Owuraku Dompreh

kuku.dompreh@warwick.ac.uk | +44 7496 894391 | LinkedIn | GitHub | SC-Eligible (UK National)

FPGA RTL Design | Bare-Metal ARM Firmware | Quantum Algorithm Research

Education

University of Warwick — Coventry, England
MEng in Computer Systems Engineering — *Sep. 2025 – Present*

- **Relevant Modules:** Computer Organisation & Architecture (72%), Electrical & Electronic Circuits, Systems Modelling & Simulation
- **Awards:** PATHWAY Undergraduate Research Scholarship (URSS)

Research & Experience

Quantum Systems Research Intern — Summer 2026
Warwick Manufacturing Group (WMG), URSS-Funded — *Coventry, England*

- Simulated Parametrised Quantum Circuits (**PQCs**) and the Quantum Approximate Optimisation Algorithm (**QAOA**) in Python/Qiskit under URSS funding to model combinatorial optimisation for financial fraud detection, achieving a Mean Squared Error (**MSE**) convergence of **1.8e-3**.
- Architected a hybrid hardware-software integration framework mapping data transfer pathways between classical infrastructure and QAOA instances, explicitly optimising gate depth to meet the physical limits of Noisy Intermediate-Scale Quantum (**NISQ**) hardware.
- Profiled computational resource scaling against classical solvers, quantifying system-level trade-offs in gate error rates and projecting an **18%** reduction in total execution latency.

Engineering Portfolio

Tang Nano 9K FPGA UART Pipeline | *Verilog, Yosys, openFPGALoader, GTKWave, Linux CLI*

- Engineered a complete Universal Asynchronous Receiver-Transmitter (**UART**) pipeline from first principles on a **9K LUT** Field-Programmable Gate Array (**FPGA**), designing a custom Verilog clock divider using mathematical counters to derive **115,200 baud** from a **27MHz** crystal oscillator with no vendor IP cores.
- Designed a 4-state Finite State Machine (**FSM**) (IDLE, START, DATA, STOP) to govern bit transmission at Register-Transfer Level (**RTL**), verifying race-condition elimination in **GTKWave** waveform traces prior to bitstream deployment via openFPGALoader.
- Maintained synthesis workstation on Ubuntu 24.04 LTS, managing Yosys and openFPGALoader toolchains from terminal with `.gitignore` discipline preventing compiled bitstream uploads to a documented GitHub portfolio.

STM32 Nucleo Bare-Metal Data Logger | *C, GCC ARM Toolchain, Python, pySerial, Matplotlib*

- Programmed the STM32F401RE ARM Cortex-M4 without the STM Hardware Abstraction Layer (**HAL**), decoding a 1,000-page reference manual to locate **AHB** bus addresses and configure RCC_AHB1ENR and GPIO registers via bitwise OR/AND-NOT operations, replacing **40KB** HAL overhead with a **3KB** bare-metal driver.
- Configured hardware timers via the Nested Vectored Interrupt Controller (**NVIC**) to trigger an Interrupt Service Routine (**ISR**) every **100ms**, driving non-blocking Analog-to-Digital Converter (**ADC**) sampling and streaming raw bytes to a threaded Python OOP class that rendered a live Matplotlib telemetry dashboard.

Technical Skills

Hardware Description: Verilog (RTL synthesis, FSM design, clock dividers, testbench verification), GTKWave
Embedded Systems: Bare-Metal C, ARM Cortex-M4, Memory-Mapped I/O, ISR/NVIC, UART, ADC, Hardware Timers
Languages: C, C++, Python (OOP, pySerial, Matplotlib, Qiskit, pandas), MATLAB, Bash, Assembly
EDA & Simulation: Yosys, openFPGALoader, NI Multisim, NI LabVIEW (installed)
Infrastructure: Git (Linux CLI), GitHub (documented portfolio), Ubuntu 24.04 LTS, Linux terminal