CAIRO UNIVERSITY
ECO RACING TEAM

Session1: Intro to STM32

CUERT

Agenda

1. Embedded Systems Concepts
2. Programming concepts
3. STM32 Cube IDE

Embedded Systems Concepts

Embedded Systems

Microcontroller Architecture

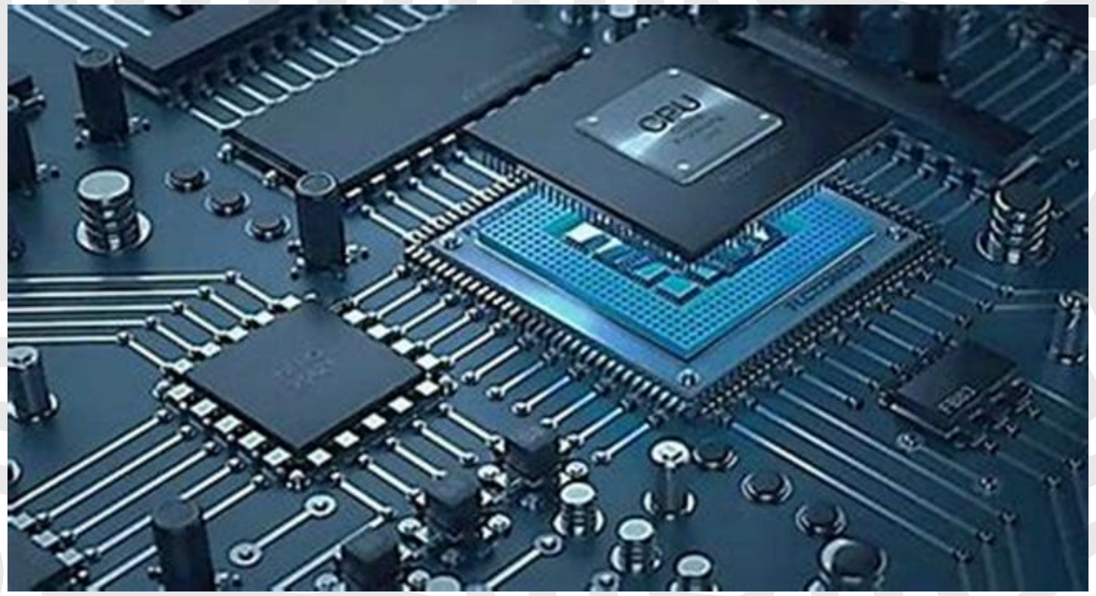
Embedded Systems Concepts

Embedded systems

a. What’s an Embedded System

- A combination of software and hardware

- Performs a specific task as a part of larger system



Embedded systems

a. What’s an Embedded System

- Embedded system is a system include software[Programing language] and hardware[Microcontroller]combination to Reach specific target.

Specific Target

Embedded System

Embedded systems

a. What’s an Embedded System

- An embedded system is a specialized computing system designed to perform dedicated functions or tasks within a larger system. Unlike general-purpose computers, embedded systems are optimized for efficiency, reliability, and real-time performance.

+

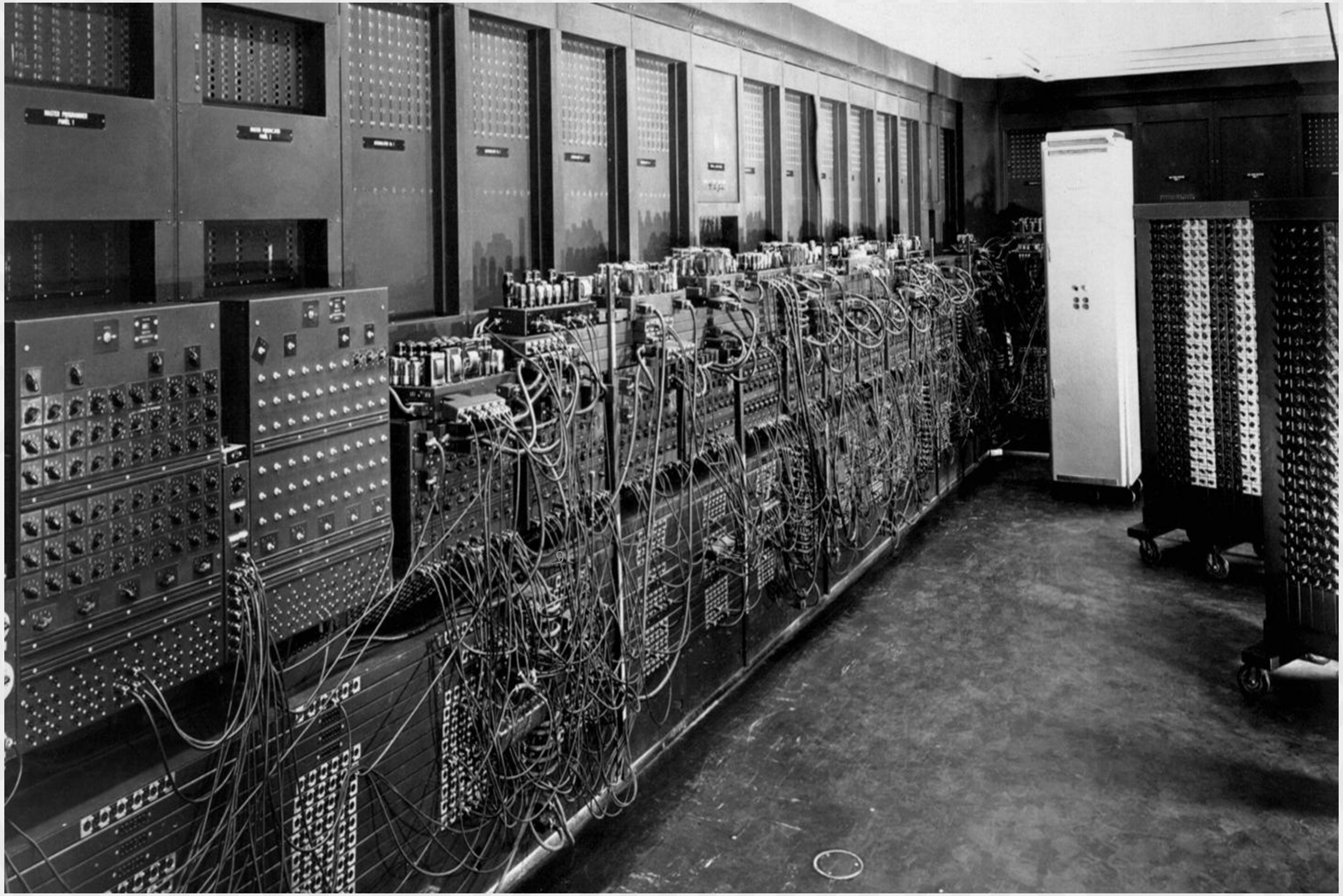
+ Specific Target = Embedded System

b. Embedded system and computing system

Embedded system	computing system
Specialized system designed for a specific task	System designed for multiple tasks and user interactions
Optimized for low power consumption	Higher power requirement
Ex: washing machines, ATMs , automotive ECUs	Ex : Laptops, desktops, smartphones

Embedded systems

Historic PC

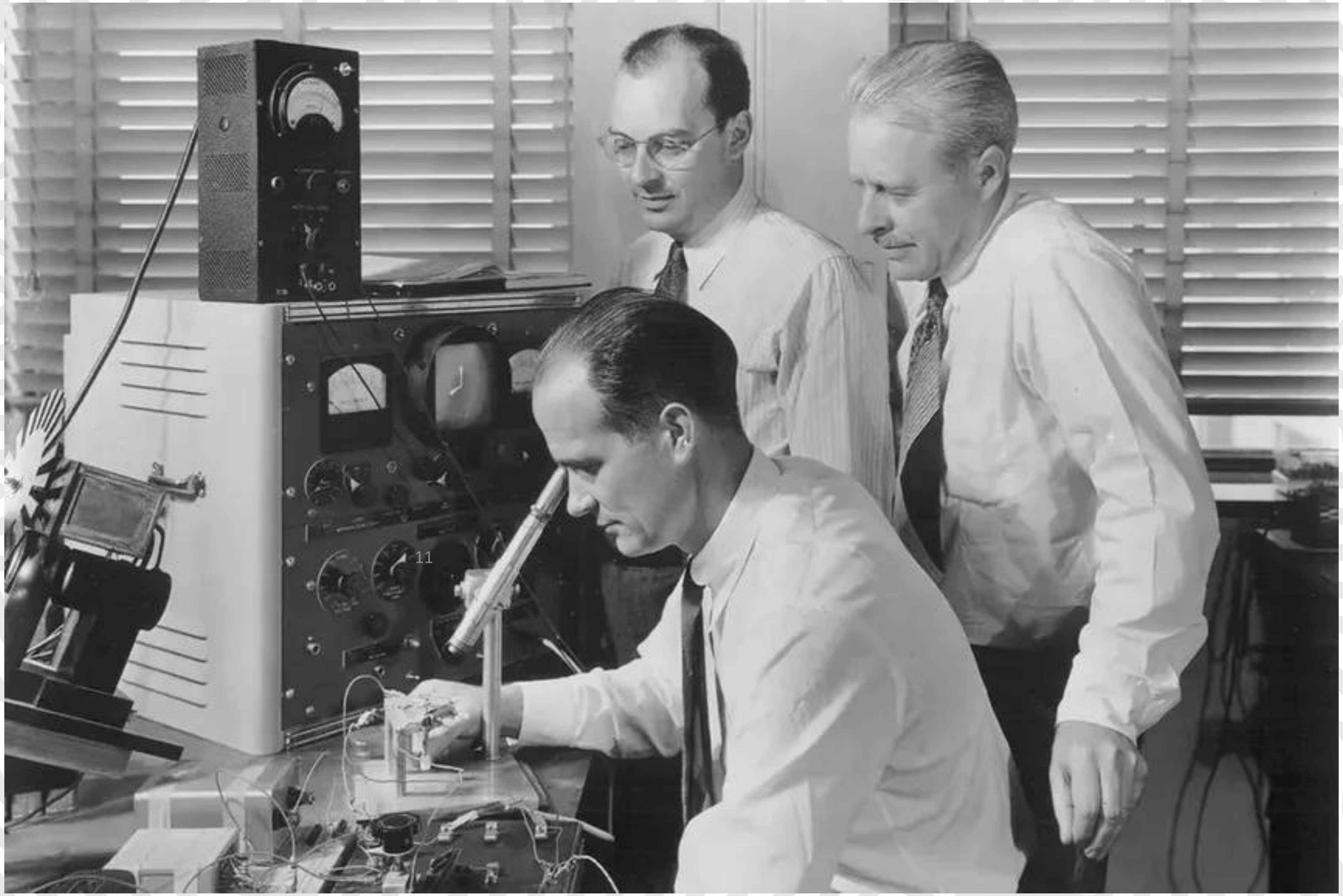


Embedded systems

Cathode Tube

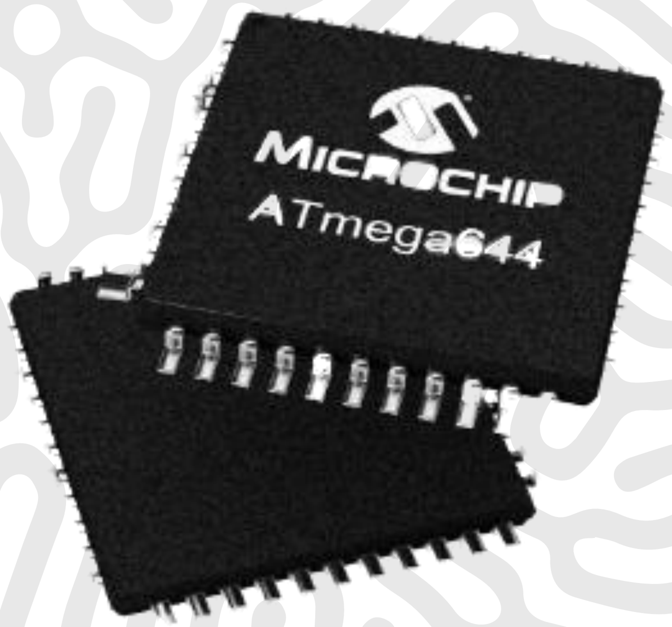
- It were used before transistors and used a lot of energy and area

Historic Transistor



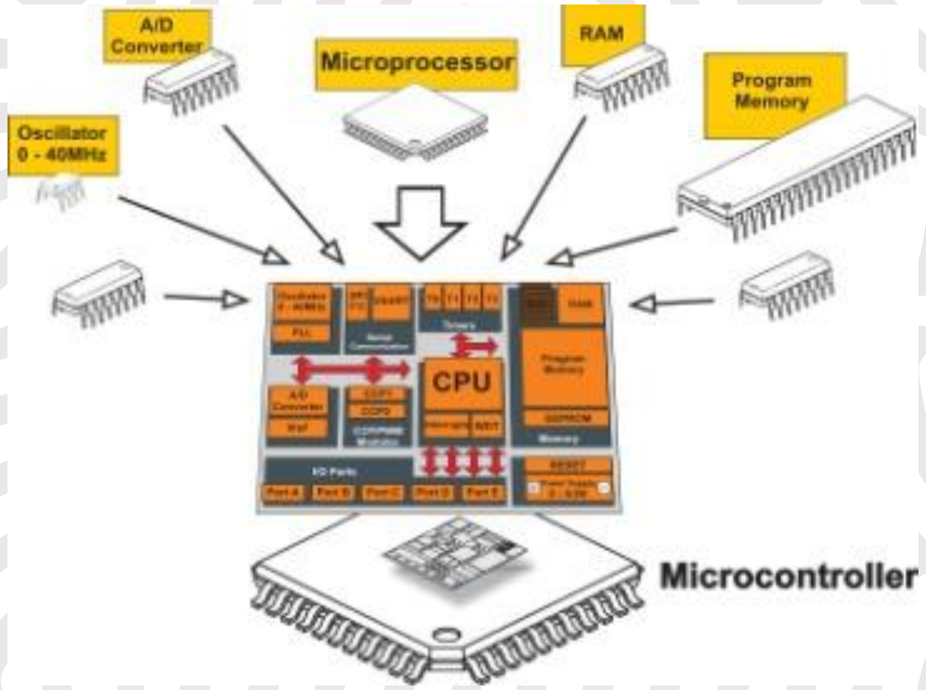
IC Chip

- IC is Integrated Circuit is an assembly of electronic components like capacitors and resistors to perform a specific function like AND gate IC



- Has different shapes but its function is known from the number printed on it and its data sheet not from its shape

c. Microprocessor vs Microcontroller



- It is the brain of the system
- Consist of CPU only

- It is a small computer on single integrated circuit containing

c. Microprocessor vs Microcontroller

MPU	MCU
A processing unit that requires external memory and peripherals	A compact integrated chip that includes a processor, memory and peripherals
Used in computing systems for general-purpose tasks	Used in embedded systems for specific tasks
High power consumption	Low power optimized for energy efficiency

d. How a microprocessor works

1-Code Writing & Compilation

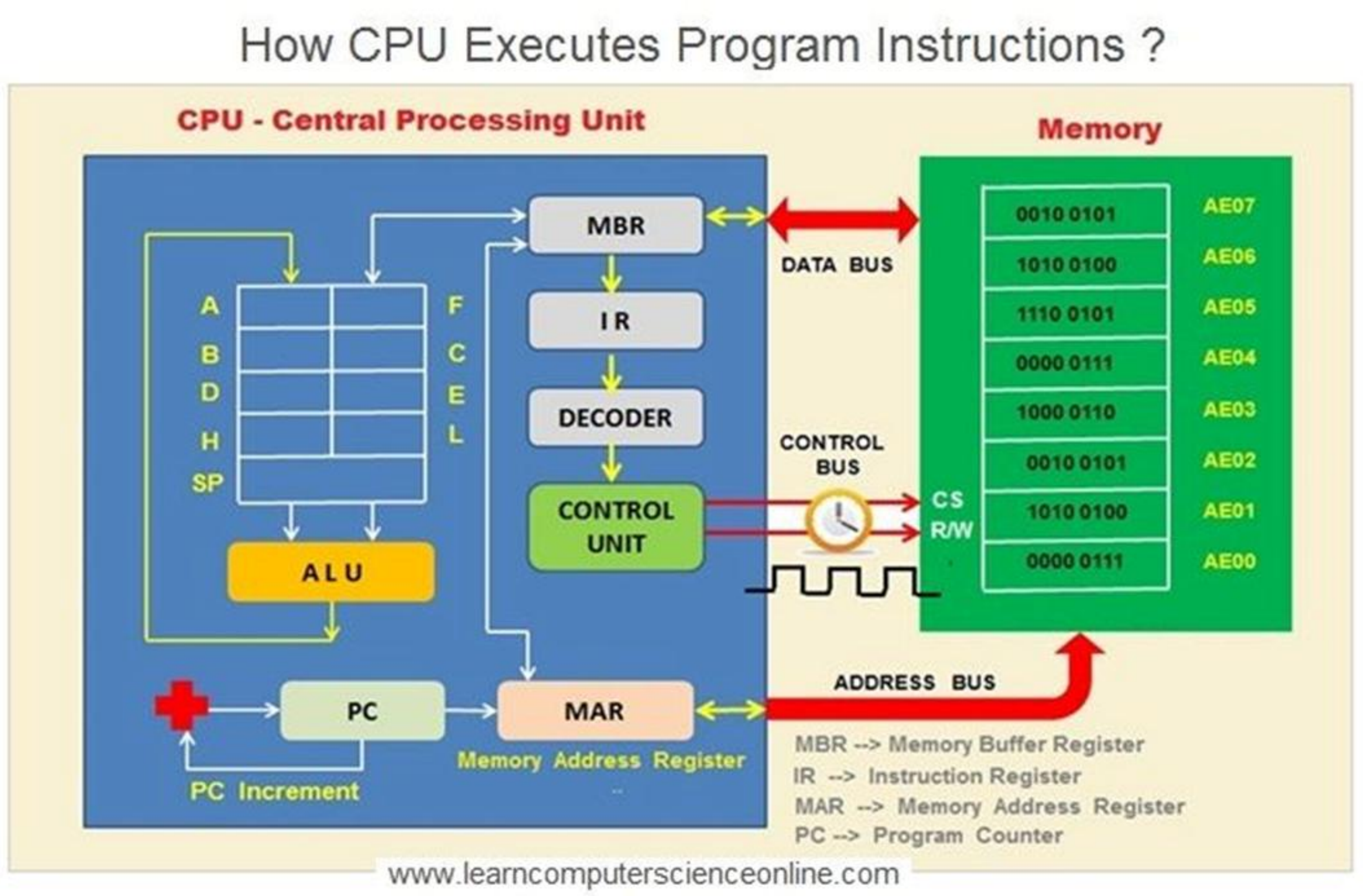
2-Code Upload to Flash Memory

3-Power-On & Reset Process

4-Fetch-Decode-Execute Cycle

5-Interrupt Handling (If Required)

6-Continuous Execution



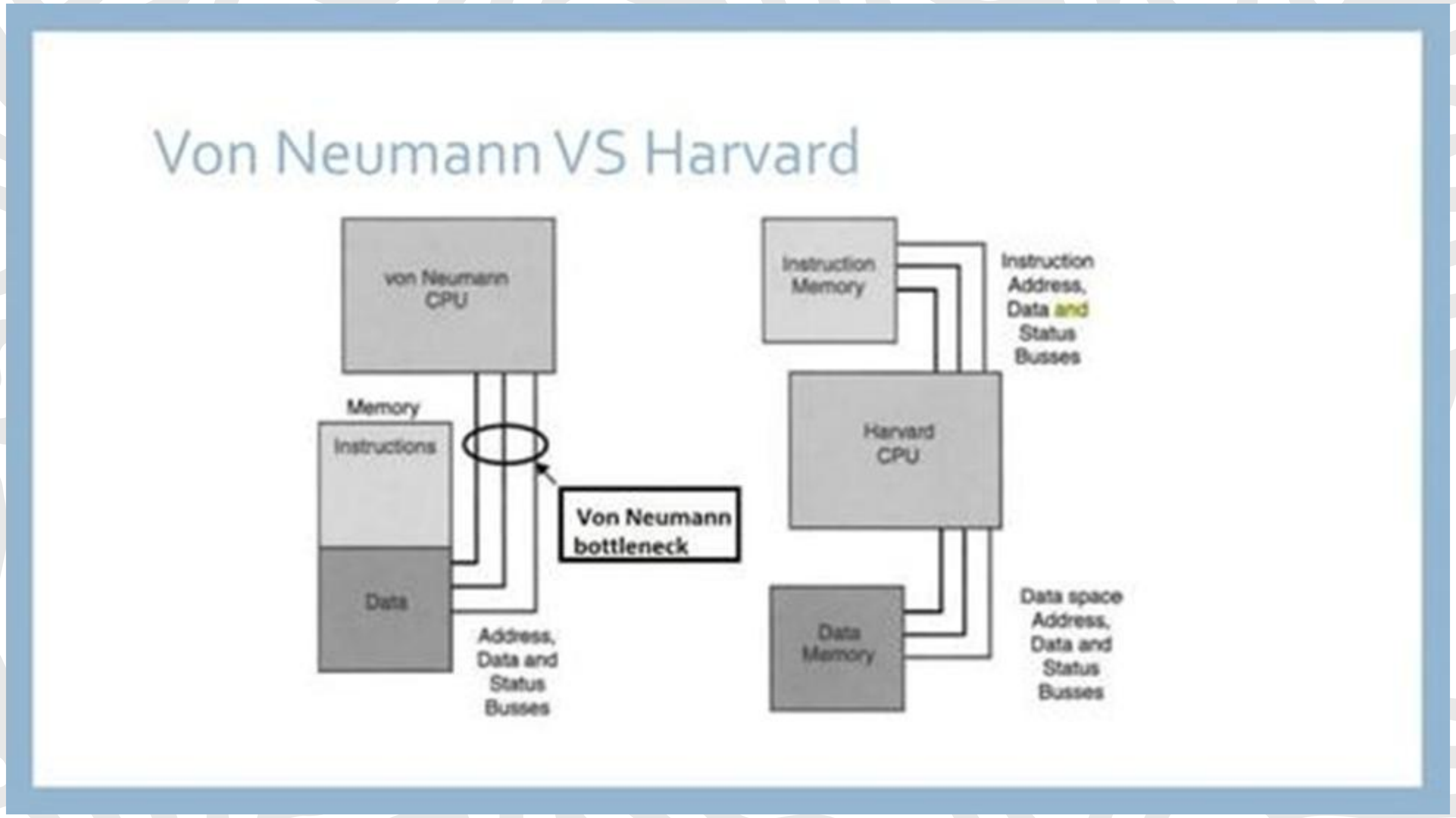
Microcontroller Architecture

CISC vs RISC

CISC vs RISC

CISC	RISC
Complex variable-length instructions	Simple fixed-length instructions
Smaller programs (fewer instructions needed)	Larger programs (more instructions needed)
Fewer general-purpose registers	More general-purpose registers

Von Neuman vs Harvard



Von Neuman vs Harvard

Von Neuman	Harvard
Unified memory for program and data	Separate memory for program and data
Slower due to bus contention	Faster due to simultaneous access
Simpler hardware	More complex hardware

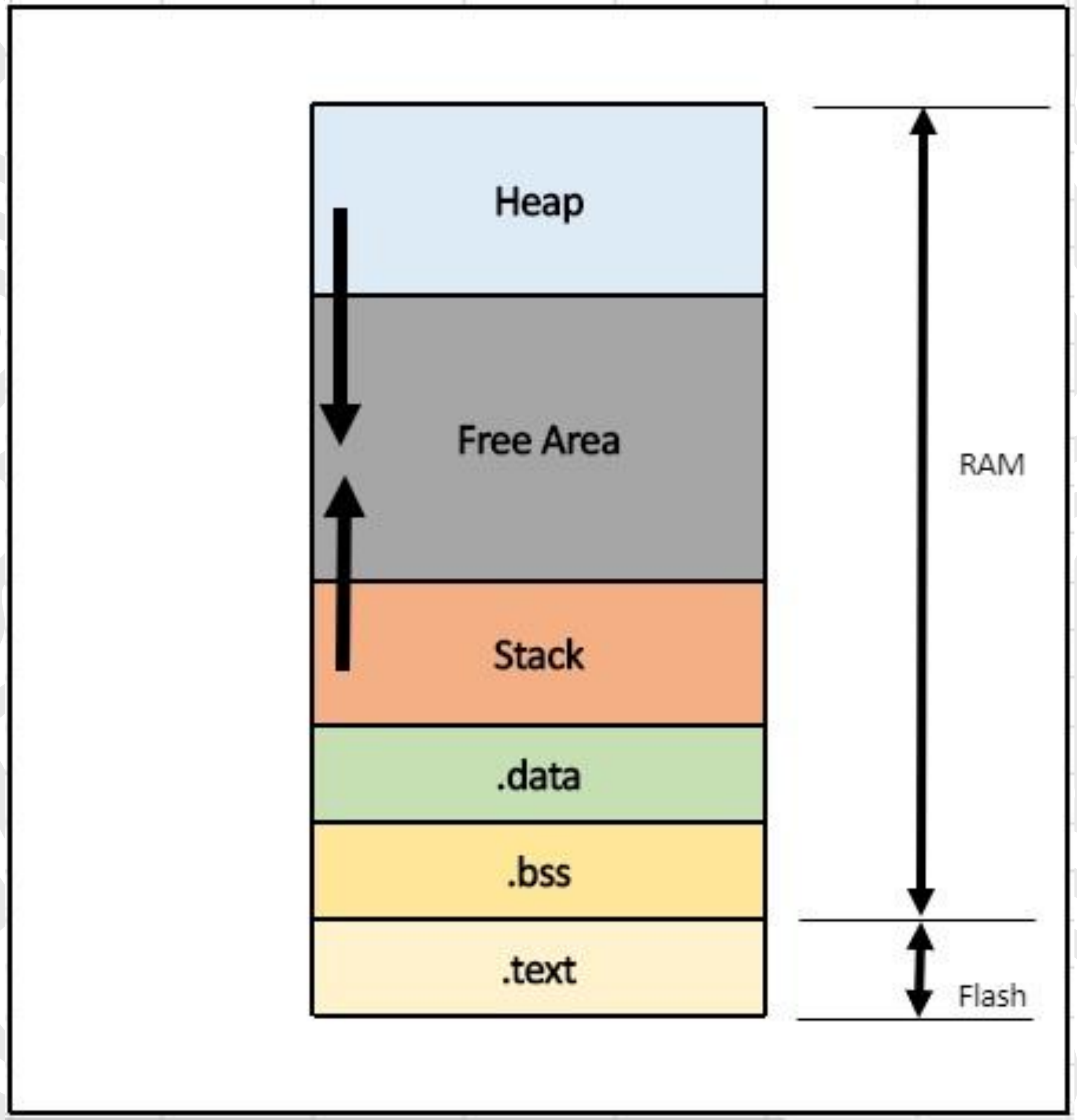
RAM vs ROM

RAM	ROM
Temporary storage for variables,stack and data	Permanent storage for program code (firmware)
Volatile(data lost when power is off)	Non-volatile(data remained after power-off)
Stores runtime variables,stack and temporary data	Stores firmware , bootloader and constants

SRAM vs DRAM

SRAM	DRAM
Transistors	Capacitors
Faster	Slower
Expensive	Cheap
Lower power	Higher Power

Memory Map



Masked ROM (MROM)

- **Description:** Pre-programmed during manufacturing. Once data is written, it cannot be modified.
- **Advantages:** Cost-effective for mass production.
- **Disadvantages**: Cannot be updated; errors require a new chip
- **Use Cases:** Embedded systems, gaming cartridges, microcontroller firmware

OTPROM (One-Time Programmable ROM)

- **Description:** Programmable by the user only once using a special device
- **Advantages:** More flexible than MROM; allows programming after manufacturing.
- **Disadvantages**: Cannot be erased or reprogrammed.
- **Use Cases:** Security keys, firmware storage in industrial applications.

EPROM (Erasable Programmable Read-Only Memory)

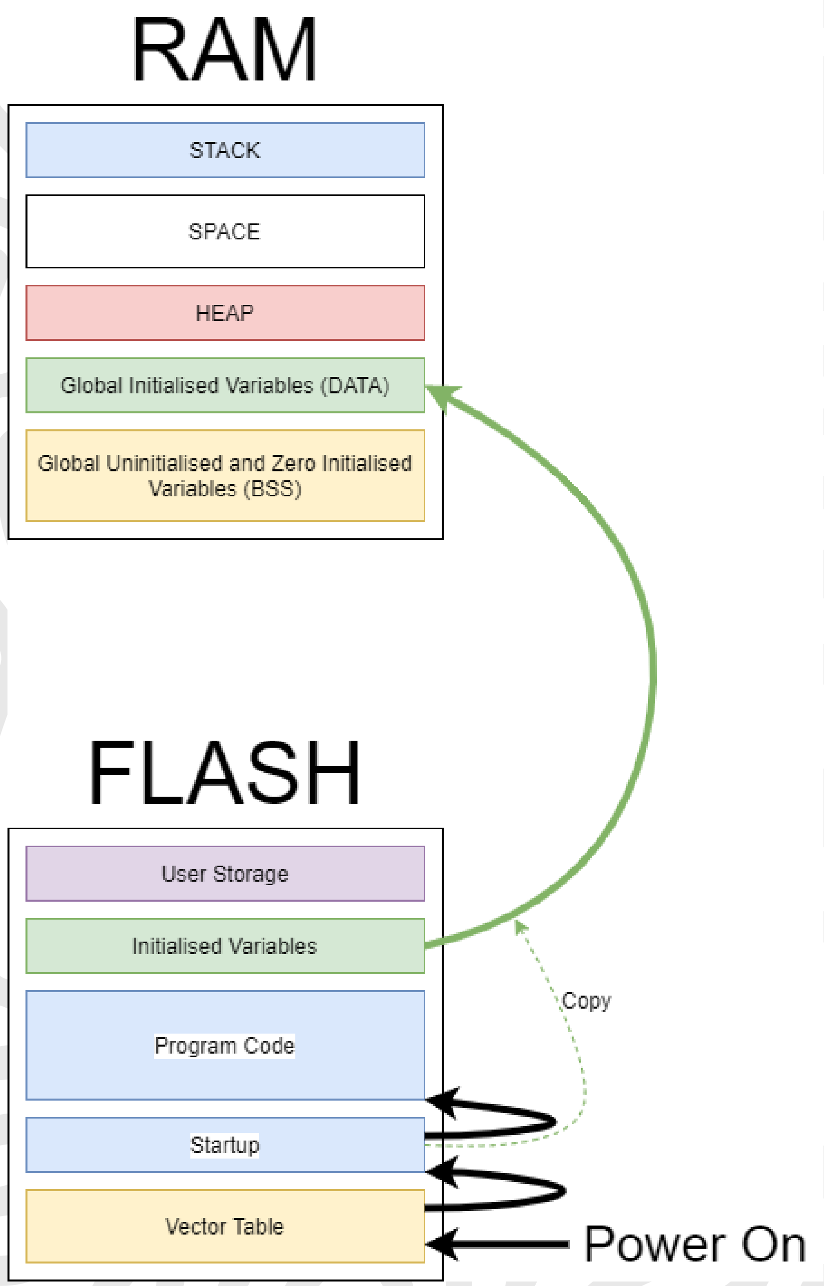
- **Description:** Can be erased using UV light and reprogrammed multiple times.
- **Advantages:** Allows updates and debugging.
- **Disadvantages**: Requires UV exposure for erasure; slower than EEPROM.
- **Use Cases:** Older BIOS chips, microcontrollers, and legacy embedded systems.

EEPROM (Electrically Erasable Programmable Read-Only Memory)

- **Description:** Can be erased and reprogrammed using electrical signals.
- **Advantages:** Byte-level erasing and rewriting, more flexible than EPROM.
- **Disadvantages**: Slower write speeds, limited write cycles.
- **Use Cases:** Microcontrollers (e.g., for storing configuration data), smart cards, RFID.

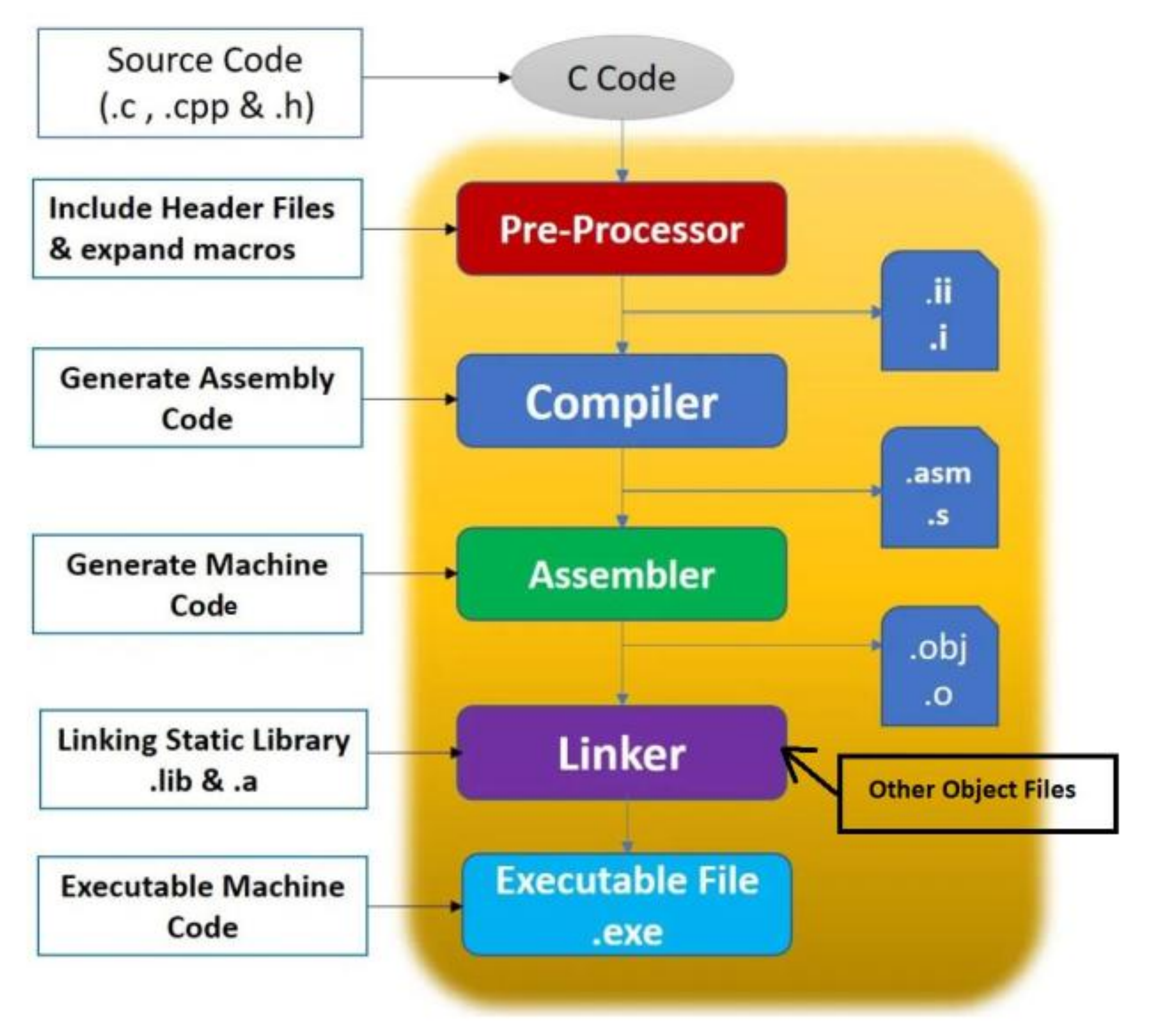
Flash Memory

- **Description:** An advanced form of EEPROM that erases and writes in blocks rather than bytes.
- **Advantages:** Faster than EEPROM, high storage density, cost-effective.
- **Disadvantages**: Limited write cycles (~10,000 to millions), block-level erasure.
- **Use Cases:** USB drives, SSDs, microcontrollers, mobile phones.



Toolchain

- A toolchain is a set of software tools used to convert source code into an executable binary for a microcontroller or processor.



Toolchain

1-Preprocessing (Preprocessor):

Handles preprocessor directives like #include, #define, and macros

2-Compilation (Compiler)

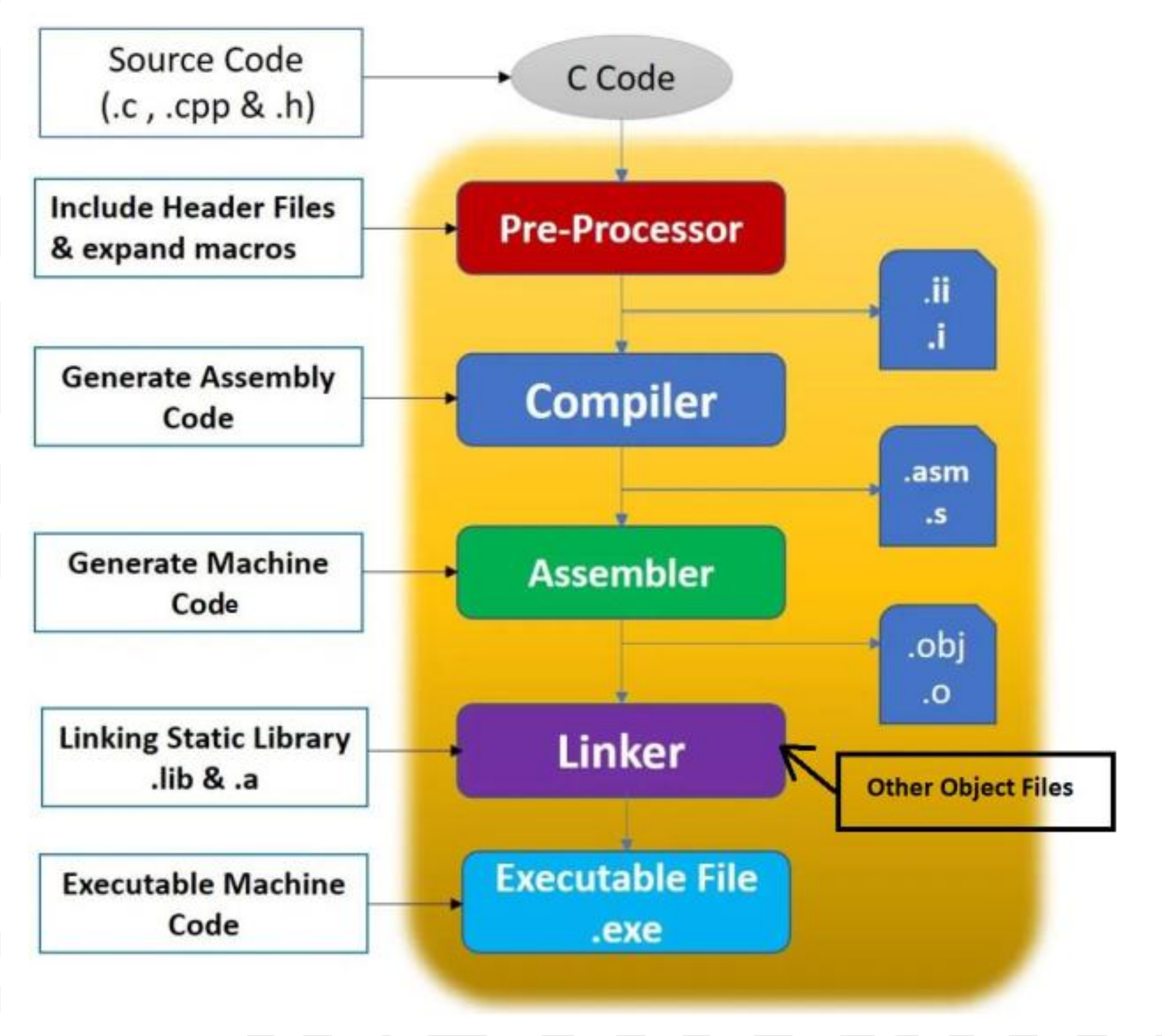
Converts preprocessed C code into Assembly code.

3. Assembly (Assembler)

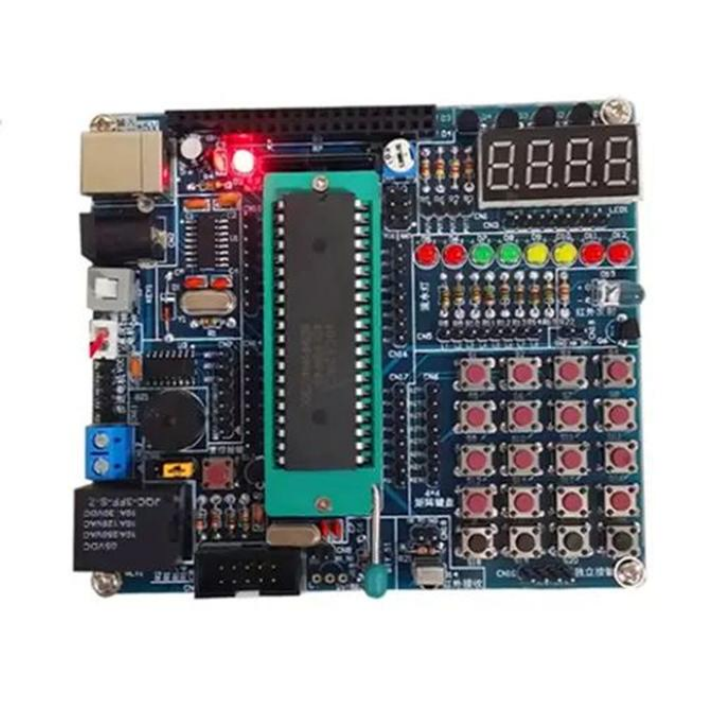
Converts Assembly code into machine code (binary instructions).

4. Linking (Linker)

Combines multiple object files and libraries into a single executable.



5. Flashing



Off-Circuit Flashing:

Offline / External Programmer

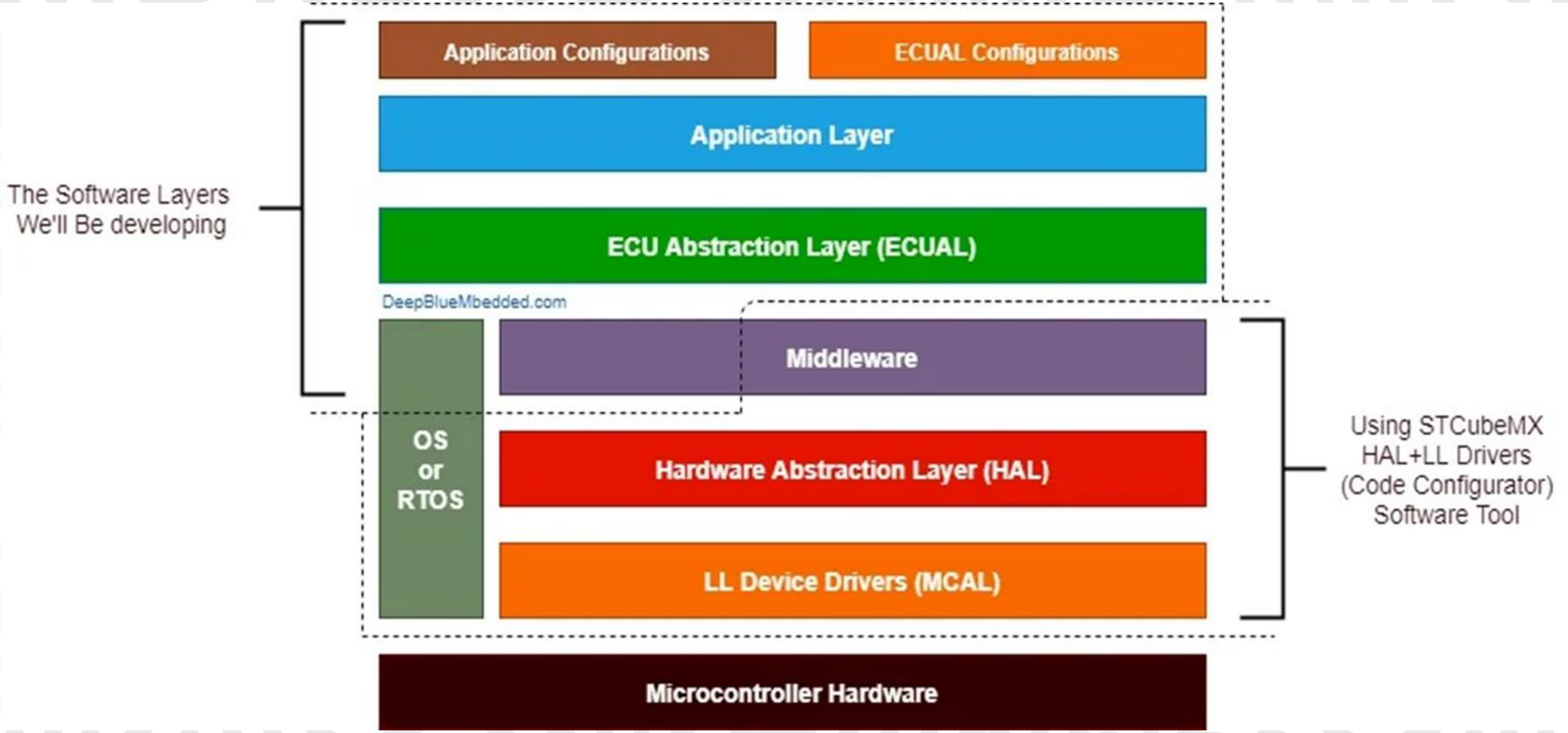
31

On-Circuit Flashing :

In-System Programming - ISP / In-

Circuit Serial Programming - ICSP

Software Architecture



Software Architecture

1-Low-Level (LL)

Drivers Directly interacts with hardware registers

2-Hardware Abstraction Layer (HAL)

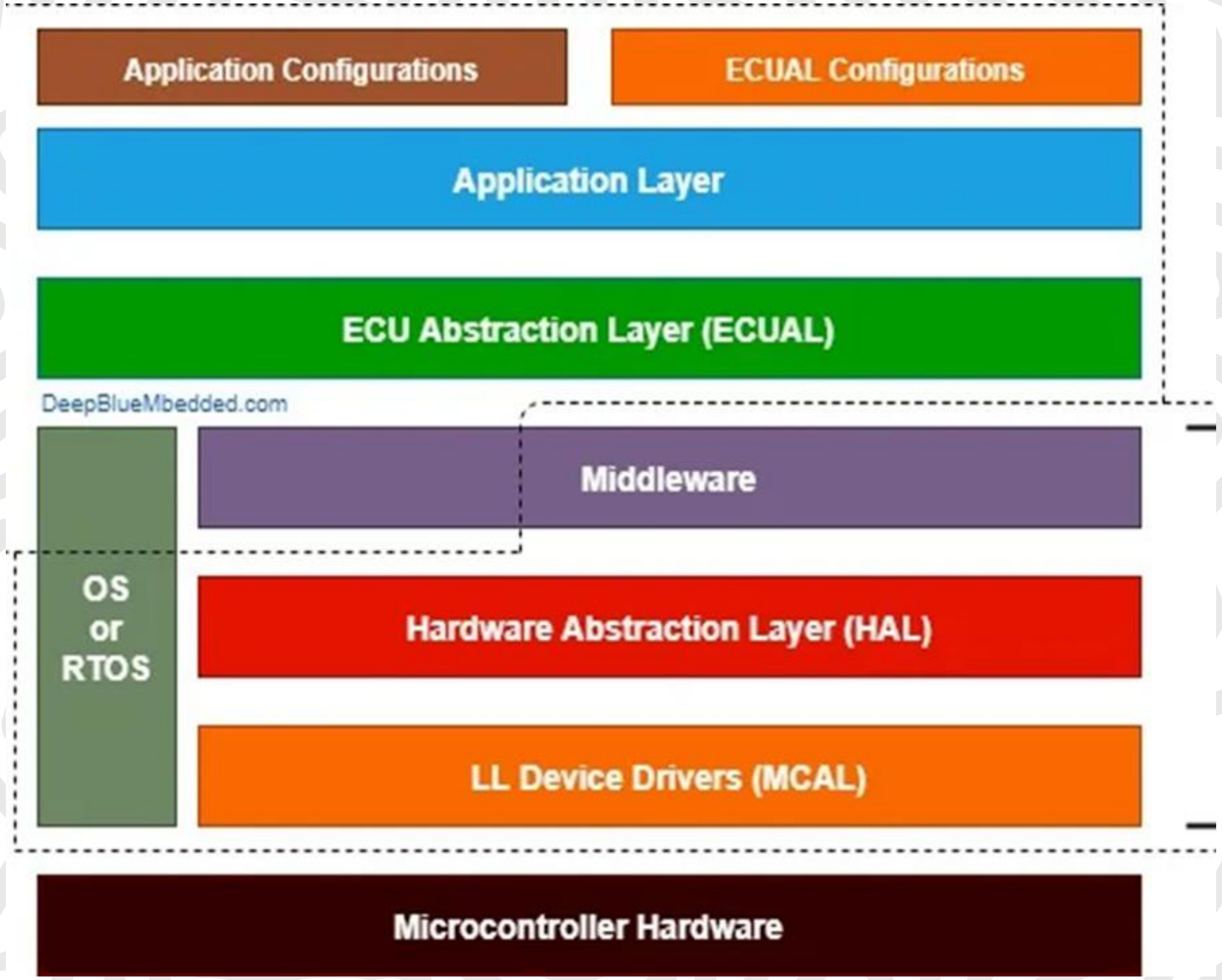
Provides an abstraction between the hardware and application.

3. Application Layer (APP)

The highest layer containing the main application logic.

4. Middleware / Libraries (LIB)

Contains the common libraries needed in most layers.



Break

loading...

57% passed

ARM Architecture

ARM Concepts

Intro to STM32 Cubes

ARM Concepts

ARM

- ARM stands for "Advanced RISC Machine". It is a family of Reduced Instruction Set Computing (RISC) architectures widely used in embedded systems, mobile devices, and high-performance computing.
- ARM processors are known for their power efficiency and scalability.

History of ARM

1-Acorn RISC Machine :

The first ARM processor, ARM1, was designed in 1985 with 25,000

transistors (compared to Intel's 80386, which had 275,000 transistors).

2-Formation of ARM Ltd

In 1990, Acorn Computers, Apple, and VLSI Technology formed

Advanced RISC Machines Ltd.

History of ARM

3-Growth and Adoption :

ARM processors started appearing in mobile phones, PDAs, and embedded systems.

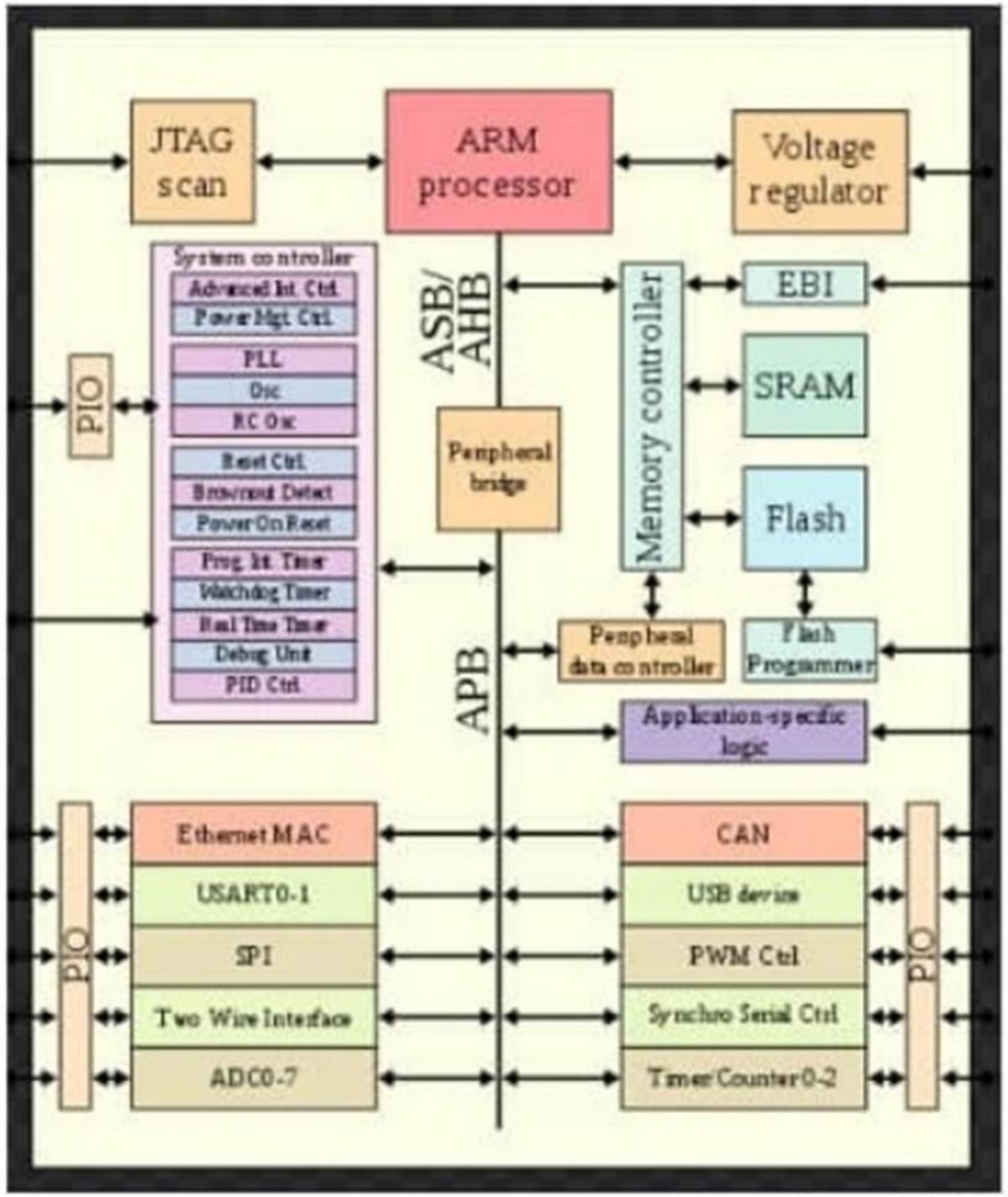
4-Smartphone Revolution :

Apple, Samsung, and Qualcomm started using ARM-based architectures.

5-Modern Era :

ARM is now used in automotive, data centers, AI applications, and even high-performance computing.

ARM Architecture



ARM Architecture

1-Voltage Regulator

A voltage regulator ensures a stable power supply to different parts of an ARM-based system. It converts input voltage to the required levels for core logic, peripherals, and I/O.

2-Parallel I/O (PIO) and Communication Peripherals

Used to interface with GPIO (General-Purpose I/O) pins that can be configured as input/output.

ARM Architecture

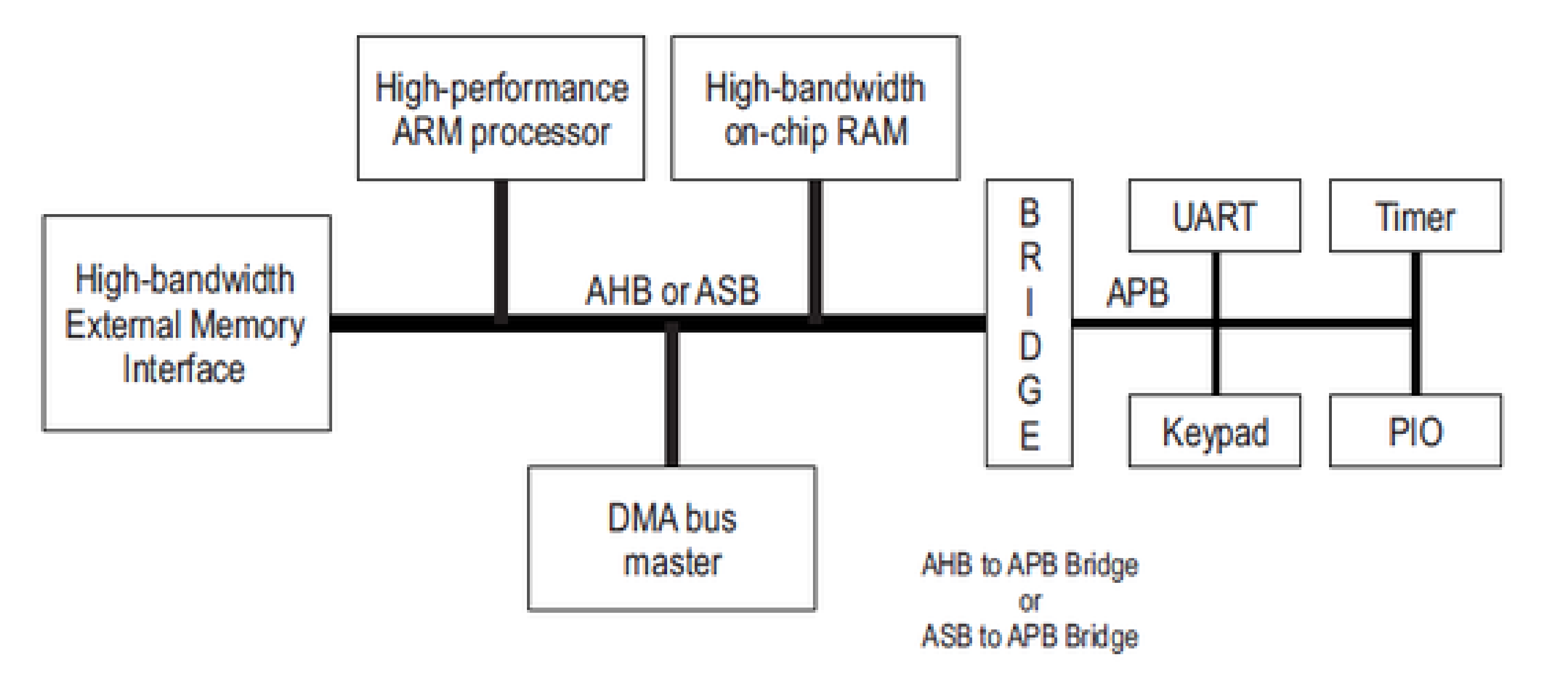
3-AMBA

AMBA (Advanced Microcontroller Bus Architecture) is a protocol developed by ARM to standardize the communication between different components of an ARM-based system. It is widely used in microcontrollers, SoCs (System-on-Chips), and embedded systems to manage high-speed data transfer and peripheral control.

ARM Architecture

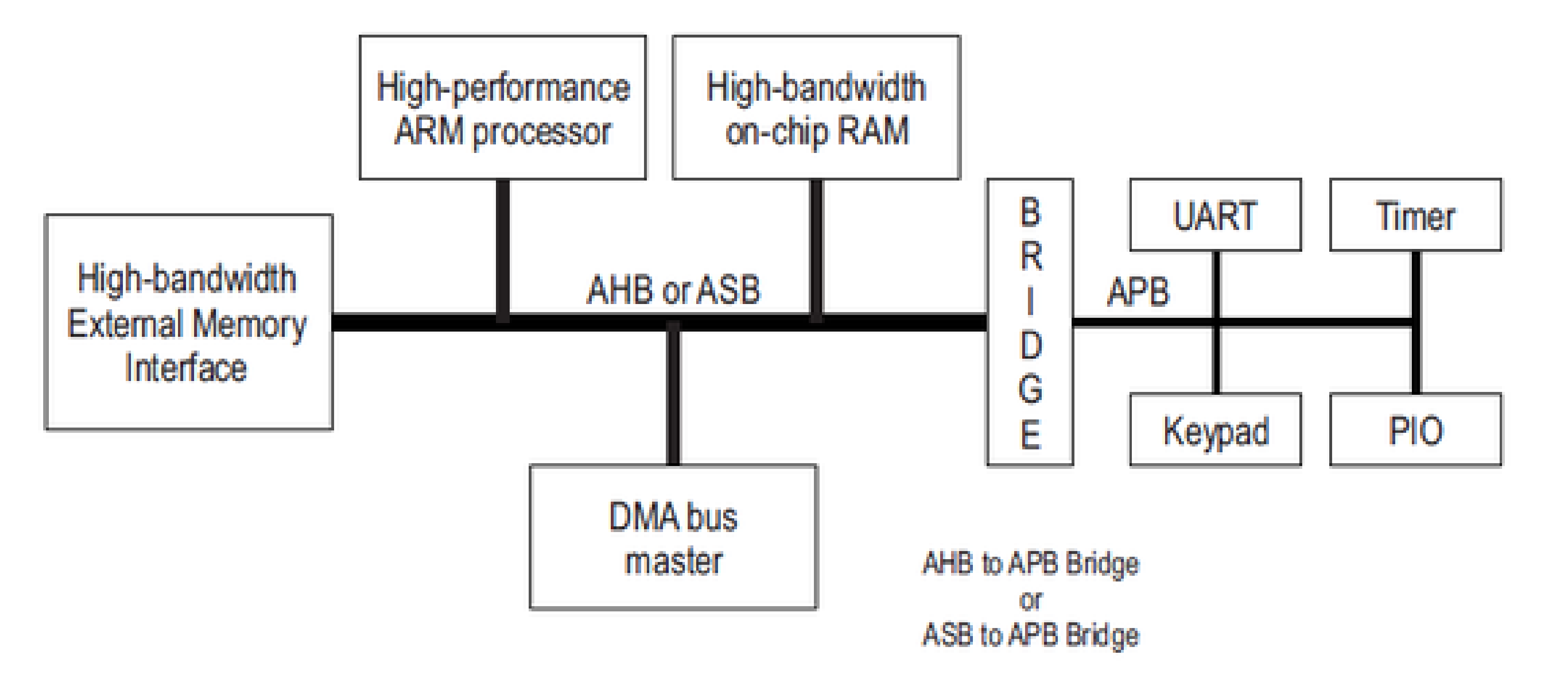
3-AMBA

Peripheral bridge: Since AHB and APB operate at different speeds, a peripheral bridge is used to connect AHB (fast bus) to APB (slow bus).



ARM Architecture

3-AMBA



AHB	APB
Fast	Slow
Complex	Simple

ARM Architecture

4-Power management

Power management in ARM-based microcontrollers involves controlling

clocks and power to various peripherals and system components.

RCC:

is a global controller for system clocks and resets.

Enable:

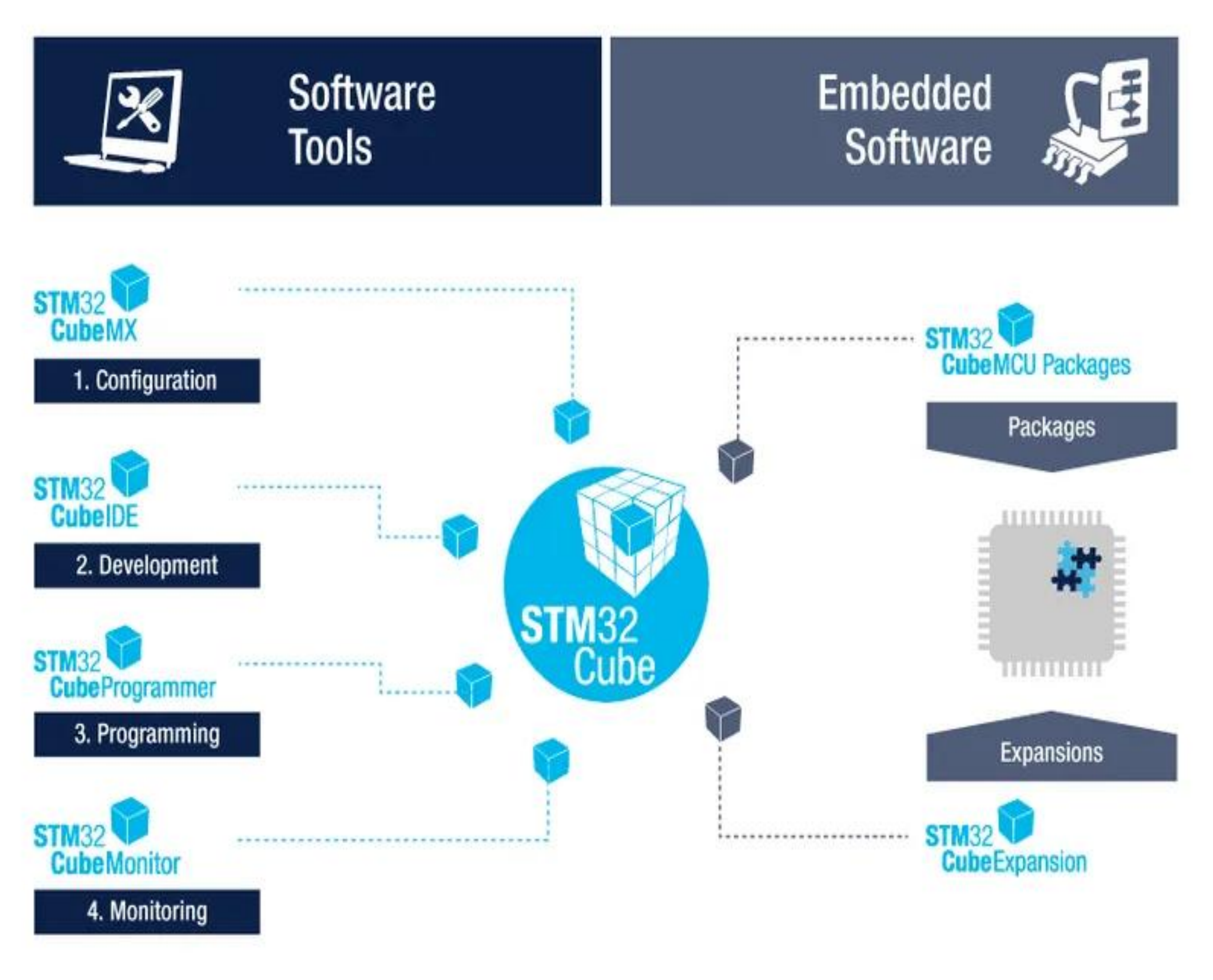
are specific to peripherals, allowing fine-grained power control.

STM32 Cube IDE

STM32 Cube IDE

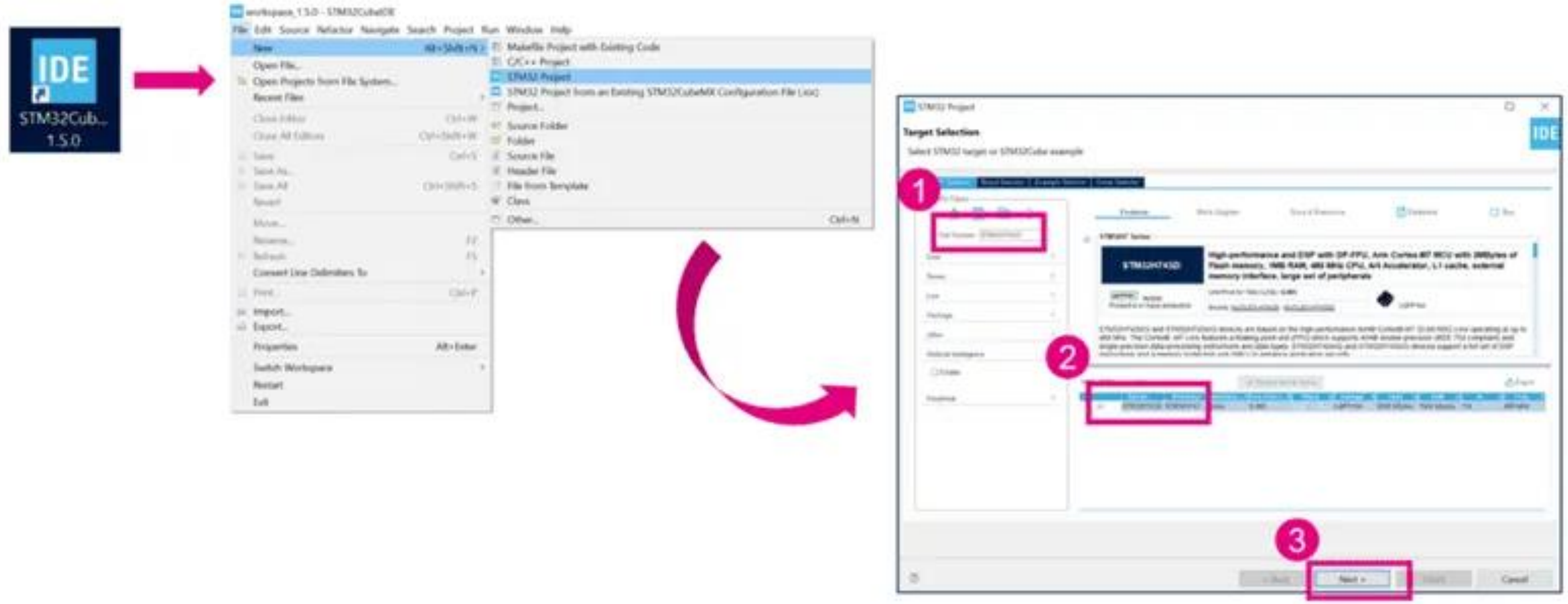
Why STM32 ?

- STM32 is widely used in the industry as MCUs are based on the ARM Cortex-M series (M0, M3, M4, M7, etc.), providing scalability, efficient processing, and strong ecosystem support.
- Strong Community & Open-Source Support as a Large developer and open-source communities provide ample resources, libraries, and third-party tools for STM32 development.

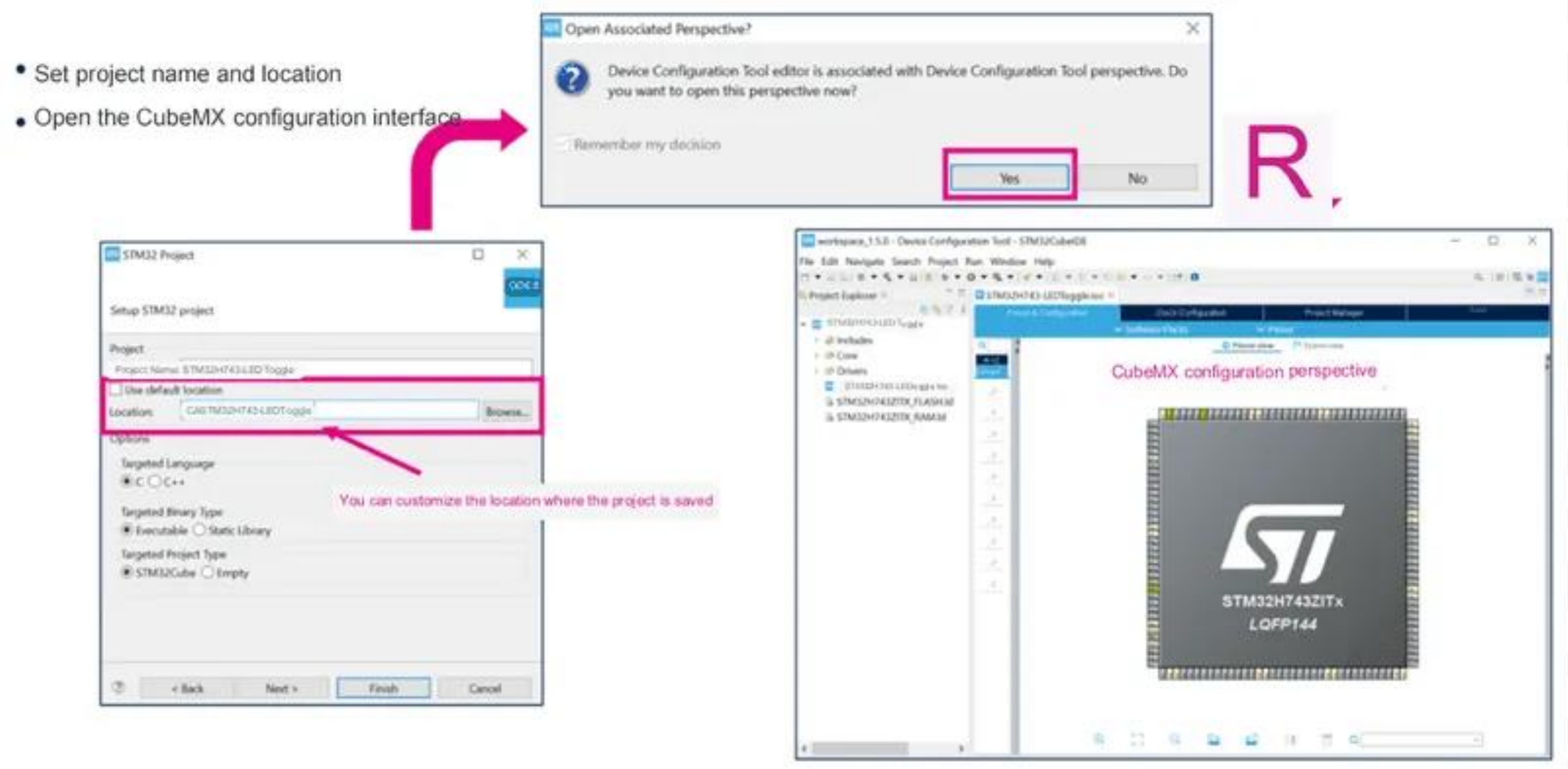


Create STM32 Project

- Create a new STM32CubeIDE project
- Select STM MCU model

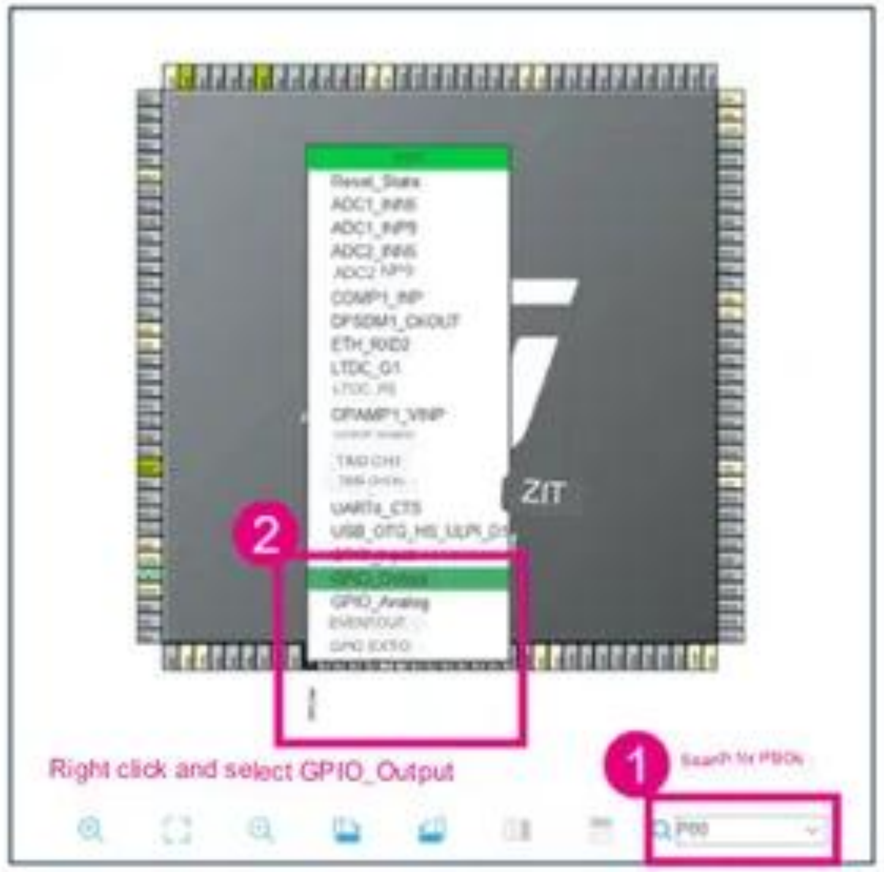


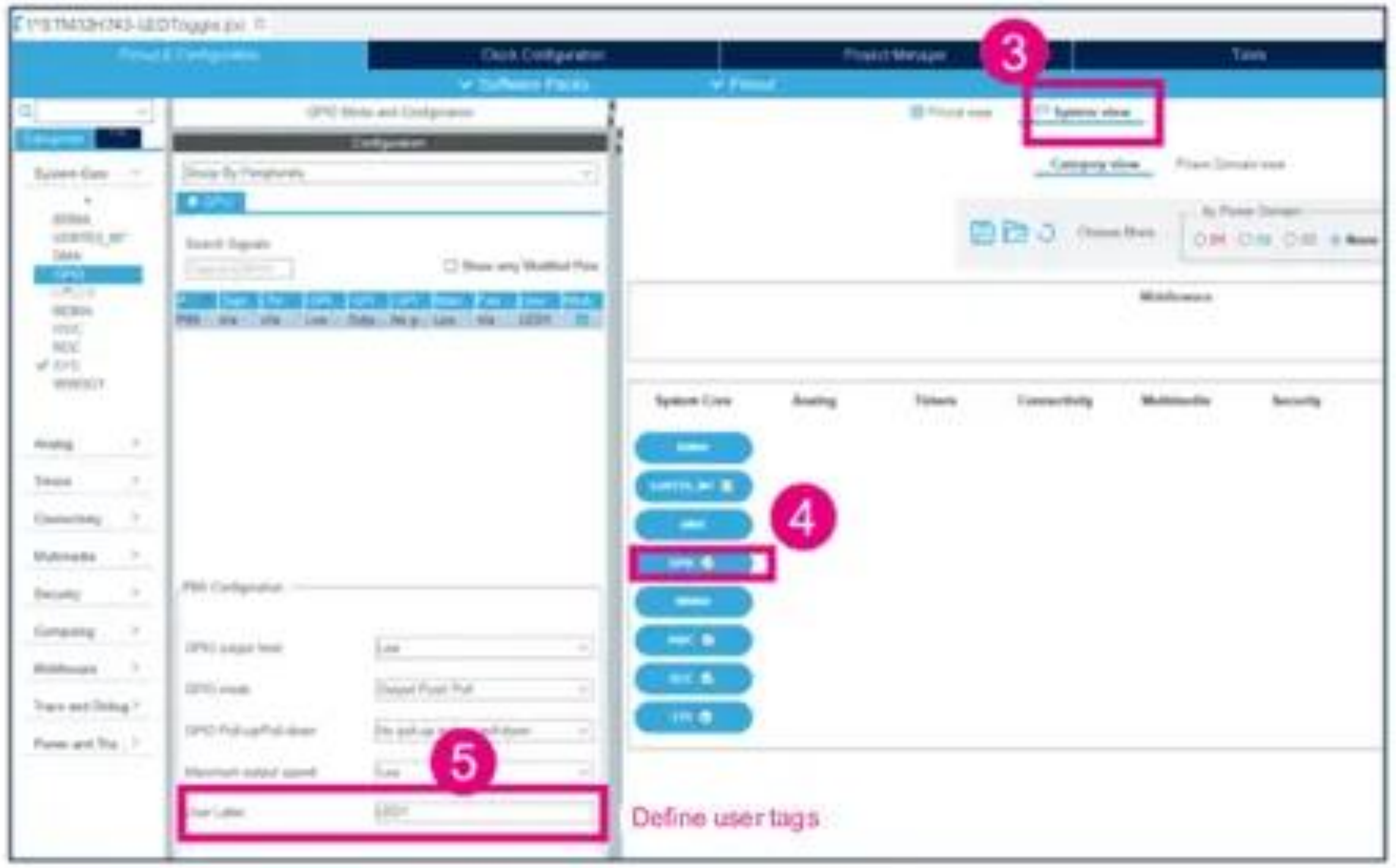
Configuration Perspective



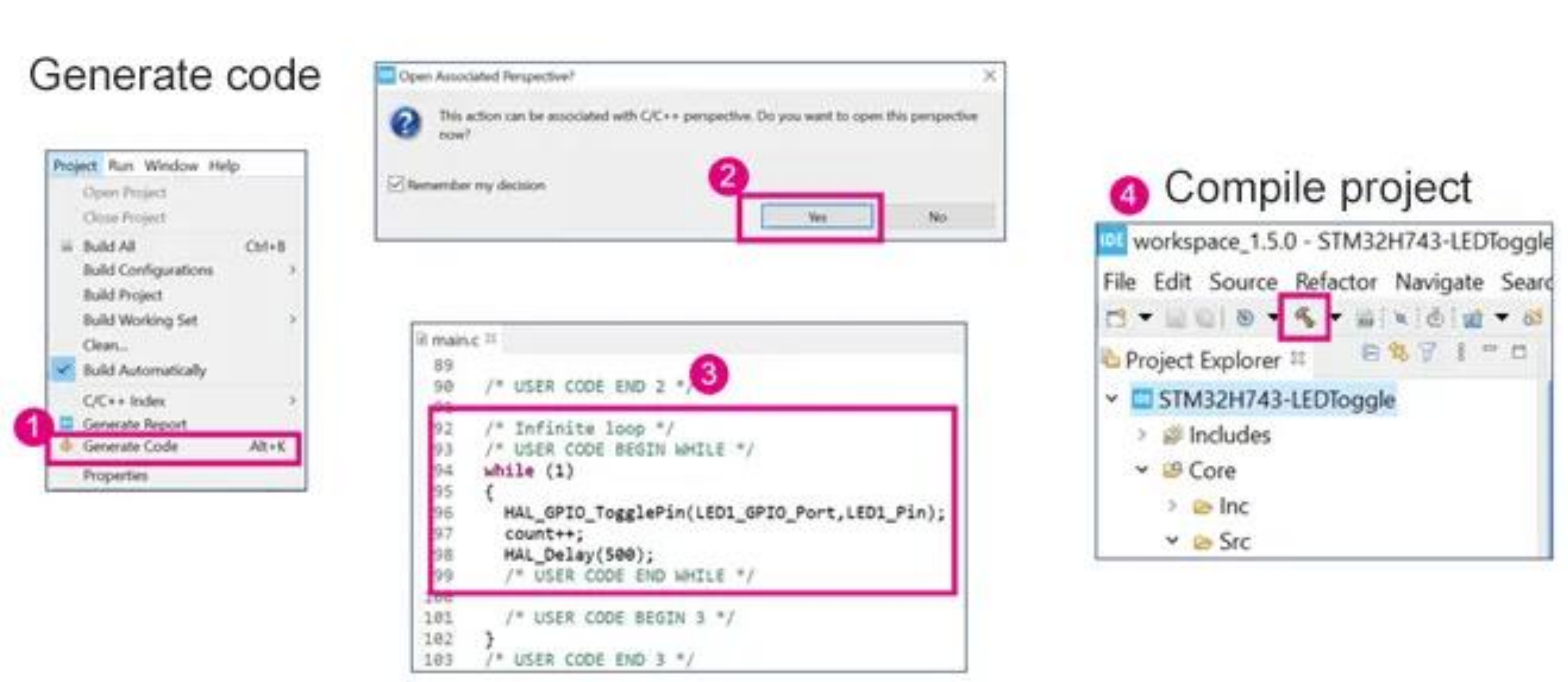
Configure GPIO

Configure GPIO port
PB0 is set as GPIO_Output
Defined in the code as LDE1

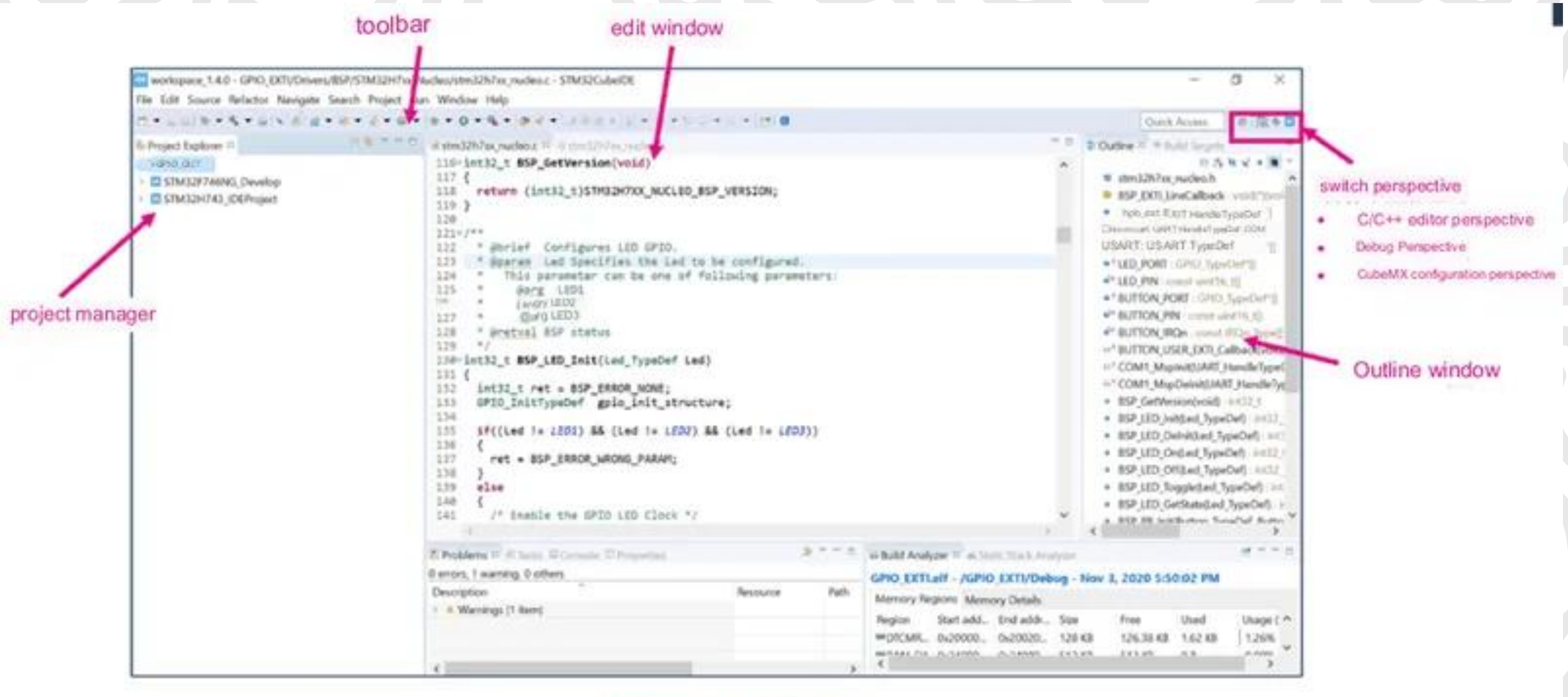




Generate Code

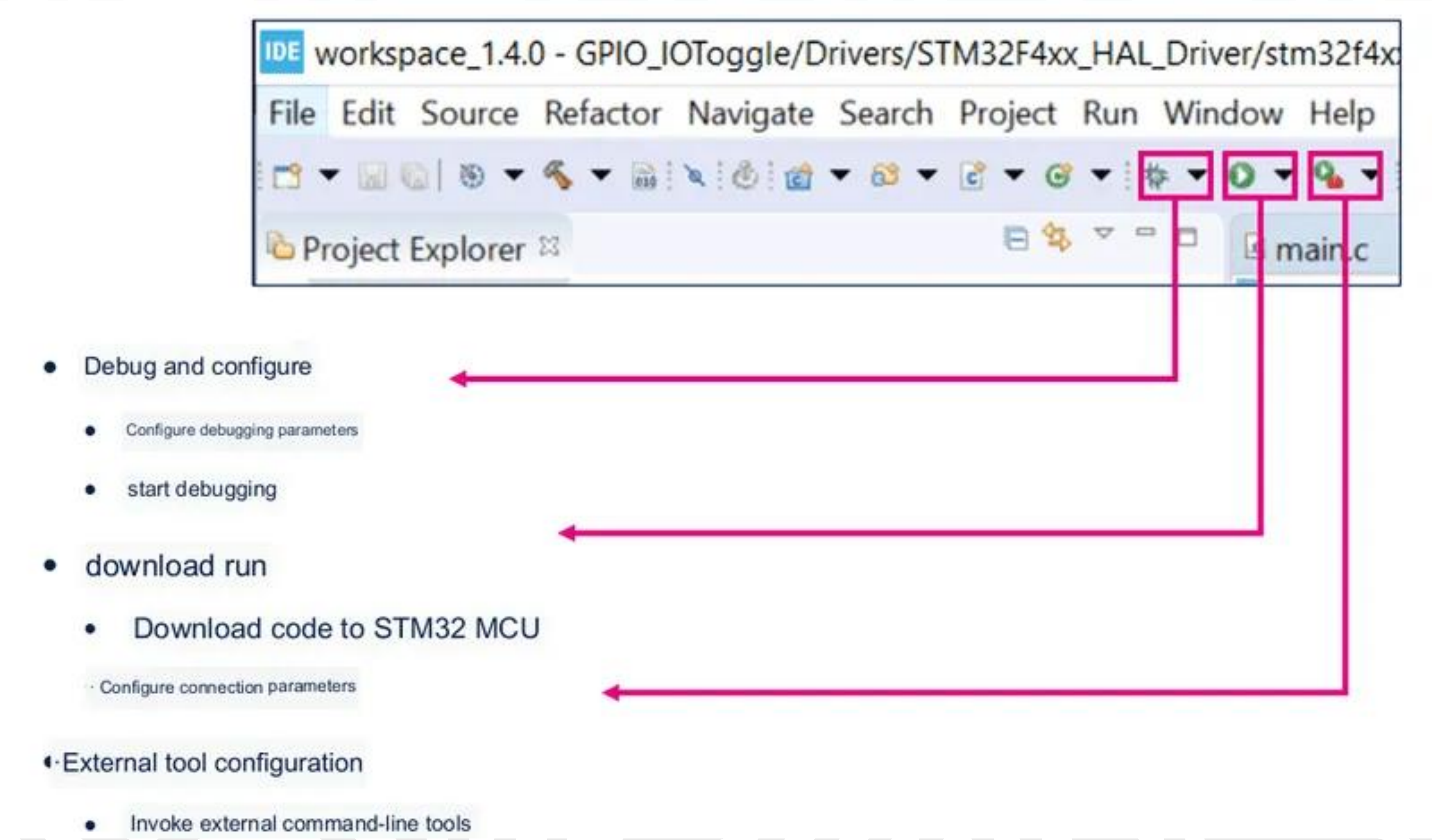


Code Perspective

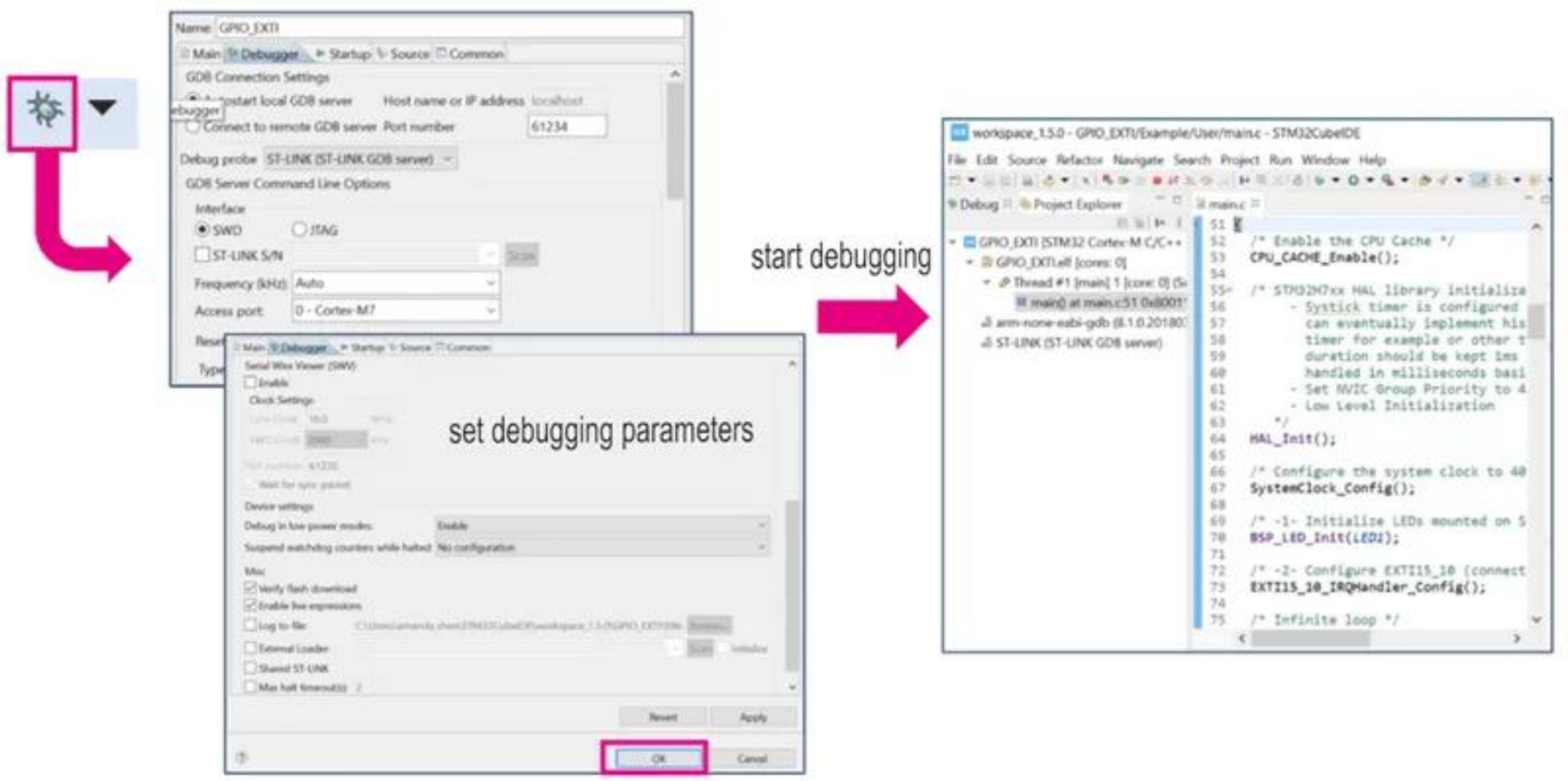


C/C++ editor perspective

Code Debugging



Start Debugging



Thank You

CAIRO UNIVERSITY
ECO-RACING TEAM

Refrences

STM32 Tutorial:

https://reversepcb.com/stm32cubeide/

Whole learning phase:

https://www.youtube.com/playlist?list=PLoiqjtgvXf9e2VJk8GWEXwECPM_7JRwkE

CAIRO UNIVERSITY ECO-RACING TEAM