

# Solution Technologique - Choix du Microcontrôleur

## Collecteur de Données Animalières

21 Octobre 2025

## 1 Introduction

Ce document présente l'analyse et la justification du choix du microcontrôleur retenu pour le projet de collecteur de données animalières. Nous détaillons les critères de sélection et comparons deux solutions plausibles : l'ESP32 et le Raspberry Pi Zero.

## 2 Historique

À la suite d'une phase de recherche concernant le contrôleur à utiliser pour ce projet, nous avons identifié deux solutions susceptibles de répondre à nos besoins : l'ESP32 et le Raspberry Pi Zero.

Pour rappel, les besoins principaux concernant le contrôleur étaient les suivants :

Critère	Description
Consommation énergétique	Minimiser la consommation pour un usage sur batterie.
Performances	Garantir une puissance suffisante pour le traitement des données.
Stockage	Disposer d'une capacité suffisante pour enregistrer temporairement les mesures locales et la photographie.
Système	Offrir un environnement logiciel simple avec facilité sur les outils de développement utilisés.
Interfaces	Proposer des entrées/sorties adaptées aux capteurs et modules externes.

<b>Connectivités</b>	Intégrer ou supporter le Wi-Fi, le Bluetooth ou d'autres moyens de communication.
<b>Prix</b>	Rester dans une gamme de coût compatible avec le budget du projet.

Ces critères nous ont permis de comparer plusieurs solutions techniques lors de la phase de benchmark.

### 3 Critères de sélection

Les critères suivants ont été considérés lors du choix final du microcontrôleur :

- Architecture matérielle
- Connectivité
- Consommation énergétique
- Nombre et type de GPIO
- Facilité de développement
- Coût global

#### 3.1 Architecture matérielle

Caractéristique	ESP32 Dev Module	Raspberry Pi Zero (W / 2 W)
<b>Processeur</b>	Dual-core Tensilica Xtensa LX6 à 240 MHz	Broadcom BCM2835 (Pi Zero) ou BCM2710A1 (Pi Zero 2 W), ARM Cortex-A53 quad-core 1 GHz
<b>Architecture</b>	32 bits (microcontrôleur)	64 bits (microprocesseur complet)
<b>Mémoire RAM</b>	520 KB SRAM interne	512 MB LPDDR2 SDRAM
<b>Mémoire Flash / Stockage</b>	4 à 16 MB (selon le module)	Dépend de la carte microSD (jusqu'à 256 GB ou plus)

<b>Système d'exploitation</b>	Aucun (bare metal), FreeR-TOS ou MicroPython	Linux (Raspberry Pi OS, Ubuntu, etc.)
-------------------------------	--	---------------------------------------

*Commentaire :* L'ESP32 offre une architecture légère idéale pour les tâches embarquées à faible consommation, tandis que le Raspberry Pi Zero dispose d'une puissance, d'une mémoire et d'une capacité de stockage nettement supérieures, mais au prix d'une complexité, d'une taille et d'une consommation accrues.

## 3.2 Connectivité

Type	ESP32	Raspberry Pi Zero (W / 2 W)
<b>Wi-Fi</b>	Oui (2.4 GHz, 802.11 b/g/n)	Oui (Wi-Fi intégré sur version W et 2 W)
<b>Bluetooth</b>	Oui (v4.2 + BLE)	Oui (v4.1 sur Pi Zero W / BLE sur Pi Zero 2 W)
<b>Ethernet</b>	Non	Non (possible via adaptateur USB)
<b>USB</b>	Micro-USB (programmation et alimentation)	Micro-USB OTG (périphériques) + Mini-HDMI vidéo

*Commentaire :* Les deux plateformes offrent une bonne connectivité sans fil, mais le Raspberry Pi Zero se distingue par sa polyvalence en périphériques tandis que l'ESP32 reste plus simple et compact.

## 3.3 Consommation énergétique

Situation	ESP32	Raspberry Pi Zero (W / 2 W)
<b>Repos (Deep Sleep)</b>	Environ 10 $\mu$ A à 150 $\mu$ A	100 mA
<b>Actif</b>	100 – 250 mA	120–200 mA (Pi Zero) / 300–400 mA (Pi Zero 2 W)

*Commentaire* : L'ESP32 consomme beaucoup moins d'énergie, ce qui en fait une solution idéale pour un système sur batterie, contrairement au Raspberry Pi Zero plus énergivore.

### 3.4 Nombre et type de GPIO

Type	ESP32	Raspberry Pi Zero (W / 2 W)
<b>GPIO disponibles</b>	= 30 broches (3.3 V)	40 broches (3.3 V)
<b>Fonctions</b>	PWM, ADC, I2C, SPI, UART, Touch Sensor	PWM, I2C, SPI, UART, GPIO numériques

*Commentaire* : Le Raspberry Pi Zero offre plus de broches, mais l'ESP32 propose une plus grande variété de fonctions intégrées comme l'ADC et les capteurs tactiles.

### 3.5 Facilité de développement

Aspect	ESP32	Raspberry Pi Zero (W / 2 W)
<b>Langages supportés</b>	C / C++ (Arduino, ESP-IDF), MicroPython, Lua	Python, C/C++, Java, Node.js, etc.
<b>Outils de développement</b>	Arduino IDE, PlatformIO, Espressif IDF	Raspberry Pi OS, VS Code, SSH, Thonny, etc.
<b>Programmation directe</b>	Via USB (UART/Serial)	Via carte microSD ou connexion SSH
<b>Communauté et ressources</b>	Très étendue (Arduino, Espressif)	Très étendue (Raspberry Pi Foundation)

*Commentaire* : L'ESP32 est plus simple à programmer pour des applications embarquées, tandis que le Raspberry Pi Zero offre un environnement Linux complet adapté à des projets plus complexes.

### 3.6 Coût global

Produit	Prix estimé (€)	Remarques
ESP32 Dev Module	4 – 20 €	Très bon rapport performance / prix, facilement disponible.
Raspberry Pi Zero W	15 – 25 €	Bon compromis puissance / prix, mais disponibilité variable.
Raspberry Pi Zero 2 W	20 – 35 €	Plus performant mais plus énergivore.

*Commentaire :* L'ESP32 est nettement plus économique, tandis que le Raspberry Pi Zero coûte plus cher et peut être difficile à se procurer selon les périodes de disponibilité.

## 4 Technologies retenues

À l'issue de l'analyse comparative réalisée entre l'ESP32 et le Raspberry Pi Zero, le choix s'est porté sur l'**ESP32** comme microcontrôleur principal pour le collecteur de données animales.

Ce choix est motivé par plusieurs facteurs clés. Tout d'abord, sa **faible consommation énergétique** constitue un avantage décisif pour une utilisation sur batterie dans un environnement isolé. L'ESP32 peut fonctionner plusieurs jours, voire semaines, en mode basse consommation, ce qui garantit une autonomie optimale sur le terrain.

Ensuite, son **intégration native du Wi-Fi et du Bluetooth** simplifie la conception matérielle et réduit le besoin de modules externes, tout en permettant la transmission sans fil des données et la configuration à distance du système.

Sur le plan du développement, l'ESP32 bénéficie d'un **écosystème logiciel très complet** (Arduino, ESP-IDF, MicroPython) et d'une large communauté d'utilisateurs, facilitant la mise en œuvre rapide du prototype et la maintenance future du système.

Concernant le **stockage**, la mémoire interne de l'ESP32 reste limitée (quelques mégaoctets selon le module), ce qui peut poser problème lors de l'enregistrement d'images complètes issues d'un capteur photo. Dans ce cas, le microcontrôleur pourrait ne pas disposer de suffisamment d'espace pour stocker temporairement ces fichiers. Une solution d'amélioration consiste

à **ajouter une carte microSD connectée via l'interface SPI**, permettant d'augmenter considérablement la capacité de stockage et de conserver localement les données avant leur transmission.

Enfin, son **coût réduit et sa disponibilité** sur le marché en font une solution économique et accessible, sans compromis majeur sur les performances nécessaires au projet.

En revanche, il est à noter que le Raspberry Pi Zero offre une puissance de calcul et une mémoire supérieures, mais celles-ci sont jugées excessives au regard des besoins du projet et pénalisantes en termes de consommation énergétique et de gestion logicielle.

**En conclusion**, l'ESP32 répond pleinement aux exigences du projet en matière de consommation, de connectivité et de simplicité de déploiement, ce qui en fait la solution la plus adaptée pour le collecteur de données animalières.