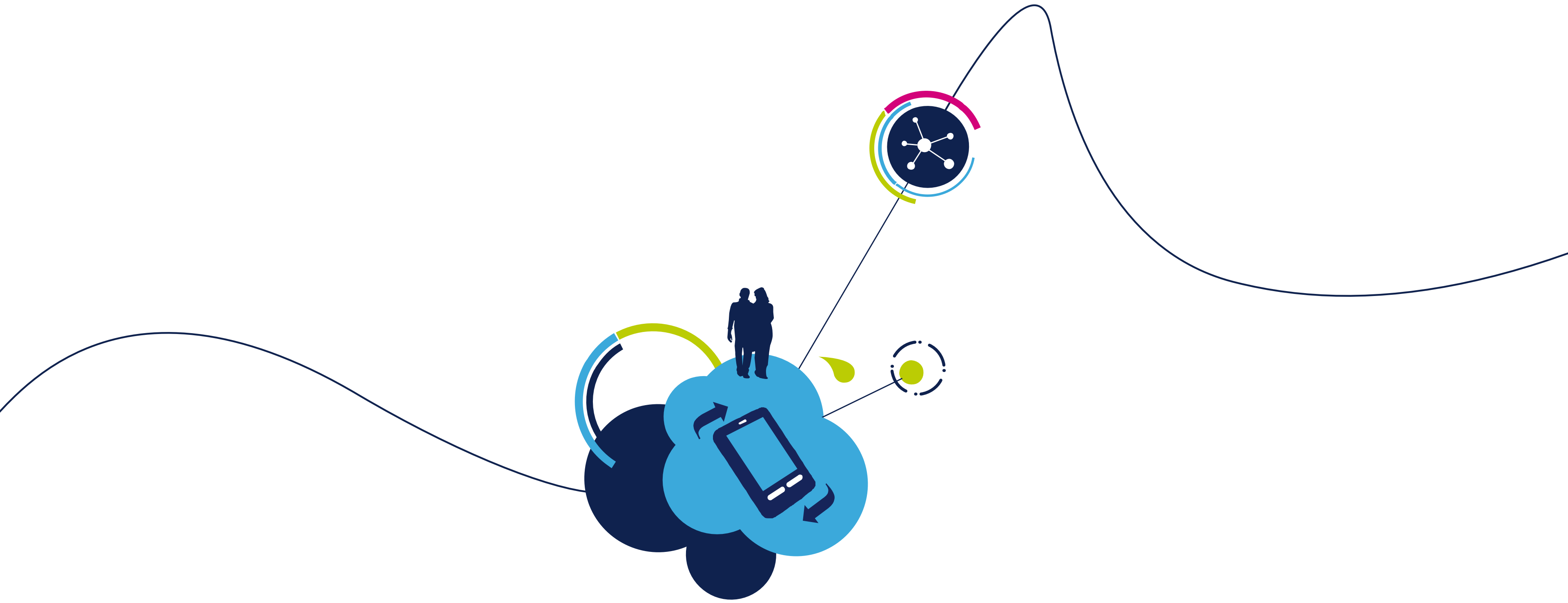


1.1.1 GPIO Lab

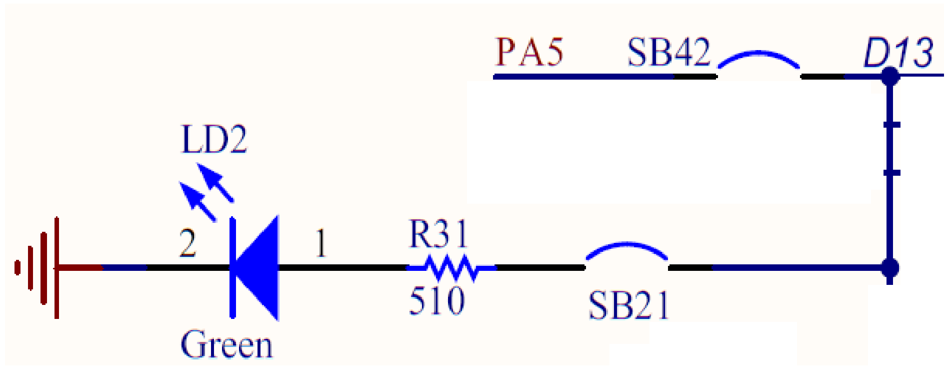
life.augmented

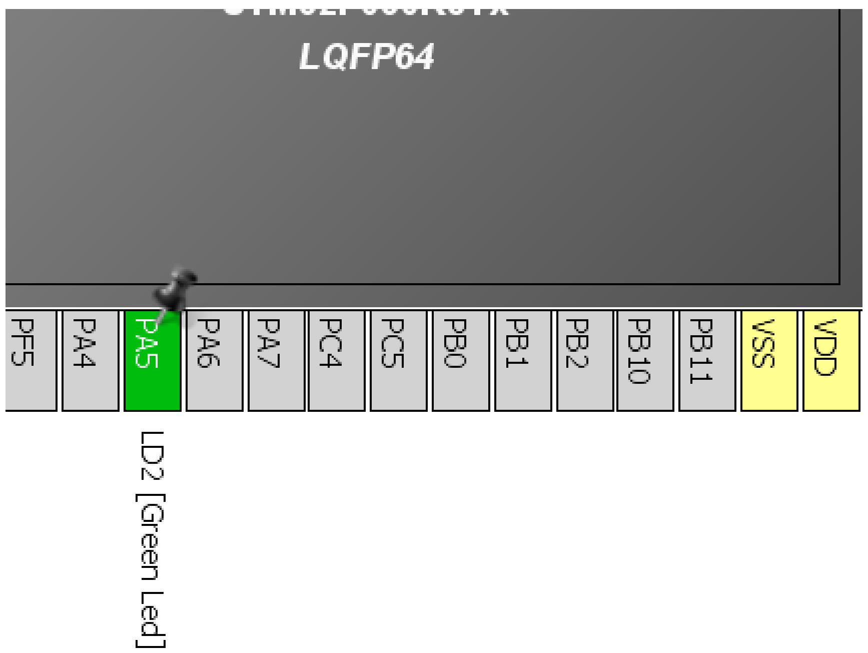
1.1.1 Configure GPIO for LED toggling

- Objective
  - Learn how to setup pin and GPIO port in CubeMX
  - How to Generate Code in CubeMX and use HAL functions
- Goal
  - Configure GPIO pin in CubeMX and Generate Code
  - Add in to project HAL_Delay function and HAL_GPIO_Toggle function
  - Verify the correct functionality on toggling LED

1.1.1 Configure GPIO for LED toggling

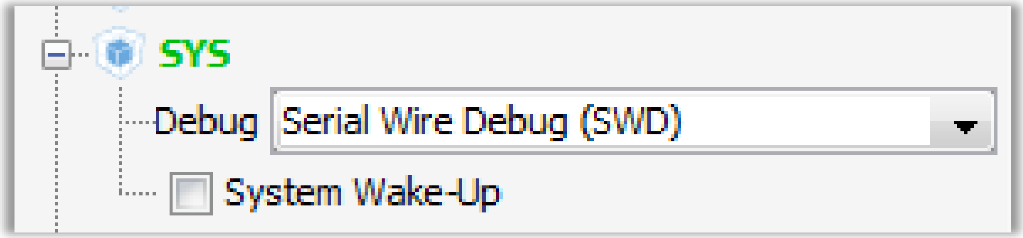
- Create project in CubeMX
  - Menu > File > New Project
  - Select STM32F0 > STM32F030 > LQFP64 > STM32F030R8
- Configure LED pin as GPIO_Output



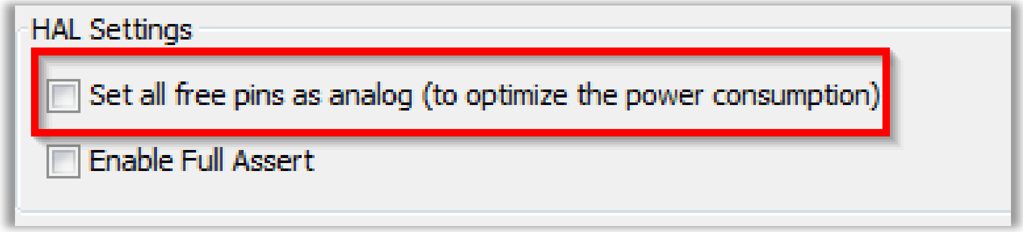


1.1.1 Configure GPIO for LED toggling

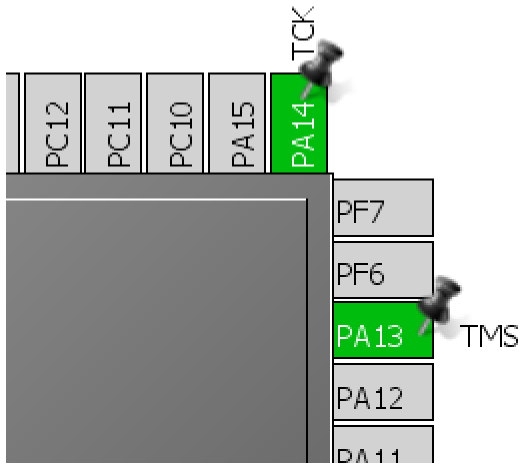
- For debug purpose is recommended to select debug pins SWD or JTAG
  - Select can be done in TAB>Pinout>SYS
  - On discovery is available only SWD option
  - If **SWD/JTAG is not selected** and the **Set all free pins as analog** (MENU>Project>Settings>TAB>Code Generator) is selected, **debug is not possible**



SYS
Debug: Serial Wire Debug (SWD)
☐ System Wake-Up

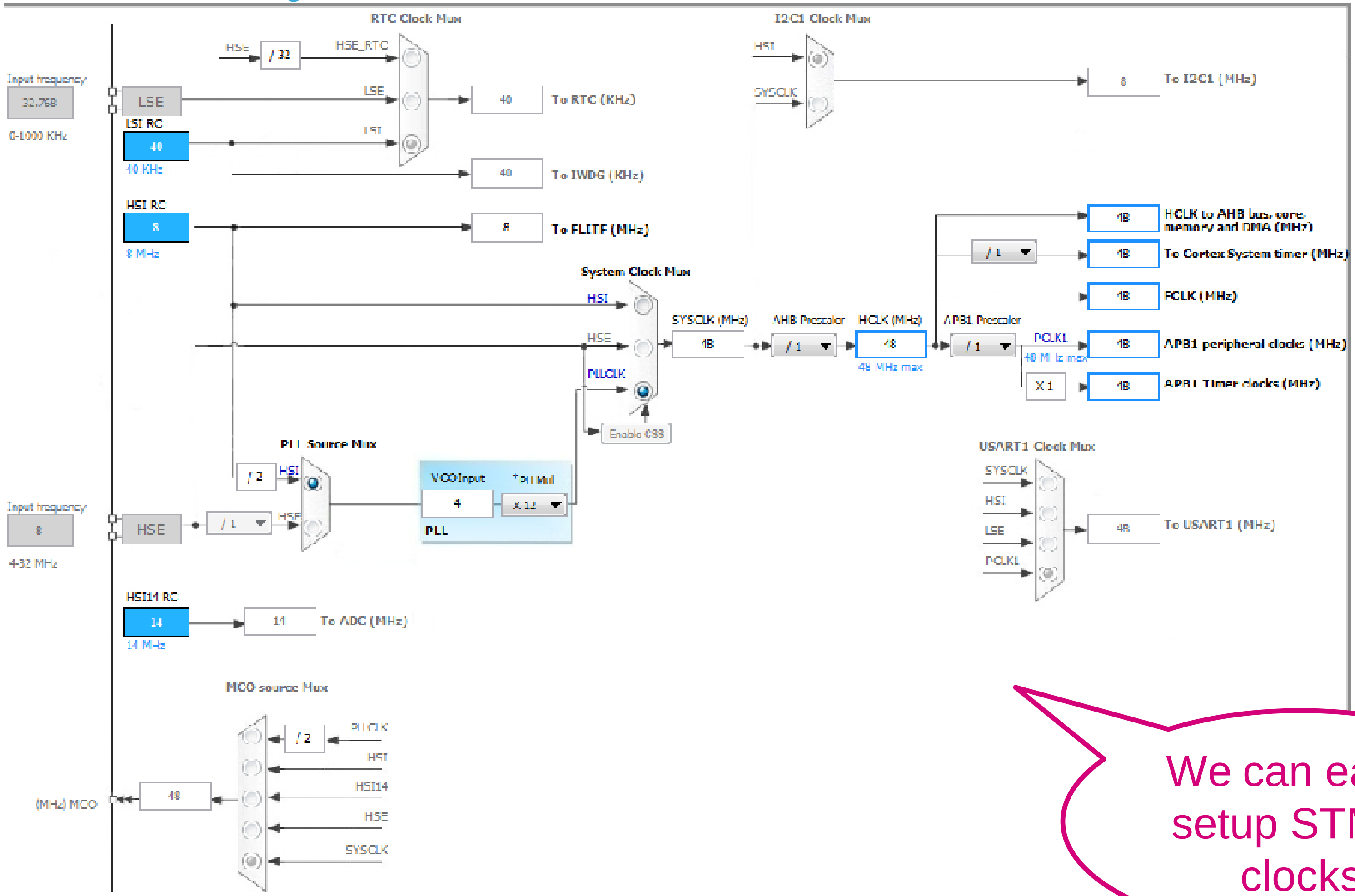


HAL Settings
☐ Set all free pins as analog (to optimize the power consumption)
☐ Enable Full Assert



1.1.1 Configure GPIO for LED toggling

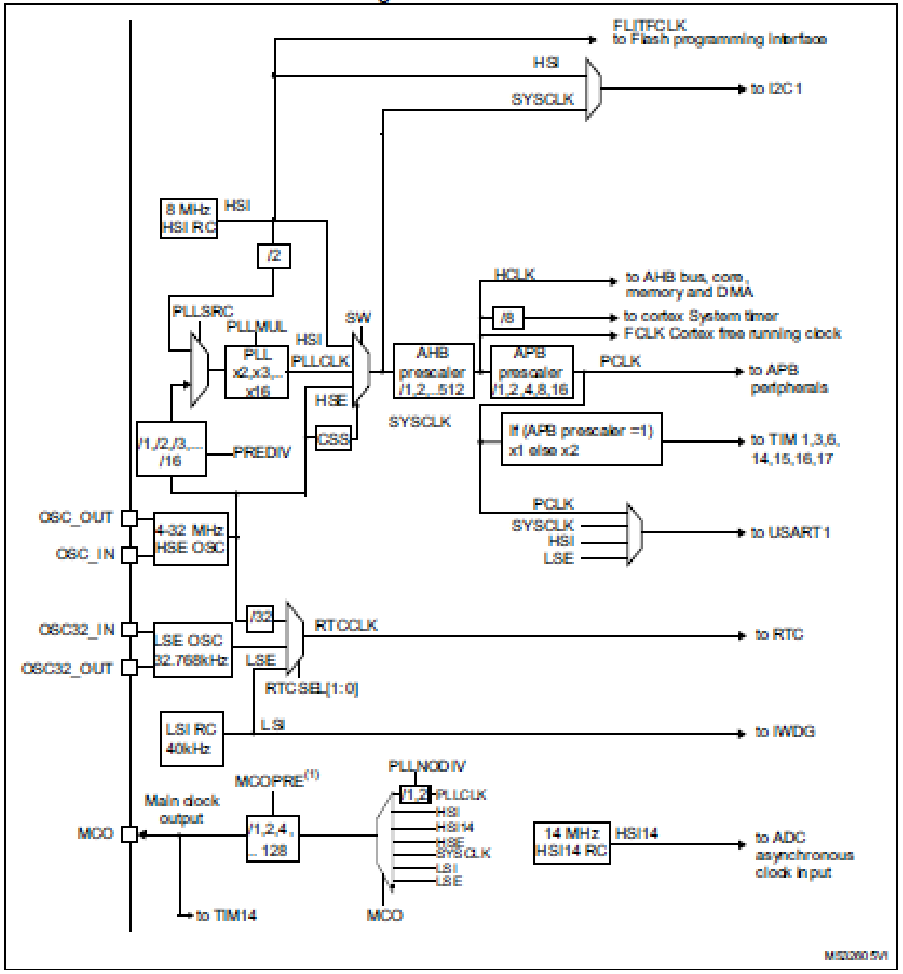
- Clock Configuration
  - TAB>Clock Configuration



1.1.1 Configure GPIO for LED toggling

- The Clock configuration tree is interactive version of tree from RM

Figure 9. Clock tree

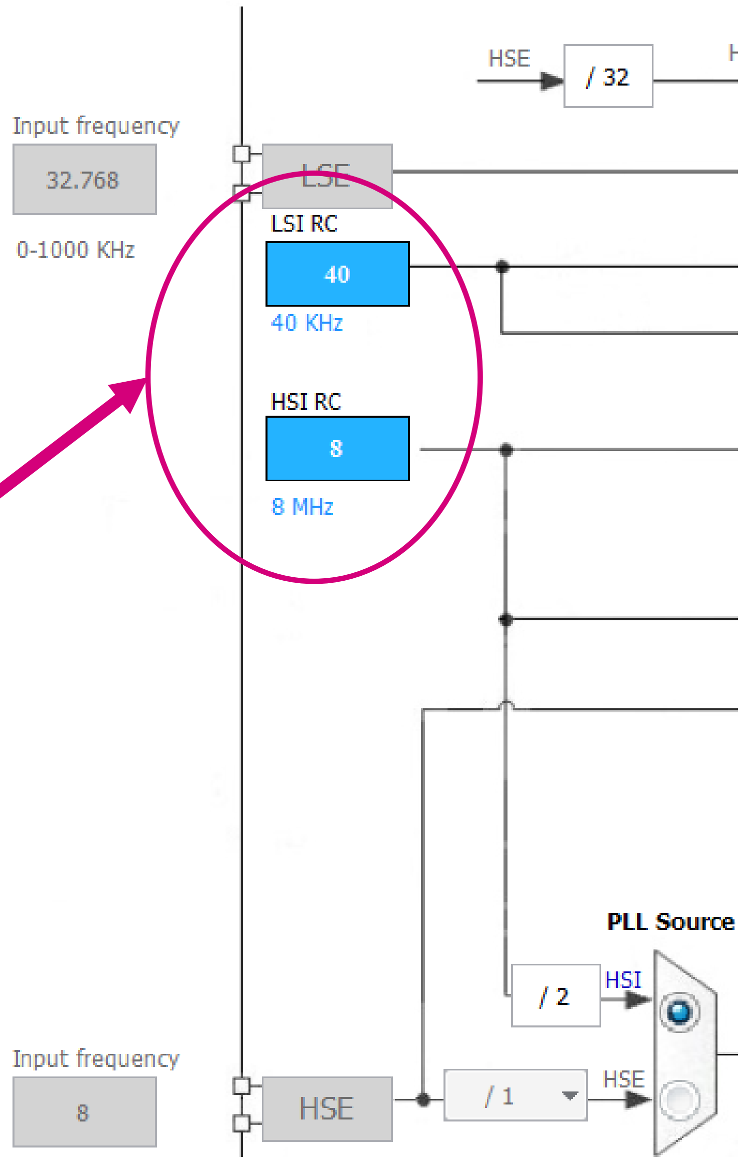


1.1.1 Configure GPIO for LED toggling

Clock Configuration overview 1

- Clock sources
  - Internal oscillators

Internal oscillators

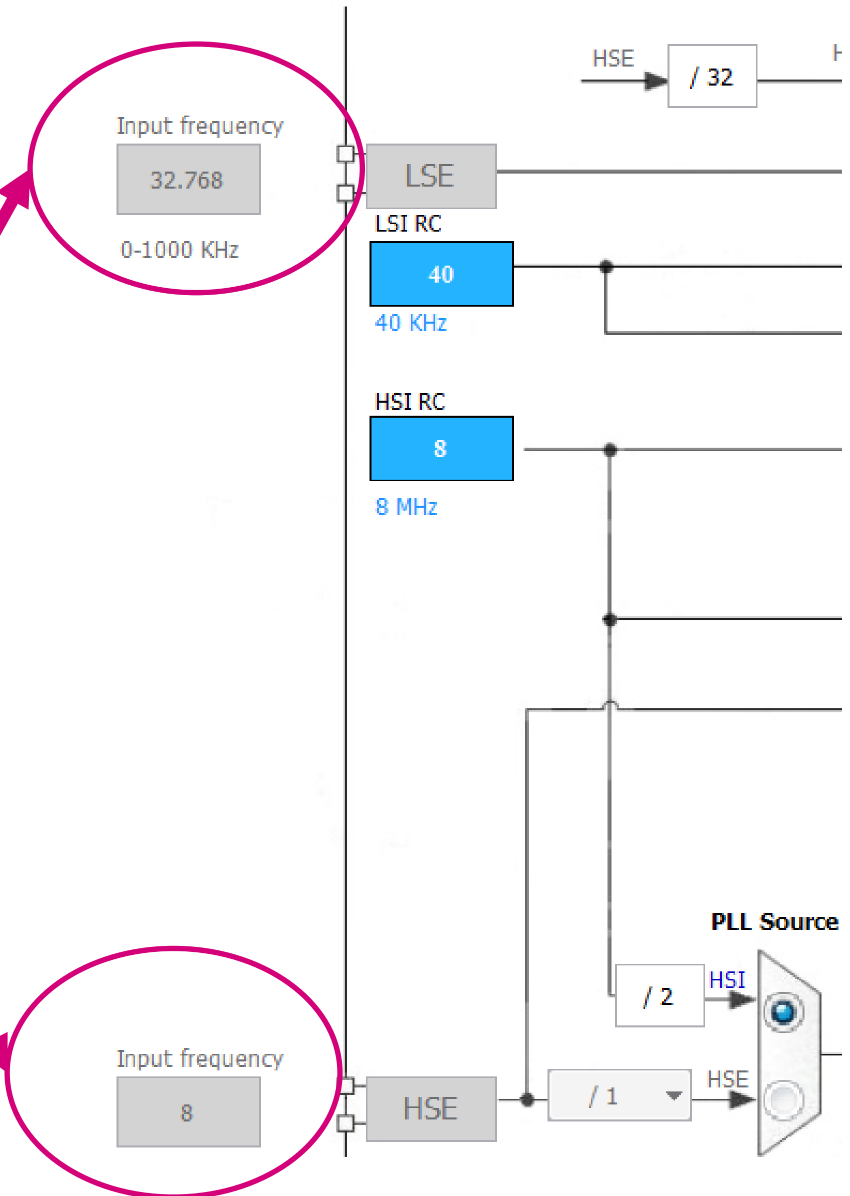


1.1.1 Configure GPIO for LED toggling

Clock Configuration overview 3

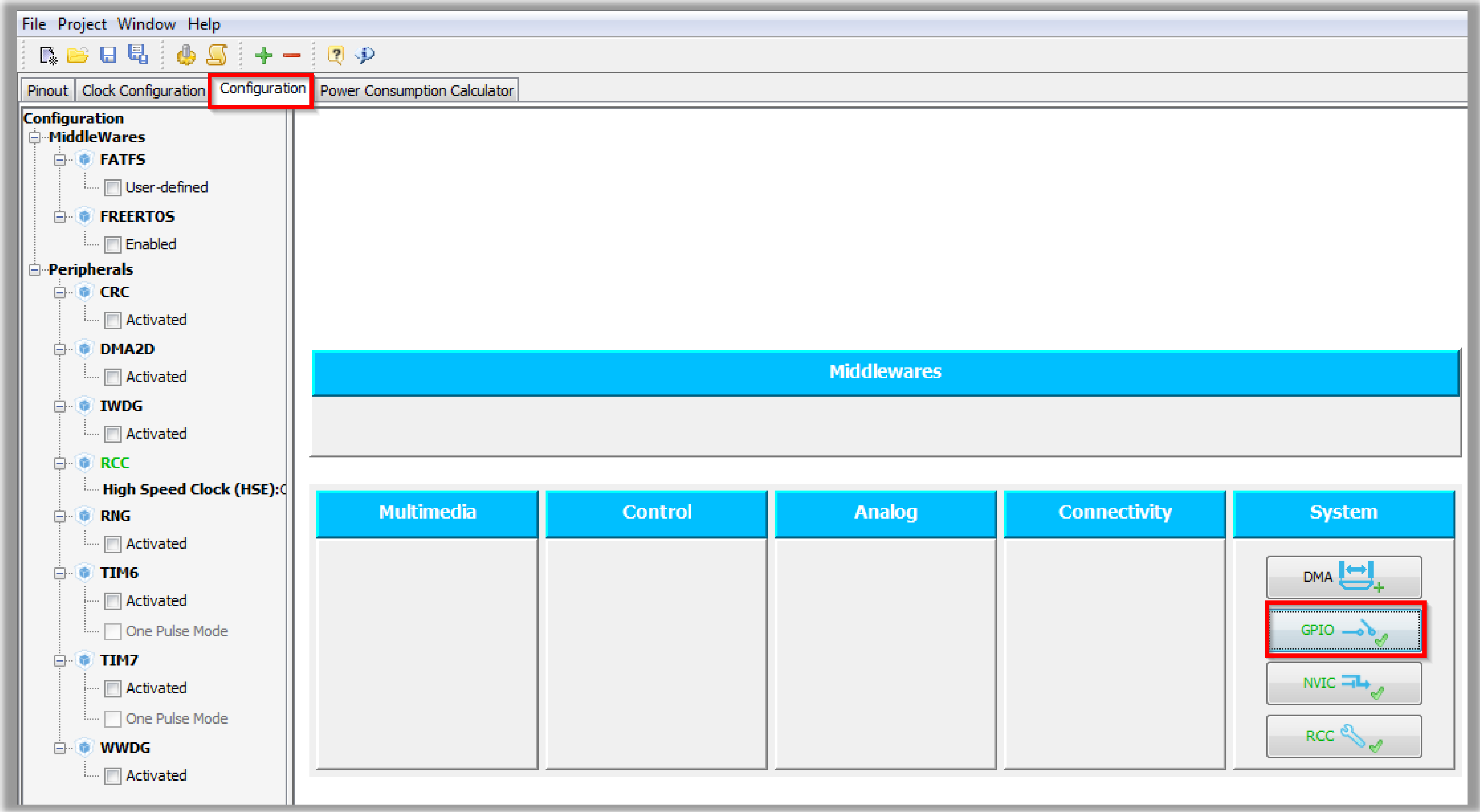
- Clock sources
  - Internal oscillators
  - External clock sources

External clock sources



1.1.1 Configure GPIO for LED toggling

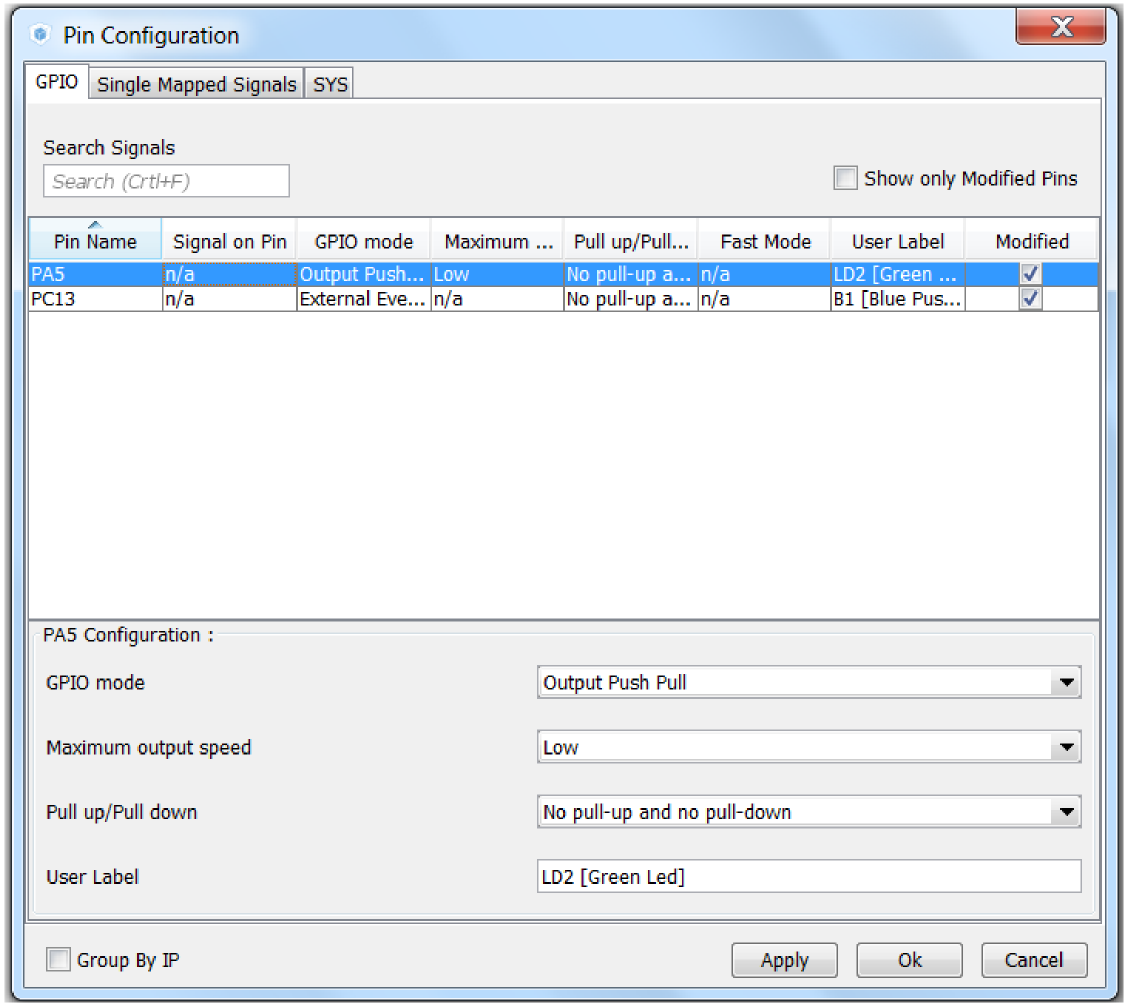
- GPIO Configuration
  - TAB>Configuration>System>GPIO



Configuration
- MiddleWares
  - FATFS
    - User-defined
  - FREERTOS
    - Enabled
- Peripherals
  - CRC
    - Activated
  - DMA2D
    - Activated
  - IWDG
    - Activated
  - RCC
    - High Speed Clock (HSE):
  - RNG
    - Activated
  - TIM6
    - Activated
    - One Pulse Mode
  - TIM7
    - Activated
    - One Pulse Mode
  - WWDG
    - Activated

1.1.1 Configure GPIO for LED toggling

- GPIO(Pin) Configuration
  - Select Push Pull mode
  - No pull-up and pull-down
  - Output speed to HIGH
    Is important for faster peripheries like SPI, USART
  - Button OK



Pin Name	Signal on Pin	GPIO mode	Maximum ...	Pull up/Pull...	Fast Mode	User Label	Modified
PA5	n/a	Output Push...	Low	No pull-up a...	n/a	LD2 [Green ...	✓
PC13	n/a	External Eve...	n/a	No pull-up a...	n/a	B1 [Blue Pus...	✓

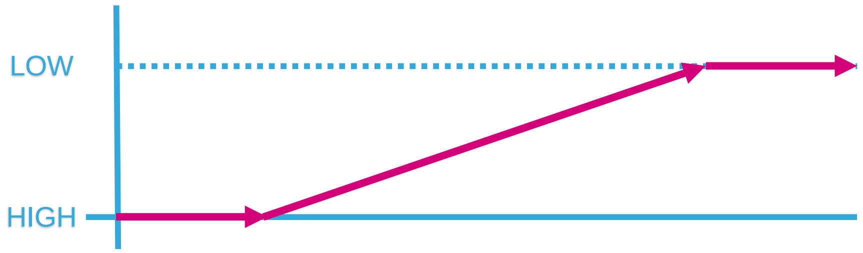
PA5 Configuration :

GPIO mode	Output Push Pull
Maximum output speed	Low
Pull up/Pull down	No pull-up and no pull-down
User Label	LD2 [Green Led]

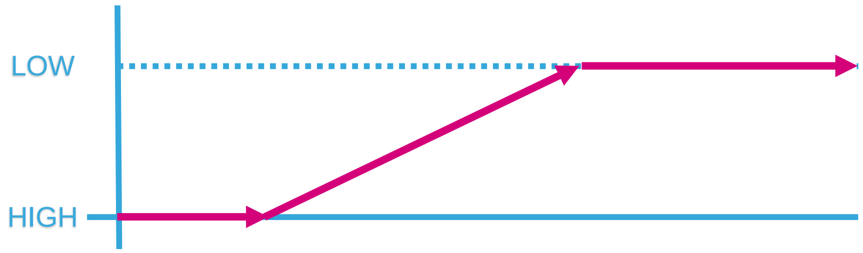
1.1.1 Configure GPIO for LED toggling

- GPIO(Pin) output speed configuration
  - Change the rising and falling edge when pin change state from high to low or low to high
  - **Higher** GPIO speed increase **EMI noise** from STM32 and increase STM32 **consumption**
  - It is good to adapt GPIO speed with periphery speed. Ex.: Toggling GPIO on 1Hz is LOW optimal settings, but SPI on 45MHz the HIGH must be set

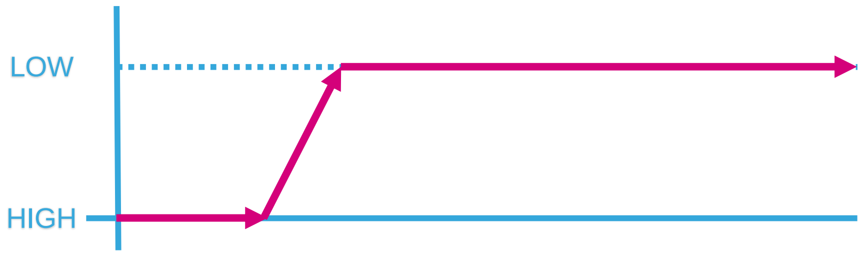
GPIO output LOW speed

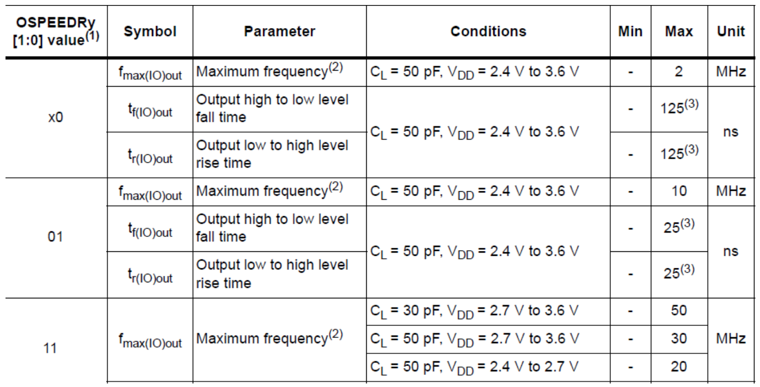


GPIO output MEDIUM speed



GPIO output HIGH speed

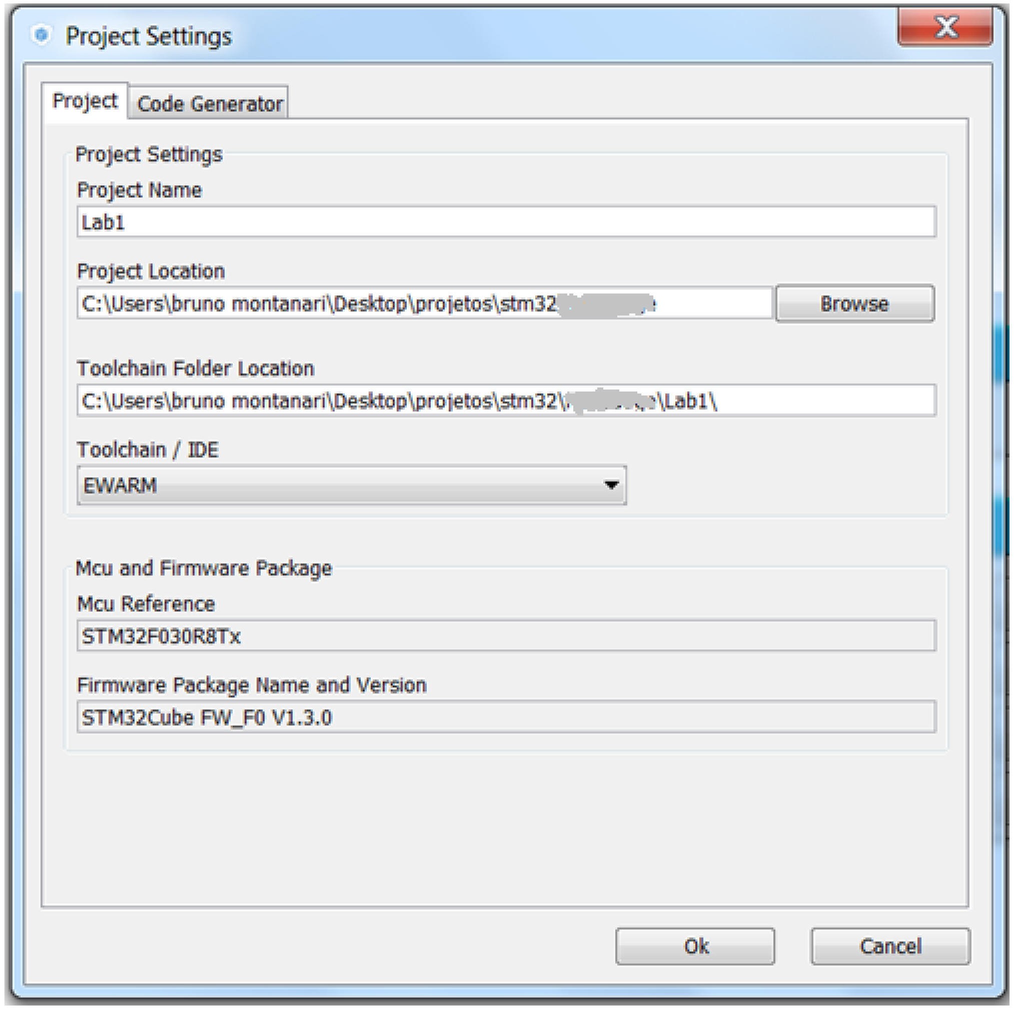




OSPEEDRy [1:0] value[1]	Symbol	Parameter	Conditions	Min	Max	Unit
x0	$f_{max(IO)out}$	Maximum frequency[2]	C_L = 50 pF, V_{DD} = 2.4 V to 3.6 V	-	2	MHz
	$t_{f(IO)out}$	Output high to low level fall time	C_L = 50 pF, V_{DD} = 2.4 V to 3.6 V	-	125[3]	ns
	$t_{r(IO)out}$	Output low to high level rise time		-	125[3]	
01	$f_{max(IO)out}$	Maximum frequency[2]	C_L = 50 pF, V_{DD} = 2.4 V to 3.6 V	-	10	MHz
	$t_{f(IO)out}$	Output high to low level fall time	C_L = 50 pF, V_{DD} = 2.4 V to 3.6 V	-	25[3]	ns
	$t_{r(IO)out}$	Output low to high level rise time		-	25[3]	
11	$f_{max(IO)out}$	Maximum frequency[2]	C_L = 30 pF, V_{DD} = 2.7 V to 3.6 V	-	50	MHz
			C_L = 50 pF, V_{DD} = 2.7 V to 3.6 V	-	30	
			C_L = 50 pF, V_{DD} = 2.4 V to 2.7 V	-	20	

1.1.1 Configure GPIO for LED toggling

- Now we set the project details for generation
  - Menu > Project > Project Settings
  - Set the project name
  - Project location
  - Type of toolchain
- Now we can Generate Code
  - Menu > Project > Generate Code



Project Settings

Project Name: Lab1

Project Location: C:\Users\bruno montanari\Desktop\projetos\stm32[redacted]

Toolchain Folder Location: C:\Users\bruno montanari\Desktop\projetos\stm32\[redacted]\Lab1\

Toolchain / IDE: EWARM

Mcu and Firmware Package

Mcu Reference: STM32F030R8Tx

Firmware Package Name and Version: STM32Cube FW_F0 V1.3.0

1.1.1 Configure GPIO for LED toggling

- Now we open the project in our IDE
  - The functions we want to put into main.c
  - Between */* USER CODE BEGIN 3 */* and */* USER CODE END 3 */* tags
  - Into infinite loop *while(1){   }*
- For toggling we need to use this functions
  - *HAL_HAL_Delay* which create specific delay
  - *HAL_GPIO_WritePin* or *HAL_GPIO_TogglePin*

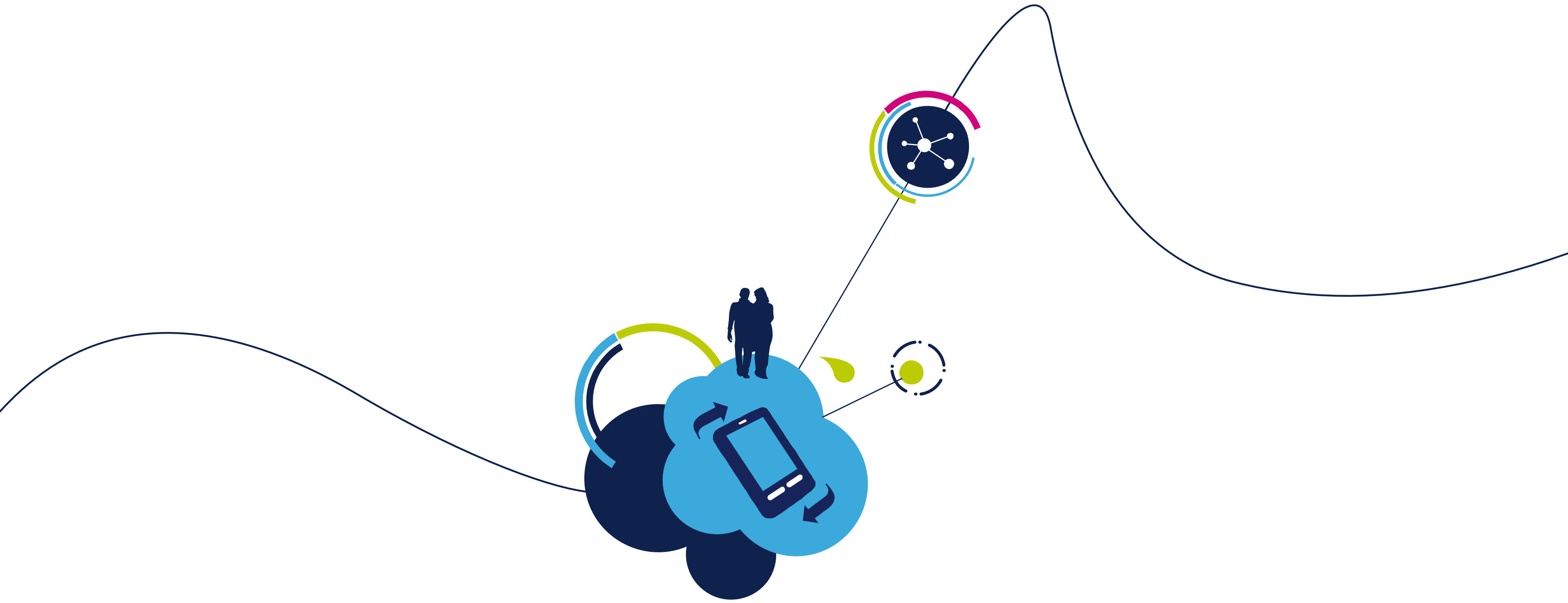
1.1.1 Configure GPIO for LED toggling

- Now we open the project in our IDE
  - The functions we want to put into main.c
  - Between */* USER CODE BEGIN 3 */* and */* USER CODE END 3 */* tags
  - Into infinite loop *while(1){ }*
- For toggling we need to use this functions
  - *HAL_HAL_Delay* which create specific delay
  - *HAL_GPIO_WritePin* or *HAL_GPIO_TogglePin*

```
/* USER CODE BEGIN 3 */
 /* Infinite loop */
 while (1)
 {
   HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
   HAL_Delay(500);

   HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
   HAL_Delay(500);
 }
 /* USER CODE END 3 */
```

life.augmented



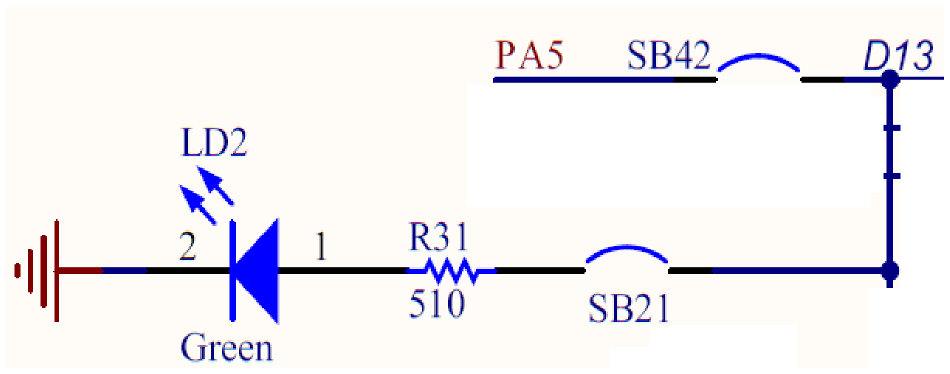
1.1.2 EXTI lab

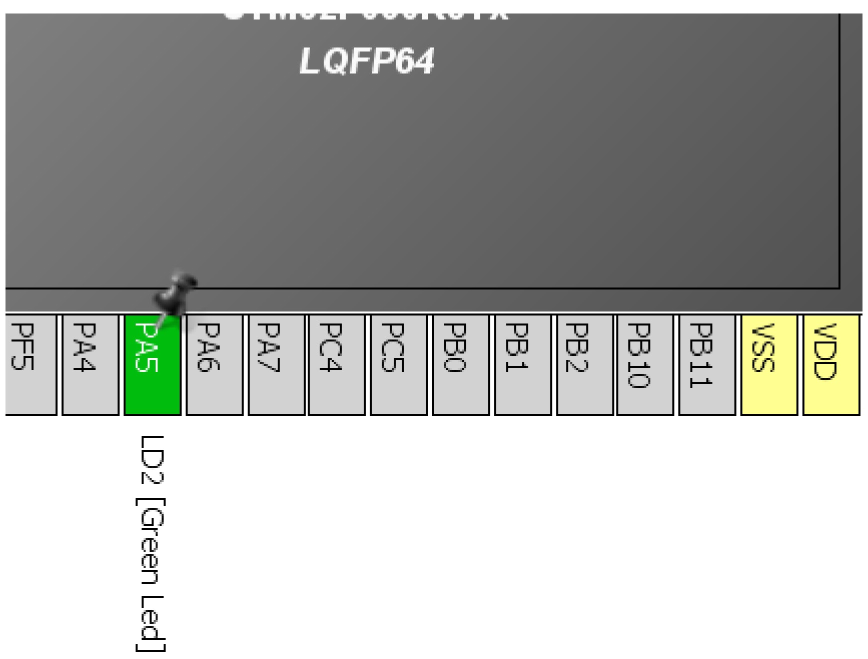
1.1.2 Configure EXTI to turn on LED

- Objective
    - Learn how to setup input pin with EXTI in CubeMX
    - How to Generate Code in CubeMX and use HAL functions
- Goal
    - Configure GPIO and EXTI pin in CubeMX and Generate Code
    - Add into project Callback function and function which turn on led
    - Verify the correct functionality by pressing button which turns on LED

1.1.2 Configure EXTI to turn on LED

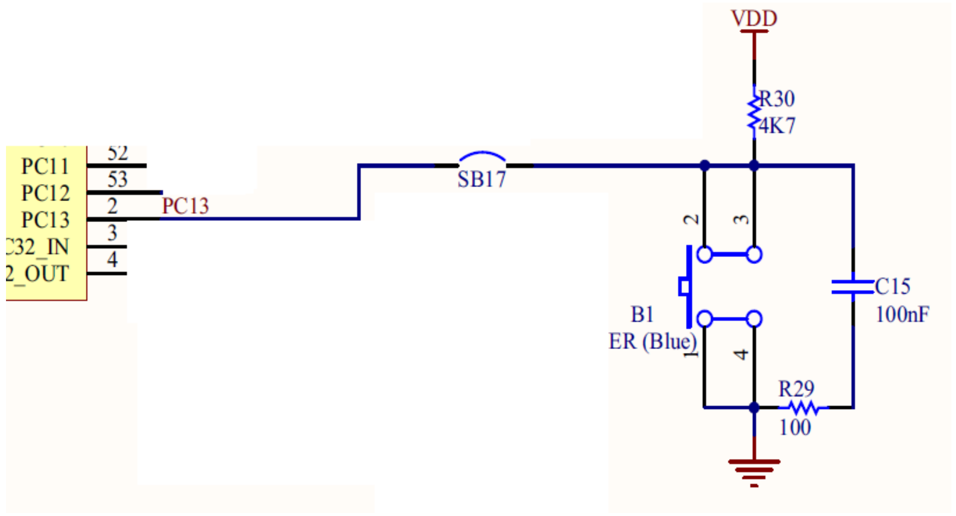
- Create project in CubeMX
  - Menu > File > New Project
  - Select STM32F0 > STM32F030 > LQFP64 > STM32F030R8
- Configure LED pin as GPIO_Output

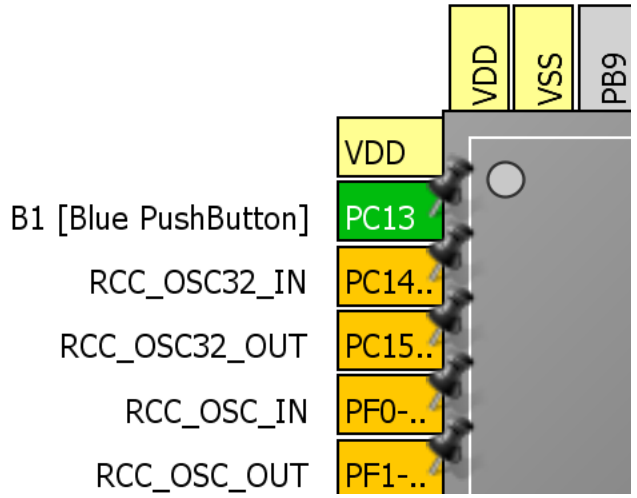




1.1.2 Configure EXTI to turn on LED

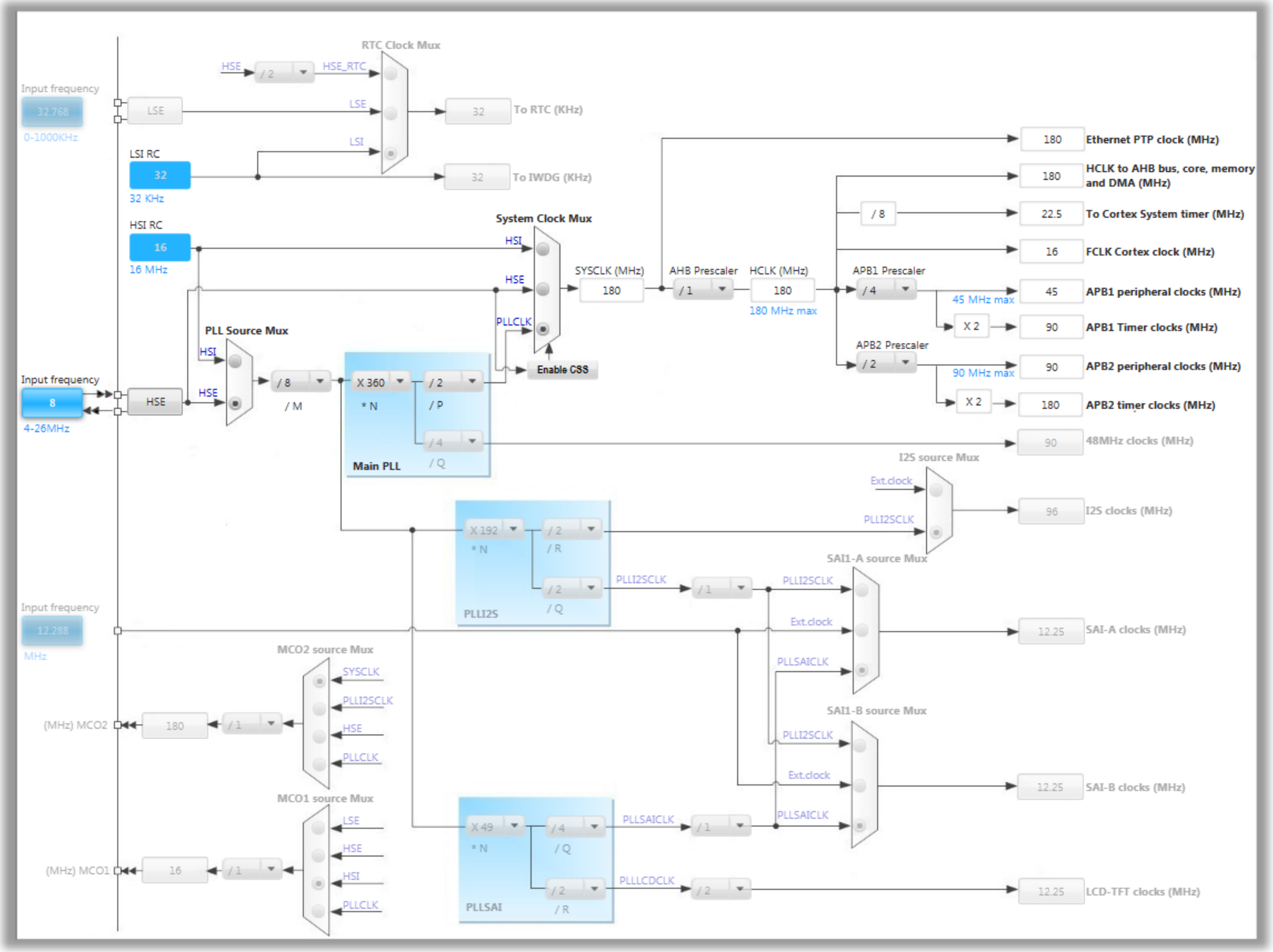
- Configure Button pin as GPIO_EXTIX





1.1.2 Configure EXTI to turn on LED

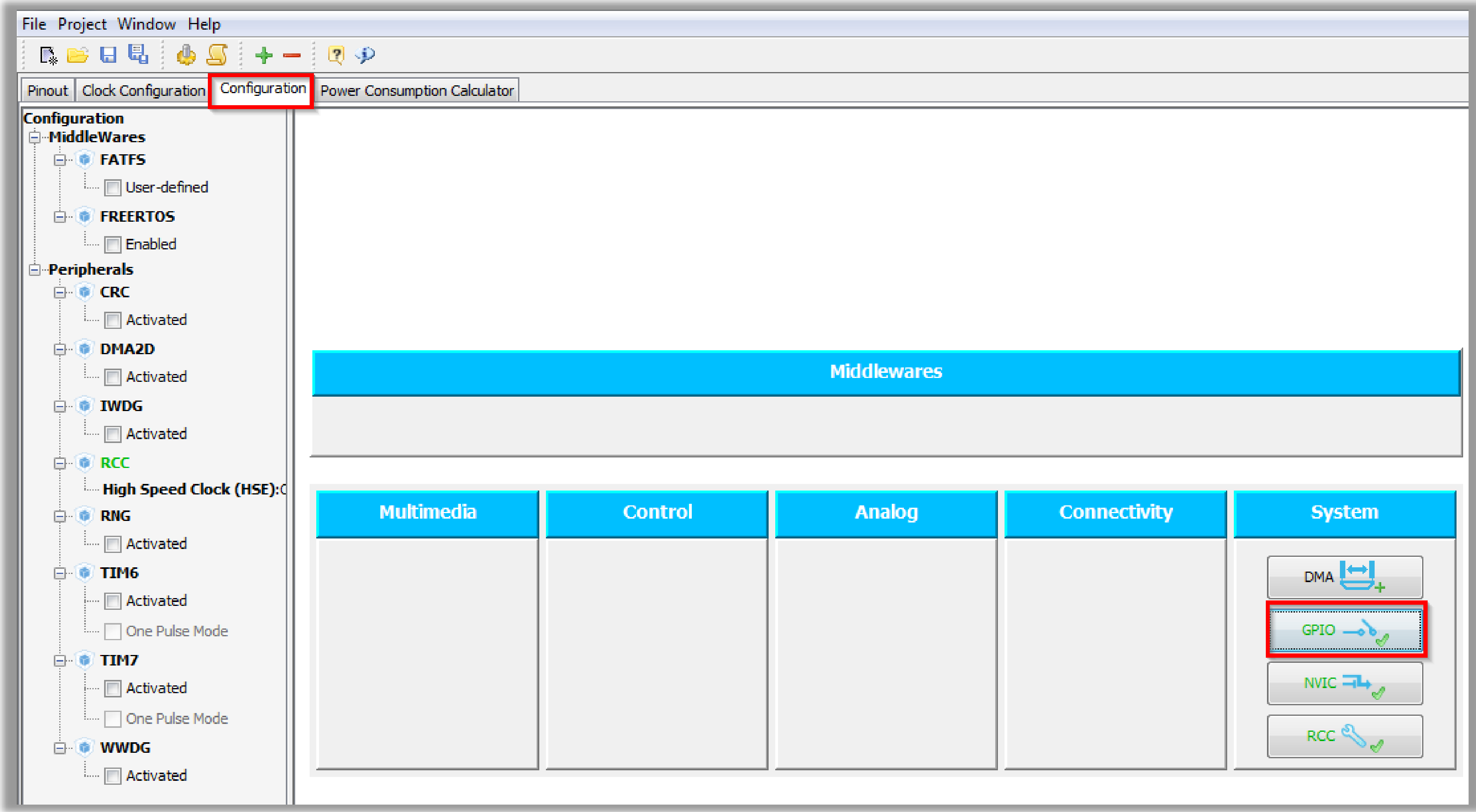
- In order to run on maximum frequency, setup clock system
- Details in lab 0



life.augmented

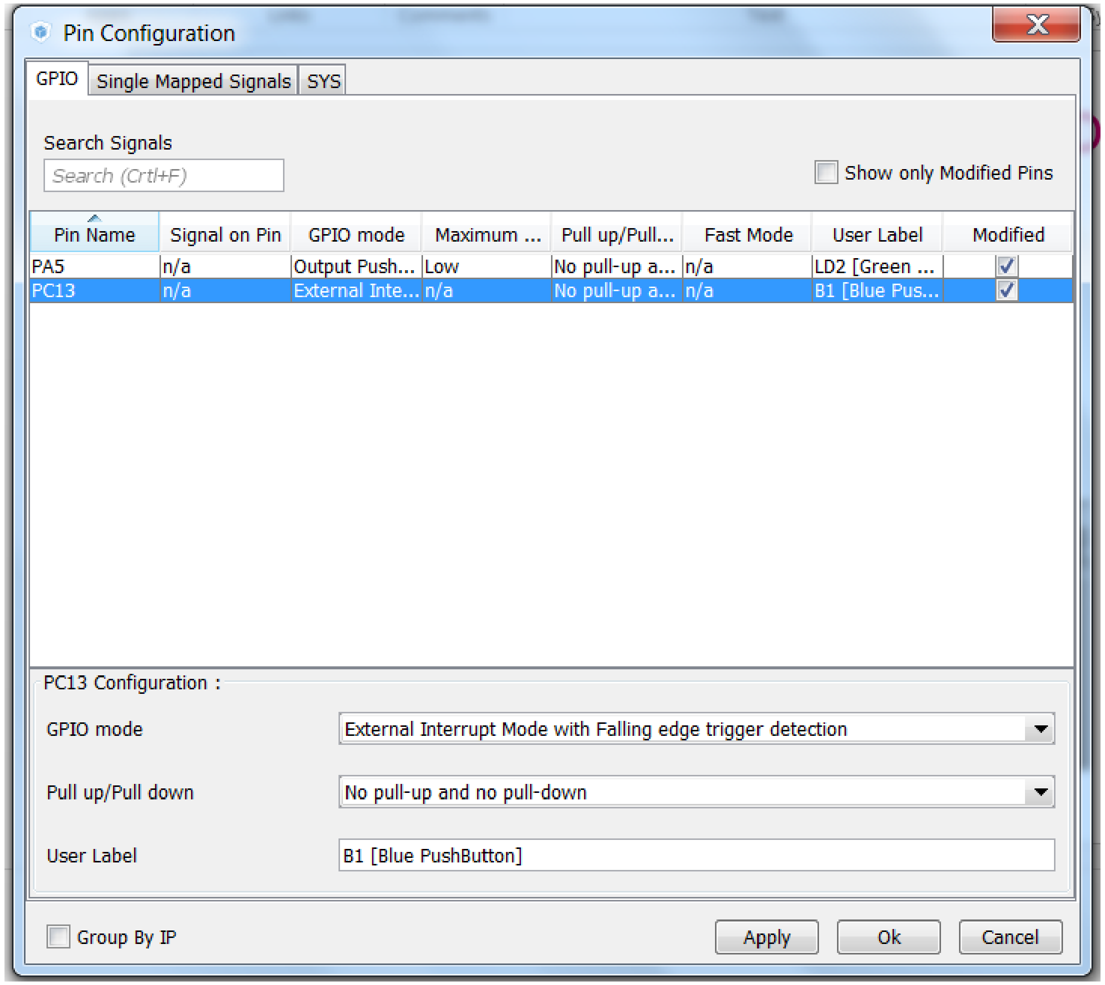
1.1.2 Configure EXTI to turn on LED

- GPIO Configuration
  - TAB>Configuration>System>GPIO



1.1.2 Configure EXTI to turn on LED

- GPIO(Pin) Configuration
  - Select External Interrupt Mode with Faling edge trigger detection
  - No pull-up or pull-down
  - Button OK



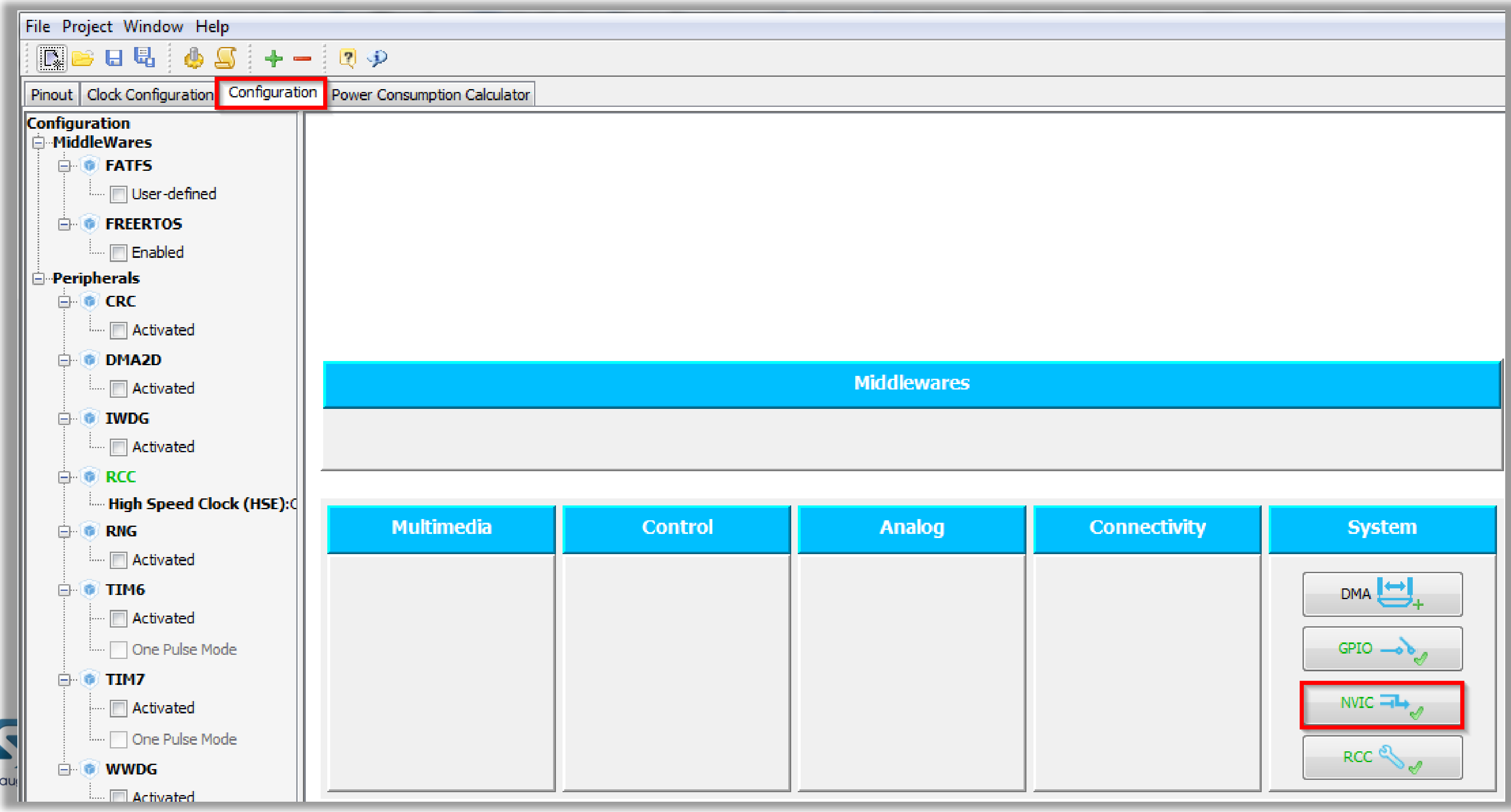
Pin Name	Signal on Pin	GPIO mode	Maximum ...	Pull up/Pull...	Fast Mode	User Label	Modified
PA5	n/a	Output Push...	Low	No pull-up a...	n/a	LD2 [Green ...	✓
PC13	n/a	External Inte...	n/a	No pull-up a...	n/a	B1 [Blue Pus...	✓

PC13 Configuration :

GPIO mode	External Interrupt Mode with Falling edge trigger detection
Pull up/Pull down	No pull-up and no pull-down
User Label	B1 [Blue PushButton]

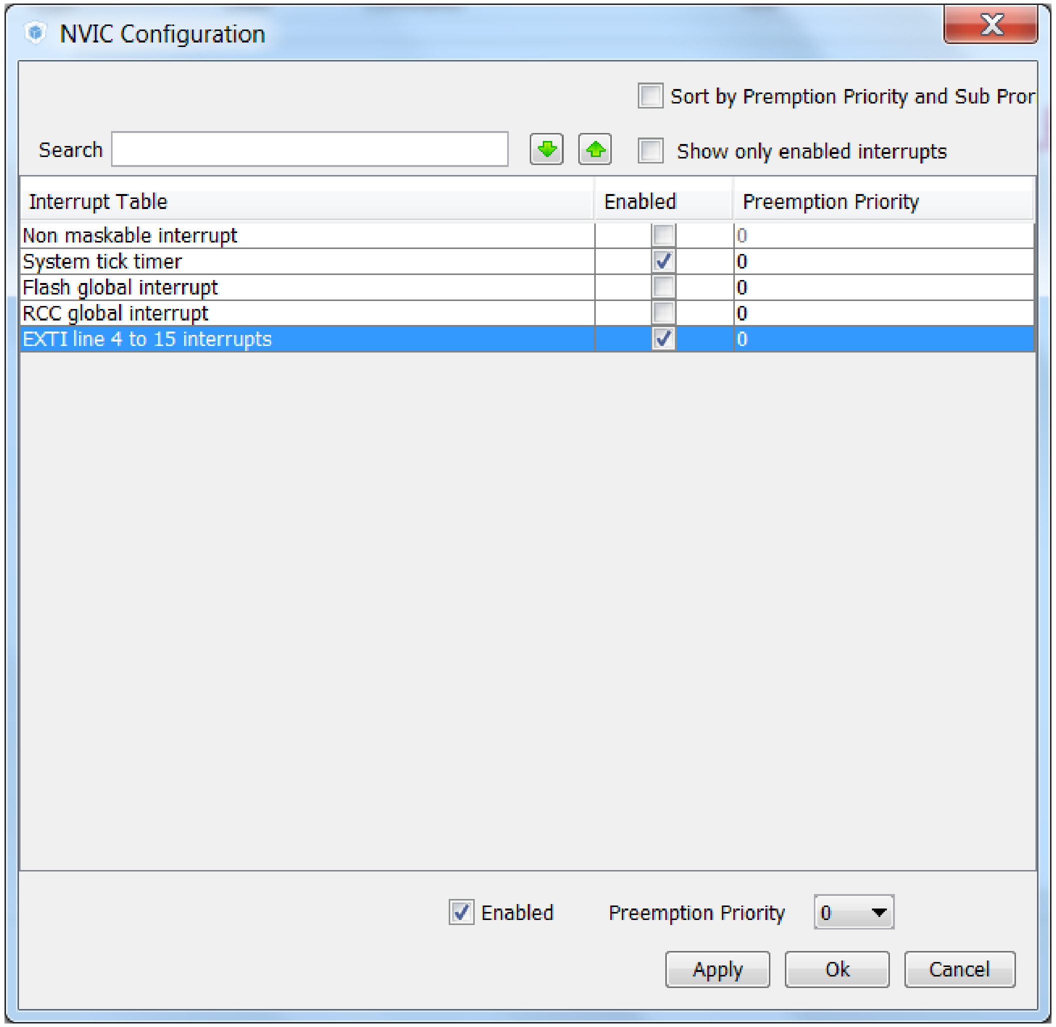
1.1.2 Configure EXTI to turn on LED

- NVIC Configuration
  - We need to enable interrupts for EXTI
  - TAB>Configuration>System>NVIC



1.1.2 Configure EXTI to turn on LED

- NVIC Configuration
  - Enable interrupt for EXTI Line4_15
  - Button OK



NVIC Configuration

☐ Sort by Premption Priority and Sub Prior

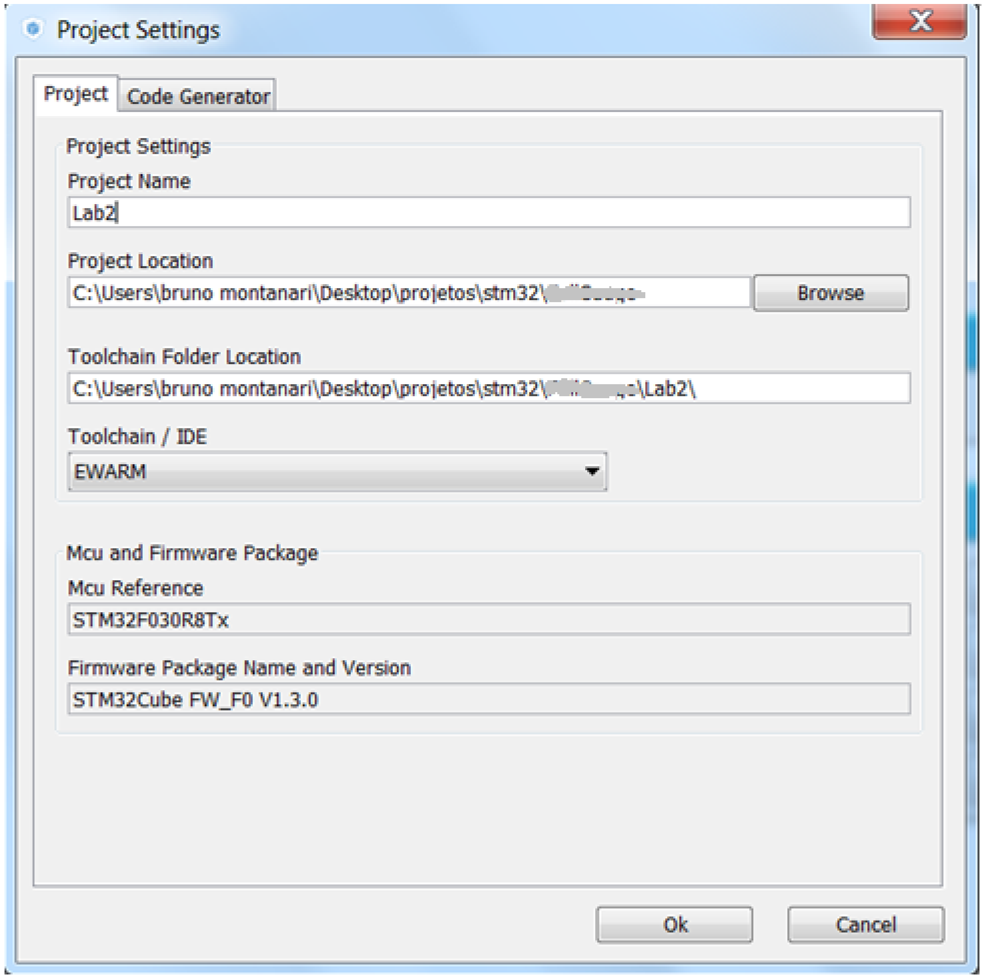
Search ☐ Show only enabled interrupts

Interrupt Table	Enabled	Preemption Priority
Non maskable interrupt	☐	0
System tick timer	✓	0
Flash global interrupt	☐	0
RCC global interrupt	☐	0
EXTI line 4 to 15 interrupts	✓	0

✓ Enabled Preemption Priority 0

1.1.2 Configure EXTI to turn on LED

- Now we set the project details for generation
  - Menu > Project > Project Settings
  - Set the project name
  - Project location
  - Type of toolchain
- Now we can Generate Code
  - Menu > Project > Generate Code



Project Settings

Project Name: Lab2

Project Location: C:\Users\bruno montanari\Desktop\projetos\stm32\[redacted]

Toolchain Folder Location: C:\Users\bruno montanari\Desktop\projetos\stm32\[redacted]\Lab2\

Toolchain / IDE: EWARM

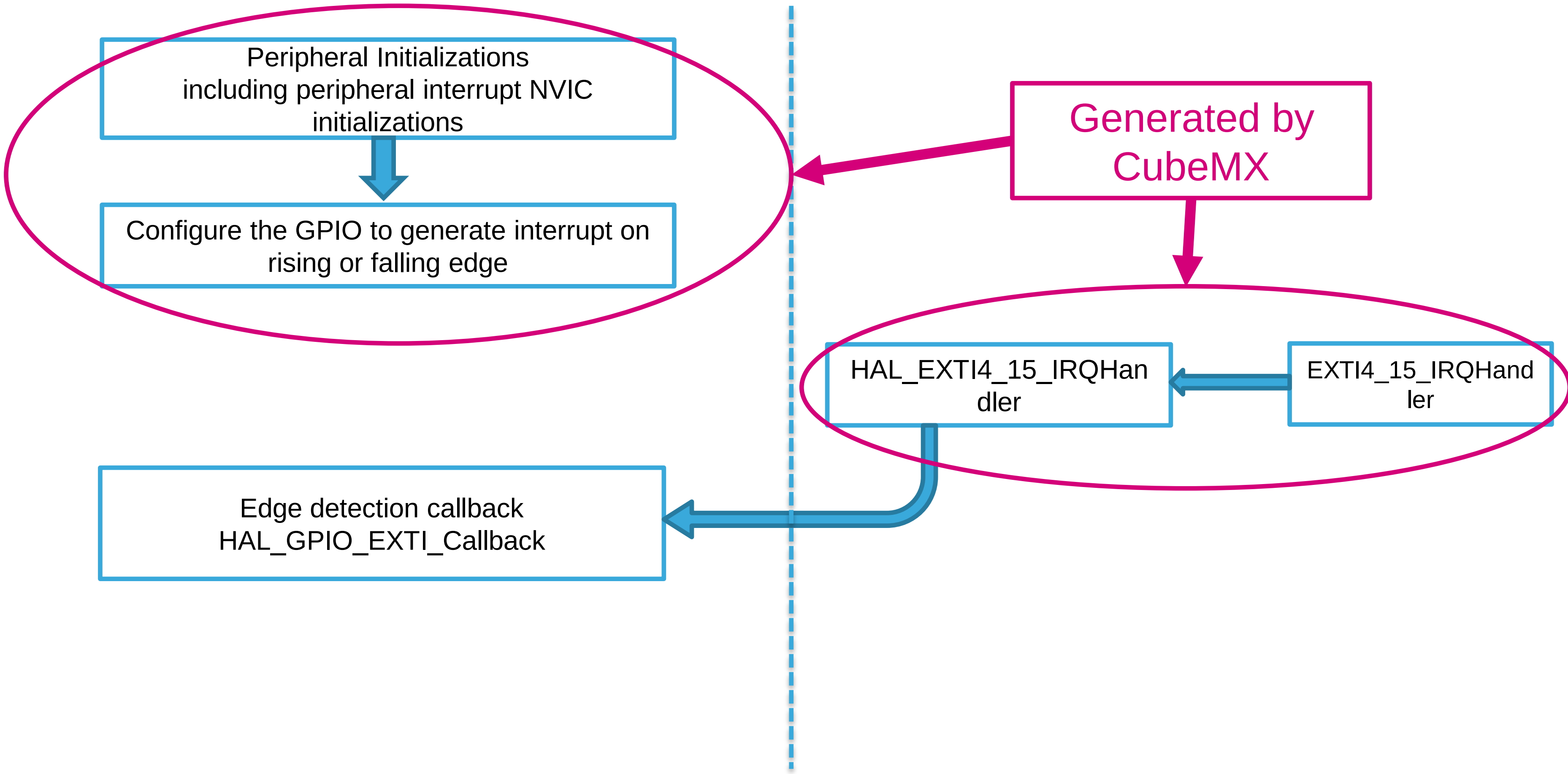
Mcu and Firmware Package

Mcu Reference: STM32F030R8Tx

Firmware Package Name and Version: STM32Cube FW_F0 V1.3.0

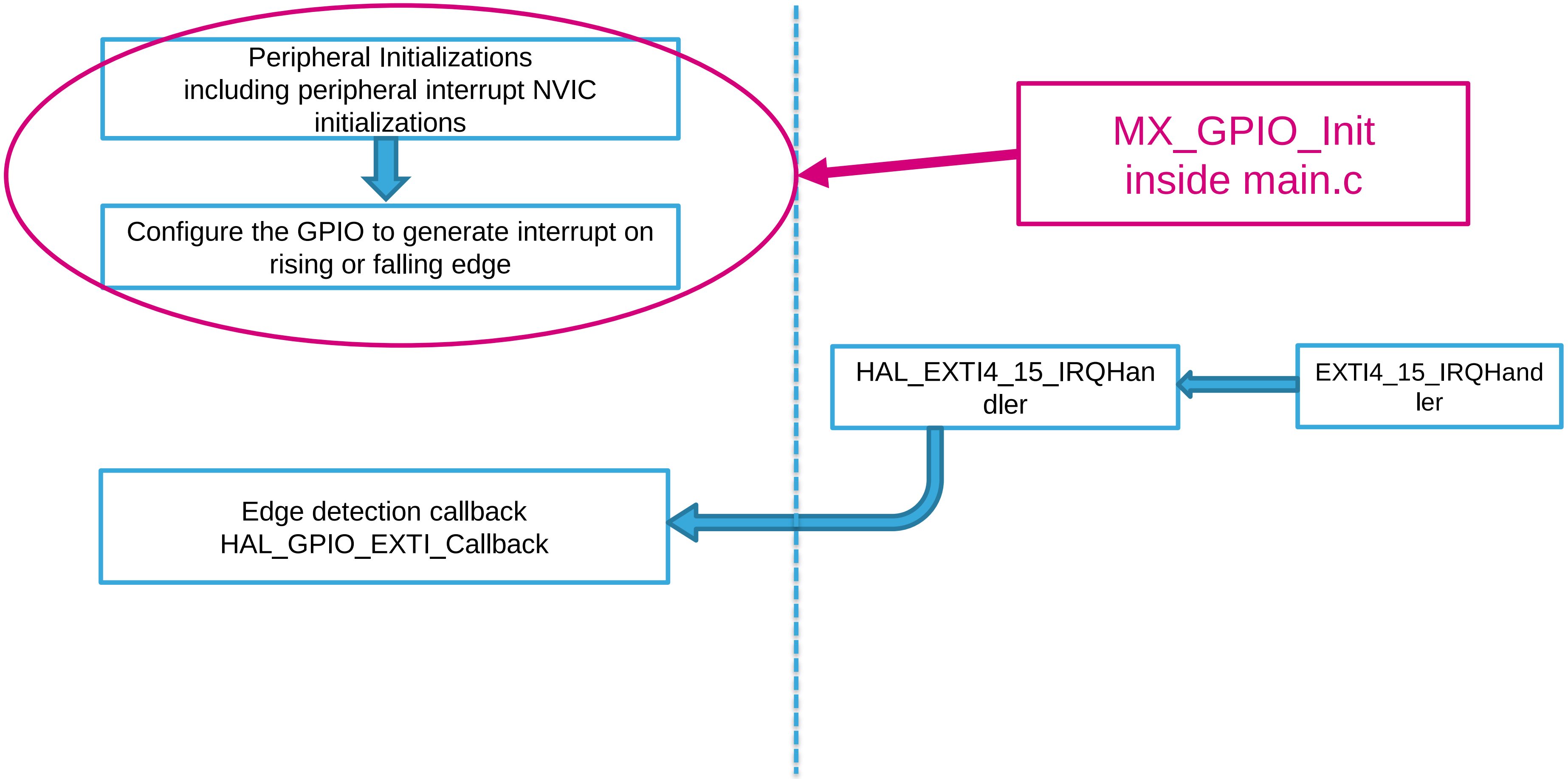
1.1.2 Configure EXTI to turn on LED

HAL Library work flow 1


Peripheral Initializations including peripheral interrupt NVIC initializations
Configure the GPIO to generate interrupt on rising or falling edge
Generated by CubeMX
HAL_EXTI4_15_IRQHandler
EXTI4_15_IRQHandler
Edge detection callback HAL_GPIO_EXTI_Callback

1.1.2 Configure EXTI to turn on LED

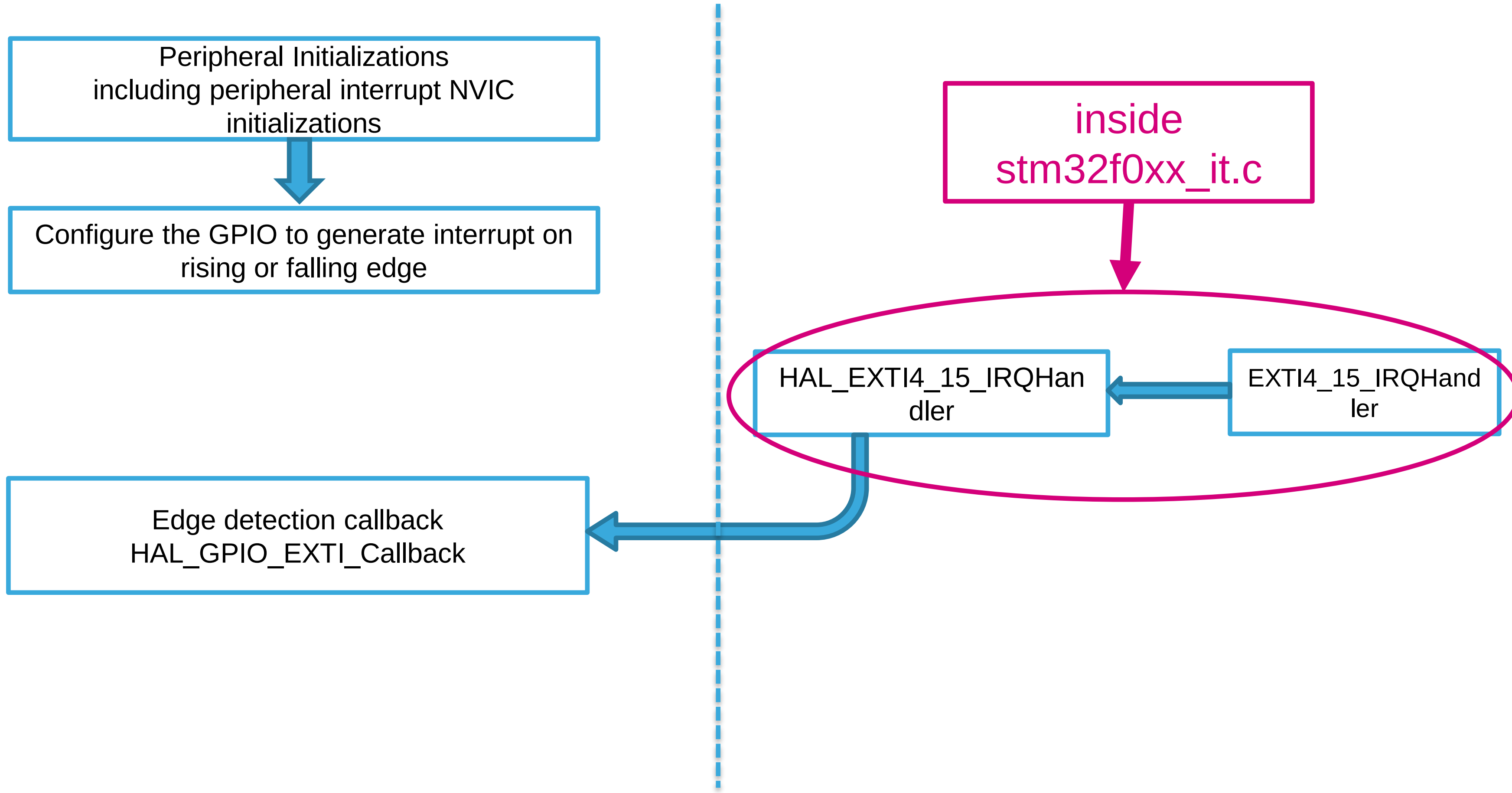
HAL Library work flow 2


Peripheral Initializations including peripheral interrupt NVIC initializations
Configure the GPIO to generate interrupt on rising or falling edge
MX_GPIO_Init inside main.c
HAL_EXTI4_15_IRQHandler
EXTI4_15_IRQHandler
Edge detection callback HAL_GPIO_EXTI_Callback

life.augmented

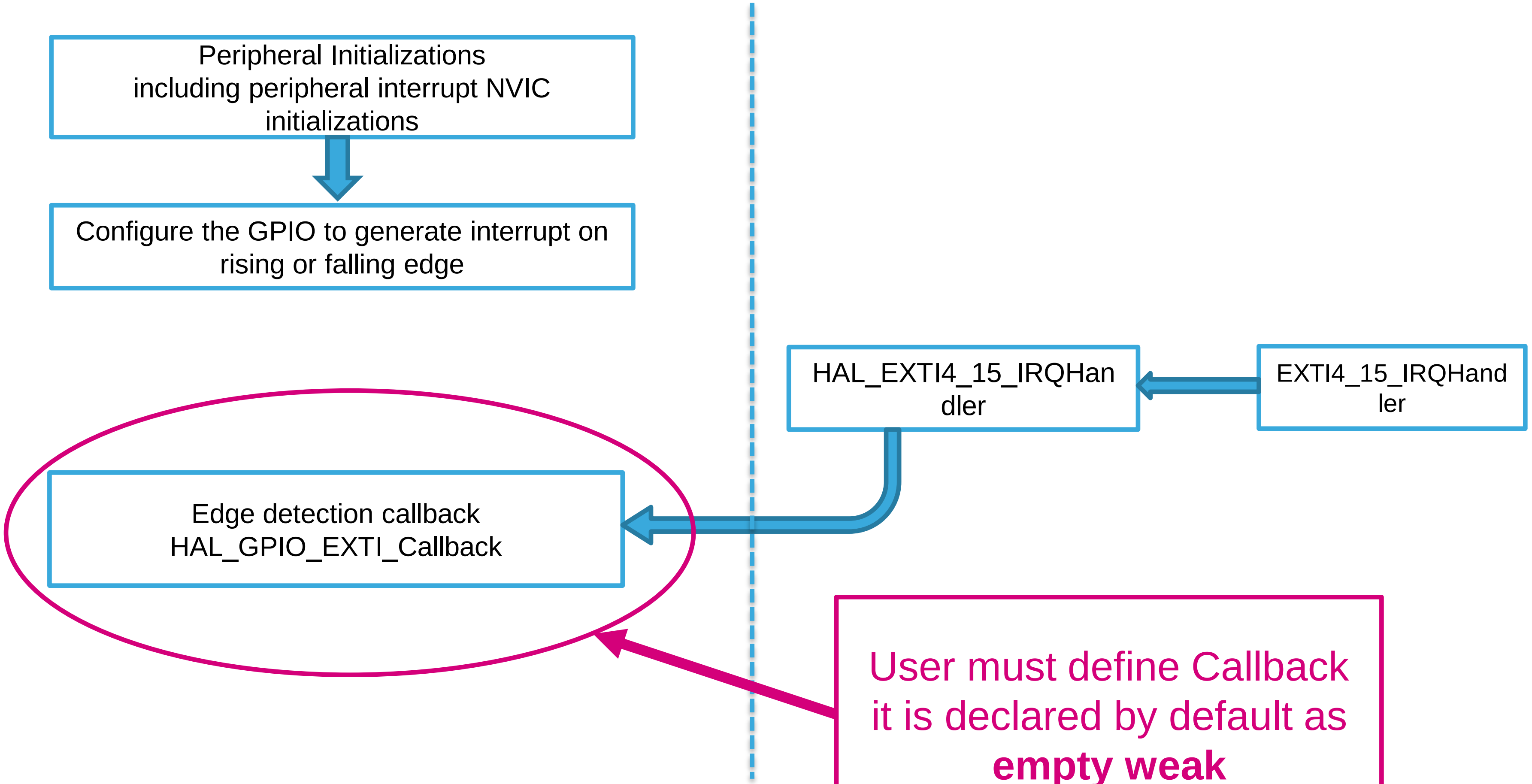
1.1.2 Configure EXTI to turn on LED

HAL Library working flow 3


Peripheral Initializations including peripheral interrupt NVIC initializations
Configure the GPIO to generate interrupt on rising or falling edge
Edge detection callback HAL_GPIO_EXTI_Callback
inside stm32f0xx_it.c
HAL_EXTI4_15_IRQHandler
EXTI4_15_IRQHandler

1.1.2 Configure EXTI to turn on LED

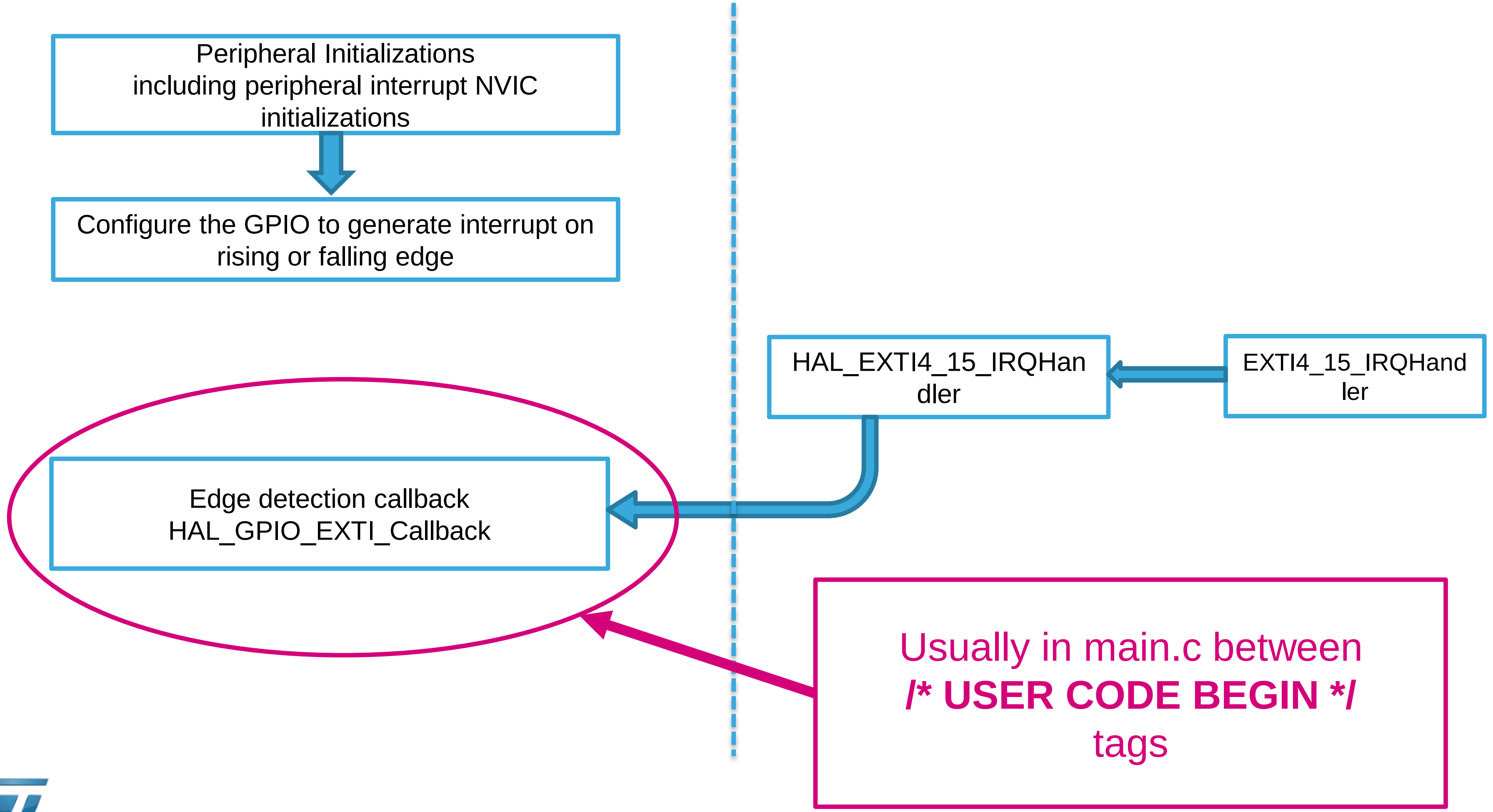
HAL Library work flow 4


Peripheral Initializations including peripheral interrupt NVIC initializations
Configure the GPIO to generate interrupt on rising or falling edge
HAL_EXTI4_15_IRQHandler
EXTI4_15_IRQHandler
Edge detection callback HAL_GPIO_EXTI_Callback

User must define Callback it is declared by default as **empty weak**

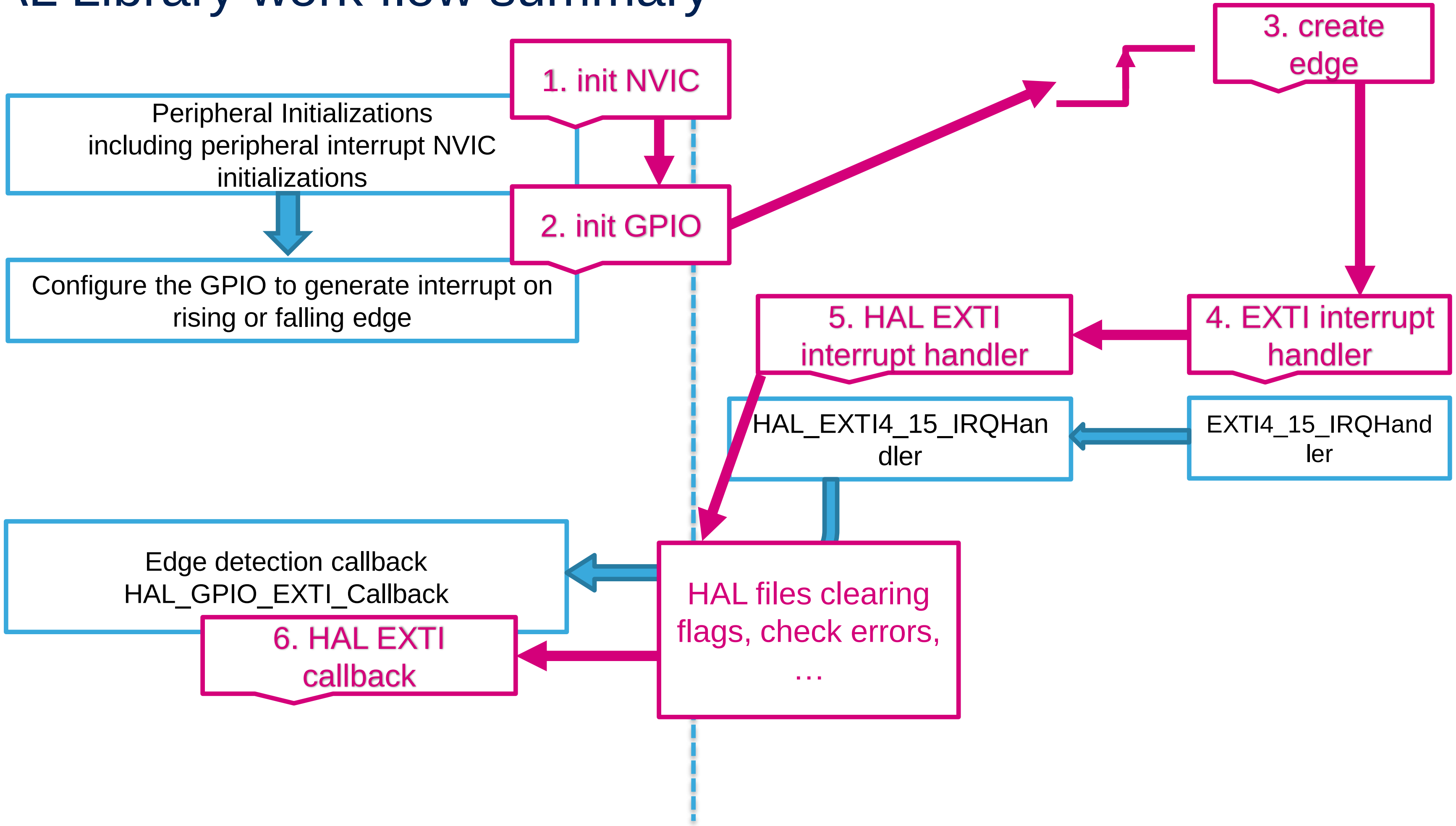
1.1.2 Configure EXTI to turn on LED

HAL Library work flow 5


Peripheral Initializations including peripheral interrupt NVIC initializations
Configure the GPIO to generate interrupt on rising or falling edge
HAL_EXTI4_15_IRQHandler
EXTI4_15_IRQHandler
Edge detection callback HAL_GPIO_EXTI_Callback
Usually in main.c between
/* USER CODE BEGIN */
tags

1.1.2 Configure EXTI to turn on LED

HAL Library work flow summary


1. init NVIC
Peripheral Initializations including peripheral interrupt NVIC initializations
2. init GPIO
Configure the GPIO to generate interrupt on rising or falling edge
3. create edge
4. EXTI interrupt handler
EXTI4_15_IRQHandler
5. HAL EXTI interrupt handler
HAL_EXTI4_15_IRQHandler
HAL files clearing flags, check errors, …
Edge detection callback HAL_GPIO_EXTI_Callback
6. HAL EXTI callback

1.1.2 Configure EXTI to turn on LED

- Now we open the project in our IDE
  - The functions we want to put into main.c
  - Between */* USER CODE BEGIN 4 */* and */* USER CODE END 4 */* tags
  - We create function which will handle the EXTI interrupts
- The HAL callback function for EXTI
  - void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
- For LED turn on we need to use this functions
  - *HAL_GPIO_WritePin*

1.1.2 Configure EXTI to turn on LED

- Now we open the project in our IDE
  - The functions we want to put into main.c
  - Between */* USER CODE BEGIN 4 */* and */* USER CODE END 4 */* tags
  - We create function which will handle the EXTI interrupts
- The HAL callback function for EXTI
  - void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
- For LED turn on we need to use this functions
  - *HAL_GPIO_WritePin*

```c
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
  if(GPIO_Pin == GPIO_PIN_13) {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
  } else {
      __NOP();
  }
}
/* USER CODE END 4 */
```

life.augmented

Appendix B Documents

B

CubeMX documentation

- CubeMX user manual UM1718
  - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00104712.pdf
- CubeMX release note RN0094
  - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00104712.pdf
- CubeMX technical note TN0072
  - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/technical_note/CD00214439.pdf

B

STM32F429i-Discovery documentation

- STM32F429i-Discovery page
  - http://www.st.com/web/en/catalog/tools/FM116/SC959/SS1532/LN1848/PF259090?s_searchtype=keyword
- STM32F429i-Discovery user manual with discovery schematics
  - http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00093903.pdf

life.augmented