La République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université Saad Dahleb Blida 1

Institut d'Aéronautique et des Etudes Spatiales (IAES) Département Etudes Spatiales

Mémoire de fin d'études

En vue de l'obtention du diplôme de **Master**
En : **Aéronautique**

Option : **Télécommunications Spatiales**

THEME

Etude et réalisation d'un système électronique de collecte de paramètres météorologiques.

Proposé et dirigé par :

Dr. AZMEDROUB Boussad

Réalisé par :

M. Lemoudaa Oussama
M. Touati Djamel

*Soutenue le **... juillet 2022**, devant le jury composé de :*

Promotion : 2021/2022

Remerciement

On remercie Dieu le tout puissant pour nous avoir donné le courage et la volonté d'achever ce travail.

On voudrait dans un premier temps exprimer toute notre reconnaissance à notre encadreur Monsieur AZMEDROUB Boussad Maitre de Conférence à l'Institut d'Aéronautique et des Etudes Spatiales. On le remercie pour sa patience tout au long de la réalisation de ce mémoire, pour l'inspiration, l'aide et le temps qu'il a bien voulu nous consacre mais surtout pour ses judicieux conseils qui ont contribué à alimenter notre réflexion, Sans lui ce mémoire n'aurait jamais vu le jour.

On adresse également nos remerciements à Monsieur TAHRAOUI Sofiane ainsi qu'à toute l´équipe pédagogique de l'institut d'aéronautique et des études spatiales de Blida qui ont contribué à la réussite de notre formation au cours de ces cinq dernières années.

Enfin, on remercie nos très chers parents, qui ont été là pour nous, nos frères et sœurs pour leurs soutiens inconditionnels et leur encouragement.

Un spécial remerciement a Bouchra .A et Lyna .B sur l'aide de faire le meilleurs choix de couleurs de l'interface graphique mais aussi pour leurs soutien morale tous les jours.

Un grand merci à nos amies Meskine Hocine qui nous a apporté son soutien moral et intellectuel tout au long de cette démarche. Aussi a tous les intervenants qui ont participé de près ou de loin a` la réalisation de ce modeste travail, on vous présente nos remerciements, notre respect et notre gratitude.

Dédicace

Je dédie ce travail à mon cher père

AMMAR LEMOUDAA

La personne la plus précieuse, et ma plus grande perte

Que dieu l'accueille dans son vaste paradis

Résumé :

Ce projet de fin d'étude de cycle master consiste à étudier et réaliser un système électronique de collecte de données météorologiques (température, humidité et pluie), avec une option de détection de flamme et de localisation de système, Ceci grâce à un ensemble de capteurs et d'un modules GPS connectées a un élément central (La carte NodeMcu) de traitement de toutes les données du système grâce à son microcontrôleur ESP8266 et de transmission à l'aide de son module Wifi. Ces deux opérations de traitement de transmission de données sont à base de langages informatiques. Ces données sont affichées en temps réel sur un afficheur LCD mais aussi sur une interface graphique dans un réseau local.

Abstract:

This end-of-study master's project consists of studying and creating an electronic system for collecting meteorological data (temperature, humidity and rain), with an option of flame detection and system location, due to a set of sensors and a GPS module connected to a central element (The NodeMcu card) for processing all system data due to its ESP8266 microcontroller and for transmission using its Wifi module. These two data transmission processing operations are based on computer languages. These data are displayed in real time on an LCD display but also on a graphic interface in a local netwok

Tables des matières

Table des figures:

Liste des abréviations :

UV : Ultra-violet
IR : Infra rouge
Wi-Fi : Wireless Fidelity
HR : humidité Relative
IDE: L'espace de développement intégré
GUI: une interface graphique
LCD : Liquid Crystal Display
DHT: Digital Humidity And Temperature
ESP: Espressif Systems
GND: Ground
GPS: Global Positioning System
I2C: Inter-Integrated Circuit
IDE: Integrated Development Environment
RXD: Received Data
SDA: Serial Data
TXD: Transmit Data
VCC: Voltage Common Collector
HTML: Hyper Text Mark-Up Language
CSS : Cascading style sheets
IP : Internet Protocol

Introduction générale :

Le besoin d'évaluation des phénomènes météorologiques tels que la température, l'humidité ou encore la luminosité jouent un rôle important dans plusieurs domaines tels que l'aéronautique, l'architecture, la conversion d'énergie...etc. Afin de prévoir le temps les météorologues ont besoin de rassembler des informations sur plusieurs échelles. Ce qui a donné lieu à la nécessité de développer de nouveaux outils d'estimation de variables météorologiques. Le progrès technologique a conduit au développement de nouveaux instruments de mesures de toutes sortes et de hautes performances. Avec cette avancée majeure, l'ingénierie météorologique a connu un nouvel essor dont l'émergence de nouveaux instruments exploités pour l'estimation des différentes grandeurs météorologiques. Dans ce mémoire nous proposons la réalisation d'un système de mesure de paramètres météorologiques. Ce système est composé de plusieurs capteurs pour l'acquisition des données, d'une carte à microcontrôleur Node Mcu ESP8266 pour le traitement des données reçues et le transfert par wifi des grandeurs mesurées qui peuvent être consulté par tout type d'appareil qui pourra se connecter au réseau local et voir les résultats en temps réel via une interface html. Pour ce faire nous avons subdivisé ce mémoire en trois chapitres:

- Le premier chapitre sera la porte de découverte et description des composants et des logiciels utilisés pour notre réalisation.
- Le deuxième chapitre se basera sur l'interconnexion entre les différents composants et l'élément central la carte NodeMcu ESP8266 pour les tester et visualiser leurs données.
- Le troisième chapitre sera pour la schématisation finale de système électronique et la création d'un site local dans le but de mieux visualiser les données en temps réel.

La conclusion synthétise le travail réalisé et ouvre la voie sur quelques perspectives qui peuvent être développées.

CHAPITRE 01 :
COMPOSANTS ET LOGICIELS UTILISEES POUR LA REALISATION

Introduction :

L'un des besoins fondamentaux dans le domaine de l'aéronautique et de l'aérospatial sont les paramètres météorologiques, qui jouent un grand rôle dans la sûreté, la sécurité et la sauvegarde des aéronefs en plus des aides à la navigation.

Notre objectif dans ce chapitre est une présentation générale des stations météos, par la suite nous allons présenter les différents modules et les capteurs et logiciels utilisés dans notre projet.

Le système électronique qui collecte ces paramètres, et permet d'une part une visualisation des paramètres sur un écran LCD, et d'autre part de transmettre les informations récoltées par les capteurs vers un réseau local. Parmi les paramètres récoltés on cite la température et l'humidité. Ce système est basé sur des capteurs et une carte électronique à microcontrôleur.

1.1Définition de La stations météo électronique

La station météo est un moyen facile d'obtenir des informations précises sur la température, le taux d'humidité, le taux d'ensoleillement d'un environnement et les prévisions climatologiques.
La station météo offre une prévision météo en temps réel utile voire indispensable pour qui a des activités extérieur tant professionnelles que de loisirs. Citons par exemple :

- les agriculteurs et les jardiniers, une bonne météo est nécessaire pour les lots récoltes, les semis… l'affichage des phases lunaires, avec l'aide d'un calendrier lunaire, leurs permettent de jardiner avec la lune ;
- les randonneurs et les vététistes, un temps clément leur garantit une sortie agréable ;
- le charpentier et les couvreurs, l'absence de précipitation est tout indiquée avant de découvrir un toit.

Selon le degré de sophistication de la station météorologique, vous pourrez effectuer différents types de mesure en fonction des appareils de mesure présents :

- le pluviomètre ;
- le capteur d'UV ;
- l'alerte intempérie comme l'orage, la pluie ou la grêle ;

- la mesure de la qualité de l'air ;
- la vitesse et la direction du vent ;
- etc [1].

1.2 Types de technologies des stations météos

On distingue 3 types :

Station météo avec capteurs :

Figure 1.1 : station météo avec capteurs [1].

Station météo pas cher, elle est la plus répandue. Constituée de capteurs intérieur et extérieur, elle transmet les relevés à une console principale par réseau intranet ou par ultrasons. La station elle-même analyse les données et gère l'affichage en temps réel. Peut proposer un historique voire un partage de relevés [1].

Station météo wifi

Figure 1.2 : station météo wifi [1].

Reçoit et transmet les informations météo diffusées. Fonctionne que sur un territoire spécifique.

Station météo interconnectée

Figure 1.3 : station météo interconnectée [1].

La plupart des stations météo. Échange ses informations avec tablettes et smartphones. Continuellement mise à jour grâce aux données des satellites, c'est une station météo très précise [1].

1.3 Description de système électronique de transmission des données

Dans notre réalisation, nous appliquerons un système électronique de transmission des données, en utilisant les différents capteurs et modules connectés à la carte ESP8266 pour envoyer les données vers un site local.

1.4 Présentation de la carte NodeMCU ESP 8266

L'ESP 8266 est un microcontrôleur permettant de se connecter en Wifi. Cette puce (le microcontrôleur ou MCU) a été développée en 2014 par la société Chinoise Espressif … d'où le nom ESP, la puce est fabriquée par une société tierce : AI-Thinker.

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 ou ESP-12E …etc.

Ce composant embarque un module Wifi, de la mémoire, une liaison série et gère des ports GPIO. Tout cela en quantité différente en fonction de la version.

Pour faciliter son utilisation des cartes ont été créées. Ces cartes permettent de connecter le microcontrôleur à un PC en USB, gèrent l'énergie et fournissent un firmware permettant de programmer le microprocesseur. On parle de carte NodeMCU ou Wemos.

En ce qui concerne NodeMCU, ce terme désigne à la fois le micro logiciel permettant de programmer le microcontrôleur en LUA et aussi la carte complète. La carte NodeMCU est basée sur le module ESP-12E qui lui-même est basé sur le microcontrôleur Esp8266.

1.4.1 Programmation de la carte NodeMCU ESP 8266

L'ESP8266 peut se programmer de plusieurs façons :

- Avec des scripts Lua, interprétés ou compilés, avec le micro logiciel NodeMCU
- En C++, avec l'IDE Arduino
- En JavaScript, avec le le micro logiciel Espruino
- En MicroPython, avec le le micro logiciel MicroPython
- En C, avec le SDK d'Espressif ou avec le SDK esp-open-sdk3

Pour notre cas nous avons utilisés une carte NodeMCU, connectée en USB à un PC et programmée avec l'IDE Arduino.
Pour que cela soit possible, il faut bien entendu avoir installé l'IDE, installé le driver windows pour que l'USB arrive à communiquer avec la carte. Installer dans l'IDE les modules et librairies qui vont permettre de compiler pour l'ESP8266.

La carte NodeMCU se met automatiquement en mode apprentissage et normalement il n'y a rien à faire avant de télé verser le programme depuis l'IDE Arduino. Dans la pratique ceci ne fonctionne pas toujours. Quand le « flashage » de la carte échoue, il faut remettre la carte dans le mode apprentissage en utilisant les boutons situés de part et d'autre du port 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

Ceci est valable pour les cartes NodeMcu. Pour les autres modèles, par exemple ESP-01, on accède au mode "flash" en connectant certaines Pins aux pôles V+ ou GND.

Le NodeMCU se connecte au PC avec un câble USB de type Type A mâle - micro USB mâle. Il existe beaucoup de ces câbles dans la nature car on les utilise souvent pour raccorder un téléphone à un chargeur. Hors, souvent, ces câbles de charge n'ont que deux fils de soudés aux fiches USB. Les fils dédiés à l'alimentation électrique. Les deux dédiés aux données ne sont pas présents. Dans ce cas, bien évidemment, il sera impossible d'injecter un programme au NodeMCU [2].

1.4.2 Alimentation électrique

La tension de fonctionnement de l'ESP8266 est de 3.3v.il faudra donc être vigilent au niveau des GPIO à ce que les composants connectés respectent cette tension.

Il existe plusieurs possibilités pour alimenter électriquement le NodeMcu, le schéma ci-dessous, qui en est extrait, montre les pins sur lesquels on peut raccorder une alimentation électrique ainsi que la tension maximum pour chacun.

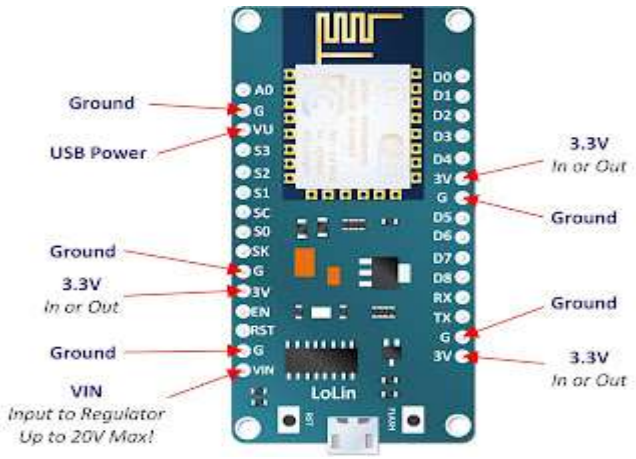


Figure 1.4 : Catre Node Mcu ESP8266 [2].

Ce qui est indiqué pour l'alimentation sur VIN ! Dans de nombreux documents, cette tension doit être comprise entre 3.3V et 9V maximum. Contrairement à ce qu'il y a d'indiqué sur l'image ci-dessus, il ne faudra en aucun cas dépasser cette tension d'entrée (9V donc !).

On voit donc que la carte s'alimente en connectant une source électrique 3.3V sur une des bornes 3.3V, en se connectant en micro USB avec câble USB ou en connectant une source sur VIN.

1.4.3 Utilisation des PIN

Les Pins GPIO de l'ESP8266 s'utilisent comme sur un Arduino. En ce qui concerne les cartes NodeMCU :

Il existe plusieurs constructeurs qui proposent des cartes Nodemcu (LoLin, Amica, …). On trouve aussi des cartes sans marques.il existe également plusieurs versions de la carte. On trouve souvent sur le net des références aux versions 0.9, 1.0, 2.0 et 3.0 (décembre 2017).

En fonction de la version de la carte, l'architecture des PIN GPIO peut varier. Mais ces variations sont souvent minimes.

L'image ci-dessous montre les « pin map » des v2 et v3. La différence est entourée en rouge.

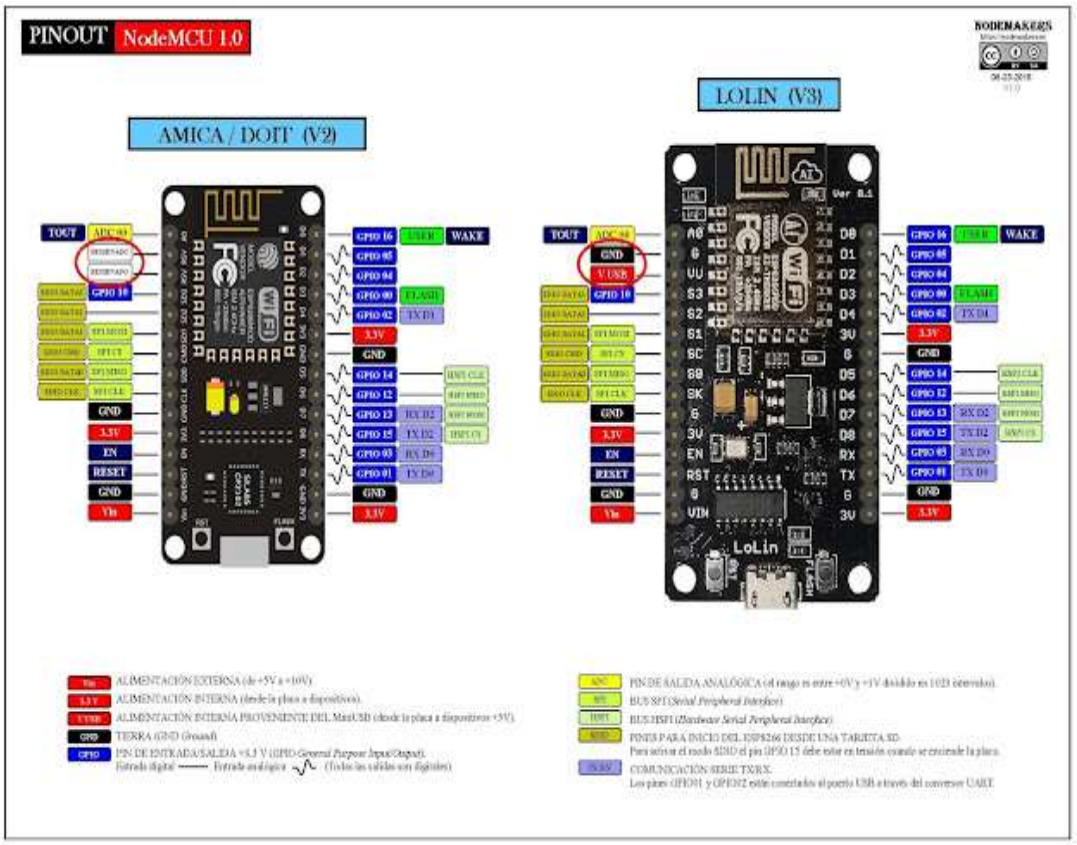


Figure 1.5 : comparaison des modules Amica et LoLin [2].

Et l'image ci-dessous montre les correspondances entre les noms des ports tels qu'indiqués sur la carte, les GPIO et les fonctions spécifiques associées à chacun :

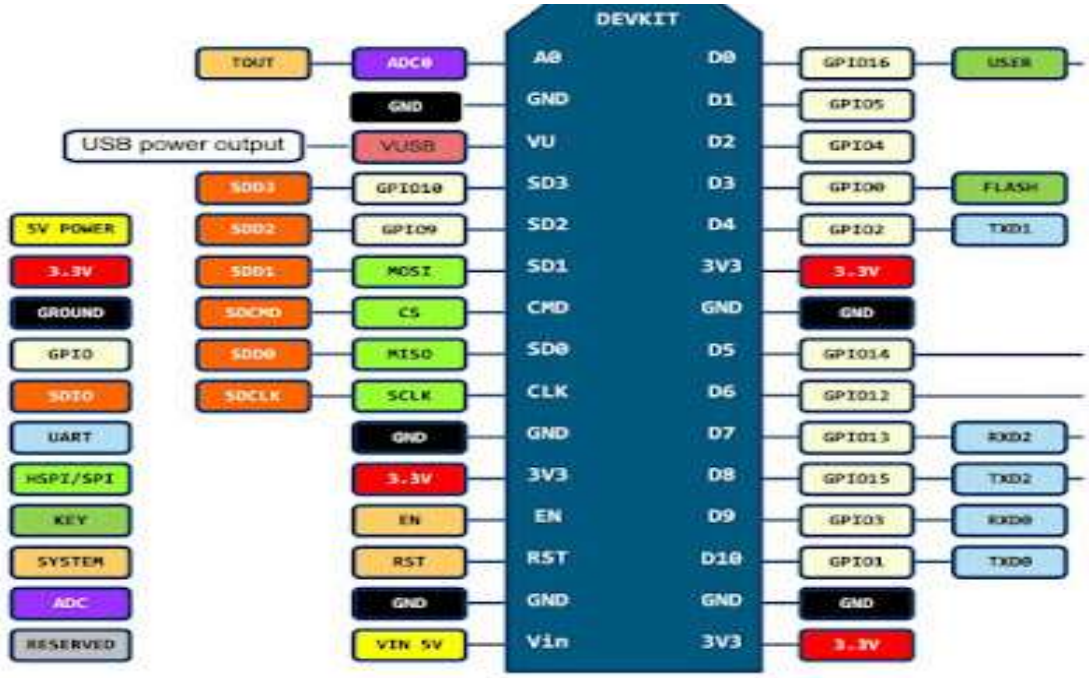


Figure 1.6 : Schéma des fonctions spécifiques associées aux ports [2].

1.5 Le logiciel Arduino IDE

Les fonctions exécutées par logiciel d'Arduino sont la capacité d'écrire et de compiler divers programmes, ainsi que de communiquer et connecter avec la carte Arduino pour transférer le fichier programme.

L'espace de développement intégré (IDE) dédié à la programmation des cartes Arduino et du langage Arduin4o comprend :

- Barre de menu, on retrouve dans tous les logiciels une interface graphique (GUI).
- Barre de bottons, cette barre permet à l'utilisateur d'accéder aux différentes fonctions de base du programme et facilite également son utilisation.
- Editor, pour écrire votre code logiciel en plus des onglets personnalisés pour la navigation.

- Zone de messages, Pour que tous les statuts des procédures en cours s'affichent, comme le montre dans la figure 5.

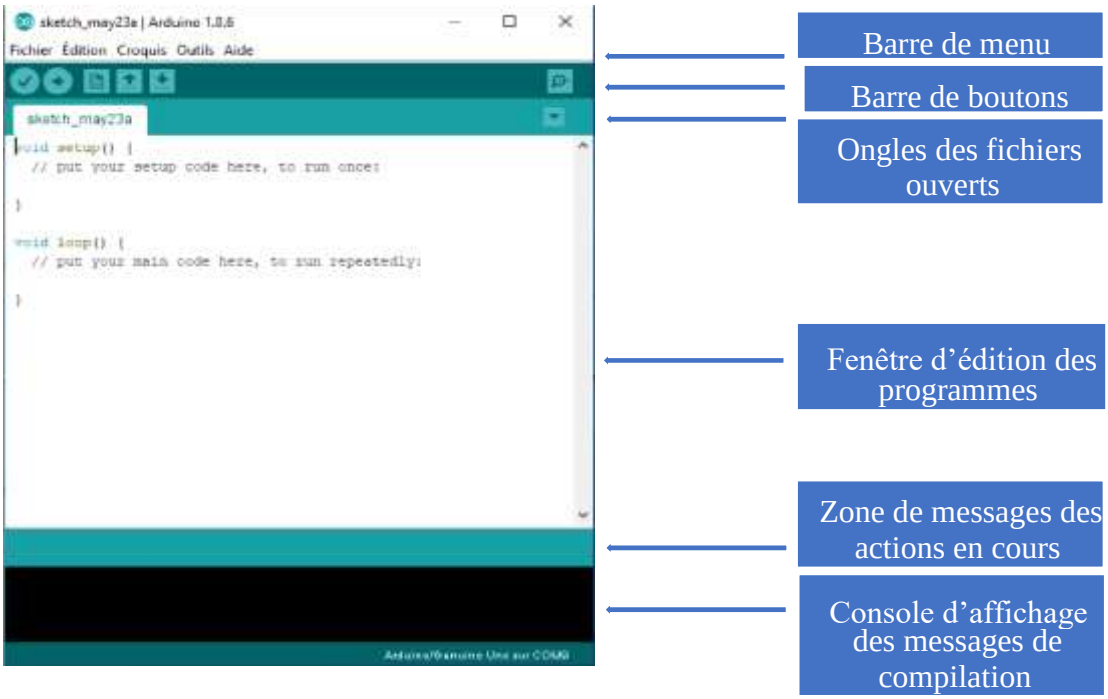


Figure 1.7 : Interface de logiciel Arduino IDE.

Console texte, Il affiche les messages liés au résultat de la compilation du programme Comme le montre dans la figure :

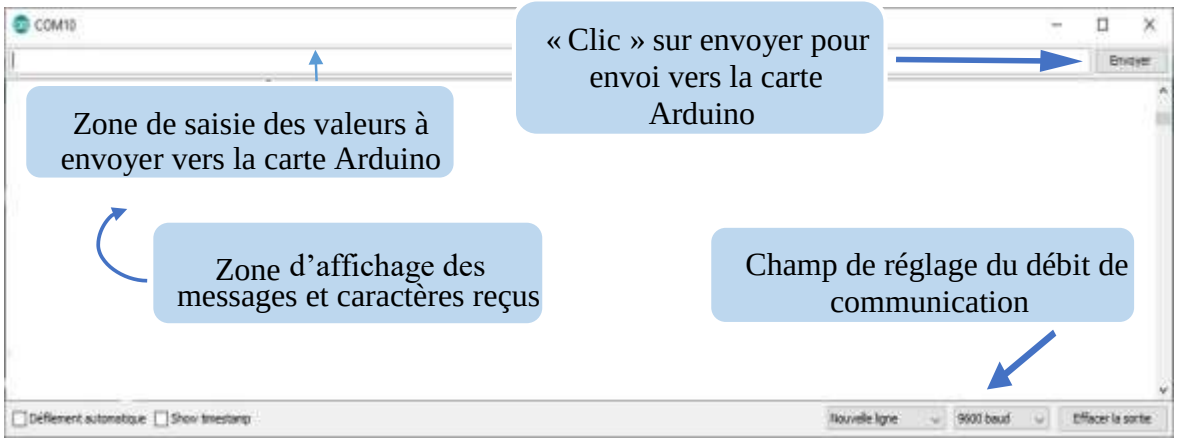


Figure 1.8 : interface de moniteur série.

Terminal série (fenêtre séparée), Cette fenêtre est destinée à l'envoi des caractères et à l'affichage des messages reçus de la carte, et elle permet également à l'utilisateur de corriger

facilement les erreurs de programme, ainsi que d'afficher l'état des variables et les résultats des transferts et des calculs.

1.6 Capteur DHT11

1.6.1 Définition d'un capteur de température

Un capteur de température est un composant qui recueille la température et / ou l'humidité de l'extérieur et le transforme en un signal numérique ou électronique qu'il envoie à une carte électronique telle qu'une carte Arduino. Il existe de nombreux types de capteurs et pour de nombreux domaines. La précision entre la température réelle et la température du capteur est l'un des principaux facteurs influençant la différenciation; Un autre facteur qui change est la température maximale et minimale qu'ils permettent, étant le capteur de température professionnel celui qui supporte le plus de degrés. Le temps de réponse, la sensibilité ou le décalage sont d'autres éléments qui différencient un capteur de température d'un autre.

Dans notre réalisation nous avons utilisés le capteur de température Arduino DHT11 :

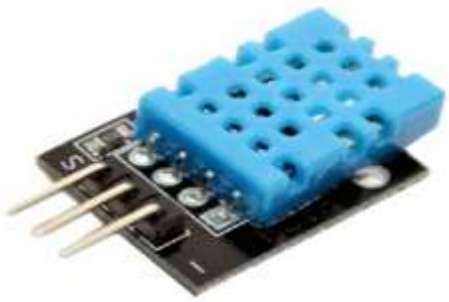


Figure 1.9 : Capteur DHT11 [3].

La mesure de la température et de l'humidité est très courante dans de nombreux projets de fabricants électroniques. Dans le bricolage, il est courant de devoir mesurer ces paramètres pour contrôler certains systèmes. Par exemple, pour pouvoir créer un système de réfrigération, d'entretien des plantes ou de climatisation qui démarre si la température ou l'humidité atteint

1.6.2 Caractéristiques techniques :

- Alimentation 3,5v à 5v
- Consommation de courant de 2,5 mA
- Signal de sortie numérique
- Plage de température de 0 ° C à 50 ° C
- Précision pour mesurer la température à 25 ° C d'une variation d'environ 2 ° C
- La résolution pour mesurer la température est de 8 bits, 1 ° C
- L'humidité peut mesurer de 20% HR à 90% HR
- Précisément pour une humidité de 5% HR pour des températures comprises entre 0 et 50 ° C
- La résolution est de 1% HR, il ne peut pas capter des variations inférieures à cela
- Fiche technique Mouser

Concernant les données, diffusé en numérique. Par conséquent, il n'est pas nécessaire de passer de l'analogique au numérique comme dans d'autres capteurs. Cela a compliqué le code à écrire dans l'IDE Arduino, mais dans ce cas, ce n'est pas nécessaire et c'est beaucoup plus facile. Bien que le capteur lui-même soit analogique, il comprend un système pour effectuer la conversion et peut être connecté directement à une entrée numérique.

Le signal analogique, qui est une variation de la tension, du capteur est converti au format numérique pour être envoyé au microcontrôleur. Il est transmis en une trame de 40 bits qui correspondent aux informations d'humidité et de température capturées par le DHT11. Les deux premiers groupes de 8 bits concernent l'humidité, c'est-à-dire le 16 bits les plus significatifs de cette trame. Puis les 2 autres groupes 8 bits restants pour la température. Autrement dit, il a deux octets pour l'humidité et deux octets pour la température. [3]

1.7 Capteur de pluie

1.7.1 Description

Le capteur utilise deux matériaux de haute qualité sur une plaque de 5,5*4,0 cm.
Le capteur est protégé contre les oxydations tout en optimisant la conductivité, la durée de vie ainsi que les performances du capteur.
La sortie du comparateur est nettoyée pour une bonne qualité de signal pour un courant de plus de 15mA. Le seuil est ajustable via potentiomètre [4].

Figure 1.10 : capteur de pluie [4].

1.7.2 Caractéristiques techniques

Le capteur fonctionne de 3,3V à 5V.

Le capteur dispose de 2 sorties: Numérique et analogique A0.

Le PCB du comparateur mesure 3.2 cm x 1.5 cm

Connexions:

1. VCC: 3-5V
2. GND
3. DO: Sortie numérique
4. AO: Sortie analogique

1.8 Module GPS NEO-6M

1.8.1 Présentation du module GPS NEO-6M

Le module GPS NEO-6M est illustré dans la figure ci-dessous. Il est livré avec une antenne externe et n'est pas livré avec des broches d'en-tête. Donc, vous devrez en obtenir et en souder.

Figure 1.11 : Module GPS NEO-6M [5].

- Ce module a une antenne externe et une EEPROM intégrée.
- Interface : RS232 TTL
- Alimentation : 3V à 5V
- Débit en bauds par défaut : 9600 bps
- Fonctionne avec des phrases NMEA standard

Le module GPS NEO-6M est également compatible avec d'autres cartes de microcontrôleur. Pour apprendre à utiliser le module GPS NEO-6M avec le Raspberry Pi, vous pouvez lire : Système d'alerte par e-mail en cas de changement d'emplacement avec Raspberry Pi et module GPS [5].

1.8.2 Caractéristiques techniques

Processus PCB Immersion Gold. Le signal est plus stable.
Puce principale: NEO-6M
Code C / A, flux de 1,023 MHz
Bandes de réception: L1 [1575,42 MHz]
Performances de positionnement
Plan 2D: 3m [moyenne]
Le 2D plat: 2m [moyenne], auxiliaire WAAS.
Dérive: <0,02 m / s
Précision de synchronisation: 1us
Système de coordonnées de référence: WGS-84
Altitude maximale: 18 000 m

Vitesse maximale: 515 m / s

Accélération: <4g

1.8.3 Propriétés électriques

Suivi sensibilité: -161dBm

Sensibilité d'acquisition: -148dBm

Temps de démarrage à froid: 38s [moyenne]

Démarrage à chaud: 35s [moyenne]

Temps de démarrage à chaud: 1s [moyenne]

Temps de récupération: 0,1 s [moyenne]

Température de fonctionnement:

-40 C à +80 C

Large plage de tension: l'alimentation principale de + 3,5 V ~ +5,5 V, puissance RTC possédée.

Tableau 1 : paramètres de module GPS NEO_6 M.

Paramètres	
Type de récepteur	50 canaux, code GPS L1 (1575.42Mhz) C / A
SBAS	WAAS / EGNOS / MSAS
Précision de la position horizontale	2,5 mCEP (SBAS : 2,0 mCEP)
Taux de mise à jour de la navigation	5 Hz maximum (1 Hz par défaut)
Temps de capture	Démarrage à froid : 27 s (le plus rapide) start Démarrage à chaud : 1 s
Sensibilité de suivi et de navigation	-161dBm
Protocole de communication	NMEA (par défaut) / UBX Binary
Vitesse de transmission série	4800, 9600 (par défaut), 19200, 38400, 57600, 115200, 230400
Température de fonctionnement	-40 °C ~ 85 °C
Tension de fonctionnement	2,7 V ~ 5,0 V (entrée d'alimentation via VCC)
Courant de fonctionnement	45mA
Impédance TXD / RXD	510Ohms

1.9 Capteur de flamme

1.9.1 Définition

Un capteur de flamme est un appareil conçu pour détecter et réagir à la présence d'une source d'incendie. Il est une partie d'un équipement de sécurité pour protéger notre environnement. Il existe différents types de méthodes de détection de flamme. Certains d'entre eux sont: Détecteur ultraviolet, détecteur proche infrarouge, détecteur infrarouge (IR), caméras thermiques infrarouges, détecteur UV / IR,… etc.

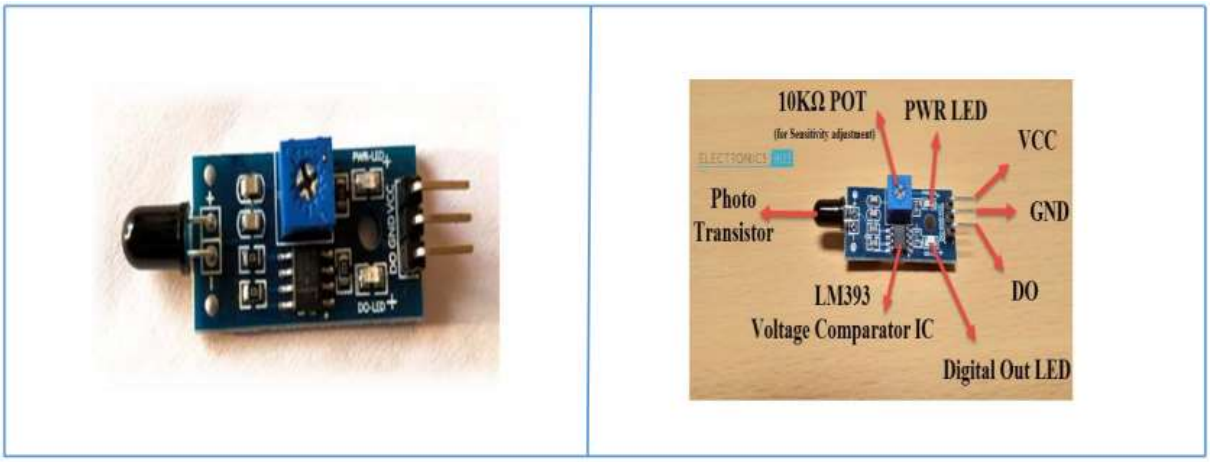


Figure 1.12 : capteur de flamme [6].

1.9.2 Principe de fonctionnement de capteur de flamme

Lorsque le feu brûle, il émet une petite quantité de lumière infrarouge, cette lumière sera reçue par la photodiode (récepteur IR) sur le module capteur. Ensuite, nous utilisons un amplificateur opérationnel pour vérifier le changement de tension aux bornes du récepteur infrarouge, de sorte que si un incendie est détecté, la broche de sortie (DO) donnera 0 V (FAIBLE) et s'il n'y a pas de feu, la broche de sortie sera de 5 V (HAUTE) .

1.9.3 Caractéristiques techniques de capteur de flamme

Le Capteur le plus sensible pour des longueurs d'onde infrarouge de la flamme entre 760 nm et 1100 nm. Il a deux sorties:

AO: sortie analogique, signaux de tension de sortie sur la résistance thermique en temps réel,
DO: lorsque la température atteint à un certain seuil, signaux de seuil de sortie haute et basse est réglable par potentiomètre.

Capteur de détection de 60 degrés Convient pour projet Arduino DIY
Tension: DC 3 ~ 5.5V
Matériel: PCB
Couleur: bleu + rouge + gris argent
Dimension du produit: 3,5 x 1,5 x 1,2 cm
Dimension de l'emballage: 80 x 41 x 15mm.

1.10 Afficheur LCD 2*16

1.10.1 Définition

LCD est l'abréviation du terme anglais "Liquid Crystal Display" qui signifie en français "Écran à cristaux liquides". D'où afficheur LCD.

L'afficheur LCD est en particulier une interface visuelle entre un système (projet) et l'homme (utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploiter par l'utilisateur d'un système [7].

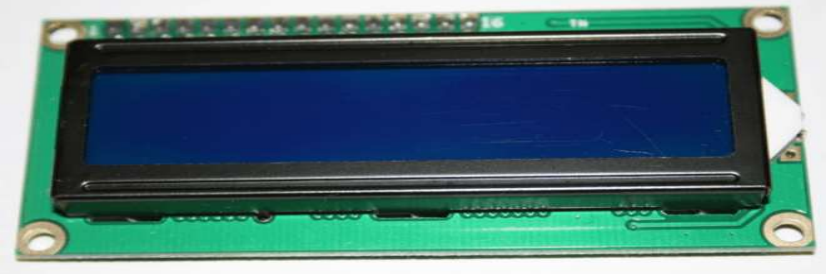


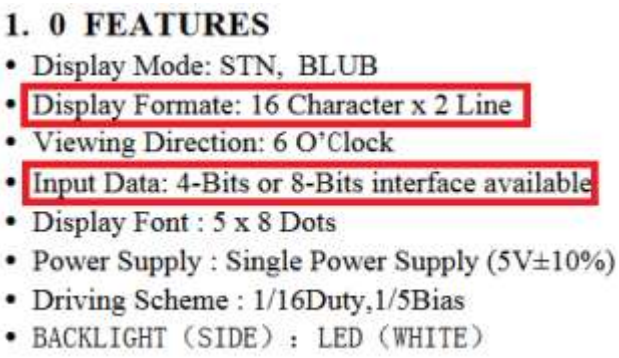
Figure 1.13 : le modèle 1602A (V2.0) [7].

Nous utiliserons le modèle 1602A (V2.0) que l'on trouve dans le kit Arduino (2012).

1.10.2 Les caractéristiques

La première information à connaître est le nombre de caractère affichable par ligne. Pour ce modèle, c'est 16 caractères sur deux lignes soit au total 32 caractères. De toute évidence, on retrouve cette information dans le datasheet sous la forme 16 X 02. Mais aussi dans la référence 1602 A.

On recueille également l'information du mode de transmission de données sur quatre (4) ou huit (8) bits [7].



1. 0 FEATURES

- Display Mode: STN, BLUB
- Display Formate: 16 Character x 2 Line
- Viewing Direction: 6 O'Clock
- Input Data: 4-Bits or 8-Bits interface available
- Display Font : 5 x 8 Dots
- Power Supply : Single Power Supply (5V±10%)
- Driving Scheme : 1/16Duty,1/5Bias
- BACKLIGHT (SIDE) : LED (WHITE)

Figure 1.14 : Nombre de caractères et bits selon le Datasheet [7].

1.10.3 Caractéristiques globales de l'afficheur LCD

En outre, les informations suivantes à connaître sont les tensions d'exploitations, caractéristiques électriques et mécaniques de l'afficheur LCD [7].

Tableau 2 : Tableaux des tensions d'exploitations, caractéristiques électriques et mécaniques [7].

2.0 ABSOLUTE MAXIMUM

Item	Symbol	Min.	Max.	Unit
Power Supply for logic	Vdd	-0.3	+7.0	V
Power supply for LCD Drive	Vlcd	Vdd-10.0	Vdd+0.3	V
Input Voltage	Vi	-0.3	Vdd+0.3	V
Operating Temperature	Ta	0	+50	℃
Storage Temperature	Tstg	-10	+60	℃

3.0ELECTRICAL CHARACTERISTICS

(Ta=25℃;Vdd=3.0V±10%,otherwise specified)

Item	Symbol	Test Condition	Min.	Typ.	Max.	Unit
Power Supply for Logic	Vdd	--	4.7	5.0	5.5	V
Operating Voltage for LCD	Vdd-Vo	--	--	5.0	--	V
Input High voltage	Vih	--	2.2	--	Vdd	V
Input Low voltage	Vil	--	-0.3	--	0.6	V
Output High voltage	Voh	-Ioh=0.2mA	2.4	--	--	V
Output Low voltage	Vol	Iol=1.2mA	--	--	0.4	V
Power supply current	Idd	Vdd=3.0v	--	1.1	--	mA

4.0 MECHANICAL PARAMETERS

Item	Description	Unit
PCB Dimension	80.0*36.0*1.6	mm
View Dimension	69.5*14.5	mm

1.10.4 - Le brochage

Le tableau ci-dessous représente bien le rôle de chaque broche et définit le brochage de l'afficheur LCD.

Tableau 3: Descriptive du brochage [7].

5. 0 PIN ASSIGNMENT

No.	Symbol	Level	Function	
1	Vss	--	0V	Power Supply
2	Vdd	--	+5V	
3	V0	--	for LCD	
4	RS	H/L	Register Select: H:Data Input L:Instruction Input	
5	R/W	H/L	H--Read L--Write	
6	E	H,H-L	Enable Signal	
7	DB0	H/L	Data bus used in 8 bit transfer	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L	Data bus for both 4 and 8 bit transfer	
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	BLA	--	BLACKLIGHT +5V	
16	BLK	--	BLACKLIGHT 0V-	

Premièrement, les broches 1, 2 et 3 sont dédiées à l'alimentation, les broches 1 et 2 pour l'alimentation générale. La broche 3 pour le branchement d'un potentiomètre qui contrôle le contraste de l'afficheur.

Deuxièmement, les broches 4, 5 et 6 pour le pilotage de la transmission des données. Elles pilotent l'écriture ou la lecture de données.

Troisièmement, les broches de 7 à 14 pour le transfert des données elles-mêmes.

Et pour finir, les broches 15 et 16 pour l'alimentation du rétro-éclairage.

Grâce à ces informations nous connaissons le brochage de l'afficheur. De plus nous avons en appui un schéma de raccordement.

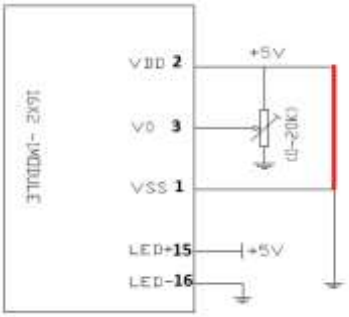


Figure 1.15 : Schéma de câblage [7].

Ce schéma illustre bien les branchements de l'alimentation principale et de ces auxiliaires. Nous avons ajouté le repérage des broches. Il y a une erreur signalée en rouge. Ne pas câbler ce fil [7].

Conclusion

Dans ce chapitre, nous avons présenté les différents types des stations météo, Ensuite nous avons essayé de découvrir d'un cote technique les différents capteurs et modules utilisés.

Dans le chapitre suivant nous montrerons le montage de chaque capteur et Nous afficherons les données obtenues séparément, puis nous afficherons la réalisation finale et le processus de transmission des données

CHAPITRE 2 :
TESTS DES COMPOSANTS ELECTRONIQUES

Introduction

Après avoir identifié les différents composants et les logiciels entrant dans ce projet et leurs critères de choix et avoir soigneusement étudié leurs caractéristiques techniques, la suite logique de cette étape va être les tests et des essais pratiques.

Ce chapitre va inclure le montage des différents capteurs et modules afin de faire les tests et les essais pour mesurer les performances de notre système.

L'interconnexion des différentes parties du système est assurée par un élément central qui est la carte NodeMcu ESP8266, cette dernière va permettre de contrôler le flux d'information entre ses périphériques.

La dernière étape consistera à intégrer notre système à télécharger les résultats obtenus via les capteurs sur l'interface Arduino, puis à les envoyer sur un site local via Wifi, ce qui permet à l'utilisateur de visualiser les données à tout moment et de pouvoir les partager

2.1 Le branchement et le test du capteur DHT11 avec la carte NodeMcu ESP8266

2.1.1 Schéma de montage

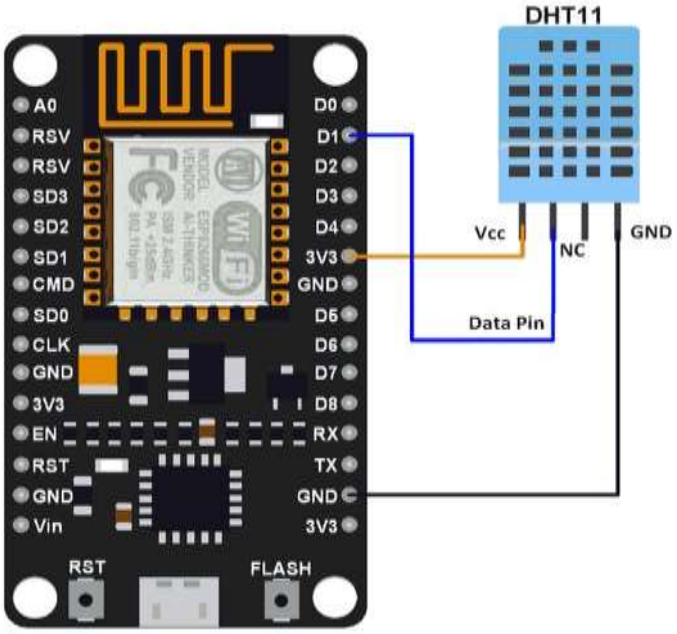


Figure 2.16 : schéma fritzing de montage

Connecter le module capteur de humidité et température directement à la carte NodeMCU en connectant la broche GND de capteur à la broche GND de NodeMCU et la broche VCC à la broche 3v3. Connectez également la sortie S à D1.

2.1.2 Code Arduino

- Lors de la connexion du capteur avec la carte, nous devons télécharger une bibliothèque <dht.h> depuis le référentiel **:** https://github.com/adafruit/DHT-sensor-library [8] et l'installer dans le fichier de bibliothèque Arduino.

```
1 #include <dht.h>
2 #include <ESP8266WiFi.h>
3 dht DHT;
```

- Puis inclure aussi la bibliothèque <ESP8266WiFi.h>, définir le branchement de la sortie S de capteur avec l'entrée D1 de Nodemcu.

- Dans la fonction setup (), commencez par le débit en bauds de 9600.

```
5 void setup(){
6   Serial.begin(9600);
7 }
```

- Dans la fonction void loop (), Après la lecture de la sortie de capteur on affiche les données de température et d'humidité avec un délai de 2000ms.

```
 8 void loop(){
 9   int chk = DHT.readl1(DHT11_PIN);
10
11   Serial.print("Temperature = ");
12   Serial.println(DHT.temperature);
13   Serial.print("Humidity = ");
14   Serial.println(DHT.humidity);
15
16
17
18   delay(2000);
19 }
```

2.1.3 Affichage sur le moniteur série

- Dans le moniteur série, on pourra voir les résultats après avoir télé verser le programme comme la montre la figure :

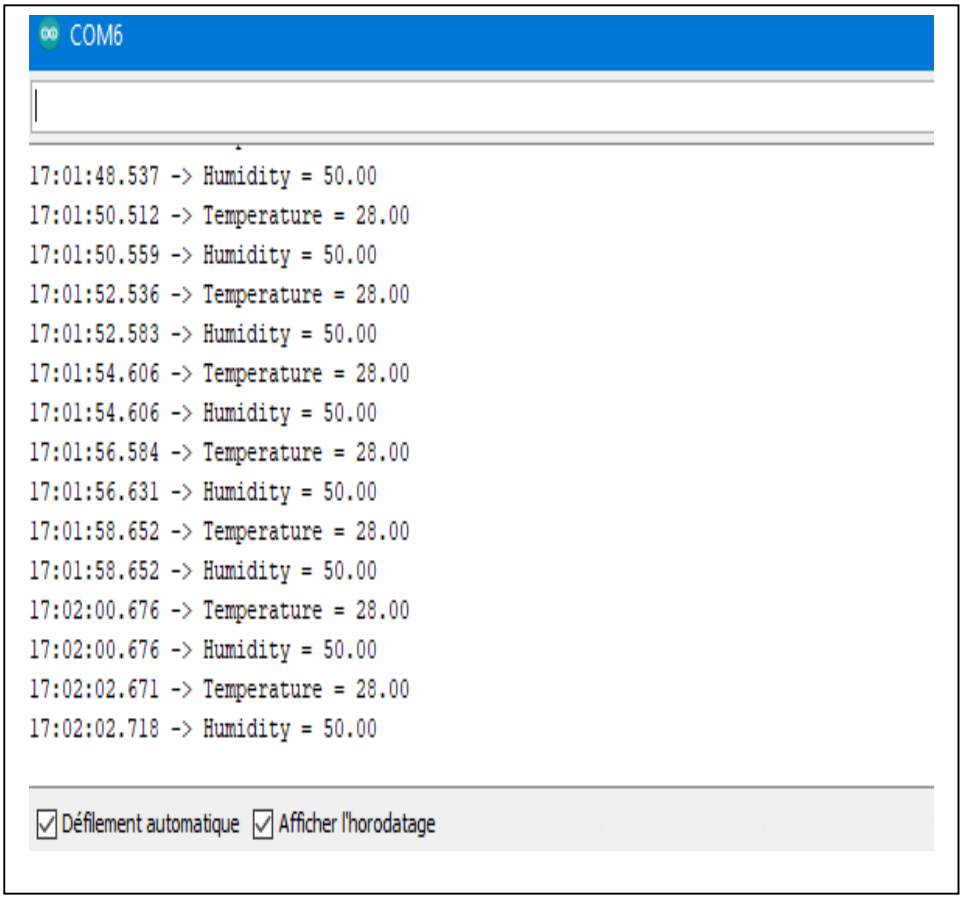


Figure 2.17 : Affichage de l'humidité et la température sur le moniteur série.

2.2 Le branchement et le test du capteur de pluie avec la carte NodeMcu ESP8266

2.2.1 Schéma de montage

Connectez le module capteur de pluie directement à la carte NodeMCU en connectant la broche GND de capteur à la broche GND de NodeMCU et la broche VCC à la broche 3v3. Connectez également D0 à D0 et A0 à A0.

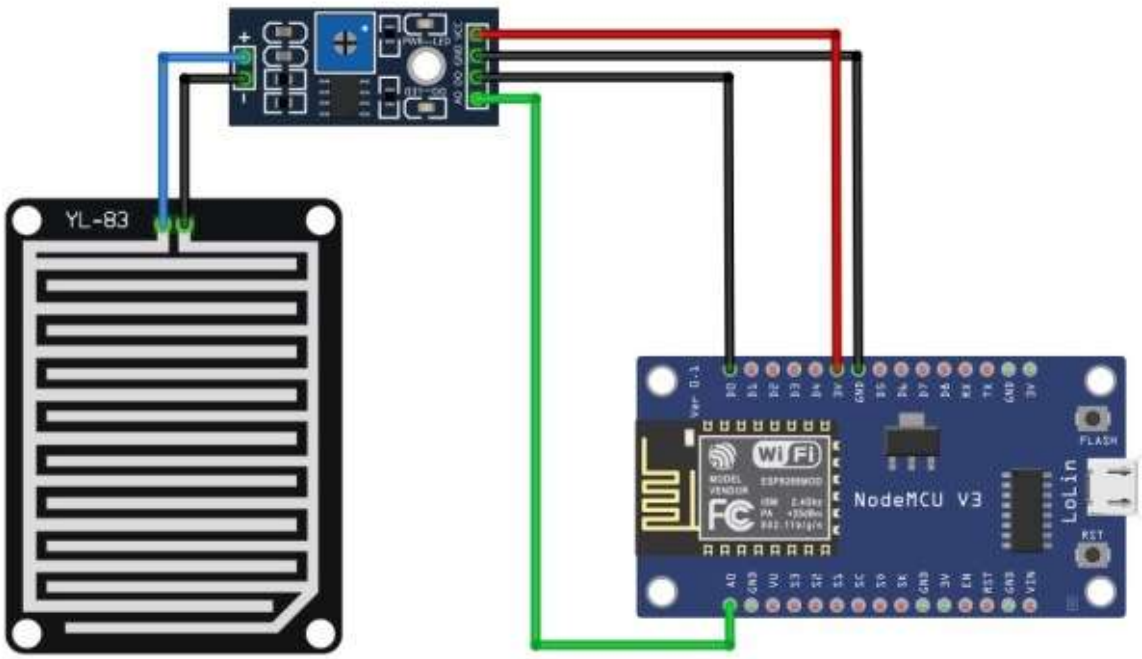


Figure 2.18 : schéma fritzing de branchement du capteur de pluie avec la carte NodeMcu ESP8266

2.2.2 Code Arduino

- On va commencer par définir les variables des entrées de capteur analogique et numérique selon leurs branchements avec la carte node mcu, avec une valeur analogique d'eau.

```
1  #include <ESP8266WiFi.h>
2  const int capteur_D = 16;
3  const int capteur_A = A0;
4
5  int val_analogique;
6
```

- Par la suite, dans la fonction setup (), on commence par le débit en bauds de 9600 et on définit A0 et D0 comme des entrées.

```
7  void setup()
8  {
9    pinMode(capteur_D, INPUT);
10   pinMode(capteur_A, INPUT);
11   Serial.begin(9600);
12 }
```

- Dans la fonction void loop (), Après la lecture de la sortie digitale de capteur de capteur grâce à une boucle on détecte s'il y a de la pluie ou pas et avec la sortie analogique on prend une valeur de niveau d'eau et on affiche les résultats avec un délai de 1000ms.

```
14 void loop()
15 {
16 if(digitalRead(capteur_D) == LOW)
17   {
18     Serial.println("Digital value : il pleut");
19     delay(10);
20   }
21 else
22   {
23     Serial.println("Digital value : pas de pluie");
24     delay(10);
25   }
26 val_analogique=analogRead(capteur_A);
27  Serial.print("Analog value : ");
28  Serial.println(val_analogique);
29  Serial.println("");
30   delay(1000);
31 }
```

2.2.3 Affichage sur le moniteur série

- Dans le moniteur série, on pourra voir les résultats après avoir télé verser le programme comme la montre la figure :

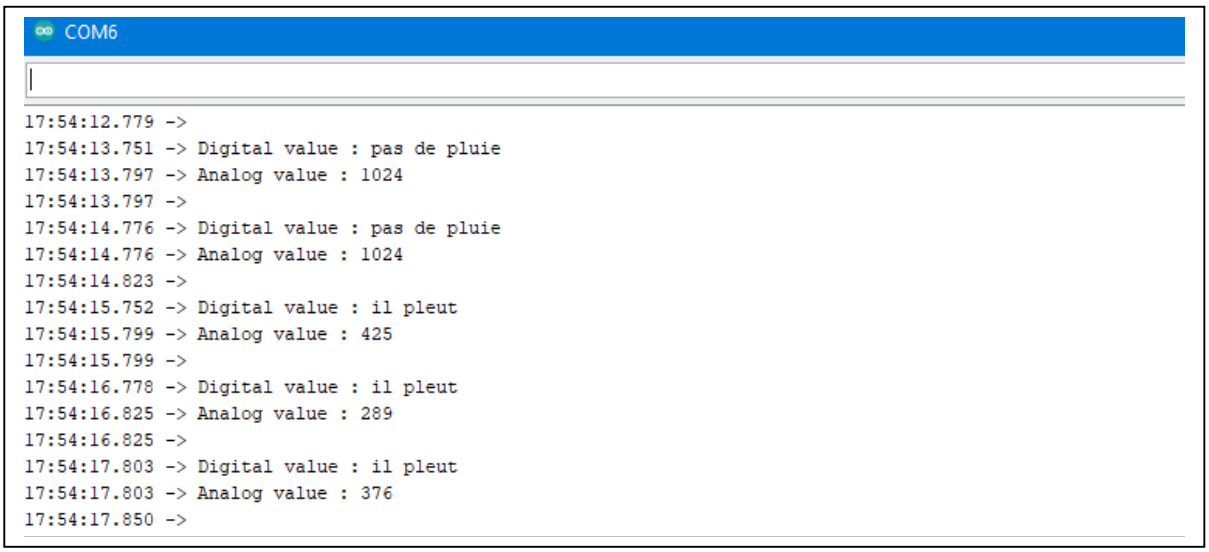


Figure 2.19 : Affichage de pluie et mise en échelle sur le moniteur série.

2.3 Le branchement et le test du capteur de flamme avec la carte NodeMcu ESP8266

2.3.1 Schéma de montage

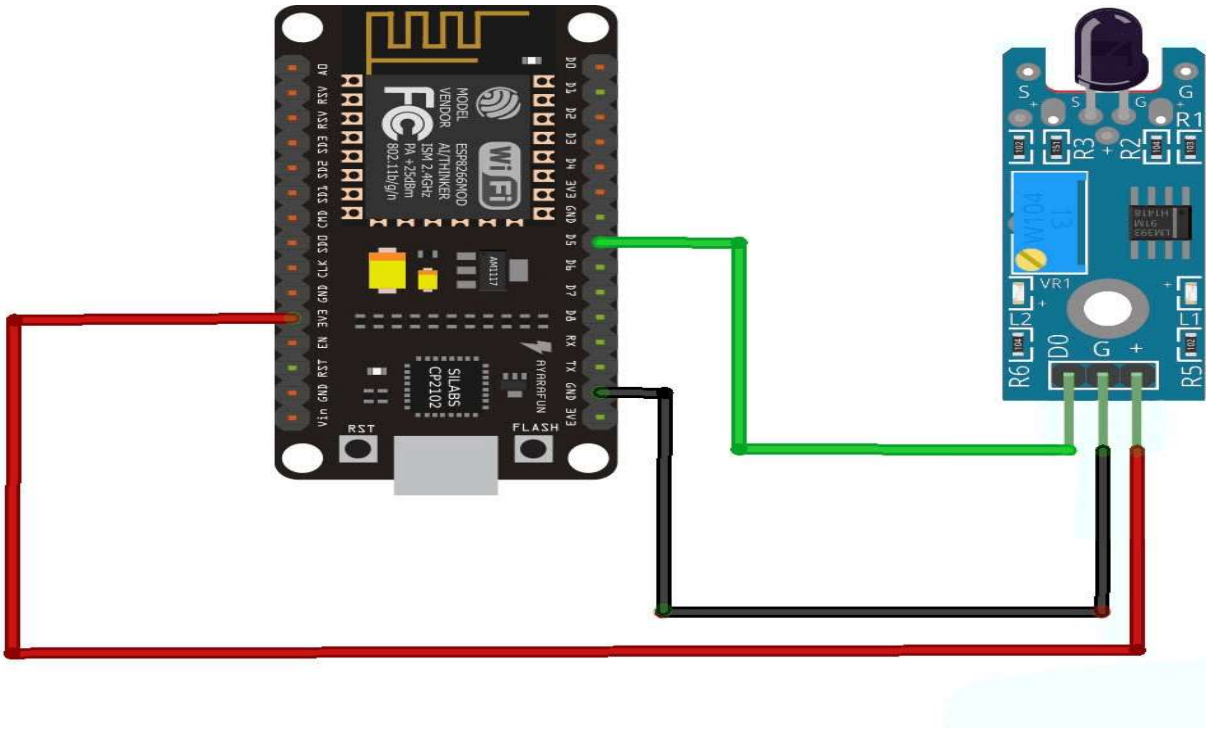


Figure 2.20 : schéma fritzing de branchement du capteur de flamme avec la carte NodeMcu ESP8266

Connectez le module de capteur de flamme directement à la carte NodeMCU en connectant la broche GND de ce dernier à la broche GND de NodeMCU et la broche VCC à la broche 3v3. Connectez également A0 à D0.

2.3.2 Code Arduino

- On va commencer par définir la variables d'entrée digitale de selon le branchement avec la carte node mcu, avec une tension haute sur cette entée.

```
#include <ESP8266WiFi.h>
int flamePin = 14;
int flame = HIGH;
```

- Par la suite, dans la fonction setup (), on commence par le débit en bauds de 9600 et on définit D0 comme des entrée.

```
4  void setup() {
5    pinMode(flamePin, INPUT);
6
7
8    Serial.begin(9600);
9  }
```

- Dans la fonction void loop (), Après la lecture de la sortie digitale de capteur de capteur grâce à une boucle on détecte s'il y a de la flamme ou pas et on affiche les résultats avec un délai de 2000ms.

  Programme comme la montre la figure :

```
10 void loop() {
11   flame = digitalRead(flamePin);//Readd the data gien by the flame sensor
12
13   if (flame== LOW)  //if it is low
14   {
15     Serial.println("cas : Attention!! flamme");
16     delay(10);
17     }
18
19       else{
20      Serial.println("cas : pas de flamme");
21      delay(10);
22     }
23 delay(2000);
24 }
```

2.3.3 Affichage sur le moniteur série

- Dans le moniteur série, on pourra voir les résultats après avoir télé verser le

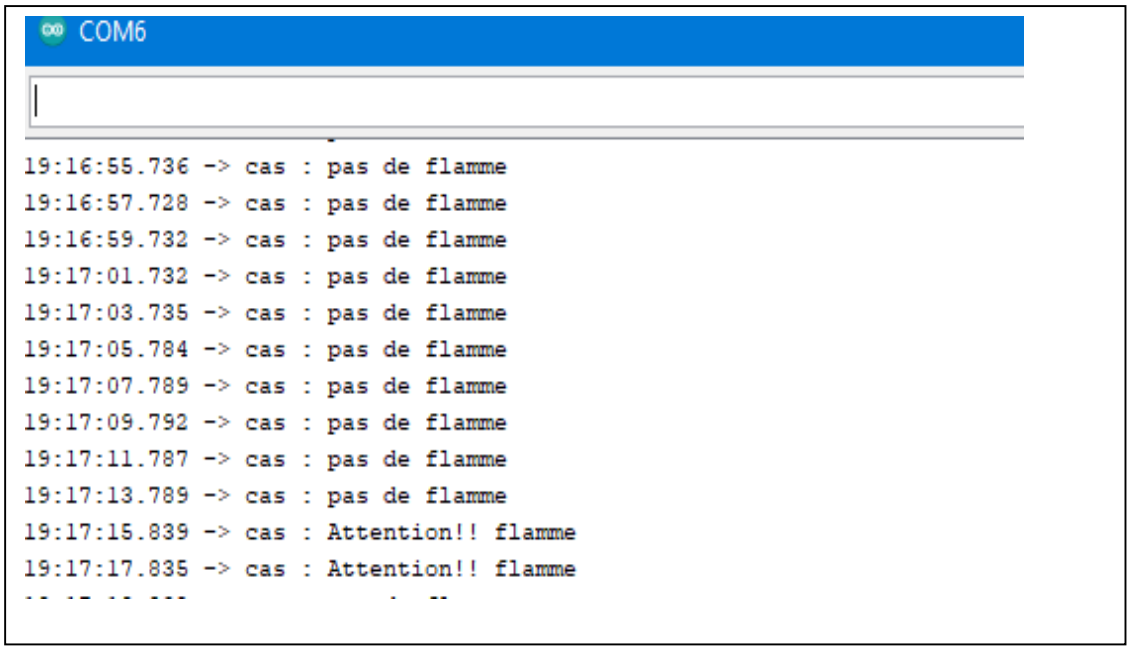


Figure 2.21 : Affichage de flamme sur le moniteur série.

2.4 Le branchement et le test du LCD 2*16 avec la carte NodeMcu ESP8266

2.4.1 Schéma de montage

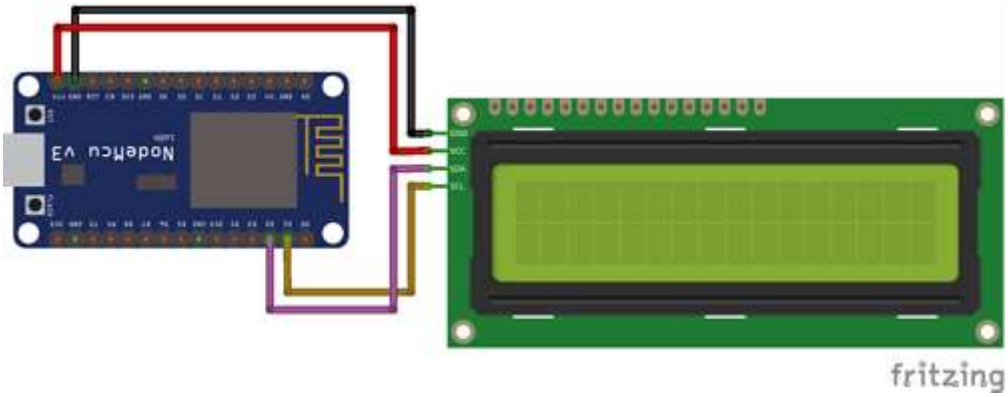


Figure 2.22 : schéma frtizing de branchement et du LCD 2*16 avec la carte NodeMcu ESP8266.

Connecter l'ecran LCD 2*16 directement à la carte NodeMCU en connectant la broche GND de ce dernier à la broche GND de NodeMCU et la broche VCC à la broche 5v(Vin). Connectez également SDA à D2 et SCL à D1.

2.4.2 Code Arduino

- Lors de la connexion de l'écran avec la carte nodemcu, nous devons télécharger une bibliothèque LiquidCrystal_I2C.h et l'installer dans le fichier Arduino. On définit les entrées de l'écran dans notre code selon le branchement :

```
1  #include "Wire.h"
2  #include "LiquidCrystal_I2C.h"
3  LiquidCrystal_I2C lcd(0x27, 16, 2);
4
```

- Par la suite, dans la fonction setup (), on lance le LCD et on le mets en rétroéclairage.

```
5  void setup()
6  {
7    lcd.begin();
8
9    lcd.backlight();
10
11 }
```

- Dans la fonction void loop (), on affiche un exemple qui va t'etre en premier« Bienvenue à IAES » et après un délai de 3000 ms il affiche « station météo M2 Telecom ».

```
13 void loop(){
14     lcd.setCursor(1, 0);
15    lcd.print("° bienvenue   ° ");
16    lcd.setCursor(1, 1);
17    lcd.print("%% a IAES %%");
18    delay(3000);
19    lcd.setCursor(1,0);
20    lcd.print("Station meteo");
21    lcd.setCursor(1,1);
22    lcd.print("M2 Telecom");
23    delay(3000);
24
25    }
```

2.4.3 Affichage sur l'écran LCD

- Après le téléversement de code on obtient les résultats suivants :

Figure 2.23 : Affichage des résultats sur l'écran LCD.

2.5 Le branchement et le test du module gps 6m NEO avec la carte NodeMcu ESP8266

2.5.1 Schéma de montage

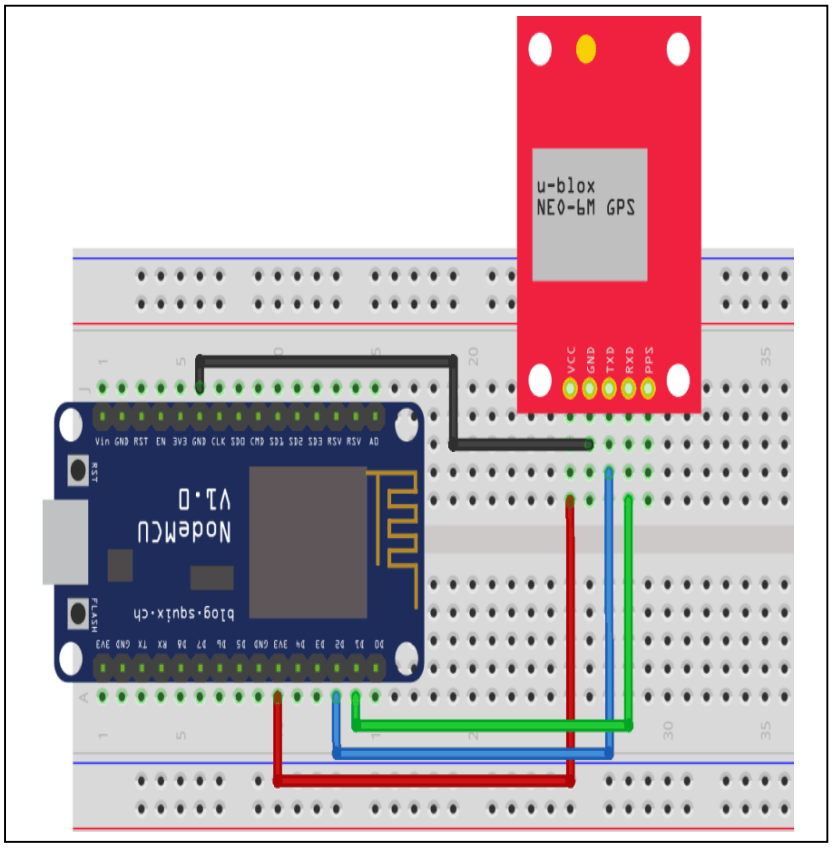


Figure 2.24 : schéma fritzing de branchement de module gps 6m NEO avec la carte NodeMcu ESP8266.

2.5.2 Code Arduino

- Le programme commence par inclure la bibliothèque TinyGPS++ nouvellement installée et la bibliothèque SoftwareSerial. Ensuite, ces broches NodeMcu sont déclarées auxquelles le module GPS NEO-6M est connecté et une variable qui stocke le débit en bauds GPS par défaut.
- La création d'un objet TinyGPSPlus aidera à accéder à des fonctions spéciales liées à la bibliothèque. Après cela, nous créons un port série logiciel appelé gpsSerial à travers lequel nous parlerons au module.

```
test2gps §
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
TinyGPSPlus gps;
int RXPin = 5, TXPin = 4;
 int GPSBaud = 9600;
SoftwareSerial gpsSerial(RXPin,TXPin);
 float latitude,longitude;
```

- Par la suite, dans la fonction setup (), nous initions la communication série avec le PC ainsi que le module GPS.

```
void setup(){
  Serial.begin(115200);
   gpsSerial.begin(GPSBaud);
}
```

- Dans la fonction void loop (), la fonction displayInfo() est appelée et imprime les informations de localisation (latitude, longitude et altitude) et UTC (date et heure) sur le moniteur série, chaque fois qu'une nouvelle phrase NMEA est encodée correctement.
- Si 5000 millisecondes s'écoulent et qu'aucun caractère n'arrive au port série du logiciel, un message d'erreur "Aucun GPS détecté" est imprimé.

```
void loop(){
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  // If 5000 milliseconds pass and there are no characters coming in
  // over the software serial port, show a "No GPS detected" error
  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println("No GPS detected");
    while(true);
  }
}
```

- Une fois le GPS est detecte nous allons commencer a afficher les resultats de longitude latitude heure et date.

```
void displayInfo()
{
  if (gps.location.isValid())
  {
    Serial.print("Latitude: ");
    Serial.println(gps.location.lat(), 6);
    latitude= gps.location.lat();
    Serial.print("Longitude: ");
    Serial.println(gps.location.lng(), 6);
    longitude= gps.location.lng();
    Serial.print("Altitude: ");
    Serial.println(gps.altitude.meters());
  }
  else
  {
    Serial.println("Location: Not Available");
  }
   Serial.print("Date: ");
  if (gps.date.isValid())
  {
    Serial.print(gps.date.month());
    Serial.print("/");
    Serial.print(gps.date.day());
    Serial.print("/");
    Serial.println(gps.date.year());
  }
  else
```

```
  }
  else
  {
    Serial.println("Not Available");
  }

  Serial.print("Time: ");
  if (gps.time.isValid())
  {
    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(":");
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(":");
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.print(gps.time.second());
    Serial.print(".");
    if (gps.time.centisecond() < 10) Serial.print(F("0"));
    Serial.println(gps.time.centisecond());
  }
  else
  {
    Serial.println("Not Available");
  }

  Serial.println();
  Serial.println();
  delay(10);
}
```

2.5.3 Affichage sur le moniteur série

- Une fois le code est teléversé le GPS peut prendre quelques minutes pour capter le résultat sera comme la figure suivante :

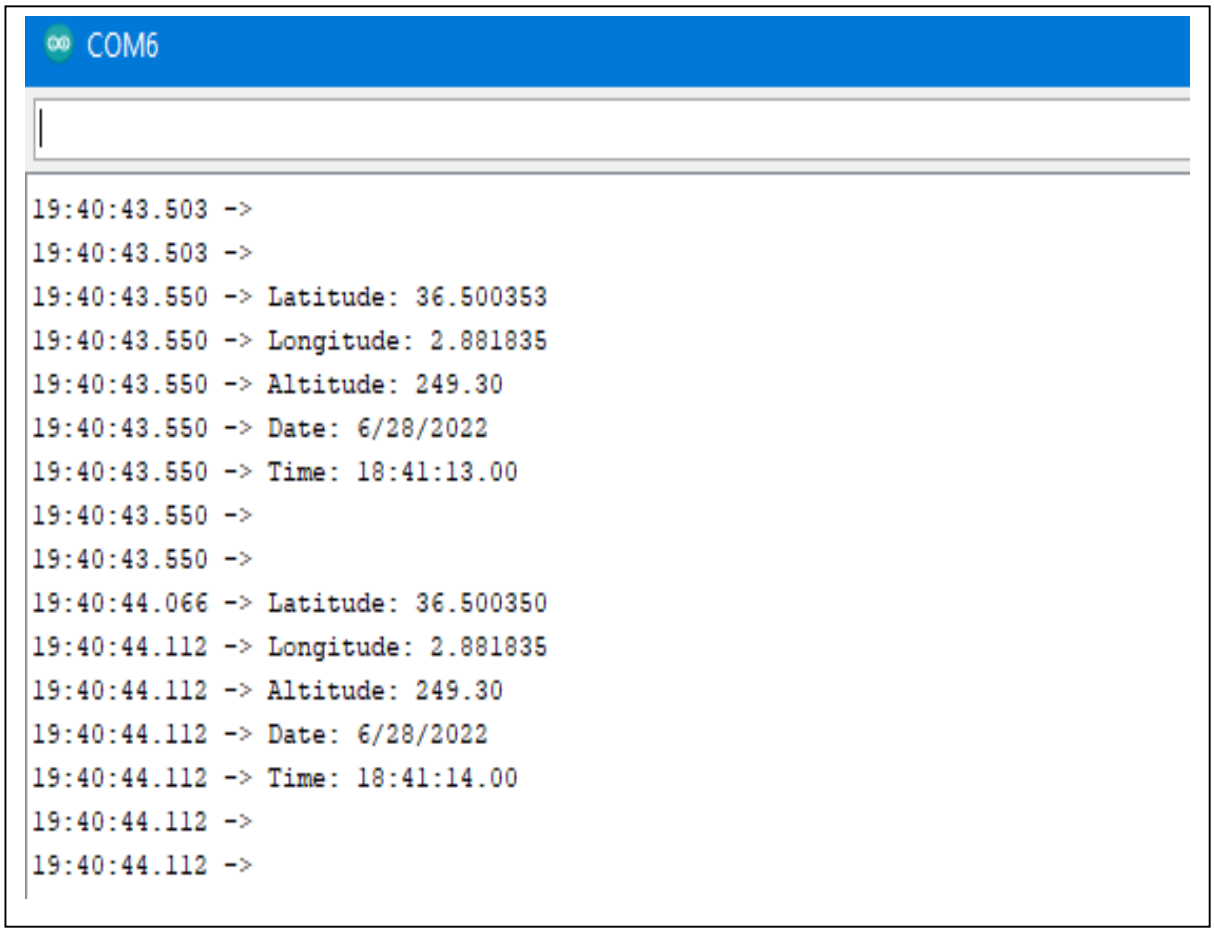


Figure 2.25 : Affichage de latitude, longitude, date et heure sur le moniteur série.

Conclusion

Dans ce chapitre nous avons mis en avant les outils utilisés pour l'aboutissement de notre projet, après avoir énuméré les outils nous avons pu remarquer que notre projet à demander des compétences et connaissances dans plusieurs domaines tels que l'électronique et l'informatique.

Nous avons donc pu dans ce chapitre lier les différents capteurs et notre écran à notre élément central la carte nodemcu esp8266, et nous allons par la suite faire collecter tous les éléments Et créer un site local qui va nous permettre de visualiser en temps réel grâce à une interface D'affichages bien définis.

CHAPITRE 3 :
CONCEPTION ET REALISATION PRATIQUE

Introduction

Après avoir testé les différents composants et les avoir connecter d'une façon individuelle à l'élément central ESP8266, la suite logique de cette étape va être la connexion totale et finale de ces éléments.

Ce chapitre va inclure le montage final des différents capteurs et modules afin de réaliser le système électronique final.

L'interconnexion des différentes parties du système est assurée par un élément central qui est la carte NodeMcu ESP8266 qui possède une fonctionnalité Wifi qui lui permet de devenir un serveur Web grâce à des étapes bien définies que nous allons découvrir.

La dernière étape consistera donc à afficher sur un site local les différentes données des capteurs et de module GPS stockées dans l'ESP8266 grâce à un code HTML associé à un code de surveillance ans le même dossier.

3.1 Langages de programmation utilisés pour notre site local

3.1.1 HTML (Hyper Text Mark-Up Language)

Figure 3.26 : Logo HTML [9].

Le HTML (« Hyper Text Mark-Up Language ») est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents [9].

Le langage HTML permet notamment la lecture de documents sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL.

On appelle World Wide Web (noté WWW) ou tout simplement Web (mot anglais signifiant toile) la "toile virtuelle" formée par les différents documents (appelés « pages web ») liés entre eux par des hyperliens.

Les pages web sont généralement organisées autour d'une page d'accueil, jouant un point central dans la navigation à l'aide des liens hypertextes. Cet ensemble cohérent de pages web liées par des liens hypertextes et articulées autour d'une page d'accueil commune est appelée site web.

Le Web est ainsi une énorme archive vivante composée d'une myriade de sites web proposant des pages web pouvant contenir du texte mis en forme, des images, des sons, des vidéo, etc [9].

3.1.2 CSS (Présentation des feuilles de style)

Figure 3.27 : Logo CSS [10].

Le concept de feuilles de style est apparu en 1996 avec la publication par le W3C d'une nouvelle recommandation intitulée « Cascading StyleSheets » (feuilles de style en cascade), notée CSS.

Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte. Il est ainsi possible de créer un groupe de titres en police Arial, de couleur verte et en italique.

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier !

Elles sont appelées « feuilles de style en cascade » (en anglais « Cascading Style Sheets ») car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade [10].

Les feuilles de style permettent notamment :

- D’obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style :
- De permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes
- Une plus grande lisibilité du HTML, car les styles sont définis à part.
- Des chargements de page plus rapides, pour les mêmes raisons que précédemment.
- Un positionnement plus rigoureux des éléments.

3.1.3 Langage JavaScript

Figure 3.28 : Logo JavaScript.

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.

JavaScript a été mis au point par Netscape en 1995. A l'origine, il se nommait Live Script et était destiné à fournir un langage de script simple au navigateur Netscape Navigator 2. Il a à l'époque longtemps été critiqué pour son manque de sécurité, son développement peu poussé et

l'absence de messages d'erreur explicites rendant dure son utilisation. Le 4 décembre 1995, suite à une association avec le constructeur Sun, Netscape rebaptise son langage Javascript (un clin d'oeil au langage Java développé par Sun). A la même époque, Microsoft mit au point le langage Jscript, un langage de script très similaire. Ainsi, pour éviter des dérives de part et d'autre, un standard a été défini pour normaliser les langages de script, il s'agit de l'ECMA 262, créé par l'organisation du même nom (ECMA, European Computer Manufactures Association).

3.1.4 Le langage JSON en informatique

JSON (JavaScript Objet Notation) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple).

Fondé sur un sous-ensemble de Javascript, JSON est un format texte totalement indépendant de tout langage. Pourtant, les conventions utilisées ne surprendront pas les codeurs familiers aux langages descendant du C tels que Javascript, Python, Pearl ou d'autres.

Autrement dit, il fonctionne un peu comme le XML (mais en moins développé) et facilite la structuration des informations présentes dans un document informatique. Comme il sert simplement à fluidifier l'échange de données, il n'est pas supposé contenir de commentaires, par exemple, ce qui le distingue d'un langage informatique à part entière. Toutefois, certaines bibliothèques en acceptent, s'ils sont écrits en JavaScript.

Le JSON fait partie des langages compréhensibles aussi bien par un esprit humain que par une machine. D'ailleurs, son apprentissage est facile et intuitif. Cependant, il reste très limité et ce qui le rend moins fiable et peu résistant en termes de sécurité [11].

3.2 Fonctionnalité wifi

3.2.1 Introduction sur la bibliothèque Wi-Fi pour ESP8266

La bibliothèque Wi-Fi pour ESP8266 a été développée sur la base du SDK ESP8266 , en utilisant les conventions de dénomination et la philosophie de fonctionnalité globale de la bibliothèque Wifi Arduino . Au fil du temps, la richesse des fonctionnalités Wi-Fi portées du SDK ESP8266 vers esp8266 / Arduino a dépassé la bibliothèque Wifi Arduino et il est devenu évident que nous aurions besoin de fournir une documentation séparée sur ce qui est nouveau et supplémentaire [12].

3.2.2 Connexion d'un ESP8266 a réseau Wifi

Pour connecter le module ESP au Wi-Fi (comme pour connecter un téléphone portable à un point d'accès), On aura besoin que de quelques lignes de code :

```
#include <ESP8266WiFi.h>

void setup()
{
Serial.begin(115200);
Serial.println();

WiFi.begin("network-name", "pass-to-network");

Serial.print("Connecting");
while (WiFi.status() != WL_CONNECTED)
    {
    delay(500);
    Serial.print(".");
    }
Serial.println();

Serial.print("Connected, IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {}
```

Dans cette ligne, on remplace par le nom et le mot de passe du réseau Wi-Fi auquel nous souhaitons connecter. Ensuite, on télécharge cette esquisse sur le module ESP et après l'ouverture moniteur série. Le résultat sera le suivant :

```
WiFi.begin("network-name", "pass-to-network")network-namepass-to-network
```

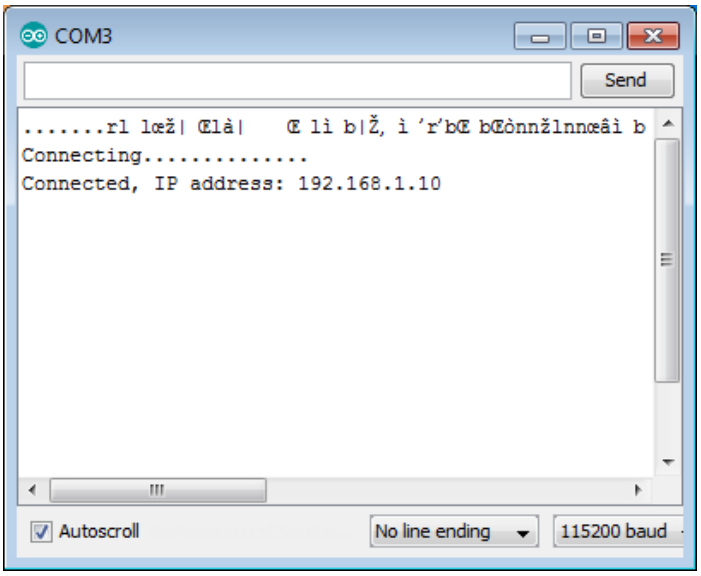


Figure 3.29 : Affichage d'adresse IP sur le moniteur série.

3.2.3 Explication de lignes de code

- Dans la première ligne du croquis, nous incluons la bibliothèque ESP8266WiFi . Cette bibliothèque fournit des routines Wi-Fi spécifiques à ESP8266 que nous appelons pour nous connecter au réseau. `#include <ESP8266WiFi.h>`
- La connexion réelle au Wi-Fi est initialisée en appelant :

```
WiFi.begin("network-name", "pass-to-network");
```

- Le processus de connexion peut prendre quelques secondes et nous vérifions si cela s'est terminé dans la boucle suivante :

```
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
```

- La `while()` boucle continuera à boucler tant que `WiFi.status()` n'est pas `WL_CONNECTED`. La boucle ne se terminera que si l'état passe à `WL_CONNECTED`.
- La dernière ligne imprimera alors l'adresse IP attribuée au module ESP par DHCP :

```
Serial.println(WiFi.localIP());
```

3.2.4 Les différents modes de fonctionnement de l'ESP8266

Les modules ESP8266 peuvent fonctionner comme une station, nous pouvons donc le connecter au réseau Wi-Fi. Il peut également fonctionner comme point d'accès logiciel (soft-AP), pour établir son propre réseau Wi-Fi. Lorsque le module ESP8266 fonctionne comme un point d'accès logiciel, nous pouvons connecter d'autres stations au module ESP. L'ESP8266 est également capable de fonctionner à la fois comme station et comme point d'accès logiciel. Cela offre la possibilité de construire par exemple des réseaux maillés [12].

Figure 3.30 : ESP8266 en mode station et point d'accès.

- **Station**

Le mode Station (STA) est utilisé pour connecter le module ESP à un réseau Wi-Fi établi par un point d'accès.

Figure 3.31 : ESP8266 en mode station [12].

La classe Station dispose de plusieurs fonctionnalités pour faciliter la gestion d'une connexion Wi-Fi. En cas de perte de connexion, l'ESP8266 se reconnectera automatiquement au dernier point d'accès utilisé, une fois qu'il sera à nouveau disponible. La même chose se produit au redémarrage du module. Cela est possible car ESP enregistre les informations d'identification du dernier point d'accès utilisé dans la mémoire flash (non volatile). En utilisant les données enregistrées, l'ESP se reconnectera également si l'esquisse a été modifiée mais que le code ne modifie pas le mode Wi-Fi ou les informations d'identification [12].

- **Point d'accès logiciel**

Un point d'accès (AP) est un appareil qui donne accès à un réseau Wi-Fi à d'autres appareils (stations) et les connecte à un réseau filaire. L'ESP8266 peut fournir des fonctionnalités similaires, sauf qu'il n'a pas d'interface avec un réseau câblé. Un tel mode de fonctionnement est appelé point d'accès souple (soft-AP). Le nombre maximum de stations pouvant être connectées simultanément au soft-AP peut être défini de 0 à 8 , mais par défaut à 4 [12].

Figure 3.32 : ESP8266 en mode point d'accès [12].

Le mode soft-AP est souvent utilisé et constitue une étape intermédiaire avant de connecter ESP à un Wi-Fi en mode station. C'est lorsque le SSID et le mot de passe de ce réseau ne sont pas connus à l'avance. L'ESP démarre d'abord en mode soft-AP, nous pouvons donc nous y connecter à l'aide d'un ordinateur portable ou d'un téléphone portable. Ensuite, nous sommes en mesure de fournir des informations d'identification au réseau cible. Ensuite, l'ESP passe en mode station et peut se connecter au Wi-Fi cible.

Analyse :

Pour connecter un téléphone mobile à un point d'accès, vous ouvrez généralement l'application des paramètres Wi-Fi, répertoriez les réseaux disponibles et choisissez le point d'accès dont vous avez besoin. Entrez ensuite un mot de passe (ou pas) et vous êtes dedans. Vous pouvez faire de même avec l'ESP. La fonctionnalité de recherche et de liste des réseaux disponibles à portée est implémentée par la classe Scan.

Client

La classe Client crée des clients qui peuvent accéder aux services fournis par les serveurs afin d'envoyer, de recevoir et de traiter des données.

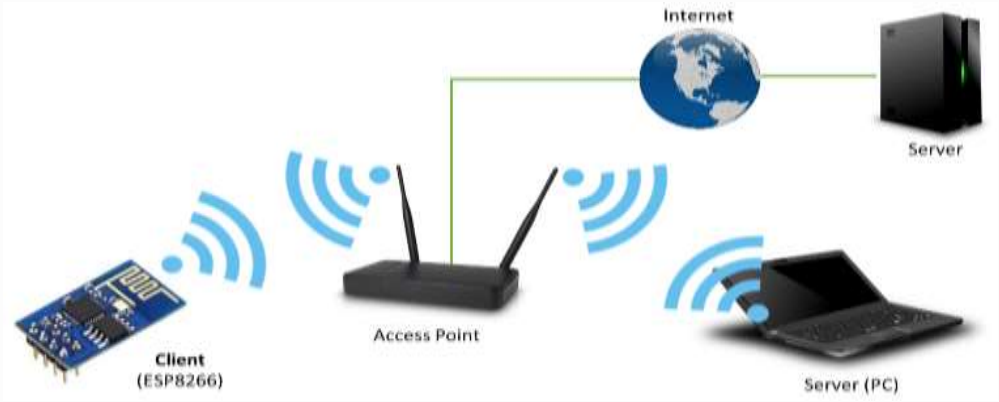


Figure 3.33 : ESP8266 en mode client [12].

Serveur

La classe Server crée des serveurs qui fournissent des fonctionnalités à d'autres programmes ou périphériques, appelés clients .

Figure 3.34 : ESP8266 en mode serveur [12].

Les clients se connectent au serveur pour envoyer et recevoir des données et accéder aux fonctionnalités fournies.

3.3 Montage électronique final de la réalisation

3.3.1 Schématisation fritzing finale

- La schématisation de montage final de notre réalisation est montrée dans les deux figures suivantes :

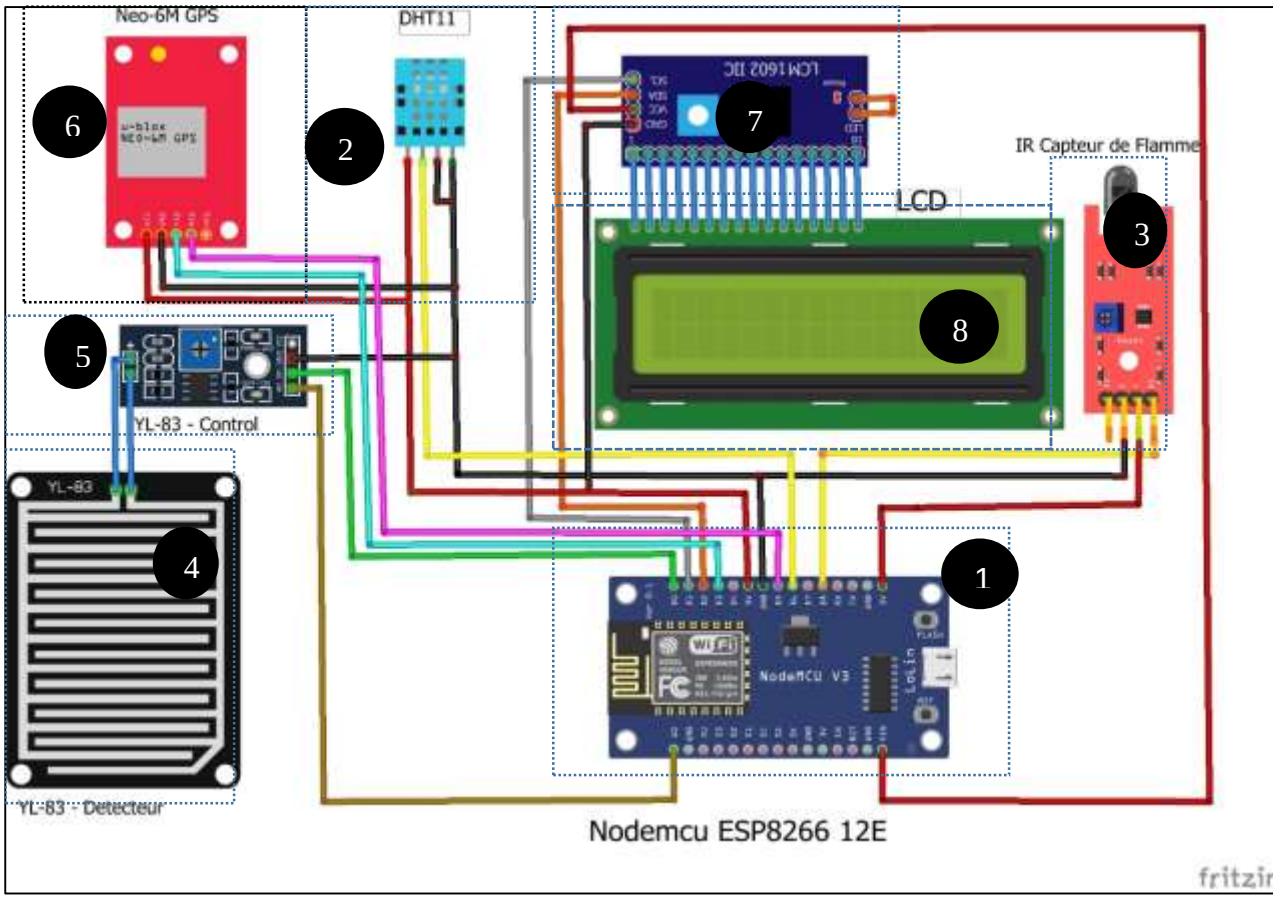


Figure 3.35 : schéma fritzing final de la réalisation.

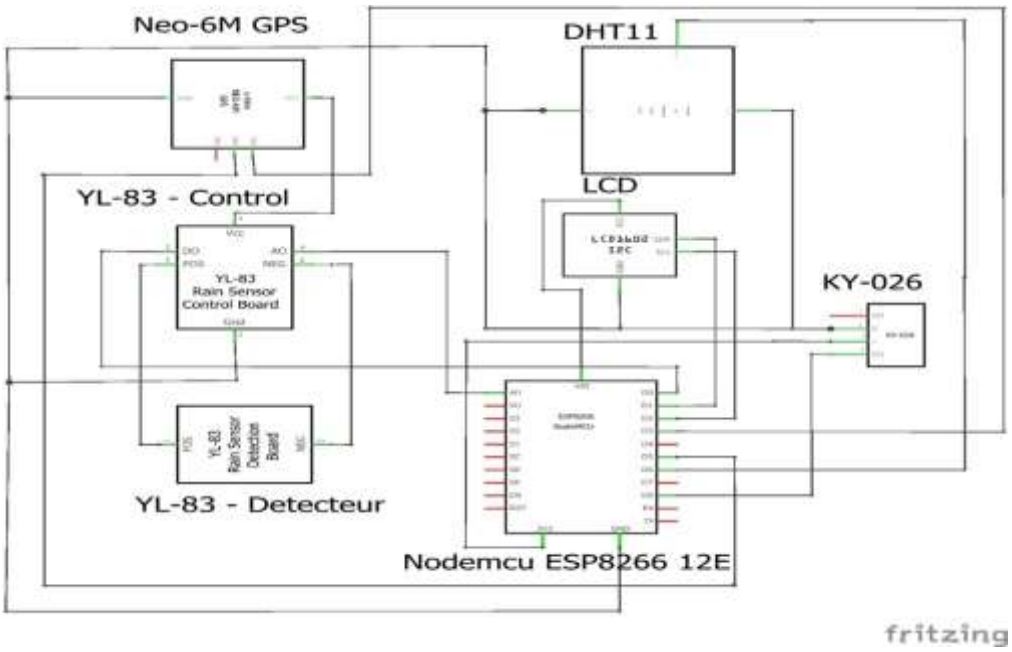


Figure 3.36 : Vue Schématique du système électronique.

- Les différentes parties du système :

1. La carte Nodemcu Esp8266 : son rôle et de recevoir le données transmets par les capteurs, traiter et lui envoyer pour les afficher sur l'afficheur LCD, et/ou par wifi pour lui afficher sur l'adresse IP occupée par le système.
2. Capteur de température dh11, il envoie les données mesurés qui sont représentée par la température et l'humidité sous forme d'un signal numérique vers l'esp8266.
3. Capteur de flamme : son fonctionnement est basé sur les ondes infra rouges, s'il reçoit un signe de flamme, il envoie un signal instantanément qui s'affichera Danger de flamme sur notre interface.
4. Dashboard rain : la surface de contacts entre le capteur de pluie et les goulettes d'eau.
5. L'unité de control de capteur de pluie, l'ensemble 4 et 5 représentent le capteur de pluie et de la neige, lorsqu'il y aura des gouttelettes d'eau sur la surface de contact, on aura un court-circuit, dont l'unité de control va transformer ce cas sous forme de deux signaux numérique et analogique vers l'ESP8266.

6. GPS NEO-6M : Il détecte la localisation actuelle du system, et d'autre paramètres telle que l'altitude et la vitesse en temps réelle, ces donnée seront envoyé sur deux liaisons RX et TX sous forme d'un signal numérique.
7. LCM 1602 IIC : une extension branchée avec l'afficheur LCD qui va simplifier le montage entre le LCD et Le ESP8266.
8. Afficheur LCD : il va nous fournir une visualisation en temps réels des données météorologiques sans besoin d'un appareil connectée, et aussi de voir l'adresse IP réservées par le système.

3.3.2 Tables de branchement

- Branchement du capteur de température et d'humidité DHT11 :

Capteur DHT11	Nodemcu Esp8266
Vcc ou (+)	V3,3
Gnd1 ou (-)	Gnd
Gnd2 ou (-)	Gnd
S ou Data	D6

- Branchement du capteur de pluie :

Capteur de Pluie	Nodemcu Esp8266
VCC	V3,3
GND	Gnd
A0	A0
D0	D0

- Branchement du capteur de flamme:

Capteur de flamme	Nodemcu Esp8266
VCC	V3,3
GND	GND
D0	D8

- Branchement du module GPS:

GPS NEO-6M	Nodemcu Esp8266
VCC	V3,3
GND	GND
TX	D3
RX	D6

- Branchement de l'afficheur LCD 2*16:

Afficheur LCD 16x2	Nodemcu Esp8266
VDD	Vin
GND	GND
SDA	D2
SCL	D1

3.4 Le système final

Après avoir mettre en place la schématisation finale de notre système électronique et la découverte des différentes modes de fonctionnement del'ESP8266 précédemment, nous avons pensé à la possibilité de voir ces données collectées par le système électronique sur un site local a temps réel et avec un affichage meilleur.

Nous avons donc schématisé ceci d'une façon synoptique comme la montre la figure suivante :

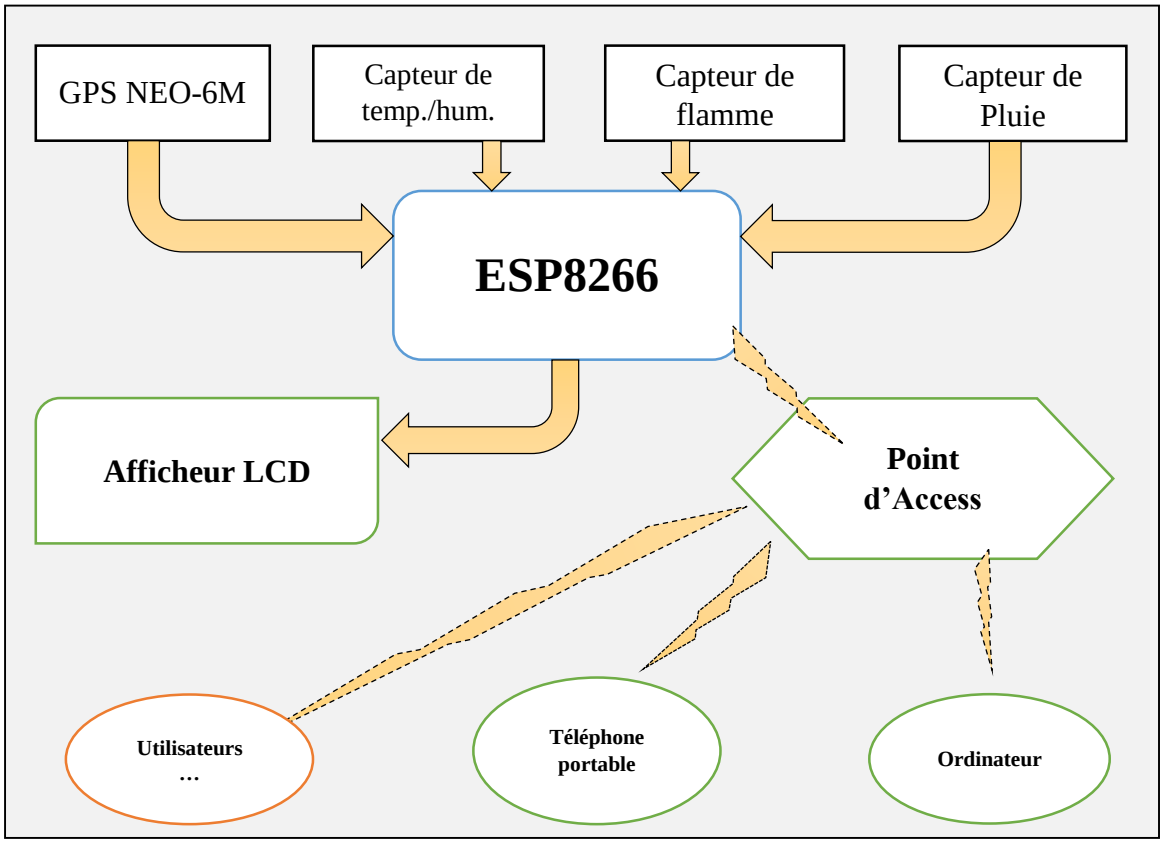


Figure 3.37 : schéma synoptique de système final.

Le système final que nous avons étudié et réalisé sera donc une Station météo de mesure de température et d'humidité, de détection de flamme et de pluie en offrant des données de

localisation en longitude et latitude, ce projet est également basé sur le serveur Web, permettant donc la visualisation de toutes les données citées précédemment soit sur un écran d'affichage LCD 2*16 ou un site local conçu pour ce but.

3.5 Test final de système

Nous avons connecté tous les capteurs selon le schéma de circuit expliqué (figure35). Ensuite, nous avons connecté notre module WiFi Nodemcu ESP8266 à l'ordinateur portable pour télécharger le code. Une fois le code téléchargé, nous avons ouvert le moniteur série, pendant les premières secondes, nous n'avons vu aucun texte sur le moniteur série, Si la même chose vous arrive, il vous suffit d'appuyer sur le bouton de réinitialisation du module WiFi Nodemcu ESP8266.

On commence à recevoir les données le premier affichage sur l'écran LCD sera une piste vers le site local c'est l'adresse IP qui va nous permettre par la suite d'accéder à notre site local.

3.5.1 Affichage de l'adresse IP sur notre afficheur

Figure 3.38 : Affichage de l'adresse IP de site local sur l'écran LCD

3.5.2 L'affichage sur le moniteur série

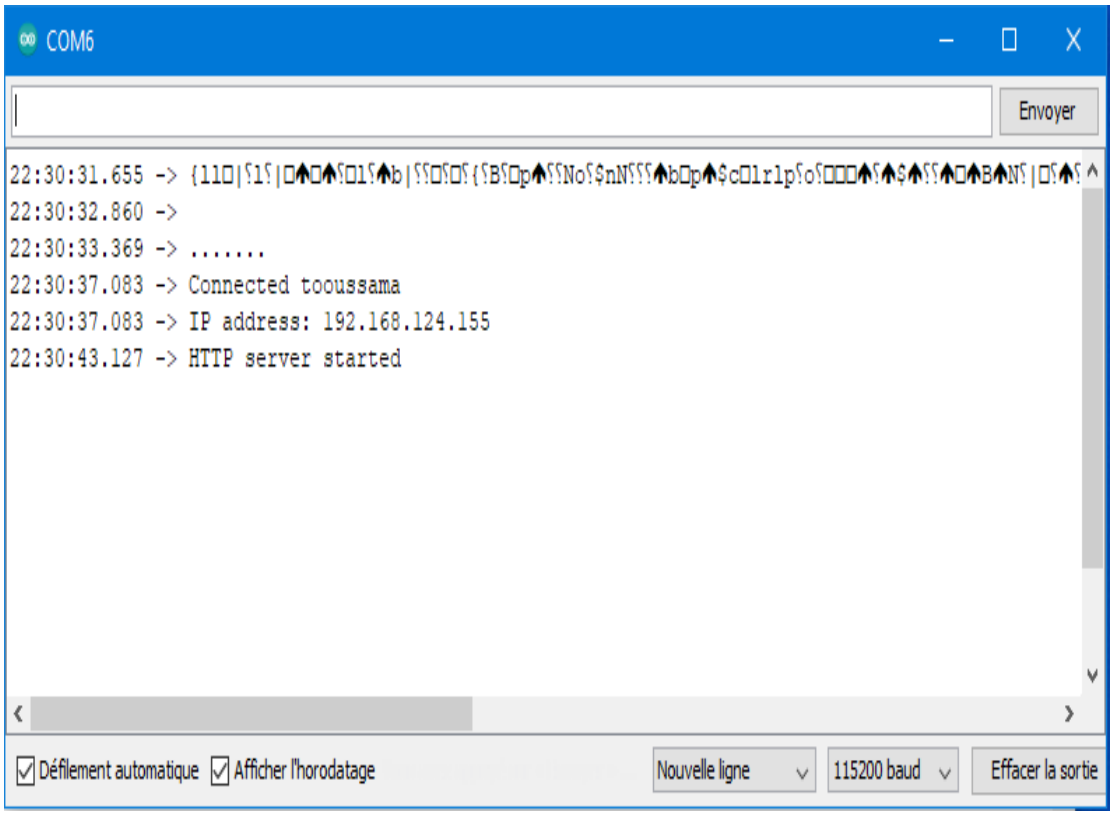


Figure 3.39 : Affichage de l'adresse IP de site local sur le moniteur série.

Ces résultats affichés sur l'écran LCD et le moniteur série assurent que le module ESP8266 est maintenant connecté au routeur WiFi, et permet également de voir l'adresse IP et le message « HTTP Server Started ».Cette adresse IP est pour accéder au site local. En la copiant sur le navigateur Web.

3.5.3 Affichage de données sur le site local

- Suivant la piste offerte par le moniteur série et l'écran LCD 2*16 et en copiant l'adresse IP 192.168.124.155 sur le navigateur le résultat est comme ceci :

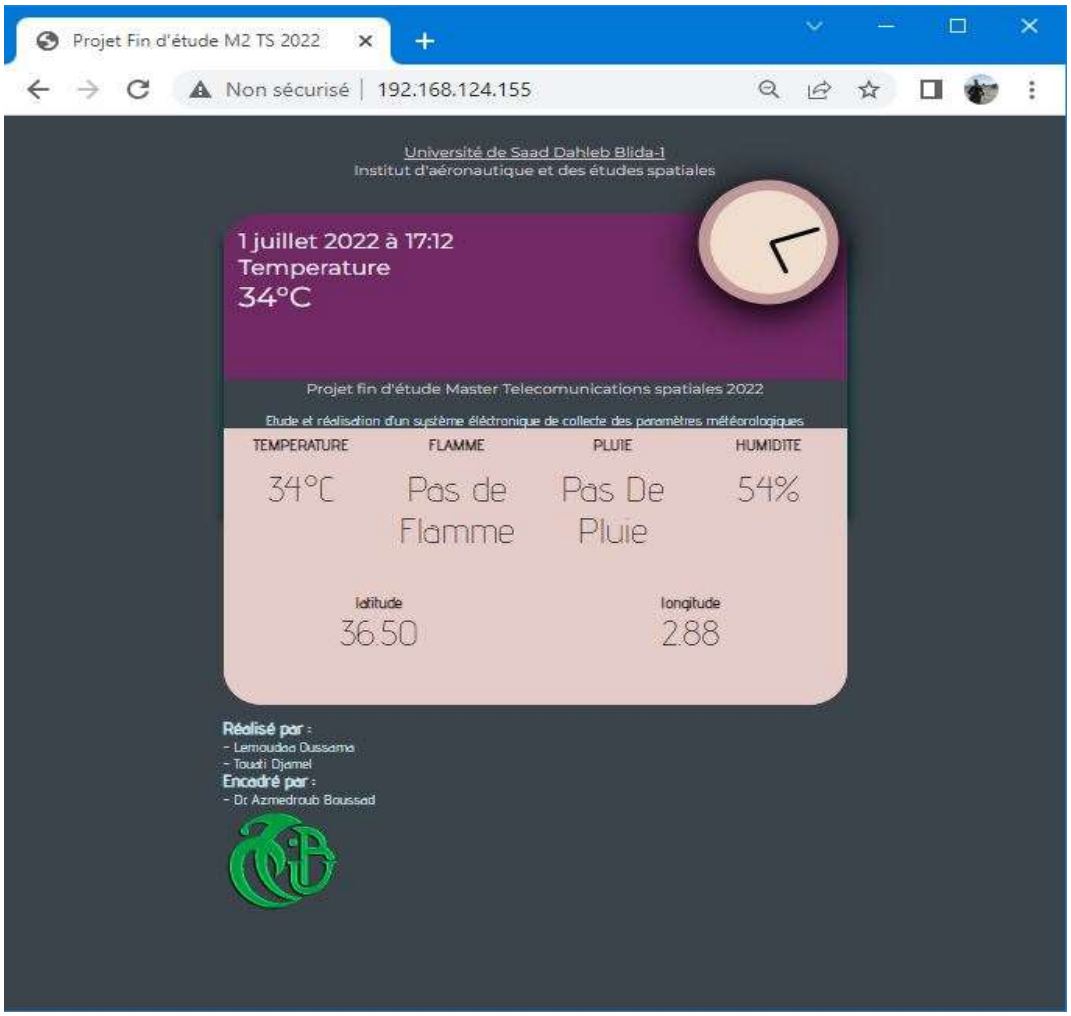


Figure 3.40 : Affichage des données sur le site local.

Notre système électronique de collecte des données météorologique basée sur le serveur Web ESP8266 affiche toutes les valeurs des capteurs souhaitées. Maintenant, allons-y et vérifions tous les capteurs un par un. En ce moment, vous pouvez voir que la température est de 34 degrés Celsius, une humidité de 54% et que l'état de la pluie est "Pas de pluie», état de flamme (pas de flamme).

3.5.4 Test capteur de pluie

Commençons d'abord par le capteur de gouttes de pluie. Actuellement, sur le site, vous pouvez voir pas de pluie. Saupoudrons un peu d'eau sur le tampon de détection des gouttes de pluie qui se compose des lignes recouvertes.

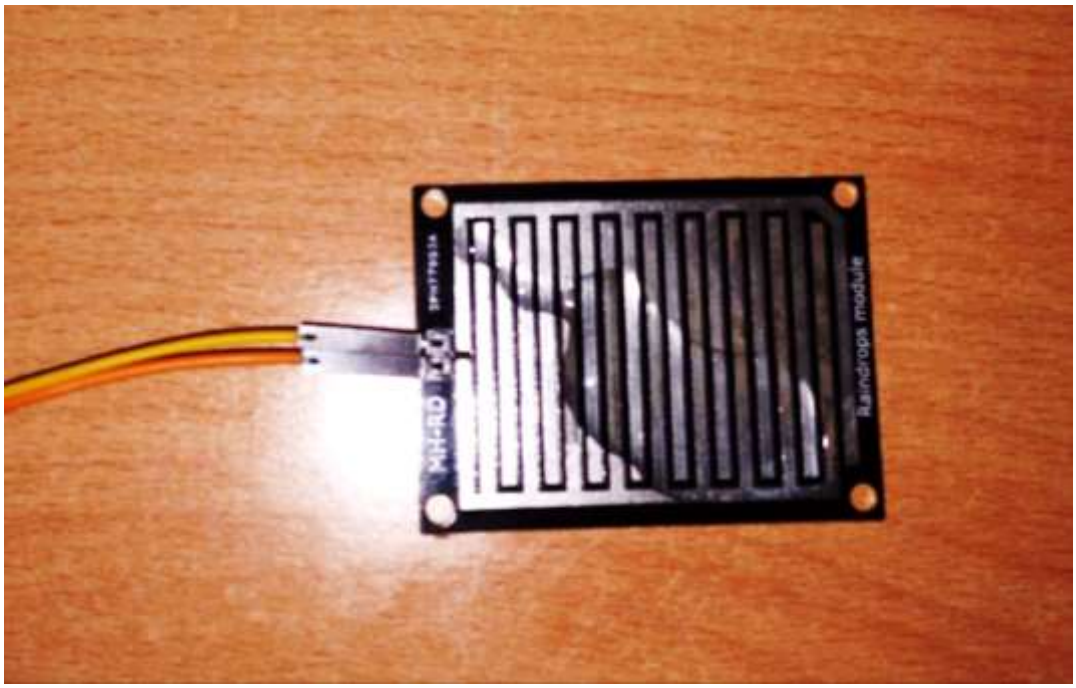


Figure 3.41 : Test de pluie.

Sur le site, vous pouvez voir que l'état de la pluie est passé de pas de pluie à il pleut. Comme illustré dans la figure suivante :

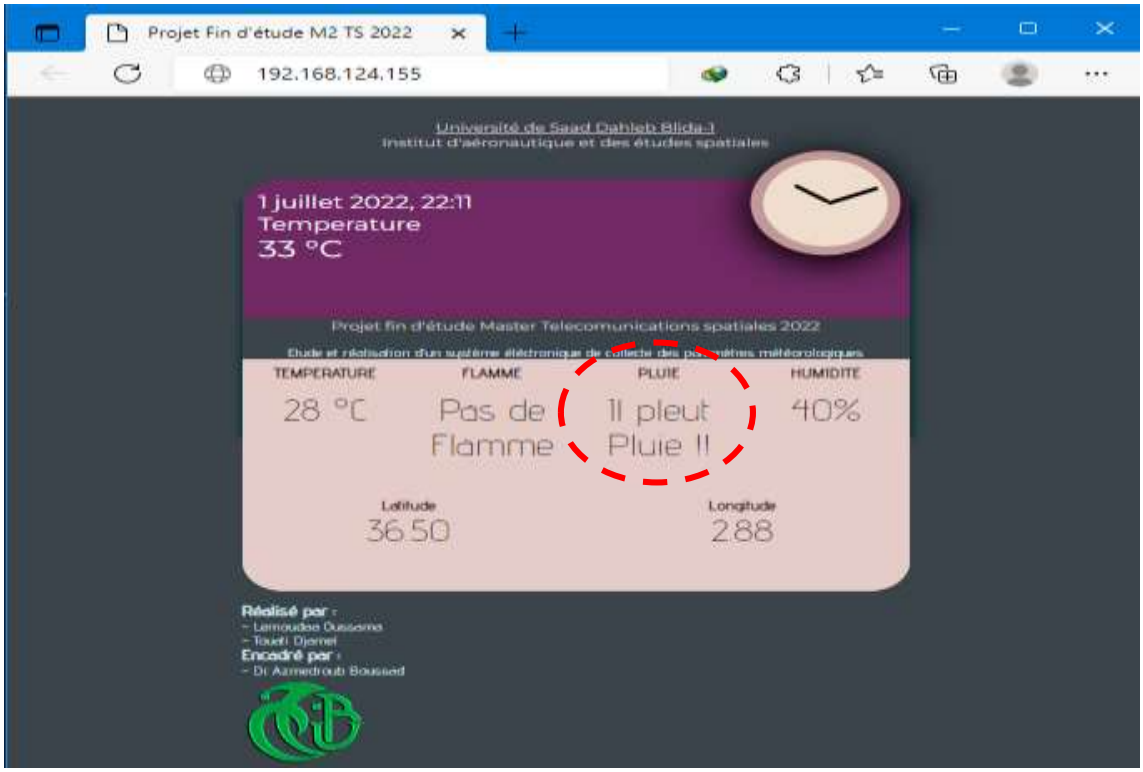


Figure 3.42 : Affichage de pluie sur le site.

3.5.5 Test de capteur d'humidité et de température DHT11

Nous avons également appliqué un peu de chaleur au capteur DHT11. Les valeurs de température et d'humidité que vous pouvez actuellement voir sur le tableau de bord de la station météo proviennent du capteur DHT11. Si vous avez utilisé le capteur DHT11, sachez que ce capteur est un peu lent. Vous pouvez voir une augmentation de la température, ce qui signifie que le capteur DHT11 fonctionne.

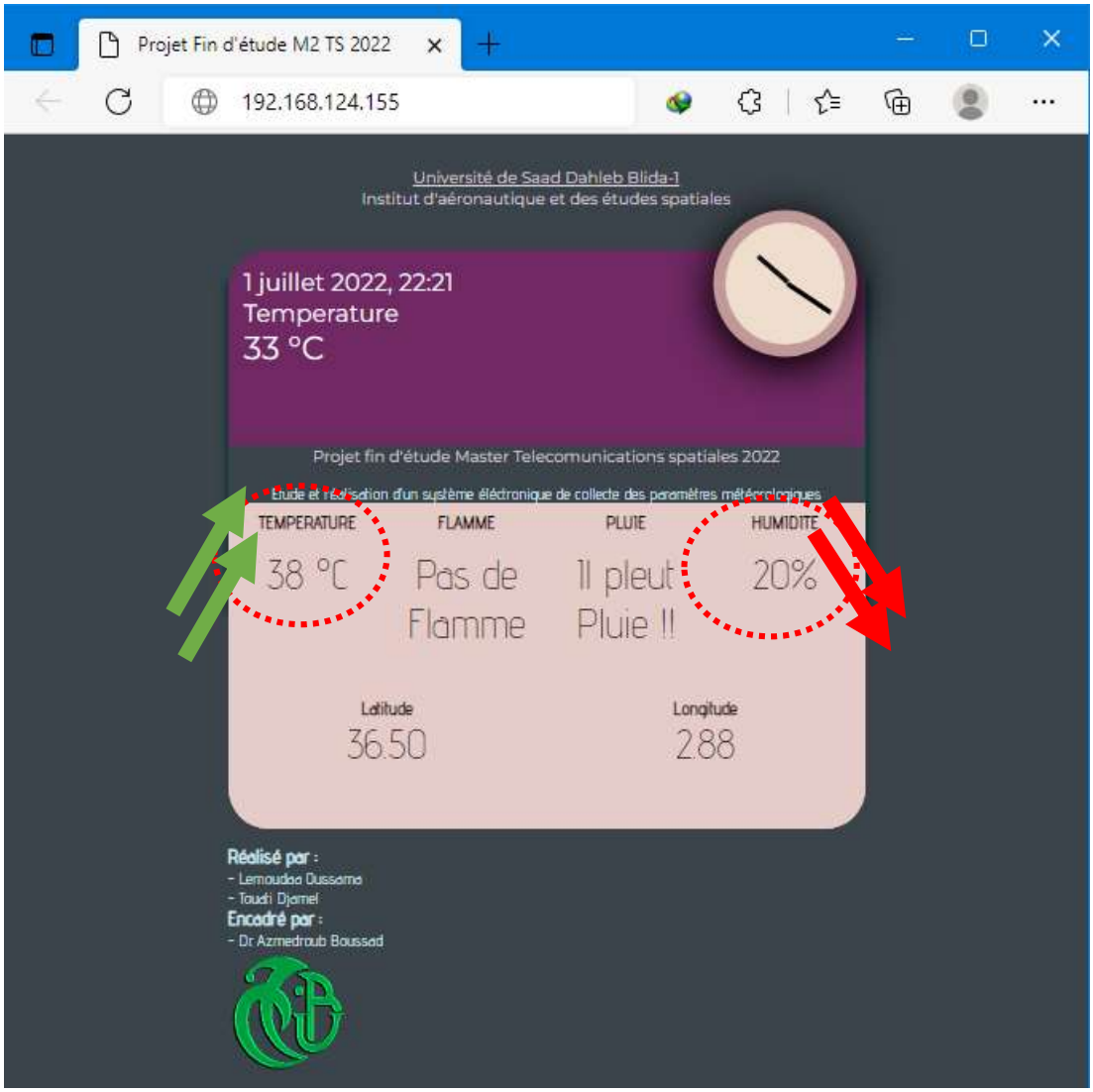


Figure 3.43 : Affichage de changement de température et d'humidité.

3.5.6 Test de capteur De flamme KY-026

Le capteur KY-026 de flamme prendre deux états, soit il détecte de flamme ou non, si il existe une flamme dans le rayon de détection, il va être lancé instantanément sur notre page, ainsi que l'afficheur LCD sur la figure suivante :

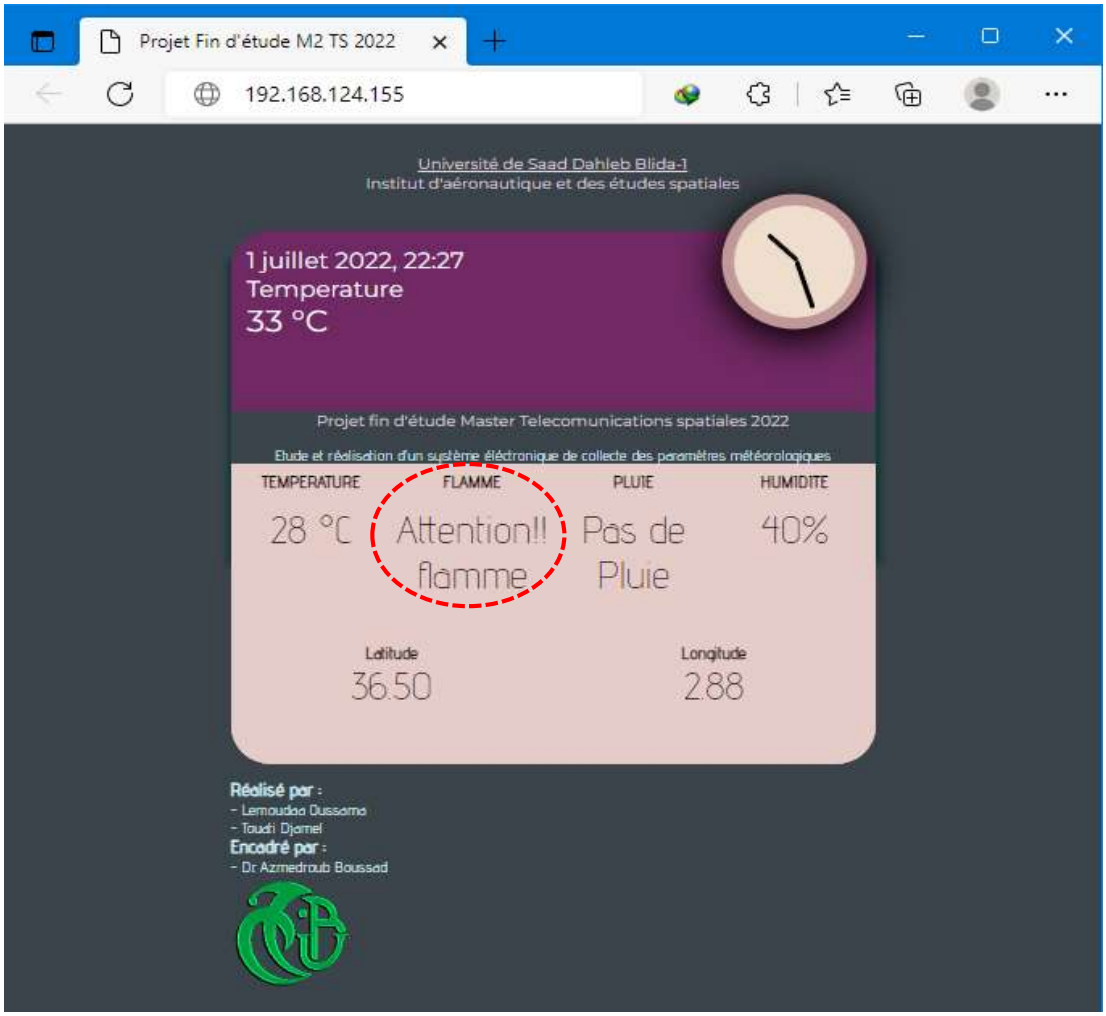


Figure 3.44 : Affichage de détection de flamme.

3.5.7 Test de l'écran LCD

Un autre dispositif caractérise notre système électronique, c'est qu'il sera toujours possible de surveiller les valeurs de température et d'humidité sur l'afficheur LCD, avec la présence ou/et l'absence de pluie et de flamme. On outre, quelle que seconde juste après le lancement de notre système, il s'affichera l'adresse IP sur afficheur du système, au lieu du moniteur série pour accéder à notre interface.

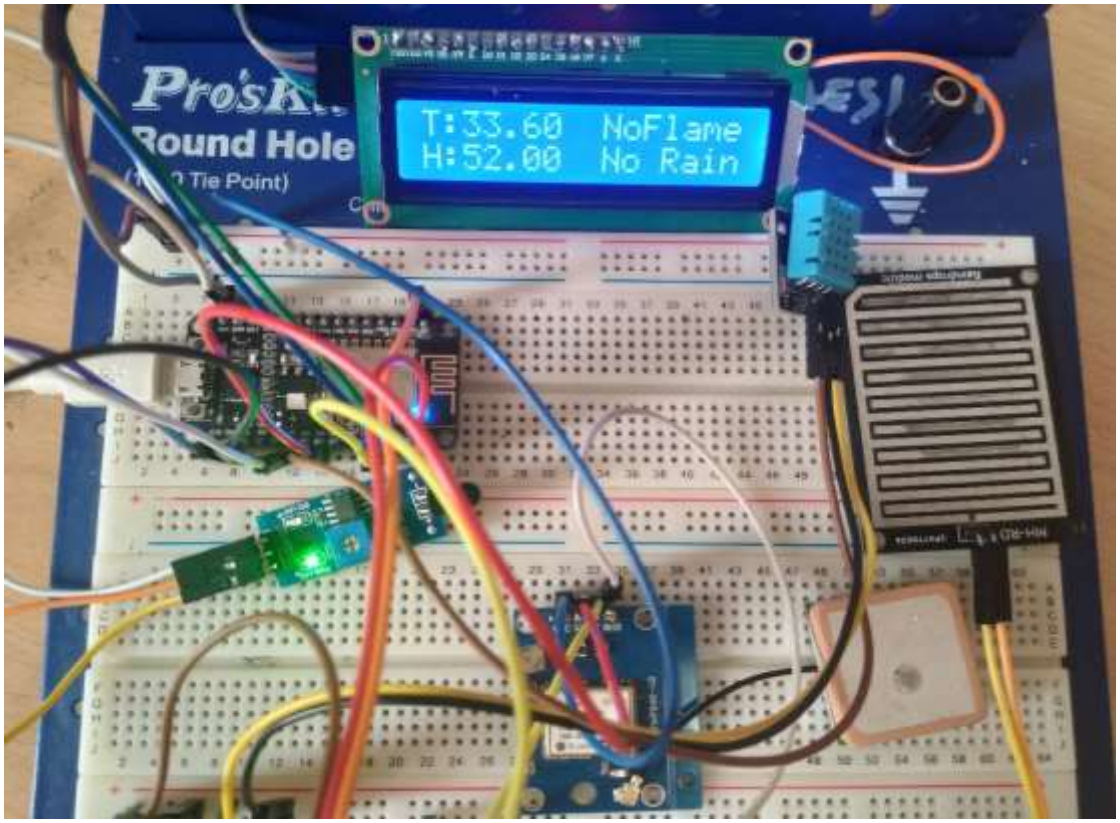


Figure 3.45 : Affichage total LCD.

3.6 Code Html et code de surveillance utilisés

Dans ce projet, le module Wifi Nodemcu ESP8266 est utilisé comme serveur Web qui stocke le code HTML (voir **Error! Reference source not found.**) et le code de surveillance (voir **Error! Reference source not found.**) de tous ces capteurs et le module GPS. Vous n'avez pas besoin d'utiliser votre ordinateur ou votre ordinateur portable comme serveur Web car vous pouvez utiliser le module Wifi ESP8266 à faible coût comme serveur Web. Les deux codes doit être dans le même dossier pour être bien exécutés.

Conclusion

Dans ce chapitre nous avons mis en avant les outils utilisés pour l'aboutissement de notre projet, après avoir énuméré les outils nous avons pu remarquer que notre projet à demander des compétences et connaissances dans plusieurs domaines tels que l'électronique et l'informatique.

Ce qui nous a permis d'améliorer nos connaissances dans ces domaines et donc nous a permis aussi de fournir une explication détaillée de la méthode de conception et de réalisation de notre projet (étude et réalisation d'un système électronique de collecte de paramètres météorologiques).

Pour finir nous avons présenté en détails la conception et la réalisation pratique de notre projet où nous avons détaillé la manière dans les différents composants sont branchés entre eux (montage ESP8266 avec les capteurs et les modules), en plus d'aborder la partie de se bénéficier de la fonctionnalité wifi, pour pouvoir visualiser les données varient en temps réel sur l'interface de site local crée.

Dans notre projet, parmi les obstacles que nous avons rencontré, la difficultés de faire un programme finale qui rassemble tous les programmes des composants seul pour travailler sur les différents capteurs en même temps et les envoyer via Wifi vers l'interface crée, c'est-à-dire relier le code de surveillance avec le code html mais nous avons dépassé cela après de nombreuses tentatives et de plus grands efforts, et à la fin nous avons obtenu le bon fonctionnement du système et les résultats auxquels nous aspirons.

Conclusion générale

Dans ce travail nous avons cherché à concevoir un système électronique de mesure de grandeurs météorologique. Il est composé d'un ensemble de capteurs qui sont : un capteur de température et d'humidité de l'air, un capteur de flamme, un capteur de pluie utilisés pour l'acquisition des données, et d'un module GPS pour déterminer la position exacte, et d'une carte à microcontrôleur NodeMcu ESP8266 dont le rôle est le traitement des données issues des différents capteurs, et de les envoyer vers un site local via wifi pour être affichées sous forme de valeurs sur ordinateur. La conception du système de mesure météorologique est passée par deux étapes principales qui sont : la réalisation des mesures avec les différents capteurs connectés à la carte Node Mcu ESP8266 et leurs transmission à distance vers l'ordinateur distant par Wifi à travers le site local. Pour conclure, nous avons réalisé des tests à l'air libre avec le système pour obtenir des mesures sur une certaine période de temps. Le système de mesure réalisé s'avère être satisfaisant. Comme perspectives de ce travail, des extensions du système de mesure peuvent être considérées tenant compte d'autres grandeurs météorologiques comme la mesure du vent et de la pression. Un système d'alarme peut être ajouté en cas de flamme aussi, une application mobile peut être conçue aussi pour la collecte des données, Un réseau de système météorologique peut être réalisé avec les mêmes procédures que nous avons utilisées pour obtenir des mesures sur une plus grande échelle. En outre les méthodes et les techniques développées pour la réalisation de ce système peuvent être améliorées pour obtenir des mesures plus précises sur les grandeurs météorologiques à mesurer. Nous espérons que ce mémoire sera utile pour les futures promotions.

Bibliographie

[1] «comment choisir sa station météo,» [En ligne]. Disponible: https://www.manomano.fr/conseil/comment-choisir-sa-station-meteo-4263. [Accès le 12 mai 2022].

[2] «lessentiel-sur-esp8266-nodemcu,» 14 janvier 2018. [En ligne]. Disponible: http://framboiseaupotager.blogspot.com/2017/12/lessentiel-sur-esp8266-nodemcu-sequence.html. [Accès le 22 avril 2022].

[3] «DHT11,» [En ligne]. Disponible: https://www.hwlibre.com/fr/dht11/. [Accès le 7 Avril 2022].

[4] «capteur de pluie,» [En ligne]. Disponible:https://letmeknow.fr/fr/environnementaux/47-capteur-de-pluie-4894479459959.html. [Accès le 25 juin 2022].

[5] «Guide du module GPS NEO-6M Arduino,» 12 january 2022. [En ligne]. Disponible: https://www.raspberryme.com/guide-du-module-gps-neo-6m-arduino/. [Accès le 20 April 2022].

[6] «Interfacing Flame Sensor with Arduino to Build a Fire Alarm System,» 2 Aout 2018. [En ligne]. [Accès le 6 April 2022].

[7] «afficheur-lcd-comment-lexploiter,» 18 April 2017. [En ligne]. [Accès le 15 Avril 2022].

[8] «Adafruit,» [En ligne]. [Accès le 04 Avril 2022].

[9] «introduction au HTML,» [En ligne]. Disponible: https://web.maths.unsw.edu.au/~lafaye/CCM/html/htmlintro.htm. [Accès le 5 mai 2022].

[10] «cascading style sheets,» [En ligne]. Disponible: https://web.maths.unsw.edu.au/~lafaye/CCM/css/cssintro.htm. [Accès le 5 mai 2022].

[11] «Json : définition et présentation de ce format de données,» [En ligne]. Disponible: https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445308-json-definition-et-presentation-de-ce-format-de-donnees/. [Accès le 15 mai 2022].

[12] «Bibliothèque ESP8266WiFi,» [En ligne]. Disponible: https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html#esp8266wifi-libra. [Accès le 17 mai 2022].