

Konstantinos Charatzas

+44 7591 072603 | Konstantinos.Charatzas@gmail.com
github.com/PseudoApoidean | linkedin.com/in/konstantinos-charatzas

SUMMARY

Embedded Software Engineer with 3 years of commercial experience delivering C firmware for connected, resource-constrained systems shipped at scale. Background spans low-level driver bring-up, protocol implementation, RTOS application code and embedded GUI in C/C++, with a consistent focus on reliability and testability.

WORK EXPERIENCE

Ohme EV, London | Apr 2023 - Sep 2025

Embedded Software Engineer

C/C++ FreeRTOS firmware for STM32 (ARM Cortex-M4) EV chargers deployed in 100k+ units across the UK/EU.

- Rewrote the device-side custom OCPP 1.6J C library's message generation, transmission and storage pipeline, reducing message loss by 95% and eliminating message mis-ordering, an essential contributor to OCPP standards compliance.
- Proposed, implemented and shipped an interrupted session detection and recovery mechanism previously rejected as too flash-write-heavy, made feasible by a new flash-optimised message storage pipeline.
- Developed features for a legacy TouchGFX (C++) GUI: new screens, implementation and polish for additional language translations, and rebuilds of existing screens. Later collaborated on architecture design and personally built 20+ screens from scratch in LVGL for new hardware, following TDD with CppUTest unit tests (C++).
- Debugged driver code for a new metering chip carrying unvetted configuration changes from the manufacturer's development branch. Via step-through debugging, byte-level decoding and datasheet analysis of SPI communication, ruled out wiring and timing issues to isolate and fix the problem, unblocking development of new hardware.
- Streamlined release workflows with lightweight automation (Bash and Python scripts, GitHub Actions), reducing manual effort.

PROJECTS

AEMlib - MQTT 3.1.1 Client Library in C99 - github.com/PseudoApoidean/aemlib

- Implemented an MQTT 3.1.1 client (QoS 0) in C99, targeted at bare-metal and RTOS systems (zero dynamic allocation, caller-owned TX/RX buffers, cooperative non-blocking `poll()`).
- Abstracted transport and time behind function-pointer interfaces, allowing the same client to run over POSIX TCP, an in-process virtual transport, or test fakes without modification or conditional compilation.
- Built a Unity test suite covering protocol encode/decode, client state machine, partial/coalesced TCP reads and keepalive timing, wired into CTest and run on every push to main via GitHub Actions.
- Validated end to end against a real broker (test.mosquitto.org) over a POSIX TCP transport, and hermetically in CI against a fake broker.

ESP32 BLE/GATT Debugging Tool - github.com/PseudoApoidean/esp32_gatt_debug

- Firmware in C on ESP-IDF that emulates a generic iTag-style BLE peripheral to aid debugging of BLE central implementations.
- Surfaces GAP events and GATT service/characteristic accesses in real time, exposing connection behaviour that is otherwise opaque from the central side.

iTag Mobile - Cross-Platform BLE Client - github.com/PseudoApoidean/itag-mobile

- Flutter/Dart application for Android and iOS that scans for, connects to and controls generic BLE tracker devices.
- Reads device state (RSSI, battery level) and writes control characteristics (alert level), exercising the same GATT profiles as the ESP32 tool above from the central side.

SKILLS

Programming Languages: C, C++, Python, Bash, VHDL

Embedded: STM32, ARM Cortex-M, FreeRTOS, ESP32 / ESP-IDF, BLE (GAP/GATT), MQTT, OCPP, device drivers, datasheet-level debugging, hardware integration, soldering

GUI: LVGL, TouchGFX, TouchGFX Designer

Build & Test: CMake, Unity, CppUTest, pytest, GitHub Actions, CI/CD, Git, GDB, IAR

Spoken Languages: English, Greek

EDUCATION

Edinburgh Napier University | 2018 - 2022

BEng (Honours) *Electronic & Electrical Engineering*