ORIGIN ACADEMY · COURSE SYLLABUS

ARM Developer

Go deep on the ARM architecture: assembly, exceptions, the MMU, caches, TrustZone and performance on Cortex-M and Cortex-A.

20 weeks · about 12 hours a week · live cohort of up to 10 · hardware kit included
Curriculum last updated 18 November 2026

Availability

This course opens soon. Join the waitlist on the course page to hear first about dates and prices.

Before you start

- C: pointers, structs and bit manipulation (our C Programming course covers this)
- Some embedded experience, or our Embedded Systems Engineer course
- English good enough to follow technical documentation and live sessions: about a credit (C6 or better) in WASSCE or NECO English Language, or an equivalent such as KCSE or IGCSE English (roughly CEFR level B1). You don't need the certificate: Phase 0 is free, so you can check for yourself
- A laptop running Linux, macOS, or Windows with WSL

How the course works

- A live cohort of up to 10 students, with a mentor.
- At least half your time is hands-on: every module ends with a lab.
- Live sessions are recorded, so you can rewatch them or catch up.
- Questions between sessions are answered within 24 hours on weekdays.
- A mentor reviews every phase project, like a pull request at work.

Phase 0 is free for everyone, before you enroll.

Curriculum

PHASE 0 · WEEKS 1–2 · FREE

ARM Architecture Essentials

A-, R- and M-profiles, registers, instruction sets and the programmer's model.

- The ARM family: A, R and M profiles
- Registers and the programmer's model
- Instruction sets: A64, A32 and Thumb
- Reading a technical reference manual

You build: Annotated memory map of a real SoC

PHASE 1 · WEEKS 3–6

ARM Assembly

Thumb-2 and AArch64 assembly, the AAPCS calling convention, mixing C and assembly.

- Thumb-2 assembly basics
- AArch64 assembly basics
- The AAPCS calling convention
- Calling assembly from C and back
- Reading compiler output

You build: Optimised assembly routines called from C

PHASE 2 · WEEKS 7–9

Exceptions & Privilege

Exception levels, vector tables, interrupts, SVC calls and fault handling.

- Exception levels and privilege
- Vector tables
- Interrupt controllers: NVIC and GIC
- Supervisor calls
- Fault handling and crash dumps

You build: Fault handler that reports crash context

PHASE 3 · WEEKS 10–12

Memory System

MMU and MPU, page tables, caches, barriers and memory ordering.

- The MPU on Cortex-M
- The MMU and translation tables
- Caches and cache maintenance
- Barriers and memory ordering

You build: Enable the MMU and caches on bare metal

PHASE 4 · WEEKS 13–14

Security: TrustZone

Secure and non-secure worlds, secure boot and trusted firmware.

- TrustZone for Cortex-A and Cortex-M
- Secure boot chains
- Trusted Firmware-A and -M

You build: Secure/non-secure partitioned application

PHASE 5 · WEEKS 15–16

Toolchains & Debug

GCC and LLVM for ARM, linker scripts, CoreSight, OpenOCD and tracing.

- GCC and LLVM for ARM
- Linker scripts and startup files
- OpenOCD and GDB

- CoreSight trace

You build: Reproducible build and debug setup

PHASE 6 · WEEKS 17–18

Performance & NEON

Profiling, pipeline effects and SIMD with NEON intrinsics.

- Profiling on ARM
- Pipelines and branch prediction
- NEON intrinsics

You build: 2x faster DSP routine with NEON

PHASE 7 · WEEKS 19–20

Capstone

Build a minimal bare-metal bootloader for a Cortex-A board.

- Bootloader design
- Image loading and verification
- Demo and review

You build: Bootloader that loads and verifies an image

Your certificate

You earn it by doing two things:

- Attend every live session. If you miss one, make it up by watching the recording and submitting that week's lab within 7 days. You can do this up to 2 times per course.
- Score more than 50% on the final evaluation at the end of the course.

Your Enterprise Readiness Rating is separate: it scores your projects, capstone and assessments so employers can see how job-ready you are.

Questions

Email academy@originclouds.io, or see the course online: https://academy.originclouds.io/courses/arm-developer