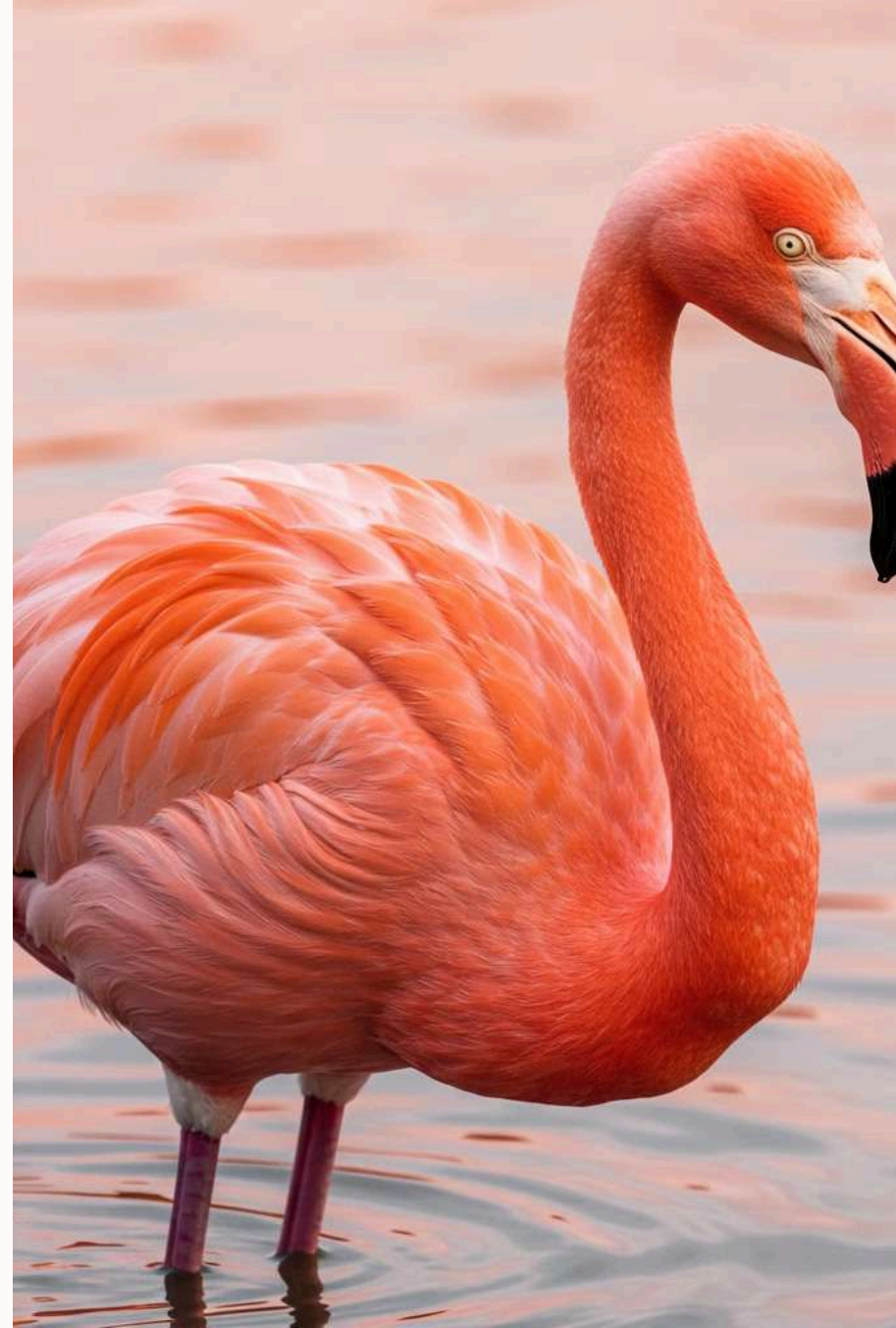


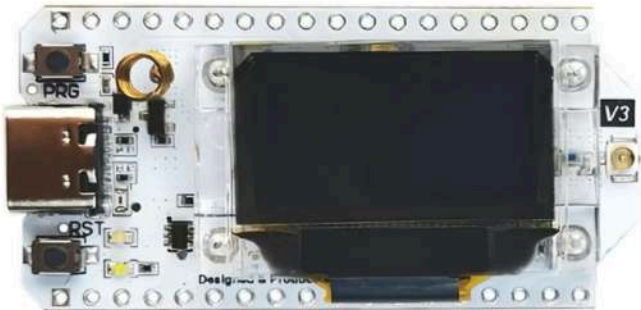


Rapport IOT

Clement HUNAULT & Roland Cédric TAYO

Avril 2025





Présentation de la carte Heltec WiFi LoRa 32

La carte Heltec WiFi LoRa 32 (V3) est un microcontrôleur polyvalent, largement utilisé dans le domaine de l'Internet of Things (IoT). Elle est dotée d'un ensemble de fonctionnalités permettant de concevoir des projets connectés et autonomes.

Caractéristiques principales



ESP32

Microcontrôleur
puissant



Connectivités

WiFi, Bluetooth, LoRa,
USB-C



Écran OLED

Affichage intégré
monochrome



Interfaces

SPI, I2C, GPIO





Utilisation dans le projet



Récolte de données

Capteurs connectés en
SPI/I2C



Affichage

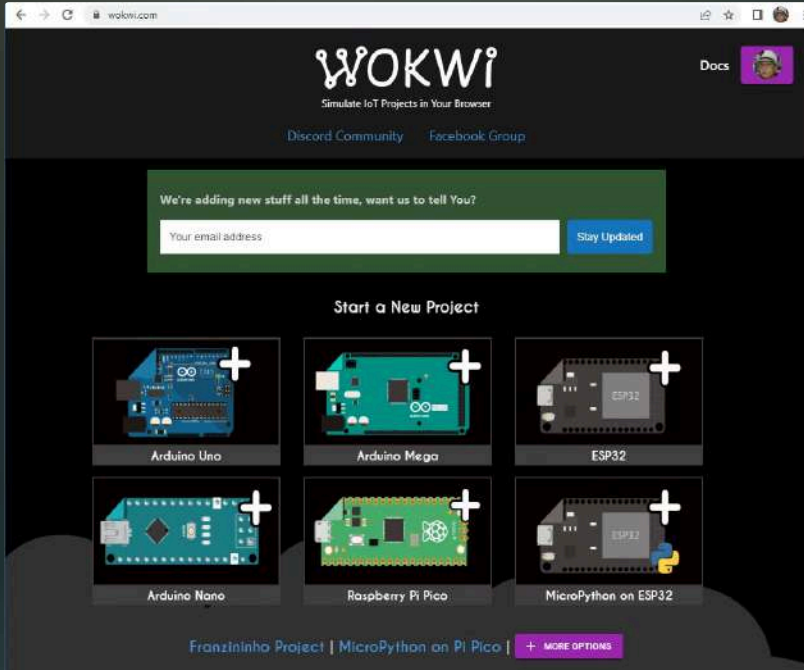
Informations sur écran
OLED



Transmission

Communication via réseau LoRa

Simulation du projet



ESP32 générique

Test logique et acquisition



OLED SSD1306

Simulation de l'affichage



Capteurs équivalents

Validation des interactions

Rappel connexion SPI

Principe

Le bus SPI (Serial Peripheral Interface) est une interface de communication synchrone permettant l'échange de données entre un maître (ici l'ESP32) et plusieurs esclaves (capteurs ou périphériques)

Chaque esclave dispose d'une broche SS spécifique, permettant au maître de choisir avec quel esclave il souhaite communiquer à un instant donné

Lignes principales

- SCLK : Signal d'horloge généré par le maître
- MOSI : Données envoyées du maître vers l'esclave
- MISO : Données envoyées de l'esclave vers le maître
- SS : Sélection de l'esclave

Rappel connexion I2C

Le bus I2C (Inter-Integrated Circuit) est un protocole de communication série synchrone permettant la communication entre un maître et plusieurs esclaves à l'aide de seulement deux lignes :

- SDA : Ligne de données (Serial Data Line) permettant l'échange bidirectionnel des données
- SCL : Ligne d'horloge (Serial Clock Line) générée par le maître

Chaque périphérique connecté au bus possède une adresse unique permettant au maître de le sélectionner. Contrairement au protocole SPI, il n'est pas nécessaire d'utiliser une broche de sélection (SS) par périphérique, ce qui réduit le nombre de fils nécessaires. Le bus I2C est particulièrement adapté aux systèmes embarqués et à la domotique, où plusieurs capteurs ou modules doivent être connectés sur un même bus



Environnement de développement Arduino

Interface

Simple et intuitive

Bibliothèques

Large catalogue compatible

Communauté

Exemples et ressources

Moniteur série

Visualisation en temps réel

Test de l'écran OLED



The screenshot shows the Arduino IDE interface. The top pane displays the code for a file named `lora.ino`. The code includes headers for SPI and LoRa, sets up a serial connection at 115200 baud, and prints "Envoi d'un message : Hello World" every 5 seconds in the loop function.

```
1 #include <SPI.h>  
2 #include <LoRa.h>  
3  
4 void setup() {  
5   Serial.begin(115200);  
6 }  
7  
8 void loop() {  
9   // Envoi d'un message  
10  Serial.println("Envoi d'un message : Hello World");  
11  delay(5000); // Envoi toutes les 5 secondes  
12 }  
13
```

The bottom pane shows the Serial Monitor window, which is set to receive data from "/dev/ttyUSB0" at 115200 baud. It displays multiple instances of the message "Envoi d'un message : Hello World".

Fonctionnement du code écran

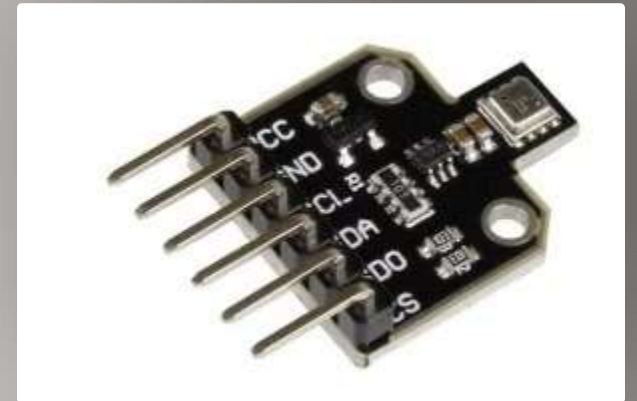
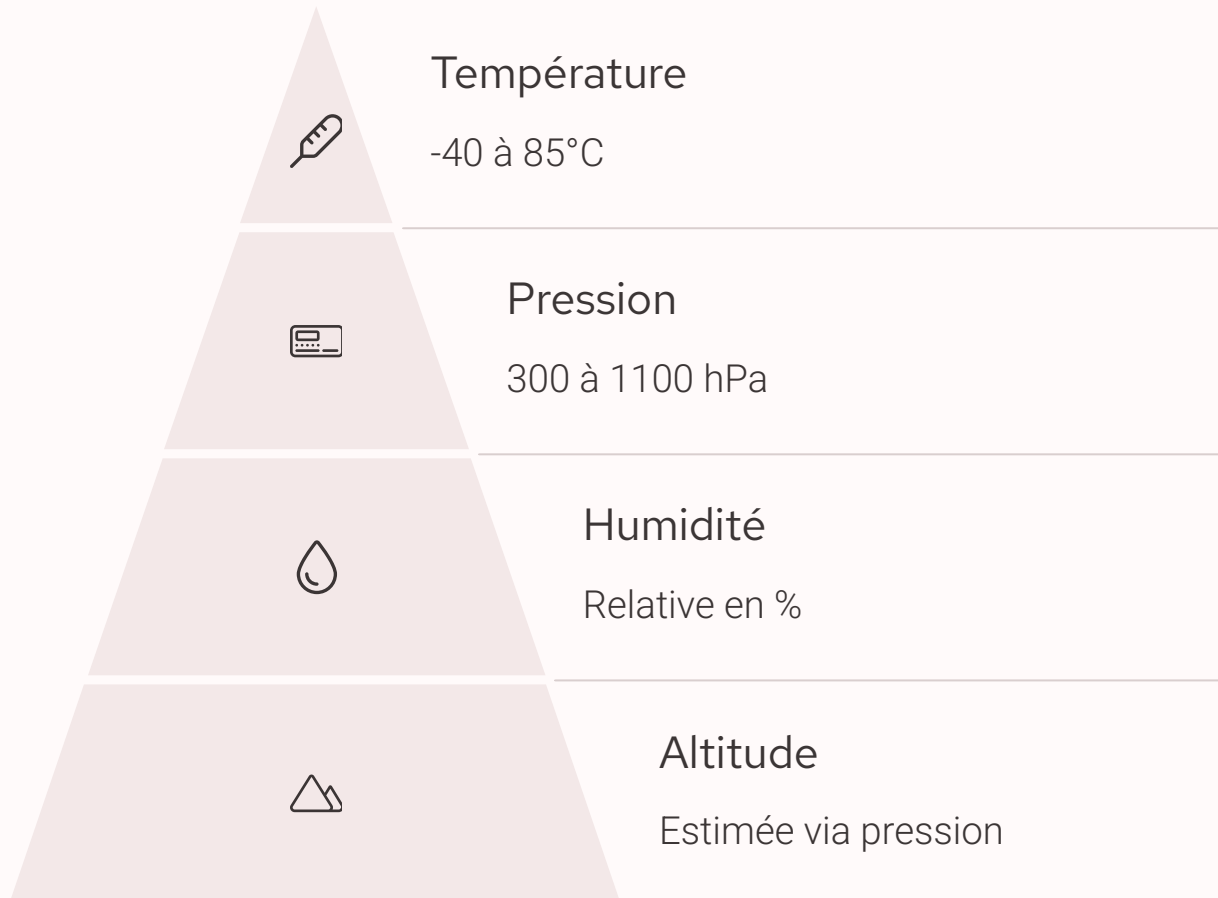
testScreen.ino

Le programme utilise les bibliothèques fournies par Heltec pour afficher successivement :

- Des lignes partant de chaque coin de l'écran (fonction drawLines())
- Des rectangles concentriques (fonction drawRect())
- Des rectangles remplis avec alternance de couleurs (fonction fillRect())
- Des cercles croissants depuis le centre de l'écran (fonction drawCircle())
- Un affichage de texte ligne par ligne (fonction printBuffer())

Ces tests permettent de s'assurer que l'écran OLED est correctement connecté et fonctionne comme attendu.

Capteur 1 : Joy-IT SEN-BME680



Fonctionnement du code BME680

testWeather.ino

Ce programme a pour but :

- D'initialiser la communication avec le capteur en SPI
- De configurer les paramètres de mesure (suréchantillonnage et filtre)
- De réaliser une lecture des données environnementales toutes les deux secondes
- D'afficher ces données sur le moniteur série : température, pression, humidité, qualité de l'air (résistance du gaz) et estimation de l'altitude

Le programme commence par inclure les bibliothèques nécessaires, puis définit les broches utilisées pour la communication SPI. Ensuite, il initialise le capteur avec les bons paramètres de mesure :

- Suréchantillonnage de la température : 8x
- Suréchantillonnage de l'humidité : 2x
- Suréchantillonnage de la pression : 4x
- Filtre IIR pour stabiliser les lectures
- Activation du capteur de gaz Dans la boucle principale, le programme effectue une lecture complète du capteur toutes les deux secondes. Les valeurs mesurées sont affichées dans la console série, ce qui permet de vérifier que le capteur fonctionne correctement

Capteur 2 : SeeedStudio TMG39931



Proximité

Détection d'objets proches



Luminosité

Jusqu'à 60 kilolux



Colorimétrie

Composantes RGBC

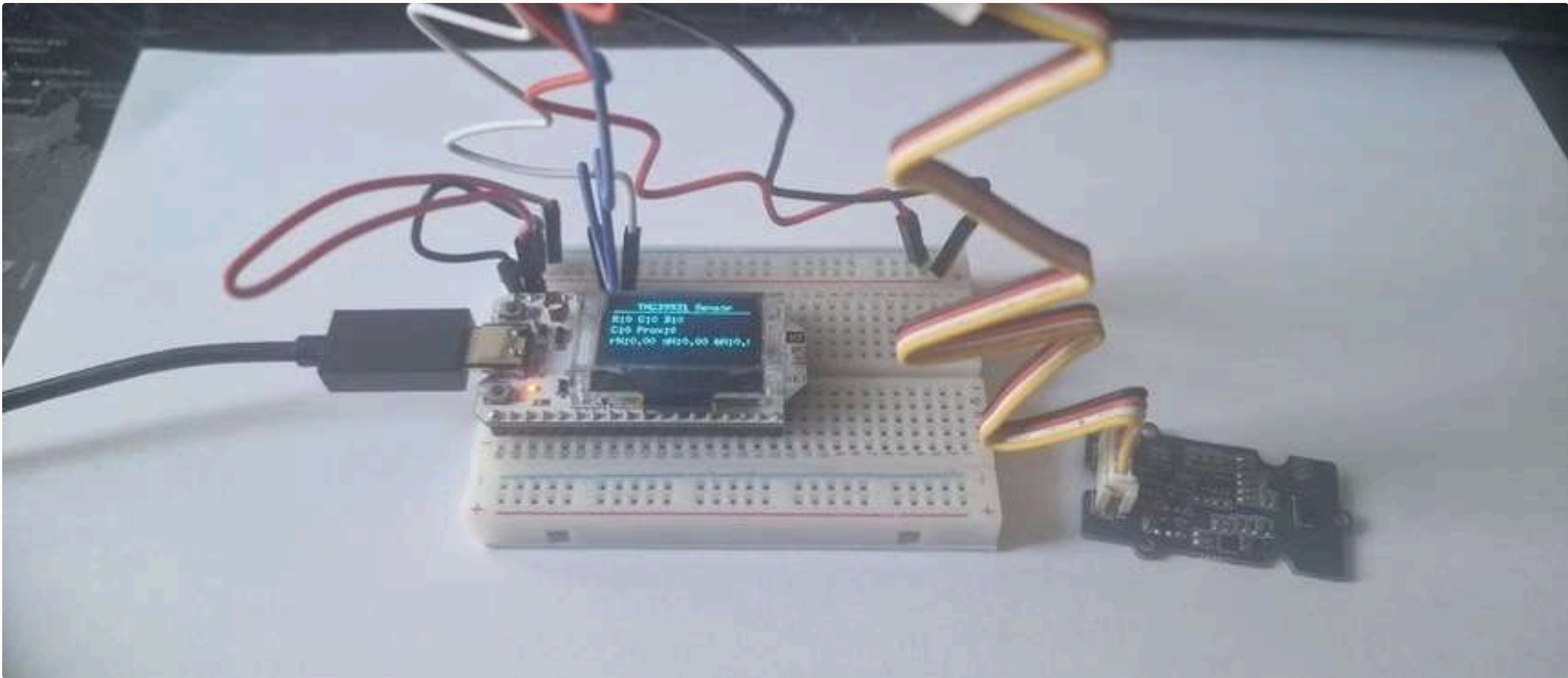


Communication

Protocole I2C



Test TMG39931



```
240 // Si le système n'est pas en pause, prendre des mesures régulièrement
241 (/ /pauseHeading)
242 {
243     readSensorData();
244 }
245
```

Serial Monitor

Serial (20000 baud) (WiringIO: 20000 baud) (115200 baud)

11:00:10.710 -> Vert: 0
11:00:10.710 -> Blanc: 0
11:00:10.710 -> Clear: 4
11:00:10.710 -> Flux lumineux (approximation): 0.00
11:00:10.710 -> Intervalle lumineux trop faible pour calculer les coordonnées chromatiques
11:00:10.710 ->
11:00:10.711 -> Prochaine mesure dans 10 secondes...
11:00:10.711 -> Appuyez sur 'g' pour mettre en pause!
11:00:21.000 ->
11:00:21.000 -> ----- Nouvelles mesures -----
11:00:21.000 -> Proximité: 2
11:00:21.000 -> Valeurs de couleur:
11:00:21.000 -> Rouge: 30
11:00:21.000 -> Vert: 20
11:00:21.000 -> Bleu: 24
11:00:21.000 -> Clear: 10
11:00:21.000 -> Flux lumineux (approximation): 34.34
11:00:21.000 -> Transposition chromatique (confocale)
11:00:21.000 -> Coordonnées chromatiques: (R:420, G:280, B:112)
11:00:21.000 -> Coordonnées CIE: x=0.129, y=0.112
11:00:21.000 ->
11:00:21.000 -> Prochaine mesure dans 10 secondes...
11:00:21.127 -> Appuyez sur 'g' pour mettre en pause!
11:00:30.323 ->
11:00:30.300 -> ----- Nouvelles mesures -----
11:00:30.300 -> Proximité: 0
11:00:30.300 -> Valeurs de couleur:
11:00:30.300 -> Rouge: 70
11:00:30.300 -> Vert: 52
11:00:30.300 -> Bleu: 54
11:00:30.300 -> Clear: 214
11:00:30.300 -> Flux lumineux (approximation): 39.83
11:00:30.300 -> Transposition chromatique (confocale)
11:00:30.300 -> Coordonnées chromatiques: (R:415, G:281, B:104)
11:00:30.300 -> Coordonnées CIE: x=0.117, y=0.112

Codes de test TMG39931

proximity.ino

Ce programme permet de mesurer la proximité d'un objet placé devant le capteur. Il effectue les opérations suivantes :

- Initialisation du capteur TMG39931 et de la communication I2C
- Configuration du capteur pour la détection de proximité
- Lecture et affichage en temps réel de la valeur de proximité sur le moniteur série.

color.ino

Ce programme permet de mesurer la luminosité et la colorimétrie de l'environnement. Il réalise les actions suivantes :

- Initialisation du capteur TMG39931
- Configuration pour l'acquisition des données de couleur et de lumière
- Lecture des valeurs des composantes Rouge, Vert, Bleu, Clair (RGBC)
- Calcul de la luminosité (en lux) et de la température de couleur (CCT)
- Affichage de ces informations sur le moniteur série.

Ces programmes de test ont permis de valider le bon fonctionnement du capteur de proximité et de luminosité

Capteur 3 : Iduino Heartbeat



Émission lumineuse

Traversant le doigt



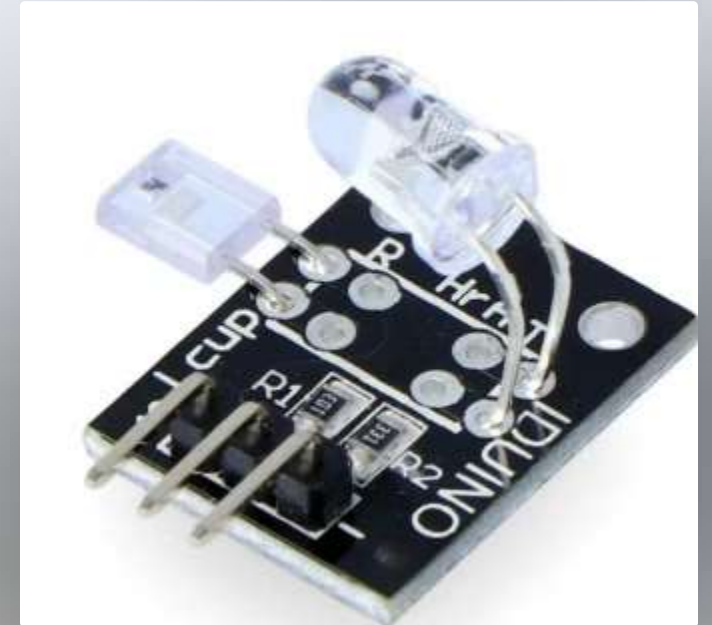
Absorption sanguine

Variation selon flux

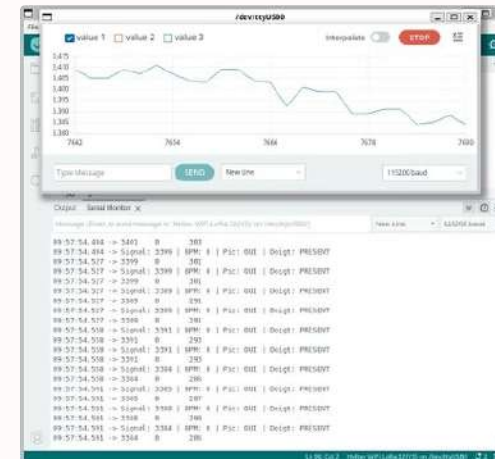
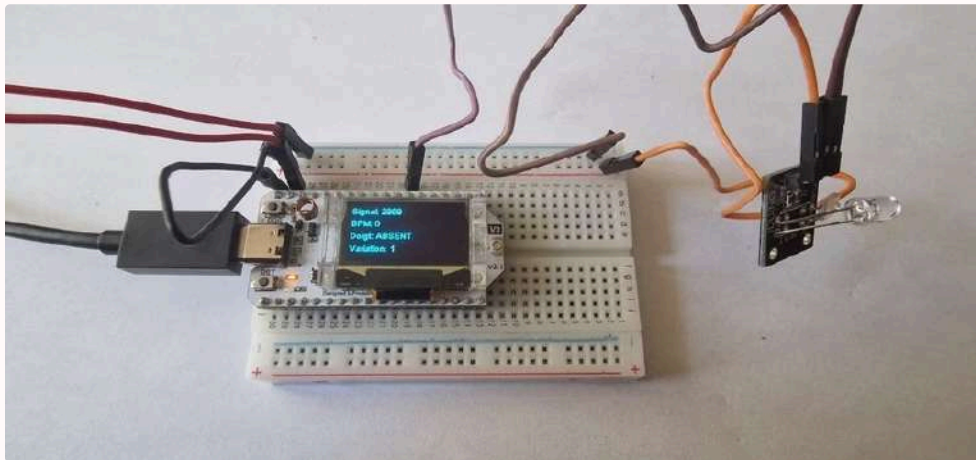


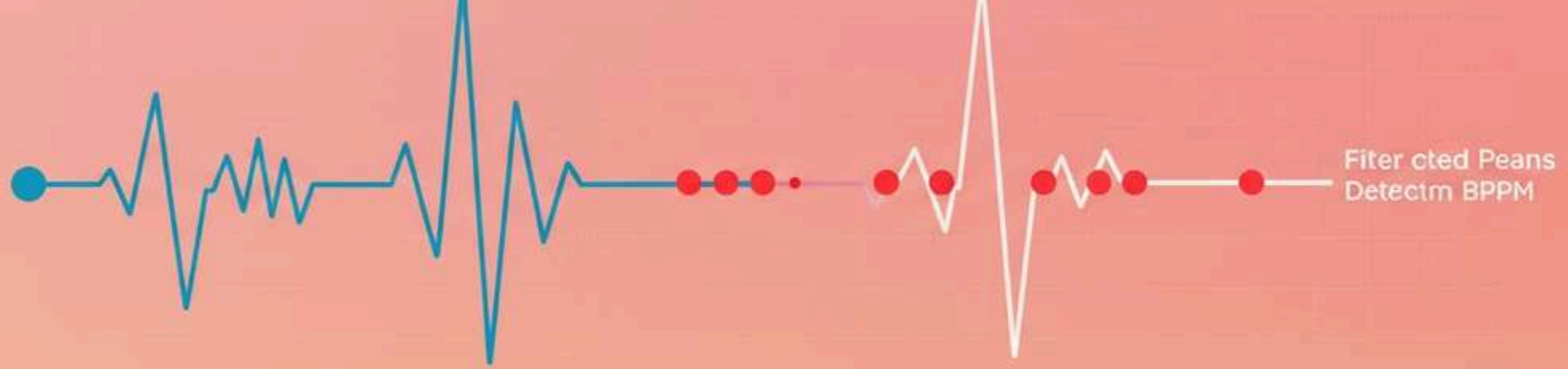
Détection pulsations

Calcul BPM



Test HeartBeat





Fonctionnement du code Heartbeat

heartbit.ino

Ce programme permet de :

- Lire les valeurs analogiques fournies par le capteur
- Filtrer et normaliser les données pour identifier les pics correspondant aux battements du cœur
- Calculer la moyenne des mesures pour réduire le bruit
- Compter le nombre de pics détectés et en déduire le BPM
- Afficher la valeur du BPM sur l'écran OLED de la carte Heltec et sur le moniteur série

L'algorithme fonctionne en analysant une série de 50 échantillons, en supprimant la moyenne pour centrer les données, puis en détectant les pics supérieurs à un seuil fixé à 0.95. Le BPM est recalculé toutes les 15 secondes

Affichage global des données



Environnement

Température, pression, humidité



Luminosité

RGBC et intensité lumineuse



Rythme cardiaque

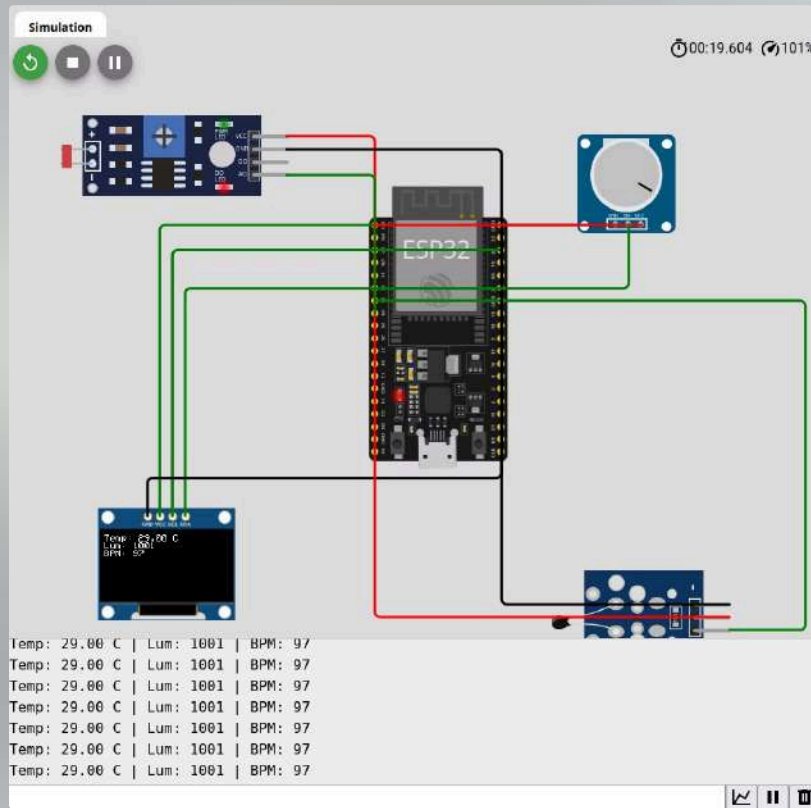
Affichage BPM



Simulation du projet

Validation du fonctionnement système avant déploiement matériel

Configuration de la simulation



1

ESP32

Microcontrôleur générique

1

OLED SSD1306

Écran simulé

3

Capteurs

NTC, luminosité, potentiomètre

Objectifs et limitations

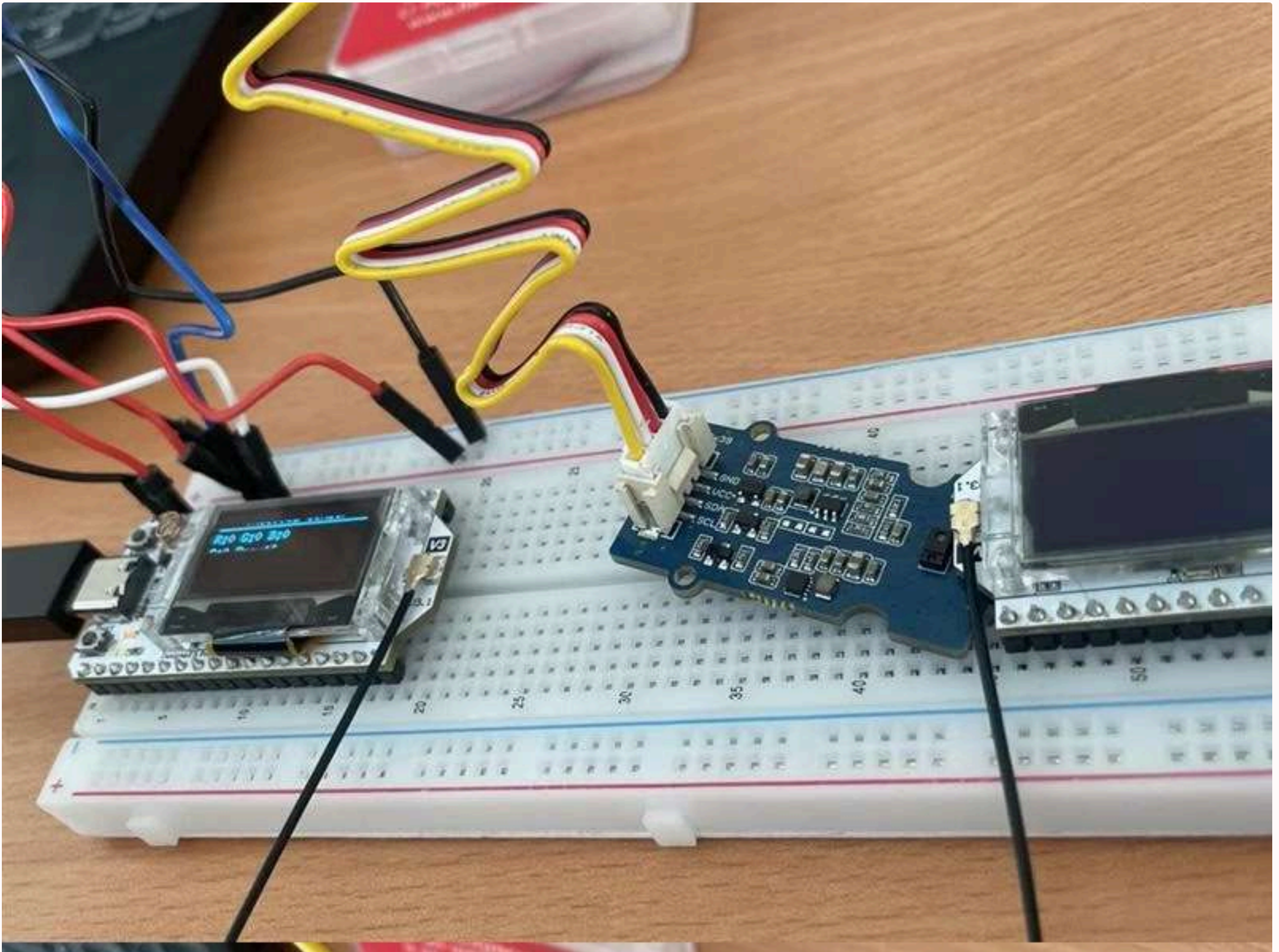
Objectifs

- Valider récupération données
- Tester affichage séquentiel
- Vérifier stabilité programme

Limitations

- Pas de communication LoRa
- Matériel Heltec non simulé
- Mesures approximatives

Affichage global des données



Fonctionnement du code d'affichage

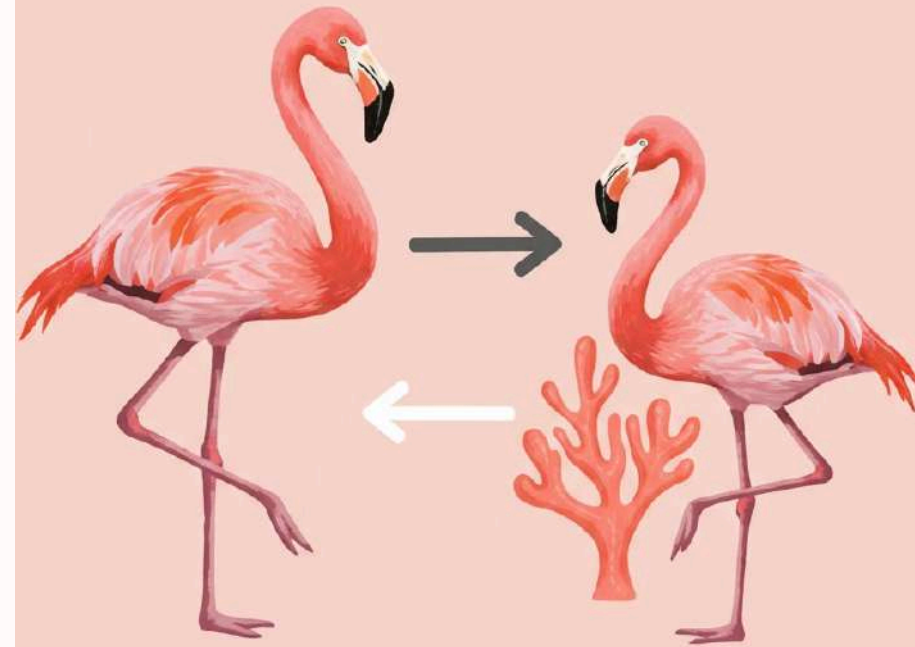
Le programme réalise les opérations suivantes :

- Lecture des données environnementales fournies par le capteur BME680 : température, pression, humidité et altitude
- Lecture des informations de luminosité et de colorimétrie (Rouge, Vert, Bleu, Clair) fournies par le capteur TMG39931
- Mesure du rythme cardiaque via le capteur Iduino Heartbeat
- Affichage de ces informations sur l'écran OLED intégré, avec un défilement automatique toutes les 4 secondes

Le programme utilise les bibliothèques spécifiques aux capteurs ainsi que la bibliothèque Heltec pour la gestion de l'affichage. Chaque ensemble de données est présenté de manière claire et lisible sur l'écran OLED. Ce code constitue l'aboutissement de l'intégration matérielle et logicielle du projet IoT, permettant de visualiser en temps réel les informations mesurées par l'ensemble des capteurs.

Communication LoRa: Émetteur et Récepteur

L'objectif de cette partie est de mettre en place un système de communication sans fil utilisant la technologie LoRa (Long Range) pour transmettre les données collectées par les capteurs connectés au microcontrôleur Heltec LoRa WiFi 32 (émetteur) vers un module ESP32 LoRa (récepteur). Les données reçues sont ensuite affichées en temps réel sur un serveur web local, accessible via un navigateur.



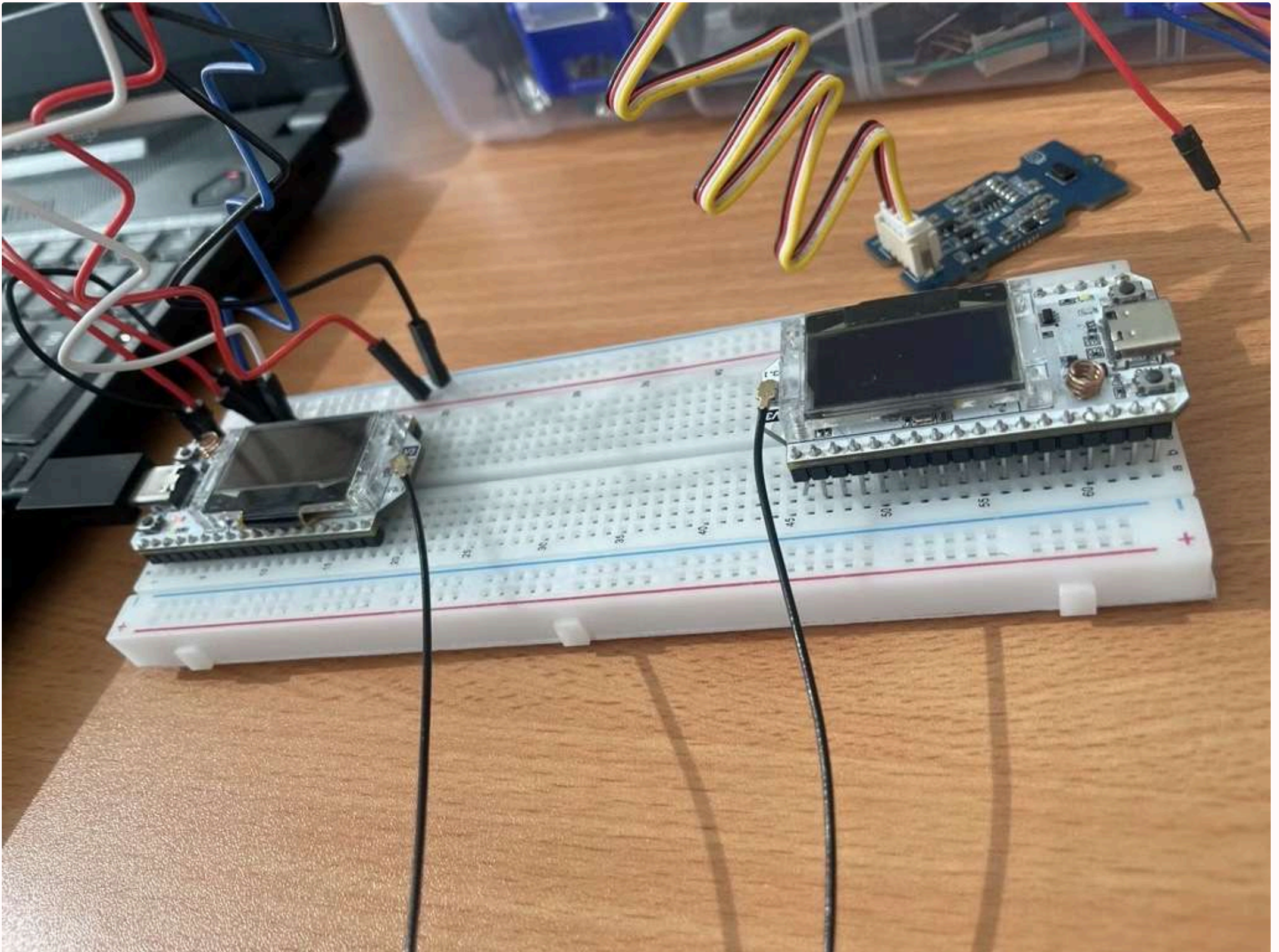
Architecture du système

Émetteur LoRa

- Capteurs connectés
- Écran OLED intégré
- Mode Deep Sleep

Récepteur LoRa

- ESP32 + module LoRa
- Réception données
- Serveur web local



Fonctionnement



Acquisition données

Lecture périodique capteurs



Affichage local

Écran OLED temporaire



Transmission LoRa

Envoi radio longue portée



Affichage web

Serveur local temps réel



Deep Sleep

Économie d'énergie

Code source

sender_lora_monitoring.ino

- Gestion capteurs
- Affichage OLED
- Transmission LoRa
- Deep Sleep

receiver_lora_webserver.ino

- Réception paquets LoRa
- Serveur web
- WebSocket temps réel



Avantages et limites

Avantages

- Longue portée
- Faible consommation
- Temps réel via WebSocket
- Mode Deep Sleep économe

Limites

- Dépendance WiFi
- Portée réduite en milieu urbain

Gestion de l'énergie: Deep Sleep

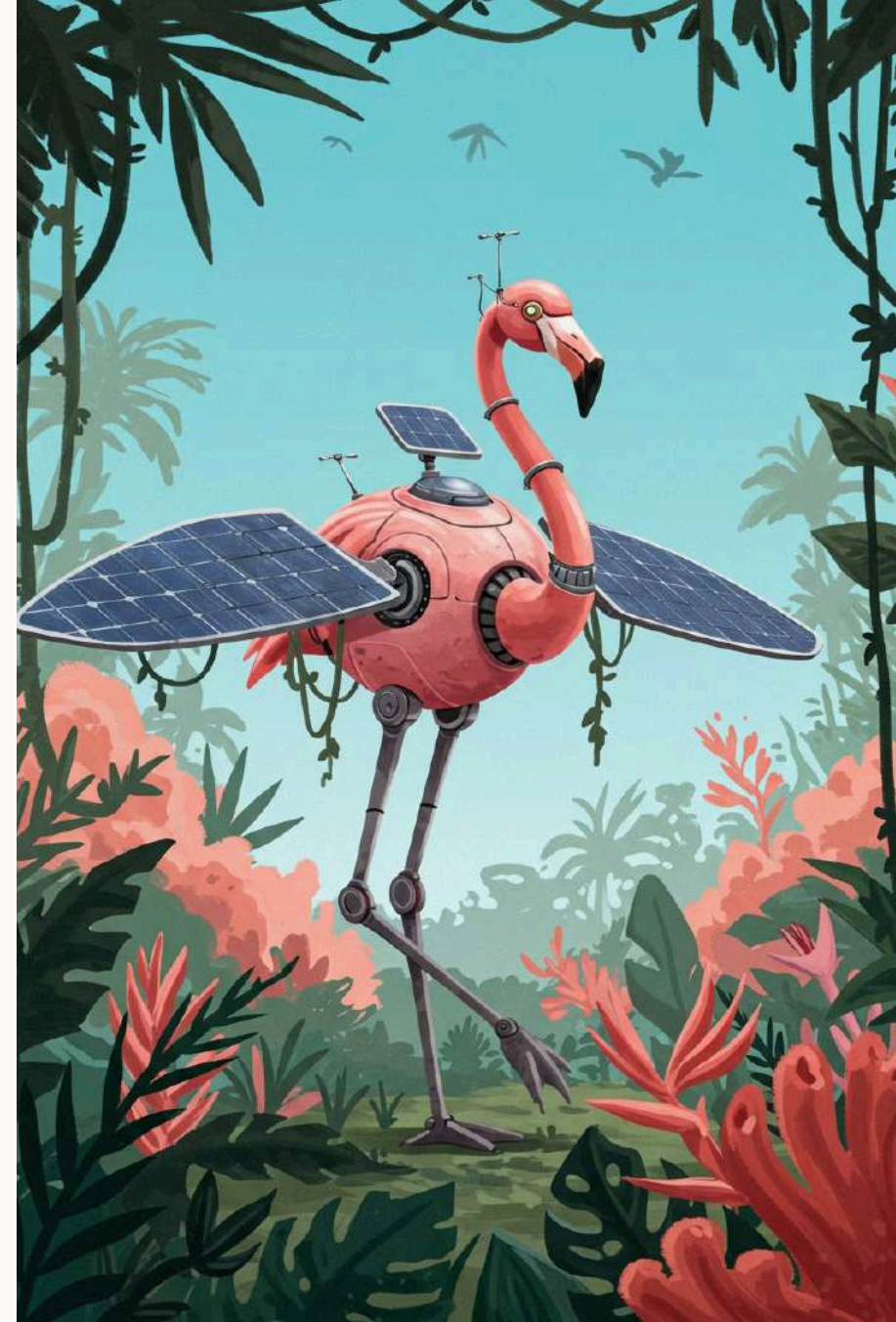
Pour optimiser l'autonomie énergétique du système, la fonctionnalité Deep Sleep de l'ESP32 a été implémentée sur l'émetteur. Après chaque cycle de mesure et de transmission, le microcontrôleur entre en veille profonde pendant une durée de 10 secondes, paramétrable. Ce mode réduit la consommation énergétique en désactivant les modules inutilisés, tout en permettant un réveil automatique grâce à un minuteur matériel (`esp_sleep_enable_timer_wakeup`). Cette approche est particulièrement adaptée aux applications IoT fonctionnant sur batterie



Projet IOT

Flaming : Système de surveillance
environnementale intelligent avec communication
LoRa

Créez un système de surveillance environnementale qui utilise le capteur
TMG39931 (capteur de lumière ambiante, RGB et de proximité/geste)
pour collecter des données, puis les transmet via LoRa entre les deux
modules Heltec



Architecture du système

Module capteur

Premier Heltec connecté au capteur TMG39931.

- Collecte données de lumière ambiante
- Mesure couleur RGB et proximité
- Transmission LoRa longue portée
- Alimentation par batterie
- Déploiement flexible

Module récepteur/passerelle

Second Heltec servant d'interface réseau.

- Réception des données LoRa
- Connexion Wi-Fi au cloud
- Affichage temps réel sur OLED
- Alimentation secteur possible
- Traitement local des données

Applications innovantes

- **Surveillance agricole intelligente:** Mesurez la lumière et les conditions environnementales dans les serres ou les champs
- **Optimisation d'éclairage urbain:** Réglage automatique de l'éclairage public en fonction de la lumière ambiante réelle
- **Détection de présence écologique:** Utilisation de la détection de proximité pour activer/désactiver des systèmes uniquement quand nécessaire



Caractéristiques techniques



Communication longue distance

La technologie LoRa permet une transmission de données sur plusieurs kilomètres, idéale pour les déploiements ruraux.



Faible consommation énergétique

Les modules fonctionnent sur batterie pendant des semaines grâce au mode Deep Sleep optimisé.



Capteur multi-fonctionnel

Le TMG39931 mesure la lumière ambiante, les couleurs RGB et détecte la proximité dans un seul composant.



Interface visuelle intégrée

L'écran OLED affiche les données en temps réel sans nécessiter d'équipement supplémentaire.



Conclusion



Système complet

Surveillance
environnementale IoT



Compétences développées

Programmation,
communication sans fil



Perspectives

Interface web, alertes, multi-émetteurs

