



Systeme de Détection de Proximité IR avec Dashboard Web Temps Réel

Architecture Edge Standalone sur M5Stack Tab5

SKandar HALOUANI SNPI5 – 2024

Développement d'un système embarqué autonome de détection et de visualisation temps réel

Introduction et Contexte du Projet

Objectifs Principaux

Ce projet vise à concevoir et implémenter un système de détection de proximité infrarouge complet, fonctionnant en mode Edge Computing totalement autonome. L'architecture repose sur le M5Stack Tab5 comme plateforme centrale, intégrant acquisition de données, traitement local, interface tactile et diffusion web temps réel.

Le système se distingue par sa capacité à opérer sans infrastructure cloud, offrant une latence minimale et une confidentialité maximale des données. L'objectif est de démontrer qu'un dispositif embarqué moderne peut gérer simultanément l'acquisition de capteurs, le traitement du signal, l'interface utilisateur locale et un serveur web avec communication bidirectionnelle.

Motivation Technique

Dans le contexte actuel de l'IoT et de l'industrie 4.0, la capacité à traiter les données à la périphérie du réseau devient cruciale. Ce projet explore les limites des microcontrôleurs ESP32 en démontrant qu'il est possible d'implémenter des fonctionnalités avancées traditionnellement réservées aux systèmes plus puissants.

L'approche Edge standalone répond à des besoins réels : réduction de la latence pour les applications critiques, protection des données sensibles, fonctionnement en environnement déconnecté, et réduction des coûts d'infrastructure cloud. Ce projet constitue une base solide pour des applications industrielles de surveillance et de détection.

Architecture Globale du Système Edge

Couche Matérielle

M5Stack Tab5 (ESP32-S3) comme unité centrale de traitement avec écran tactile IPS 7" intégré

- Capteur IR digital sur GPIO 1
- Haut-parleur intégré pour alarmes
- Interface tactile capacitive

Couche Logicielle

Firmware embarqué développé avec Arduino Framework et bibliothèques optimisées

- Serveur HTTP/WebSocket asynchrone
- Gestionnaire d'états temps réel
- Algorithmes de filtrage signal

Couche Communication

Point d'accès WiFi autonome créé par smartphone en mode hotspot

- Protocol HTTP pour pages statiques
- WebSocket pour flux temps réel

Couche Présentation

Interfaces utilisateur dual : locale sur Tab5 et distante via navigateur web

- Dashboard responsive HTML5/CSS3
- Visualisations animées Canvas
- Contrôles interactifs

Cette architecture en couches garantit une séparation claire des responsabilités tout en optimisant les performances du système embarqué. Chaque composant peut être testé et maintenu indépendamment, facilitant le développement itératif et le débogage.

Justification de l'Architecture Choisie

Architecture 3 – Tab5 Standalone Edge Computing

Après analyse comparative de plusieurs architectures possibles (cloud centralisé, edge-cloud hybride, standalone complet), l'architecture standalone a été retenue pour ses avantages décisifs dans le contexte du projet.

Avantages Techniques Majeurs

- **Latence ultra-faible** : Traitement local permettant des temps de réponse inférieurs à 50ms entre détection et affichage, crucial pour les applications de sécurité
- **Autonomie complète** : Aucune dépendance à une infrastructure externe, fonctionnement possible en environnement isolé ou mobile
- **Sécurité renforcée** : Les données restent confinées au réseau local, éliminant les risques liés aux transmissions Internet et au stockage cloud
- **Coût d'exploitation nul** : Pas d'abonnement cloud, pas de frais de bande passante, investissement matériel unique
- **Simplicité de déploiement** : Configuration minimale, pas de dépendances à des services tiers

Contraintes Acceptées

Cette architecture impose des limitations en termes de puissance de calcul et de stockage, mais ces contraintes sont largement compensées par les avantages pour l'application visée. La portée WiFi limite la distance d'accès au dashboard, mais reste suffisante pour un usage local ou semi-mobile.

<50ms

Latence système

Du capteur à l'affichage

100%

Autonomie

Fonctionnement sans cloud

Schéma d'Architecture Système Complet



Flux de Données et Communication

Communication Bidirectionnelle : Le système établit un canal WebSocket persistant permettant la transmission temps réel des données de détection vers le client web, ainsi que la réception des commandes utilisateur (ajustement sensibilité, activation/désactivation alarme).

Synchronisation Multi-Interface : L'état du système est maintenu de façon cohérente entre l'interface tactile Tab5 et le dashboard web. Toute modification sur l'une des interfaces se reflète instantanément sur l'autre grâce au gestionnaire d'états centralisé.

Protocoles Utilisés :

- HTTP/1.1 pour servir les ressources statiques (HTML, CSS, JavaScript)
- WebSocket (RFC 6455) pour le streaming bidirectionnel temps réel
- JSON comme format d'échange structuré de données
- mDNS pour la découverte du service sur le réseau local

Schéma de Câblage et Connexions Hardware

Connexion du Capteur IR

Le capteur infrarouge de proximité est un module digital simple fournissant une sortie binaire (HIGH/LOW) en fonction de la détection d'obstacle. Il est alimenté directement par les rails du Tab5.

Brochage Détaillé

- **GPIO 1 (Tab5) → Signal (Capteur IR)** : Ligne de données digitale avec résistance pull-up interne activée
- **3.3V (Tab5) → VCC (Capteur IR)** : Alimentation stabilisée 3.3V, courant max 50mA
- **GND (Tab5) → GND (Capteur IR)** : Masse commune du système

Caractéristiques Électriques

Le capteur fonctionne en logique 3.3V compatible ESP32. Le GPIO 1 est configuré en mode INPUT_PULLUP, assurant un état stable HIGH en l'absence de détection. Lors d'une détection, le capteur tire la ligne à LOW (0V).

Une capacité de découplage de 100nF est recommandée près du capteur pour filtrer le bruit sur l'alimentation, bien que le système fonctionne correctement sans dans la plupart des cas.

📌 **Note Importante** : Le GPIO 1 du Tab5 est un port général polyvalent ne présentant pas de conflit avec les périphériques internes (contrairement aux GPIOs réservés pour l'écran ou le SPI Flash). Vérifier la documentation Tab5 avant modification du brochage.

Configuration Logicielle

```
const int IR_PIN = 1;pinMode(IR_PIN, INPUT_PULLUP);// Lecture : LOW =  
détectionbool detected = !digitalRead(IR_PIN);
```

Pipeline Complet de Traitement du Signal

01

Acquisition Signal Brut

Lecture GPIO toutes les 10ms via interruption ou polling rapide, capture de l'état binaire du capteur IR

03

Validation Stabilité

Vérification que l'état détecté persiste pendant au moins 3 lectures consécutives avant de considérer le changement comme valide

05

Diffusion Multi-Canal

Propagation simultanée vers l'affichage Tab5, le serveur WebSocket, le système d'alarme et le buffer d'historique

Ce pipeline garantit une détection fiable tout en optimisant les performances du microcontrôleur ESP32-S3. La latence totale du pipeline reste inférieure à 70ms dans le pire cas, permettant une réactivité temps réel pour l'utilisateur.

02

Filtrage Anti-Rebond

Application d'un algorithme de debouncing avec fenêtre temporelle de 50ms pour éliminer les faux positifs dus aux vibrations ou instabilités du capteur

04

Mise à Jour État

Actualisation de la variable globale d'état système et déclenchement des callbacks associés (interface, alarme, historique)

06

Enregistrement Historique

Stockage dans un buffer circulaire en RAM des 100 dernières valeurs avec timestamp pour génération du graphique temporel

Flowchart Détaillé du Système



Interface Locale sur M5Stack Tab5

Fonctionnalités de l'Interface Tactile

L'interface locale exploite pleinement les capacités du grand écran IPS 7" du Tab5 pour offrir une visualisation riche et des contrôles intuitifs directement sur le dispositif.

Éléments Visuels Principaux

- **Indicateur de Status** : Grande icône colorée en haut de l'écran affichant l'état actuel (vert = inactif, rouge = détection) avec animation pulsante lors des événements
- **Graphique Temps Réel** : Tracé linéaire des 60 dernières secondes de détection, mise à jour fluide toutes les 150ms, axes temporels automatiques
- **Compteurs Numériques** : Affichage du nombre total de détections depuis le démarrage et du temps de détection cumulé
- **Zone de Contrôle Tactile** : Boutons virtuels pour activation/désactivation alarme, reset compteurs, accès paramètres

Slider de Sensibilité Dynamique

Un contrôle slider horizontal permet d'ajuster en temps réel le seuil de sensibilité de détection. Les valeurs vont de 1 (très sensible, détections courtes acceptées) à 10 (peu sensible, nécessite détection prolongée). Le changement est appliqué immédiatement sans redémarrage.

L'interface locale constitue le point de contrôle principal du système, fonctionnant même sans connexion WiFi active. Elle offre une autonomie complète pour la surveillance et la configuration.



150ms

Rafraîchissement

Fréquence mise à jour graphique

📄 **Optimisation Graphique** : Pour maintenir la fluidité sur l'ESP32-S3, le graphique utilise un buffer circulaire optimisé et des techniques de double buffering évitant le flickering. Le rendu complet prend moins de 20ms, laissant 80% du temps CPU disponible pour les autres tâches.

Dashboard Web et Visualisations Avancées

Interface Web Responsive et Animations Canvas

Radar Animé Temps Réel

La visualisation phare du dashboard est un radar circulaire animé dessiné avec Canvas HTML5. Un faisceau lumineux tourne en continu à 360°, simulant un balayage radar professionnel. Lors d'une détection, des points rouges apparaissent sur la circonférence avec effet de fade-out progressif.

Implémentation Technique : Le radar utilise `requestAnimationFrame()` pour une animation fluide à 60fps. Les calculs trigonométriques sont optimisés avec des tables précalculées. La détection déclenche l'ajout de "blips" qui persistent 2-3 secondes avec opacité décroissante, créant un effet de traînée visuel impressionnant.

Bargraph Vertical

Un indicateur bargraph vertical affiche l'intensité ou la fréquence des détections récentes sur une échelle de 0 à 100%. Les barres sont colorées en dégradé vert-jaune-rouge selon l'intensité, avec animation de croissance/décroissance fluide grâce aux transitions CSS3.

Communication WebSocket Bidirectionnelle

Le dashboard établit une connexion WebSocket persistante au chargement de la page. Le protocole assure une latence minimale pour la transmission des événements de détection.

Messages Émis par le Serveur :

- `{"type": "detection", "value": true}` lors d'un événement
- `{"type": "stats", "count": 42, "uptime": 3600}` pour les statistiques
- `{"type": "history", "data": [...]}` pour le buffer historique

Messages du Client :

- `{"cmd": "sensitivity", "val": 7}` pour ajuster la sensibilité
- `{"cmd": "alarm", "state": "on"}` pour contrôler l'alarme

Le protocole inclut des heartbeats toutes les 30 secondes pour détecter les déconnexions et tenter une reconnexion automatique.