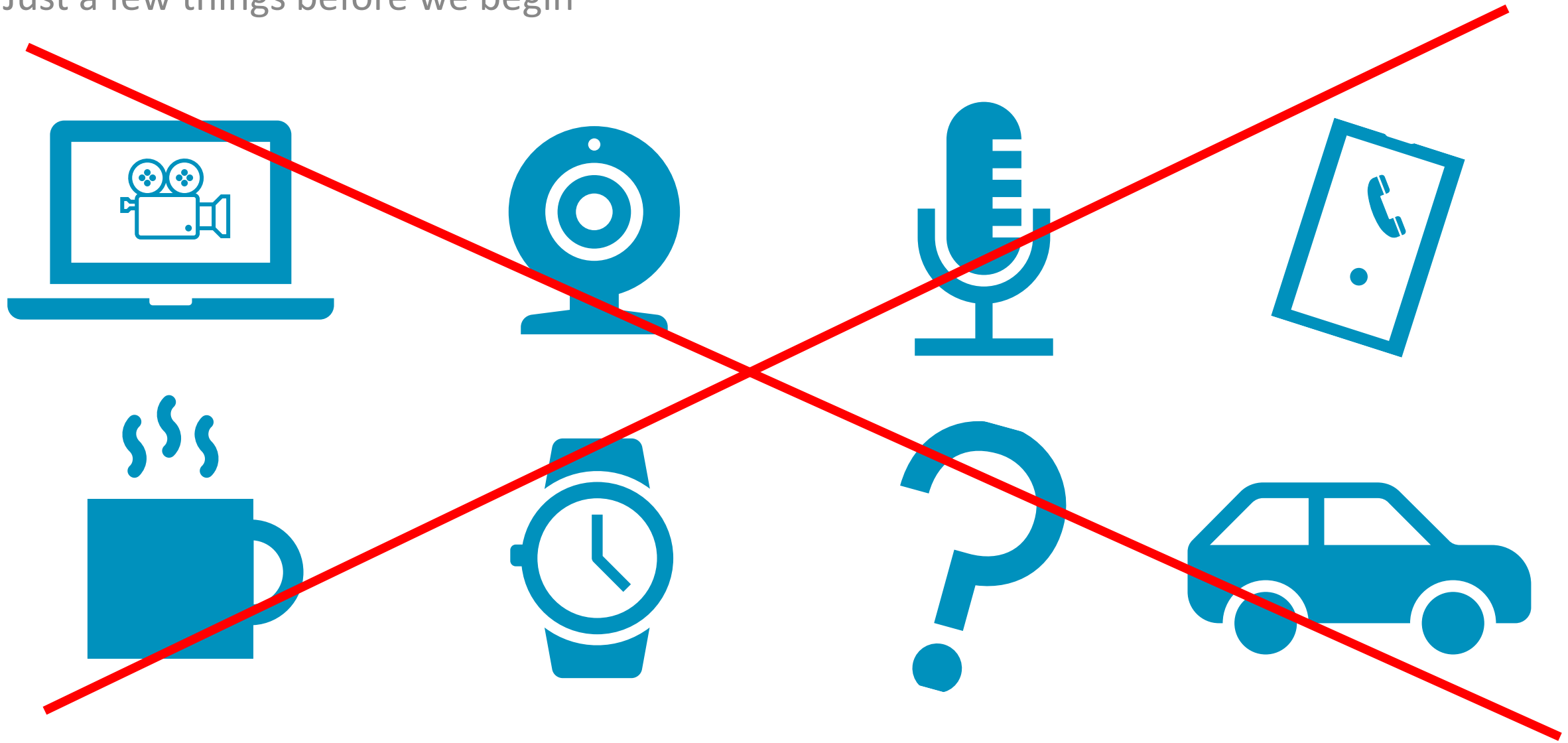
Introduction to the Arm Cortex-M Architecture and Software Development Specialization

Virtual house rules

Just a few things before we begin



What does Arm do?

Arm founded in November 1990

- Advanced RISC Machines

Best known for its range of RISC processor cores designs

- Other products – fabric IP, software tools, models, cell libraries - to help partners develop and ship Arm-based SoCs

Company headquarters in Cambridge, UK

- Processor design centers in Cambridge, Austin, Sophia Antipolis and Hsinchu
- Sales, support, and engineering offices all over the world

Arm does *not* manufacture silicon

More information about Arm and our offices on our web site:

- https://www.arm.com/company

Arm Architecture Profiles

Cortex-A

Highest performance

Designed for high-level operating systems

Cortex-R

Faster responsiveness

Designed for high performance, hard real-time applications

Cortex-M

Smallest/lowest power

Designed for discrete processing and microcontrollers

SecurCore

Tamper resistant

Designed for physical security

Development of the Arm Architecture

Halfword and signed halfword/byte support
System mode
Thumb instruction set

Improved Arm/Thumb Interworking
CLZ
Saturated arithmetic
DSP multiply-accumulate instructions

SIMD Instructions
Multi-processing
v6 Memory architecture
Unaligned data support
Extensions
Thumb-2 (v6T2)
TrustZone (v6Z)
Multicore (v6K)
Thumb-only (v6-M)

Thumb-2
NEON
TrustZone
Virtualization
Architecture Profiles
v7-A (Applications):
NEON
v7-R (Real-time):
Hardware divide
v7-M (Microcontroller):
Hardware divide, Thumb-only

64-bit registers
Privilege Levels
New exception model
New memory model
New instructions
Armv7 compatible

v8-M (Microcontroller)
32-bit only
Baseline / Mainline
TrustZone for Armv8-M

Note that implementations of the same architecture can be different:

- Cortex-M3 - architecture v7-M with a 3-stage integer pipeline
- Cortex-M7 - architecture v7-M with a 6-stage integer pipeline

Arm Cortex Advanced Processors

Architectural compatibility across a wide application range

Arm Cortex®-A family:

- Applications processors for feature-rich OS and 3rd party applications

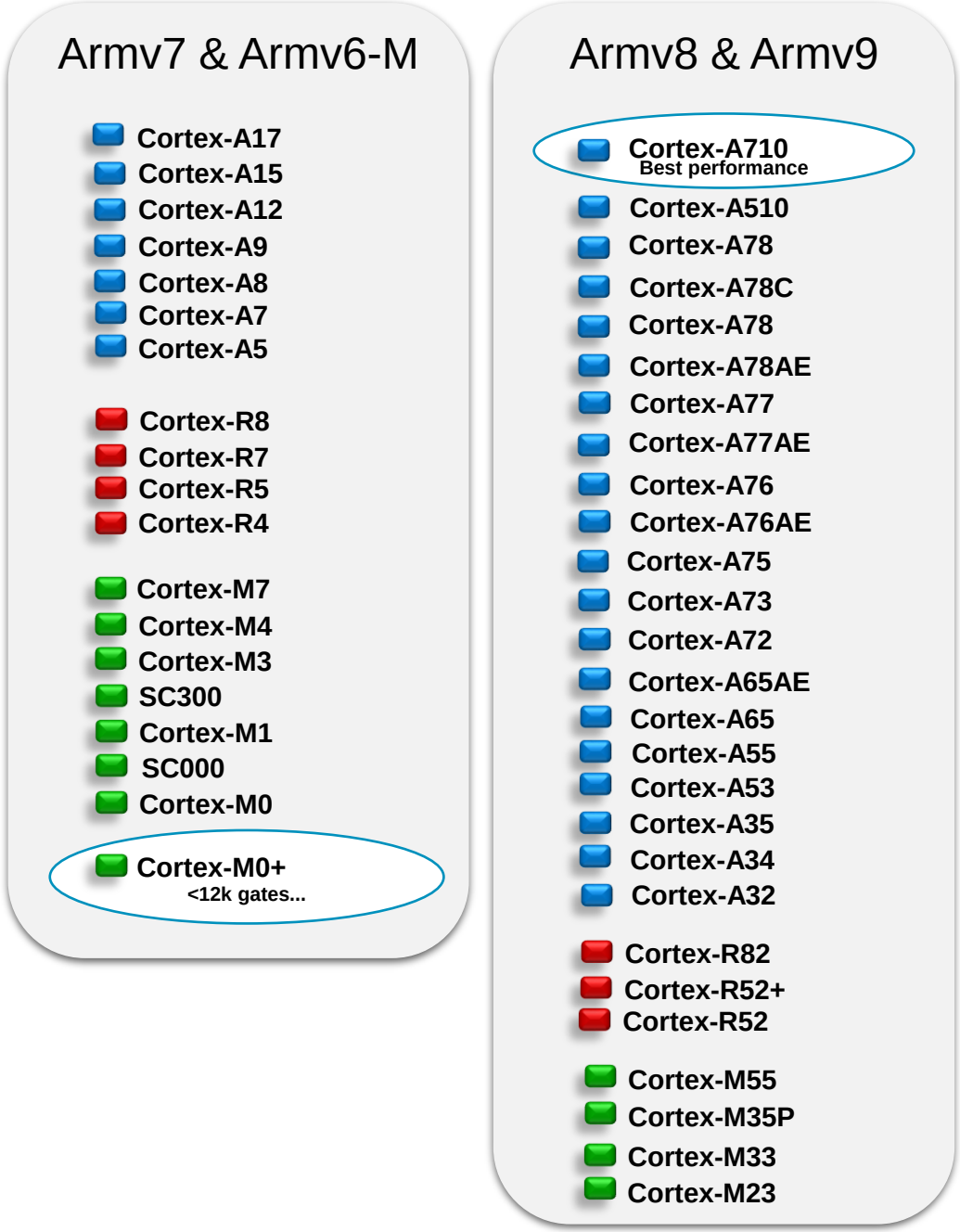
Arm Cortex-R family:

- Embedded processors for real-time signal processing, control applications

Arm Cortex-M family:

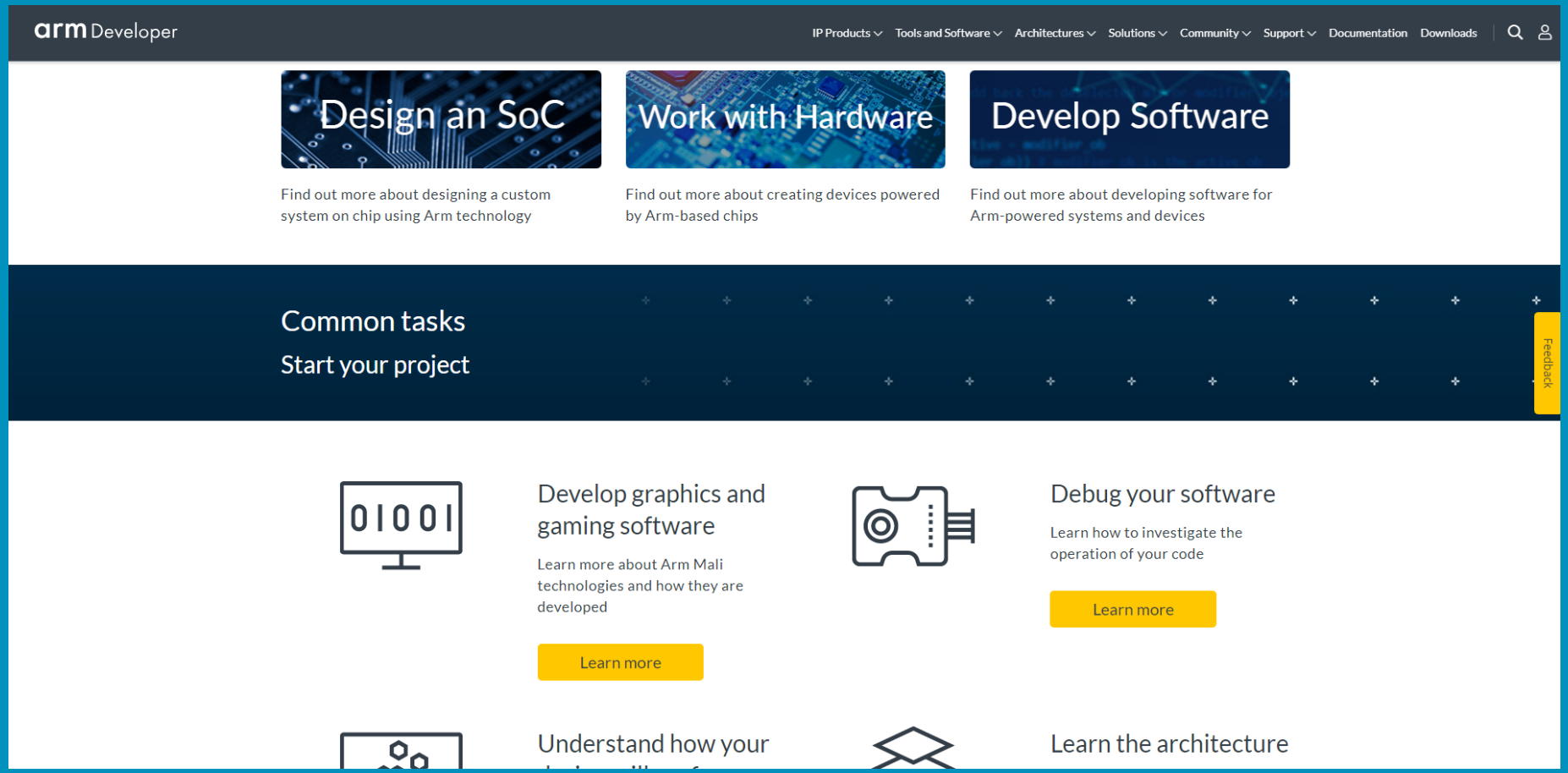
- Microcontroller-oriented processors for MCU, ASSP, and SoC applications
- Arm SecurCore™ and Cortex-M35P - Tamper-resistant security

Arm Neoverse and Cortex-X processors also fall under the Application profile



Arm Developer

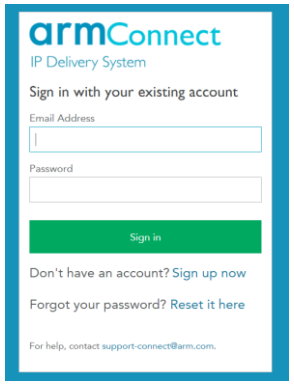
Developer resources website

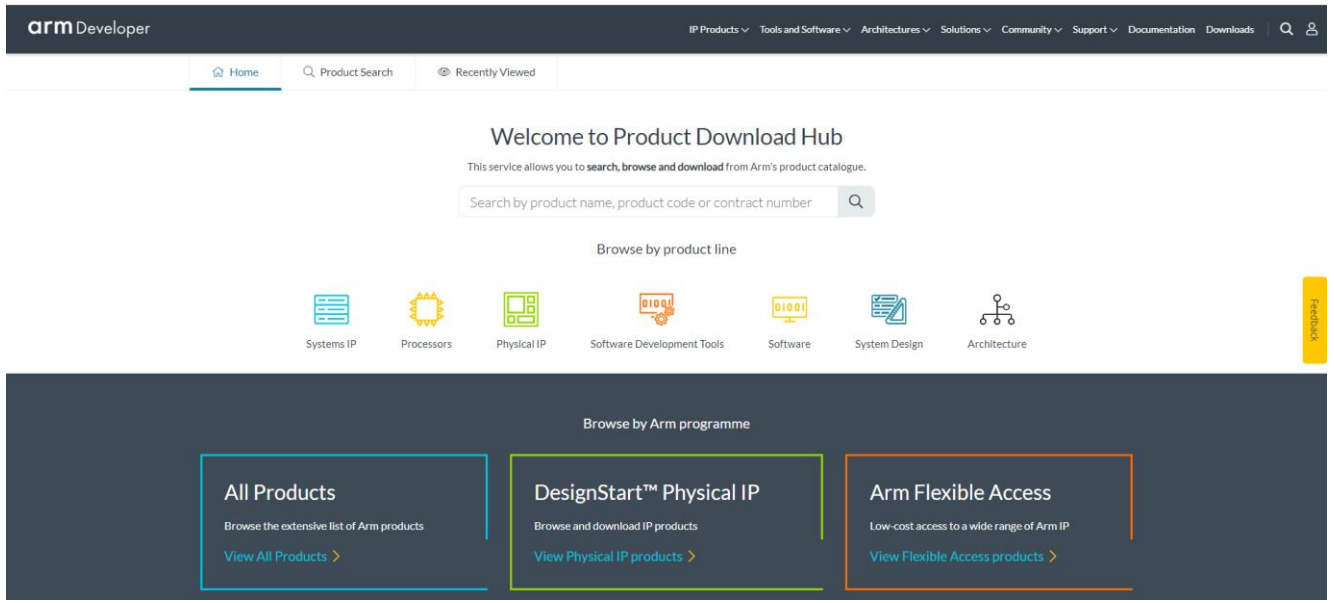


Downloading Arm products

Products bundles are currently delivered via two IP delivery systems:

- https://developer.arm.com/downloads
- https://connect.arm.com





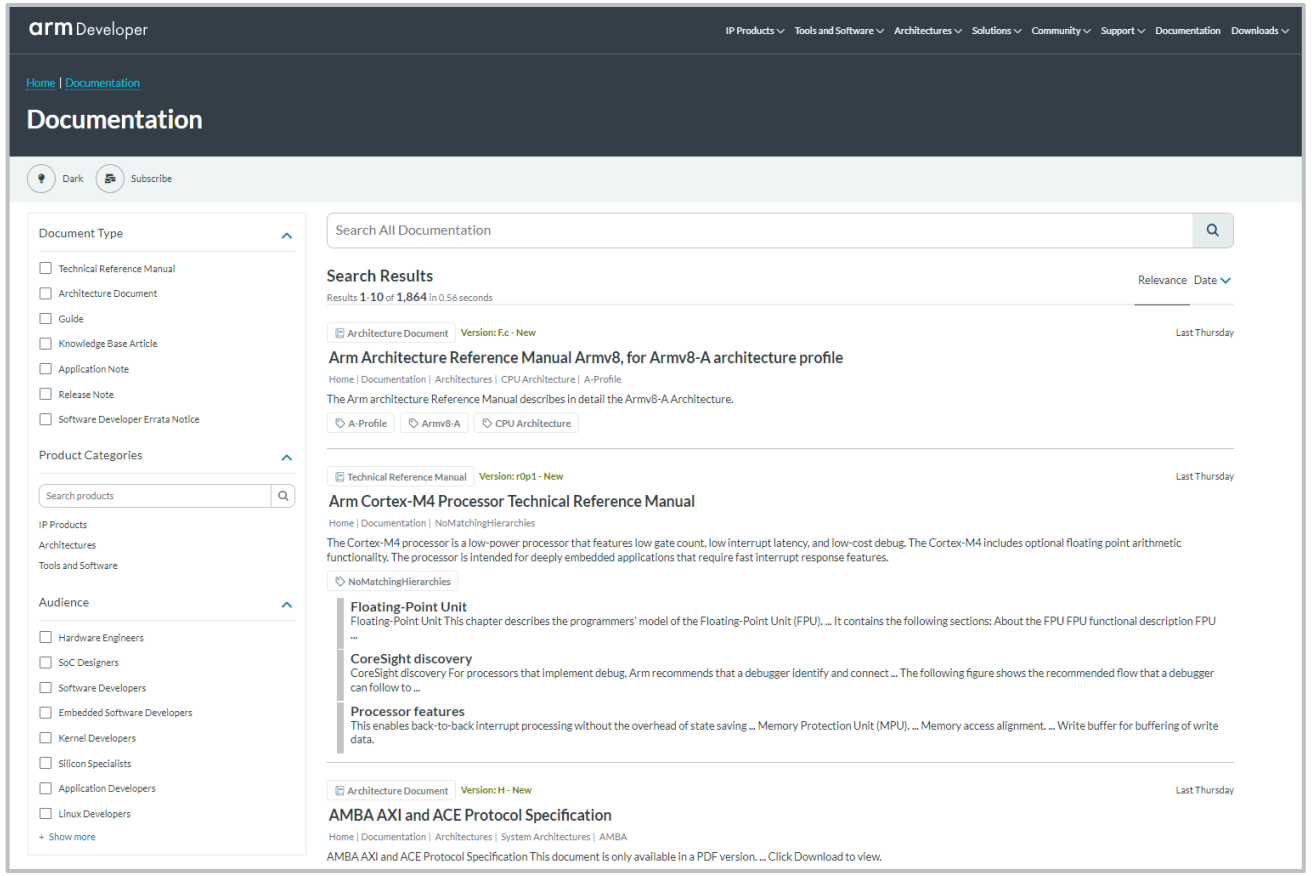
Most of Arm's IP is restricted to licensees

Non-restricted access to other downloads (license sometimes required)

Arm Documentation

Useful sections include

- Arm architecture
- Cortex-A/R/M series processors
- Arm Technical Support Knowledge Articles
- Application Notes and Tutorials
- Developer Guides and Articles

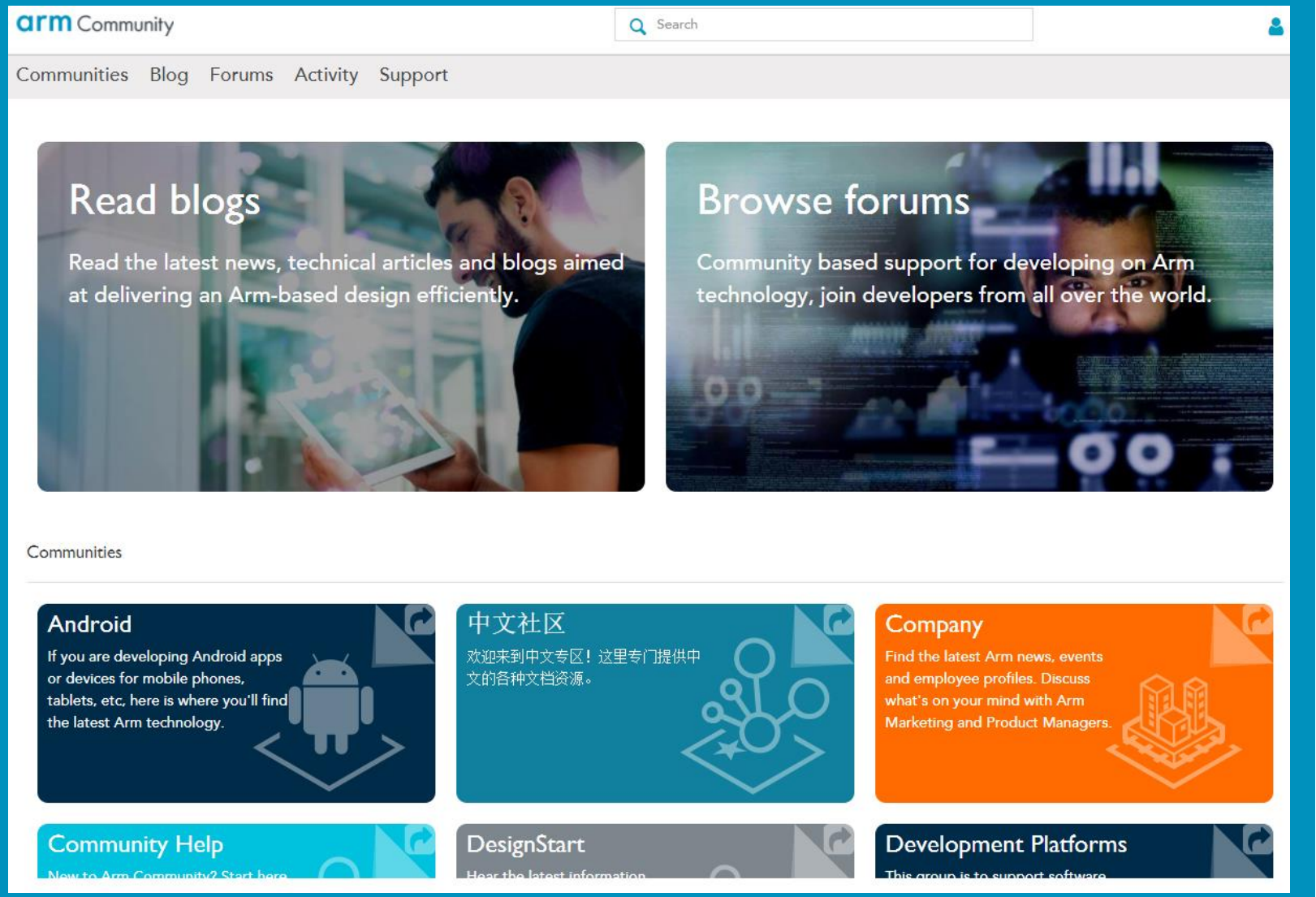


https://developer.arm.com/documentation

Arm Community

- Arm blogs
- Forum posts

https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/cortex-m-resources



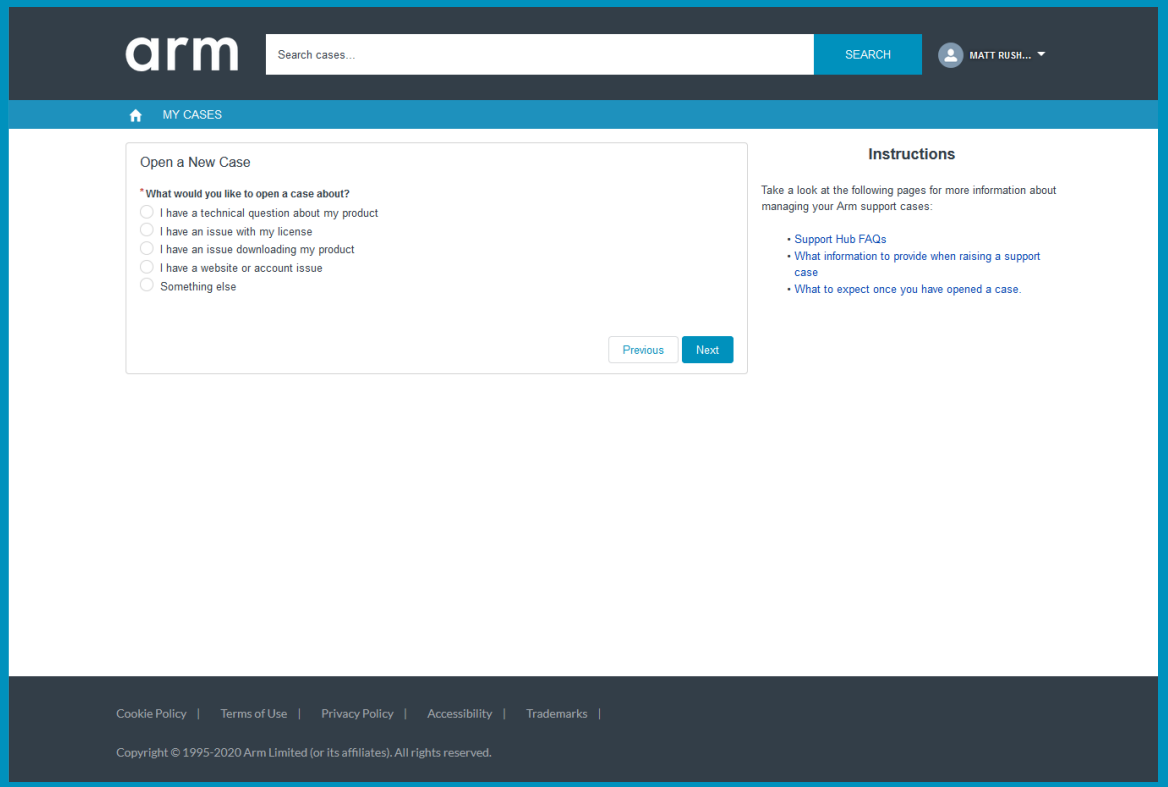
http://community.arm.com/

Technical Support

Arm Support: **https://developer.arm.com/support**

Technical Support Options

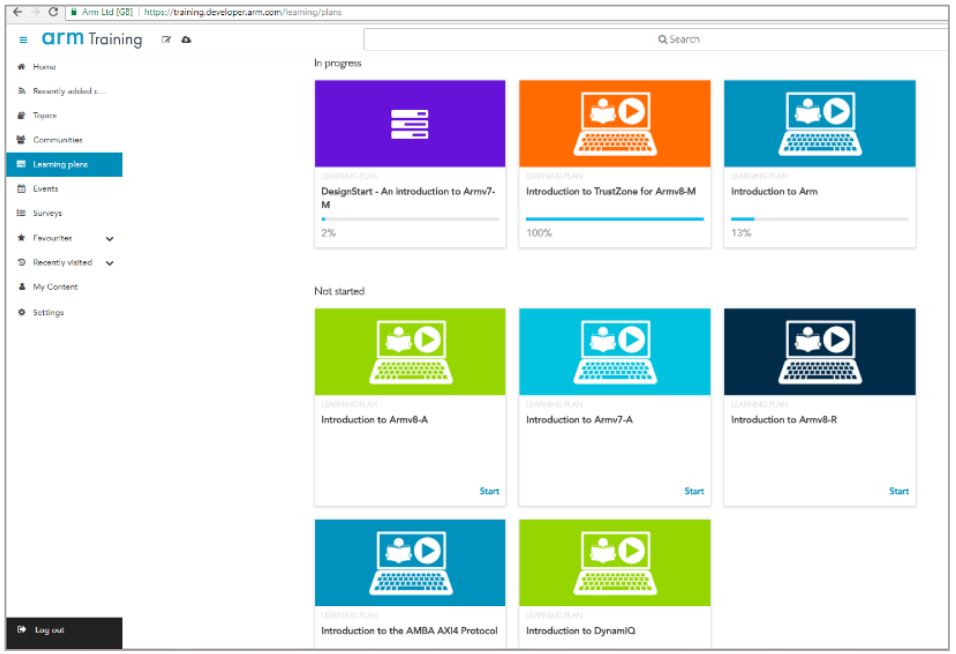
Features	Standard Support	Enhanced Support
Unlimited number of questions/cases	✓	✓
Case tracking system	✓	✓
Maintenance	✓	✓
Online documentation	✓	✓
Named AEs		✓
Proactive project engagement		✓
Onsite presence available		✓

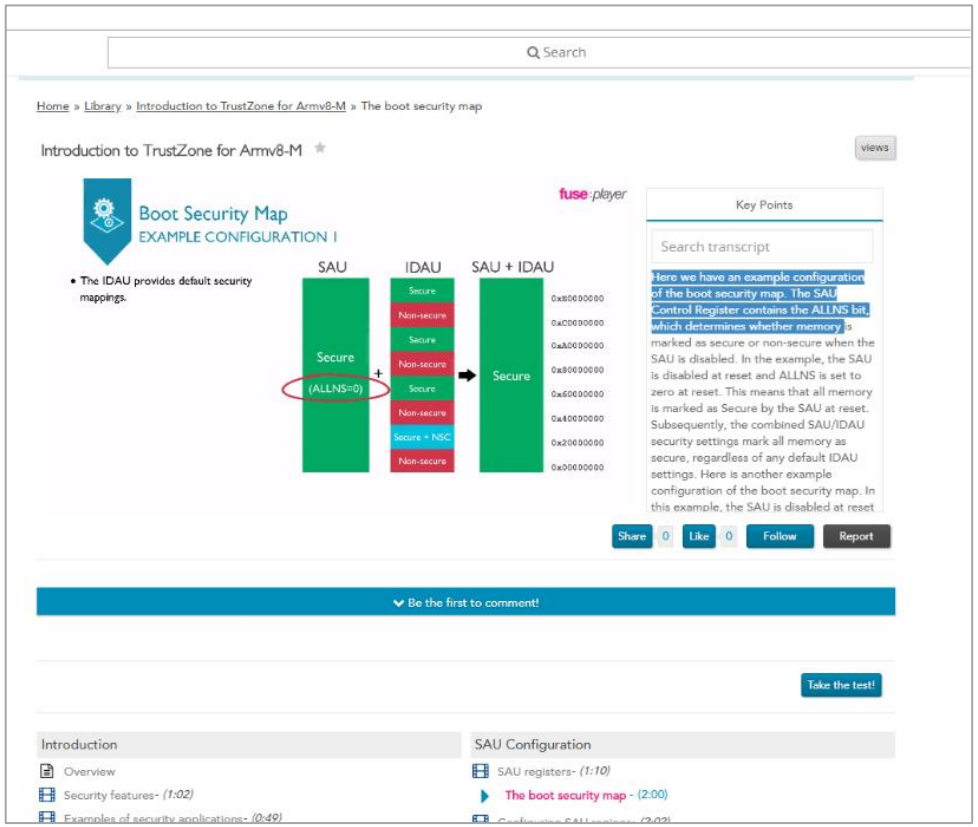


Arm Training

Arm offers different types of training: https://developer.arm.com/training

- Face-to-face training
- Virtual training
- Online learning: https://developer.training.arm.com





Fundamental course objectives

At the end of this course you will be able to:

- Describe the implementation differences and configuration options of the different Cortex-M processors.
- Describe the programming model (register set, modes and states of the processor).
- Describe standards and APIs that support the architecture.
- Interpret and program in assembly language.
- Target applications at the default system memory.
- Configure the MPU to change the default system memory map.
- Handle processor exceptions such as interrupts and faults.
- Build an application targeting a Cortex-M system.
- Debug an application running on a Cortex-M system.

Further course objectives

At the end of this course you will be able to:

- Develop software that can safely access memory which is shared between multiple threads or processors.
- Utilise and manage caches in software.
- Optimise software to make use of the optional DSP Extension.
- Describe how floating-point hardware can help speed-up software that contains floating-point operations.
- Partition memory to allow secure and non-secure applications to run on the same device.
- Program the SysTick timer for different use cases.
- Authenticate pointers and identify branch targets to make software more secure.
- Optimise software to make use of the optional Microcontroller Vector Extension (Helium).

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה