

Advanced Robot Control

Microcontroller architecture

Wojciech Domski

Chair of Cybernetics and Robotics,
Wrocław University of Science and Technology

Presentation compiled for taking notes during lecture



Wrocław University
of Science and Technology



- 1 Introduction
- 2 System components
 - Processor
 - Memory
 - Clock system
 - Reset system
 - Watchdog
 - Peripherals
 - Interrupts
 - Direct Memory Access (DMA)
 - Debugger
- 3 Power modes



Microcontroller – What is it? (1/3)

Is a microcontroller the same as processor?



Microcontroller – What is it? (2/3)

A microcontroller (MCU) is a devices which has:

- Processor (CPU, core) – a central processing unit. In simpler microcontrollers e.g 8-bit microcontroller the unit is represented by ALU (Arithmetic Logical Unit).
- Memory – usually MCU has three types of memory: program, data and non-volatile data memory.
- Clock generator and Real-Time Clock – a clock generator system, usually equipped with a Phase-Locked Loop to increase the system clock frequency.
- Reset system – a subsystem responsible for resetting the MCU.
- Watchdog – a peripheral which delivers robustness to the system against software failures.



Microcontroller – What is it? (3/3)

- Distribution logic – a network of connections between all subsystems in MCU.
- System bus – connects the processor and the memory together. Depending on the design it can also connect to peripherals.
- Peripherals – additional system components e.g. timers, ADC/DCA, SPI, I2C, etc.
- Interrupt block – a subsystem responsible for capturing the information about internal and external interrupts.
- DMA – Direct Memory Access, a system responsible for transferring data between system's components without processor.
- Debugger – a debug interface with direct access to



System architecture (1/2)

Figure: General system architecture outline [4]



System architecture (2/2)

Figure: STM32 system architecture (STM32F303XB/C) [2]



Processor (1/2)

Figure: Block diagram of the Cortex-M3/M4 processor [4]



Processor (2/2)

Types of instructions for ST MCUs

- General data processing and I/O control tasks (Cortex-M0/M0+/M1).
- Advanced data processing and bit field manipulation (Cortex-M3).
- DSP – Single Instruction Multiple Data (SIMD) and fast Multiply Accumulate Operation (MAC) (Cortex-M4).
- FPU (Cortex-M4F).



Memory

Memory can be divided into three groups in respect to the purpose:

- program memory to store instructions and data,
- data memory,
- non-volatile data memory (not always present).



Clock generator

The MCU is usually equipped with two clock sources:

- 1 A build-in RC oscillator.
- 2 An external signal from an external crystal/ceramic resonator or external clock signal.



Prescaler

The system clock signal is distributed to other system components and its frequency can be also divided by a prescaler, if necessary.



Low frequency clock signal

Some microcontrollers provide an additional low frequency clock signals or low frequency inputs. This signal is used to clock the Real-Time Clock (RTC) subsystem but also can be but used for Watchdog or auto-wakeup functions.



Clock tree

Figure: Clock tree for Cortex-M3 based ST MCU [1]



Reset types

There are three types of reset:

- system reset,
- power reset,
- backup domain reset.



System reset

A system reset sets all registers to their reset values. However, there are some exceptions. The source of reset can be identified by checking the appropriate register.



Power reset

The power reset can happen when one of following occurs:

- Power cycle.
- Exit from Standby mode.



Reset system

Figure: Reset circuit [1]



Watchdog

A watchdog is a specific timer which can cause a reset of the MCU. The main purpose of a watchdog is to detect a software failure. There are two types of watchdogs

- Independent watchdog – can trigger a system reset.
- Window watchdog – can trigger a system reset or an interrupt.



Peripherals

Each microcontroller can be equipped with a different set of different peripherals. Some of the most common peripherals are listed below:

- GPIO,
- ADC,
- DAC,
- timers,
- SPI,
- I2C,
- CAN,
- SDIO,
- USB,
- **USART/UART,**
- **FSMC.**



Interrupts

The main purpose of an interrupt is to change the current program flow because (typically) a hardware event have occurred.



Enabling interrupts

Figure: Sources of exceptions in a MCU [4]



Enabling interrupts

To enable an interrupt following sequence is required:

- Set up priority level (this is optional, after reset all interrupt levels are reset to default value of 0).
- Enable the interrupt generation in the specific peripheral.
- Enable the interrupt in the Nested Vectored Interrupt Controller (NVIC).



Extended interrupts and events controller (EXTI)

The extended interrupts and events controller (EXTI) manages the external and internal asynchronous events/interrupts and generates the event request to the CPU/Interrupt Controller and a wake-up request to the Power Manager [2].



Priorities and groups

Depending on the importance of an interrupt a specific level is assigned to the interrupt.

The lower the priority number the more important the interrupt is.

Figure: An example of division in priority level register [4]



Pre-emption (1/1)

Pre-emption is a situation when a microcontroller is already in ISR and another interruption happens. What is more, the new interrupt is handled by the processor and when it finishes it returns to the previous ISR.



Masking (1/1)

Interrupt masking is a technique which allows to temporarily all interrupts so a critical part of code could be executed without interruption.



DMA (1/3)

Direct Memory Access (DMA) is a controller which allows a memory-memory and memory-peripheral data transfer without involvement from processor.



DMA (2/3)

Figure: STM32 system architecture (STM32F303XB/C) [2]



DMA (3/3)

Figure: STM32 DMA block diagram (STM32F303XB/C) [2]



Debugger (1/2)

Debug interface allows to stop on a given instruction fetch (**breakpoint**) or data access (**watchpoint**).

Debug interfaces allows to freeze instruction execution while the processor and peripherals state can be examined.



Debugger (2/2)

Figure: STM32 debug block diagram (STM32F303XB/C) [2]



Power modes (1/2)

ST microcontrollers can be in one of following operation modes:

- Run mode – normal operation. All system components are running.
- Sleep mode – only main peripherals like core, NVIC, SysTick, etc. are running.
- Stop mode – all clocks are stopped.
- Standby mode – the 1.8V domain is powered off.



Power modes (2/2)

Figure: STM32 Low-power modes possible transitions (STM32L4x5/6) [3]



Wake up

It is always possible to exit low-power mode, however, it varies depending on the mode.



Quiz (1/1)

Calculate group number as the rest from dividing the Student ID number by 4.

Example

Student ID number is 123456, thus the group is 0.

Take last 2 digits from Student ID number (56) and calculate the rest from dividing by 4 ($56 \% 4 = 0$).

Write down your name, Student ID number and group.



Literature (1/2)



ST.

STM32F100xx advanced ARM-based 32-bit MCUs, Reference manual.

ST, 2010.



ST.

STM32F303xB/C/D/E, STM32F303x6/8, STM32F328x8, STM32F358xC, STM32F398xE advanced ARM[®]-based MCUs, Reference manual.

ST, 2017.



ST.

STM32L4x5 and STM32L4x6 advanced ARM[®]-based 32-bit MCUs, Reference manual.

ST, 2017.





J. Yiu.

*The Definitive Guide to ARM[®] Cortex[®]-M3 and
Cortex[®]-M4 Processors.*

Newnes, 2013.

