
```

```
display.drawPixel(*x_pos + 17, 12, SSD1306_WHITE);
display.drawPixel(*x_pos + 17, 13, SSD1306_WHITE);
display.drawPixel(*x_pos + 17, 14, SSD1306_WHITE);
display.drawPixel(*x_pos + 17, 15, SSD1306_WHITE);
//display.display();
//delay(1);
display.drawPixel(*x_pos + 18, 15, SSD1306_WHITE);
display.drawPixel(*x_pos + 18, 14, SSD1306_WHITE);
display.drawPixel(*x_pos + 18, 13, SSD1306_WHITE);
display.drawPixel(*x_pos + 18, 12, SSD1306_WHITE);
//display.display();
//delay(1);
display.drawPixel(*x_pos + 19, 11, SSD1306_WHITE);
display.drawPixel(*x_pos + 19, 10, SSD1306_WHITE);
display.drawPixel(*x_pos + 19, 9, SSD1306_WHITE);
display.drawPixel(*x_pos + 19, 8, SSD1306_WHITE);
//display.display();
//delay(1);
/*****/
display.drawPixel(*x_pos + 20, 8, SSD1306_WHITE);
display.drawPixel(*x_pos + 21, 8, SSD1306_WHITE);
//display.display();
//delay(1);
/*****/
display.drawPixel(*x_pos + 22, 7, SSD1306_WHITE);
```

```
display.drawPixel(*x_pos + 23, 6, SSD1306_WHITE);  
display.drawPixel(*x_pos + 24, 6, SSD1306_WHITE);  
display.drawPixel(*x_pos + 25, 7, SSD1306_WHITE);  
//display.display();  
//delay(1);  
/*****/  
display.drawPixel(*x_pos+26,8,SSD1306_WHITE);  
display.drawPixel(*x_pos + 27, 8, SSD1306_WHITE);  
display.drawPixel(*x_pos + 28, 8, SSD1306_WHITE);  
display.drawPixel(*x_pos + 29, 8, SSD1306_WHITE);  
display.drawPixel(*x_pos + 30, 8, SSD1306_WHITE); // ----  
*x_pos = *x_pos + 30;  
display.display();  
delay(1);  
}
```

III.2.5 montage final

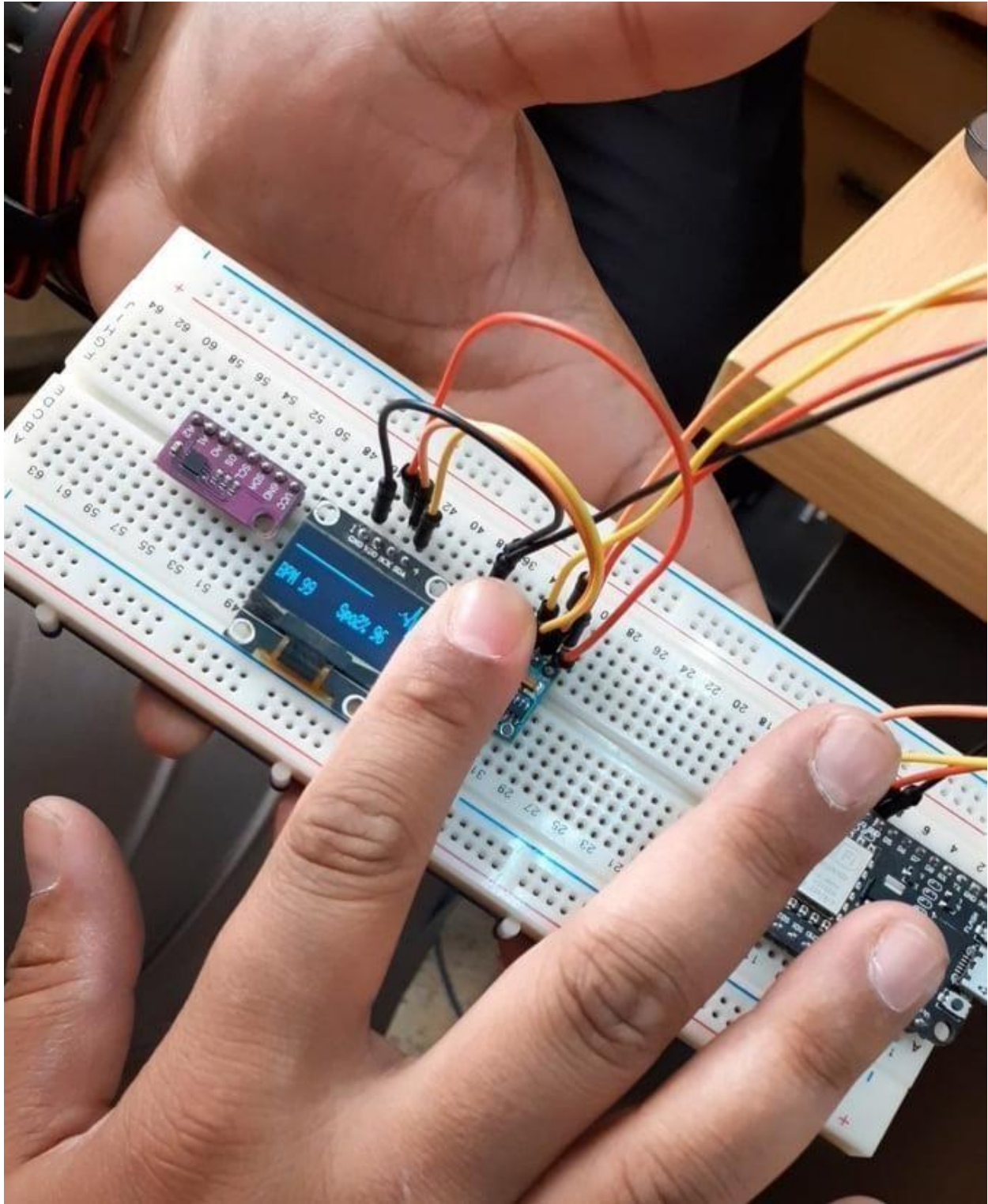


Figure III.9.Montage finale

III.2.6 Code du programme final

```
#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <Wire.h>

#include "Protocentral_MAX30205.h"

#include "MAX30100_PulseOximeter.h"

MAX30205 tempSensor;

#define REPORTING_PERIOD_MS 3000

/*Put your SSID & Password*/

const char* ssid = "AndroidGH"; // Enter SSID here

const char* password = "hamza1986"; //Enter Password here

// PulseOximeter is the higher level interface to the sensor

// it offers:

// * beat detection reporting

// * heart rate calculation

// * SpO2 (oxidation level) calculation

PulseOximeter pox;

uint32_t tsLastReport = 0;

// Callback (registered below) fired when a pulse is detected

// Callback (registered below) fired when a pulse is detected

void onBeatDetected()

{

    Serial.println("Beat!");

}
```

```
ESP8266WebServer server(80);

float HeartRate;

float SpO2;

void setup()
{
  Wire.begin();

  Serial.begin(115200);
  delay(100);
  Serial.println("Connecting to ");
  Serial.println(ssid);

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
```

```
server.begin();

Serial.println("HTTP server started");

// Initialize the PulseOximeter instance
// Failures are generally due to an improper I2C wiring, missing power supply
// or wrong target chip

// The default current for the IR LED is 50mA and it could be changed
// by uncommenting the following line. Check MAX30100_Registers.h for all the
// available options.
// pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
// Register a callback for the beat detection
// or wrong target chip
Serial.println("Initializing pulse oximeter");

if (!pox.begin()) {
    Serial.println("FAILED");
    for(;;);
} else {
    Serial.println("SUCCESS");
}

pox.setOnBeatDetectedCallback(onBeatDetected);

while(!tempSensor.scanAvailableSensors()){
    Serial.println("Couldn't find the temperature sensor, please connect the sensor.");
    delay(1000);
}
```

```
tempSensor.begin();  
  
}  
  
void loop()  
{  
    // Make sure to call update as fast as possible  
  
    //orwrongtargetchip  
  
    pox.update();  
  
  
    HeartRate =pox.getHeartRate();// Gets the values of the Heart rate  
    SpO2 = pox.getSpO2();// Gets the values of the Spo2  
  
  
    // Asynchronously dump heart rate and oxidation levels to the serial  
    // For both, a value of 0 means "invalid"  
    if(millis()-tsLastReport>REPORTING_PERIOD_MS){  
        Serial.print("Heart rate:");  
  
        //Serial.print(pox.getHeartRate());  
  
        Serial.print(HeartRate);  
  
        Serial.print("bpm / SpO2:");  
  
        //Serial.print(pox.getSpO2());  
  
        Serial.print(SpO2);  
  
        Serial.println("%");  
  
        float temp = tempSensor.getTemperature();// read temperature for every 100ms  
  
        Serial.print(temp ,2);  
  
        Serial.println("c" );  
  
        tsLastReport = millis();  
  
    };  
};
```

```
//if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

//delay(1500);

//}

server.handleClient();

}

void handle_OnConnect() {

pox.update();

HeartRate =pox.getHeartRate();// Gets the values of the Heart rate

SpO2 = pox.getSpO2();// Gets the values of the Spo2

float temp = tempSensor.getTemperature();// read temperature for every 100ms

server.send(200, "text/html", SendHTML(temp, HeartRate,SpO2));

}

void handle_NotFound(){

server.send(404, "text/plain", "Not found");

}

String SendHTML(float tempStat,float HearRateStat,float So2Stat ){

String ptr = "<!DOCTYPE html> <html>\n";

ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";

ptr += "<title>ESP8266 Weather Report</title>\n";

ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";

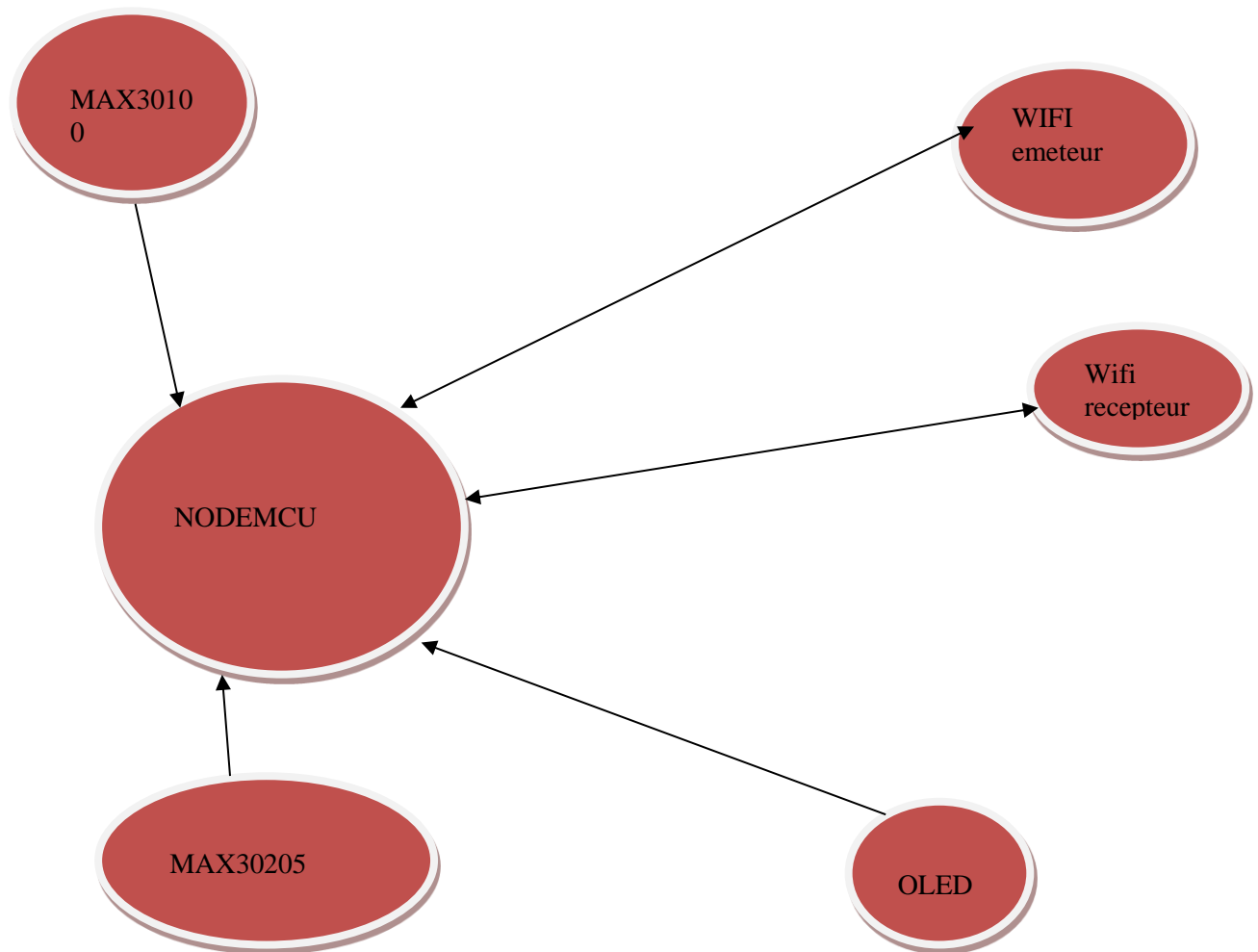
ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";

ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";

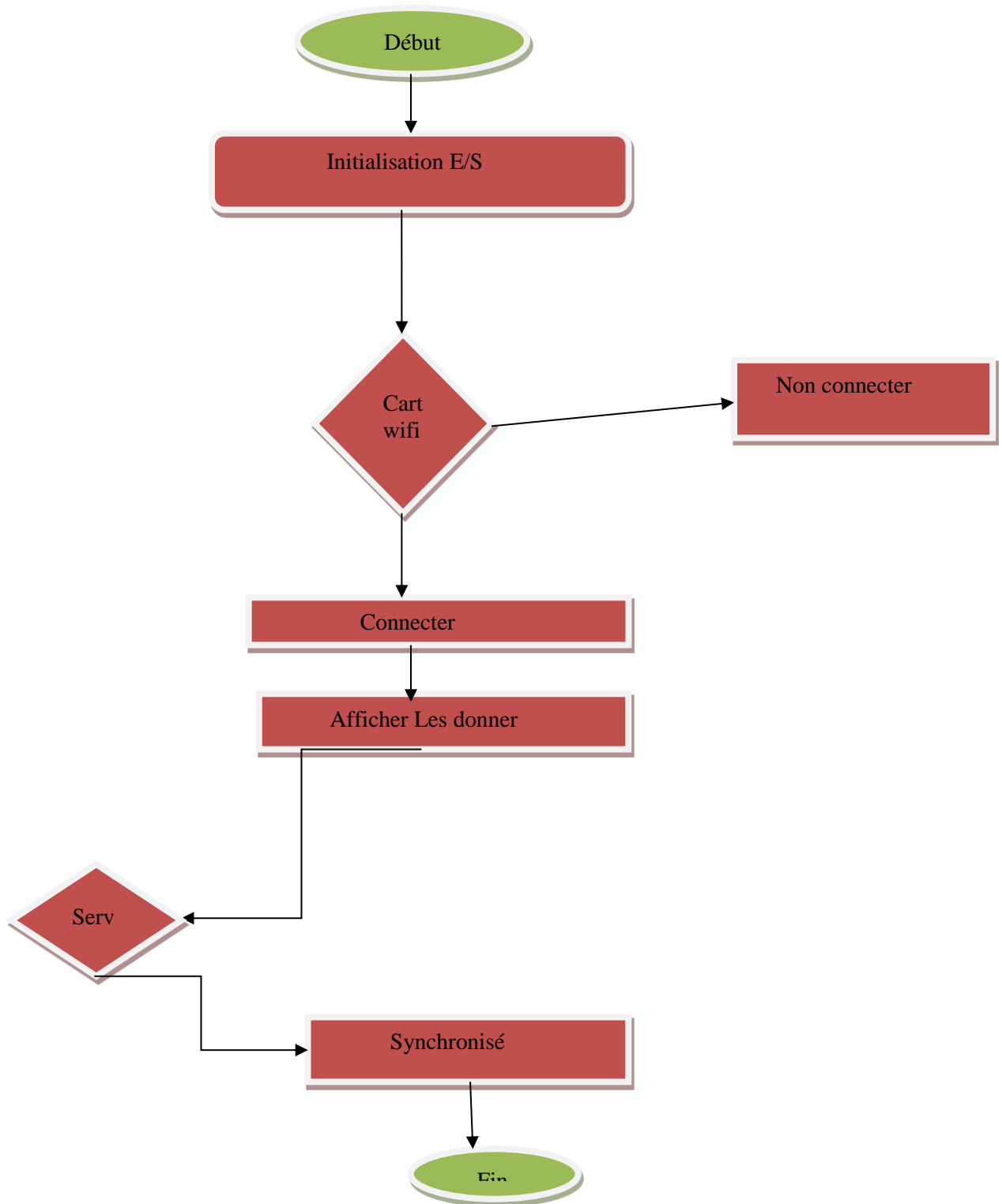
ptr += "</style>\n";
```

```
ptr+="/</head>\n";  
  
ptr += "<body>\n";  
  
ptr += "<div id=\"webpage\">\n";  
  
ptr += "<h1>ESP8266 NodeMCU Health Report</h1>\n";  
  
  
ptr += "<p>Temperature ";  
ptr += (int)tempStat;  
ptr += "<p>Heart Rate: ";  
ptr += (int)HearRateStat;  
ptr += "BPM</p>";  
ptr += "<p>So2: ";  
ptr += (int)So2Stat;  
ptr += "%</p>";  
  
  
ptr += "</div>\n";  
ptr += "</body>\n";  
ptr += "</html>\n";  
return ptr;  
}
```

III .3.1 schéma synoptique



III .3.2 l organigramme



III.4 Conclusion

Dans cette partie du travail nous avons décrit l'avancement des différentes étapes de notre système. Nous avons donné un schéma fonctionnel global pour expliquer le fonctionnement des deux capteurs avec la carte wifi (ESP8266).

Cette réalisation nous a permis de faire connaissance avec les composants électroniques et nous avons appris les bases de programmation de la carte Nodemcu.

Enfin nous pouvons dire que nous avons atteint notre objectif.

Conclusion générale

Conclusion Générale

Au cours de ces études que nous avons menées, nous avons découvert que les systèmes embarqués sont utilisés dans une variété de technologies dans toutes les industries, même dans le domaine médical, et nous envisageons le développement de cette technologie en particulier pour les appareils portables et certaines des fonctions les plus courantes pour les bracelets. Les appareils tels que les trackers de santé, montres intelligentes et de fitness incluent la surveillance des pas, de la distance parcourue, des pas gravis, des calories brûlées, des statistiques d'activité, du temps et de la qualité du sommeil, de la fréquence cardiaque, du niveau de forme physique, des itinéraires d'entraînement, de la localisation GPS, etc. Ces appareils ne sont pas seulement utiles pour le suivi ces types d'habitudes des utilisateurs, mais la technologie est également bénéfique pour les entreprises et diverses industries, comme les soins de santé et la fabrication, pour améliorer la productivité sur le lieu de travail, résoudre rapidement les problèmes et assurer l'efficacité des organisations. Même si sa petite taille peut causer des difficultés à exécuter certaines notifications avec un simple bouton ou un toucher du doigt.

Étant donné que les systèmes embarqués sont partout et que nous ne pouvons pas nous en éloigner, nous avons donc essayé dans ce mémoire de les présenter ainsi que leur évolution quotidienne, en particulier les systèmes de télésurveillance qui représentent une grande partie de nos vies et avec leur partie de programmation intéressante, nous avons essayé de réaliser l'un des systèmes les plus demandés surtout dans le domaine médical, qui est un système de télésurveillance des patients à distance basée sur NodeMcu Wifi afin d'envoyer des informations depuis des capteurs en temps réel à la salle du garde des médecins.

Cette conclusion résume tout le travail pratique que nous avons fait nous avons présenté le principe de fonctionnement de notre prototype en analysant tous les résultats obtenus, nous avons pu évaluer tous les paramètres essentiels pour le suivi des patients, avec ces procédés le médecin pourra prendre une décision meilleure à propos de ce dernier. Notre projet est une initiation estimable dans notre pays pour le développement et l'amélioration de la qualité de vie chez les personnes et établir un accès rapide pour le contrôle du patient.

Conclusion Générale

Pour La réalisation de notre projet on a passé par quatre étapes d'étude :

- ✓ L'installation de la bibliothèque Nodemcu dans l'Arduino
- ✓ Téléchargement de la bibliothèque des deux capteurs :
 - Capteur MAX30100 : détecteur de la saturation en oxygène et la fréquence cardiaque
 - Capteur MAX30205 : détecteur de la température corporelle
- ✓ Testé chaque capteur avec la carte wifi (ESP8266) et le résultat du test de chaque capteur sera afficher dans un site web et aussi sur un OLED
- ✓ Puis en a testé les deux capteurs avec la carte wifi en même temps et le résultat des deux capteurs sera afficher dans le même site web

Le projet que nous avons réalisé, se veut juste d'être un système sur lequel l'on pourrait se baser pour la réalisation de systèmes beaucoup plus performants et complets.

Ce travail nous a permis d'enrichir notre savoir dans le domaine de la télésurveillance. En effet, nous avons eu l'opportunité de découvrir les différents aspects de la conception des projets électronique et les outils de développements de l'étude jusqu'à la réalisation des objectifs demandés, où nous avons réalisé un système de mesure comme solution pour le contrôle et le diagnostic des patients

Référence bibliographique

- [1] F.KRIEF , livre ‘’ Les systèmes embarqués communicants : mobilité, sécurité, autonomie ‘’ , réseaux et télécommunication , France ,2008. (20/05/2021)
- [2] <https://www.gotronic.fr/art-carte-arduino-nano-12422.htm> (18/06/2021)
- [3] <https://www.generationrobots.com/fr/401275-carte-arduino-due.html> (18/06/2021)
- [4] <https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm> (18/06/2021)
- [5] <https://www.generationrobots.com/fr/401353-carte-arduino-leonardo.html> (19/06/2021)
- [6] (MEKHALFIA Toufik et GHADBANE Toufik,Etude et réalisation d'un système de commande à distance des installations électriques pour la domotique, ELECTRONIQUE DES SYSTEMES EMBARQUES, MOHAMED BOUDIAF - M'SILA,2017-2018)
- [7] (ZENNOUCHE Kahina HADJ ALI Zineb ,Etude et réalisation d'un système de contrôle pour la sécurité, le confort et l'environnement d'une maison intelligente-Smart house, Electronique des Systèmes Embarqués,Univ-AKLI MOHAND OULHADJ-BOUIRA ,2018-2019)
- [8] <https://www.electronique-mixte.fr/wp-content/uploads/2018/07/PFE-Rapport-de-projet-de-fin-d'étude-39-1> (26/06/2021)
- [9] <https://michel.re/esp8266> (28/06/2021)
- [10] <https://fr.aliexpress.com/item/1005001520232701.html> (28/06/2021)
- [11] <https://www.ouedkniss.com/capteur-fr> (29/06/2021)
- [12] <https://www.futura-sciences.com/tech/definitions/technologie-systeme-embarque-15282/> (20/05/2021)
- [13] http://www.magoie.net/coursUniv/coursUniv_146_pdf.pdf (02/07/2021)
- [14]<http://www.lossendiere.com/wp-content/uploads/2017/09/comparatif-technique-cartes-arduino.pdf> (les différentes cartes Arduino) (02/07/2021)
- [15] <https://michel.re/esp8266/> (05/07/2021)
- [16] (OUALI ALAMI MOHAMMED/Memoire/ ingénieur dtetat en systems electroniques et telecommunications/Univ-SIDI MOHAMED BENABDELLAH/2015_216)

Référence bibliographique

[17] https://www.tutorialspoint.com/embedded_systems/es (22/05/2021)

[18] <https://www.passeportsante.net/fr/Actualites/Dossiers/DossierComplexe.aspx?doc=Tout-ce-que-vous-devez-savoir-sur-la-teledrdecine> (22/05/2021)

[19] (Melle. SIDI YKHLEF Asma Mr. LASLA Abderrahim/mémoire /Master en informatique/Univ-ABOU BEKR BELKAID-Tlemcen/2014_2015)

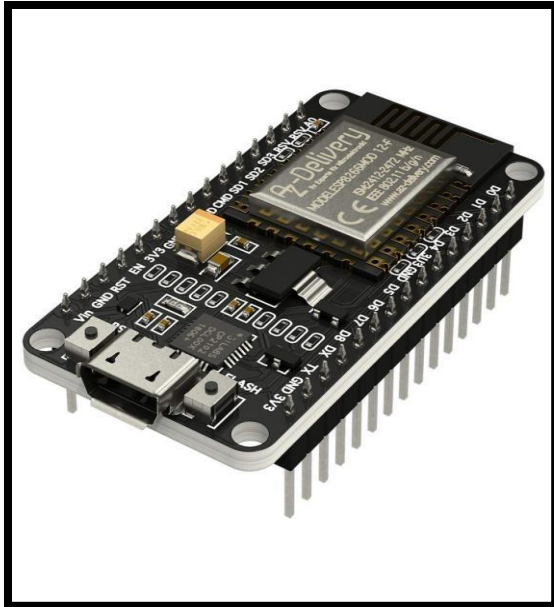
[20] <http://dictionnaire.sensagent.leparisien.fr> (11/06/2021)

[21] https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742_decouverte-de-larduino/3414_presentation-darduino (18/06/2021)

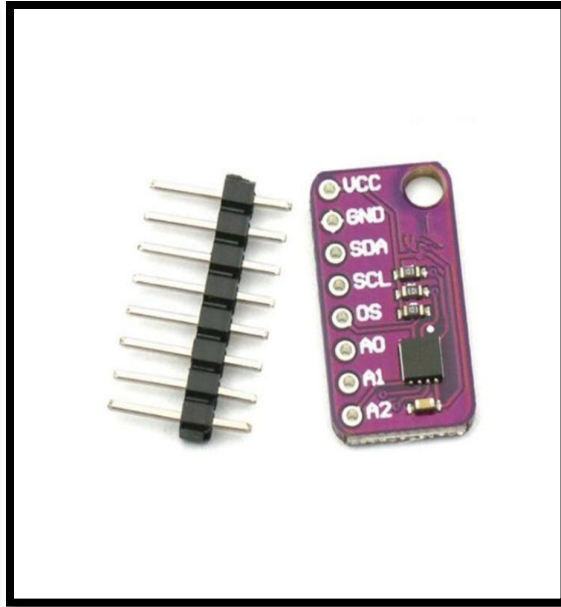
[22] <http://domotics.fr/index.php/2019/02/25/les-differentes-cartes-arduino/> (18/06/2021)

[23] <https://www.gotronic.fr/art-carte-arduino-uno-12420.htm> (18/06/2021)

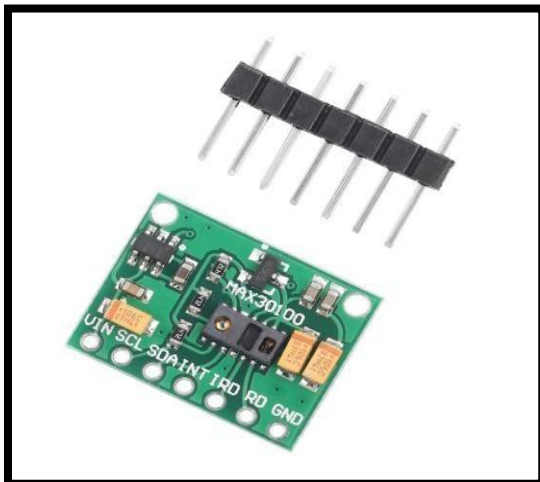
1. les composants utilisés dans ce projet



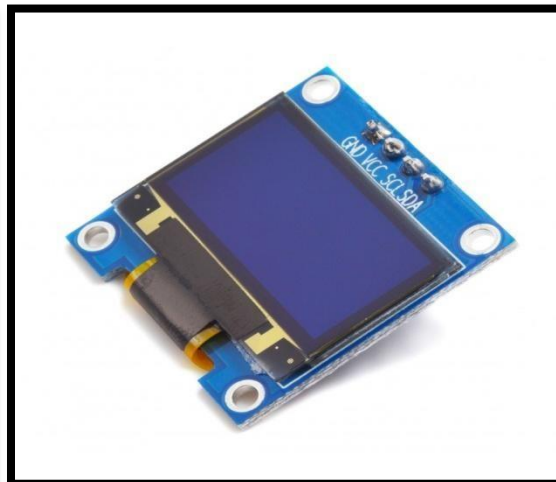
Nodemcu(esp8266)



MAX30205



MAX30100

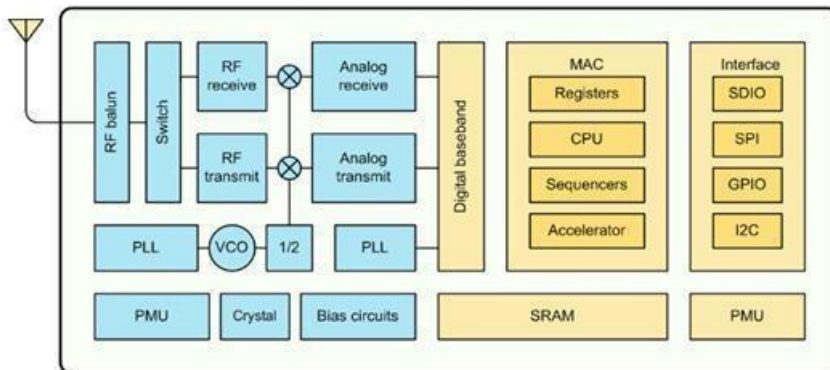


OLED(SSD1306)

2. datasheet

Circuit ESP8266

- Espressif Systems (Shanghai) Pte. Ltd.
- 32 pin Ultra Low Power
- WIFI 2.4GHz b,g,n WPA/WPA2
- 80/160Mhz (pll)
- 32 bits RTOS inclus (pas visible)
- 80% de la puissance disponible (non wifi)

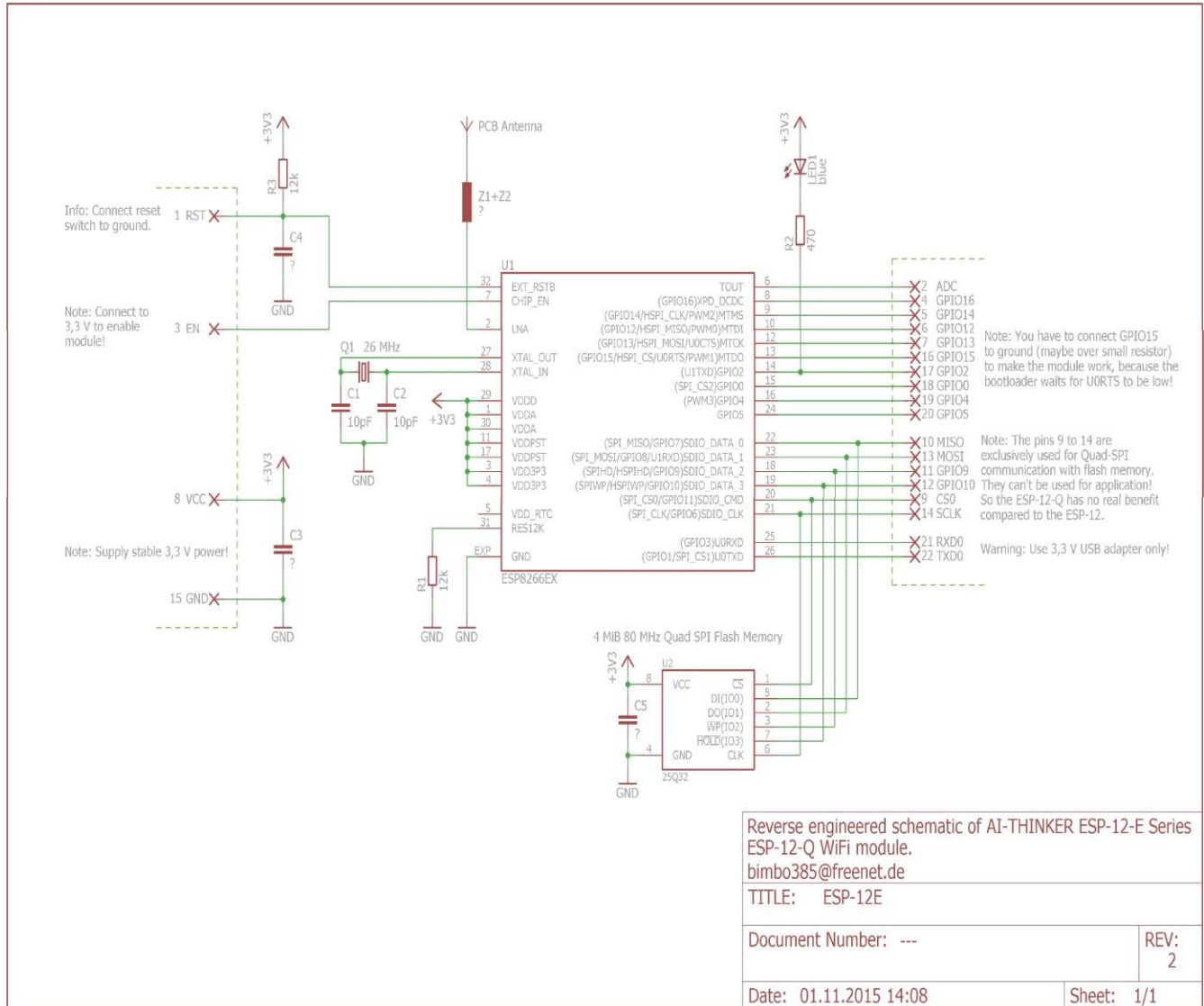


- Flash externe 512b-16Mb

connexion de ESP8266:

- Reset (High=run, Low=reset)
- ADC
- CH_PD, Enable (High=on, Low=off)
- GPIO16/D0/USER/WEAK
- GPIO14/D5/SPI-SCK
- GPIO12/D6/MISO
- GPIO13/D7/MOSI
- GPIO1/D10 TX0
- GPIO3/D9 RX0
- GPIO5 /D1/SCL(I2C)
- GPIO4 /D2/SDA(I2C)
- GPIO0/D3/FLASH
- GPIO2/D4
- GPIO15/D8/TX2/SPI-CS

2.3ESP8266 Schéma:



Reverse engineered schematic of AI-THINKER ESP-12-E Series ESP-12-Q WiFi module.
 bimbo385@freenet.de
 TITLE: ESP-12E

| | |
|------------------------|------------|
| Document Number: --- | REV: 2 |
| Date: 01.11.2015 14:08 | Sheet: 1/1 |

3. Capteur MAX30205

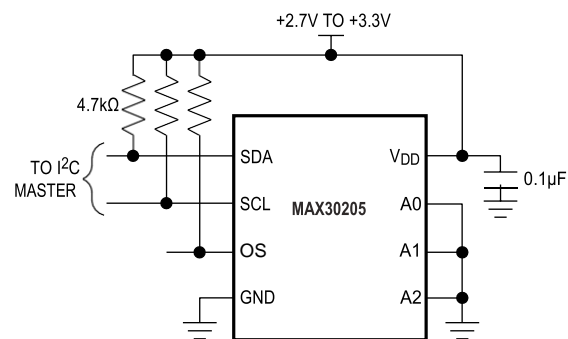
Features

- ✚ High Accuracy and Low-Voltage Operation Aids Designers in Meeting Error and Power Budgets
 - ✓ 0.1°C Accuracy (37°C to 39°C)
 - ✓ 16-Bit (0.00390625°C) Temperature Resolution
 - ✓ 2.7V to 3.3V Supply Voltage Range
- ✚ One-Shot and Shutdown Modes Help Reduce Power Usage
- ✚ 600µA (typ) Operating Supply Current
- ✚ Digital Functions Make Integration Easier into Any System
 - ✓ Selectable Timeout Prevents Bus Lockup
 - ✓ Separate Open-Drain OS Output Operates as Interrupt or Comparator/Thermostat Output

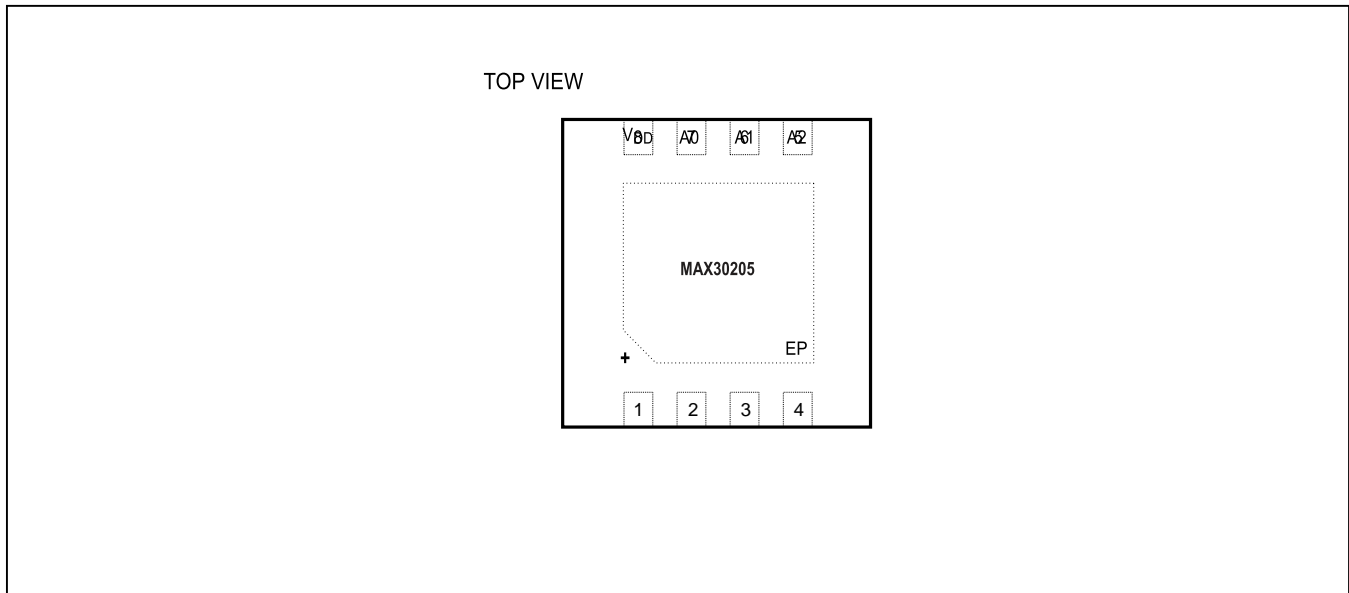
Applications

- Fitness
- Médical

Typical application



3.4 Pins de configurations



4. capteur MAX30100

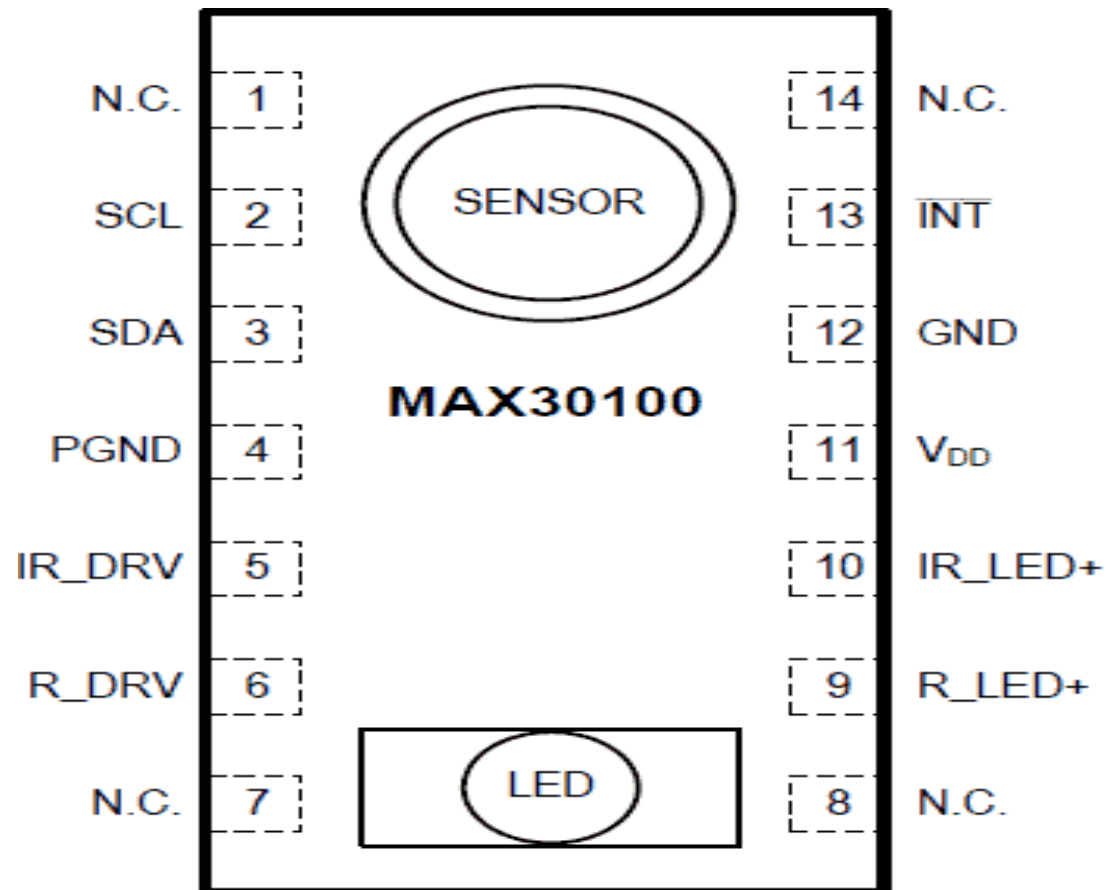
Features

- ✚ Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
 - ✓ Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
 - ✓ Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
- ✚ Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
 - ✓ Programmable Sample Rate and LED Current for Power Savings
 - ✓ Ultra-Low Shutdown Current (0.7 μ A, typ)
- ✚ Advanced Functionality Improves Measurement Performance High SNR Provides Robust Motion Artifact Resilience
 - ✓ Integrated Ambient Light Cancellation
 - ✓ High Sample Rate Capability
 - ✓ Fast Data Output Capability

Applications

- Wearable Devices
- Fitness Assistant Devices
- Medical Monitoring Devices

Pins de configuration



5.OLED SSD11306:

This is an OLED monochrome 128x64 dot matrix display module with I2C Interface. Comparing to LCD, OLED screens are way more competitive, which has a number of advantages such as high brightness, self-emission, high contrast ratio, wide viewing angle, wide temperature range, and low power consumption.

➤ Features:

- ✓ Interface: I2C (3.3V logic level)
- ✓ Resolution: 128*64
- ✓ Angle of view: >160 degree
- ✓ Display color: White Display
- ✓ Dimension: 0.96inch
- ✓ Driver IC: SSD1306
- ✓ Power supply: 3.3V~5VDC
- ✓ Operating temperature: -20°C~70°C
- ✓ Application: smart watch, MP3, thermometer, instruments, DIY projects, etc.

Les pins de configuration :

| Pin No: | Pin Name: | Description |
|---------|---------------------|---|
| 1 | Ground (Gnd) | Connected to the ground of the circuit |
| 2 | Supply (Vdd,Vcc,5V) | Can be powered by either 3.3V or 5V |
| 3 | SCK (D0,SCL,CLK) | The display supports both IIC and SPI, for which clock is supplied through this pin |
| 4 | SDA (D1,MOSI) | This is the data pin of the both, it can either be used for IIC or for SPI |
| 5 | RES(RST,RESET) | When held to ground momentarily this pin resets the module |
| 6 | DC (A0) | This is command pin, can either be used for SPI or for IIC |
| 7 | Chip Select (CS) | Normally held low, used only when more than one SPI device is connected to MCU |

Schema d'OLED SSD1306 I2C

