

Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

Hands-On 1-2

STM32CubeIDE: Eclipse, STM32CubMX, STM32F7 HAL

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

STM32CubeIDE

The **STM32CubeIDE** is an integrated development environment for STM32 [Ref_11-x]:

- Integration of **STM32CubeMX**:
 - STM32 microcontroller, microprocessor, development platform and example project selection
 - Pinout, clock, peripheral, and middleware configuration
 - Project creation and generation of the initialization code
 - Software and middleware completed with enhanced STM32Cube Expansion Packages
- Based on Eclipse/CDT, with support for Eclipse add-ons, GNU C/C++ for Arm toolchain and GDB debugger
- Additional advanced debug features including:
 - CPU core, peripheral register, and memory views
 - Live variable watch view
 - System analysis and real-time tracing (SWV)
 - CPU fault analysis tool
 - RTOS-aware debug support
- Support for ST-LINK (STMicroelectronics) and J-Link (SEGGER) debug probes
- Import project from Atollic TrueSTUDIO and AC6 System Workbench for STM32 (SW4STM32)
- Multi-OS support: Windows, Linux, and macOS, 64-bit versions only

CDT: C/C++ Development Tooling

STM32CubeIDE

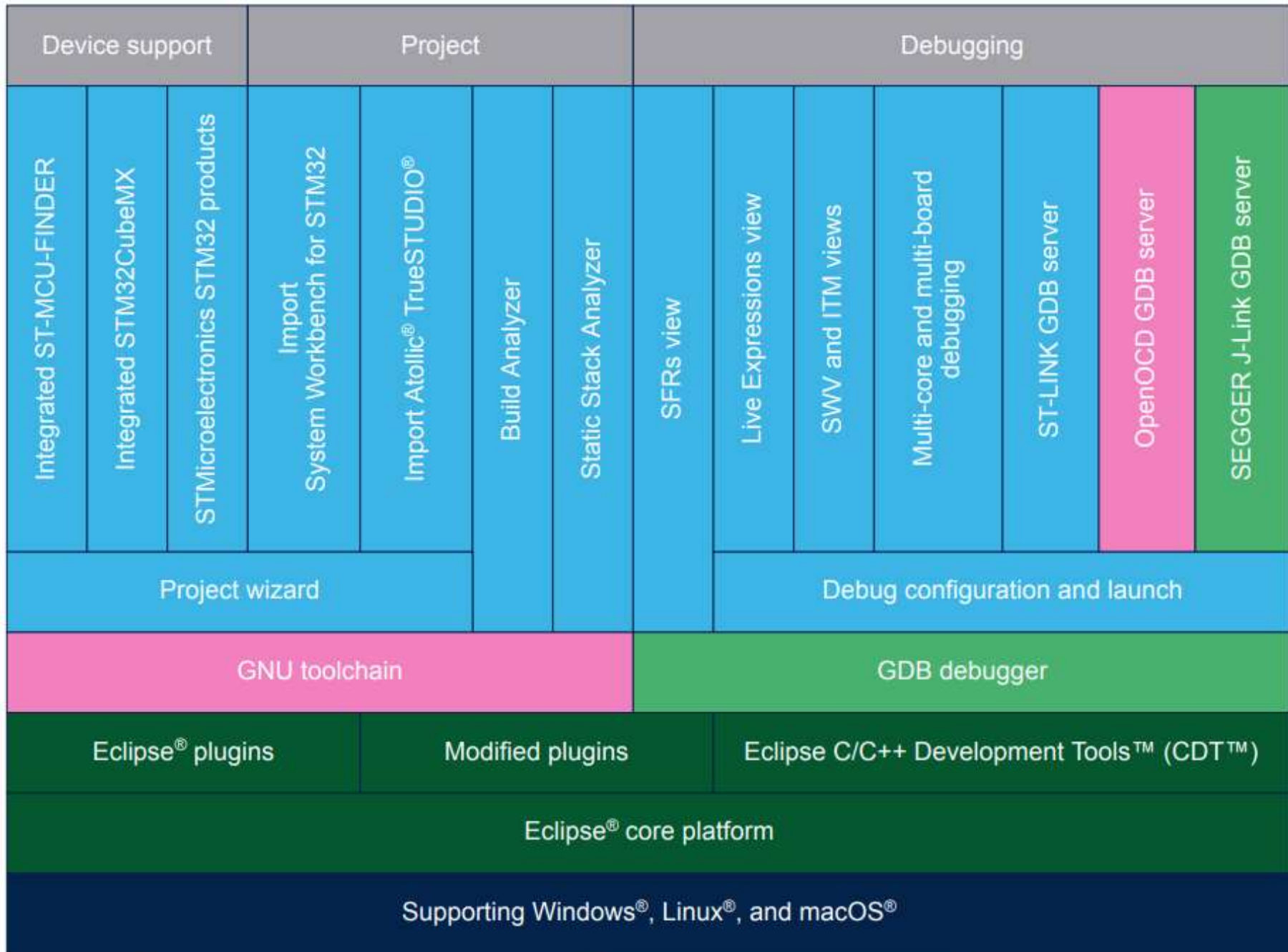
STM32CubeIDE: STM32Cube MCU and MPU Packages

STM32Cube MCU and MPU packages (e.g., **STM32CubeF7** for **STM32F7** Series) include

- STM32Cube Hardware Abstraction Layer (**HAL**) – an STM32 abstraction layer embedded software, ensuring maximized portability across the STM32 portfolio
- STM32Cube **Low Layer APIs**, ensuring the best performance and footprints with a high degree of user control over hardware
- A consistent set of **middleware components** such as RTOS, CAN, USB, TCP/IP, Graphics
- Embedded **software utilities** with full sets of peripheral and applicative **examples**

Note: STM32Cube is an original initiative of STMicroelectronics to significantly improve designer productivity by reducing development effort, time, and cost. It covers the whole STM32 portfolio, which includes STM32CubeIDE, STM32CubeMX, STM32CubeProg, STM32CubeMonitor (power, RF, USB-PD), STM32Cube MCU and MPU Packages, and STM32Cube Expansion Packages.

STM32CubeIDE Key Features [Ref_11-6]



Legend:

Specific STM32CubeIDE functions

Open-based updated by ST

Base technology platform

STM32CubeIDE main function groups

Third-party solutions

Operating systems

STM32CubeIDE Window

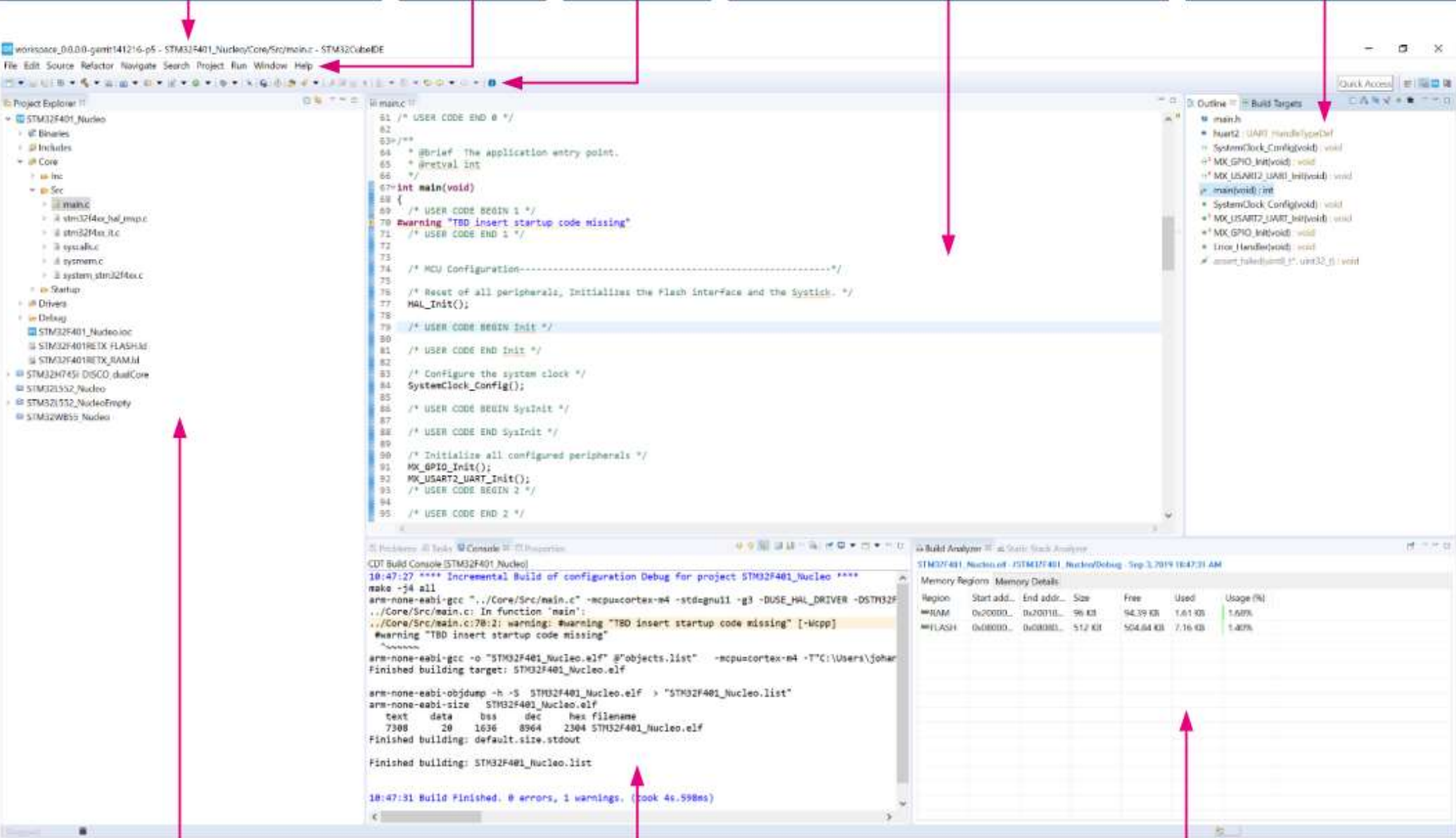
Name of current workspace

Menu bar

Toolbar

Editor area

Outline view



Project Explorer

Console view

Build Analyzer

STM32CubeIDE

STM32CubeIDE basics

Learning of STM32F767ZI and STM32CubeIDE

FreeRTOS on STM32 v2

STM32 MOOCs (Massive Open Online Courses)

STM32F7 workshop MOOC

https://www.st.com/content/st_com/en/support/learning/stm32-education/stm32-moocs.html

STM32F7 Online Training

https://www.st.com/content/st_com/en/support/learning/stm32-education/stm32f7-online-training.html

- Read the document

[Ref_02-1, Ref_02-2] DS11532 & ES0334, data-sheet & errata-sheet of STM32F767xx,

[Ref_03] RM0410, reference manual of STM32F76xxx,

[Ref_05-1] UM1974, user manual of STM32 Nucleo-F767ZI board,

[Ref_07-1] UM1718, user manual of STM32CubeMX,

[Ref_08] UM1905, user manual of STM32F7 HAL and Low-Layer Drivers,

[Ref_11-6] UM2609, user guide of STM32CubeIDE, and

other related manuals.

STM32CubeIDE

Key Features of STM32CubeMX **[Ref_07-1]**, User Manual **UM1718**

- MCU/MPU Selector
 - Filter by family, package, peripherals or memory sizes.
- Pinout Configuration
 - Choose peripherals, assign GPIO, and alternate functions to pins.
- Clock Configuration
 - Choose oscillator, set PLL, and set clock frequencies and clock dividers.
- Peripheral and Middleware Parameters
- Power Consumption Calculator
- Code Generation
 - Generate or re-generate code while keeping user code intact.

STM32CubeIDE

Start a New STM32 Project – Target Selection

Target Selection

⚠ STM32 target or STM32Cube example selection is required

MCU/MPU Selector Board Selector Example Selector Cross Selector

Matching of ST MCUs to other vendors' MCUs.

MCU/MPU Filters

★ 📁 🔍 ↺

Part Number

Core >

Series >

Line >

Package >

Other >

Peripheral >

MCU/MPU can be selected by:

- Part Number (STM32F767ZI)
- Core (ARM Cortex-M7)
- Series (STM32F7)
- Line (STM32F7x7)
- Package (LQFP144)
- Other (Price, IO, Eeprom, Flash, Ram, Freq.)
- Peripherals (CAN, Ethernet)

STM32CubeIDE

Start a New STM32 Project → Target Selection → MCU/MPU Selector → type "f767"

STM32 Project

Target Selection

STM32 target or STM32Cube example selection is required

MCU/MPU Selector | Board Selector | Example Selector | Cross Selector

MCU/MPU Filters

Commercial Part Number: f767

PRODUCT INFO

MEMORY

TIMER

ANALOG

COMMUNICATION IN

USB INTERFACE

EXTERNAL MEMORY

OTHER INTERFACE

GRAPHICS

STM32F767BGT6

STM32F767BIT6

STM32F767IGK6

STM32F767IGT6

STM32F767IHK6

STM32F767IIT6

STM32F767NGH6

STM32F767NIH6

STM32F767NIH6TR

STM32F767NIH7

STM32F767VGH6

STM32F767VGT6

STM32F767VGT7

STM32F767VIH6

STM32F767VIT6

STM32F767VIT7

STM32F767ZGT6

STM32F767ZIT6

Features

Block Diagram

Docs & Resources

Datasheet

Buy

STM32F7 Series

STM32F767ZIT6

High-performance and DSP with FPU, Arm Cortex-M7 MCU with 2 Mbytes of Flash memory, 216 MHz CPU, Art Accelerator, L1 cache, SDRAM, TFT, JPEG codec, DFSDM

ACTIVE

Product is in mass production

Unit Price for 10kU (US\$): 9.6322

LQFP 144 20x20x1.

MCUs/MPUs List: 18 items

+ Display similar items

Export

	Commercial Part ...	Part No	Refer...	Ma...X	Unit ...X	Board X	Pac...X	Flash X	RAM X
☆	STM32F767VGT6	STM32F767VGT6	STM...	Active	8.5152		LQF...	1024...	512...
☆	STM32F767VGT7		STM...	Active	8.9883		TFB...	2048...	512...
☆	STM32F767VIH6		STM...	Active	8.9883		LQF...	2048...	512...
☆	STM32F767VIT6	STM32F767VI	STM...	Active	8.9883		LQF...	2048...	512...
☆	STM32F767VIT7		STM...	Active	9.6175		LQF...	2048...	512...
☆	STM32F767ZGT6	STM32F767ZG	STM...	Active	8.602		LQF...	1024...	512...
☆	STM32F767ZIT6	STM32F767ZI	STM...	Active	9.6322	NUCLEO	LQF...	2048...	512...

Select

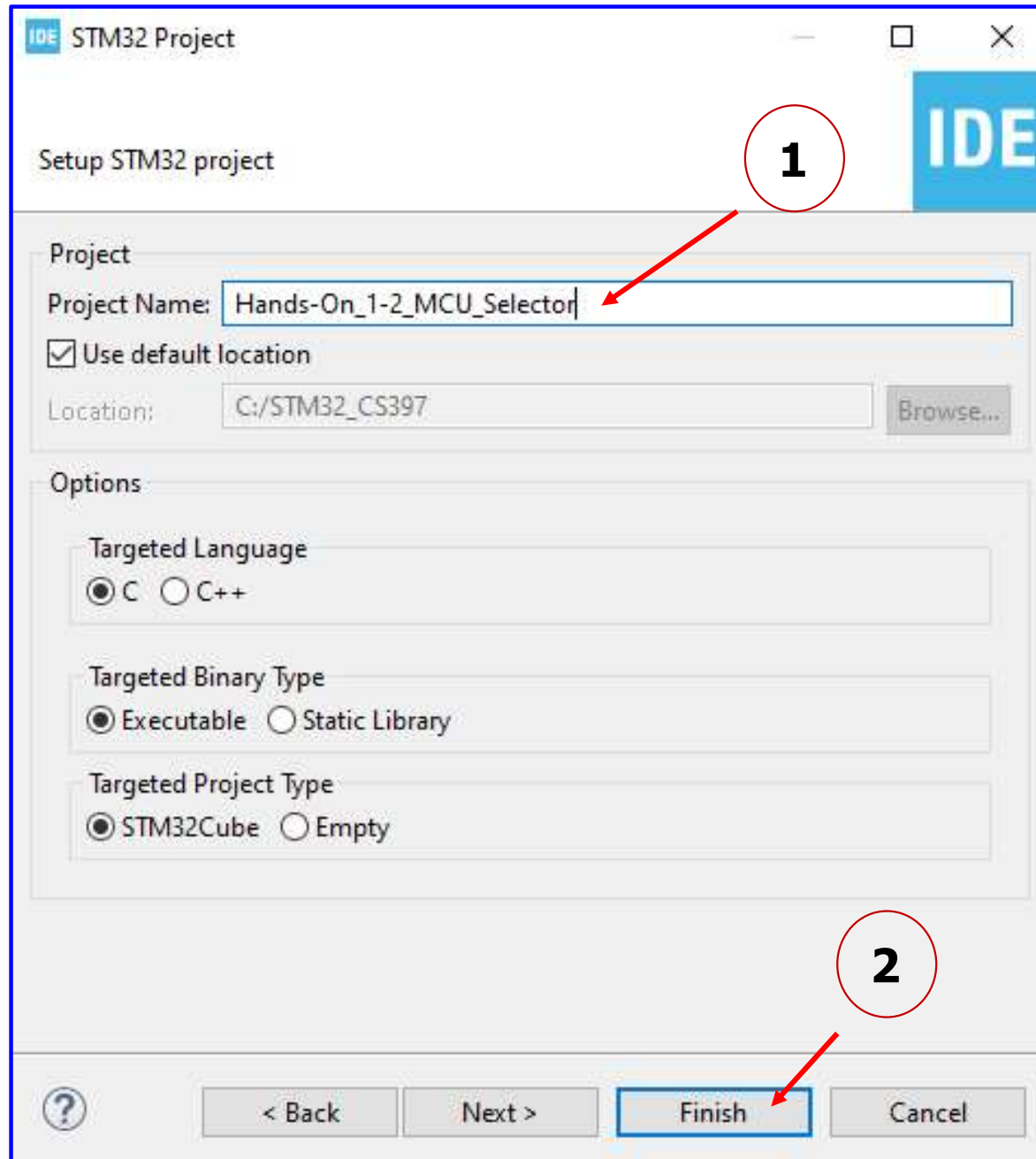
Next >

Finish

Cancel

STM32CubeIDE

Start a New STM32 Project → Enter Project Name: Hands-On_1-2_MCU_Selector



Hands-On_1-2_MCU_Selector.ioc

STM32CubeIDE

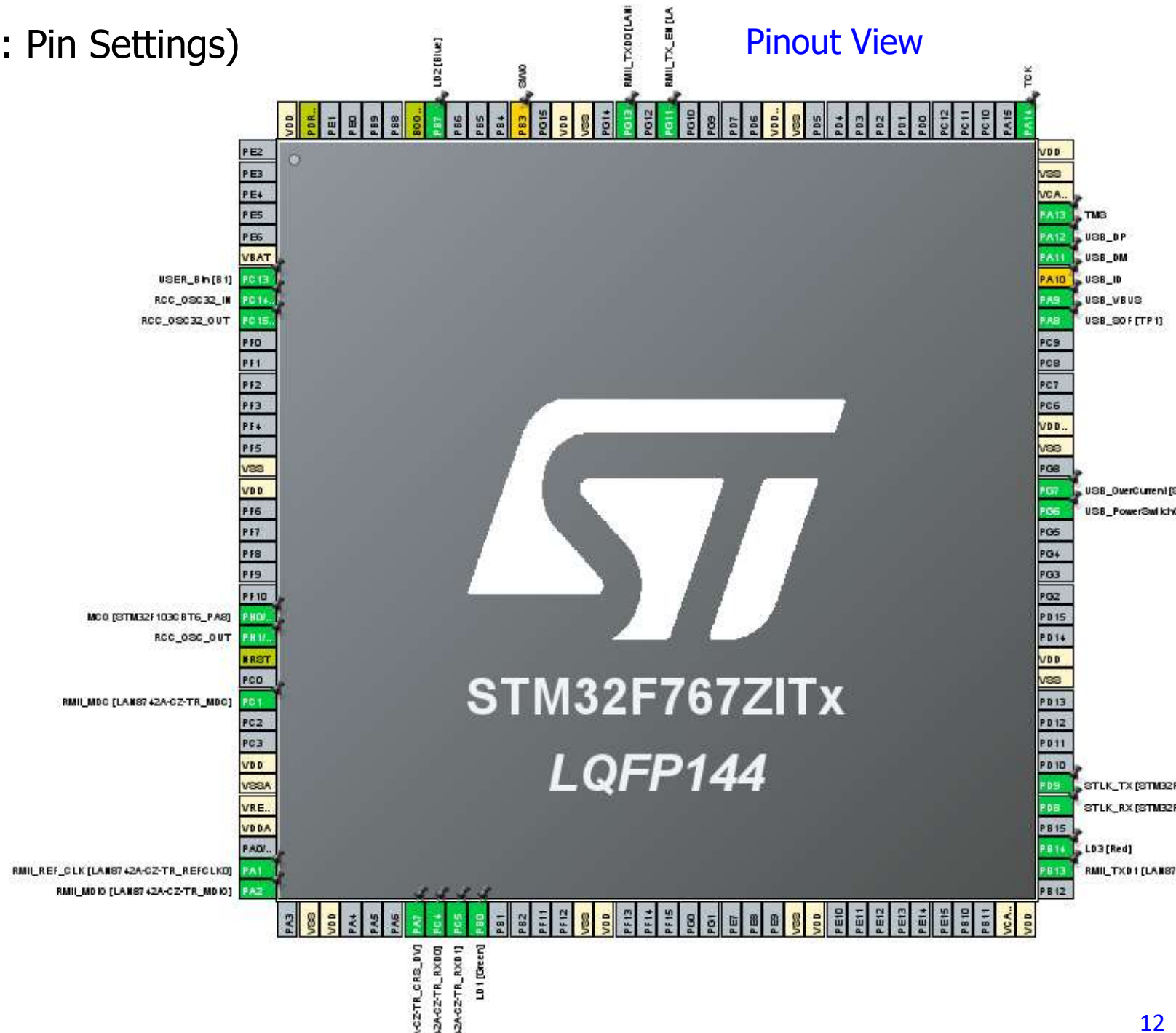
Pinout View

(STM32CubeMX file)



(Nucleo – F767ZI: Pin Settings)

Pinout View



Configure CAN1

STM32CubeIDE

Hands-On_1-2_MCU_Selector.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

Categories: A-Z

Connectivity

1 CAN1

CAN2
CAN3
ETH
FMC
I2C1
I2C2
I2C3
I2C4
MDIOS
QUADSPI
SDMMC1
SDMMC2
SPI1
SPI2
SPI3
SPI4
SPI5
SPI6
UART4
UART5
UART7
UART8
USART1

CAN1 Mode and Configuration

Mode

☒ Activated **2**

Configuration

Reset Configuration

NVIC Settings | GPIO Settings
Parameter Settings | User Constants

Configure the below parameters :

Search (Ctrl+F)

Bit Timings Parameters

Prescaler (for Time Qu... 16
Time Quantum 1000.0 ns
Time Quanta in Bit Seg... 1 Time
Time Quanta in Bit Seg... 1 Time
Time for one Bit 3000 ns
Baud Rate 333333 bit/s
ReSynchronization Ju... 1 Time

Basic Parameters

Time Triggered Commu... Disable
Automatic Bus-Off Man... Disable
Automatic Wake-Up M... Disable
Automatic Retransmis... Disable
Receive Fifo Locked M... Disable

Pinout view | System view

Note: After configured the pins, default labels are given. Right-click on the pin to select "Enter User Label" to change the name if needed.

(Left-click on PC12)

Configure GPIO Pin

PC12
Reset_State
DCMI_D9
I2S3_SD
SDMMC1_CK
SPI3_MOSI
SYS_TRACED3
UART5_TX
USART3_CK
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI12

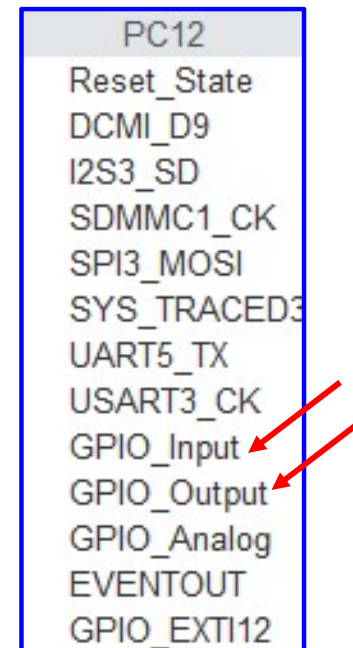
PC11
PC10
PA15
PA14

VDD
VSS
VCAP..
PA13
PA12
PA11
PA10

3 CAN1_TX
CAN1_RX

Pin Assignment

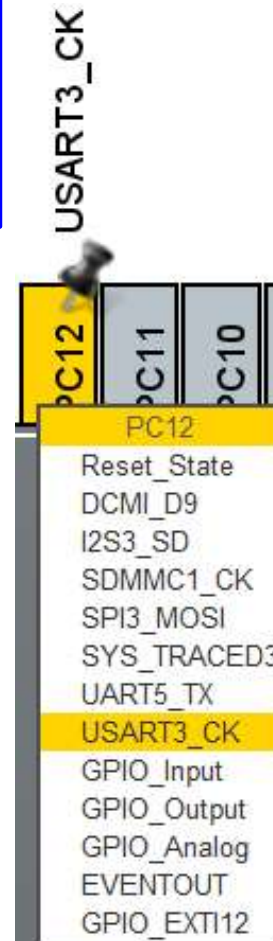
- **Pin assignment:** select a peripheral and assign pins to I/Os.
- E.g., when the **CAN1** peripheral is selected and activated, the tool automatically assigns the **default pins**. If default pins are not preferred, they can be transferred to **alternate locations, i.e.,** re-mapping of pins is possible.
- **Independent GPIOs** such as digital I/Os can be configured **manually**.
- Pinout selection/configuration:
 - **Peripheral tree** (e.g., CAN1)
 - **Manually**



PC12
Reset_State
DCMI_D9
I2S3_SD
SDMMC1_CK
SPI3_MOSI
SYS_TRACED3
UART5_TX
USART3_CK
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI12

Pin Assignment

- When more pins are used for specific functions, the choice of remaining configurations for other peripherals decreases. The limitations are indicated by **icon changes** on other peripheral **modes**.
- **Right-click** on the selected/green pin to **enter user label**, **pin/un-pin**, and **stack/un-stack**. <https://www.youtube.com/watch?v=d6k55AgiyVY>
- If a pinout is selected **without a particular peripheral enabled** or if there is any other problem with the pinout, the pin turns **orange** instead of **green**.
- Pins can be set/moved directly from the pinout view.
 - **Left-click** on the pin to display the list of possible signals (**alternate functions**) and select one. This works for GPIOs which have no peripherals assigned.
 - **To see alternate pins**, **CTRL+Left-click** on the pin. You can then drag and drop the pin to the new pin (while **holding the CTRL key**).



STM32CubeIDE

Pin Assignment

- There are different possible states for peripheral modes:
 - **Dimmed:** The **mode is not available** because it requires another mode to be set. Place the mouse pointer over the dimmed mode to see the reason - it may require a disabled clock source or have other peripheral dependencies.
 - **Orange:** The **mode is available with limitations** because some options are blocked by conflicts. For example, the USART may not be configured to synchronous mode because all selectable clock pins are taken.
 - **Red:** **Signals required for this mode cannot be mapped to the pinout.** This may occur, for example, if a crucial signal has all its alternate pins used by other peripherals.
 - **Green:** **Periphery is assigned properly to pinout.**
- Ignore unused pins: the code generator can set them to power-saving analog mode.

[Ref_07-1] UM1718: Table 5. Component list, mode icons and color schemes (page 70,71)





Plain black text
Example: UART5
Gray italic text
Example: LWIP
 
Example::  ETH

Example:  USB_OTG_HS
Example:  USB_OTG_HS

Example:  I2C2
Example: IRTIM

Table 5. Component list, mode icons and color schemes









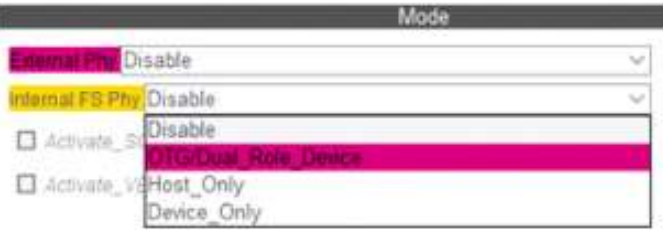


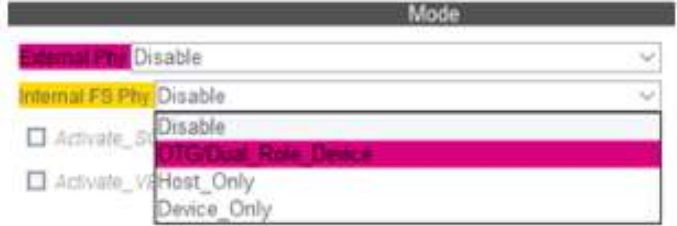

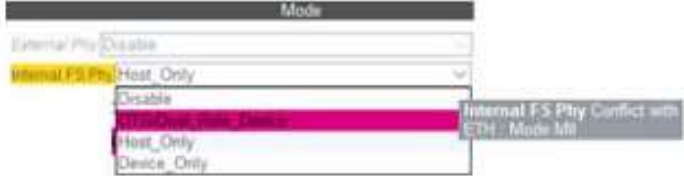


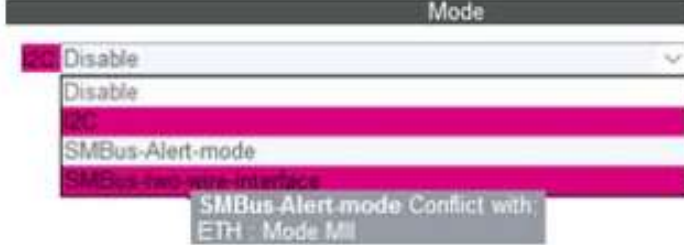

Display	Component status	Corresponding Mode view / Tooltips
Plain black text Example: UART5	The peripheral is not configured (no mode is set) and all modes are available.	
Gray italic text Example: LWIP	Peripheral is not available because some constraints are not solved. See tooltip.	
  Example::  ETH	The peripheral is configured (at least one mode is set) and all other modes are available. The green check mark indicates that all parameters are properly configured, a cross indicates they are not.	
 Example:  USB_OTG_HS	The peripheral is not configured (no mode is set) and at least one of its modes is unavailable.	

Table 5. Component list, mode icons and color schemes (continued)

 Example:  USB_OTG_HS	The peripheral is not configured (no mode is set) and at least one of its modes is unavailable.	
Example:  USB_OTG_HS	The peripheral is configured (one mode is set) and at least one of its other modes is unavailable.	
 Example:  I2C2	The peripheral is not configured (no mode is set) and no mode is available. Move the mouse over the peripheral name to display the tooltip describing the conflict.	
Example: IRTIM	Peripheral is not available because of constraints.	

Pin Assignment

- Why are some pins highlighted in **yellow** or in **light green** in the Pinout view?

Why cannot I change the function of some pins (when I click some pins, nothing happens)?

Ans: These pins are specific pins (such as power supply or BOOT) which are not available as peripheral signals.

Acronyms and Definitions [Ref_03]

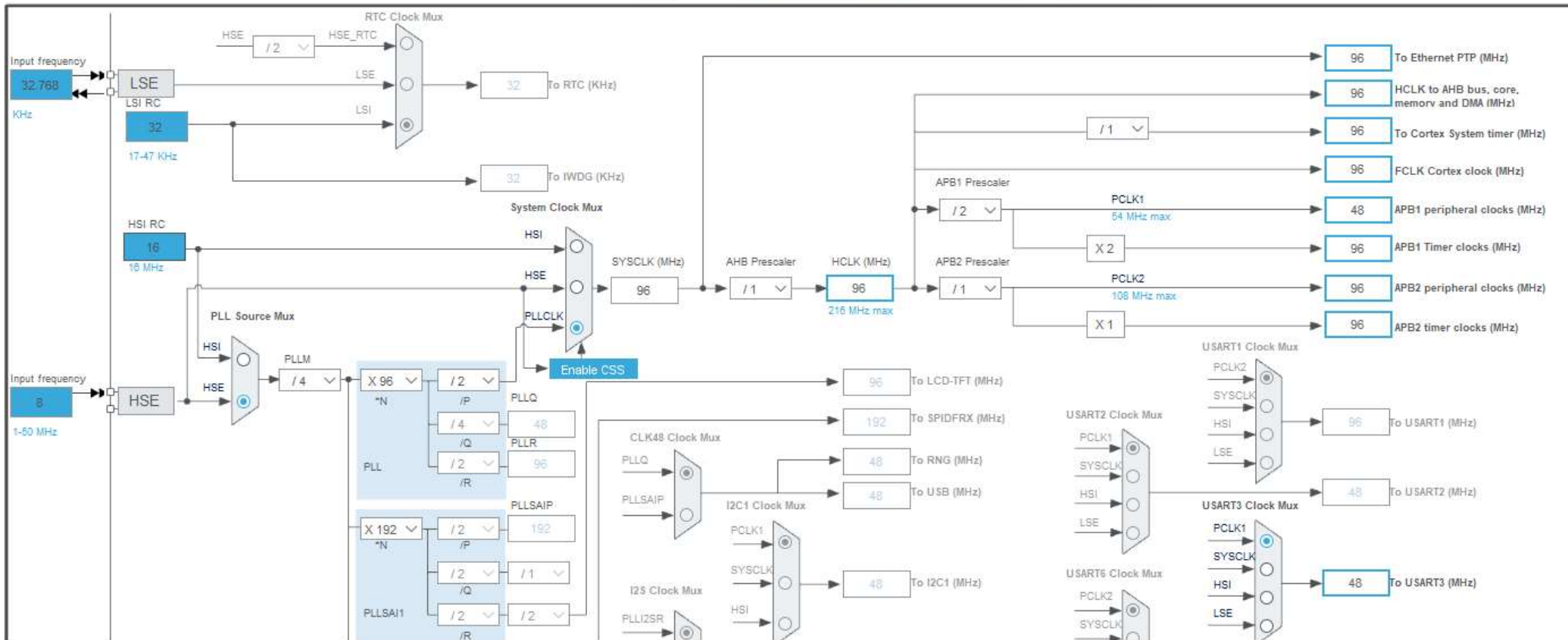
Pinout & Configuration

Clock Configuration

Project Manager

Tools

Resolve Clock Issues



LSI: Low Speed Internal
 HSI: High Speed Internal
 RTC: Real-Time Clock
 LSE: Low Speed External
 HSE: High Speed External
 PLL: Phase Locked Loop
 CSS: Clock Security System

IWDG: Independent Watchdog
 SYSCLK: System Clock
 AHB: Advanced High-Performance Bus
 APB: Advanced Peripheral Bus
 HCLK: AHB Clock
 PCLK: Peripheral Clock
 PTP: Precision Time Protocol

Acronyms and Definitions (1/2) [Ref_03]

Acronym Definition

ADC	Analog-to-digital converter
ANSI	American National Standards Institute
API	Application Programming Interface
BSP	Board Support Package
CAN	Controller area network
CEC	Consumer electronic controller
CMSIS	Cortex Microcontroller Software Interface Standard
CPU	Central Processing Unit
CRC	CRC calculation unit
DAC	Digital to analogue converter
DFSDM	Digital filter sigma delta modulator
DMA	Direct Memory Access
DSI	Display serial interface
ETH	Ethernet controller
EXTI	External interrupt/event controller
FLASH	Flash memory
FMC	Flexible memory controller
GPIO	General purpose I/Os
HAL	Hardware abstraction layer
HCD	USB Host Controller Driver
I2C	Inter-integrated circuit
I2S	Inter-integrated sound
IRDA	InfraRed Data Association
IWDG	Independent watchdog

Acronym Definition

LCD	Liquid Crystal Display Controller
LTDC	LCD TFT Display Controller
MDIOS	Management data input/output (MDIO) slave
MMC	Multi-Media Card
MSP	MCU Specific Package
NAND	NAND Flash memory
NOR	Nor Flash memory
NVIC	Nested Vectored Interrupt Controller
PCD	USB Peripheral Controller Driver
PPP	STM32 peripheral or block
PWR	Power controller
QSPI	QUADSPI Flash memory Interface
RCC	Reset and clock controller
RTC	Real-time clock
SAI	Serial Audio Interface
SD	Secure Digital
SRAM	SRAM external memory
SMARTCARD	Smartcard IC
SPI	Serial Peripheral interface
SPDIFRX	Sony/Philips Digital InterFace Receiver
SysTick	System tick timer
TIM	Advanced-control, general-purpose or basic timer
UART	Universal asynchronous receiver/transmitter
USART	Universal synchronous receiver/transmitter
USB	Universal Serial Bus
WWDG	Window watchdog

Acronyms and Definitions(2/2) [Ref_03]

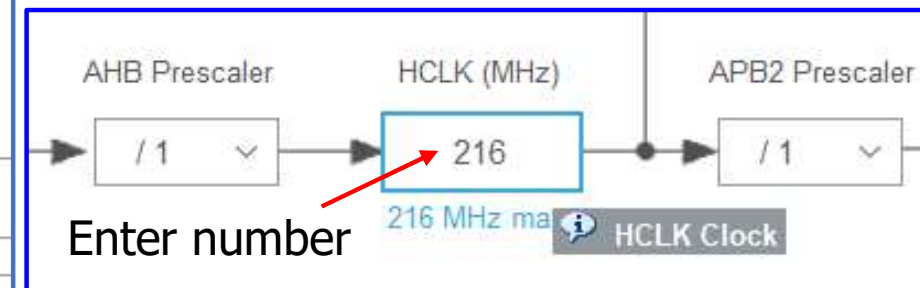
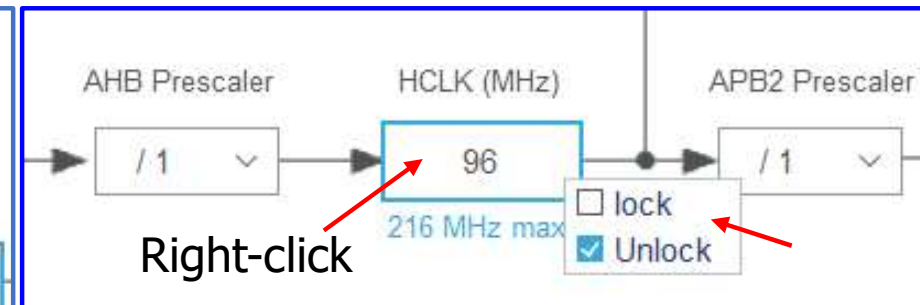
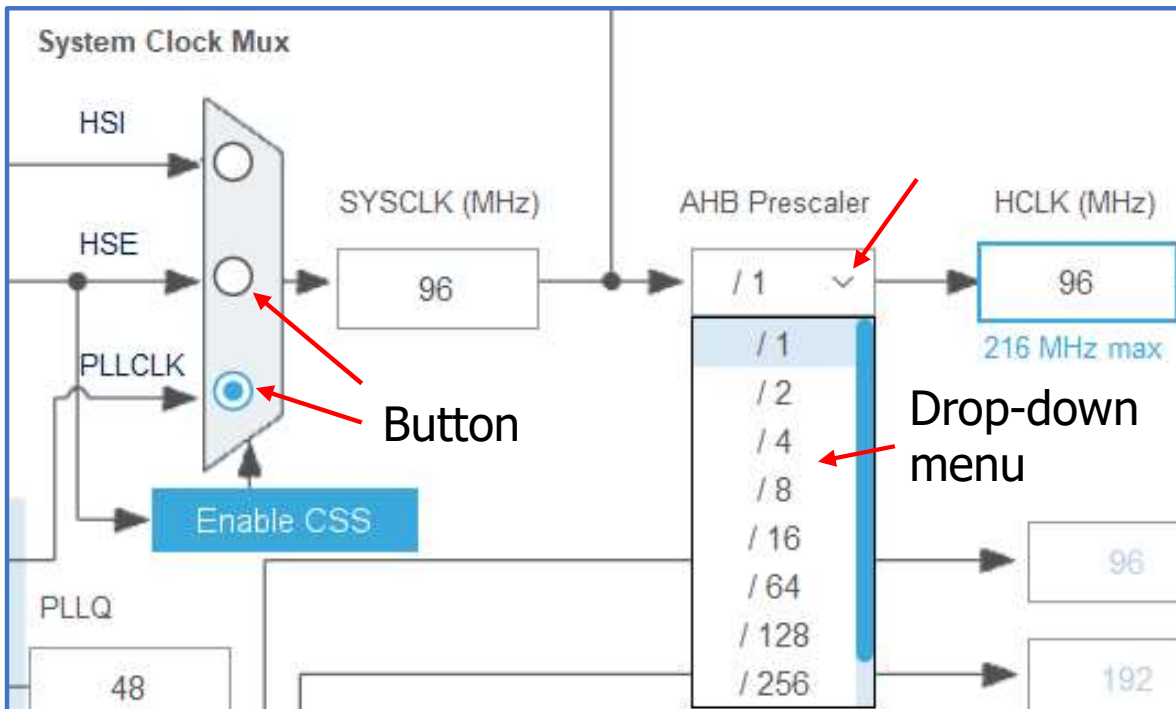
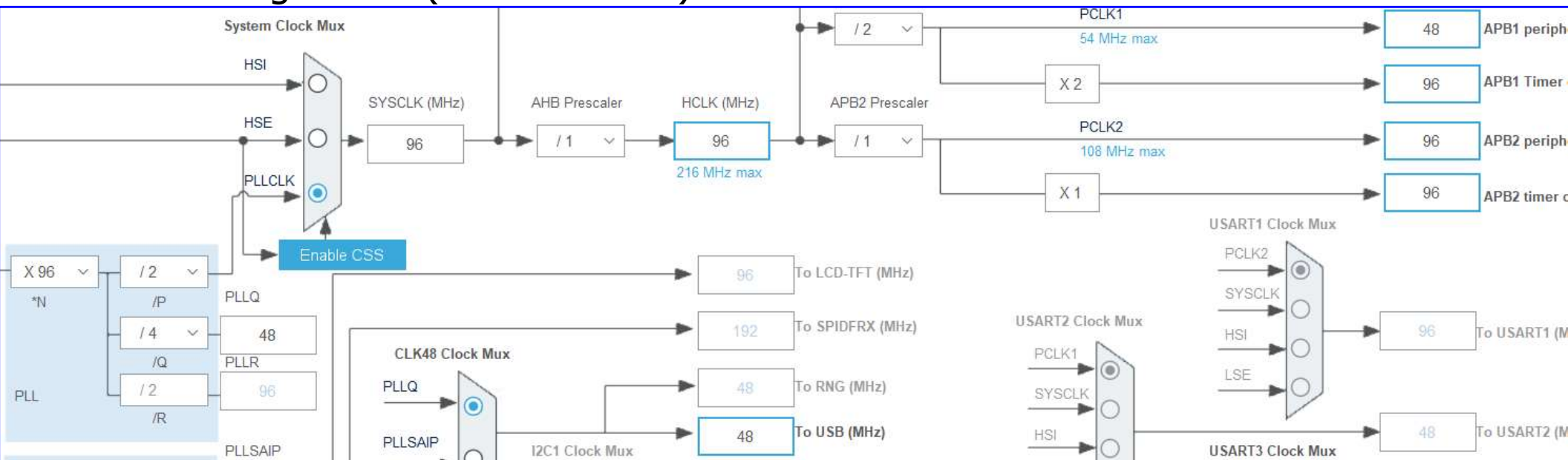
Acronym	Definition	Acronym	Definition
AHB	Advanced High-Performance Bus	Debug:	
APB	Advanced Peripheral Bus	AHB-AP	AHB Access Port
AXI	Advanced Extensible Interface	APB-AP	APB Access Port
AXIM	AXI Interconnect Matrix	AXI-AP	AXI Access Port
		AP	Access Port
Word	data of 32-bit length	FPB	Breakpoint Unit
Half-word	data of 16-bit length	CSTF	CoreSight Trace Funnel
Byte	data of 8-bit length	CTI	Cross Trigger Interface
Double word	data of 64-bit length.	CTM	Cross Trigger Matrix
Flash word	data of 256-bit length	DBGMCU	Microcontroller Debug Unit
		DWT	Data Watchpoint and Trace
CD	CPU Power and Clock Domain	ETF	Embedded Trace FIFO
CRS	Clock Recovery System	ETM	Embedded Trace Macrocell
IAP	In-Application Programming	ITM	Instrumentation Trace Macrocell
ICP	In-Circuit Programming	JTAG	Joint Test Action Group
PCROP	Proprietary Code Readout Protection	JTAG-DP	JTAG Debug Port
RDP	Readout Protection	TPIU	Trace Port Interface Unit
SRD	Smart Run Power and Clock Domain	SW-DP	Serial-Wire Debug Port
SYSCFG	System Configuration Controller	SWJ-DP	Serial-Wire and JTAG Debug Port
		SWD	Serial-Wire Debug
		SWO	Serial-Wire Output

STM32CubeIDE

Clock Configuration

- The **clock configuration tab** provides a **schematic overview** of the clock paths, along with all clock sources, dividers, and multipliers.
- Actual clock speeds are visible.
- **Active and enabled clock signals** are highlighted in **blue**.
- **Drop-down menus** and **buttons** serve to modify the actual clock configuration.
- If a configured value is out of bounds, it immediately turns **red** to **highlight a problem**.
- It also works the other way; **enter the required clock speed in a blue frame** and the software will attempt to reconfigure multipliers and dividers to provide the requested value.
- **Right-click** on a **clock value in blue to lock** it to prevent modifications.

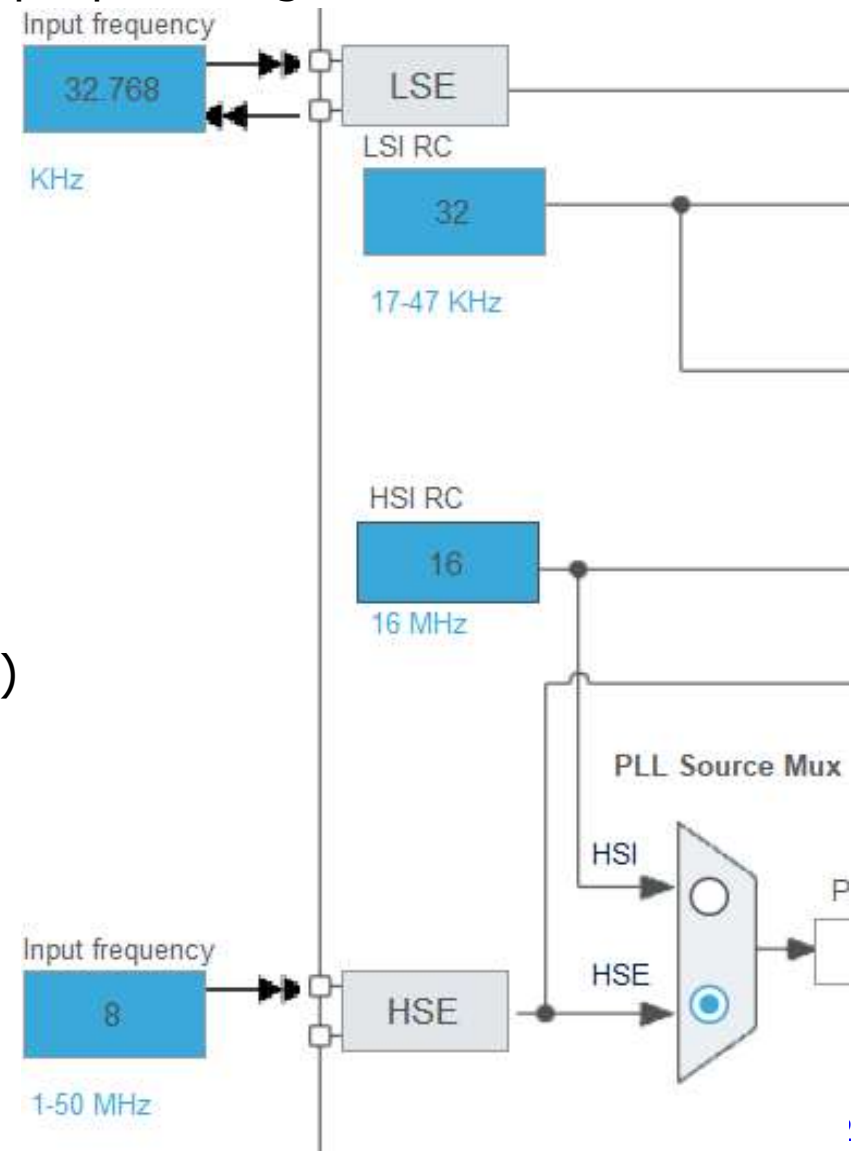
Clock Configuration (STM32F767ZI)



STM32CubeIDE

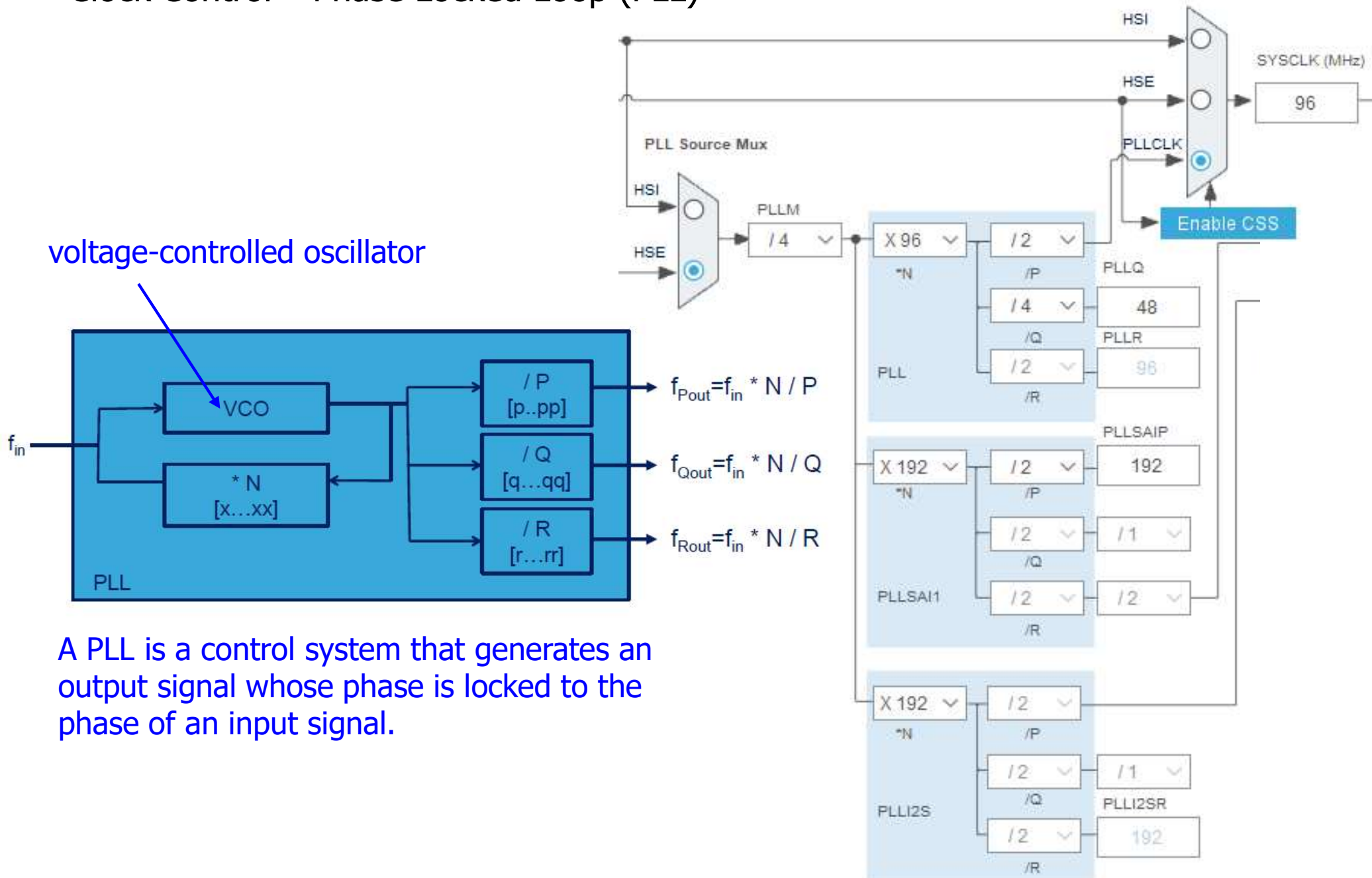
Clock Control

- All peripherals need clock from **RCC (Reset and Clock Controller)** to work properly.
- Without clocks, it is not possible to change or read peripheral registers.
- The clock sources:
 - HSI (16 MHz)
 - LSI (32 kHz)
 - HSE (8, 1 – 50 MHz)
 - LSE (32.768 kHz)
- **PLL** (Phase Locked Loop) from HSI or HSE
- **SYSCLK** for Ethernet PTP (Precision Time Protocol)
- System clock source multiplexer
 - **AHB (HCLK)**
 - **APB1 or APB2 (PCLK1 or PCLK2)**



Clock Control – Phase Locked Loop (PLL)

voltage-controlled oscillator



A PLL is a control system that generates an output signal whose phase is locked to the phase of an input signal.

HAL GPIO Generic Driver, Section 28, User Manual UM1905 , [Ref_08]

- GPIO HAL APIs are the following:

HAL_GPIO_Init()

HAL_GPIO_DeInit()

HAL_GPIO_ReadPin()

HAL_GPIO_WritePin()

HAL_GPIO_TogglePin()

HAL_GPIO_LockPin()

[Refer latest UM1905 for more and updated information.](#)

In addition to standard GPIO modes, the pin mode can be configured as **EXTI** with interrupt or event generation.

When selecting **EXTI** mode with interrupt generation, the program will call

HAL_GPIO_EXTI_IRQHandler() implemented in **stm32f7xx_it.c**


and the user must implement

HAL_GPIO_EXTI_Callback()

Structure field	Description (old document)
Pin	Specifies the GPIO pins to be configured. Possible values: GPIO_PIN_x or GPIO_PIN_All, where x[0..15]
Mode	Specifies the operating mode for the selected pins: GPIO mode or EXTI mode. Possible values are: <ul style="list-style-type: none"> <u>GPIO mode</u> <ul style="list-style-type: none"> GPIO_MODE_INPUT : Input floating GPIO_MODE_OUTPUT_PP : Output push-pull GPIO_MODE_OUTPUT_OD : Output open drain GPIO_MODE_AF_PP : Alternate function push-pull GPIO_MODE_AF_OD : Alternate function open drain GPIO_MODE_ANALOG : Analog mode <u>External Interrupt mode</u> <ul style="list-style-type: none"> GPIO_MODE_IT_RISING : Rising edge trigger detection GPIO_MODE_IT_FALLING : Falling edge trigger detection GPIO_MODE_IT_RISING_FALLING : Rising/Falling edge trigger detection <u>External Event mode</u> <ul style="list-style-type: none"> GPIO_MODE_EVT_RISING : Rising edge trigger detection GPIO_MODE_EVT_FALLING : Falling edge trigger detection GPIO_MODE_EVT_RISING_FALLING: Rising/Falling edge trigger detection
Pull	Specifies the Pull-up or Pull-down activation for the selected pins. Possible values are: GPIO_NOPULL GPIO_PULLUP GPIO_PULLDOWN


GPIO_InitTypeDef structure field

GPIOs (old document) - Examples

Structure field	Description
Speed	<p>Specifies the speed for the selected pins</p> <p>Possible values are:</p> <p>GPIO_SPEED_LOW  GPIO_SPEED_FREQ_LOW</p> <p>GPIO_SPEED_MEDIUM</p> <p>GPIO_SPEED_HIGH</p> <p>GPIO_SPEED_FREQ_VERY_HIGH</p> <p>GPIO_InitTypeDef structure field</p>

Please find below typical GPIO configuration examples:

- Configuring GPIOs as output push-pull to drive external LEDs

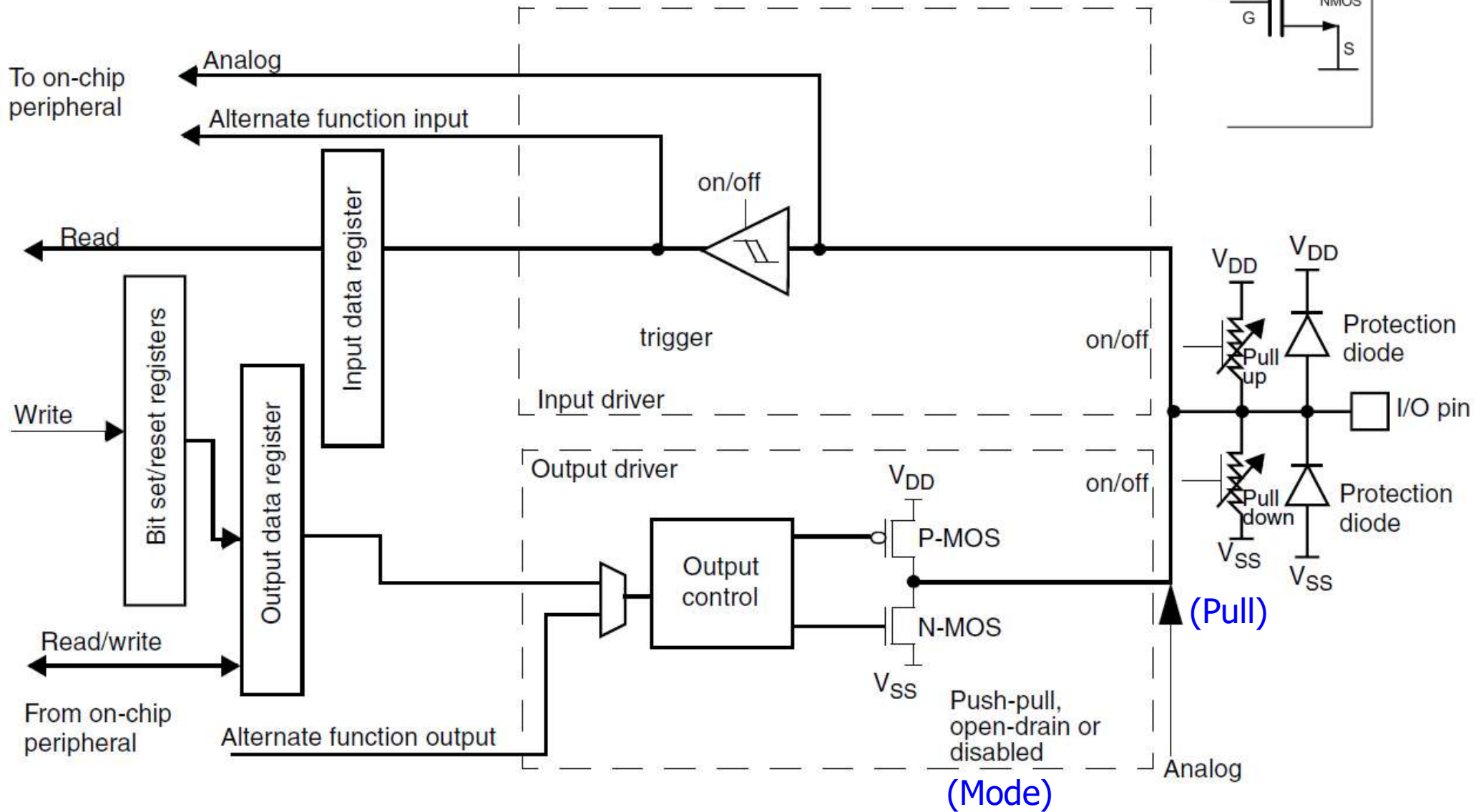
```
GPIO_InitStruct.Pin = GPIO_PIN_12 | GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_MEDIUM;  GPIO_SPEED_FREQ_MEDIUM
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
```

- Configuring PA0 as external interrupt with falling edge sensitivity:

```
GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;
GPIO_InitStructure.Pull = GPIO_NOPULL;
GPIO_InitStructure.Pin = GPIO_PIN_0;
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
```


STM32CubeMX

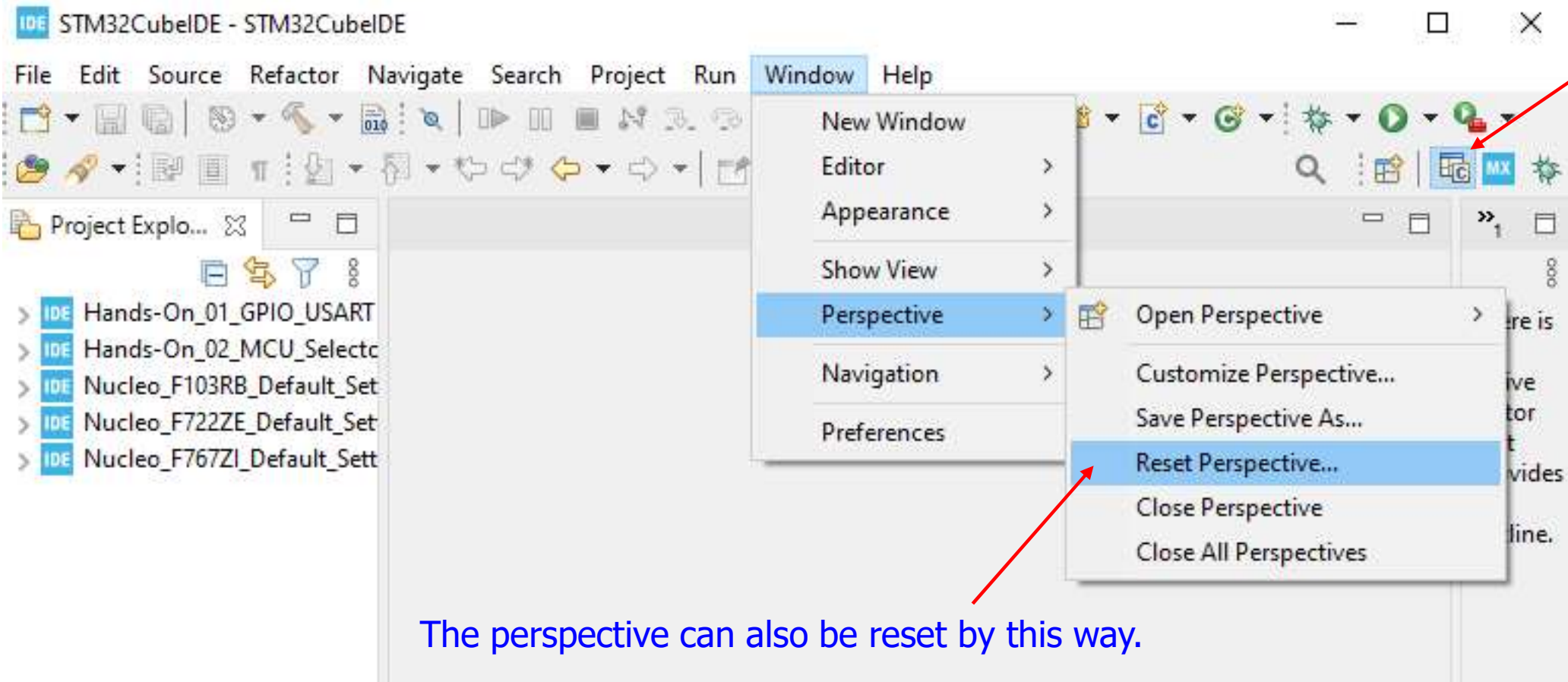
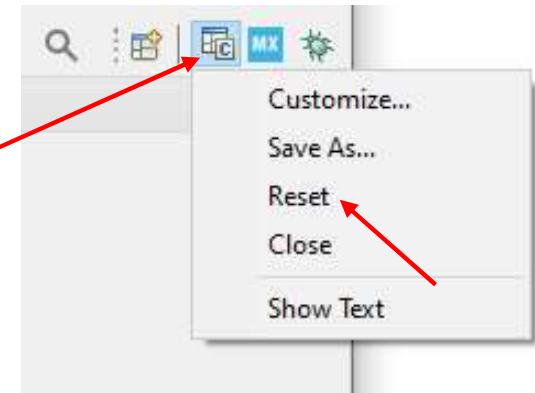
GPIOs (Reference Manual [Ref_03])



Basic structure of an I/O port bit

STM32CubeIDE: Reset Perspective

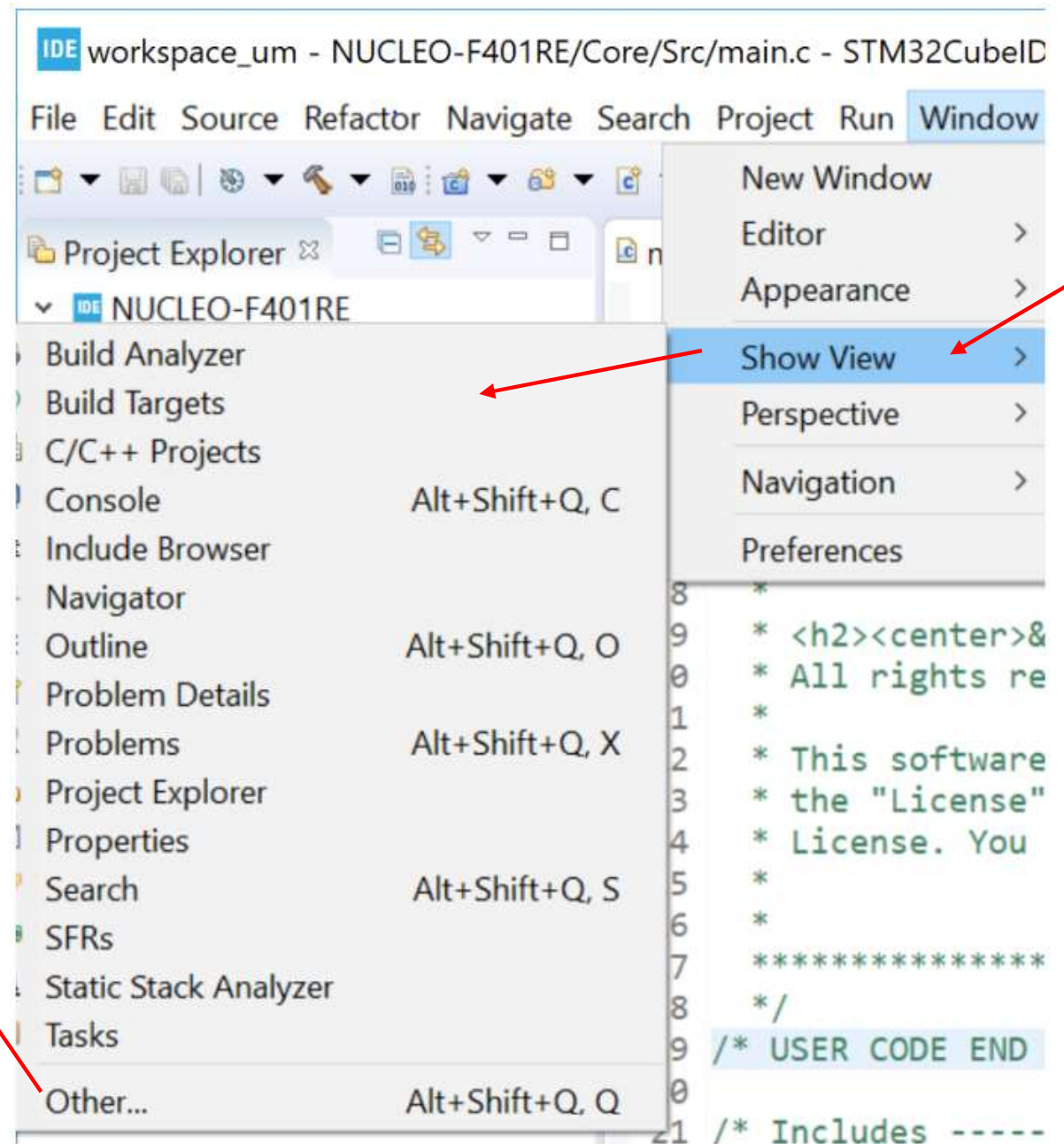
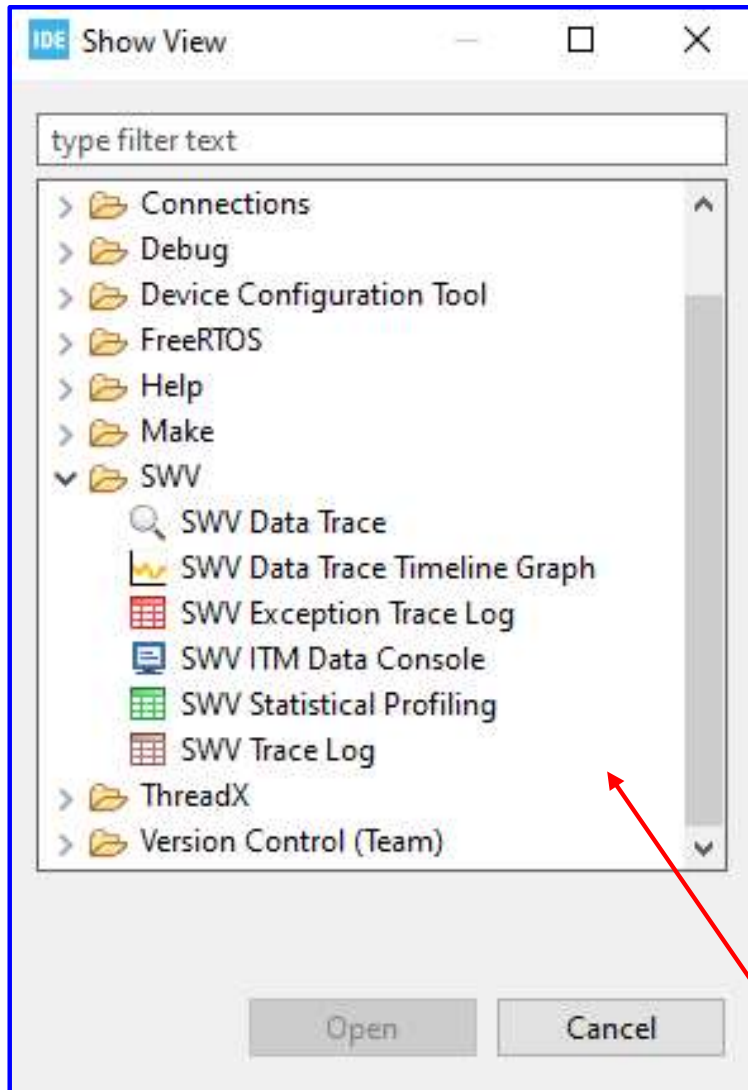
There is a need to reset perspective. The perspective is reset by right-clicking the **perspective icon** in the toolbar and selecting [**Reset**] from the list.



The perspective can also be reset by this way.

STM32CubeIDE: Views

Views can be opened from the [Window]>[Show View] menu by selecting one of the views in the list.



STM32CubeIDE

STM32CubeIDE: Keyboard Shortcuts

Press [**Ctrl+Shift+L**] to open the cheat sheet in Editor.

Activate Editor	F12
Add Block Comment	Ctrl+Shift+ /
Add Include	Ctrl+Shift+N
Align const qualifiers	Ctrl+Shift+A
Backward History	Alt+Left
Build All	Ctrl+B
Build Target Build	Shift+F9
Close	Ctrl+F4
Close All	Ctrl+Shift+F4
Collapse	Ctrl+Numpad_Subtract
Collapse All	Ctrl+Shift+Numpad_Divide
Content Assist	Ctrl+Space
Context Information	Ctrl+Shift+Space
Copy	Ctrl+Insert
Copy Lines	Ctrl+Alt+Down
Copy Qualified Name	Ctrl+Alt+Shift+C
Cut	Shift+Delete
Debug	F11
Declaration	Ctrl+G
Delete	Delete
Delete Line	Ctrl+D
Delete Next Word	Ctrl+Delete
Delete Previous Word	Ctrl+Backspace
Delete to End of Line	Ctrl+Shift+Delete


Press 'Ctrl+Shift+L' to open the preference page

Surround With Quick Menu	Alt+Shift+Z
Switch to Editor	Ctrl+Shift+E
Text End	Ctrl+End
Text Start	Ctrl+Home
To Lower Case	Ctrl+Shift+Y
To Upper Case	Ctrl+Shift+X
Toggle Block Selection	Alt+Shift+A
Toggle Breakpoint	Ctrl+Shift+B
Toggle Comment	Ctrl+Shift+C
Toggle Folding	Ctrl+Numpad_Divide
Toggle Full Screen	Alt+F11
Toggle Function - Refactoring	Alt+Shift+T
Toggle Insert Mode	Ctrl+Shift+Insert
Toggle Mark Occurrences	Alt+Shift+O
Toggle Overwrite	Insert
Toggle Source/Header	Ctrl+Tab
Toggle Split Editor (Horizontal)	Ctrl+_
Toggle Split Editor (Vertical)	Ctrl+{
Toggle Word Wrap	Alt+Shift+Y
Undo	Ctrl+Z
Use Step Filters	Shift+F5
Word Completion	Alt+/
Zoom In	Ctrl+=
Zoom Out	Ctrl+-

Press 'Ctrl+Shift+L' to open the preference page

STM32CubeIDE


KEYBOARD SHORCUT OVERVIEW

- CTRL+SHIFT+L List keyboard shortcuts List all defined keyboard shortcuts
- 

NAVIGATING FILES AND C SYMBOLS

- CTRL+SHIFT+R Open resource Find files from any perspective
- CTRL+H Search for keyword Search for keyword in defined scope. Possible to use reg.exp.
- ALT+ENTER View properties Views the properties for the selected resource.
- CTRL+PGUP/DN Switch editor Switches to an open editor to the left/right
- CTRL+E Select editor Move to open editor by filtering text or selecting in menu
- CTRL+SHIF+T Search for elements Search for elements (functions, symbols, ...) in Workspace resources

NAVIGATING INFORMATION IN FILES

- CTRL+O Quick Outline Navigate big files from perspectives lacking outline view
 - CTRL+L Go to Line Go to line in editor
 - CTRL+F Search inside context Search within the file currently active in editor
 - CTRL+ALT+I Open Include Browser Open include browser for the current resource
 - CTRL+ALT+H Open Call hierarchy See how function calls made to and from selected function
 - CTRL+SPACE Code completion Code completion/parameter hints depending on context
 - Parameter hints
- 

STM32CubeIDE

CODE FORMATTING AND REFACTORING

- | | | | |
|---|--------------|----------------------|--|
| • | SHIFT+ALT+A | Toggle block select | Edit one column across multiple rows |
| • | CTRL+I | Indent Line | Indent source code line according to defined format rules |
| • | CTRL+SHIFT+F | Format selected code | Format source code according to defined format rules |
| • | SHIFT+ALT+R | Quick renaming | Renames any C symbol across all files in all open projects |


VERSION CONTROL

- | | | | |
|---|------------|------------------|--|
| • | CTRL+ALT+C | Commit resources | Commits modified files within active context |
|---|------------|------------------|--|

DEBUGGING

- | | | | |
|---|-----|---------------|--|
| • | F11 | Debug project | Starts a debug session of project currently active |
|---|-----|---------------|--|

GOOD TO KNOW

- | | | |
|---|---------------------------------------|--|
| • | Window → Preferences → General → Keys | allows you to define your own keyboard shortcuts, also allows you to choose other keyboard shortcut schemes: Emacs, Microsoft VisualStudio, etc... |
|---|---------------------------------------|--|
- 

STM32CubeIDE

STM32CubeIDE: Keyboard Shortcuts

Examples

Command	Binding	Where
Copy	Ctrl+C	In dialogs and windows
Cut	Ctrl+X	In dialogs and windows
Paste	Ctrl+V	In dialogs and windows
Debug	F11	In windows
Open declaration	F3	In C/C++ editor
References	Ctrl+Shift+G	In C/C++ editor/views
Find and open files	Ctrl+Shift+R	In C/C++ editor/views
Toggle selection mode normal/block	Alt+Shift+A	In C/C++ editor/views
Zoom In	Ctrl++	Editing text
Zoom Out	Ctrl+-	Editing text

Editor zoom in and zoom out

It is possible to increase or decrease the default font size for text editors by pressing **Ctrl++** and **Ctrl+-**:

- **Ctrl++** : zoom in text [**Ctrl+Shift++**]
- **Ctrl+-** : zoom out text

*If a keyboard with a numeric keypad is used and the + or – keys are pressed on the numeric keypad, use the **Shift** key in addition to make the zoom work (**Ctrl+Shift+** or **Ctrl+Shift-**).*

STM32CubeIDE

STM32CubeIDE User Manual, **UM2609**, [Ref_11-6]:

- **Compare files**, section 1.8.6, pages 31 – 32,
- **Exporting projects**, section 2.9, pages 104 – 106, and
- **Importing existing projects**, section 2.10, pages 106 – 111.