

# Pierre Ollivier

Paris, France | pollivie@student.42.fr | 07 85 61 33 47 | linkedin.com/in/pollivie | github.com/pierrelgol

## Sommaire

---

Étudiant Ecole 42 - Paris - RNCP 7.

Spécialisé en C/C++/Zig/ AVR, ESP32 et Linux. Passionné par la conception de code robuste et temps réel, particulièrement dans le secteur de la défense.

## Compétences

---

**Outils:** C, C++, Zig, Bash, Rust, Python, Java, Git, Qt, SDL, SQL, Pytorch, Numpy

**Plateformes:** Linux, Windows, AVR, ESP32, Raspberry Pi, RTOS (FreeRTOS, Zephyr), Docker

**Langues:** Français (C2), Anglais (C2), Espagnole (B1), Russe (A1)

## Expérience

---

**Ingénieur Logiciel Embarqué (Alternance)**, Thales – France

Sep 2025 – présent

- Refonte du code legacy C++ vers C pur, supprimant Libcamera/OpenCV pour faciliter la cross-compilation et réduire le temps de build (6m → 20s).
- Livraison d'un kernel Linux sur-mesure avec drivers propriétaires, réduisant le démarrage (22s → 3s) et la consommation (450mA → 280mA).
- Architecture d'un pipeline V4L2/DMA zéro-copie optimisé via intrinsèques NEON, atteignant un temps réel souple avec une empreinte mémoire minime (200Mo → 6Mo).
- Conception d'un pipeline MLOps complet pour YOLO11 : génération de données synthétiques, entraînement curriculum et export TensorRT.

**Militaire**, Armée de Terre - 13ème DBLE – France

2017 – 2018

## Projets

---

**Compilateur C11 en Zig**

2025

- Développement from scratch d'un compilateur C11 complet en Zig.
- Robuste et capable de compiler des projets réels d'envergure comme MySQL.

**ZigCleaner (zc)**

2025

- Utilitaire de nettoyage de fichiers haute performance développé en Zig.
- Scanne 200 000+ répertoires en < 1s via un pool de threads optimisé.
- Suppression récursive d'artefacts de build pour libérer de l'espace disque.

**Développement embarqué sur ATMega328p**

2025

- Implémentation de drivers bas-niveau pour I2C, SPI et UART avec manipulation directe des registres.
- Gestion des interruptions matérielles et développement de pilotes pour divers capteurs.

**Développement Netcode Multithreaded**

2024

- Simulation Lock-Step de pointe et architecture Thread-per-game en Zig.
- Création d'un format de sérialisation binaire sur-mesure minimisant la latence client (< 1 ms).
- Optimisation de l'empreinte mémoire (< 512 Kb RAM).

## Éducation

---

**42 Paris**, in Spécialisation bas-niveau / embarqué / Kernel – Paris, France

2023 – 2025

**UBS Vannes**, in Licence Économie appliquée – Vannes, France

2020 – 2021