

STM32 ????? ? 3 ??GPIO

????????????????

? ?????

1. `GPIO` `GPIO` - `GPIO`
2. `EXTI` - `EXTI`
3. `GPIO` - `GPIO`

? GPIO ?????

GPIO ?????

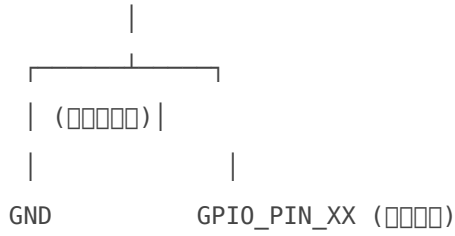
GPIO `GPIO` / `GPIO`

<code>GPIO_MODE_OUTPUT_PP</code>	<code>GPIO</code> Push-Pull	<code>GPIO</code> LED <code>GPIO</code>
<code>GPIO_MODE_INPUT</code>	<code>GPIO</code>	<code>GPIO</code>
<code>GPIO_MODE_INPUT_PU</code>	<code>GPIO</code>	<code>GPIO</code>
<code>GPIO_MODE_INPUT_PD</code>	<code>GPIO</code>	<code>GPIO</code>
<code>GPIO_MODE_AF_PP</code>	<code>GPIO</code>	UART <code>GPIO</code> SPI <code>GPIO</code>
<code>GPIO_MODE_IT_FALLING</code>	<code>GPIO</code>	<code>GPIO</code>

??????

`GPIO`





XXXX

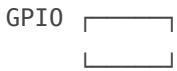
- XXXXX GPIO = XXX (3.3V) → GPIO_PIN_SET
- XXXXX GPIO = XXX (0V) → GPIO_PIN_RESET

?????Bouncing???

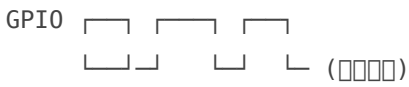
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

/XXXXXXXXXXXX

XXXXXXXXXXXX



XXXXXXXXXXXX



XXXXXXXXXXXXXXXXXXXXXXXXXXXX

?????

XX	XX	XX	XX
XXXX	XXXX	XX CPU XX	□
XXXX	XXXX	XX RC XX	□
XX +XX	XX	XXXX	□

XXXX

XXXX

XXXX

?? ?????

????

Component	Value	Notes
Push Button	1	
220Ω Resistor	1	GPIO
	2	GPIO

???

Pin	Signal	Notes
A	3.3V	
B	PA0	GPIO
PA0	GND	
LED	PA5	

CubeMX PA0

- **GPIO mode:** Input
- **Pull:** Pull-up
- **Label:** KEY_Input

???????



?? CubeMX ?????

?? 1?GPIO ?????

1. CubeMX Pinout **PA0**
2. PA0 **GPIO_Input**
3. **System Core** **GPIO**
4. **GPIOA** PA0
 - **GPIO mode:** Input
 - **Pull:** Pull-up
 - **Speed:** Medium
 - **User Label:** KEY_Input

?? 2?????????EXTI?

1. Pinout **PA0**
2. **GPIO_EXTIO**
3. **System Core** → **NVIC**
4. **EXTI line0 interrupt** **Enabled**
5.
 - **Preemption Priority:** 0
 - **Sub Priority:** 0

?? 3????????????? 2 ?????

SysTick HAL_Delay

?? 4???????

Project → **Generate Code**

? ?????

main.c - ???? + ???

```

/* STM32 Lesson 03 - Key Press with Debouncing
 *  LED 

```

```

*   *
*/

#include "main.h"
#include "gpio.h"

/*   */
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin);
void Debounce_Key(void);

/*   */
#define DEBOUNCE_DELAY_MS 20 /*   */
#define DEBOUNCE_SAMPLES 3 /*   */

/*   */
volatile uint8_t key_pressed = 0; /*   */
volatile uint32_t last_interrupt_time = 0; /*   */
uint8_t led_state = 0; /* LED 0=1= */

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();

    /* LED */
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
    led_state = 0;

    while (1)
    {
        /*   */
        if (key_pressed)
        {
            Debounce_Key();
            key_pressed = 0; /*   */
        }
    }
}

```

```

    /*  */
}
}

/**
 * @brief GPIO 
 * 
 * @param GPIO_Pin: 
 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == GPIO_PIN_0)
    {
        /* 50ms */
        uint32_t current_time = HAL_GetTick();

        if (current_time - last_interrupt_time > 50)
        {
            key_pressed = 1; /*  */
            last_interrupt_time = current_time;
        }
    }
}

/**
 * @brief 
 * 
 */
void Debounce_Key(void)
{
    uint8_t stable_count = 0;

    /*  */
    for (uint8_t i = 0; i < DEBOUNCE_SAMPLES; i++)
    {
        /*  */
        GPIO_PinState key_state = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);

        /*  */
        if (key_state == GPIO_PIN_RESET)

```

```

    {
        stable_count++;
    }

    /*          */
    HAL_Delay(DEBOUNCE_DELAY_MS);
}

/*          */
if (stable_count == DEBOUNCE_SAMPLES)
{
    /* LED  */
    led_state = !led_state;
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, led_state ? GPIO_PIN_SET : GPIO_PIN_RESET);
}
}

/**
 * @brief          2
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 8;
    RCC_OscInitStruct.PLL.PLLN = 336;
    RCC_OscInitStruct.PLL.PLLP = 2;
    RCC_OscInitStruct.PLL.PLLQ = 7;

    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
}

```

```

}

RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
                             | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief GPIO
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO */
    __HAL_RCC_GPIOA_CLK_ENABLE();

    /* PA5 LED */
    GPIO_InitStruct.Pin = GPIO_PIN_5;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    /* PA0 + */
    GPIO_InitStruct.Pin = GPIO_PIN_0;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING; /* */
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

    /* EXTI0 */

```

```

HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI0_IRQn);
}

/**
 * @brief EXTI0_IRQn
 */
void EXTI0_IRQHandler(void)
{
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
}

/**
 * @brief EXTI0_IRQn
 */
void Error_Handler(void)
{
    while (1);
}

```

??????

????

□ □□□□

1. □□□□□ **User Button** □ **PC13** □ □□□□
2. □□□□□ LED □
3. □□□□□ LED □
4. □□□□□ LED □□□

△ □□□□ □ LED □□□□□□□□

????????

□	□	□□□
□□□□	□□□□□ GPIO □□□	□□ CubeMX □ EXTIO_IRQn □ Enabled

GPIO	GPIO	GPIO
LED GPIO	GPIO	GPIO DEBOUNCE_DELAY_MS 30-50ms
GPIO	GPIO	GPIO key_pressed GPIO
GPIO EXTI0_IRQHandler GPIO	GPIO HAL GPIO	GPIO

????????

GPIO Debug GPIO

```

/* GPIO */
GPIO_PinState key_state = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
// key_state GPIO_PIN_RESET (0) = GPIO
// GPIO_PIN_SET (1) = GPIO

```

? ? ? ? ?

????????

STM32 GPIO Priority Grouping

- **Preemption Priority**
 - 0 > 1 > 2 ...
 -
- **Sub Priority**

```

/* */
HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0); /* =0, =0 */

```

????????

GPIO

```

#define ANTI_BOUNCE_TIME 50 /* 50ms */

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    static uint32_t last_time = 0;

```

```

uint32_t now = HAL_GetTick();

/* 50ms */
if (now - last_time > ANTI_BOUNCE_TIME)
{
    key_pressed = 1;
    last_time = now;
}
}

```

?????

?? 1??????

XXXXXXXXXX

UART XXXXXXXX

UART XXXX

?? 2??????

XXXXXXXXXX

2 XXXXXXXX

```

void Handle_Long_Press(void)
{
    uint32_t press_time = HAL_GetTick();

    while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) == GPIO_PIN_RESET)
    {
        if (HAL_GetTick() - press_time > 2000) /* 2 s */
        {
            /*  */
            return;
        }
    }
}
}

```

?? 3???????

XXXXXXXXXXXXXXXXXXXX

