



SYSTÈME D'ALARME INCENDIE INTELLIGENT

GCSE 2
2025/2026

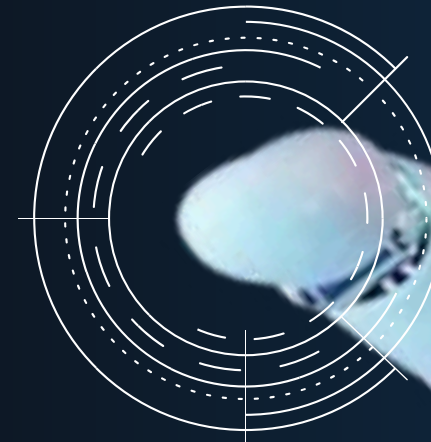
Encadrée par
Pr Moutakki

Présentée par
Amine BELAMINE
Saad EL GUELYOUY

Start

SOMMAIRE

1. CHAPITRE I : ÉTUDE BIBLIOGRAPHIQUE
ET TECHNOLOGIQUE
2. CHAPITRE II : ANALYSE ET
CONCEPTION MATÉRIELLE
3. CHAPITRE III : DÉVELOPPEMENT
LOGICIEL
4. CHAPITRE IV : RÉALISATION, TESTS ET
RÉSULTATS
5. CONCLUSION GÉNÉRALE ET
PERSPECTIVES





INTRODUCTION



Ce projet propose une solution de sécurité intelligente et connectée, basée sur le microcontrôleur ESP32.

Il surveille en permanence les risques d'incendie, de gaz et de surchauffe. Contrairement aux alarmes classiques, ce système ne se contente pas de sonner : il offre un suivi en temps réel via une page Web et envoie une alerte immédiate sur Telegram en cas de danger, assurant une protection totale, même à distance.

PROBLÉMATIQUE

Comment concevoir un système de sécurité abordable, capable de détecter plusieurs types de dangers (incendie, gaz, surchauffe) et d'alerter l'utilisateur instantanément, où qu'il soit dans le monde ?



OBJECTIFS DU PROJET

Ce projet vise à créer un prototype fonctionnel capable de :

- Surveiller en permanence la température, la présence de gaz/fumée et les départs de feu.
- Afficher les données en local sur un écran LCD et à distance sur une interface Web moderne.
- Alerter via une notification Telegram sur smartphone et une alarme sonore/visuelle en cas de danger.
- Éviter les faux positifs grâce à un algorithme de vérification logiciel.

ANALYSE MATÉRIELLE (HARDWARE)

Analyse Matérielle (Hardware).

Le système repose sur le microcontrôleur ESP32, choisi pour sa double connectivité Wi-Fi/Bluetooth et sa puissance de calcul.

les composants utilisées

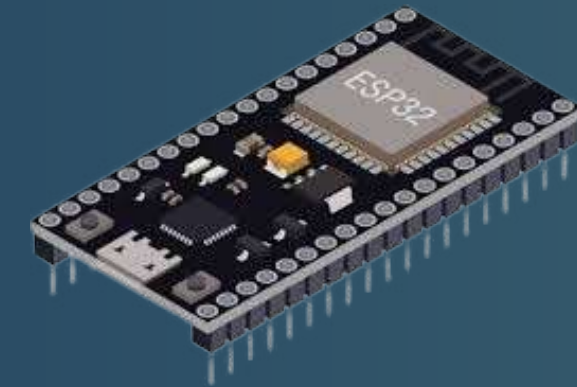
- ESP32 : Contrôle l'ensemble du système et la communication.
- DHT11 : Mesure la température et l'humidité.
- Capteur de flamme : Détecte la présence de feu.
- Capteur de gaz : Détecte la fumée et les gaz.
- Buzzer & LED : Signalent une alerte sonore et visuelle.
- LCD I2C : Affiche l'état du système.

Composant	Pin ESP32	Type de signal
Capteur Flamme	GPIO 4	Entrée Numérique (Digital)
Capteur Fumée	GPIO 1	Entrée Analogique (ADC)
Capteur DHT11	GPIO 15	Entrée Numérique (OneWire)
Buzzer	GPIO 16	Sortie (PWM/Tone)
LED	GPIO 17	Sortie Numérique
LCD (SDA)	GPIO 8	Communication I2C
LCD (SCL)	GPIO 9	Communication I2C

COMPOSANTS MATÉRIELS (HARDWARE)

Microcontrôleur ESP32

Le cerveau du système. Microcontrôleur bi-cœur avec Wi-Fi et Bluetooth intégré, gère la logique de détection et pilote tous les périphériques via ses interfaces multiples (UART, I2C, ADC).



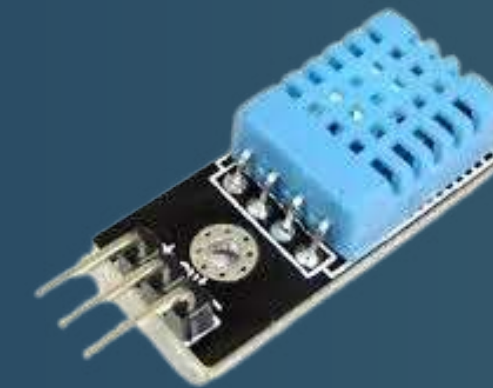
Capteur de Gaz

Détecte la présence de fumée et de gaz combustibles. Utilisé en lecture analogique pour une mesure précise de la concentration.



Capteur de Température DHT11

Mesure la température ambiante et l'humidité. Sert d'indicateur préventif de surchauffe.



COMPOSANTS MATÉRIELS (HARDWARE)

Capteur de Flamme

Détecte le rayonnement infrarouge émis par une flamme nue, confirmant un incendie déclaré.



Affichage Écran LCD avec Module I2C

Affiche le statut du système (OK/ALARME) et les mesures clés (Température) en temps réel. Le module I2C réduit le câblage à deux fils.



Alerte Sonore Buzzer Piézoélectrique Actif

Fournit une alarme sonore forte et immédiate en cas de danger détecté.



COMPOSANTS MATÉRIELS (HARDWARE)

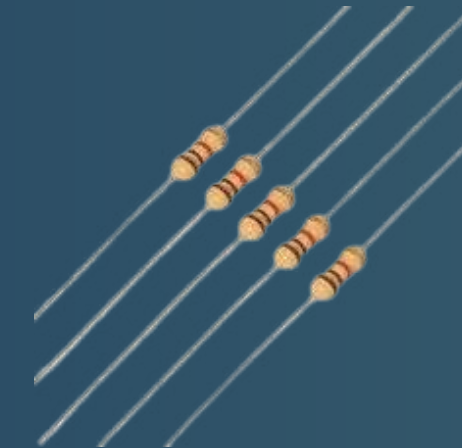
Témoin Visuel LEDs

Indiquent l'état du système : Vert pour normal, Rouge pour alerte. Utilisées avec des résistances de limitation de courant.



Résistance

Principalement utilisé pour protéger les LEDs (Rouge et Verte) en limitant le courant qu'elles reçoivent de l'ESP32.



ARCHITECTURE LOGICIELLE (SOFTWARE)

Le programme est développé en C++ sous l'IDE Arduino. Il utilise une architecture "boucle principale" (Loop) non bloquante pour gérer le serveur web et les capteurs simultanément.

Bibliothèques Utilisées

- *WiFi.h & WiFiClientSecure.h : Pour la connexion au routeur et la communication sécurisée (SSL) avec les serveurs de Telegram.*
- *WebServer.h : Pour transformer l'ESP32 en un serveur HTTP capable de servir des pages Web.*
- *UniversalTelegramBot.h : Pour interagir simplement avec l'API Telegram Bot.*
- *DHT.h & LiquidCrystal_I2C.h : Pilotes pour les capteurs et l'écran.*

ARCHITECTURE LOGICIELLE (SOFTWARE)

Algorithme de Détection Intelligent

Pour éviter que l'alarme ne se déclenche à cause d'une interférence lumineuse (sur le capteur de flamme), un algorithme de vérification temporelle a été implémenté

1. Si le capteur détecte une flamme (HIGH), on enregistre l'heure (millis()).
2. On attend une confirmation de 500ms.
3. Si après 500ms le signal est toujours présent, l'alarme est validée. Sinon, c'est considéré comme un faux positif.

Seuils de Sécurité

Les seuils d'alerte ont été calibrés comme suit :

- Gaz/Fumée : Valeur analogique > 2500 (sur 4095).
- Température : > 45.0°C.
- Feu : Détection logique HIGH confirmée.

INTERFACE UTILISATEUR & EXPÉRIENCE (UX)

Interface Web "Cyberpunk" (V4.0).

Contrairement aux interfaces brutes habituelles sur Arduino, ce projet intègre une interface HTML5/CSS3 soignée directement dans le code C++.

- Design : Style futuriste sombre, polices "Orbitron" importées de Google Fonts.
- Glassmorphism : Les cartes de données utilisent des effets de transparence et de flou.
- Réactivité : La page se rafraîchit automatiquement toutes les 3 secondes (meta refresh).
- Dynamisme : Les couleurs changent (Cyan vers Rouge) et les éléments "pulsent" (animation CSS) en cas de danger.

Notifications Telegram

L'utilisation de Telegram permet une alerte instantanée. Le bot, identifié par son Token unique, envoie un message formaté avec des émojis précisant la nature exacte du danger (Feu, Gaz ou Chaleur) directement sur le téléphone de l'utilisateur (Chat ID sécurisé).

CÂBLAGE ET ALIMENTATION

Tests et Résultats

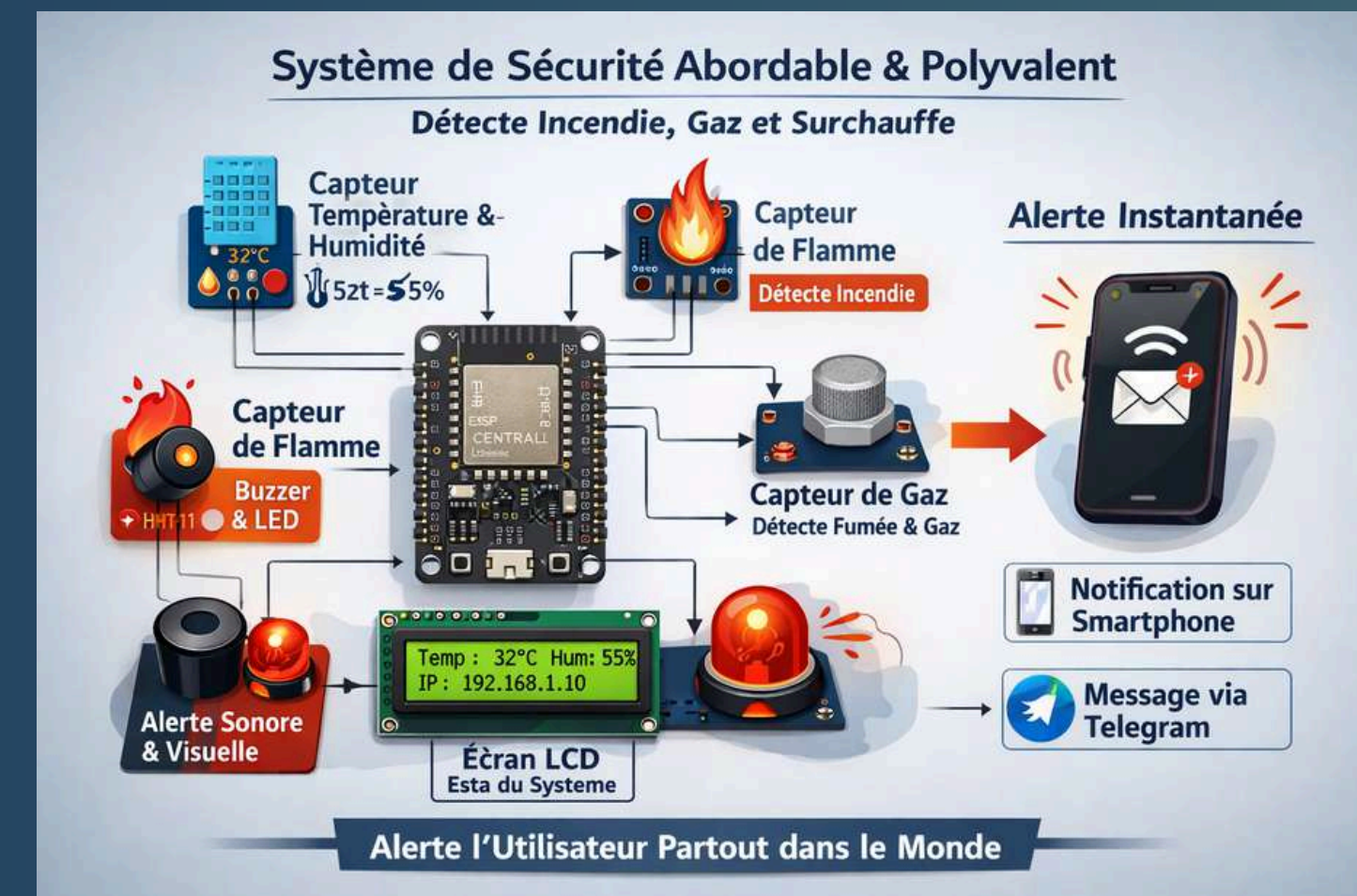
Scénario Nominal (Tout va bien)

- LCD : Affiche la température, le niveau de gaz et l'IP défilante.
- Web : Affiche "SYSTÈME SÉCURISÉ" en vert/cyan.
- Actuateurs : LED éteinte, Silence.
-

Scénario d'Alerte (Test avec briquet/gaz)

Dès qu'un seuil est franchi :

1. Réaction immédiate (<200ms) : Le Buzzer émet un son rythmé et la LED clignote.
2. Notification (<2s) : Un message "🚨 ALERTE DANGER !" arrive sur Telegram.
3. Web : L'interface passe au ROUGE et affiche "DANGER CRITIQUE".





CONCLUSION

6.1 Bilan

Ce projet a permis de réaliser un système de sécurité complet, autonome et connecté. Il démontre la puissance de l'ESP32 pour gérer simultanément l'acquisition de données physiques, un serveur web graphique complexe et une communication cloud sécurisée.

6.2 Améliorations futures

Pour passer du prototype au produit commercialisable, nous pourrions ajouter :

- **Batterie de secours :** Pour fonctionner même en cas de coupure de courant.
- **Module GSM (SIM800L) :** Pour envoyer des SMS si le Wi-Fi est coupé.
- **Boîtier imprimé en 3D :** Pour protéger les composants électroniques.
- **Application Mobile dédiée :** Pour remplacer la page Web et avoir des notifications push natives.



THANK

YOU

FOR

YOUR

ATTENTION