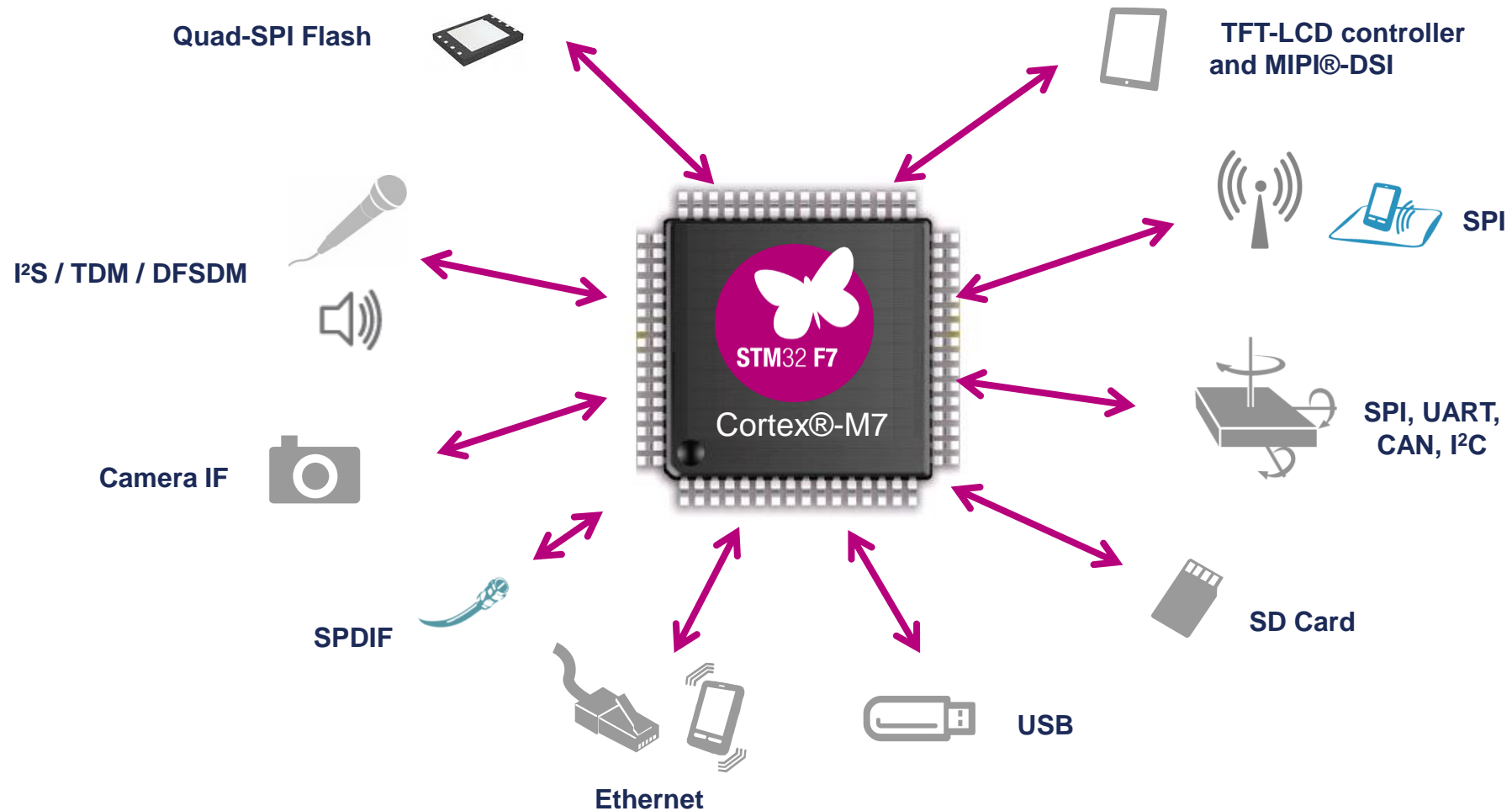


STM32 Overview



Ing. Davide Giacalone
Advanced Research Engineer, SRA

STM32 is more than a CPU



High Performance

STM32F2 398 CoreMark 120 MHz Cortex-M3	STM32F4 608 CoreMark 180 MHz Cortex-M4	STM32F7 1082 CoreMark 216 MHz Cortex-M7	STM32H7 Up to 3224 CoreMark Up to 550 MHz Cortex-M7 240 MHz Cortex-M4
---	---	--	---

Mainstream

STM32G0 142 CoreMark 64 MHz Cortex-M0+	STM32G4 ● 569 CoreMark 170 MHz Cortex-M4	
STM32F0 106 CoreMark 48 MHz Cortex-M0	STM32F1 177 CoreMark 72 MHz Cortex-M3	STM32F3 ● 245 CoreMark 72 MHz Cortex-M4

● Optimized for mixed-signal applications

Ultra-low-power

STM32L4+ 409 CoreMark 120 MHz Cortex-M4		STM32U5 651 CoreMark 160 MHz Cortex-M33	
STM32L0 75 CoreMark 32 MHz Cortex-M0+	STM32L1 93 CoreMark 32 MHz Cortex-M3	STM32L4 273 CoreMark 80 MHz Cortex-M4	STM32L5 443 CoreMark 110 MHz Cortex-M33

Wireless

STM32WL 162 CoreMark 48 MHz Cortex-M4 48 MHz Cortex-M0+	STM32WB ● 216 CoreMark 64 MHz Cortex-M4 32 MHz Cortex-M0+
---	---

● Cortex-M0+ Radio co-processor

- ART Accelerator™
- SDIO
- USART, SPI, I2C
- PS + 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

Product line	F _{CPU} (MHz)	Flash (KB)	RAM (KB)	Ethernet I/F IEEE 1588	2x CAN	Camera I/F	SDRAM I/F	Dual Quad-SPI	SAI	SPDIF RX	Chrom-ART Graphic Accelerator™	TFT LCD Controller	MIPI DSI
Advanced lines													
STM32F469 ²	180	512 K to 2056 K	384	•	•	•	•	•	•		•	•	•
STM32F429 ²	180	512 K to 2056 K	256	•	•	•	•		•		•	•	
STM32F427 ²	180	1024 K to 2056 K	256	•	•	•	•		•		•		
Foundation lines													
STM32F446	180	256 K to 512 K	128		•	•	•	•	•	•			
STM32F407 ²	168	512 K to 1024 K	192	•	•	•							
STM32F405 ²	168	512 K to 1024 K	192		•								
Product line	F _{CPU} (MHz)	Flash (KB)	RAM (KB)	RUN current (µA/MHz)	STOP current (µA)	Small package (mm)	FSMC (NOR/PSRAM/LCD) support	QSPI	DFSDM	DAC	TRNG	DMA Batch Acquisition mode	USB 2.0 OTG FS
Access lines													
STM32F401	84	128 K to 512 K	up to 96	Down to 128	Down to 10	Down to 3x3							•
STM32F410	100	64 K to 128 K	32	Down to 89	Down to 6	Down to 2.553x 2.579				•	•	BAM	-
STM32F411	100	256 K to 512 K	128	Down to 100	Down to 12	Down to 3.034x 3.22						BAM	•
STM32F412	100	512 K to 1024 K	256	Down to 112	Down to 18	Down to 3.653x 3.651	•	•	•		•	BAM	• +LPM ¹
STM32F413 ²	100	1024 K to 1536 K	320	Down to 115	Down to 18	Down to 3.951x 4.039	•	•	•	•	•	BAM	• +LPM ¹

Note: 1. Link Power Management

2. The same devices are also found with embedded HW AES encryption (128/256-bit)

STM32 Tools

- **Hardware Development Tools**

- STM32 Nucleo boards
- STM32 Discovery kits
- STM32 Eval boards

- **STM32 Docs**

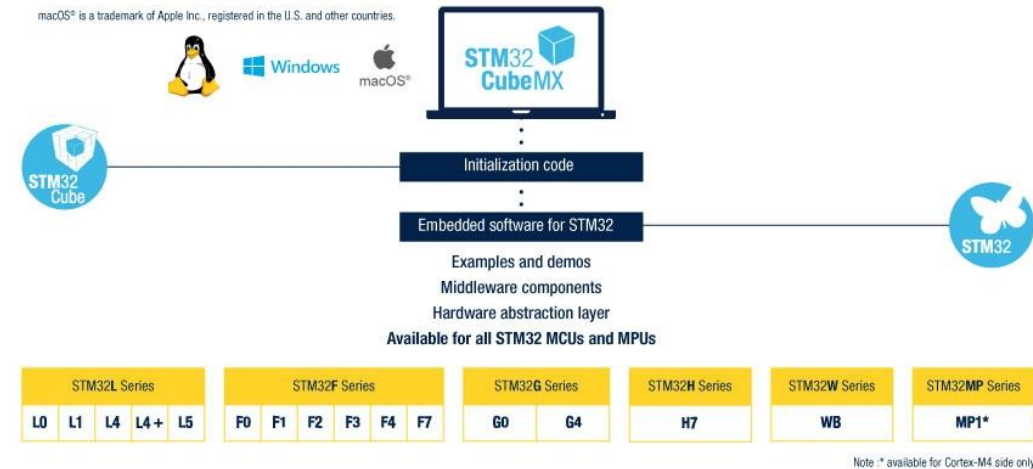
- Datasheet
- Reference Manual

- **STM32CubeMX**

- Graphical tool
- Easy configuration of STM32 microcontrollers peripherals
- C code generation, compliant with STM32 MCU

- **STM32CubeIDE**

- C/C++ development platform
- Code generation, compilation, and debug for STM32 microcontrollers



STM32 boards

- Developing your own board
- Using existing STMicroelectronics boards
 - STM32 Nucleo boards
 - STM32 Discovery kits
 - STM32 Eval boards

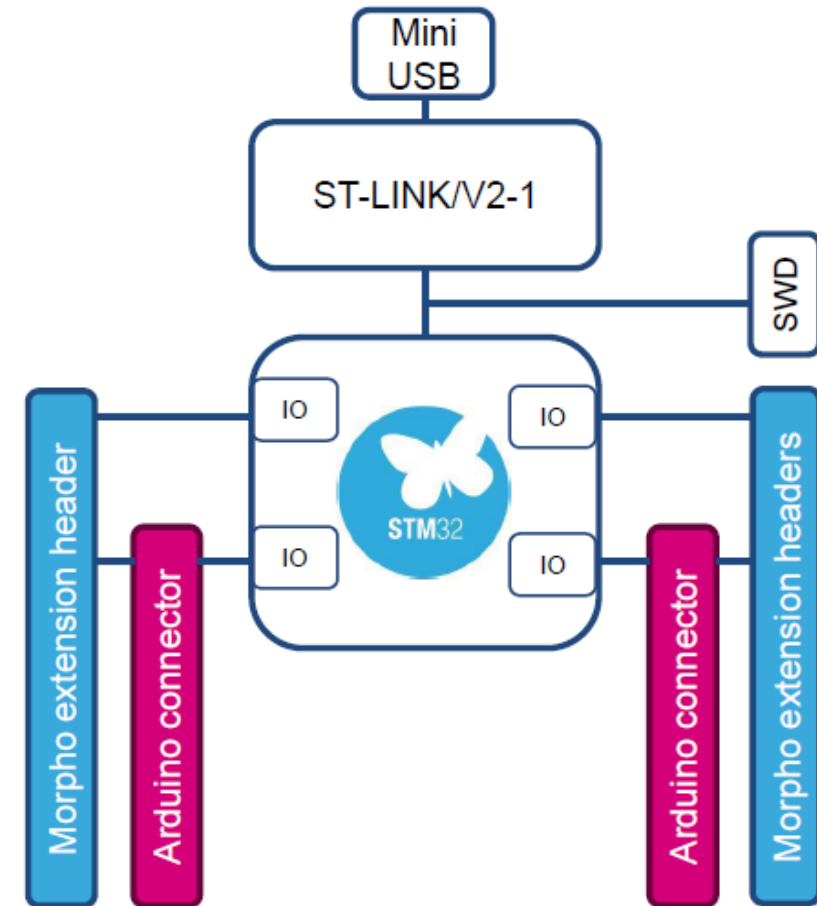


Board	ST-Link	ARDUINO connector	Display
Custom	No		
STM32 Nucleo	Yes (V2.1/V3)	x	
STM32 Discovery	Yes (V2.1/V3)	x	x
STM32 Eval	Yes (V2.1/V3)	x	x

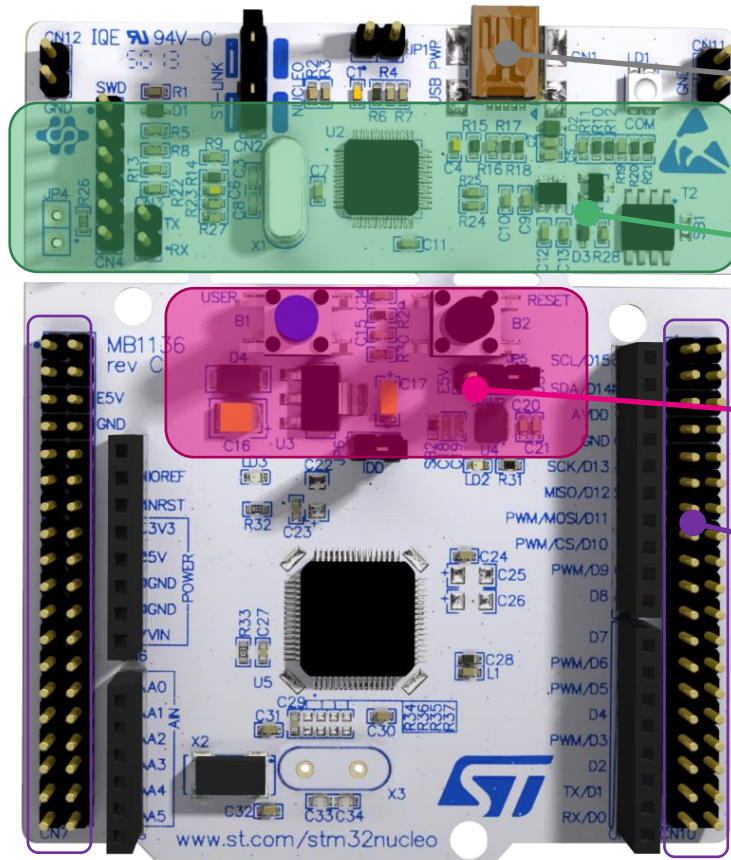
STM32 Nucleo Structure



- Each Nucleo board is based on one of the STM32 MCU product lines
- Two types of extension resources :
 - **Arduino** Uno v3 connectivity.
 - **Morpho** headers for easy access to all MCU peripherals.
- Integrated ST-LINK/V2-1 debugger and programmer :
 - supports drag-and-drop flash programming.
 - can target on-board STM32 or external STM32-based application.



STM32 Nucleo key assets



Flexible board power supply
Through USB or external source

Integrated ST-Link/V2-1
Mass storage device flash programming

2 push buttons, 2 color LEDs

Direct access to all STM32 I/Os:
through Morpho extension headers

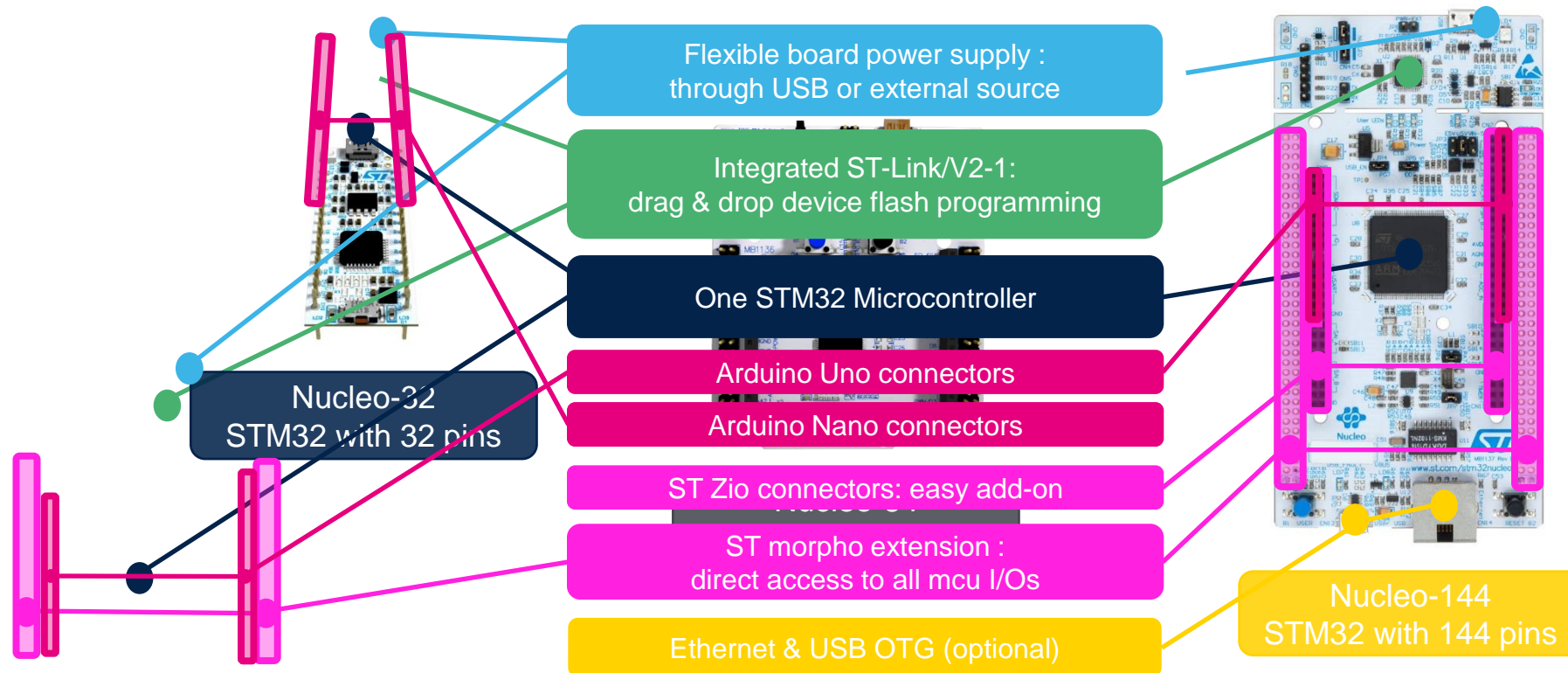
Flexible prototyping

Unlimited expansion capabilities

Simply expose the whole STM32 portfolio to the communities



Enlarging the STM32Nucleo family to cover whole STM32 portfolio



STM32 complete product range from ultra-low power to high performance



Examples of Nucleo Expansion boards

Sense



Motion & environmental
sensors

Proximity sensor

Microphone

Connect



BLE

Wi-Fi

Sub-GHz

NFC

Power
Drive



Power management

LED boost

Move
Actuate



Motor drive

Actuator

Translate



Audio amplifier

OpAmp

Motion &
Environmental
sensors



BLE



Dynamic NFC



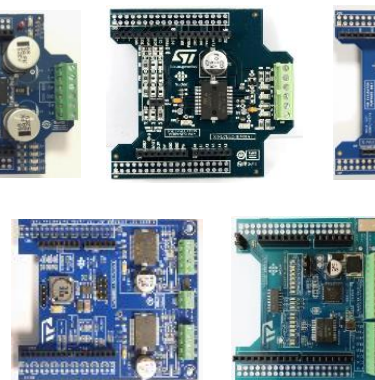
LoRa



LED driver



Motor driver



Audio Microphones

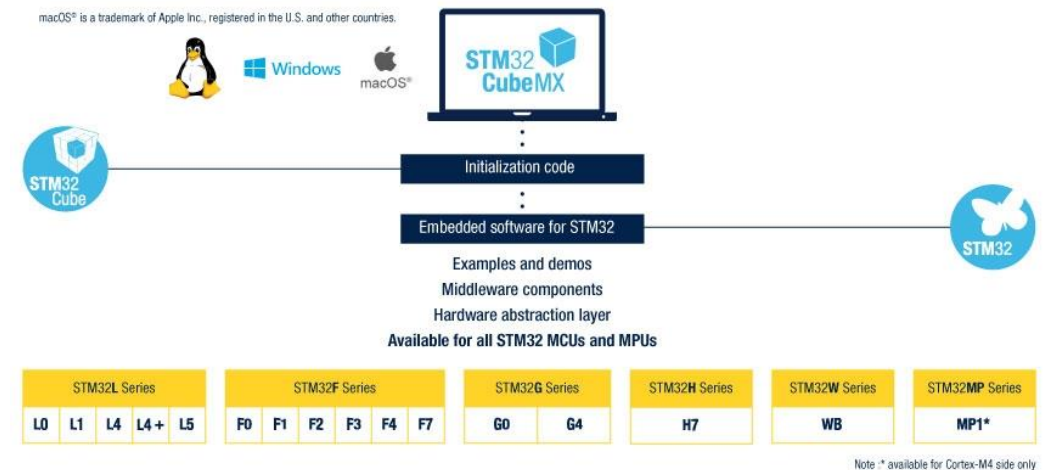


Security



For more information on STM32 Nucleo expansion boards: <http://www.st.com/en/ecosystems/stm32-nucleo-expansion-boards.html>

- Updater settings and manage embedded software packages
- Create new project (standalone MCU or board)
- Pinout
- Peripheral configuration
- Clock tree
- NVIC configuration
- Additional Software
- Project generation



Create new project

The screenshot displays the STM32CubeMX software interface. At the top, there is a menu bar with 'File', 'Window', and 'Help'. Below the menu bar is a 'Home' button. The main workspace is divided into three panels. The left panel, titled 'Existing Projects', contains a list of 'Recent Opened Projects' and a section for 'Other Projects'. The 'Recent Opened Projects' list includes 'Dotara.ioc', 'FW.ioc', 'Stand Alone.ioc', 'Nucleo-H743ZI settings.ioc', and 'Stand Alone.ioc', each with a last modified date and a small 'MX' icon. The right panel, titled 'New Project', contains a section 'I need to :' with three options: 'Start My project from MCU', 'Start My project from ST Board', and 'Start My project from Example'. Each option has a corresponding 'ACCESS TO [selector] SELECTOR' button. The bottom panel, titled 'Manage software installations', contains a section 'Check for STM32CubeMX and embedded software packages...' with a 'CHECK FOR UPDATES' button, and a section 'Install or remove embedded software packages' with an 'INSTALL / REMOVE' button. At the bottom of the interface, there is a banner for 'Build your certified safety system with STM32 and STM8' featuring logos for 'SIL Ready', 'ASIL Ready', 'ClassB Ready', and 'Partner Program'.

MCUs selector

Board selector

Create new project – MCUs selector

Link to Documentation and web page

You can filter MCUs with:

- Part Number
- Core
- Series
- Lines
- Peripherals
-



New Project from a MCU/MPU

MCU/MPU Selector Board Selector Example Selector Cross Selector

MCU/MPU Filters

Part Number

Core >

Series >

Line >

Package >

Other >

Peripheral >

Features Block Diagram Docs & Resources Datasheet Buy Start Project

★ STM32F4 Series

STM32F401RE STM32 Dynamic Efficiency MCU, Arm Cortex-M4 core with DSP and FPU, up to 512 Kbytes of Flash memory, 84 MHz CPU, Art Accelerator

ACTIVE Active
Product is in mass production

Unit Price for 10kU (US\$): 2.745

Board: [NUCLEO-F401RE](#)

LQFP64

The STM32F401xD/xE devices are based on the high-performance ARM®Cortex® -M4 32-bit RISC core operating at a frequency of up to 84 MHz. Its Cortex®-M4 core features a Floating point unit (FPU) single precision which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. The STM32F401xD/xE incorporate high-speed embedded memories (512 Kbytes of Flash memory, 96 Kbytes of SRAM), and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB buses and a 32-bit multi-AHB bus matrix. All devices offer one 12-bit ADC, a low-power RTC, six general-purpose 16-bit timers including one PWM timer for motor control, two general-purpose 32-bit timers. They also feature standard and advanced communication interfaces.

Features

MCUs/MPUs List: 201 items [Display similar items](#) [Export](#)

	Part No	Reference	Marketing Status	Unit Price for 10k...	Board	Package	Flash	RAM	IO	Freq.
★	STM32F401CD	STM32F401...	Active	2.384		UFQFPN48	384 kBytes	96 kBytes	36	84 MHz
★	STM32F401CE	STM32F401...	Active	2.639		UFQFPN48	512 kBytes	96 kBytes	36	84 MHz
★	STM32F401RB	STM32F401...	Active	1.979		LQFP64	128 kBytes	64 kBytes	50	84 MHz
★	STM32F401RC	STM32F401...	Active	2.235		LQFP64	256 kBytes	64 kBytes	50	84 MHz
★	STM32F401RD	STM32F401...	Active	2.49		LQFP64	384 kBytes	96 kBytes	50	84 MHz
★	STM32F401RE	STM32F401...	Active	2.745	NUCLEO-F401RE	LQFP64	512 kBytes	96 kBytes	50	84 MHz
★	STM32F401VB	STM32F401...	Active	2.32		UFBGA100	128 kBytes	64 kBytes	81	84 MHz
★	STM32F401VC	STM32F401...	Active	2.575		UFBGA100	256 kBytes	64 kBytes	81	84 MHz
★	STM32F401VD	STM32F401...	Active	2.831		UFBGA100	384 kBytes	96 kBytes	81	84 MHz
★	STM32F401VE	STM32F401...	Active	3.086		UFBGA100	512 kBytes	96 kBytes	81	84 MHz

Create new project – Board selector

Link to Documentation and web page



You can filter Boards with:

- Part Number
- Vendor
- Type
- MCU series



New Project from a MCU/MPU

MCU/MPU Selector Board Selector Example Selector Cross Selector

Board Filters

Commercial Part Number

Vendor

Type

Check/Uncheck All

- ☐ Discovery Kit
- ☐ Evaluation Board
- ☐ Evaluation Kit
- ☐ Nucleo USB Dongle
- ☐ Nucleo-144
- ☐ Nucleo-32
- ☒ Nucleo-64
- ☐ Nucleo-RF Kit

MCU/MPU Series

Check/Uncheck All

- ☐ STM32F0
- ☐ STM32F1
- ☐ STM32F2
- ☐ STM32F3
- ☒ STM32F4
- ☐ STM32F7
- ☐ STM32G0
- ☐ STM32G4
- ☐ STM32H7

Features Large Picture Docs & Resources Datasheet Buy Start Project

STM32F4 Series

NUCLEO-F401RE **STMicroelectronics NUCLEO-F401RE Board Support and Examples**

ACTIVE Active
Product is in mass production

Part Number : NUCLEO-F401RE
Commercial Part Number : NUCLEO-F401RE

Unit Price (US\$) : 13.0
Mounted Device : [STM32F401RETx](#)

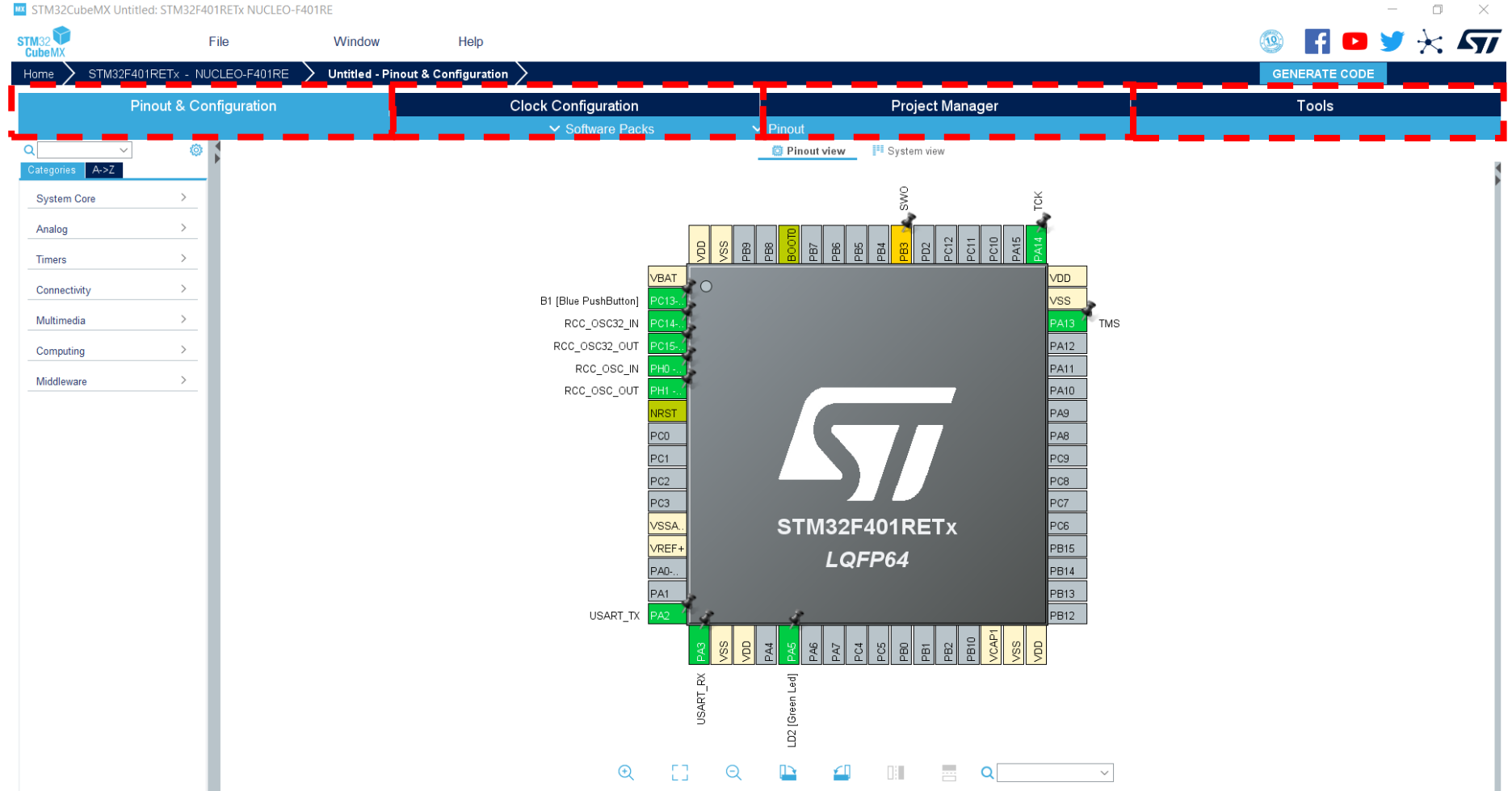
The STM32 Nucleo-64 boards provide an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. For the compatible boards, the external SMPS significantly reduces power consumption in Run mode. The Arduino™ Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields. The STM32 Nucleo-64 board does not require any separate probe as it integrates the ST-LINK debugger/programmer. The STM32 Nucleo-64 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Boards List: 4 items

	Overview	Commercial Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		NUCLEO-F401RE	Nucleo-64	Active	13.0	STM32F401RETx
☆		NUCLEO-F410RB	Nucleo-64	Active	13.0	STM32F410RBTx
☆		NUCLEO-F411RE	Nucleo-64	Active	13.0	STM32F411RETx
☆		NUCLEO-F446RE	Nucleo-64	Active	14.0	STM32F446RETx

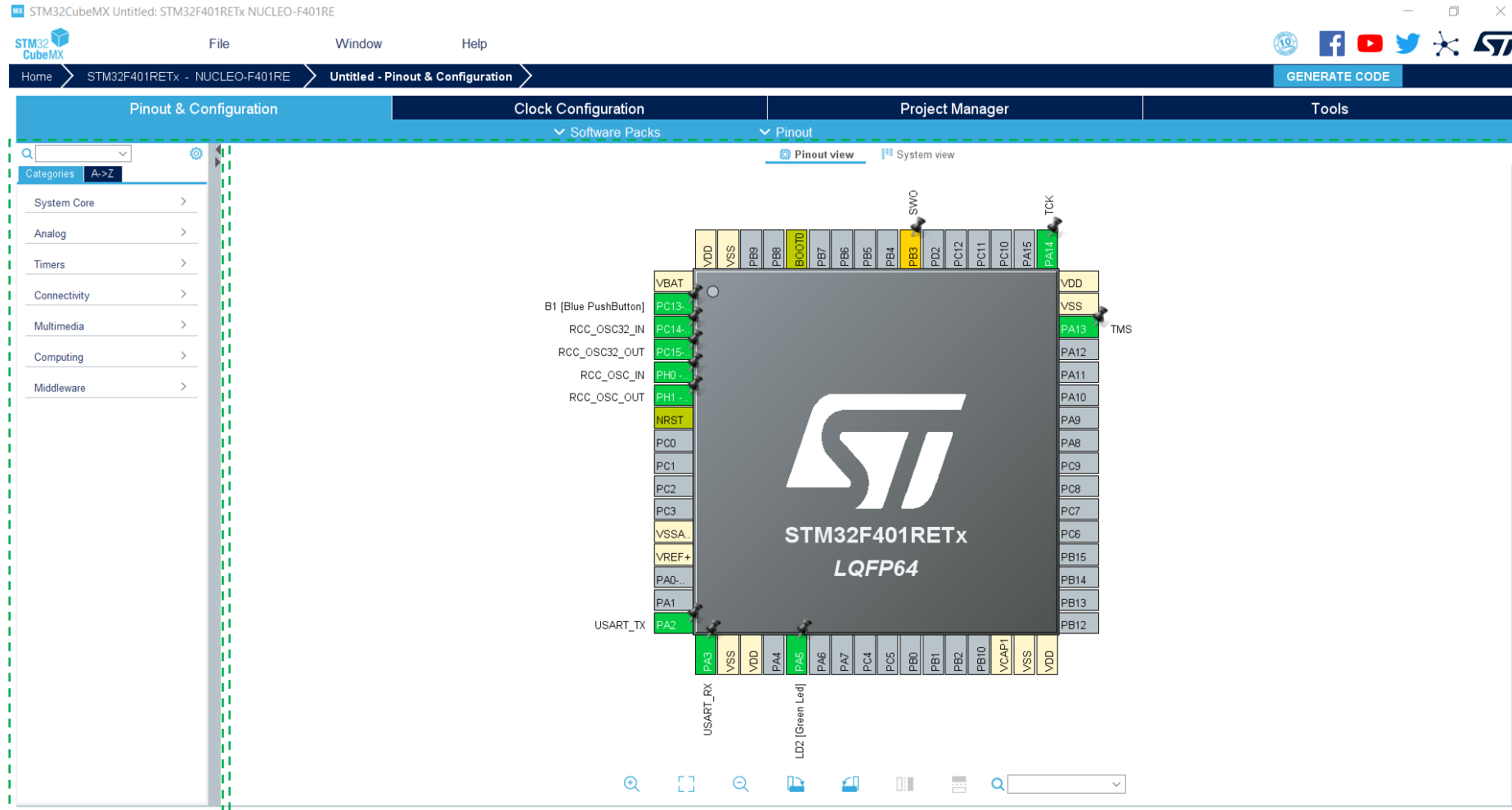
CubeMX view

- Pinout & Configuration
- Clock Configuration
- Project Manager
- Tools



Pinout & Configuration view

Peripherals
Configuration
section



MCU pinout
section

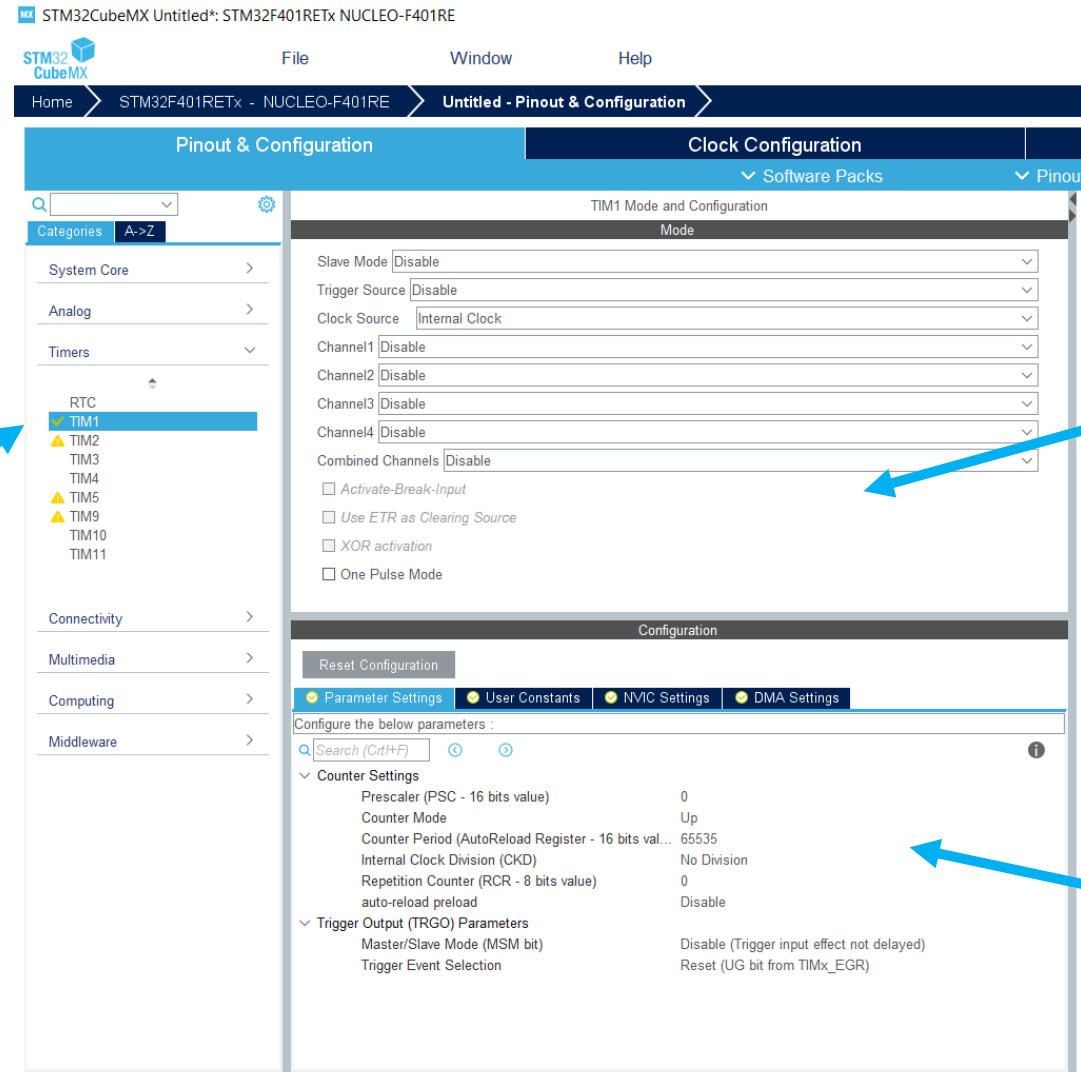
Peripherals Configuration section (set parameters)

From the **Peripherals Configuration section**, the user can select the peripherals from a list and configure the peripheral modes required for the application. STM32CubeMX assigns and configures the pins accordingly.

Peripheral selection

Peripheral mode selection

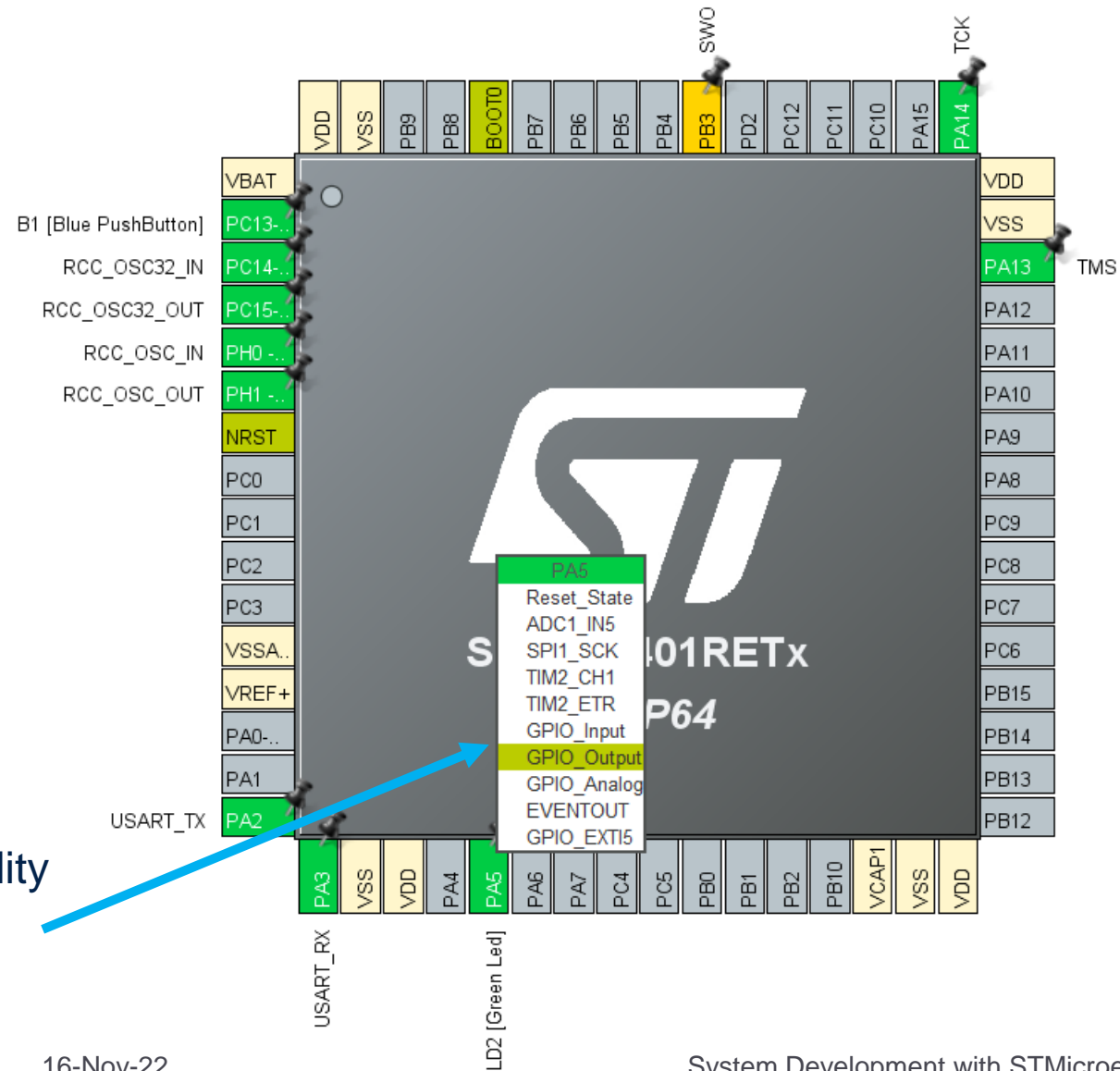
Peripheral parameters setting



MCU pinout section

For more advanced users, it is also possible to directly map a peripheral function to a physical pin using the Chip view. The signals can be locked on pins to prevent STM32CubeMX conflict solver from moving the signal to another pin.

Pin functionality assignment



Peripherals Configuration section (set interrupts)

Interrupts Configuration

System Interrupts

Interrupts list

Interrupts activation

Preemption and sub priority settings

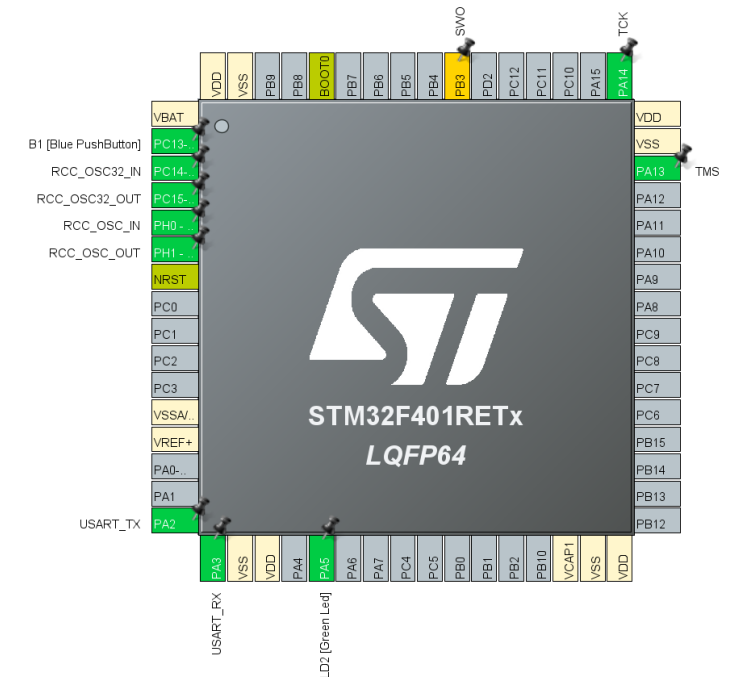
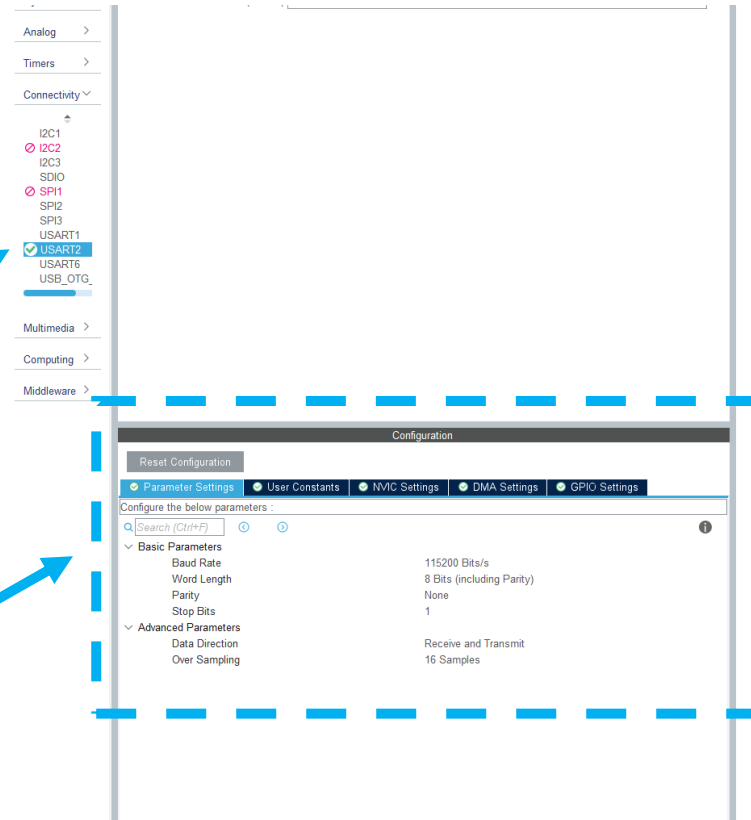
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Pre-fetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
TIM2 global interrupt	<input type="checkbox"/>	0	0
USART2 global interrupt	<input type="checkbox"/>	0	0
EXTI line[15:10] interrupts	<input type="checkbox"/>	0	0
FPU global interrupt	<input type="checkbox"/>	0	0

Hello World on uart (1/2)

- On CubeMx, check Parameter Settings values for Usart2

Usart2

Settings



Hello World on uart (2/2)

```
56 /* USER CODE BEGIN PFP */  
57  
58 int _write(int fd, char* ptr, int len) {  
59     HAL_UART_Transmit(&huart2, (uint8_t *) ptr, len, HAL_MAX_DELAY);  
60     return len;  
61 }  
62
```

```
/* USER CODE BEGIN 2 */
```

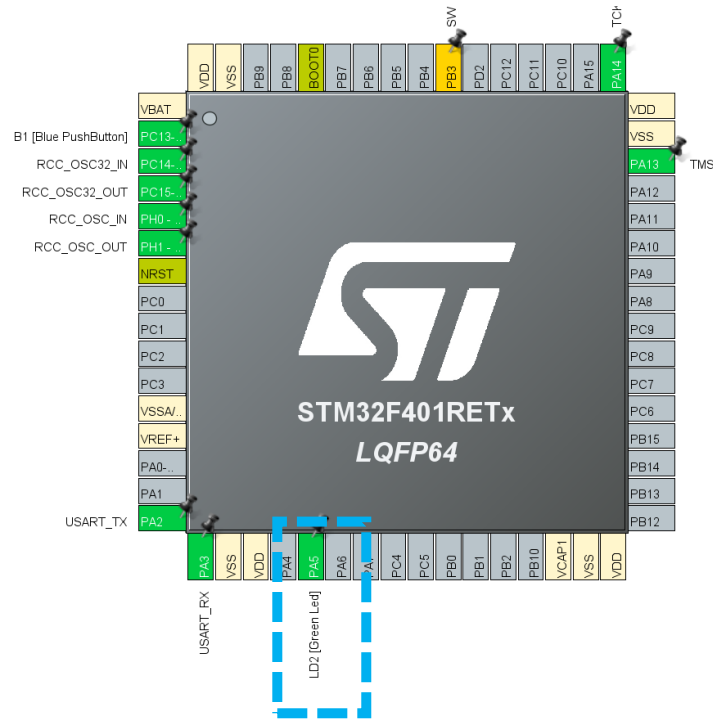
```
printf("hello\r\n");
```

- On STM32CubeIDE:
 - Define the _write function
 - Write the printf function

Led blinking, solution 1 (not precise)

- On CubeMx, verify PA5 is set as GPIO_output

- Toggle the pin status
- Add a delay



```
90  /* USER CODE BEGIN 2 */
91
92  /* USER CODE END 2 */
93
94  /* Infinite loop */
95  /* USER CODE BEGIN WHILE */
96  while (1)
97  {
98      HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
99      HAL_Delay(500);
100
101      /* USER CODE END WHILE */
102
103      /* USER CODE BEGIN 3 */
104  }
105  /* USER CODE END 3 */
```

Led blinking, with timer and interrupt (1/2)

- Enable timer 10 and set its params
- Enable the related interrupt

1. Click on TIM10

2. Enable the flag

3. Set params

The screenshot shows the STM32CubeMX interface. On the left, the 'Timers' category is expanded, and TIM10 is selected. A blue arrow points to TIM10. In the main area, the 'TIM10 Mode and Configuration' window is open. The 'Mode' tab is selected, and the 'Activated' checkbox is checked. A blue arrow points to the 'Activated' checkbox. Below the 'Mode' tab, the 'Configuration' tab is selected, and the 'Counter Settings' section is expanded. A blue dashed box highlights the 'Counter Settings' section, which includes parameters like Prescaler (PSC - 16 bits value), Counter Mode, Counter Period (AutoReload Register - 16 bits value), Internal Clock Division (CKD), and auto-reload preload. A blue arrow points to the 'Counter Settings' section.

4. Open the NVIC Settings and enable the tim10 global interrupt

The screenshot shows the 'NVIC Interrupt Table' in the STM32CubeMX interface. The 'NVIC Settings' tab is selected. The table lists the 'TIM1 update interrupt and TIM10 global interrupt'. The 'Enabled' checkbox is checked, and the 'Preemption Priority' and 'Sub Priority' are both set to 0. A blue arrow points to the 'Enabled' checkbox, and another blue arrow points to the 'TIM1 update interrupt and TIM10 global interrupt' row.

Interrupt	Enabled	Preemption Priority	Sub Priority
TIM1 update interrupt and TIM10 global interrupt	<input checked="" type="checkbox"/>	0	0

Led blinking, with timer and interrupt (2/2)

```
55 /* USER CODE BEGIN PFP */
56
57 volatile uint8_t time_elapsed = 0;
58
59 /* USER CODE END PFP */
```

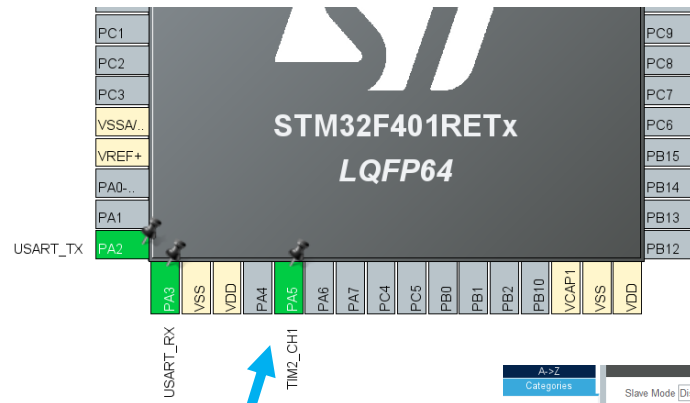
```
96 /* USER CODE BEGIN 2 */
97
98 HAL_TIM_Base_Start_IT(&htim10);
99
100 /* USER CODE END 2 */
```

```
262 /* USER CODE BEGIN 4 */
263
264 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
265 {
266     if(htim == &htim10)
267     {
268         time_elapsed = 1;
269     }
270 }
271
272
273 /* USER CODE END 4 */
```

```
100 /* USER CODE END 2 */
101
102 /* Infinite loop */
103 /* USER CODE BEGIN WHILE */
104 while (1)
105 {
106     if(time_elapsed)
107     {
108         time_elapsed = 0;
109         HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
110     }
111
112 /* USER CODE END WHILE */
```

- Declare a global flag
- Make the timer 10 start in interrupt mode
- Redefine a callback function related to the elapsed time and set the flag
- Manage the flag in the main loop

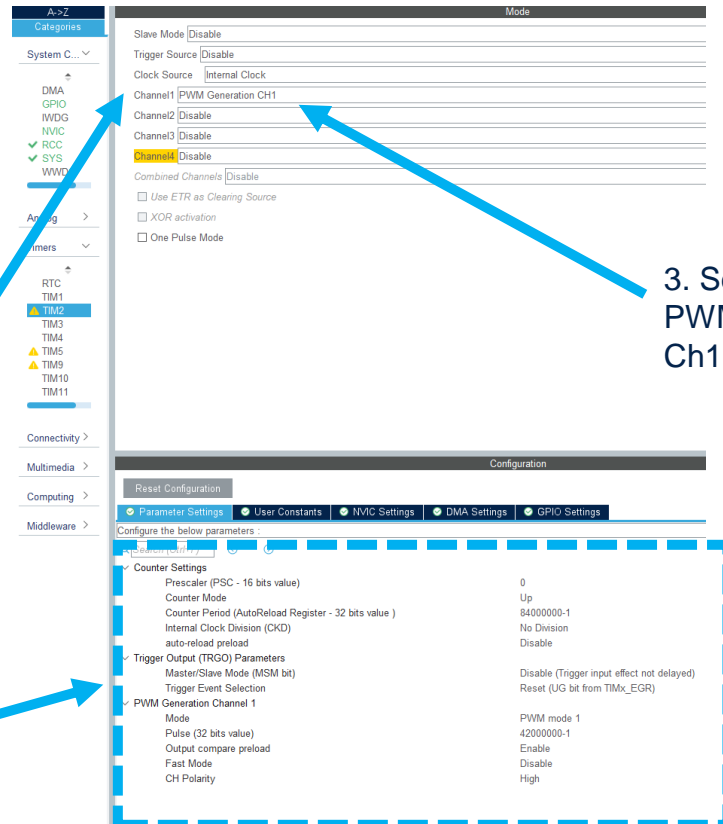
Led blinking, with PWM



1. Set PA5 as TIM2_CH1

2. Enable Clock Source

4. Adjust Counter period and Pulse values



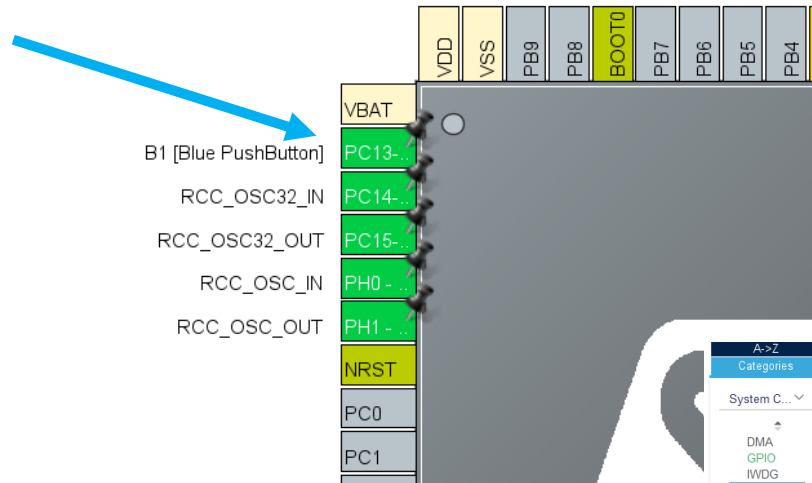
3. Set channel 1 as PWM Generation Ch1

```
104  
105 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  
106  
107 /* USER CODE END 2 */  
108
```

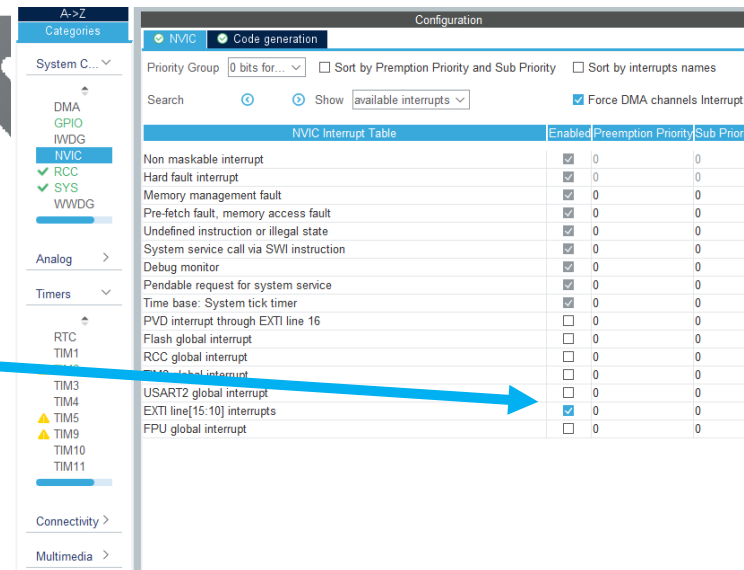
5. In the c code, let the timer 2 pwm start
Remember to remove all the references to LD2

Button Interrupt (1/2)

1. Check the PC13 pin is set as GPIO_EXTI13



2. Enable the EXTI line[15:10] interrupts



Button Interrupt (2/2)

```
46 /* USER CODE BEGIN PV */
47 volatile uint8_t button_pressed = 0;
```

```
298 /* USER CODE BEGIN 4 */
299
300 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
301 {
302     /* Prevent unused argument(s) compilation warning */
303     if(GPIO_Pin == GPIO_PIN_13)
304     {
305         button_pressed = 1;
306     }
307 }
```

```
109 /* Infinite loop */
110 /* USER CODE BEGIN WHILE */
111 while (1)
112 {
113     if(button_pressed)
114     {
115         button_pressed = 0;
116         printf("yes, you pressed a button!\r\n");
117     }
118 }
```

- Declare a global flag
- Redefine a callback function related to the pressed button and set the flag
- Manage the flag in the main loop