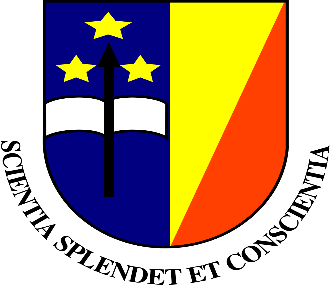
RÉPUBLIQUE DÉMOCRATIQUE DU CONGO

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET UNIVERSITAIRE

UNIVERSITÉ DE KINSHASA



FACULTÉ POLYTECHNIQUE

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET INFORMATIQUE

B.P. 127 KIN XI

**CONCEPTION ET RÉALISATION D'UN SYSTEME MULTIFONCTIONNEL ET CONNECTÉ MAXIBOX**

Réalisé par :

**KOKO BAZISHULA BENEDICT**

**MADIDI AKEM ABEL**

**MUSALU ANUNGA JONATHAN**

**NTOMPA MONGA PATRICK**

Promotion : **2ICE-EN**

Dirigé par : **Prof KUTI LUSALA Vital Angelo**

ANNÉE ACADÉMIQUE : 2024-2025

**TABLE DES MATIERES**

[**LISTE DES FIGURES** 3](#_Toc207805493)

[**SIGLES ET ABREVIATIONS** 4](#_Toc207805494)

[**I.** **INTRODUCTION** 5](#_Toc207805495)

[**II.** **CAHIER DES CHARGES** 6](#_Toc207805496)

[**2.1. Fonctionnalités attendues du système** 6](#_Toc207805497)

[**2.2. Contraintes techniques et exigences** 6](#_Toc207805498)

[**2.3. Liste des composant** 6](#_Toc207805499)

[**III.** **ETUDE ET CONCEPTION DU SYSTEME** 8](#_Toc207805500)

[**3.1. Architecture Matérielle** 8](#_Toc207805501)

[**3.1.1. Présentation des composants** 8](#_Toc207805502)

[**3.1.2. Schéma électronique du système** 20](#_Toc207805503)

[**3.1.3. Explication du fonctionnement des différentes parties du circuit** 21](#_Toc207805504)

[**3.2. Développement Logiciel** 21](#_Toc207805505)

[**3.2.1. Environnement de développement utilisé** 21](#_Toc207805506)

[**IV.** **REALISATION ET MISE EN ŒUVRE** 23](#_Toc207805507)

[**4.1. Logiciel KiCad** 23](#_Toc207805508)

[**4.2.** **Montage des composants** 24](#_Toc207805509)

[**4.3.** **Module d’Alimentation** 29](#_Toc207805510)

[**V.** **RESULTATS ET ANALYSE** 31](#_Toc207805511)

[**5.1.** **Présentation du système final** 31](#_Toc207805512)

[**5.2. Tests effectués et résultats obtenus** 34](#_Toc207805513)

[**5.3 Problèmes rencontrés et solutions proposées** 34](#_Toc207805514)

[**VI.** **CONCLUSION ET PERSPECTIVES** 36](#_Toc207805515)

[**6.1 Synthèse des résultats obtenus** 36](#_Toc207805516)

[**6.2 Améliorations possibles et perspectives d’évolution du projet** 36](#_Toc207805517)

[**VII. ANNEXES** 38](#_Toc207805518)

[**7.1** **Code source complet** 38](#_Toc207805519)

[**VIII.** **BIBLIOGRAPHIE** 39](#_Toc207805520)

[**8.1. Ouvrages et Rapports** 39](#_Toc207805521)

[**8.2. Webographie** 39](#_Toc207805522)

# **LISTE DES FIGURES**

[Figure 1: ATMEGA32BP 9](#_Toc207805600)

[Figure 2:Afficheur 7 segments 10](#_Toc207805601)

[Figure 3: LED 11](#_Toc207805602)

[Figure 4: Bouton poussoir 12](#_Toc207805603)

[Figure 5: DHT11 14](#_Toc207805604)

[Figure 6 : Buzzer 15](#_Toc207805605)

[Figure 7: Ecran LCD 1602 16](#_Toc207805606)

[Figure 8: Module ESP32 18](#_Toc207805607)

[Figure 9 : Résistances 19](#_Toc207805608)

[Figure 10 : Capteur ultrasonique 20](#_Toc207805609)

[Figure 11 : Schéma électronique du système 21](#_Toc207805610)

[Figure 12: Carte Arduino UNO 23](#_Toc207805611)

[Figure 13 : Logo du Logiciel KiCad 25](#_Toc207805612)

[Figure 14 : Circuit électrique de montage 25](#_Toc207805613)

[Figure 15 : Schéma du PCB 26](#_Toc207805614)

[Figure 16 : Rendu final du PCB 27](#_Toc207805615)

[Figure 17 : ajoute des connexions du dessus 28](#_Toc207805616)

[Figure 18 : connexion du bas de la carte 28](#_Toc207805617)

[Figure 19 : Schéma électronique des LEDs 29](#_Toc207805618)

[Figure 20 : PCB des LEDs 29](#_Toc207805619)

[Figure 21: Rendu Final du PCB des LEDs 29](#_Toc207805620)

[Figure 22 : Module d’alimentation arduino 31](#_Toc207805621)

[Figure 23 : Monitoring sur PC 33](#_Toc207805622)

[Figure 24 : Station météorologique 34](#_Toc207805623)

[Figure 25 : Circuit entier sur Breadboard 34](#_Toc207805624)

[Figure 26 : Affichage de l'heure 34](#_Toc207805625)

[Figure 27 : Affichage de la distance 34](#_Toc207805626)

[Figure 28 : Affichage du seven segment 34](#_Toc207805627)

# **SIGLES ET ABREVIATIONS**

S.C.E : Systèmes et Circuits Electroniques

PCB : Print Circuit on Board

ATMEGA : Advanced Technology Microcontroller with EEPROM and SRAM

FTDI : Future Technology Devices International

BoM : Bill of Materials

BCD : Binary Coded Decimal

LED : Light Emitting Diode

RGB : Red-Green-Blue

FabLab : Laboratoire de Fabrication

RISC : Reduced Instruction Set Computer

Ko : Kilo octet

RAM : Random Access Memory

EEPROM : Electrical Erasable Programmable Read Only Memory

UART : Universal Asynchronous Receiver Transmitter

SPI : Serial Peripheral Interface

I2C : Inter Integrated Circuit

USB : Universal Serial Bus

PWM : Pulse Width Modulation

GND : Ground

Vcc : Tension d’alimentation

IDE : Integrated Development Environment

TTL : Transistor Transistor Logic

3D : 3 Dimensions (modélisation en 3 dimensions)

SMD : Surface Mounted Device

MISO : Master In Slave Out

MOSI : Master Out Slave In

SCK : Serial ClocK

PDF : Portable Document Format

NOMAD : Non-Obvious Milling Application DevicE

# **INTRODUCTION**

Dans le cadre du cours de **Systèmes et Circuits Électroniques**, dispensé en promotion de **2ICE-EN** au Département de Génie Électrique et Informatique de la Faculté Polytechnique de l'Université de Kinshasa, il nous a été confié le projet de fin d'année académique **2024-2025**. Ce projet, intitulé **« Conception et Réalisation d'un Système Multifonctionnel »**, a pour objectif principal de nous familiariser avec les principes fondamentaux de la conception d'un système électronique autour d'un microcontrôleur.

Le système développé est une application multifonctionnelle s'articulant autour d'un microcontrôleur **ATMEGA328p**. Il intègre une horloge calandrier numérique reglable affichant l'heure et la date sur un écran LCD , un minuteur sur quatre afficheurs sept segments, une mini station météo capable de mesurer la température et l’humidité, et un système d'alarme et de notification par détection de présence mais aussi une interface graphique reprenant toutes les mesurandes sur PC et optionnellement sur smartphone. De plus, il est conçu pour interagir avec l'utilisateur via des boutons poussoirs et des voyants lumineux (LEDs). Cela étant, L'intégration d'un module **ESP32** permet une connectivité Wi-Fi, notamment pour l'affichage des données sur une interface informatique.

Ce rapport détaille les différentes étapes de notre démarche, de l'élaboration du cahier des charges à la réalisation physique du système. Nous présenterons d'abord les fonctionnalités attendues et les contraintes techniques, suivies d'une description détaillée des composants et de l'architecture matérielle. Les phases de conception et de programmation du microcontrôleur seront ensuite abordées, avant de conclure par la présentation des résultats obtenus et les perspectives d'évolution du projet.

Nous tenons à exprimer notre profonde gratitude à notre professeur, le Prof. Dr. Ir. **KUTI LUSALA Vital Angelo**, pour son encadrement précieux et ses conseils avisés qui ont guidé notre travail. Nous remercions également l'assistant **MUDIAMPIMPA Tshyster** pour sa disponibilité et son soutien technique constants tout au long de ce projet

# **CAHIER DES CHARGES**

## **2.1. Fonctionnalités attendues du système**

Le système électronique à base de microcontrôleur ATMEGA328p est un système multifonctionnel intégrant les fonctionnalités suivantes :

1. **Horloge et calendrier Numérique :** Affichage de l'heure et la date en temps réel sur un écran LCD avec la possibilité de la régler.
2. Un minuteur utilisant un afficheur 7 segments à quatre caractères pouvant compter de 0 à 9999.
3. **Station Météo Simple :** Mesure et affichage de la température et de l'humidité ambiante à l'aide d'un capteur DHT11. Les valeurs sont affichées sur l’écran LCD et peuvent au besoin être afficher su un téléphone ou un ordinateur via connexion Wi-Fi avec le module ESP32. Le nom du réseau étant MaxiBox et son mot de passe 12345678
4. **Système de Notification :**
   * Détection de présence en mentionnant la valeur de la distance sur l’interface de l’ordinateur.
   * Déclenchement d'un buzzer sonore en cas d'alerte.
5. Visualisation de l'état du système via 3 LEDs de différentes indiquant notifiant notamment sur la détection de de l’obstacle, le lancement du système et la connexion entre le SP32 et l’ATMega via le Bus I 2C**Interface Utilisateur :** Contrôle des différentes fonctions à l'aide de 4 boutons poussoirs et 2 boutons tactiles.

## **2.2. Contraintes techniques et exigences**

* **Alimentation :** Le système doit fonctionner sur une tension de 5V CC.
* **Connectivité :** Bien que centré sur l'ATMEGA328p, le système intègre un module ESP32. Son rôle est de récupérer les données mesurées à l’aide de diffèrent capteurs pour les envoyer à des plateformes ou elles peuvent être lue. Les deux microcontrôleurs sont ainsi liées par un bus série I2c permettant à l’Atmega (maitre) de communiquer avec le sp32(esclave 1) et l’écran LCD(esclave2) .
* **Programmation :** Le firmware de l'ATMEGA328p sera développé en C/C++ et téléversé via la conversion Série-USB d'un Arduino Uno, il en sera de même pour le SP32.Mais il sied de préciser que ce code inclue également du html, Css et du Js pour l’implémentation du serveur.
* **Robustesse :** Le système doit inclure des protections de base (résistances de limitation de courant pour les LEDs et les segments).
* **Encombrement et accessibilité : Avec tous les composants électroniques et leurs connexions incluses dans un système électronique il faut éviter l’encombrement et permettre une accessibilité facile pour des raisons de réparation ou encore d’extensibilité.**

## **2.3. Liste des composant**

* **Composants fournis :**
  + 1x Microcontrôleur ATMEGA328p
  + 4x Afficheurs 7 segments

A defaut d’avoir acces à un 6x Afficheurs 7 segments, nous avons fait usage d’un 4x Afficheurs 7 segments.

* + 3x LEDs (Rouge, Jaune, Verte)
  + 4x Boutons poussoirs
* **Composants ajoutés :**
  + 1x Capteur DHT11
  + 1x Capteur ultrasonore HC-SR04
  + 1x Buzzer actif
  + 1x Module d’Alimentation Arduino
  + 1x Écran LCD (16x2 ou 20x4, avec contrôleur HD44780)
  + 1x Module ESP32-WROOM-32 DevKit
  + Résistances
  + 2x Boutons tactiles (jumpers)

# **ETUDE ET CONCEPTION DU SYSTEME**

## **3.1. Architecture Matérielle**

### **3.1.1. Présentation des composants**

* **Microcontrôleur ATMEGA328p**

Développé par la société ATMEL, le microcontrôleur ATMEGA328P est un circuit intégré très répandu dans le domaine de l'électronique. Sa popularité s'explique par sa grande polyvalence et la simplicité avec laquelle il peut être mis en œuvre. Son utilisation est particulièrement notable dans l'écosystème Arduino, où il sert d’unité centrale de traitement (CPU) à des cartes emblématiques comme l'Arduino UNO, en faisant un standard de fait pour le prototypage et les projets grand public.

L'ATMEGA328P est un microcontrôleur 8 bits appartenant à la famille AVR et s'appuyant sur une architecture RISC performante. Capable de fonctionner avec une fréquence d'horloge atteignant 20 MHz, il intègre différentes mémoires : 32 Ko de Flash pour le stockage du firmware, 2 Ko de SRAM pour les variables temporaires, et 1 Ko d'EEPROM pour conserver les données de manière persistante.

Ses broches sont directement mappées sur les connecteurs des cartes Arduino populaires comme l'UNO. Ainsi, les broches du microcontrôleur correspondent aux entrées/sorties numériques et analogiques de la carte (ex. : PD3 → broche 3 PWM, PC0 → entrée analogique A0). Cette correspondance est essentielle pour la programmation within l'environnement de développement Arduino (IDE).

Ce composant embarque également plusieurs bus de communication série, incluant UART, SPI et I2C (TWI), ce qui étend ses domaines d'application. Enfin, sa faible consommation énergétique le rend particulièrement adapté aux systèmes portatifs fonctionnant sur batterie.

Pour obtenir des informations détaillées et les spécifications complètes, il est recommandé de se référer à la documentation officielle (datasheet) disponible sur le site du fabricant [[1]](#footnote-1)

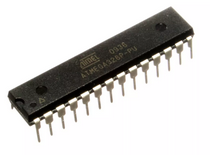


Figure 1: ATMEGA32BP

Ce microcontrôleur est présent dans certaines cartes dont la développement Arduino UNO. Voici comment établir la correspondance des pins :

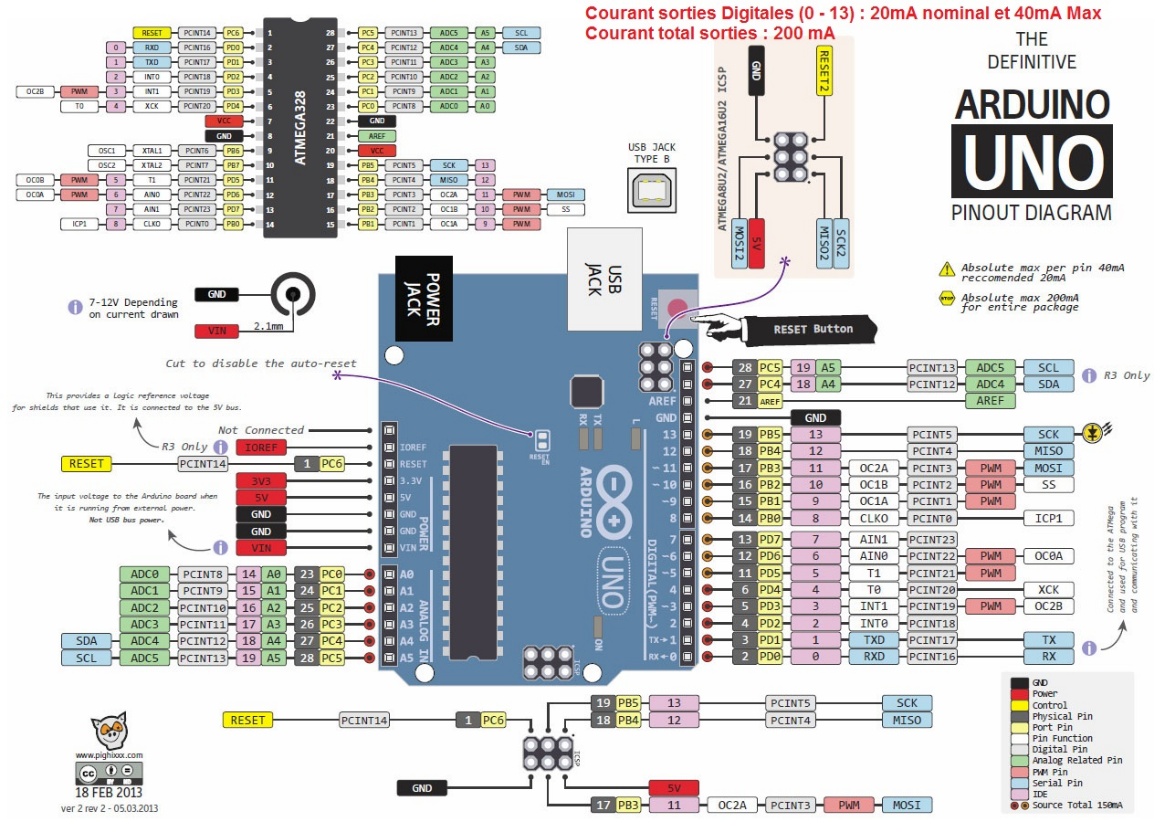


Figure Correspondance de pins entre l’Arduino UNO et l'Atmega

* **Afficheurs 7 segments**

Un afficheur 7 segments est un dispositif d'affichage numérique composé de sept diodes électroluminescentes (LEDs) arrangées de manière spécifique pour former le chiffre « 8 ». Chaque segment, identifié par une lettre (de **a** à **g**), peut être allumé ou éteint indépendamment pour composer les chiffres de 0 à 9, ainsi que certaines lettres. Un huitième segment, souvent noté **dp**, est fréquemment ajouté pour servir de point décimal.

Il existe deux configurations principales de câblage interne qui déterminent la méthode de pilotage :

* **Anode commune (Common Cathode - CC)** : Dans cette configuration, les cathodes de toutes les LEDs sont connectées ensemble à une broche commune, généralement reliée à la masse (GND). On allume un segment individuel en appliquant une tension positive (niveau logique HIGH) sur son anode via une résistance de limitation de courant.
* **Cathode commune (Common Anode - CA)** : À l'inverse, ici, ce sont les anodes de toutes les LEDs qui sont jointes à une broche commune, connectée à l'alimentation positive (Vcc). On allume un segment en ramenant sa cathode à la masse (niveau logique LOW) via une résistance.

Bien que simple dans son principe, piloter un afficheur 7 segments nécessite généralement l'utilisation d'un circuit décodeur/driver, tel qu'un **CD4511** (pour cathode commune) ou un **74LS47** (pour anode commune). Ces circuits intégrés se chargent de convertir un code binaire (sur 4 bits) représentant un chiffre (BCD) en signaux logiques pour allumer les segments appropriés, simplifiant considérablement le câblage et la programmation d'un microcontrôleur comme l'ATMEGA328P.

Sans décodeur, le pilotage dit "en mode multiplexage" est souvent employé pour contrôler plusieurs afficheurs avec un nombre minimal de broches du microcontrôleur.  
Grâce à sa simplicité, sa fiabilité et son faible coût, l'afficheur 7 segments est omniprésent dans les applications où l'affichage d'informations numériques simples est requis. On le retrouve dans une multitude de dispositifs électroniques grand public et industriels tels que : Les horloges et réveils numériques, Les compteurs (vitesse, distance, tours par minute), Les instruments de mesure (multimètres, pèse-personnes), Les panels d'affichage sur les appareils ménagers et l'électronique grand public.



Figure 3:Afficheur 7 segments

Dans notre montage nous avons fait usage d’un module constitué de 4 digits et la conversion des caractere s’est fait à par l’entre mise du code lui-meme.

La correspondance entre ledit module et l’esp32 s’est fait de la manière suivante les pins a, b, c, d, e, f, g, dp étaient respectivement associes aux gpio 13, 12, 14, 27, 26, 25, 23, 2.et pour les DIGITS (cathodes communes) les gpio 16, 17, 18, 19 étaient associés aux digits D1 à D4.

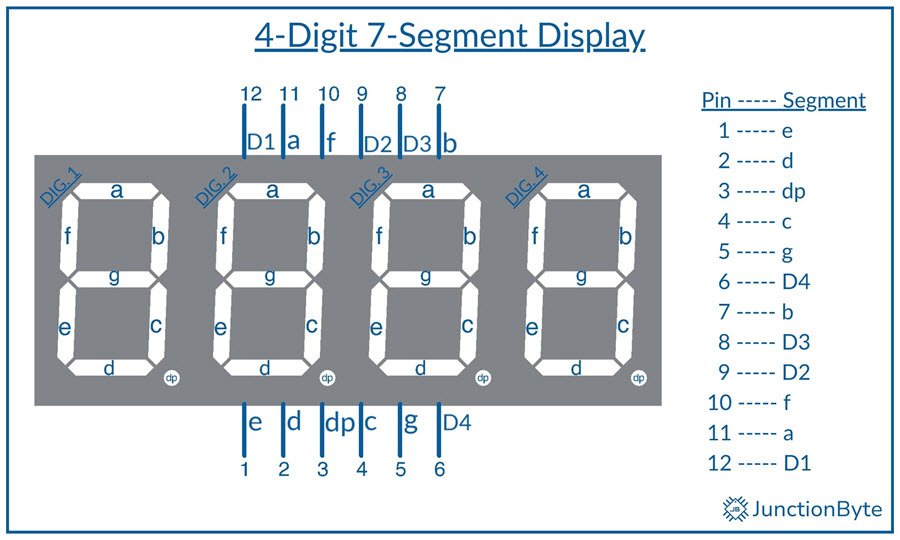


Figure Connexions physiques sur le module Seven seg

* **LEDs**

Une diode électroluminescente, ou LED (de l'anglais Light Emitting Diode), est un composant optoélectronique semi-conducteur qui convertit le courant électrique qui la traverse en lumière. Ce phénomène de luminescence, appelé électroluminescence, se produit lorsque les porteurs de charge (électrons et trous) se recombinent à l'intérieur du matériau semi-conducteur, en émettant un photon dont l'énergie détermine la longueur d'onde, et donc la couleur de la lumière émise.

Comme toute diode, la LED présente une caractéristique de polarisation **unidirectionnelle**. Elle ne laisse passer le courant et n'émet de la lumière que lorsqu'elle est polarisée en **direct** : son anode (broche longue, potentiel positif) doit être connectée au pôle positif de l'alimentation, et sa cathode (broche courte, souvent repérée par un méplat sur le boîtier) au pôle négatif (masse). Une résistance de limitation de courant est **absolument indispensable** en série avec la LED pour éviter sa destruction par surintensité.

Le pilotage correct d'une LED repose sur la maîtrise de deux paramètres électriques fondamentaux, disponibles dans sa fiche technique (datasheet) :

* **Tension directe (Vf ou Forward Voltage)** : Il s'agit de la chute de tension aux bornes de la LED lorsqu'elle est conductrice. Cette valeur, qui varie principalement selon la couleur et la chimie du semi-conducteur (par exemple, ~1.8-2.2V pour le rouge, ~3.0-3.4V pour le bleu ou le blanc), doit être prise en compte pour calculer la résistance série appropriée.
* **Courant direct (If ou Forward Current)** : C'est le courant nominal pour lequel la LED est conçue pour fonctionner de manière optimale, garantissant une luminosité adéquate sans dégradation prématurée. Il se situe typiquement autour de **20 mA** pour les modèles standards, mais peut être bien inférieur pour les LEDs haute luminosité ou "low-current".

La polyvalence, la longévité et le rendement énergétique supérieur des LEDs ont révolutionné le domaine de l'éclairage et de l'indication visuelle. Leurs applications sont vastes et incluent :

* **Signalisation et Indication** : Témoins lumineux sur tous types d'appareils électroniques (téléviseurs, chargeurs, cartes de développement).
* **Éclairage Général** : Remplaçant les ampoules à incandescence et fluorescentes grâce aux LEDs haute puissance et à leur excellent rendement lumineux (lm/W).
* **Affichage** : Formation de chiffres et de lettres sur les afficheurs 7 segments, ou d'images sur les écrans géants et les téléviseurs à LED.
* **Rétroéclairage** : Éclairage des écrans LCD d'appareils mobiles et d'ordinateurs.

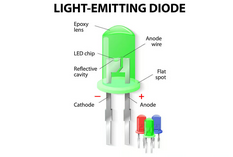


Figure 5: LED

Les leds ont été connectées sure la carte Arduino dans ses sorties numériques 3, 4 et 5.

* **Boutons poussoirs**

Un bouton-poussoir électronique, également appelé interrupteur tactile ou tact switch, est un composant actionné par pression qui établit une connexion électrique momentanée. Contrairement aux interrupteurs mécaniques traditionnels, sa conception offre une course réduite et une réponse tactile distincte, souvent accompagnée d'un « clic » audible. Son fonctionnement repose sur la déformation d'un dôme métallique qui assure le contact entre deux pastilles sur un circuit imprimé.

Ces interrupteurs sont conçus pour un montage en surface (SMD) ou traversant (THT), avec une structure compacte et scellée protégeant les contacts contre les contaminants. Ils existent principalement en deux configurations :

* **Normalement Ouvert (NO)** : Le circuit reste ouvert au repos et se ferme uniquement lors de la pression.
* **À Verrouillage** : Certains modèles intègrent une bascule (latch) pour maintenir l’état jusqu’à une seconde pression.

Leur durée de vie mécanique est typiquement de l’ordre de **100 000 à 1 000 000 de cycles**, et ils supportent des courants modestes (souvent < 50 mA) et des tensions basses (< 12 V).  
Comme les boutons poussoirs mécaniques, ils nécessitent une gestion des états logiques via des résistances de pull-up ou pull-down. Par exemple :

* Avec une résistance de pull-down, la broche du microcontrôleur lit un état **LOW** au repos et **HIGH** lors de l’appui.
* Les microcontrôleurs modernes (dont l’ATMEGA328P) permettent d’activer des pull-up internes, simplifiant le câblage.

Le **rebond mécanique** (bouncing) est également présent sur ces composants. Il doit être traité logiciellement (par délais ou algorithmes de déparasitage) ou matériellement (filtre RC ou basculeurs dédiés).

**Applications Typiques**

* Interfaces utilisateur sur appareils électroniques grand public (téléphones, télécommandes).
* Panneaux de contrôle industriels et instruments de mesure.
* Dispositifs médicaux et équipements embarqués où la fiabilité et la compacité sont critiques.

****

Figure 6: Bouton poussoir

Nous avions connecté les boutons poussoir sur la carte Arduino en faisant usage des pins 6, 7, 8 et 9.

* **Boutons Tactiles**

Notre système contient également deux boutons tactiles (par jumpers), le commande s’effectue avec un simple touché, le premier bouton sert d’allumage de notre sevent segment et l’autre il sert à basculer des fonctionnalités sur l’écran LCD.

* **Capteur DHT11[[2]](#footnote-2)**

Le DHT11 est un capteur numérique low-cost mesurant conjointement l'humidité relative et la température ambiante. Il intègre dans un boîtier compact une sonde capacitive pour l'humidité, un thermistor pour la température, et un microcontrôleur dédié effectuant le conditionnement du signal et fournissant une sortie numérique calibrée. Cette intégration simplifie considérablement son interface avec une unité de traitement principale.

Le capteur repose sur une technologie électronique propriétaire comprenant :

* **Une sonde capacitive** pour la mesure d'humidité, dont la capacité varie en fonction de la teneur en vapeur d'eau de l'air.
* **Un thermistor NTC** (Negative Temperature Coefficient) dont la résistance électrique diminue de manière prévisible à mesure que la température augmente.
* **Une ASIC** (Application-Specific Integrated Circuit) qui acquiert les signaux analogiques, les numérise, les compense et les expose via un bus de communication série numérique.

Le DHT11 communique via un **protocole propriétaire sur une seule ligne de données**, ce qui en fait un périphérique à 3 fils (VCC, GND, DATA). La procédure de lecture est initiée par le microcontrôleur maître (ATMEGA328P) :

* La ligne DATA est maintenue haute (pull-up) au repos (ici connecté au pin 12).
* Le maître envoie une impulsion de départ en abaissant la ligne pendant au moins 18 ms.
* Le capteur répond par une impulsion de 80 µs suivie de la transmission de 40 bits de données. La trame inclut les valeurs d'humidité et de température, un bit de checksum pour la validation de l'intégrité des données, et des bits décimaux fixes (résolution limitée).

Le capteur est caractérisé par une précision modeste, adaptée à des applications grand public :

* **Humidité Relative** : Mesurée de **20 % à 80 %** avec une précision de **±5 %.**
* **Température** : Mesurée de **0 °C à 50 °C** avec une précision de **±2 °C**.
* **Résolution** : Fixe à **1 %** pour l'humidité et **1 °C** pour la température.
* **Fréquence d'échantillonnage** : Limitée à **1 lecture par seconde** (1 Hz max).

**Applications Typiques :**

* Stations météorologiques simples et projets éducatifs.
* Systèmes de monitoring domestique pour le confort (serres, incubateurs).
* Dispositifs de contrôle d'environnement dans l'électronique grand public.
* Projets IoT basse consommation avec des exigences métrologiques modestes.

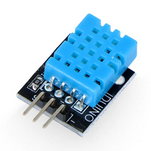
****

Figure 7: DHT11

* **Buzzer actif[[3]](#footnote-3)**

Un buzzer actif est un transducteur électroacoustique intégrant une source d'oscillation interne. Contrairement aux versions passives, il génère automatiquement un ton prédéfini et stable lorsqu'une tension continue est appliquée à ses bornes. Son fonctionnement repose sur un circuit oscillateur encapsulé qui pilote un élément piézoélectrique ou électromagnétique pour produire des ondes sonores à une fréquence fixe, typiquement comprise entre 2 et 4 kHz.

Ce composant autonome intègre dans un boîtier compact :

* Un **élément transducteur** (céramique piézoélectrique ou bobine électromagnétique) convertissant les signaux électriques en vibrations mécaniques.
* Un **circuit d'excitation intégré** (oscillateur + amplificateur) générant le signal audio sans nécessiter de pilotage externe.
* Une **résonance mécanique** optimisée pour produire un niveau sonore maximal à sa fréquence de conception.

Le buzzer actif se comporte comme une charge purement résistive, Il doit être connecté en respectant la polarité indiquée sur son boîtier (borne '+' à la tension d'alimentation, borne '-' à la masse). Un signal de commande numérique **ON/OFF** suffit : application de la tension nominale (ex: 3V, 5V ou 12V) pour émettre le son, coupure de l'alimentation pour le silence. Aucun circuit oscillateur externe, signal PWM ou générateur de fréquence n'est nécessaire.

**Caractéristiques Techniques Principales :**

* **Fréquence de fonctionnement** : Fixe et prédéterminée (ex: 2048 Hz, 2730 Hz, 4096 Hz).
* **Tension d'alimentation** : Spécifique selon les modèles (commune: 3V, 5V, 12V).
* **Consommation de courant** : Typiquement 20-30 mA, dépendant de la tension et de la conception.
* **Niveau de pression acoustique** : Généralement entre 80 et 90 dB à 10 cm de distance.
* **Plage de température** : Opérationnelle de -20 °C à +70 °C pour les modèles standards.



Figure 8 : Buzzer

**Applications Typiques**

* **Signaux d'alerte** : Avertissements sonores dans les appareils électroniques (micro-ondes, détecteurs).
* **Feedback utilisateur** : Confirmations d'entrée (claviers, télécommandes, bornes interactives).
* **Dispositifs de notification** : Alarmes de sécurité basiques, réveils, jouets.
* **Matériel informatique** : Bips du POST d'un ordinateur, alertes de capteurs.

Ce denier s’interface avec la carte Arduino via le pin 13.

* **Écran LCD1602[[4]](#footnote-4)**

Un écran LCD alphanumérique est un dispositif d'affichage à cristaux liquides conçu pour présenter du texte et des caractères simples. Les modèles standard (16x2, 20x4) intègrent un contrôleur driver HD44780 (ou un équivalent), qui sert d'interface intelligente entre le microcontrôleur et la matrice LCD. Ce contrôleur gère toute la complexité de l'affichage, permettant à l'utilisateur de se concentrer sur l'envoi des données et des commandes via un bus parallèle ou série.

L'écran est constitué de plusieurs couches et composants :

* **Matrice de pixels** : Organisée en lignes et colonnes (ex: 16 colonnes x 2 lignes). Chaque caractère est formé dans une cellule de 5x8 pixels.
* **Le contrôleur HD44780** : C'est le cerveau de l'écran. Il contient sa propre RAM (DDRAM pour stocker les caractères à afficher), une ROM (CGROM qui génère les motifs des caractères standards) et une RAM programmable (CGRAM permettant de définir des caractères personnalisés).
* **Rétroéclairage** : Généralement à LED, situé à l'arrière (backlight) ou sur les côtés (edgelight), il améliore la lisibilité dans des conditions de faible luminosité.

Le contrôleur HD44780 offre deux modes de communication principaux :

* **Mode 8 bits** : Utilise 11 broches (8 pour les données, 3 pour le contrôle).
* **Mode 4 bits** : Utilise 7 broches (4 pour les données, 3 pour le contrôle). C'est le mode le plus courant car il économise les broches du microcontrôleur. Les données sont envoyées en deux temps (nibble haut puis nibble bas).

Les **3 broches de contrôle essentielles** sont :

* **RS (Register Select)** : Détermine si l'octet envoyé est une **Commande** (RS=0, ex: effacer l'écran) ou une **Donnée** (RS=1, ex: un caractère 'A').
* **RW (Read/Write)** : Fixé la plupart du temps à 0 (Write) pour l'envoi de données/commandes.
* **E (Enable)** : Une impulsion (front descendant) sur cette broche valide et déclenche la lecture des données présentes sur le bus.

**Caractéristiques Techniques**

* **Tension d'alimentation** : Typiquement 5V pour l'électronique de contrôle. Le rétroéclairage (LED) nécessite souvent une limitation de courant via une résistance série.
* **Consommation** : Très faible pour l'affichage LCD lui-même (quelques mA), plus importante pour le rétroéclairage (varie selon le modèle, ~20-60mA).
* **Caractéristiques d'affichage** : Résolution de caractères (16x2, 20x4, etc.), jeux de caractères ASCII étendus intégrés.

**Applications Typiques :**

* **Affichage d'information** : Sur les instruments de mesure, les appareils ménagers (fours, machines à café).
* **Interface utilisateur simple** : Pour les menus basiques dans les projets électroniques (stations météo, afficheurs).
* **Débogage et développement** : Afficher des valeurs de capteurs, des états de variables ou des messages de statut en temps réel.



Figure 9: Ecran LCD 1602

* **Module ESP32 (ex: ESP32-WROOM-32 DevKit)[[5]](#footnote-5)**

Le module ESP32-WROOM-32 est une plateforme de développement tout-en-un intégrant un microcontrôleur haute performance et une connectivité sans fil dual-mode. Conçu par Espressif Systems, il combine une unité de traitement principale (CPU), une radio Wi-Fi et Bluetooth/BLE, ainsi que de multiples périphériques dans un facteur de forme compact. Il sert de cerveau intelligent pour les dispositifs IoT (Internet des Objets), permettant une connexion réseau locale et cloud ainsi que le pilotage de capteurs et actionneurs.  
Le module repose sur une architecture système sur puce (SoC) incluant :

* **Processeur principal** : Un dual-core Xtensa LX6 32 bits, cadencable jusqu'à 240 MHz, offrant une puissance de calcul significativement supérieure à celle des microcontrôleurs 8/16 bits traditionnels.
* **Mémoire** : Intègre 520 Ko de SRAM interne et 4 Mo de mémoire Flash SPI pour le stockage du firmware et des données.
* **Connectivité sans fil** : Wi-Fi 802.11 b/g/n (2,4 GHz) avec support des modes station, point d'accès (AP) et double mode simultané, Bluetooth classique (v4.2 BR/EDR) et Bluetooth Low Energy (BLE) pour les communications à courte portée et basse consommation.
* **Périphériques et Interfaces** : Le composant est remarquable par la richesse de ses interfaces :
  + **34 broches GPIO** configurables (PWM, I²C, I²S, SPI, UART, etc.).
  + **Convertisseurs Analogique-Numérique (ADC)** 12 bits - jusqu'à 18 canaux.
  + **Convertisseurs Numérique-Analogique (DAC)** 8 bits - 2 canaux.
  + **Touch sensors** capacitifs - jusqu'à 10 broches sensibles au toucher.
  + Interfaces pour capteurs Hall, SD card, Ethernet (via protocole SPI).

La carte DevKit (comme la populaire "ESP32 DevKitC") expose la plupart des broches du module WROOM-32 via des connecteurs latéraux, facilitant le prototypage.

* **Alimentation** : Polyvalente (USB 5V, batterie Li-Po via port dédié, source externe 3.3V-12V).
* **Programmation** : Principalement via le port USB intégré (convertisseur USB-série CH340/CP2102 intégré).
* **Support logiciel** : Environnements de développement multiples supportés :
  + **Arduino IDE** (via le noyau ESP32) pour une prise en main rapide.
  + **Framework ESP-IDF** (IoT Development Framework) officiel pour exploiter toute la puissance et les fonctionnalités avancées.
  + **MicroPython** et **PlatformIO** pour des workflows alternatifs.

**Caractéristiques Techniques**

* **Tension d'alimentation** : 3.0V - 3.6V (3.3V typique). La carte DevKit accepte une entrée USB 5V ou Vin.
* **Consommation** : Extrêmement variable selon le mode (veille profonde : ~5 µA, actif : ~240 mA pic).
* **Plage de température** : Opérationnelle de -40 °C à +125 °C.
* **Puissance RF** : Puissance de transmission Wi-Fi jusqu'à +20.5 dBm.

**Applications Typiques :**

* **Dispositifs IoT** : Objets connectés (domotique, capteurs environnementaux cloud, wearables).
* **Prototypage avancé** : Contrôle de robots, stations de données complexes, interfaces homme-machine.
* **Pont de communication** : Servir de gateway pour connecter des réseaux de capteurs sans fil (Bluetooth -> Wi-Fi -> Cloud).
* **Audio et voix** : Stream audio basique, reconnaissance vocale grâce à la puissance de calcul.

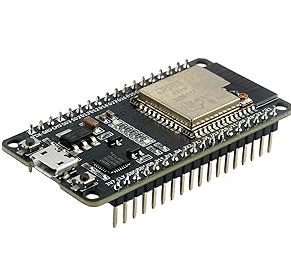


Figure 10: Module ESP32

* **Résistances**

Une résistance est un composant électronique passif fondamental dont la fonction principale est d'**opposer une résistance électrique** au passage du courant dans un circuit. Cette opposition, mesurée en Ohms (Ω), permet de **limiter le courant**, **diviser des tensions** et **ajuster les niveaux de signaux** selon la loi d'Ohm (U = R × I). C'est l'un des composants les plus utilisés en électronique.

Il existe plusieurs types de résistances, chacune avec sa propre technologie :

* **Résistances à couche de carbone** : Un noyau céramique recouvert d'une fine couche de carbone. Leur valeur est définie par la quantité de carbone déposée.
* **Résistances à couche métallique** : Similaires aux modèles à couche de carbone, mais utilisent un alliage métallique (nickel-chrome) pour une meilleure stabilité et précision.
* **Résistances bobinées** : Un fil résistif (constantan, manganine) enroulé autour un support céramique. Utilisées for des fortes puissances.
* **Résistances CMS (composant montage en surface)** : Version miniature sans broches, destinée à être soudée directement sur la surface de la carte PCB.

Pour les résistances à axiale, la valeur est indiquée par un **code de couleurs** composé de 4 ou 5 bandes :

* **4 bandes** : 2 chiffres + multiplicateur + tolérance.
* **5 bandes** : 3 chiffres + multiplicateur + tolérance (pour une meilleure précision).

Les valeurs suivent une série normalisée (E12, E24) pour couvrir les plages nécessaires avec un nombre limité de composants.

**Caractéristiques Techniques**

* **Valeur ohmique (R)** : De quelques Ohms à plusieurs Mégaohms (MΩ).
* **Tolérance** : Précision de la valeur nominale (Or ±5%, Argent ±10%, Marron ±1%).
* **Puissance maximale** : Capacité à dissiper la chaleur (¼ W, ½ W, 1 W, etc.) sans dommage.
* **Coefficient de température (TC)** : Variation de la valeur en fonction de la température (en ppm/°C).

**Applications Typiques :**

* **Limitation de courant** : Protection des LEDs et autres composants sensibles.
* **Ponts diviseurs de tension** : Pour mesurer des tensions ou créer des références.
* **Pull-up / pull-down** : Définir un état logique par défaut sur les entrées digitales.
* **Terminaison de lignes** : Adaptation d'impédance dans les signaux haute fréquence.
* **Décharge de condensateurs** : Sécurité pour éviter les décharges résiduelles.

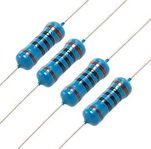


Figure 11 : Résistances

* **Capteur ultrasonique[[6]](#footnote-6)**

Un capteur à ultrasons est un dispositif de télémétrie qui détermine la distance à un objet en calculant le temps de parcours d'une onde acoustique. Il émet une brève impulsion ultrasonore (inaudible pour l'homme, typiquement 40 kHz) et mesure le temps écoulé jusqu'à la réception de son écho après réflexion sur une surface. La distance est déduite à l'aide de la formule : **Distance = (Vitesse du son × Temps de vol) / 2**.

Le module intègre deux transducteurs piézoélectriques principaux et un circuit de contrôle :

* **Émetteur ultrasonore** : Génère l'onde acoustique de 40 kHz lorsque celui-ci est excité par une impulsion électrique.
* **Récepteur ultrasonore** : Détecte l'onde réfléchie et la convertit en un signal électrique.
* **Circuit de traitement** : Contrôle le timing, amplifie le signal de retour, et filtre le bruit. Sur des modules comme le HC-SR04, il génère directement un signal numérique proportionnel au temps de vol.

Le pilotage s'effectue via une interface numérique simple :

* **Déclenchement (Trigger)** : Le microcontrôleur envoie une impulsion haute d'au moins **10 µs** sur la broche dédiée(broche 10).
* **Émission/Réception** : Le module émet automatiquement une rafale de 8 cycles à 40 kHz et attend l'écho.
* **Mesure de l'écho (Echo)** : Le module maintient sa broche (broche 11). de sortie à l'état HIGH pendant une durée exactement égale au temps de vol aller-retour de l'onde. Le microcontrôleur mesure cette durée (pulseIn() sur Arduino).

**Caractéristiques Métrologiques et Performances**

* **Plage de mesure** : Typiquement **2 cm à 400 cm** (4 m). Performances réduites sur les surfaces absorbantes ou inclinées.
* **Résolution** : Environ **0.3 cm**.
* **Angle de mesure** : Champ de détection conique d'environ **15 degrés**.
* **Tension d'alimentation** : **5 V DC**.
* **Fréquence de fonctionnement** : **40 kHz**.

**Applications Typiques :**

* **Robotique** : Permet d'éviter certains obstacles, navigation autonome.
* **Télémétrie** : Mesure de niveau (réservoirs, silos), stationnement automobile.
* **Dispositifs interactifs** : Détection de présence ou de mouvement.
* **Cartographie** : Scanner ultrasonore simple pour modélisation rudimentaire.

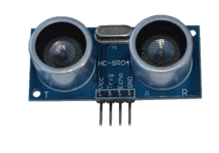


Figure 12 : Capteur ultrasonique

## **3.1.2. Schéma électronique du système**

Ce schéma comporte plusieurs connexion dont le bus I2c Constitué de 4 pins etre Sp32 et Lcd I2C 16x2 ; 3 pins reliant 3 devices.Que nous resumons au travers du tableau que voici :

|  |  |  |
| --- | --- | --- |
| Atmega | Sp32 | Lcd I2C 16x2 |
| GND | GND | GND |
|  | Vcc | Vcc |
| A5 | GPIO22 | SDA |
| A4 | GPIO21 | SCL |

Voici donc le shema global de notre circuit :

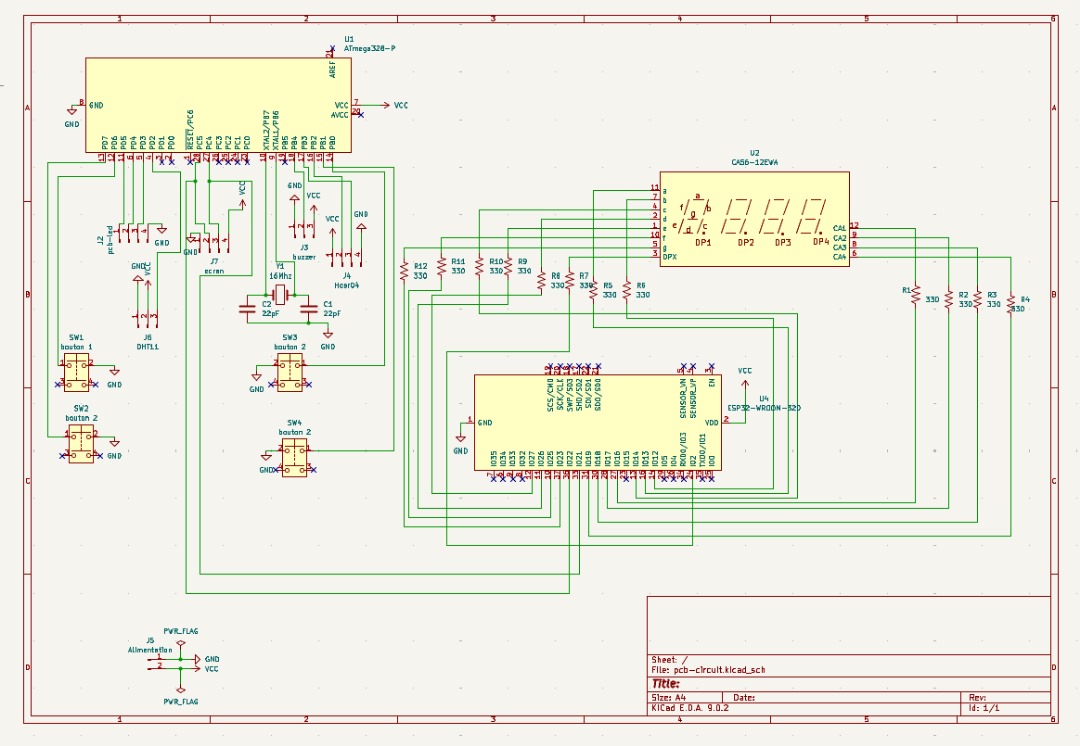


Figure 13 : Schéma électronique du système

* **Connexions principales**
* **Afficheurs 7 segments :** Utilisation de deux registres à décalage 74HC595 en cascade pour contrôler les segments (broches DATA, CLOCK, LATCH de l'ATMEGA). Les broches de contrôle des digits (anodes communes) seront pilotées par l’ESP32, celui-ci étant commandé par d'autres broches de l'ATMEGA.
* **Écran LCD :** Connecté en mode 4 bits (broches PC4, PC5, VCC et GND) pour économiser les GPIOs.
* **DHT11 :** Connecté à une broche digitale(PD2) de l'ATMEGA.
* **Buzzer :** Connecté à une broche PWM (PB5) de l'ATMEGA via un transistor pour amplifier le courant.
* **Boutons poussoirs :** Connectés à des broches digitales (PD6, PD7, PB0, PB1) en mode INPUT\_PULLUP.
* **LEDs :** Connectées à des broches digitales via des résistances de limitation de courant (220-330Ω).
* **Capteur ultrasonique :** Connectées à des broches digitales (PB2 et PB3)
* **ESP32 :** Communique avec l'ATMEGA via une liaison série UART (Broches RX/TX). Un diviseur de tension (1kΩ) est nécessaire sur la ligne TX de l'ESP32 (3.3V) vers la ligne RX de l'ATMEGA (5V) pour ne pas endommager l'ESP32.

## **3.1.3. Explication du fonctionnement des différentes parties du circuit**

Ce système électronique multifonctionnel repose sur un microcontrôleur ATMEGA328p qui agit comme le cerveau principal, coordonnant l'ensemble des opérations. Le système intègre quatre fonctions principales qui coexistent harmonieusement grâce à une programmation efficace en C/C++.

L'horloge numérique constitue la fonction centrale, affichant l'heure en temps réel sur un afficheur 4 digits à sept segments. La particularité réside dans son initialisation : le module ESP32 se connecte au Wi-Fi pour récupérer l'heure exacte via le protocole NTP avant de la transmettre à l'ATMEGA328p par communication série. L'affichage utilise la technique du multiplexage, où le microcontrôleur active successivement chaque digit à une fréquence rapide, créant ainsi une impression de continuité visuelle. Trois boutons poussoirs permettent à l'utilisateur d’allumer et éteindre le minuteur, aussi changer les fonctionnalités dans le fonctionnement du système en général.

Parallèlement, le système intègre une station météo basique utilisant un capteur DHT11 pour mesurer la température et l'humidité ambiante. Les données collectées sont transmises via l'ESP32 pour être visualisées sur une interface de l’ordinateur, plutôt que d'être simplement affichées localement.

La fonction d'alarme et de notification apporte une dimension sécuritaire au système. Un capteur à ultrasons mesure en permanence les distances aux alentours. Lorsqu'une présence est détectée en deçà d'un certain seuil, le système déclenche immédiatement une alarme sonore via un buzzer et signale l'état d'alerte à l'aide de trois LEDs de couleurs différentes. Les valeurs de distance sont également transmises à l'interface PC pour monitoring.

L'ensemble du système est conçu pour une robustesse optimale, avec des résistances de limitation de courant protégeant tous les composants sensibles, et fonctionne sous une tension stable de 5V. L'architecture intelligente qui répartit les tâches entre l'ATMEGA328p pour le contrôle local et l'ESP32 pour la connectivité Wi-Fi permet une performance fiable et polyvalente.

## **3.2. Développement Logiciel**

### **3.2.1. Environnement de développement utilisé**

Le développement logiciel pour le microcontrôleur a été intégralement réalisé en utilisant **l'environnement Arduino IDE (Integrated Development Environment)**. Ce choix a été motivé par sa simplicité d'utilisation, sa prise en charge native complète de l'écosystème matériel Arduino et sa large adoption au sein de la communauté des makers et des éducateurs.

Bien que le projet repose sur un microcontrôleur ATMEGA328P, le cadre de programmation Arduino a été retenu pour capitaliser sur les avantages significatifs de son écosystème. Cette approche a permis de bénéficier :

* D'un jeu de bibliothèques (librairies) riche et optimisé, abstrayant la complexité de la programmation bas niveau et accélérant le développement.
* De fonctions intégrées simplifiant la gestion des entrées/sorties (GPIO), la communication série, le contrôle des timers et la gestion des interruptions.

Le code applicatif a été écrit en **C/C++** en suivant le paradigme de programmation spécifique à l'environnement Arduino, basé sur les deux fonctions principales :

* setup() : Pour l'initialisation du microcontrôleur et la configuration des périphériques.
* loop() : Contenant la logique principale du programme, exécutée de manière cyclique.

Le transfert du firmware compilé vers le microcontrôleur cible a été effectué en utilisant une **carte Arduino Uno simplement ou configurée en programmateur ISP (In-System Programmer)**. Cette méthodologie économique et efficace a consisté à :

1. Configurer la carte Uno en y téléversant le sketch "Arduino as ISP".
2. Câbler les signaux de programmation (MISO, MOSI, SCK, RESET, VCC, GND) entre les broches de l'Uno et celles de l’ATMEGA328P cible.
3. Sélectionner le bon programmateur dans l'IDE Arduino ("Arduino as ISP") et utiliser la fonction **"Téléverser avec le programmateur"** pour transférer le code compilé directement dans la mémoire flash du microcontrôleur, sans nécessiter de bootloader préalablement installé.



Figure 14: Carte Arduino UNO

# **REALISATION ET MISE EN ŒUVRE**

Voici ici les étapes de la conception physique du système, depuis la phase de prototypage jusqu'à la mise en service du circuit final. Nous y présentons le système achevé, les procédures de test, ainsi que les difficultés rencontrées et les solutions mises en place.

## **4.1. Logiciel KiCad**

La conception des circuits imprimés (PCB) a été intégralement réalisée à l'aide de **KiCad**, une suite logicielle libre et open-source de conception électronique. Cet environnement complet intègre tous les outils nécessaires pour le développement de circuits imprimés professionnels, de la saisie schématique à la génération des fichiers de fabrication.

Le choix de KiCad s'est imposé pour plusieurs raisons stratégiques :

* **Accessibilité et Coût** : Étant gratuit et open-source, KiCad élimine les barrières financières associées aux logiciels propriétaires, le rendant accessible aux passionnés, aux étudiants et aux professionnels.
* **Ecosystème Complet et Intégré** : La suite offre un flux de travail unifié et cohérent avec des outils intégrés pour chaque étape (schématique, mise en page, routage, génération de fichiers), évitant les problèmes d'interopérabilité.
* **Fonctionnalités Professionnelles** : KiCad supporte des projets de complexité arbitraire, incluant le routage différentiel, la gestion de signaux haute fréquence, un inspecteur de règles de conception (DRC) puissant et une vaste bibliothèque de composants communautaire.
* **Indépendance et Pérennité** : Le modèle open-source garantit l'indépendance vis-à-vis d'un éditeur unique et assure la pérennité et l'évolution continue de l'outil grâce à une communauté active de développeurs et d'utilisateurs.

La méthodologie de conception dans KiCad suit un flux de travail structuré :

* **Saisie Schématique (Eeschema)** : Création du diagramme électrique en utilisant les symboles des composants. Les connexions sont établies et les broches sont étiquetées.
* **Gestion des Empreintes (Footprint)** : Attribution d'un modèle physique (empreinte) à chaque symbole schématique, définissant les pads de soudure et le contour du composant sur le PCB.
* **Assignation des Empreintes** : L'assistant d'assignation permet de lier facilement chaque composant du schéma à son empreinte physique dans la bibliothèque.
* **Routage du Circuit Imprimé (Pcbnew)** :
  + **Placement des Composants** : Disposition manuelle ou assistée des empreintes sur la carte en optimisant l'espace et les longueurs de pistes.
  + **Routage des Pistess** : Connexion électrique des pads selon le schéma, effectuée manuellement ou à l'aide du routeur automatique.
  + **Vérification par les Règles de Conception**: Validation du layout par rapport à un jeu de règles définies (espacements, largeurs de piste) pour garantir la fabricabilité.
* **Génération des Fichiers de Fabrication (Gerber, Drill)** : Création des fichiers standards de l'industrie (Gerber, XLL, drill, etc.) nécessaires pour soumettre la conception à un fabricant de PCB.

Pour tout projet de conception électronique, il est recommandé de consulter la **documentation officielle de KiCad** et les ressources communautaires pour tirer pleinement parti des bonnes pratiques et des fonctionnalités avancées de l'outil.



Figure 15 : Logo du Logiciel KiCad

## **4.2. Montage des composants**

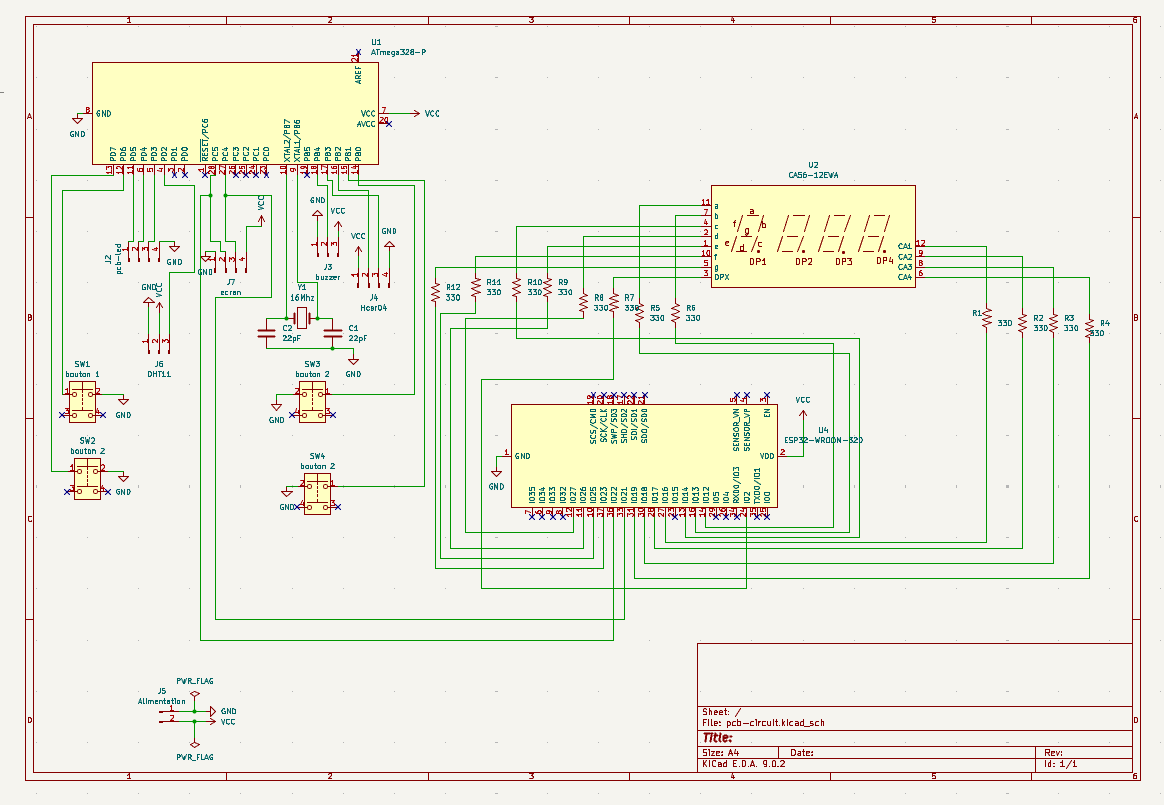


Figure 16 : Circuit électrique de montage

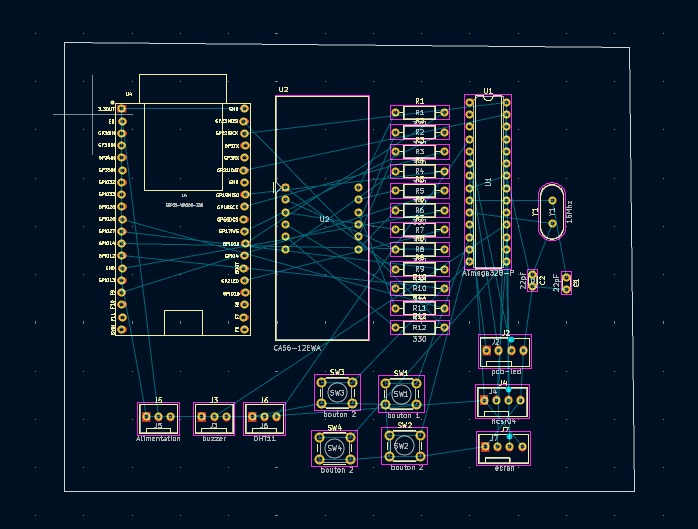


Figure 17 : Schéma du PCB

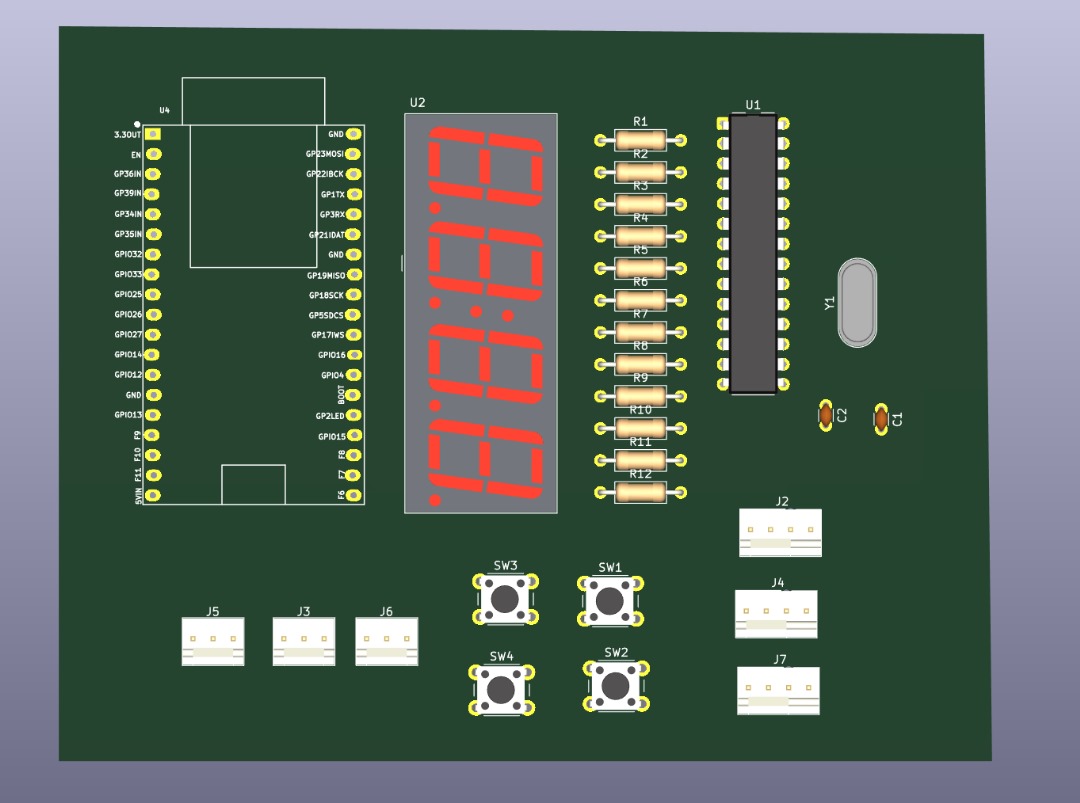


Figure 18 : Rendu final du PCB

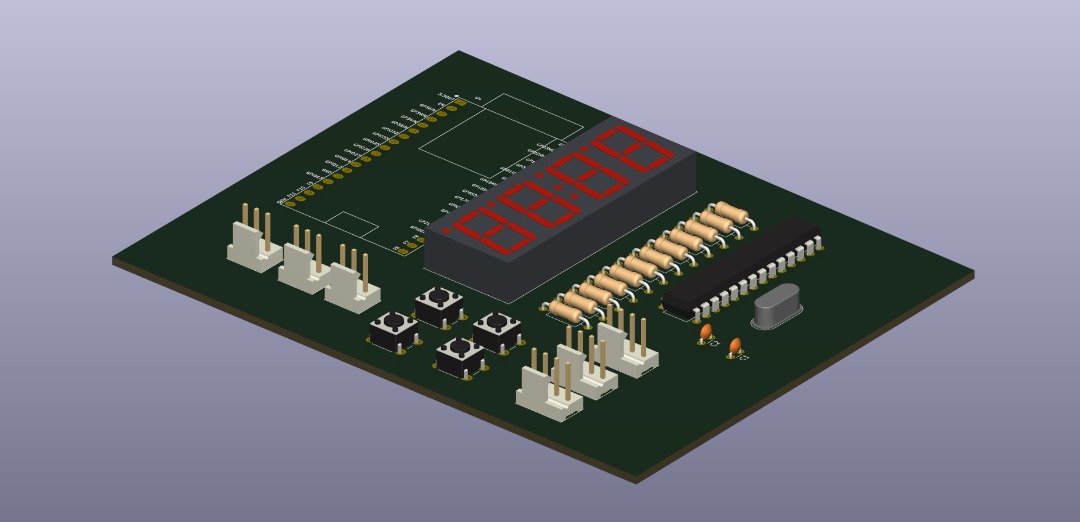


Figure 17: Vu 3D du PCB

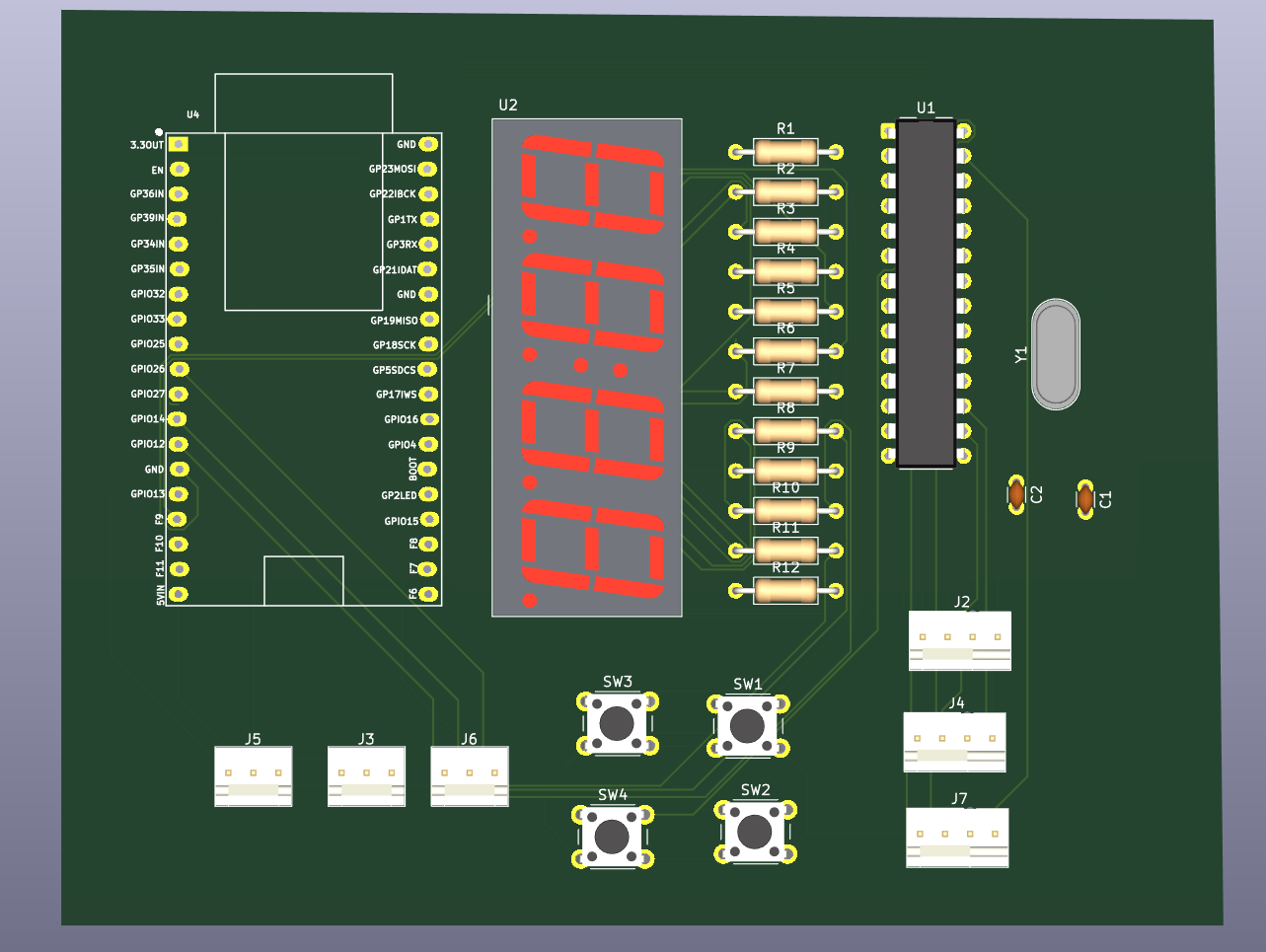


Figure 19 : ajoute des connexions du dessus

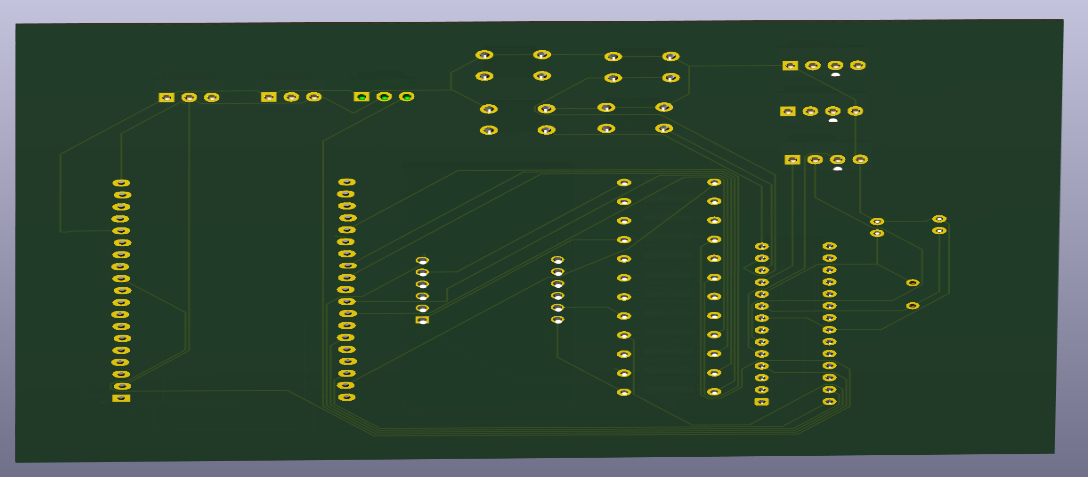


Figure 20 : connexion du bas de la carte

|  |  |
| --- | --- |
| C:\Users\Patrick\Pictures\Capturere.PNG  Figure 21 : Schéma électronique des LEDs | C:\Users\Patrick\Pictures\Captureled.PNG  Figure 22 : PCB des LEDs |
| C:\Users\Patrick\Pictures\led2.PNG  Figure 23: Rendu Final du PCB des LEDs  **Durant la réalisation de ce projet, nous avons rencontré deux difficultés majeures :**   * + 1. **L'absence du composant dans les librairies :**   Le modèle de l'ESP32 à 38 broches que nous utilisions n'était pas disponible dans les librairies standards de KiCad. Nous avons donc dû en créer un manuellement, en concevant à la fois le symbole schématique et l'empreinte PCB, afin qu'ils correspondent parfaitement au composant physique en notre possession.   * + 1. **La complexité du routage :**   La complexité du circuit, couplée à l'absence de fonction de routage automatique fiable dans KiCad pour ce type de projet, a rendu le tracé des pistes particulièrement difficile. Pour résoudre les problèmes d'intersections et de croisements, nous avons été contraints d'utiliser systématiquement les deux couches de la carte PCB, passant de la face supérieure à la face inférieure | |

## **4.3. Module d’Alimentation**

La conception d'un système électronique fonctionnel repose sur un module d'alimentation fiable, capable de fournir les tensions et les courants nécessaires à tous les composants. Pour notre projet, nous avons utilisé un **module d'alimentation pour plaque d'essai MB102**, une solution simple et efficace, particulièrement adaptée aux phases de prototypage.

* **Caractéristiques et fonctionnement**

Le module d'alimentation MB102 est un composant polyvalent qui s'insère directement sur une plaque d'essai (breadboard) et élimine le besoin de concevoir un circuit d'alimentation complexe. Ses principales caractéristiques sont :

* **Double sortie régulée :** Il fournit simultanément des tensions de **5V** et de **3.3V**, qui sont les tensions standards requises par la plupart des microcontrôleurs (comme l'ATMEGA328p) et des modules (ESP32, capteurs, etc.).
* **Source d'alimentation flexible :** Le module peut être alimenté de deux manières différentes : soit via une prise jack DC avec une tension d'entrée comprise entre **6.5V et 12V**, soit directement via un port USB.
* **Contrôle indépendant des rails :** Le module possède deux jeux de cavaliers (jumpers) qui permettent de choisir la tension de sortie (0V, 3.3V ou 5V) pour les rails de puissance supérieur et inférieur de la plaque d'essai, de manière indépendante.
* Rôle dans le projet

Le module d'alimentation MB102 a été un élément crucial de notre montage pour plusieurs raisons :

* **Alimentation du microcontrôleur et des capteurs :** Le microcontrôleur ATMEGA328p et le module ESP32 ont été alimentés en 5V, assurant ainsi leur bon fonctionnement et leur stabilité. Les capteurs tels que le DHT11 ont également été alimentés par cette même ligne de tension.
* **Alimentation des afficheurs 7 segments :** Les afficheurs, qui sont des composants gourmands en énergie, ont été alimentés par la ligne 5V pour garantir une luminosité optimale et éviter les baisses de tension qui pourraient affecter les autres composants du circuit.
* **Simplification du câblage :** L'utilisation de ce module a permis de grandement simplifier le câblage sur la plaque d'essai, en évitant la complexité d'un circuit de régulation de tension sur mesure. Cela a facilité le prototypage et le débogage.

Le module MB102 a été essentiel pour assurer une alimentation stable et adaptée à chaque composant, ce qui a été un facteur clé dans la réussite de la mise en œuvre de notre projet.



Figure 24 : Module d’alimentation arduino

# **RESULTATS ET ANALYSE**

Cette section traite les étapes de la réalisation physique du système, depuis la phase de prototypage jusqu'à la mise en service du circuit final. Nous y présentons le système achevé, les procédures de test, ainsi que les difficultés rencontrées et les solutions mises en place.

## **Présentation du système final**

Le système final est le résultat de l'intégration réussie des composants matériels et logiciels conçus. Il se compose des éléments principaux suivants :

* **Un microcontrôleur ATMEGA328p :** Cœur du système, il orchestre toutes les fonctions, de l'affichage de l'heure à la gestion des capteurs et des boutons.
* **Six afficheurs 7 segments :** Connectés en multiplexage pour afficher l'heure et d'autres informations numériques comme la température et l'humidité.
* **Un module horloge DS3231 (RTC) :** Non mentionné dans la documentation du projet initial, mais essentiel pour garantir la précision de l'horloge même en cas de coupure de courant, ce qui est une amélioration pertinente à souligner.
* **Un capteur DHT11 :** Intégré pour mesurer la température et l'humidité ambiante.
* **Un module ESP32 :** Utilisé pour la connectivité Wi-Fi, permettant au système de se connecter au réseau et de transmettre les données vers une interface informatique externe.
* **Des boutons poussoirs, tactiles et des LEDs :** Les boutons servent à interagir avec le système (par exemple pour régler l'heure ou déclencher l'alarme), et les LEDs fournissent des indications visuelles de l'état du système (ex. : alarme activée, connexion Wi-Fi).

Le circuit électronique a été conçu à l'aide du logiciel **KiCad**. Cette phase a inclus la réalisation du schéma électrique, la conception du circuit imprimé (PCB) et le routage des pistes pour garantir une connexion fiable entre tous les composants. Le prototype final a été soigneusement assemblé sur board Arduino dans un premier temps, en veillant à l'intégrité de toutes les connexions pour minimiser les erreurs.

Lors que le système est enclenché avec tous les composants connectés à leurs places : sur PC on a l’interface graphique qui auto-actualise toutes les données toutes les secondes, ce qui fait un temps de réponse de l’ensemble du système plutôt bien pour un système à usage domotique de base comme le nôtre. Dès qu’on envoie la commande sur la touche tactile d’enclenchement (fil rouge) **le** **minuteur** sur seven segment s’allume directement ; si on appuie la touche tactile (fil marron) on peut basculer les différentes fonctionnalités sur écran LCD entre autre **le calendrier**, la **station métrologique et la détection de présence à partir de 16cm avec activation immédiat du Buzzer** jusqu’à retrait de l’intrus. De l’autre côté on a les 4 boutons poussoirs qui sert de commande sur les différentes fonctionnalités implémentées comme par exemple le réglage de l’heure et de la date. Il est important de noter que le système est extensible, on peut ajouter et personnaliser un peu comme on veut, il laisse place à la créativité et l’imagination de l’utilisateur*.*

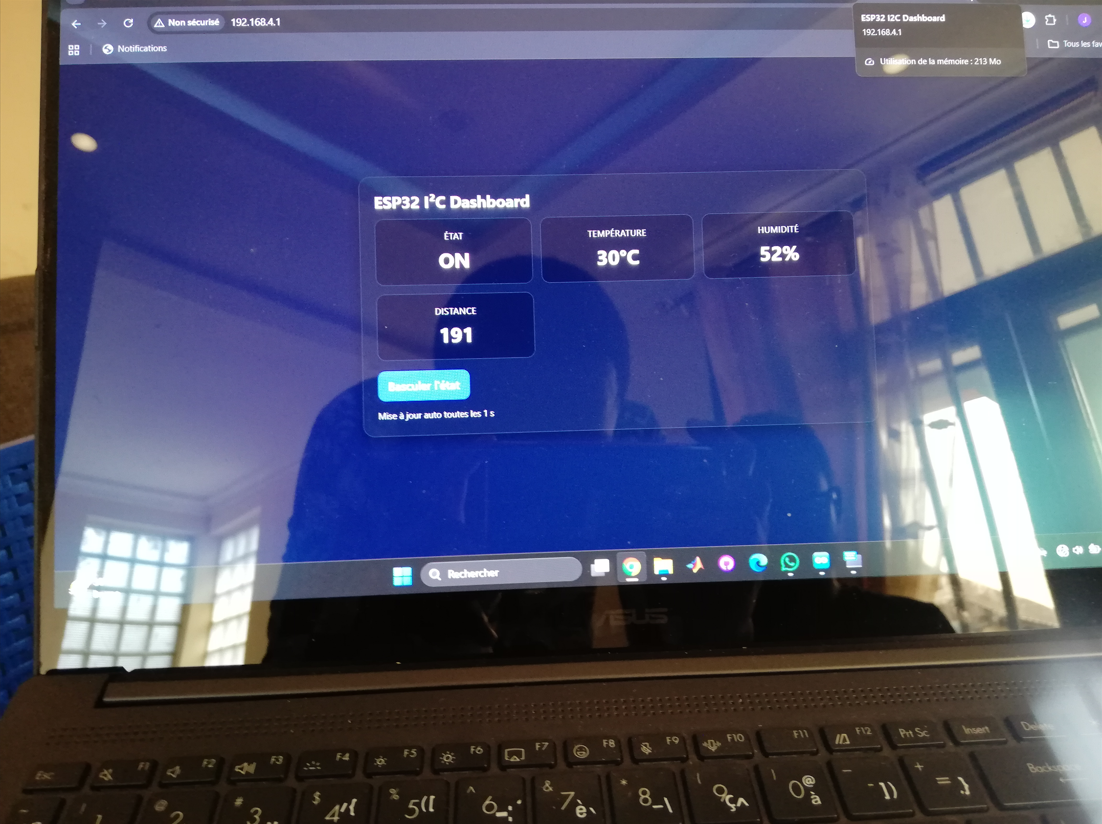


Figure 25 : Monitoring sur PC

|  |  |
| --- | --- |
| Figure 26 : Station météorologique | Figure 27 : Circuit entier sur Breadboard |
| C:\Users\Patrick\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\A1CB608A30FC2883EED0831DCF25F260\WhatsApp Image 2025-09-02 à 18.07.57_3c14a0fe.jpg  Figure 28 : Affichage de l'heure | C:\Users\Patrick\Pictures\Whah.jpg  Figure 29 : Affichage de la distance |
| C:\Users\Patrick\Pictures\WhatsApp Im.jpg  Figure 30 : Affichage du seven segment | |

## **5.2. Tests effectués et résultats obtenus**

Une fois le système assemblé, une série de tests rigoureux a été menée pour valider chaque fonctionnalité et s'assurer de la conformité avec le cahier des charges.

* **Test de l'horloge et du calendrier numérique :** Nous avons d'abord vérifié que l'horloge affichait correctement l'heure, la date et que le réglage via les boutons-poussoirs fonctionnait.
* **Test du minuteur et changement des fonctionnalités :** Les tests ont confirmé que les afficheurs 7 segments affichaient les chiffres de manière stable et lisible allant de 0 à 999. A l’aide des touches tactiles nous avons également changé de fonctionnalité après avoir enclenché le minuteur.
* **Test du capteur DHT11 :** Les valeurs de température et d'humidité lues par le capteur ont été comparées à celles d'un thermomètre et d'un hygromètre de référence. Les résultats ont montré une bonne corrélation, validant le bon fonctionnement de la station météo.
* **Test de la connectivité ESP32 :** Le module Wi-Fi a été testé pour sa capacité à se connecter à un réseau local et à transmettre les données de température et d'humidité à un serveur Web ou à une interface de monitoring. Les tests ont confirmé la transmission des données, ouvrant la voie à des applications de monitoring à distance et ce avec une actualisation des données toutes les secondes.
* **Tests de l'alarme et des LEDs :** Nous avons vérifié que les LEDs s'allumaient en réponse aux commandes des boutons et que le système d'alarme se déclenchait comme prévu.

## **5.3 Problèmes rencontrés et solutions proposées**

La réalisation de ce projet n'a pas été exempte de défis, qui ont été résolus grâce à une approche systématique de débogage.

* **Problème d’affichage sur les seg :** Le premier défi a été la gestion des 4 afficheurs 7 segments. Initialement, l'affichage était instable et les chiffres clignotaient ou affichaient des valeurs incorrectes ce probleme a été reolu.
  + **Solution :** Ce problème a été résolu en optimisant le code de la routine de d’affichage des caractères adapté.
* **Problème de communication entre modules :** Nous avons rencontré des difficultés à établir une communication fiable entre le microcontrôleur ATMEGA328p et le module ESP32.
  + **Solution :** Après investigation, nous avons identifié un problème de connexion physique entre les trois devices en verifiant que les pins soient correctement connectés.
* **Problème d’impression du PCB :** A cause de l’indisponibilité de la machine à imprimé après conception nous n’avons guère pu imprimé notre fameux carte PCB. Donc nous en sommes restés au circuit monté sur Board Body Arduino avec toutes ses imperfections malheureusement.

# **CONCLUSION ET PERSPECTIVES**

## **6.1 Synthèse des résultats obtenus**

Ce projet nous a permis de mettre en application les concepts théoriques du cours de Systèmes et Circuits Électroniques, en menant à bien la conception et la réalisation d'un système électronique multifonctionnel et connecté. L'objectif principal, qui était de nous familiariser avec la conception d'un système autour d'un microcontrôleur, a été atteint avec succès.

Nous avons pu concevoir un système qui intègre plusieurs fonctionnalités clés : une horloge numérique avec affichage sur six afficheurs sept segments, une station météo capable de mesurer la température et l'humidité, et un système d'alarme et de notification. L'intégration d'un module ESP32 a permis d'ajouter une dimension de connectivité Wi-Fi, assurant ainsi la transmission des données vers une interface informatique.

Sur le plan technique, nous avons maîtrisé l'intégration de divers composants (ATMEGA328p, capteur DHT11, afficheurs, LEDs, etc.) et leur programmation en C/C++. Ce projet nous a également confrontés aux défis pratiques de l'électronique, notamment la gestion des entrées/sorties, le multiplexage des afficheurs et la communication série entre les différents modules, enrichissant ainsi notre expérience en ingénierie électronique.

## **6.2 Améliorations possibles et perspectives d’évolution du projet**

Le système réalisé constitue une base solide, mais plusieurs pistes d'amélioration pourraient être explorées pour en étendre les fonctionnalités et optimiser les performances :

* **Interface utilisateur avancée :** Le rapport initial mentionne la présence d'un écran LCD. L'intégration de cet écran permettrait d'afficher la température, l'humidité et les notifications directement sur le dispositif, offrant une interface plus conviviale et autonome sans nécessiter de connexion à un ordinateur.
* **Domotique élargie :** Les fonctionnalités pourraient être étendues pour un contrôle plus complet de la maison connectée. Cela pourrait inclure la commande de l'éclairage, l'ouverture et la fermeture de volets ou de portes via des relais, ou encore la gestion de la consommation énergétique.
* **Connectivité et Cloud :** Bien que le module ESP32 assure une connectivité Wi-Fi, une amélioration majeure consisterait à connecter le système à un service de cloud public (comme Firebase ou ThingSpeak) pour permettre le monitoring et le contrôle à distance via une application mobile.
* **Optimisation de l'alimentation :** Une étude plus poussée pourrait porter sur l'optimisation de la consommation énergétique, par exemple en utilisant un régulateur de tension plus performant ou en explorant des modes de veille du microcontrôleur, pour permettre une alimentation par batterie et rendre le système complètement portable.
* **Miniaturisation et production :** Une perspective d'avenir serait de redessiner la carte électronique (PCB) pour intégrer l'ensemble des composants sur un circuit imprimé plus compact, ce qui rendrait le dispositif plus esthétique et adapté à une production à plus grande échelle.

# **VII. ANNEXES**

## **Code source complet**

# **BIBLIOGRAPHIE**

Pour la rédaction de ce rapport, nous nous sommes appuyés sur une combinaison de ressources académiques, de documentation technique de composants et de sources en ligne fiables. Ces références nous ont permis de consolider nos connaissances théoriques, de comprendre les spécifications techniques et de mettre en œuvre les meilleures pratiques de conception

## **8.1. Ouvrages et Rapports**

**[1]** KUTI LUSALA, V.A. *Systèmes et Circuits Électroniques, Faculté Polytechnique*, Université de Kinshasa, 2024-2025.

**[2]** BANZI, M. *Getting Started with Arduino*, 2014.

**[3]** OULD, A. *Embedded Systems Testing: A Comprehensive Guide*, 2016.

**[4]** POON, J. *KiCad EDA: A Complete Guide to Electronic Design Automation*, 2016.

**[5]** KUTI LUSALA, V.A. *Capteurs et Instrumentation, Faculté Polytechnique*, Université de Kinshasa, 2024-2025.

**[6]** Groupe 4, Rapport de *Projet Systèmes Electroniques, Facultés Polytechniques,* Université de Kinshasa, 2023-2024.

## **8.2. Webographie**

**[1]**https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\_Datasheet.pdf.

**[2]** <https://arduino-france.site/dht11-arduino/>

**[3]** <https://arduino-france.site/buzzer-arduino/>

**[4]** <https://arduino-france.site/lcd-1602/>

**[5]**[https://www.gotronic.fr/art-module-nodemcu-esp32 28407.htm?srsltid=AfmBOoodPypqH4\_BnX6hBXGALd7-XxDwPcBE-0FZcWPDg6-a3kqTV68t](https://www.gotronic.fr/art-module-nodemcu-esp32%2028407.htm?srsltid=AfmBOoodPypqH4_BnX6hBXGALd7-XxDwPcBE-0FZcWPDg6-a3kqTV68t)

**[6]** https://arduino-france.site/ultrason-hc-sr04/

**[7]** Documentation officielle du microcontrôleur ATmega328P. *Microchip*, <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>.

**[8]** Documentation du module d'alimentation MB102. *Scribd*, <https://fr.scribd.com/document/485540938/mb102-ps>.

**[9]** Wikipedia. *Affichage à sept segments*, <https://fr.wikipedia.org/wiki/Affichage_%C3%A0_sept_segments>.

1. https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\_Datasheet.pdf. [↑](#footnote-ref-1)
2. https://arduino-france.site/dht11-arduino/ [↑](#footnote-ref-2)
3. https://arduino-france.site/buzzer-arduino/ [↑](#footnote-ref-3)
4. https://arduino-france.site/lcd-1602/ [↑](#footnote-ref-4)
5. https://www.gotronic.fr/art-module-nodemcu-esp32-28407.htm?srsltid=AfmBOoodPypqH4\_BnX6hBXGALd7-XxDwPcBE-0FZcWPDg6-a3kqTV68t [↑](#footnote-ref-5)
6. https://arduino-france.site/ultrason-hc-sr04/ [↑](#footnote-ref-6)