

# EXTI lab 2

# 2 Configure EXTI which turns on LED

45

- Objective

- Learn how to setup input pin with EXTI in CubeMX
- How to Generate Code in CubeMX and use HAL functions

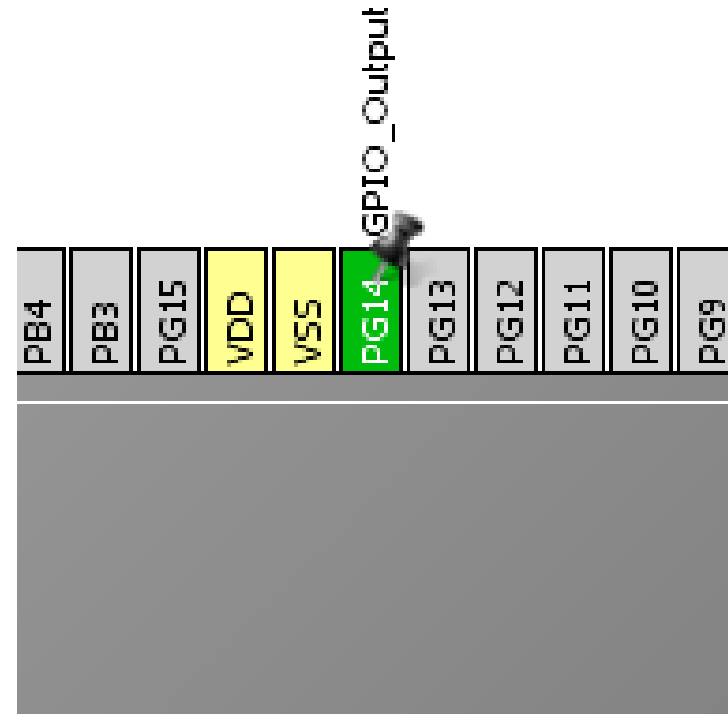
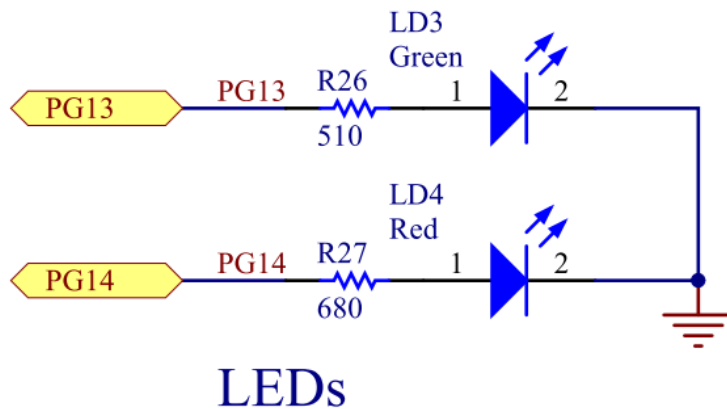
- Goal

- 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

# 2 Configure EXTI which turns on LED

46

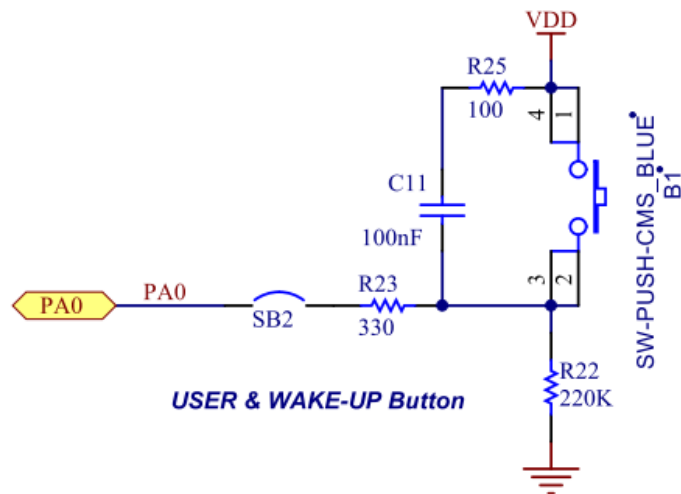
- Create project in CubeMX
  - Menu > File > New Project
  - Select STM32F4 > STM32F429/439 > LQFP144 > STM32F439ZITx
- Configure LED pin as GPIO\_Output
- Configure Button pin as GPIO\_EXTIX



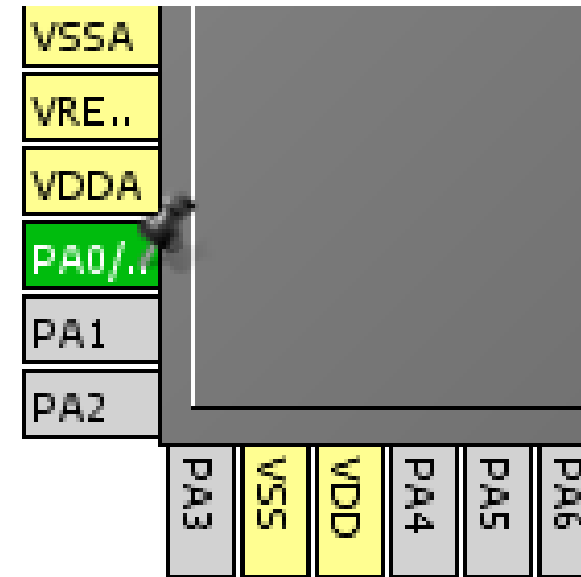
# 2 Configure EXTI which turns on LED

47

- Create project in CubeMX
  - Menu > File > New Project
  - Select STM32F4 > STM32F429/439 > LQFP144 > STM32F439ZITx
- Configure LED pin as GPIO\_Output
- Configure Button pin as GPIO\_EXTIX



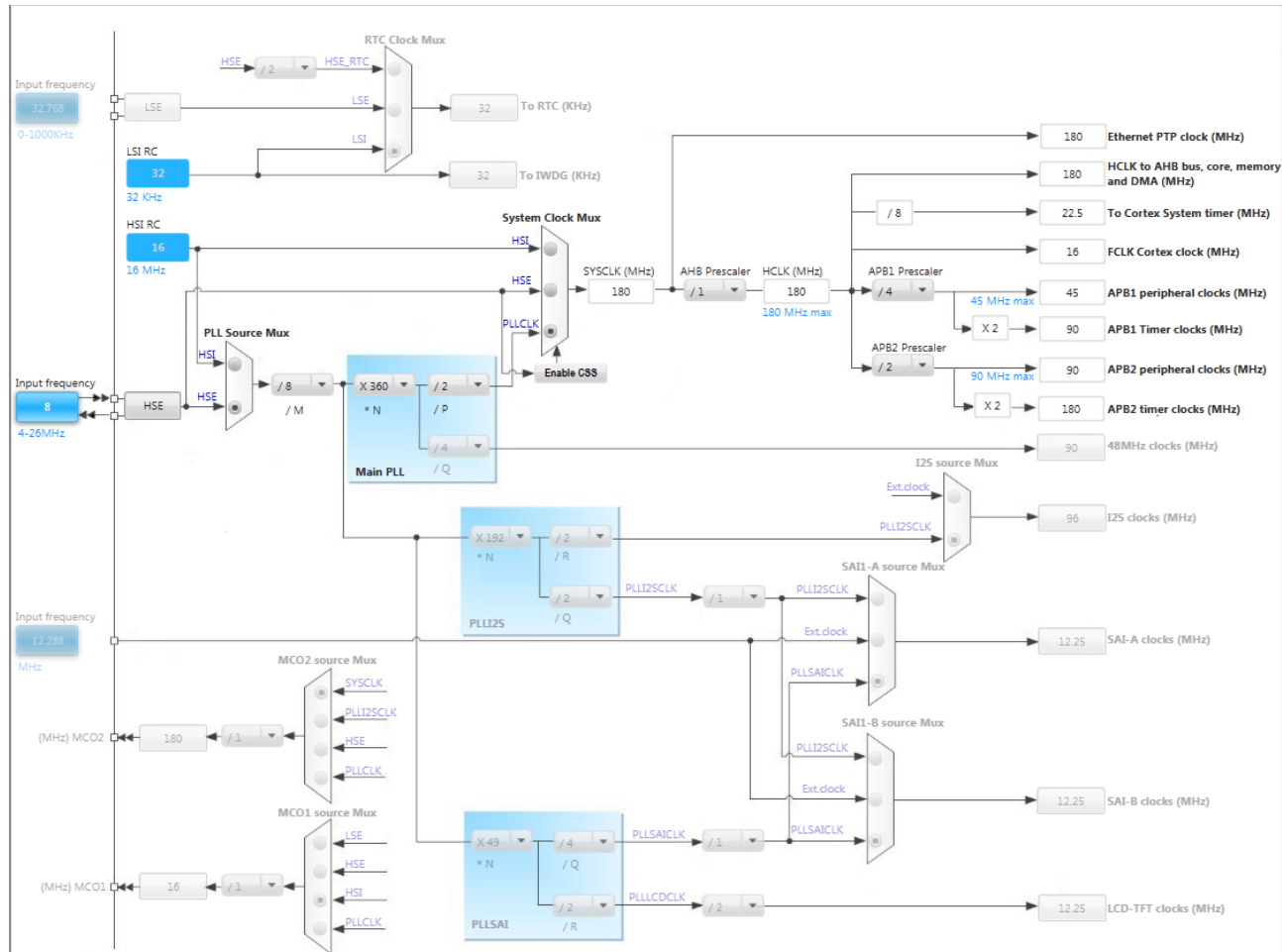
GPIO\_EXTI0



# 2 Configure EXTI which turns on LED

48

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



# 2 Configure EXTI which turns on LED

49

- GPIO Configuration
  - TAB>Configuration>System>GPIO

The screenshot shows the STM32CubeMX configuration tool interface. The 'Configuration' tab is selected and highlighted with a red box. The left sidebar shows a tree view of configuration options, including MiddleWares (FATFS, FREERTOS) and Peripherals (CRC, DMA2D, IWDG, RCC, RNG, TIM6, TIM7, WWDG). The main area displays a configuration table with columns for Multimedia, Control, Analog, Connectivity, and System. The 'System' column contains a list of peripherals: DMA, GPIO, NVIC, and RCC. The 'GPIO' entry is highlighted with a red box, indicating it is the selected peripheral for configuration.

Multimedia	Control	Analog	Connectivity	System
				DMA
				GPIO
				NVIC
				RCC

# 2 Configure EXTI which turns on LED

50

- GPIO(Pin) Configuration

- Select External Interrupt Mode with Rising edge trigger detection
- No pull-up or pull-down
- PG14 can be let in default settings
- Button OK

The image shows a 'Pin Configuration' window with two tabs: 'GPIO' and 'RCC'. The 'GPIO' tab is active. It features a 'Search Signals' field with the placeholder text 'Search (Ctrl+F)' and a checkbox labeled 'Show only Modified Pins'. Below this is a table with the following columns: 'Pin Name', 'Signal on Pin', 'GPIO mode', 'GPIO Pull-up/Pu...', 'Maximum output...', 'User Label', and 'Modified'. The table contains two rows: 'PA0/WKUP' and 'PG14'. The 'PA0/WKUP' row is highlighted with a red border. Below the table, there is a section titled 'PA0/WKUP Configuration :'. It contains three fields: 'GPIO mode' with a dropdown menu showing 'External Interrupt Mode with Rising edge trigger detection', 'GPIO Pull-up/Pull-down' with a dropdown menu showing 'No pull-up and no pull-down', and 'User Label' with an empty text box. At the bottom of the window, there is a checkbox labeled 'Group By IP' and three buttons: 'Apply', 'Ok', and 'Cancel'.

Pin Name	Signal on Pin	GPIO mode	GPIO Pull-up/Pu...	Maximum output...	User Label	Modified
PA0/WKUP	n/a	External Interrup...	No pull-up and no...	n/a		<input checked="" type="checkbox"/>
PG14	n/a	Output Push Pull	No pull-up and no...	High		<input checked="" type="checkbox"/>

PA0/WKUP Configuration :

GPIO mode: External Interrupt Mode with Rising edge trigger detection

GPIO Pull-up/Pull-down: No pull-up and no pull-down

User Label:

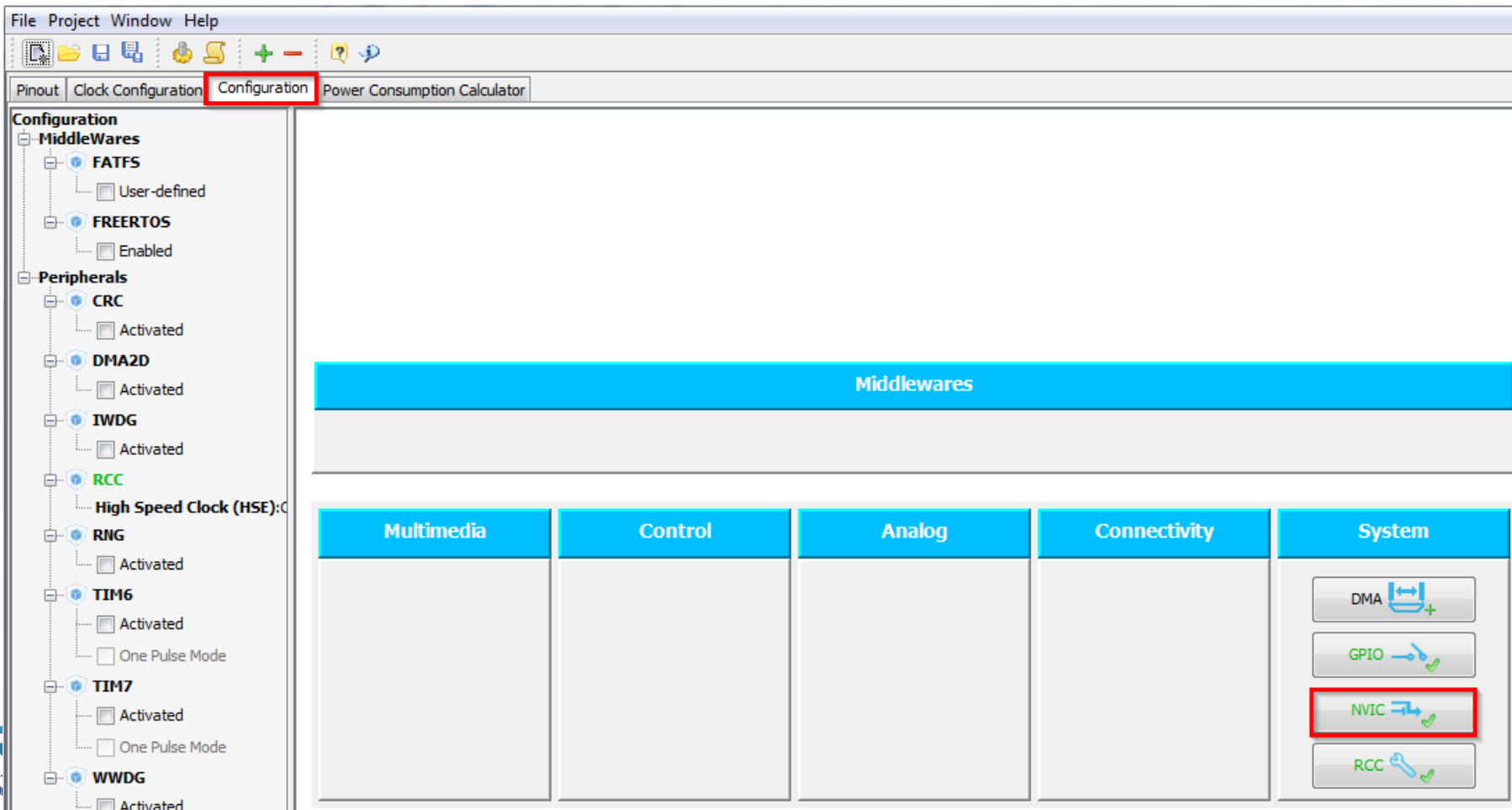
☐ Group By IP

Apply Ok Cancel

# 2 Configure EXTI which turns on LED

51

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



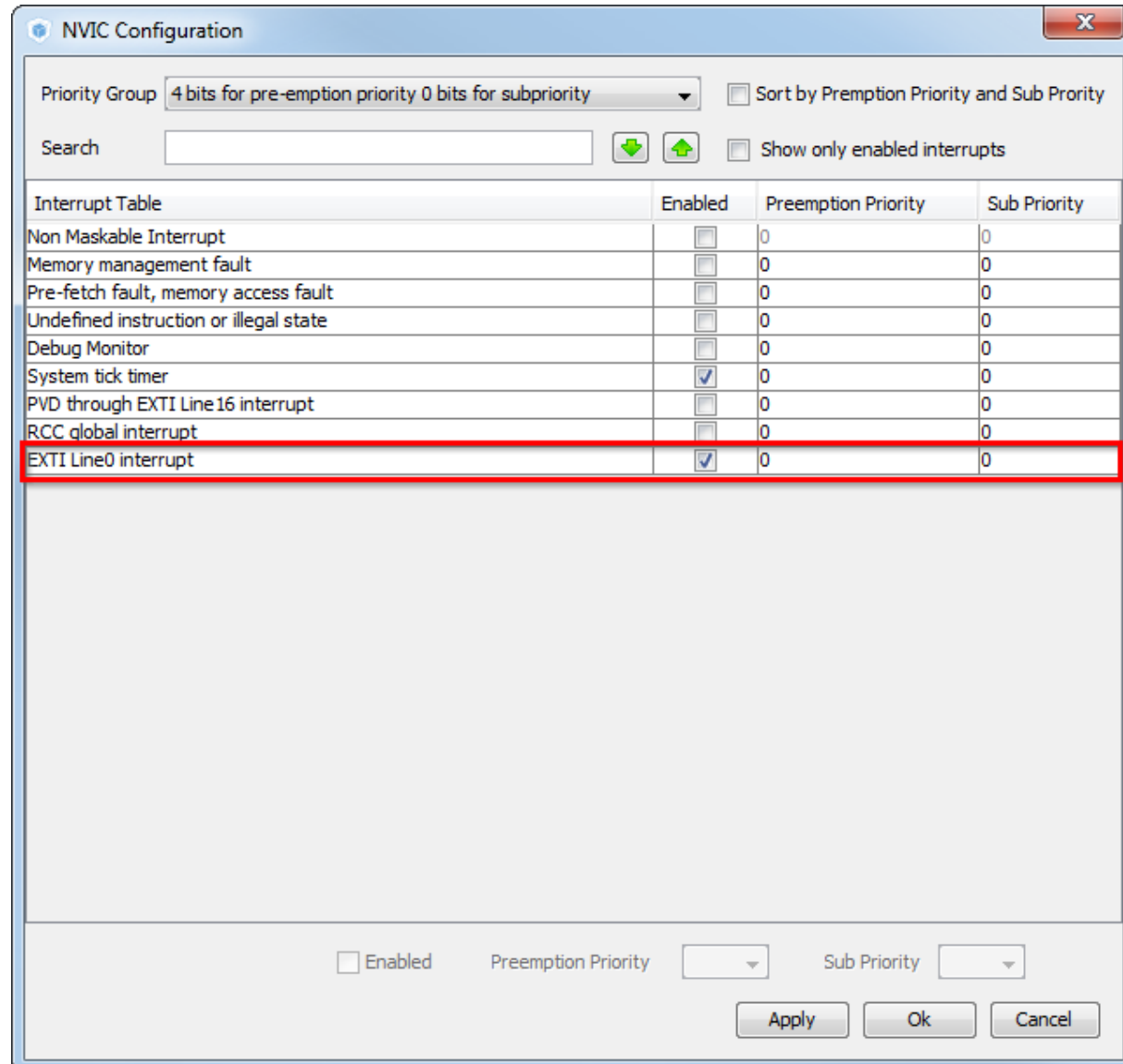


# 2 Configure EXTI which turns on LED

52

- NVIC Configuration

- Enable interrupt for EXTI Line0
- Button OK



The image shows the NVIC Configuration window. At the top, the 'Priority Group' is set to '4 bits for pre-emption priority 0 bits for subpriority'. There are checkboxes for 'Sort by Preemption Priority and Sub Priority' and 'Show only enabled interrupts'. A search bar is present. Below is a table of interrupts. The 'EXTI Line0 interrupt' row is highlighted with a red border, showing it is 'Enabled' with a checkmark in the 'Enabled' column, and 'Preemption Priority' and 'Sub Priority' are both set to 0. At the bottom, there are fields for 'Enabled' (checkbox), 'Preemption Priority' (dropdown), and 'Sub Priority' (dropdown), along with 'Apply', 'Ok', and 'Cancel' buttons.

Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non Maskable Interrupt	<input type="checkbox"/>	0	0
Memory management fault	<input type="checkbox"/>	0	0
Pre-fetch fault, memory access fault	<input type="checkbox"/>	0	0
Undefined instruction or illegal state	<input type="checkbox"/>	0	0
Debug Monitor	<input type="checkbox"/>	0	0
System tick timer	<input checked="" type="checkbox"/>	0	0
PVD through EXTI Line16 interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
EXTI Line0 interrupt	<input checked="" type="checkbox"/>	0	0

# 2 Configure EXTI which turns on LED

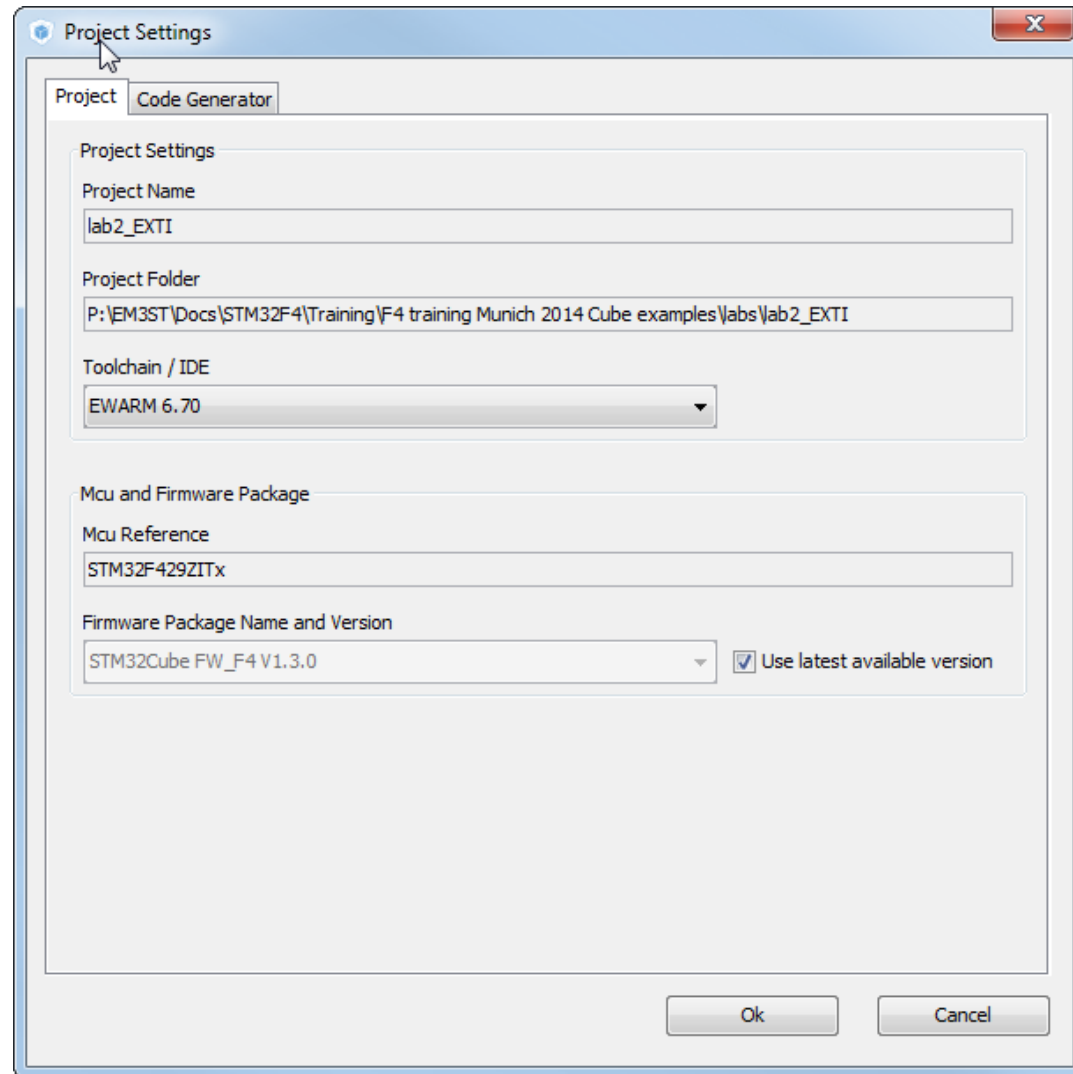
53

- 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

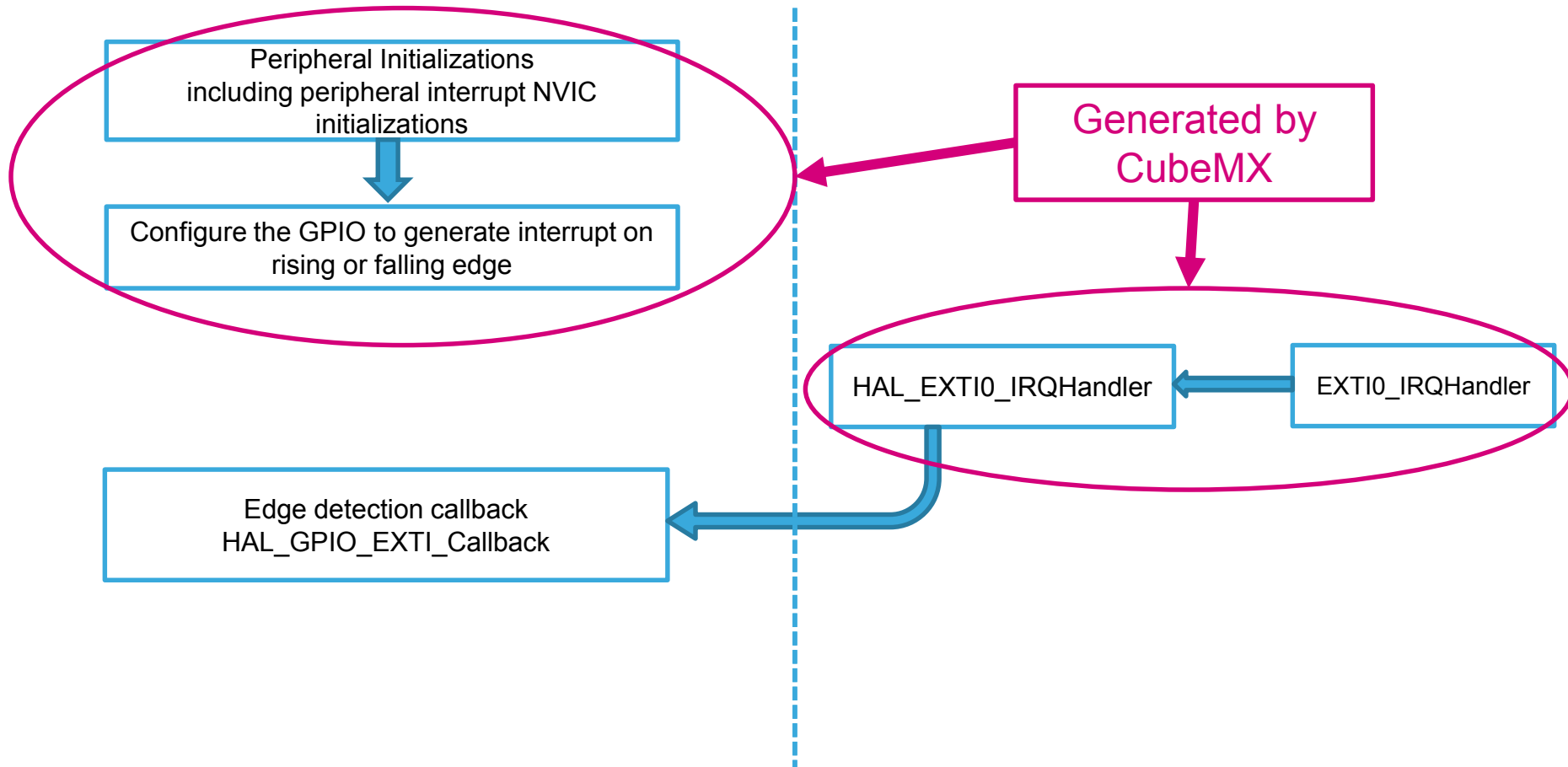
- Menu > Project > Generate Code



# 2 Configure EXTI which turns on LED

54

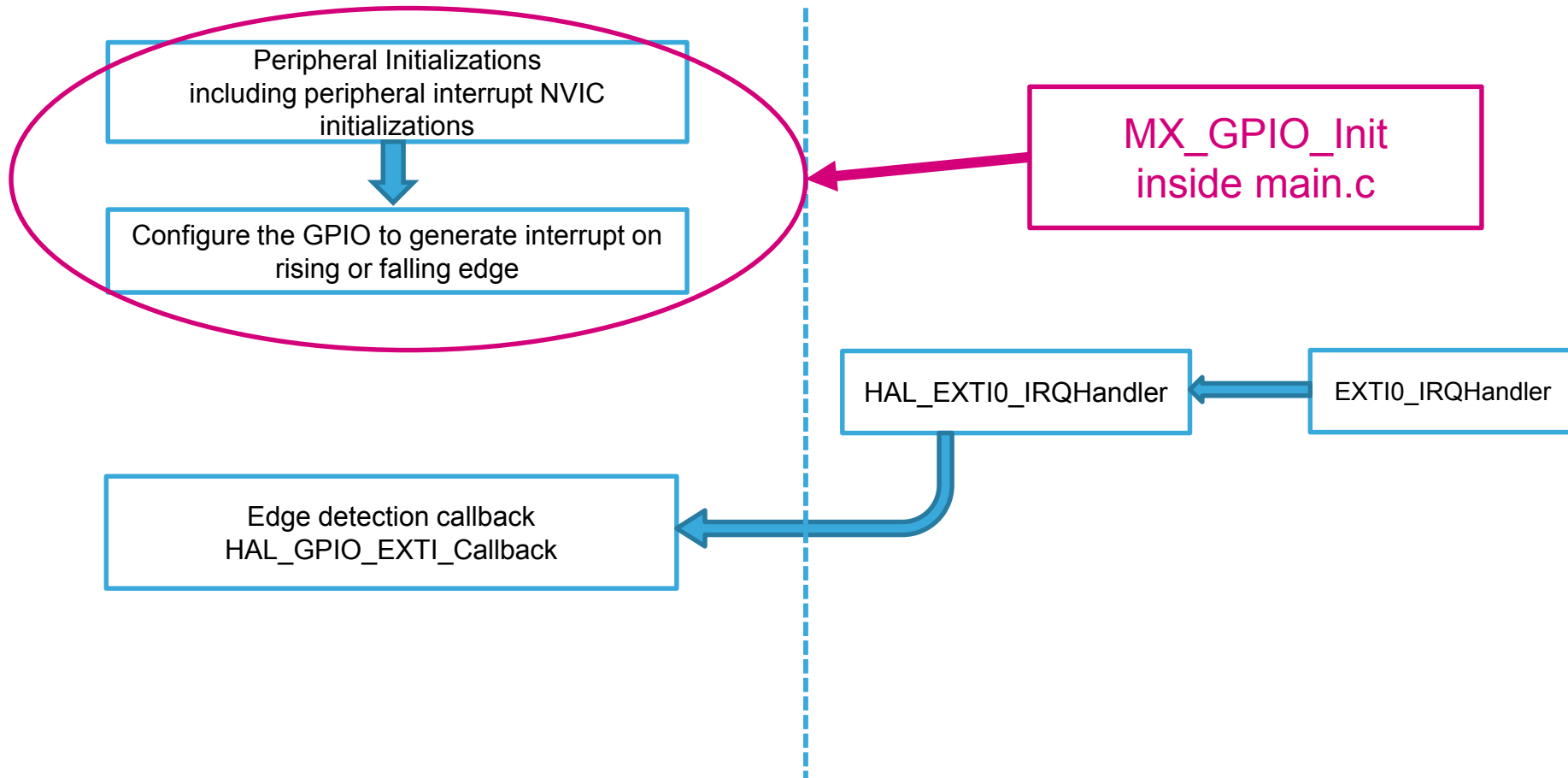
## HAL Library work flow 1



# 2 Configure EXTI which turns on LED

55

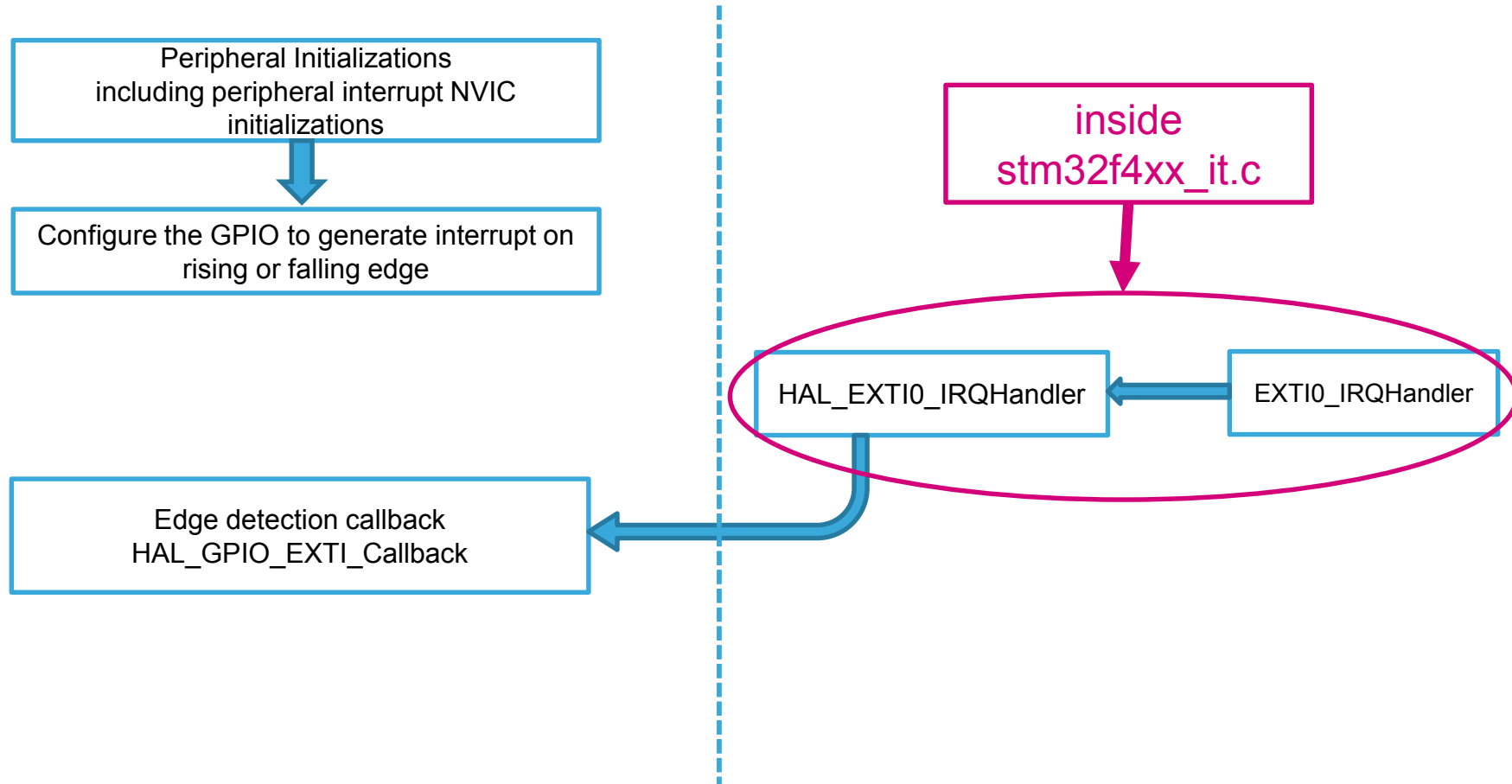
## HAL Library work flow 2



# 2 Configure EXTI which turns on LED

56

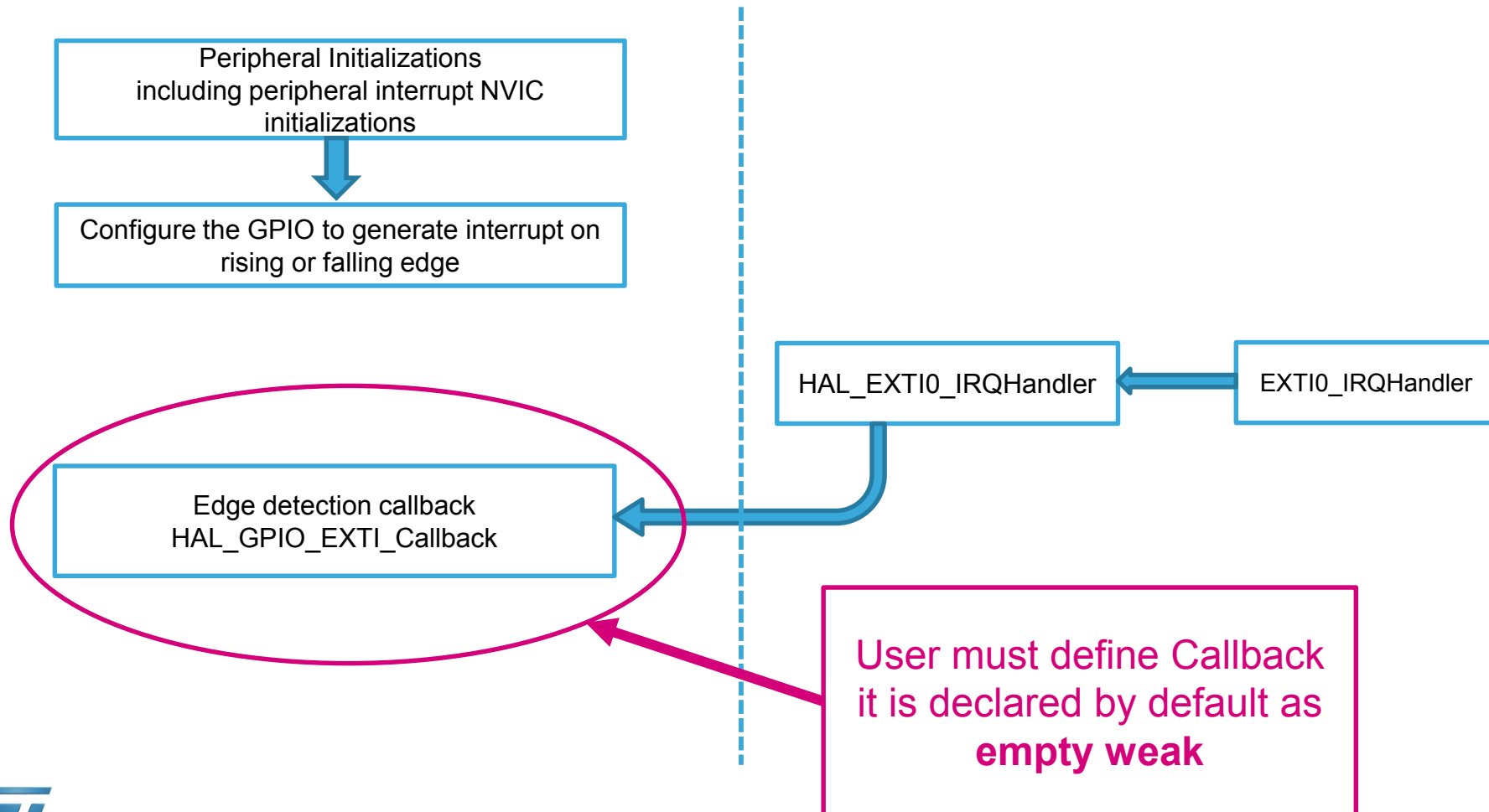
## HAL Library working flow 3



# 2 Configure EXTI which turns on LED

57

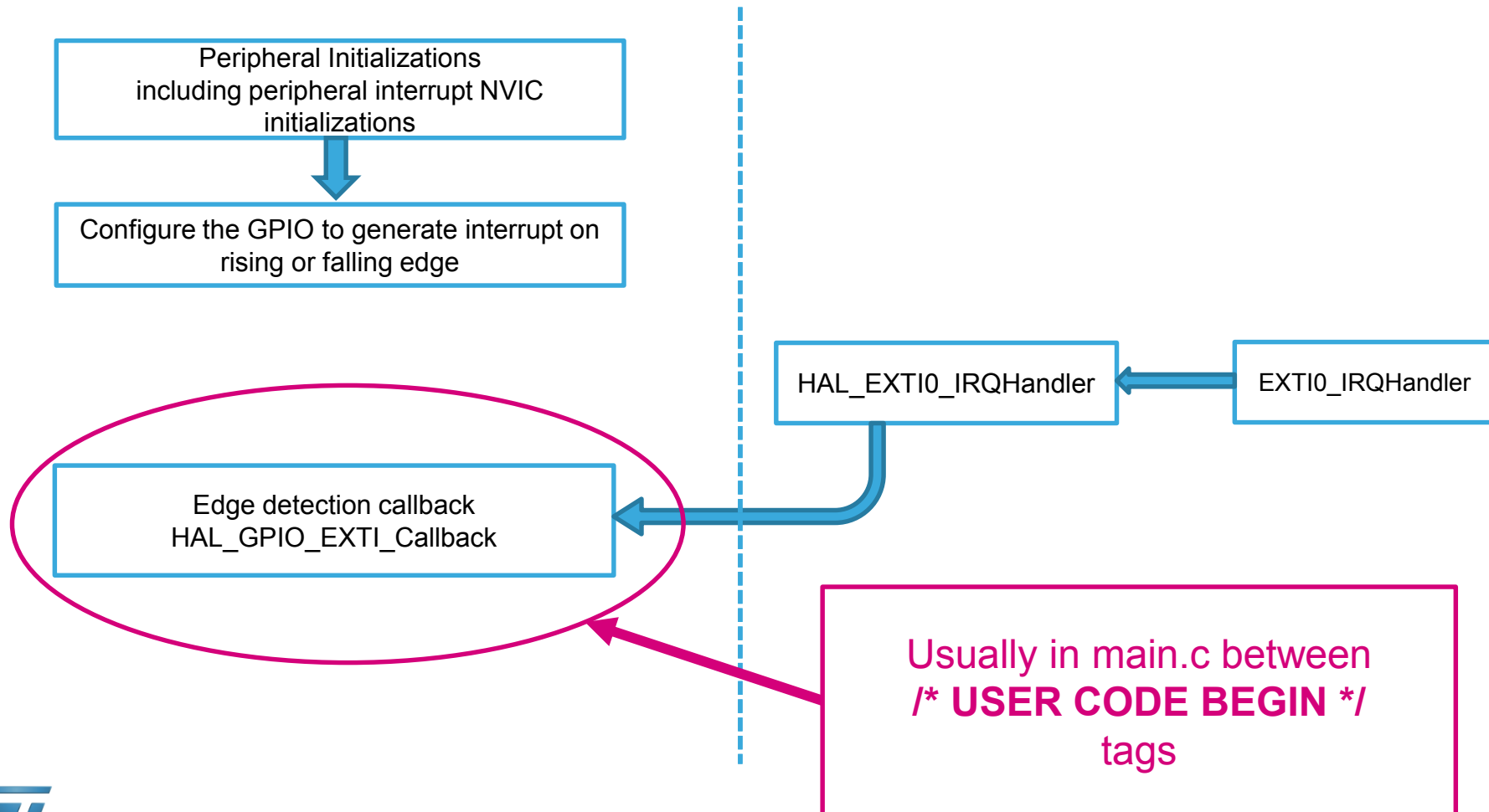
## HAL Library work flow 4



# 2 Configure EXTI which turns on LED

58

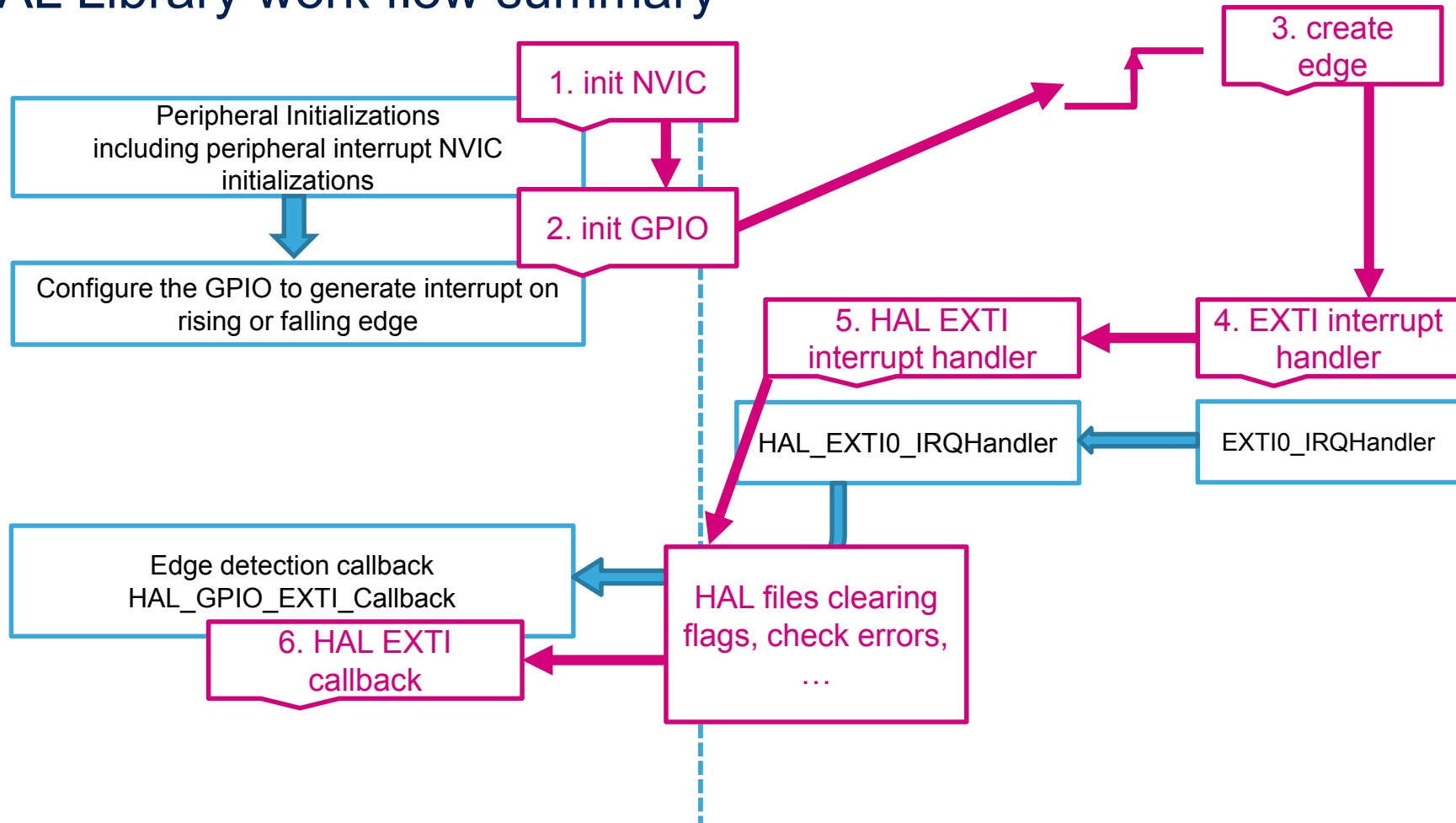
## HAL Library work flow 5



# 2 Configure EXTI which turns on LED

59

## HAL Library work flow summary





# 2 Configure EXTI which turns on LED

60

- 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`

# 2 Configure EXTI which turns on LED

61

- 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`

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_0) {
        HAL_GPIO_WritePin(GPIOG, GPIO_PIN_14, GPIO_PIN_SET);
    } else {
        __NOP();
    }
}
/* USER CODE END 4 */
```