

Tom Schammo

 Tom Schammo |  Tom Schammo |  tom@tomschammo.com |  +352 621 48 0040

AUSBILDUNG

2019 - 2024 B.Sc. (Informatik) an der **Universität Tübingen**

(Note: 2,36)

Kurse umfassen:

- Programming Ultra Low Power Architectures: Ein M.Sc. Kurs, der Studierenden Stromspaarzustände, Hardware-Treiber, Busse und Hardware-Interrupts näher bringt.
- Massively Parallel Computing: Ein M.Sc. Kurs, in dem die Programmierung von GPUs mittels CUDA gelehrt wird, um diverse Probleme effizienter zu lösen, wie z.B. Matrixmultiplikation oder die Manipulation von Datenstrukturen.
- Teamprojekt: Ein Kurs, in dem Studierende lernen ein Software-Projekt im Team zu planen und umzusetzen. Insbesondere wird die Versionskontrolle mittels git erlernt.
- Abschlussarbeit: Das Ziel meiner Abschlussarbeit war die (Audio-)Schlüsselworterkennung auf eingebetteten Geräten zu verbessern.

BERUFSERFAHRUNG

Universität Tübingen

Nov 2023 - Feb 2025

Wissenschaftlicher Mitarbeiter

Im Rahmen eines Forschungsprojekts zu selbstfahrenden Autos habe ich die vollständige Entwicklung einer [Lidar-Punktwolken-Augmentierungsbibliothek](#) übernommen. Diese stellt verschiedene SOTA-Augmentierungsmethoden durch ein einfach zu bedienendes Python-Frontend bereit, während die Leistung aufgrund des effizienten C++-Backends erheblich verbessert wird.

Als alleiniger Entwickler war ich für die konzeptionelle Planung, Programmierung und Testung der Bibliothek verantwortlich, wodurch ich meine Fähigkeiten entwickelte, mittelgroße Projekte eigenständig von Anfang bis Ende durchzuführen.

Octoshrew Ltd.

Nov 2022 - Nov 2023

Junior Entwickler

Als Junior Entwickler war ich an der Entwicklung modernster Pfadplanungs- und autonomen Steuerungssysteme für Drohnen beteiligt. Insbesondere war ich für die Pfadfindung verantwortlich und entwickelte die Datenstrukturen und Algorithmen, um eine sichere und effiziente Navigation im 3D-Raum zu gewährleisten, wodurch ich meine Fähigkeiten in Algorithmen und Leistungsoptimierung/Parallelisierung verbesserte.

PROJEKTE

Implementierung und Analyse verschiedener Eingabequellen auf der Ultratrail-Architektur

Für meine Abschlussarbeit arbeitete ich an der Implementierung eines Rust-Treibers für den Ultratrail AI-Beschleuniger auf dem PULPissimo-Board (ein Entwicklungsboard für einen von der ETH Zürich entwickelten RISC-V-Chip), um eine effiziente Schlüsselworterkennung in Audio zu ermöglichen.

Durch die Verbesserung der Sicherheit mittels der Programmiersprache Rust gegenüber des bestehenden C-Treibers verbesserte ich meine Fähigkeiten und erweiterte mein Wissen in der eingebetteten Programmierung sowie in der Fehlersuche auf Hardwareebene mithilfe von Werkzeugen wie Oszilloskopen oder Logikanalysatoren.

Programming Ultra Low Power Architectures

Ich baute einen 'Wecker' also proof-of-concept mit einem Entwicklungsboard und Peripheriegeräten (LED-Display, Tastatur, Lautsprecher). Der Wecker nutzte Tiefschlafzustände, um sehr energieeffizient zu sein und überlebte 12 Stunden allein auf einem kleinen Kondensator. Zum Vergleich: Mit einer AA-Batterie würde der Wecker 130'000 bis 220'000 Stunden im Energiesparmodus halten.

Das Projekt bot mir die Möglichkeit, Einblicke in die Nutzung von Hardwaredokumentationen sowie in die Entwicklung und Fehlersuche von Hardwaretreibern und eingebetteten Anwendungen zu gewinnen. Zusätzlich bot es die Gelegenheit, die Nutzung von Prozessor-Schlafzuständen und Werkzeugen zur Messung des Stromverbrauchs zu erkunden, was zu verbesserten batteriebetriebenen Designs führte.

FÄHIGKEITEN

Sprachen:	Deutsch (Muttersprache), Luxemburgisch (Muttersprache), Englisch (C2), Französisch (B2)
Programmiersprachen:	C, C++, CUDA, Python, Rust, RISC-V Assembler, Mips Assembler
Entwicklungstools:	Make, CMake, Git, GitHub, GitLab, Linux, Docker, Logic Analyzer, Oszilloskop
Frameworks:	Libtorch, Boost, Google Test, ROS, PyTest, Numpy