POLITECNICO
DI MILANO

www.polimi.it

ARM – Cortex microcontrollers

SENSOR SYSTEMS

Dr. Federica Villa

Summary

- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

Beep: 05 – C-programming

Summary

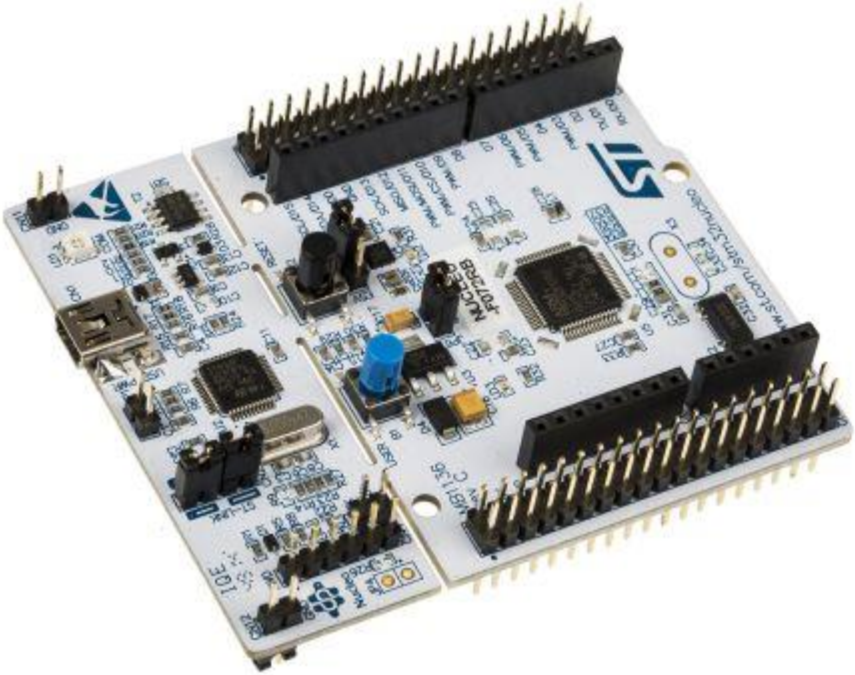
- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

Embedded systems definition

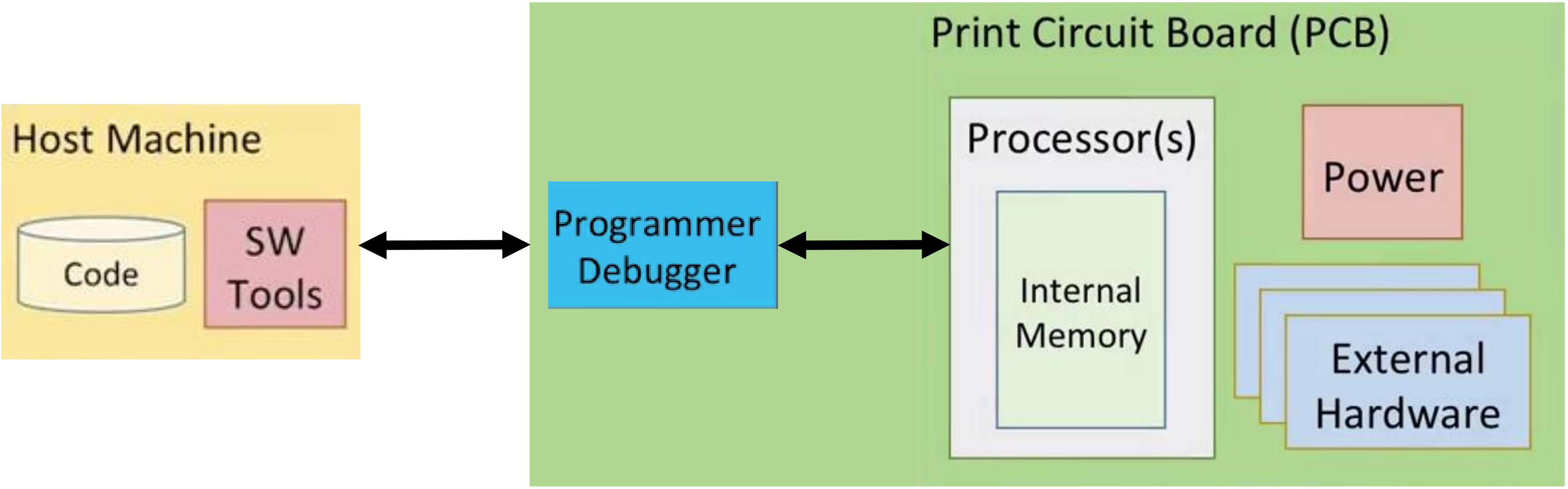
An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system.

Limited resources:

- Processing
- Memory
- Peripherals



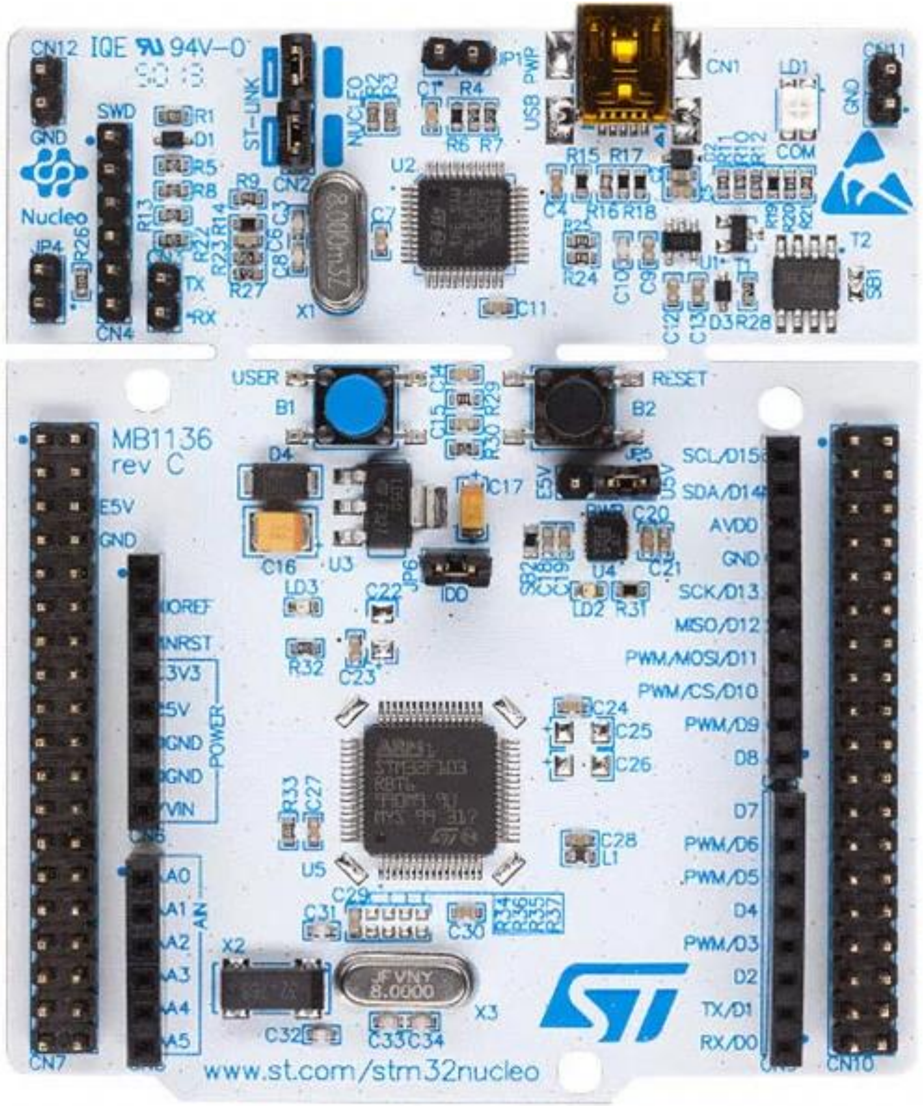
Development platform

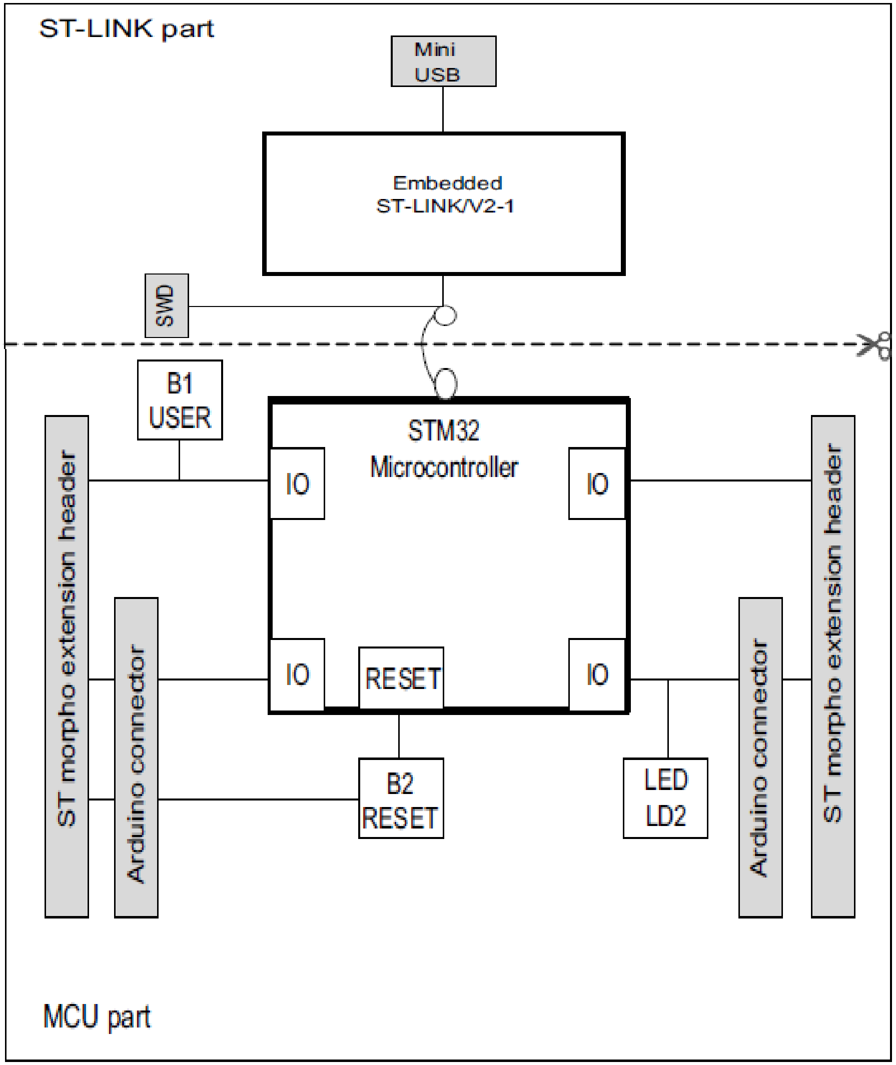


Software engineer's tools

Hardware (end-user)

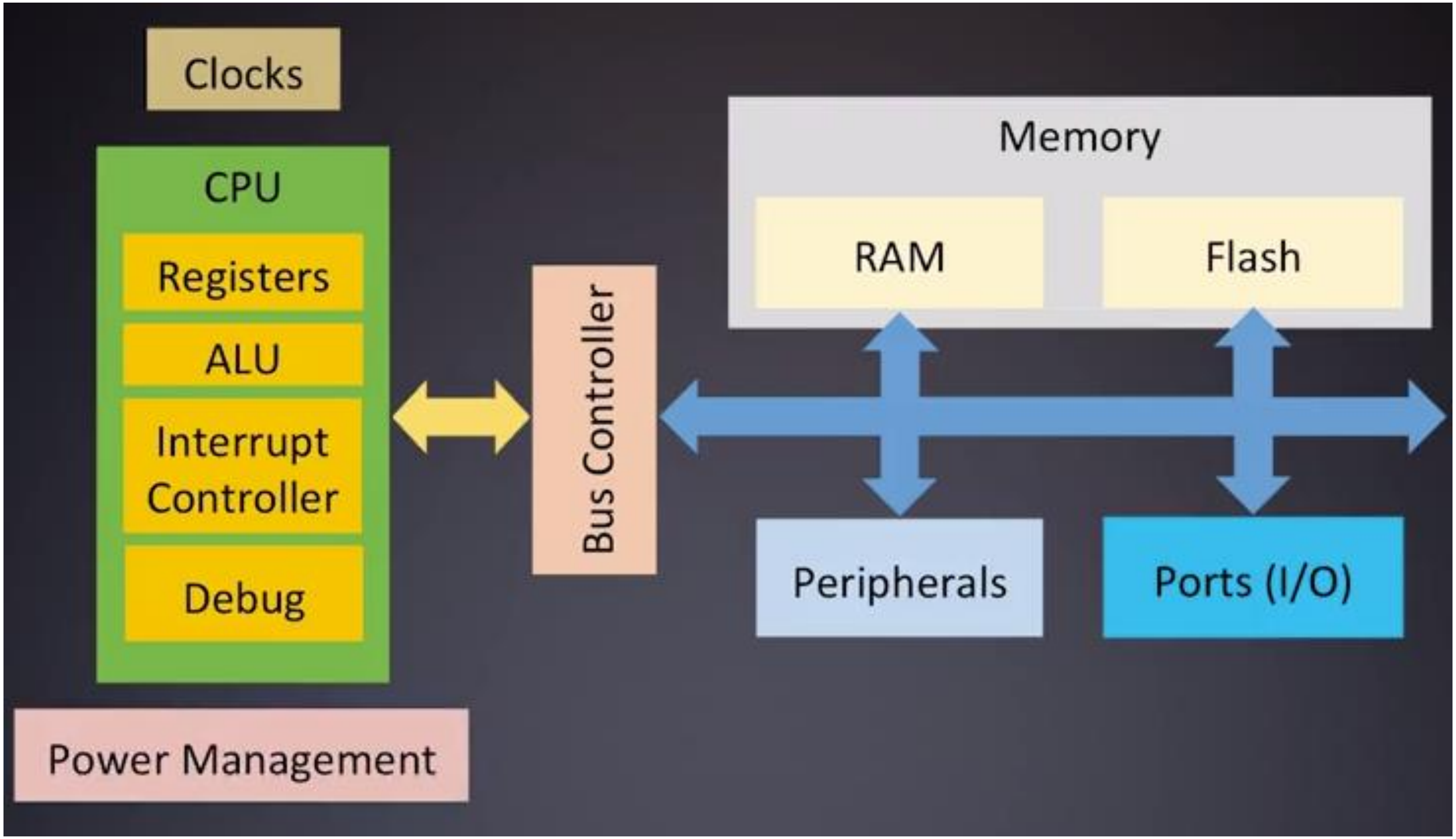
Example of a development platform





Microcontroller definition

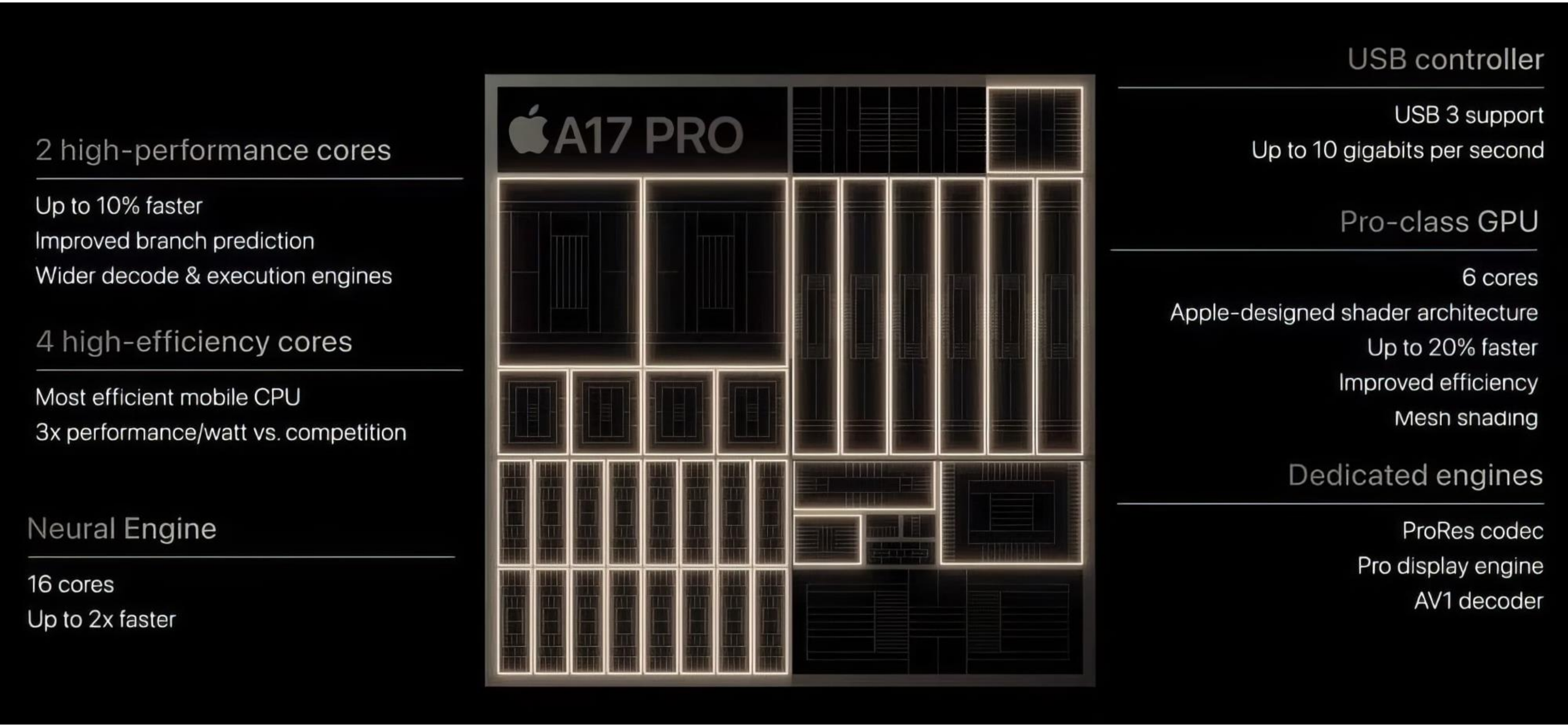
A microcontroller is a microprocessor
with added functionality
such as memory and peripheral hardware.



Microcontroller peripherals

- Communication:
  - I2C
  - UART
  - SPI
  - Ethernet
  - CAN
  - USB
- Timing
  - clock generator
  - timers
  - counters
  - Pulse Width Modulators (PWM)
  - Watchdog
- Analog signal processing:
  - ADC
  - DAC
  - voltage reference
  - analog comparators
  - analog sensors
- Input / Output
- Power control circuits
- …

A17 Pro



2 high-performance cores

Up to 10% faster
Improved branch prediction
Wider decode & execution engines

4 high-efficiency cores

Most efficient mobile CPU
3x performance/watt vs. competition

Neural Engine

16 cores
Up to 2x faster

USB controller

USB 3 support
Up to 10 gigabits per second

Pro-class GPU

6 cores
Apple-designed shader architecture
Up to 20% faster
Improved efficiency
Mesh shading

Dedicated engines

ProRes codec
Pro display engine
AV1 decoder

Why ARM?

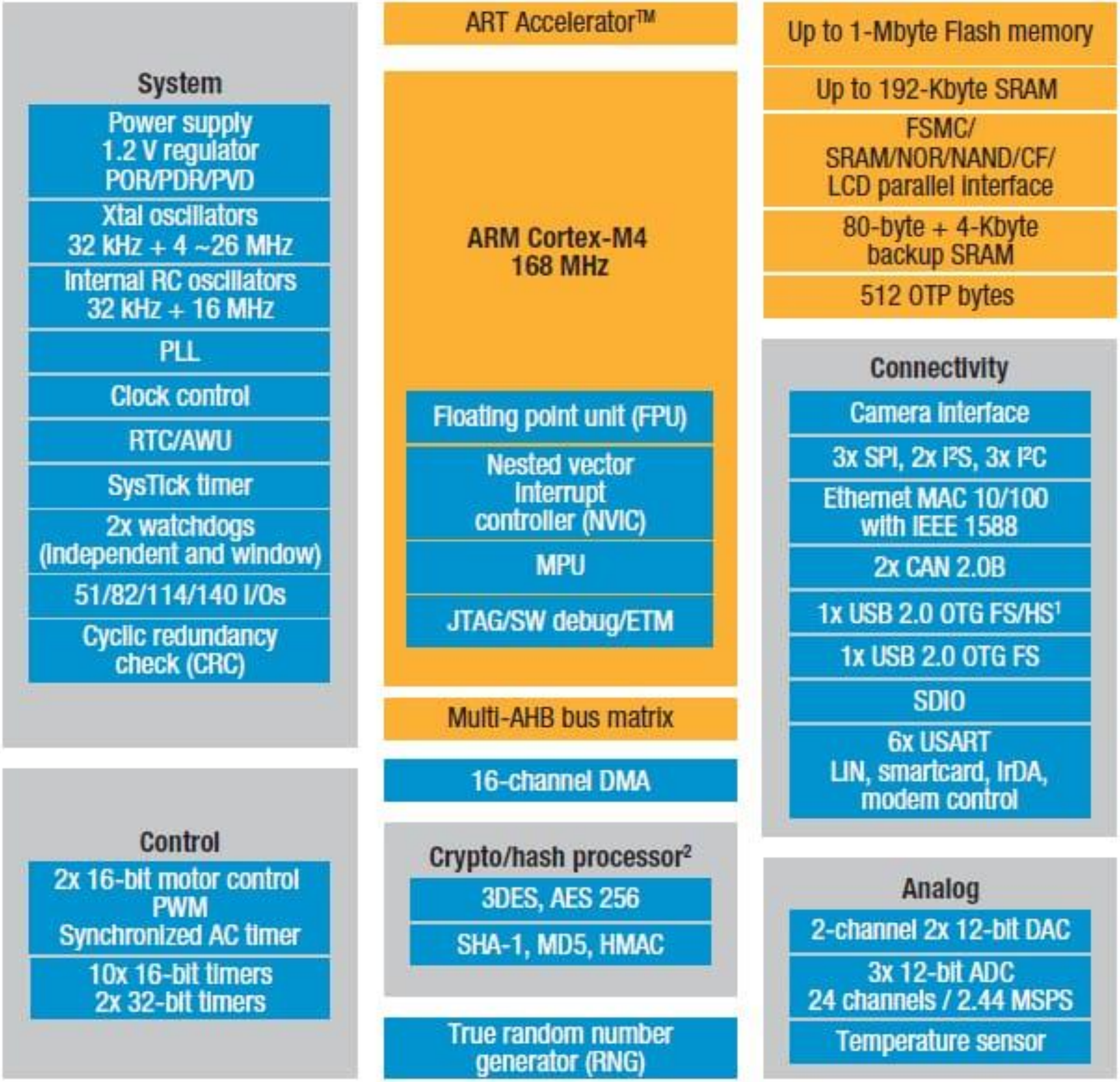
ST life.augmented
NXP
MICROCHIP
CYPRESS EMBEDDED IN TOMORROW™
TEXAS INSTRUMENTS
ANALOG DEVICES
EPSON EXCEED YOUR VISION
GHI electronics
infineon
maxim integrated™
SILICON LABS
VORAGO TECHNOLOGIES
TOSHIBA
ROHM SEMICONDUCTOR
WIZnet
zilog An IXYS Company

Arm Holdings provides the I.P. for the core processor

ARM – Cortex → 32 bit registers

STM32F4 Block diagram

yellow= ARM
gray=other suppliers (stm)



System

- Power supply 1.2 V regulator POR/PDR/PVD
- Xtal oscillators 32 kHz + 4 ~26 MHz
- Internal RC oscillators 32 kHz + 16 MHz
- PLL
- Clock control
- RTC/AWU
- SysTick timer
- 2x watchdogs (independent and window)
- 51/82/114/140 I/Os
- Cyclic redundancy check (CRC)

Control

- 2x 16-bit motor control PWM Synchronized AC timer
- 10x 16-bit timers 2x 32-bit timers

ART Accelerator™

ARM Cortex-M4 168 MHz

- Floating point unit (FPU)
- Nested vector interrupt controller (NVIC)
- MPU
- JTAG/SW debug/ETM

Multi-AHB bus matrix

16-channel DMA

Crypto/hash processor[2]

- 3DES, AES 256
- SHA-1, MD5, HMAC

True random number generator (RNG)

Up to 1-Mbyte Flash memory

Up to 192-Kbyte SRAM

FSMC/ SRAM/NOR/NAND/CF/ LCD parallel interface

80-byte + 4-Kbyte backup SRAM

512 OTP bytes

Connectivity

- Camera interface
- 3x SPI, 2x I^2S, 3x I^2C
- Ethernet MAC 10/100 with IEEE 1588
- 2x CAN 2.0B
- 1x USB 2.0 OTG FS/HS[1]
- 1x USB 2.0 OTG FS
- SDIO
- 6x USART LIN, smartcard, IrDA, modem control

Analog

- 2-channel 2x 12-bit DAC
- 3x 12-bit ADC 24 channels / 2.44 MSPS
- Temperature sensor

RISC vs CISC

ARM = Advanced RISC Machine

↓

Reduced Instruction Set Computer
(Very simple instructions that execute within a single cycle, at high clock speed)

vs

CISC = Complex Instruction Set Computer

(more complex instructions to complete a task in as few lines as possible)

RISC vs CISC: an example

Pick the ball command to a pet

CISC:

Pick the ball

RISC:

Track the toy

Pick the toy up

Get back to the owner

Hamd over the toy to the owner

How to execute 1800 + 1801

CISC:

ADD 1800, 1801

Memory locations

RISC:

Load X, 1800

Load Y, 1801

ADD X, Y

Store 1800, X

RISC vs CISC

Cisc

- Little work from the compiler to pass from high-level language into assembly (and few memory needed)

- Short codes

- More clock cycles

- Pipeline more difficult

- More transistors used to store instructions

Risc

- Long codes and more RAM needed (and it must also be fast)

- More general purpose hardware

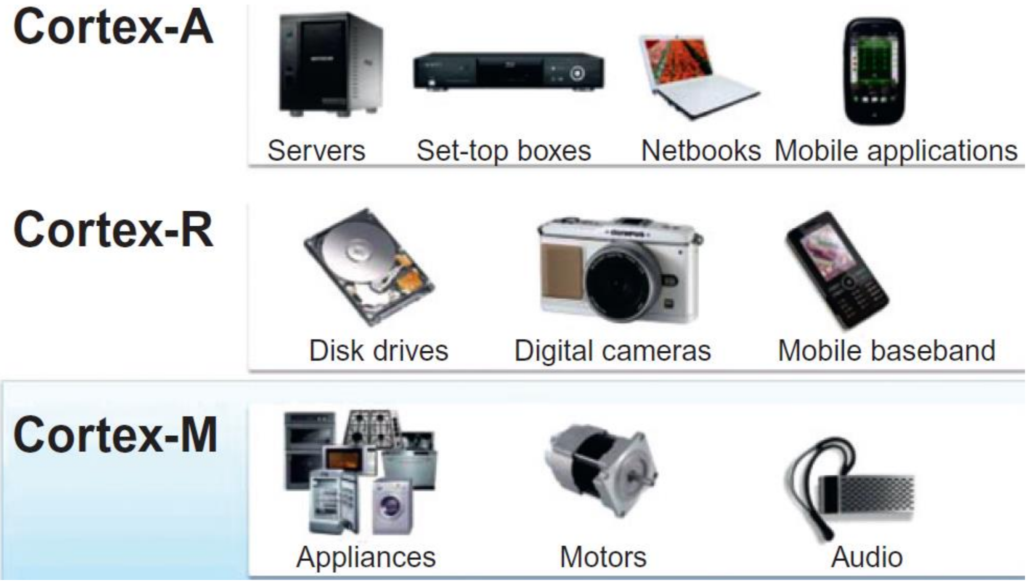
- One cycle per instruction (or even less)

- Pipeline easier

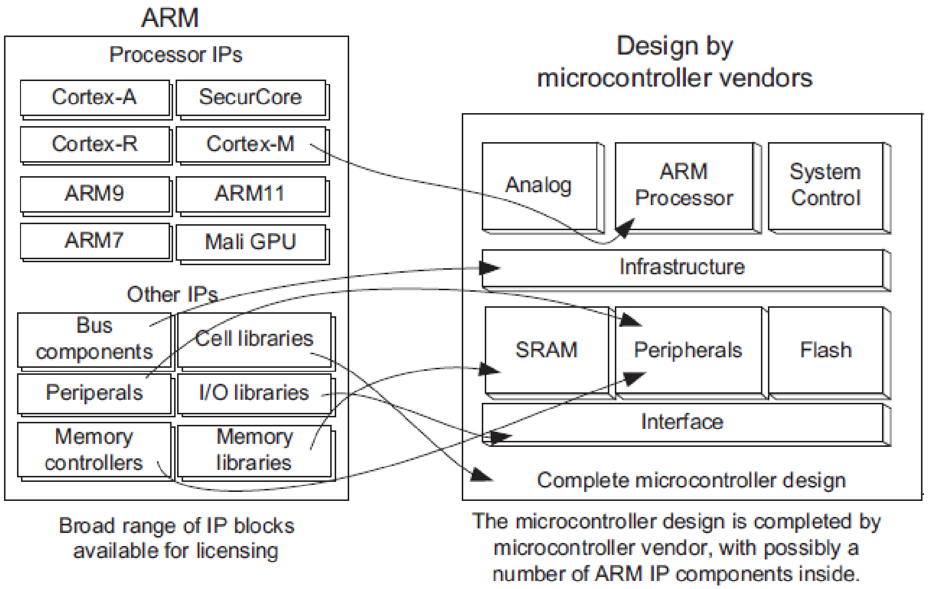
- More transistors used for register memories

ARM-Cortex families

Different «profiles»
- A: application (high-end)
- R: real time (more focus on timing performance)
- M: more wide-range and mainstream



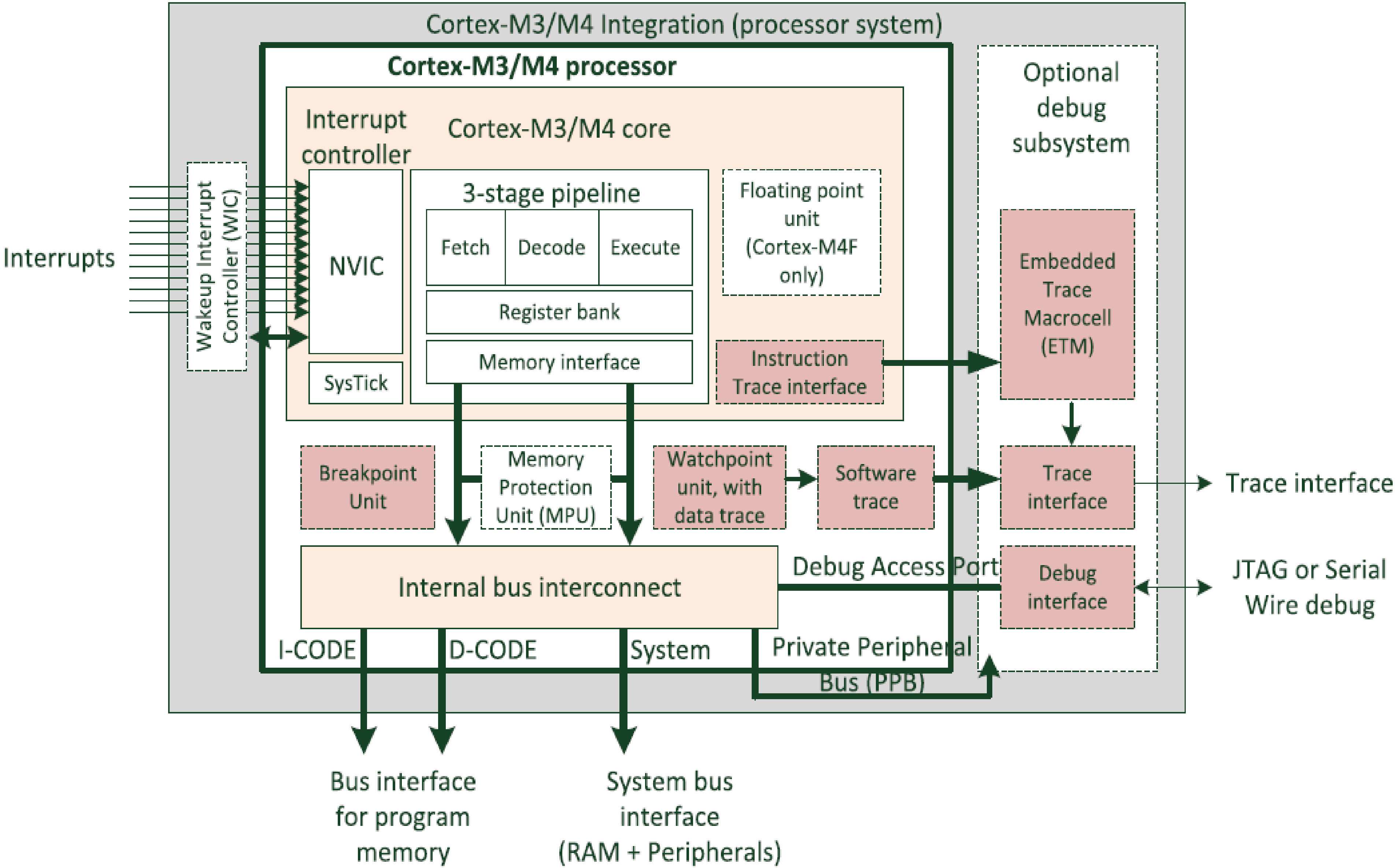
Business model:
Intellectual
Property
Licensing



Broad range of IP blocks available for licensing

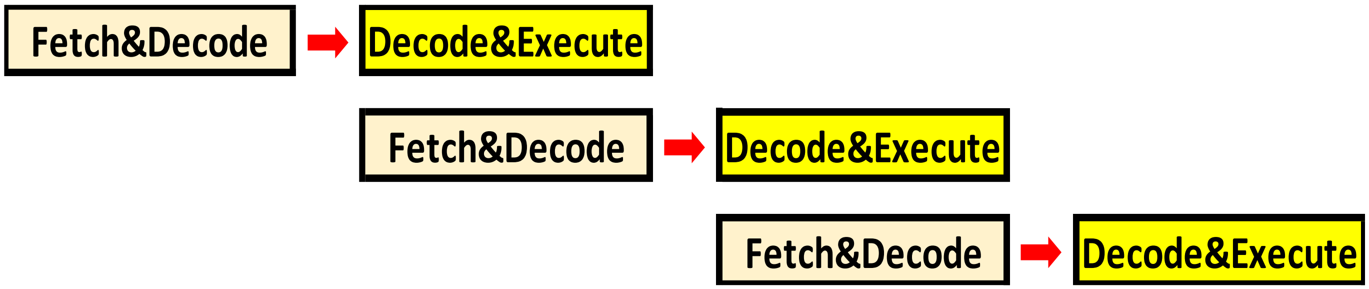
The microcontroller design is completed by microcontroller vendor, with possibly a number of ARM IP components inside.

ARM-Cortex M3 and M4 architectures

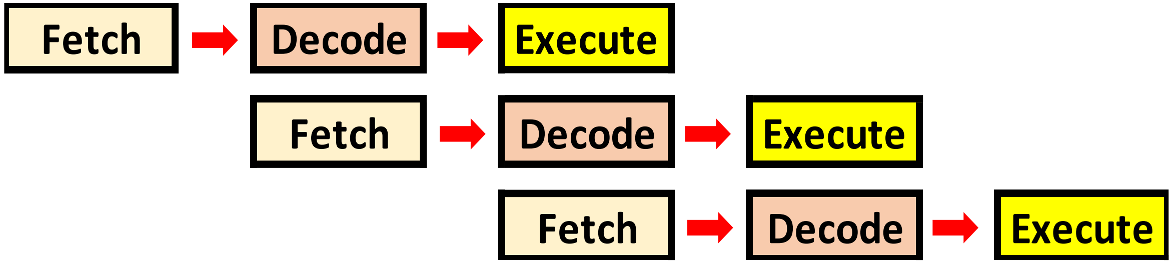

Cortex-M3/M4 Integration (processor system)
Cortex-M3/M4 processor
Interrupt controller
Cortex-M3/M4 core
Optional debug subsystem
Interrupts
Wakeup Interrupt Controller (WIC)
NVIC
SysTick
3-stage pipeline
Fetch
Decode
Execute
Register bank
Memory interface
Floating point unit (Cortex-M4F only)
Instruction Trace interface
Embedded Trace Macrocell (ETM)
Breakpoint Unit
Memory Protection Unit (MPU)
Watchpoint unit, with data trace
Software trace
Trace interface
Trace interface
Internal bus interconnect
Debug Access Port
Debug interface
JTAG or Serial Wire debug
I-CODE
D-CODE
System
Private Peripheral Bus (PPB)
Bus interface for program memory
System bus interface (RAM + Peripherals)

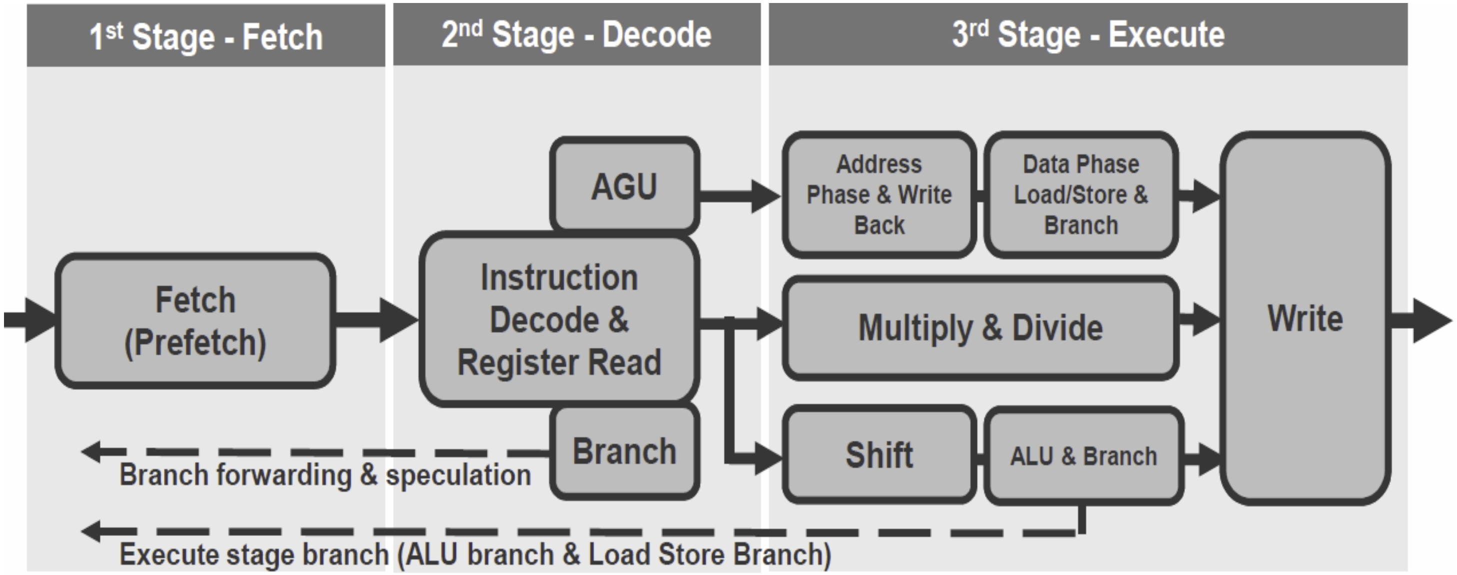
ARM-Cortex pipelining

2-level (M0+)



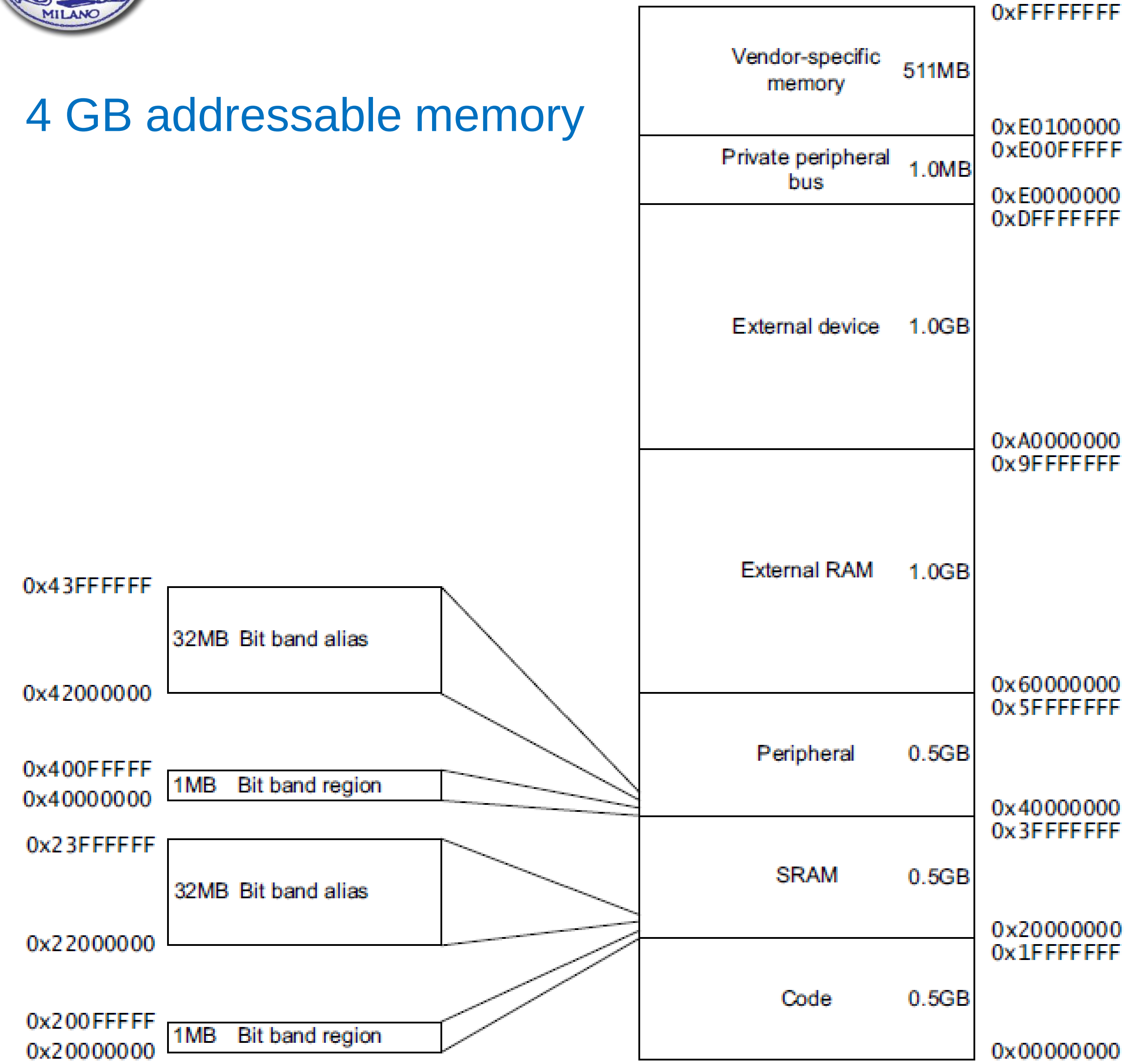
3-level (M3/4)




1st Stage - Fetch
2nd Stage - Decode
3rd Stage - Execute
Fetch (Prefetch)
AGU
Instruction Decode & Register Read
Branch
Address Phase & Write Back
Data Phase Load/Store & Branch
Multiply & Divide
Shift
ALU & Branch
Write
Branch forwarding & speculation
Execute stage branch (ALU branch & Load Store Branch)

ARM-Cortex M4 memory map

4 GB addressable memory



0xFFFFFFFF

Vendor-specific memory 511MB

0xE0100000

0xE00FFFFF

Private peripheral bus 1.0MB

0xE0000000

0xDFFFFFFF

External device 1.0GB

0xA0000000

0x9FFFFFFF

External RAM 1.0GB

0x60000000

0x5FFFFFFF

Peripheral 0.5GB

0x40000000

0x3FFFFFFF

SRAM 0.5GB

0x20000000

0x1FFFFFFF

Code 0.5GB

0x00000000

0x43FFFFFF

32MB Bit band alias

0x42000000

0x400FFFFF

1MB Bit band region

0x40000000

0x23FFFFFF

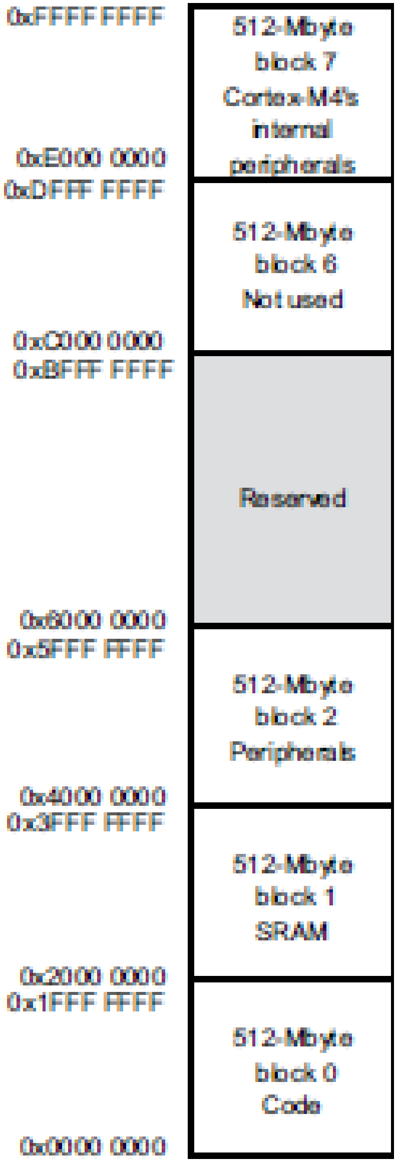
32MB Bit band alias

0x22000000

0x200FFFFF

1MB Bit band region

0x20000000



0xFFFF FFFF

512-Mbyte block 7 Cortex-M4's internal peripherals

0xE000 0000

0xDFFF FFFF

512-Mbyte block 6 Not used

0xC000 0000

0xBFFF FFFF

Reserved

0x6000 0000

0x5FFF FFFF

512-Mbyte block 2 Peripherals

0x4000 0000

0x3FFF FFFF

512-Mbyte block 1 SRAM

0x2000 0000

0x1FFF FFFF

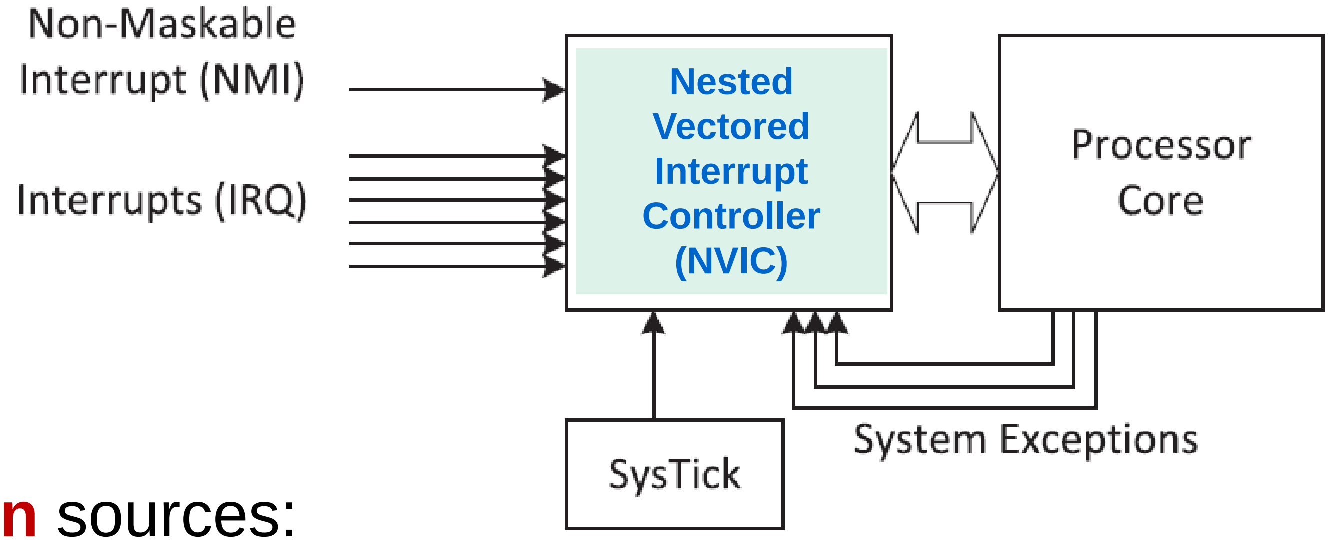
512-Mbyte block 0 Code

0x0000 0000

Summary

- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

ARM-Cortex M4 exceptions and interrupts



Exception sources:

Reset invoked on power up or warm reset, stops processor; restart is privileged Thread mode execution.

NMI interrupt by a peripheral or by software, highest priority exception other than reset; permanently enabled.

HardFault occurs because of an error during exception processing.

MemManage because of a memory protection related fault, for both instruction and data transactions.

BusFault because of a memory related fault for instruction or data transaction, e.g. error detected on a memory bus.

UsageFault related to instruction faults, such as: undefined instruction; illegal unaligned access; division by zero; etc..

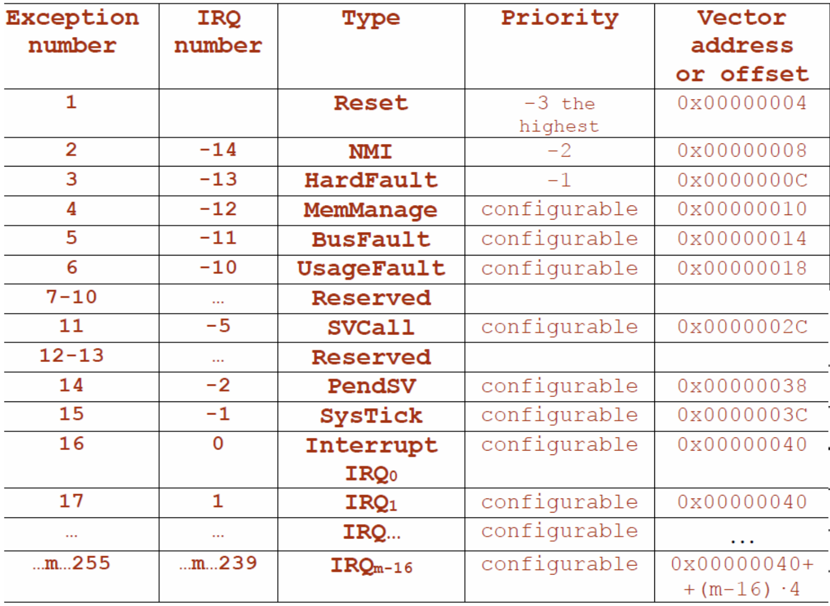
SVCall SuperVisor Call triggered by the SVC instruction (e.g. to access OS kernel functions and device drivers).

PendSV interrupt-driven request for system-level service (e.g. used in OS for context switching).

SysTick generated by the system timer when it reaches zero (e.g. used as system tick for Real-Time Clock).

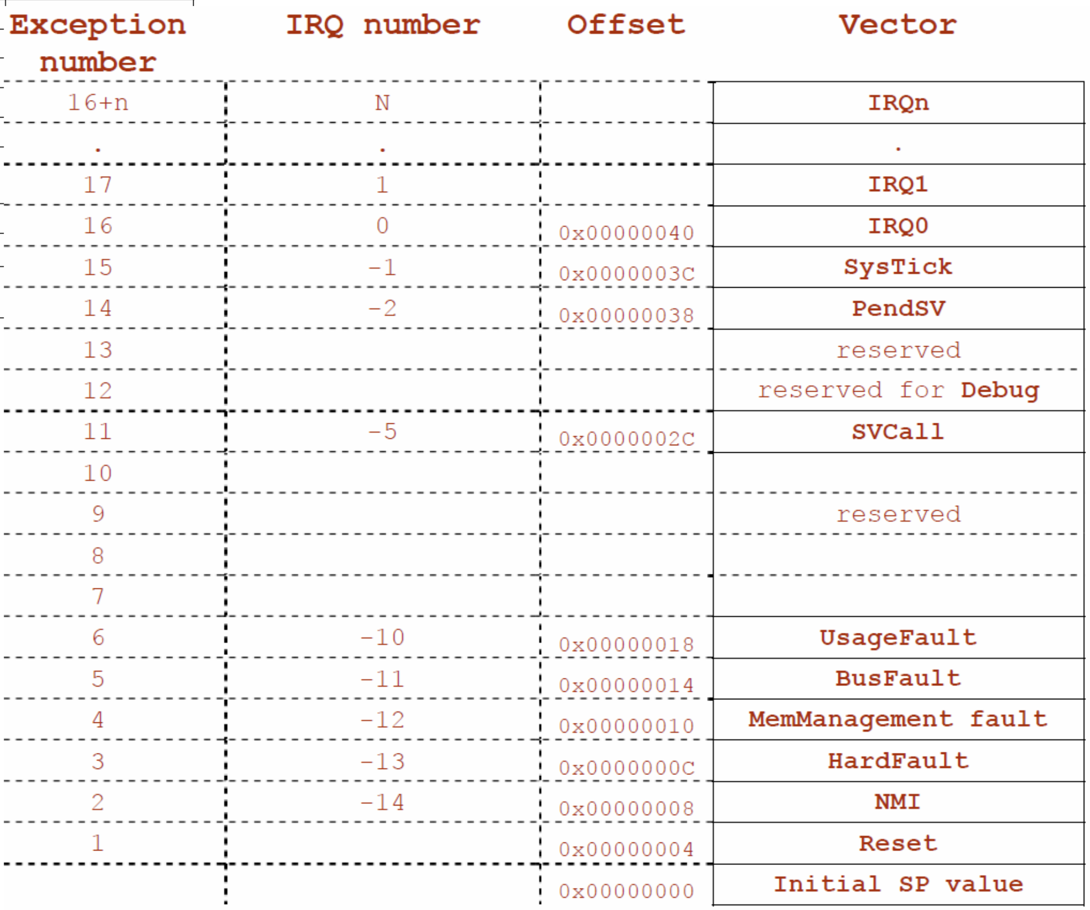
Interrupt IRQ Interrupt Servicing Request signalled by a peripheral.

ARM-Cortex M4 interrupt vector



Exception number	IRQ number	Type	Priority	Vector address or offset	Activation
1		Reset	-3 the highest	0x00000004	Asynchronous
2	-14	NMI	-2	0x00000008	Asynchronous
3	-13	HardFault	-1	0x0000000C	
4	-12	MemManage	configurable	0x00000010	Synchronous
5	-11	BusFault	configurable	0x00000014	Synch/Asynch
6	-10	UsageFault	configurable	0x00000018	Synchronous
7-10	…	Reserved			
11	-5	SVCall	configurable	0x0000002C	
12-13	…	Reserved			
14	-2	PendSV	configurable	0x00000038	
15	-1	SysTick	configurable	0x0000003C	
16	0	Interrupt IRQ$_0$	configurable	0x00000040	
17	1	IRQ$_1$	configurable	0x00000040	
…	…	IRQ…	configurable	…	
…m…255	…m…239	IRQ$_{m-16}$	configurable	0x00000040+ +(m-16)·4	

Exception priorities



Exception number	IRQ number	Offset	Vector
16+n	N		IRQn
.	.		.
17	1		IRQ1
16	0	0x00000040	IRQ0
15	-1	0x0000003C	SysTick
14	-2	0x00000038	PendSV
13			reserved
12			reserved for Debug
11	-5	0x0000002C	SVCall
10			reserved
9			
8			
7			
6	-10	0x00000018	UsageFault
5	-11	0x00000014	BusFault
4	-12	0x00000010	MemManagement fault
3	-13	0x0000000C	HardFault
2	-14	0x00000008	NMI
1		0x00000004	Reset
		0x00000000	Initial SP value

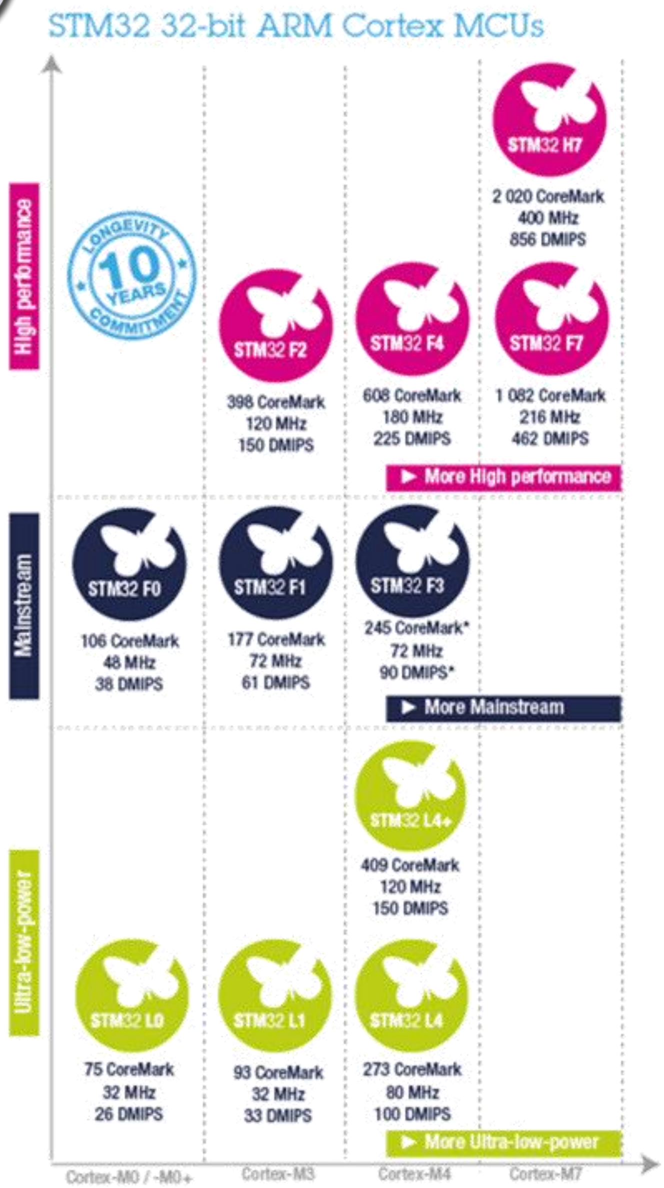
Vector Table

(start address for all exception handlers)

Summary

- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

STM32 Family



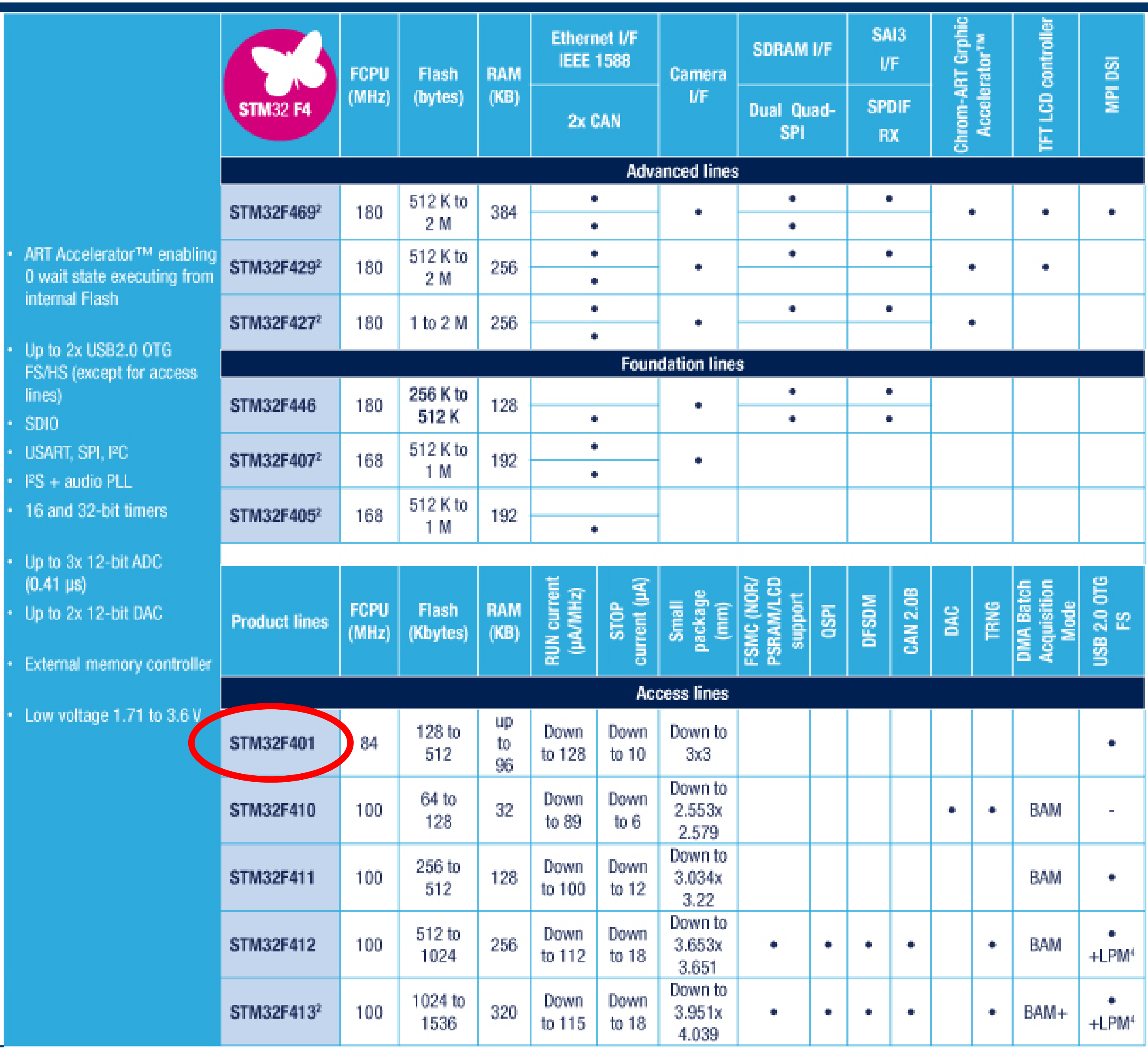
STM32F4:

- High performance
- up to 180 MHz clock

STM32F4

STM32F4xx:

- Cortex M4 core



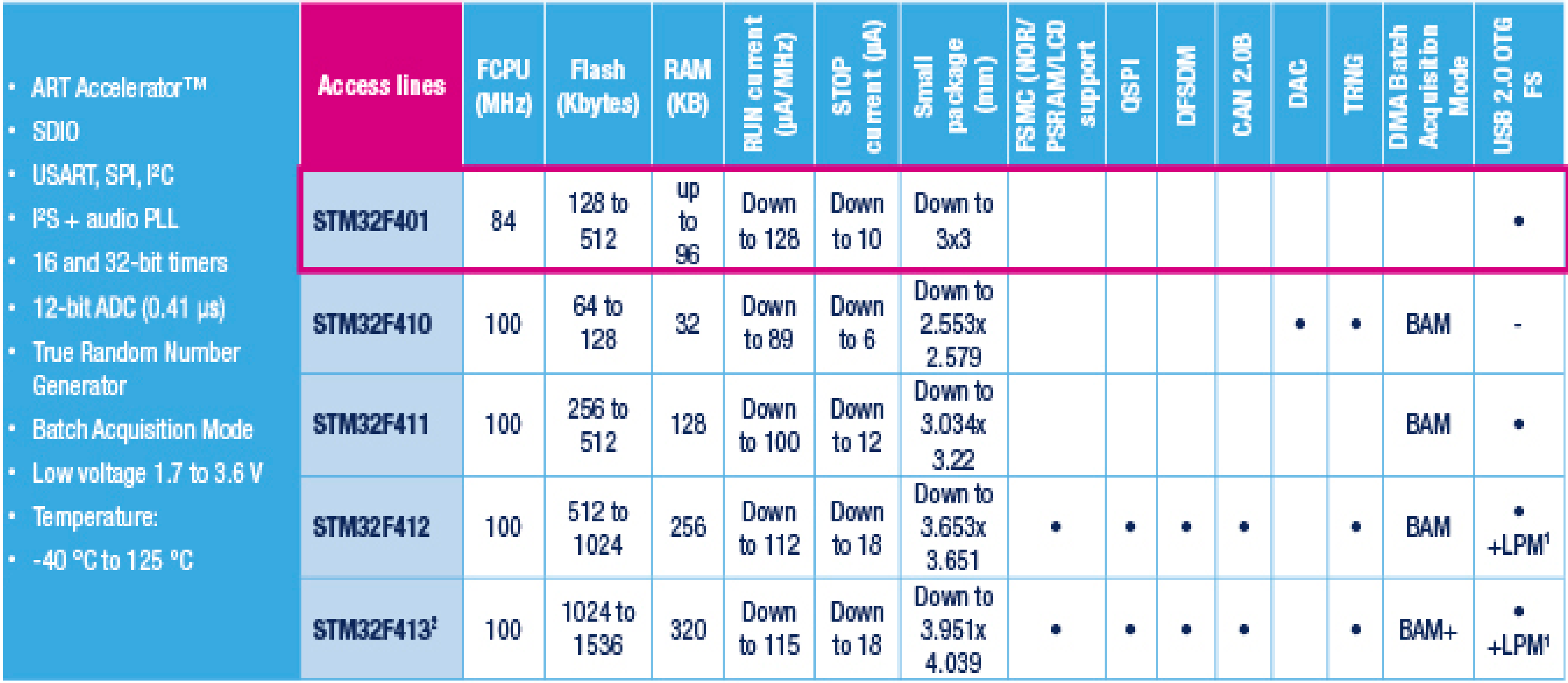
- ART Accelerator™ enabling 0 wait state executing from internal Flash
- Up to 2x USB2.0 OTG FS/HS (except for access lines)
- SDIO
- USART, SPI, I^2C
- I^2S + audio PLL
- 16 and 32-bit timers
- Up to 3x 12-bit ADC (0.41 µs)
- Up to 2x 12-bit DAC
- External memory controller
- Low voltage 1.71 to 3.6 V

STM32 F4	FCPU (MHz)	Flash (bytes)	RAM (KB)	Ethernet I/F IEEE 1588	Camera I/F	SDRAM I/F	SAI3 I/F	Chrom-ART Grphic Accelerator™	TFT LCD controller	MPI DSI
				2x CAN		Dual Quad-SPI	SPDIF RX			
Advanced lines										
STM32F469[2]	180	512 K to 2 M	384	• •	•	• •	•	•	•	•
STM32F429[2]	180	512 K to 2 M	256	• •	•	•	•	•	•	
STM32F427[2]	180	1 to 2 M	256	• •	•	•	•	•		
Foundation lines										
STM32F446	180	256 K to 512 K	128	•	•	• •	• •			
STM32F407[2]	168	512 K to 1 M	192	• •	•					
STM32F405[2]	168	512 K to 1 M	192	•						

Product lines	FCPU (MHz)	Flash (Kbytes)	RAM (KB)	RUN current (µA/MHz)	STOP current (µA)	Small package (mm)	FSMC (NOR/ PSRAM/LCD support	QSPI	DFSDM	CAN 2.0B	DAC	TRNG	DMA Batch Acquisition Mode	USB 2.0 OTG FS
Access lines														
STM32F401	84	128 to 512	up to 96	Down to 128	Down to 10	Down to 3x3								•
STM32F410	100	64 to 128	32	Down to 89	Down to 6	Down to 2.553x 2.579					•	•	BAM	-
STM32F411	100	256 to 512	128	Down to 100	Down to 12	Down to 3.034x 3.22							BAM	•
STM32F412	100	512 to 1024	256	Down to 112	Down to 18	Down to 3.653x 3.651	•	•	•	•		•	BAM	• +LPM[4]
STM32F413[2]	100	1024 to 1536	320	Down to 115	Down to 18	Down to 3.951x 4.039	•	•	•	•		•	BAM+	• +LPM[4]

STM32F401

STM32F4xx Access line: the entry-level microcontrollers

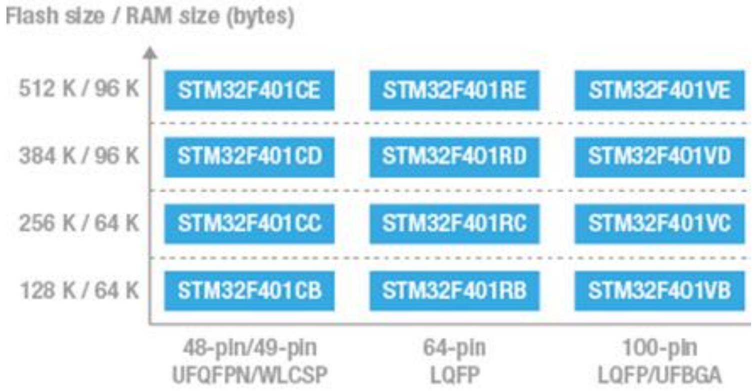


- ART Accelerator™
- SDIO
- USART, SPI, I²C
- I²S + audio PLL
- 16 and 32-bit timers
- 12-bit ADC (0.41 µs)
- True Random Number Generator
- Batch Acquisition Mode
- Low voltage 1.7 to 3.6 V
- Temperature:
- -40 °C to 125 °C

Access lines	FCPU (MHz)	Flash (Kbytes)	RAM (KB)	RUN current (µA/MHz)	STOP current (µA)	Small package (mm)	FSMC (NOR/ PSRAM/LCD support	QSPI	DFSDM	CAN 2.0B	DAC	TRNG	DMA Batch Acquisition Mode	USB 2.0 OTG FS
STM32F401	84	128 to 512	up to 96	Down to 128	Down to 10	Down to 3x3								•
STM32F410	100	64 to 128	32	Down to 89	Down to 6	Down to 2.553x 2.579					•	•	BAM	-
STM32F411	100	256 to 512	128	Down to 100	Down to 12	Down to 3.034x 3.22							BAM	•
STM32F412	100	512 to 1024	256	Down to 112	Down to 18	Down to 3.653x 3.651	•	•	•	•		•	BAM	• +LPM[1]
STM32F413[2]	100	1024 to 1536	320	Down to 115	Down to 18	Down to 3.951x 4.039	•	•	•	•		•	BAM+	• +LPM[1]

STM32F401RE

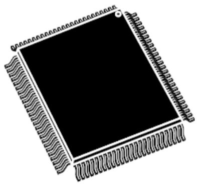
STM32F401



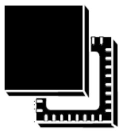
Flash size / RAM size (bytes)

	48-pin/49-pin UFQFPN/WLCSP	64-pin LQFP	100-pin LQFP/UFBGA
512 K / 96 K	STM32F401CE	STM32F401RE	STM32F401VE
384 K / 96 K	STM32F401CD	STM32F401RD	STM32F401VD
256 K / 64 K	STM32F401CC	STM32F401RC	STM32F401VC
128 K / 64 K	STM32F401CB	STM32F401RB	STM32F401VB

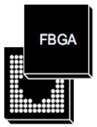
WLCSP49
(3.06 x 3.06 mm)



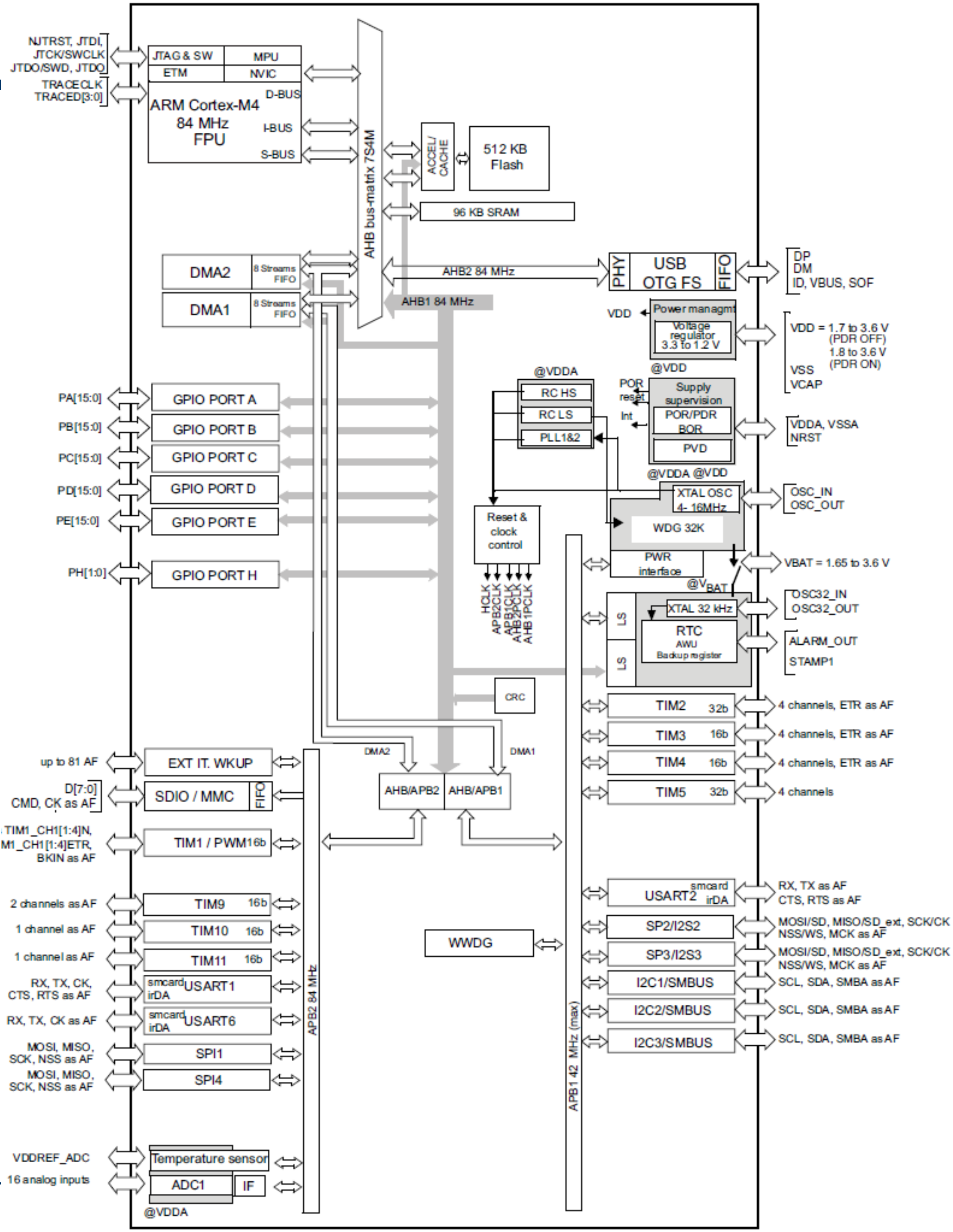
LQFP100 (14 × 14 mm
LQFP64 (10 × 10 mm)



UFQFPN48
(7 × 7 mm)



UFBGA100
(7 × 7 mm)



Clocks

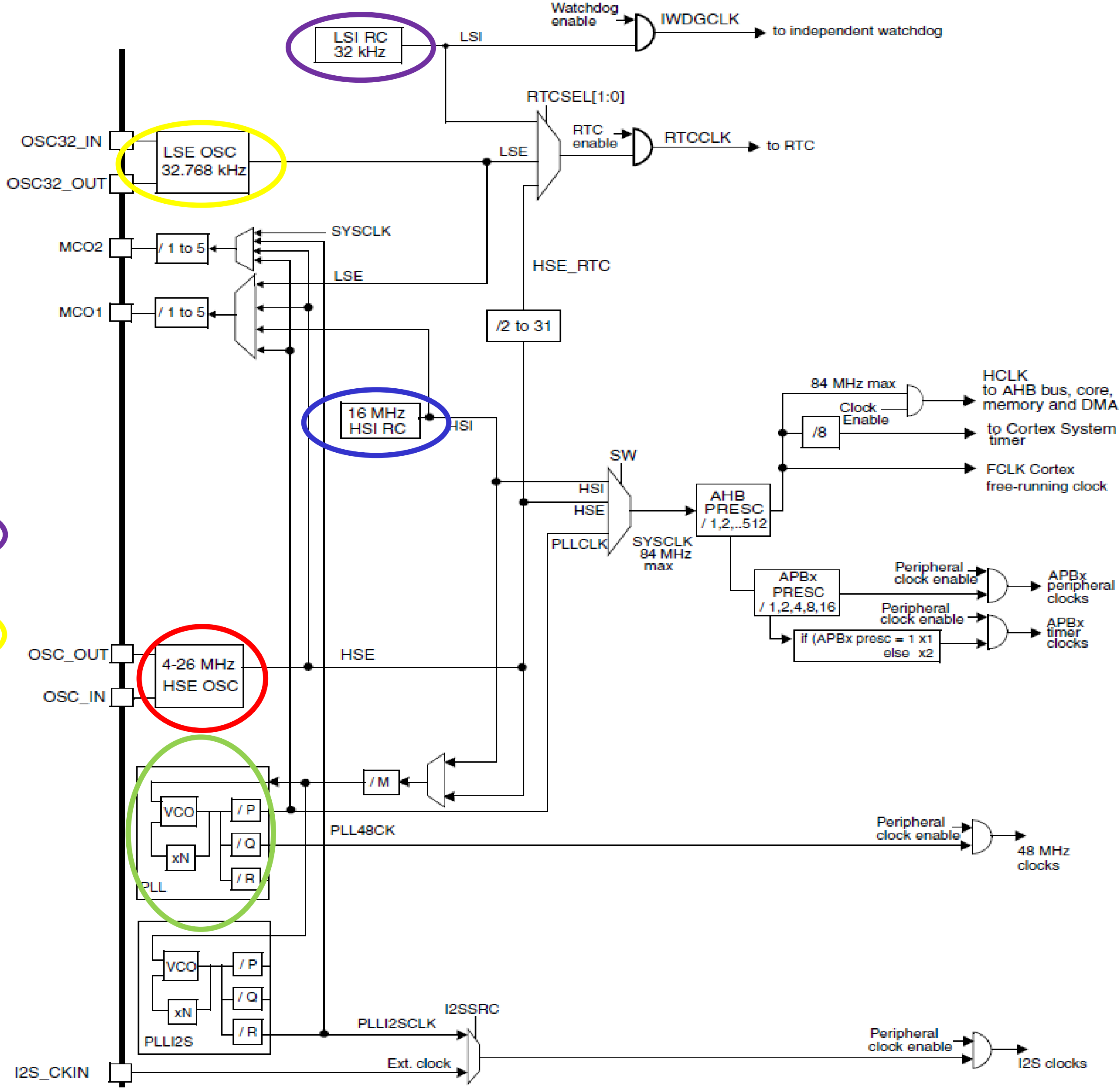
3 clock sources for system clock (SYSCLK):

- HSI oscillator clock
- HSE oscillator clock
- main PLL clock.

2 secondary clock sources:

- a 32 kHz low-speed internal RC (LSI RC)
- a 32.768 kHz low-speed external crystal (LSE crystal)

Each source can be switched on/off independently when not used



Timer modules

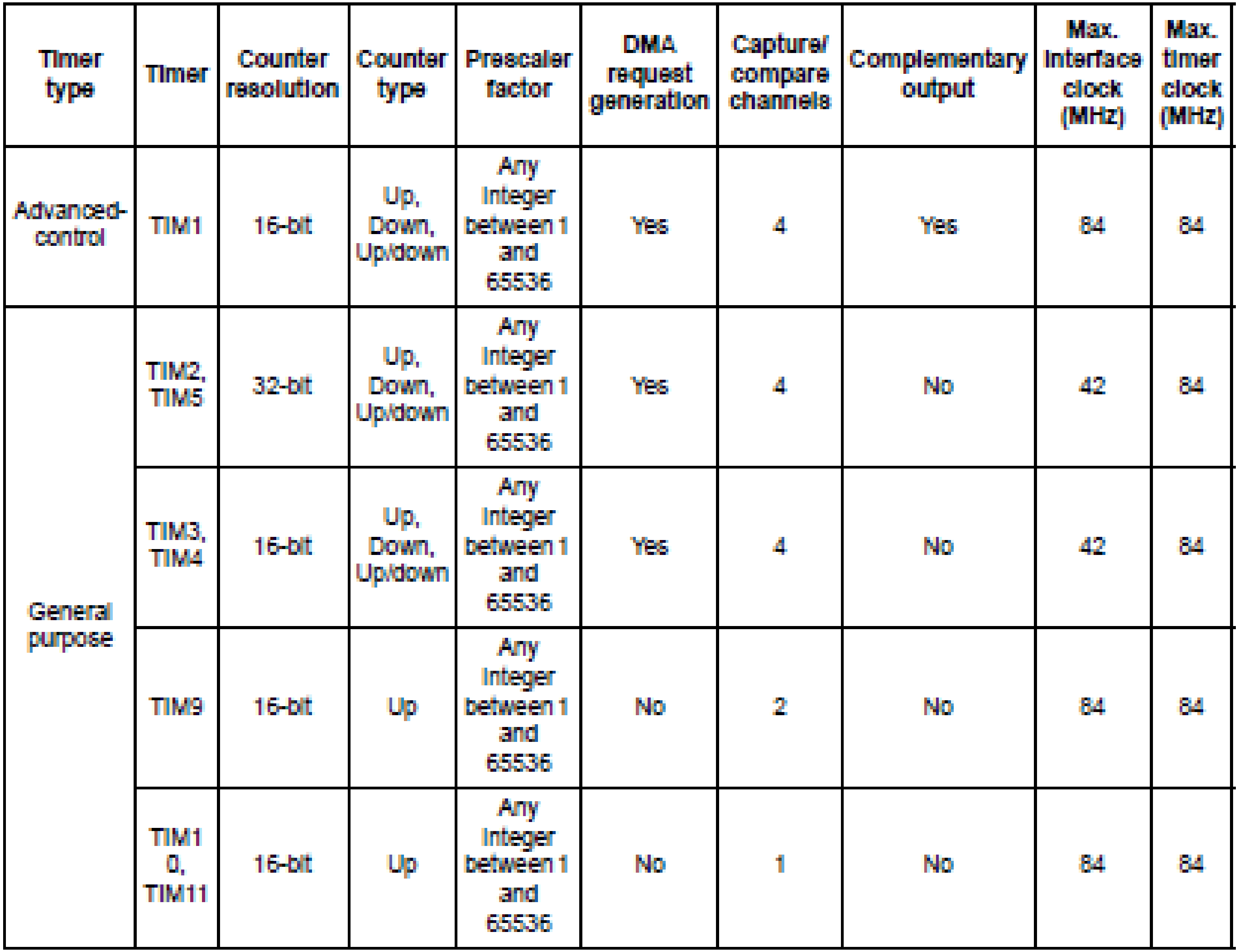
Timer modules include:

- reference clock (internal or external)
- prescaler (reduce the clock frequency)
- counter
- comparison registers

Timer modules are used for:

- generating interrupts
- generating pulse trains
- driving Pulse Width Modulation (PWM) outputs
- measuring pulse length of input signals
- counting events

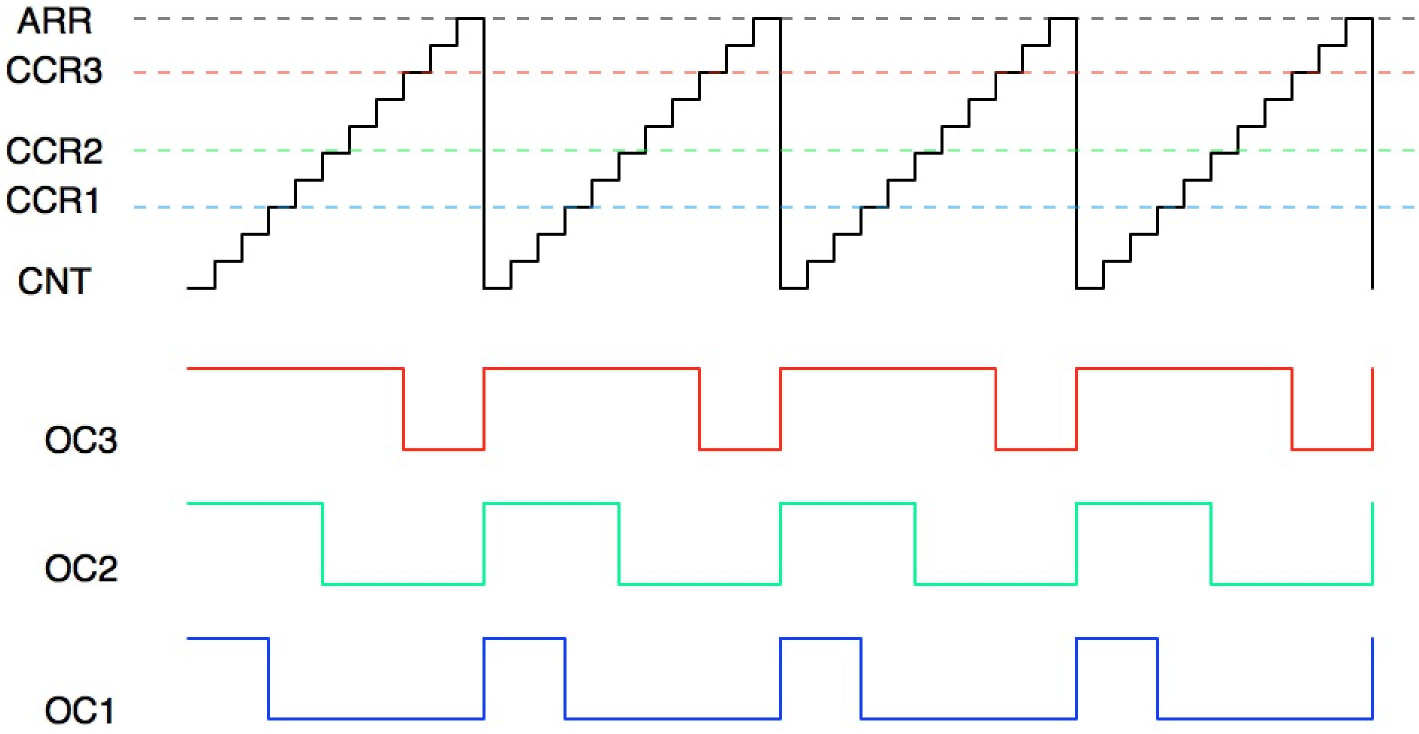
Timers feature



Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/ compare channels	Complementary output	Max. Interface clock (MHz)	Max. timer clock (MHz)
Advanced-control	TIM1	16-bit	Up, Down, Up/down	Any Integer between 1 and 65536	Yes	4	Yes	84	84
General purpose	TIM2, TIM5	32-bit	Up, Down, Up/down	Any Integer between 1 and 65536	Yes	4	No	42	84
	TIM3, TIM4	16-bit	Up, Down, Up/down	Any Integer between 1 and 65536	Yes	4	No	42	84
	TIM9	16-bit	Up	Any Integer between 1 and 65536	No	2	No	84	84
	TIM10, TIM11	16-bit	Up	Any Integer between 1 and 65536	No	1	No	84	84

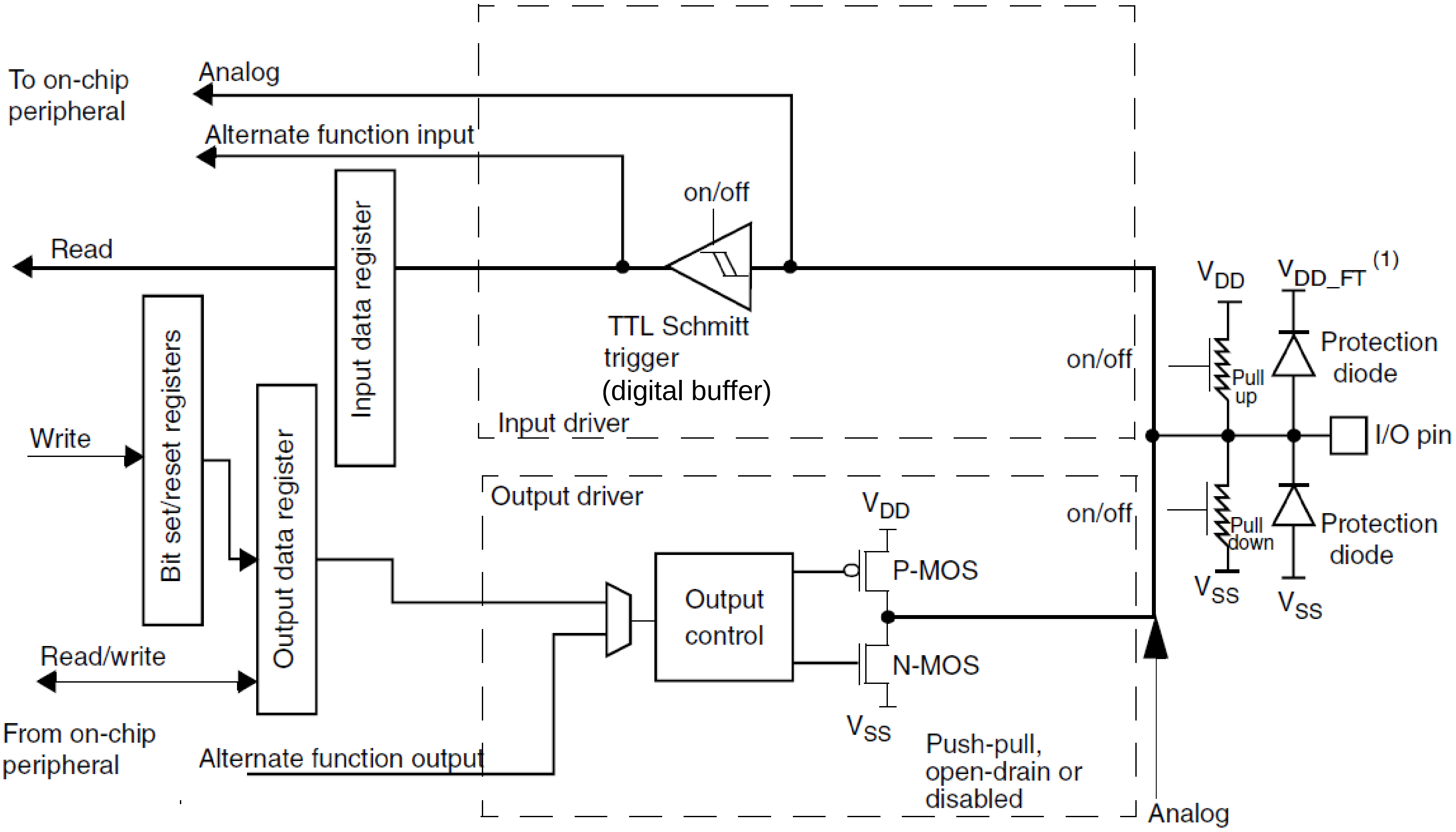
PWM generation

Three PWM signals from the Output Compare Channels of a general purpose timer



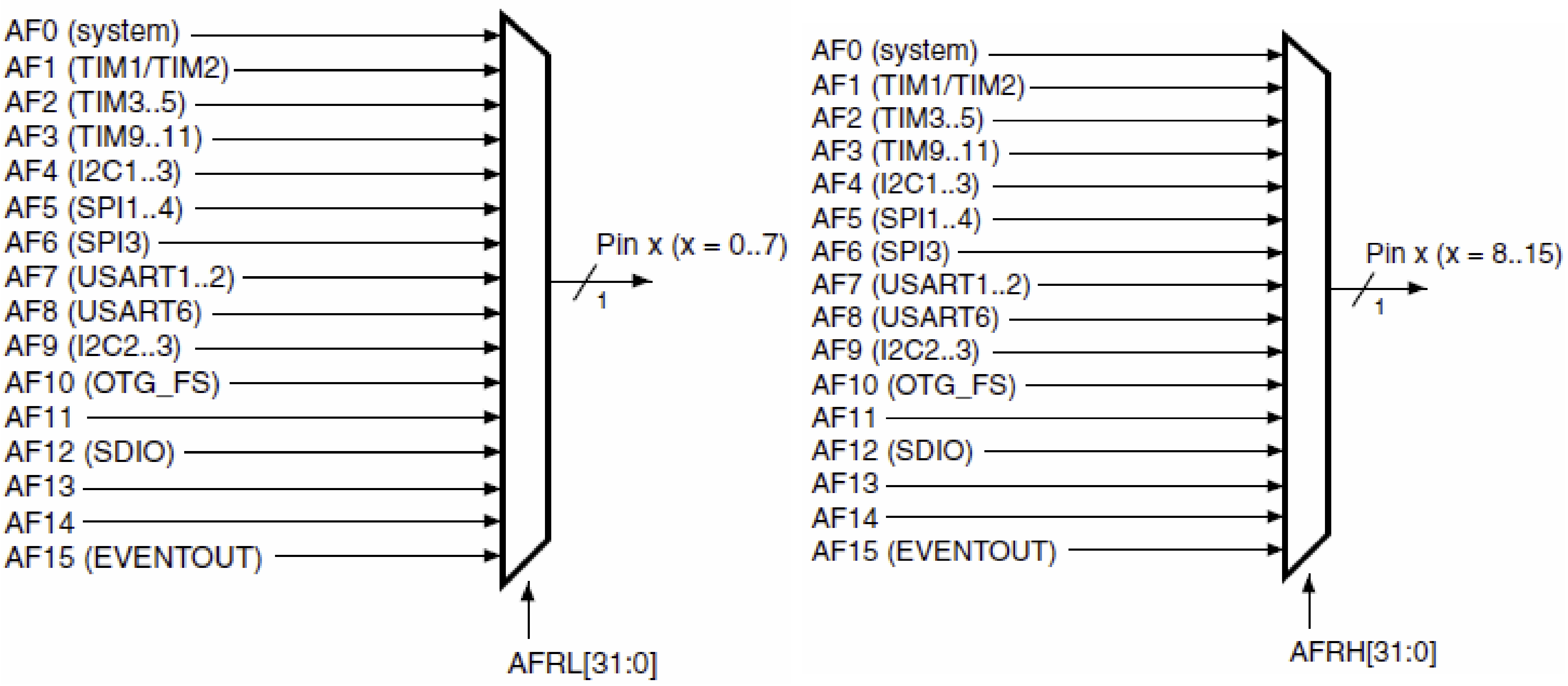
counters can operate in Up, Down or Up/Down mode

General Purpose Input/Outputs

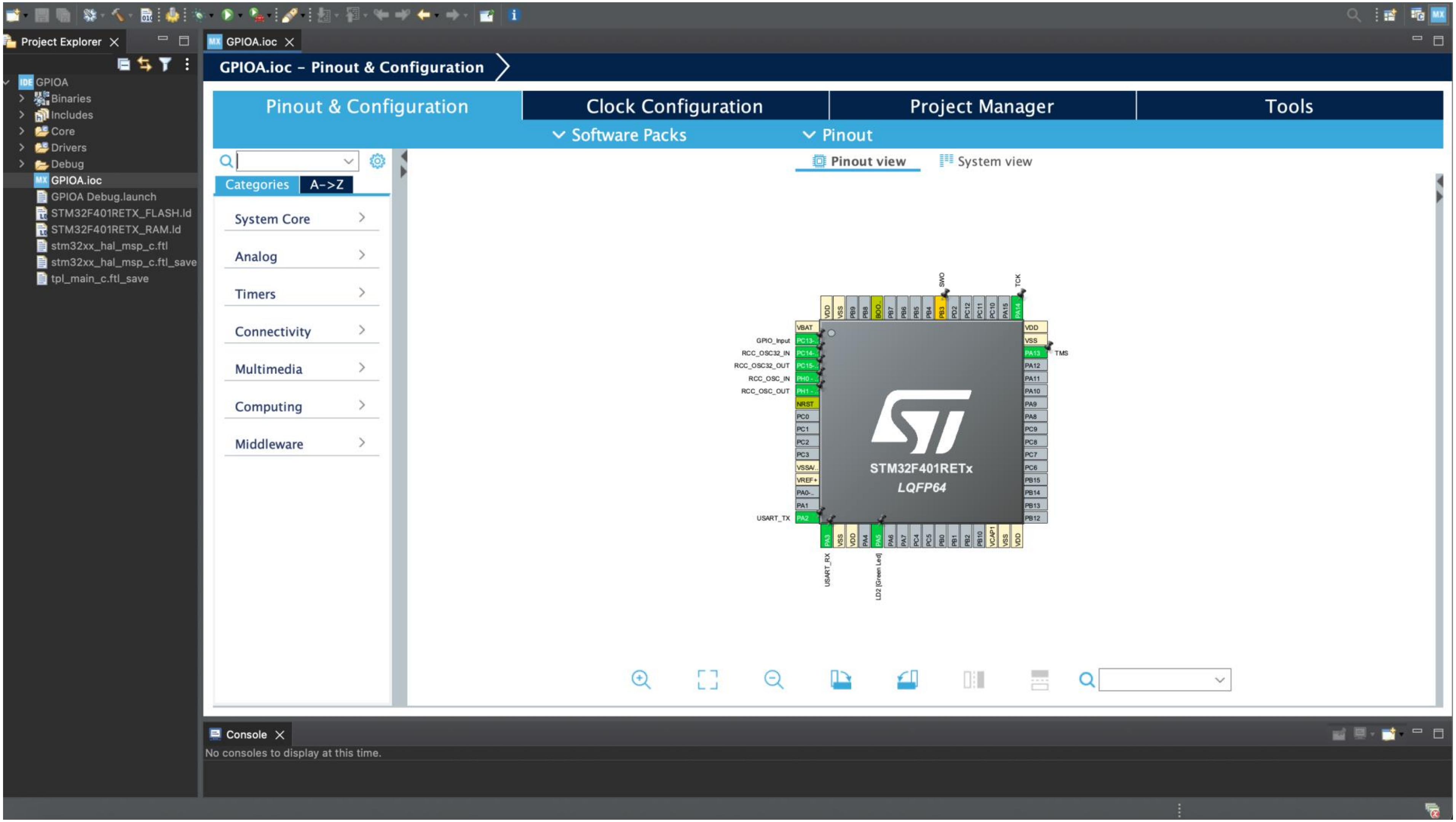


GPIO:
6 ports (A…E, H)
each port 16 PIN

Alternate Functions



Graphical User Interface (GUI)



No consoles to display at this time.

STM32 programming

Multiple ways to write programs for STM microcontrollers:

HAL Hardware Abstraction Layer drivers:

- ✓ Hide the MCU and peripheral complexity
- ✓ Maximize software portability across different microcontrollers
- ✓ Exist for all peripherals
- × Larger overhead → performance penalty

LL Low-Level drivers:

- ✓ Better optimization → improved performance
- × Less portability
- × Require deep knowledge of the MCU and peripheral specifications

Detailed description available in ST's UM1725 User manual – NOT required for this course

Summary

- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

Development board

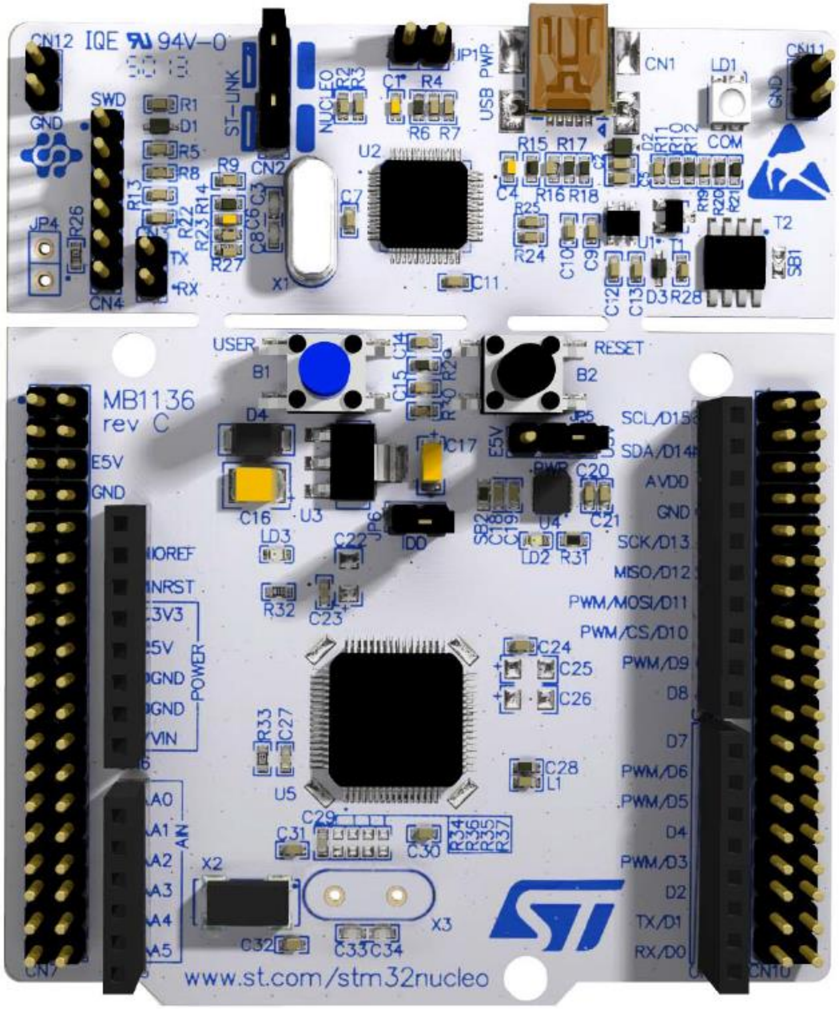
NUCLEO board
by STMicroelectronics

for STM32F family in LQFP64 package

Provided with libraries and examples.

Divided in:

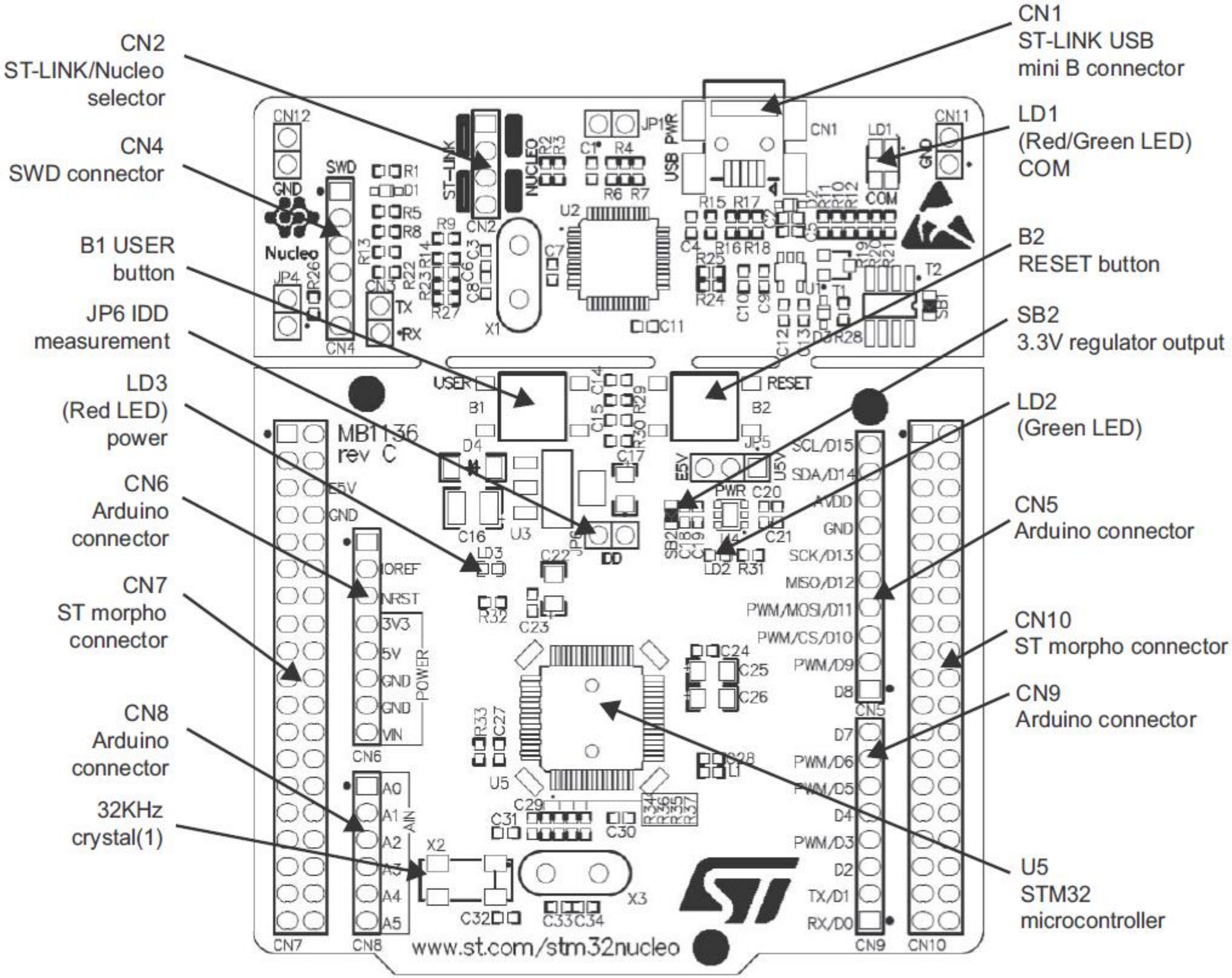
- microcontroller part
- removable ST-LINK debugger/programmer interface



Nucleo board features

- connector for Arduino Uno Revision 3
- general purpose connector (ST morpho extension)
  male pin headers accessible on both side of the board
  full access to STM32 I/Os
- 3 LEDs:
  - LD1 (COM) for USB communication (green, red)
  - LD2 available for user (connected to PA5) (green)
  - LD3 MCU powered, +5V power is available (red)
- 2 pushbuttons:
  - B1 for USER connected to PC13
  - B2 for RESET
- LSE (low-speed external) oscillator at 32.768 kHz
- 3 interfaces supported by the USB
  (Virtual Com port, Mass storage, Debug port)

Nucleo board layout



CN2
ST-LINK/Nucleo
selector

CN4
SWD connector

B1 USER
button

JP6 IDD
measurement

LD3
(Red LED)
power

CN6
Arduino
connector

CN7
ST morpho
connector

CN8
Arduino
connector

32KHz
crystal(1)

CN1
ST-LINK USB
mini B connector

LD1
(Red/Green LED)
COM

B2
RESET button

SB2
3.3V regulator output

LD2
(Green LED)

CN5
Arduino connector

CN10
ST morpho connector

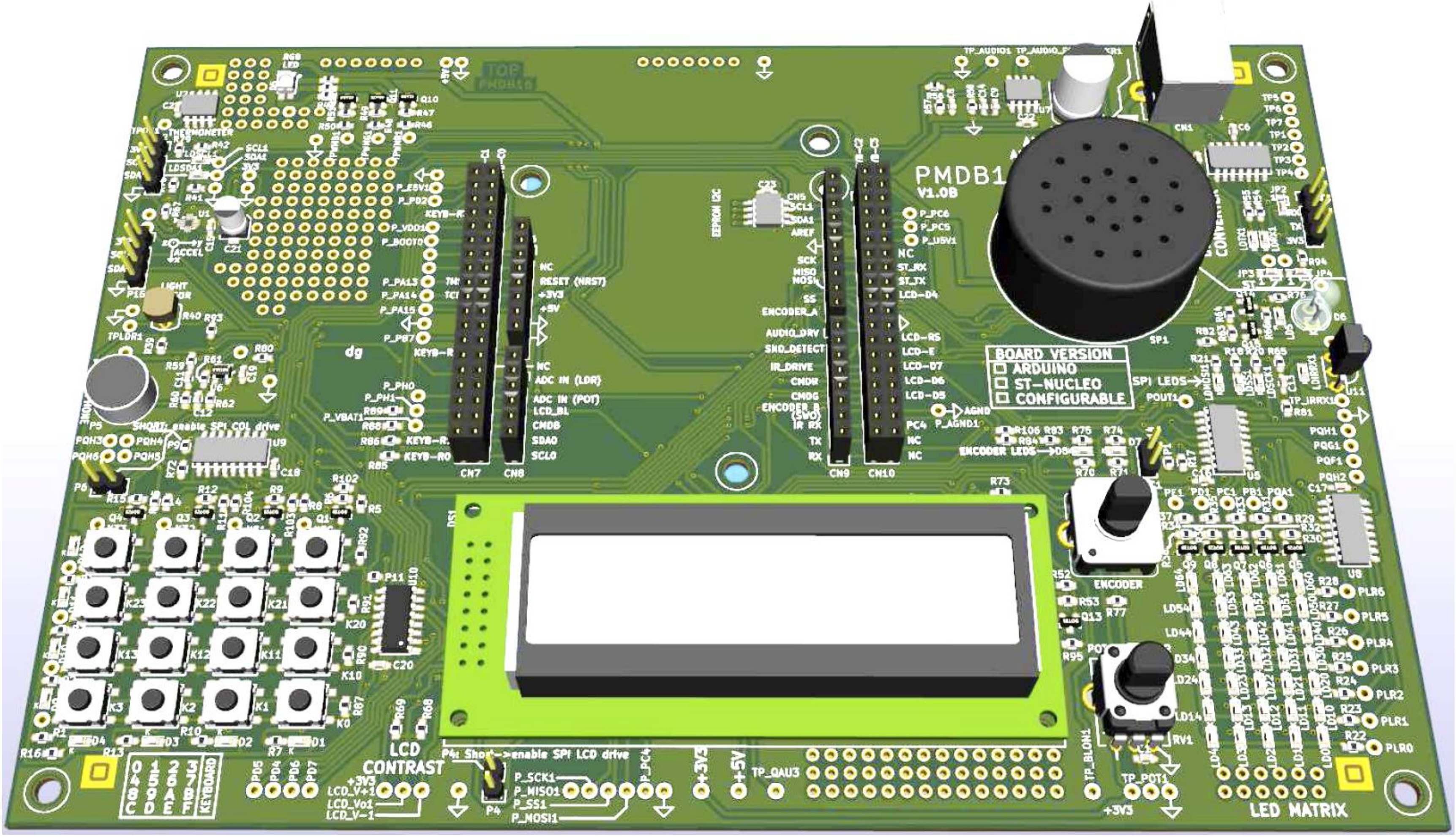
CN9
Arduino connector

U5
STM32
microcontroller

Summary

- ARM-Cortex cores
- Exceptions, Interrupts, and Faults
- STM32 microcontroller
- NUCLEO board
- POLIMI expansion board

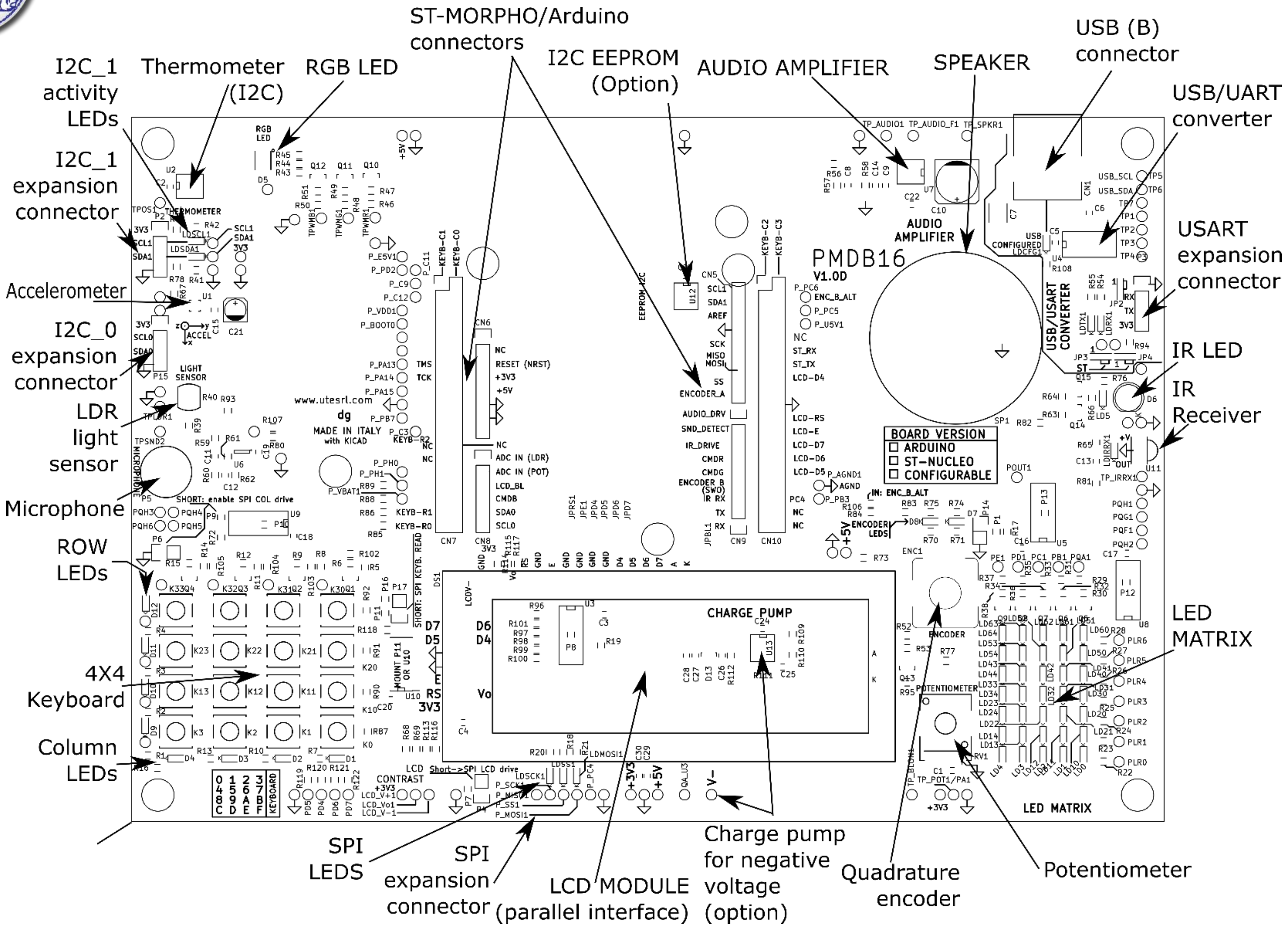
POLIMI expansion board



POLIMI board features

- standard 16x2 LCD alphanumeric module
- 5x7 LEDs array area, driven by two shift registers connected to the SPI interface
- 4x4 keypad, connected directly to the GPIOs
- incremental encoder
- microphone and comparator
- amplifier and speaker, connected to PWM
- RGB led, connected to three PWM-configurable outputs
- light sensor connected to ADC
- potentiometer connected to ADC
- thermometer, which communicates through I^{2}C peripheral
- accelerometer, which communicates through I^{2}C peripheral
- IR led and IR receiver

POLIMI board layout



ST-MORPHO/Arduino connectors

I2C EEPROM (Option)

AUDIO AMPLIFIER

SPEAKER

USB (B) connector

USB/UART converter

USART expansion connector

IR LED

IR Receiver

LED MATRIX

I2C_1 activity LEDs

Thermometer (I2C)

RGB LED

I2C_1 expansion connector

Accelerometer

I2C_0 expansion connector

LDR light sensor

Microphone

ROW LEDs

4X4 Keyboard

Column LEDs

SPI LEDS

SPI expansion connector

LCD MODULE (parallel interface)

Charge pump for negative voltage (option)

Quadrature encoder

Potentiometer