3a laboratory work

GPIO EXTI

1. Aim

- Learn how to use STM32CubeIDE for programming of STM32 microcontrollers.
- Use basic in Embedded C language.
- Learn how to read GPIO and use EXTI interrupts.

1. Theory

In this project we are going to configure the GPIO that is connected to the user button as External Interrupt (EXTI) with rising edge trigger.

We will also configure the Interrupt Controller: the NVIC.

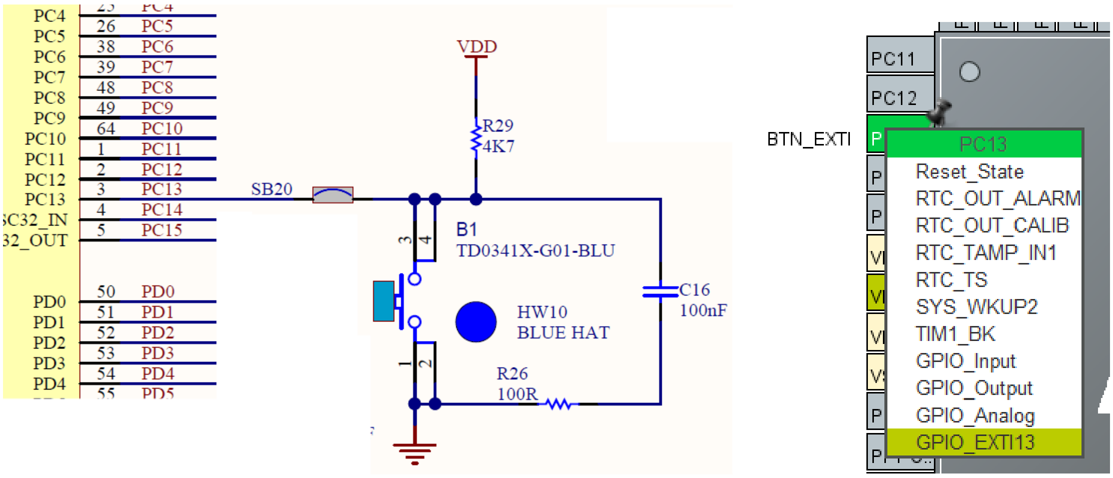
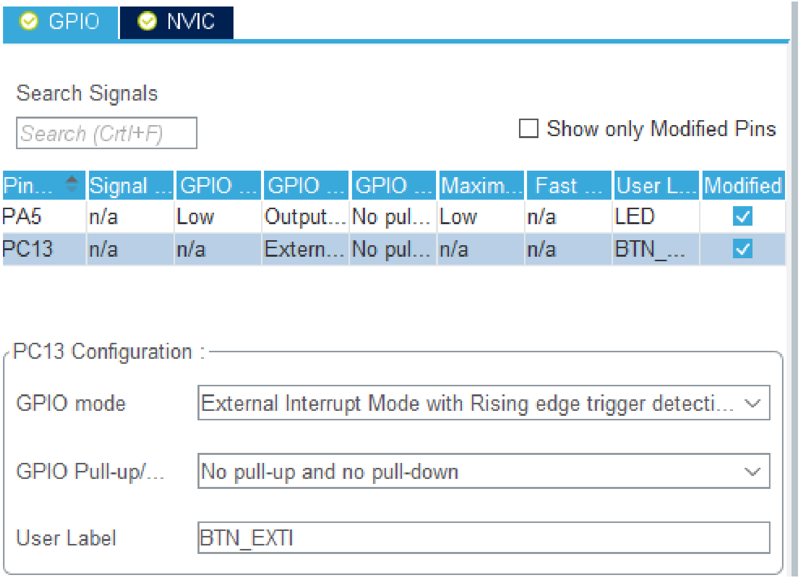


Fig 1. GPIO EXTI selection

Make sure GPIO mode is “External Interrupt Mode with Rising edge trigger detection”



Search Signals

Search (Crtl+F)

□ Show only Modified Pins

Pin...	Signal ...	GPIO ...	GPIO ...	GPIO ...	Maxim...	Fast ...	User L...	Modified
PA5	n/a	Low	Output...	No pul...	Low	n/a	LED	☑
PC13	n/a	n/a	Extern...	No pul...	n/a	n/a	BTN_...	☑

PC13 Configuration :

GPIO mode: External Interrupt Mode with Rising edge trigger detecti...

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

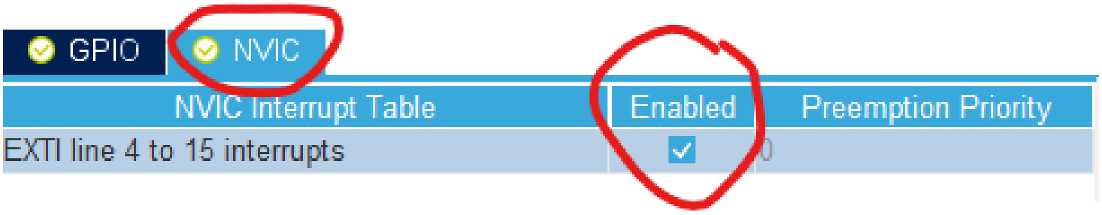
User Label: BTN_EXTI

Fig 2. GPIO EXTI settings

Select **NVIC** under System View

Fig 3. Interrupt controller tab

Enable “EXTI line 4 to 15 interrupts” (by checking the box)



NVIC Interrupt Table	Enabled	Preemption Priority
EXTI line 4 to 15 interrupts	☑	0

Fig 4. EXTI interrupt enable

Eternal Interrupt

Create flag variable.

```
/* USER CODE BEGIN PV */
uint8_t PC13_flag = 0;
/* USER CODE END PV */
```

main.c

Analyse file **`STM32G0xx_HAL_GPIO.h`** here you can see declarations of all functions you can use with GPIO.

```
/* IO operation functions *****************************************************/
GPIO_PinState     HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
void              HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin,
GPIO_PinState PinState);
void              HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
HAL_StatusTypeDef HAL_GPIO_LockPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);

void              HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin);
void              HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin);
void              HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin);
```

STM32G0xx_HAL_GPIO.h

In comparison with **`HAL_GPIO_ReadPin or HAL_GPIO_WritePin,`** EXTI functions are called automatically when the external event is presented. In our case, this is a Push-button connected to the PC13 pin of the microcontroller. When the button is pressed, on the microcontroller pin we create a **falling edge** (signal goes from +3.3V to 0V) and when we realize the push button, we create a **rising edge** (signal goes from 0V to +3.3V). When one of these events is presented, functions **HAL_GPIO_EXTI_Rising_Callback or HAL_GPIO_EXTI_Falling_Callback** are automatically called. But these functions are empty and if you will not do some actions inside – you will not see this happening. For example, you can choose one of EXTI functions from STM32G0xx_HAL_GPIO.h and fill it in with your code. In the example, below, when the button is released, code increases the variable `PC13_flag`. When `PC13_flag` is more than 10, the Pin is set to High and the variable is reset to 0.

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Rising_Callback(uint16_t GPIO_Pin) {
            PC13_flag++;
            if (PC13_flag > 10) {
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
                        PC13_flag = 0;
            } else {
                        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
            }
}
/* USER CODE END 4 */
```

main.c

2. Tasks

2.1. Create and setup STM32G070RB project.

2.2. Analyse the Nucleo board schematic from DM00452640 pdf file.

2.3. Build a circuit with two push buttons and EXTI. Count the first push-button presses and store it in X variable. If you press two times second push button - blink a LED X many times.

2.4. Write the code to change the brightness of the two LED's using two push buttons. One button press should increase the brightness of the first LED and decrease the brightness of the next and another button should do the same in opposite with changing brightness in bugger steps with each press.

2.5. Use EXTI to Write the code for toggling the LED if the push button was shortly pressed. If the button is pressed and held for 5 seconds, the LED should start blinking.

1. Report content

1) Title.
2) Main blocks of source code for tasks with comments.
3) Conclusions.

1. References

1. https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32g0-series/stm32g0x0-value-line/stm32g070rb.html