

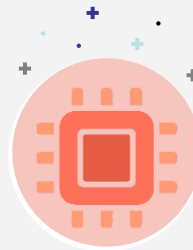


In The Name of God

Faculty of Electrical Engineering
Microprocessor Laboratory



V1 (@2025)



Microprocessor LAB

Lecture 5:

External Interrupt In STM32

By: M.Razeghizadeh

Start now!

یک ریزپردازنده می‌تواند چندین ماژول را از طریق دو روش زیر سرویس‌دهی کند:

1. وقفه (Interrupt):

هرگاه یکی از ماژول‌ها به سرویس نیاز داشته باشد، با ارسال یک سیگنال وقفه، پردازنده را مطلع می‌کند. وقتی پردازنده این سیگنال را دریافت می‌کند، اجرای برنامه‌ی فعلی خود را موقتاً متوقف کرده و به سرویس‌دهی ماژول مربوطه می‌پردازد.

2. پرس‌وجو (Polling):

در این روش، پردازنده به‌صورت مداوم وضعیت ماژول‌ها را بررسی می‌کند. هرگاه وضعیت خاصی (نیاز به سرویس) در یکی از ماژول‌ها شناسایی شود، پردازنده عملیات سرویس‌دهی به آن ماژول را انجام می‌دهد.

Polling vs Interrupt

POLLING



INTERRUPT





Polling vs Interrupt

```
int main()
{
    while(1){
        . . .
    }
}

OnSwitch_ISR{
    getData()
}
```

Interrupt

```
int main()
{
    while(1){

        if(switch == on ){
            getData(); }

        . . .
    }
}
```

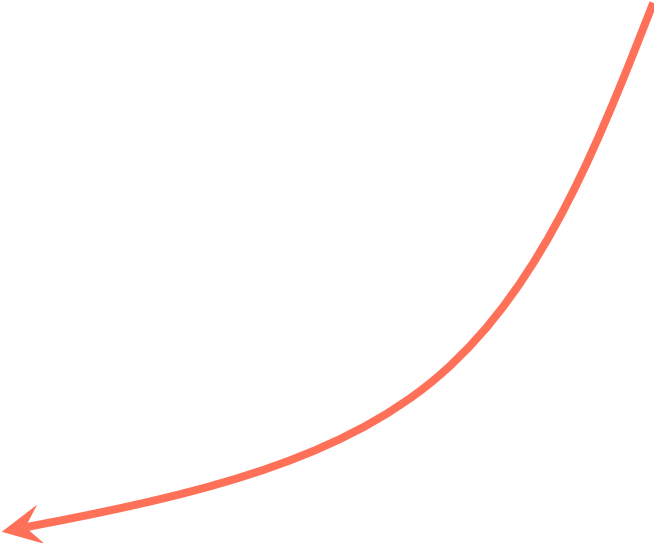
Polling

روال سرویس وقفه (Interrupt Service Routine):

تابعی که هنگام وقوع یک وقفه اجرا می‌شود، روال سرویس وقفه یا مدیر وقفه (Interrupt Handler) نام دارد. این تابع وظیفه دارد پس از دریافت سیگنال وقفه، کار لازم را برای پاسخ‌گویی به آن انجام دهد و سپس کنترل را به برنامه‌ی اصلی بازگرداند.

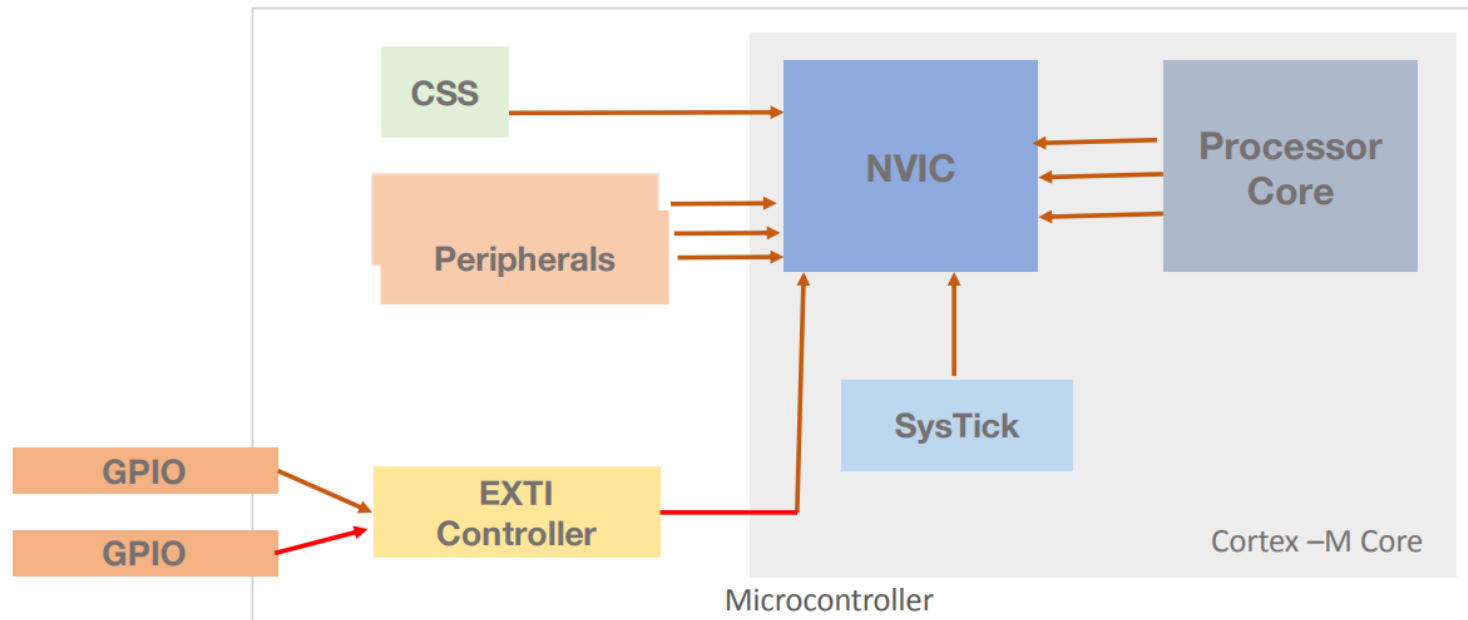
```
int main()
{
    while(1){
        . . .
    }
}

OnSwitch_ISR{
    getData()
}
```



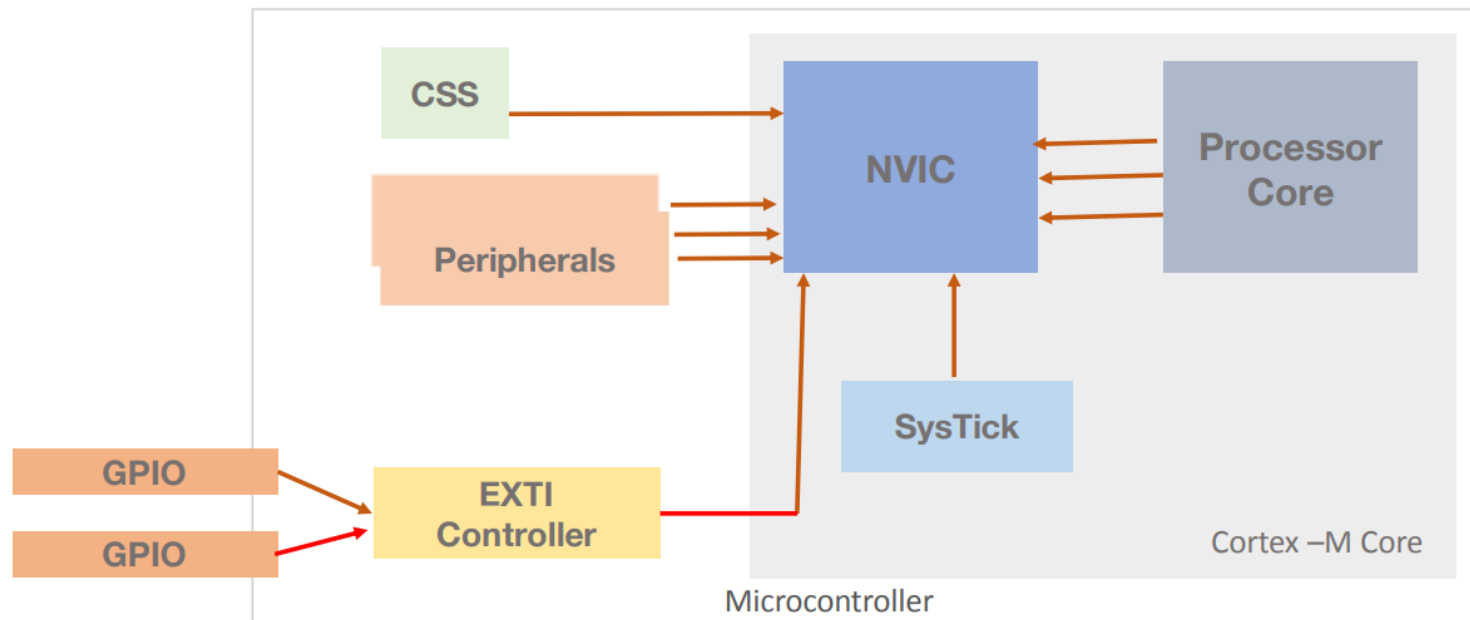
کنترل کننده وقفه‌های تو در تو (Nested Vector Interrupt Controller)

NVIC یک سخت‌افزار اختصاصی درون میکروکنترلرهای خانواده Cortex است. این واحد وظیفه دارد مدیریت و کنترل وقفه‌ها را بر عهده بگیرد و ترتیب و اولویت اجرای آن‌ها را مشخص کند.



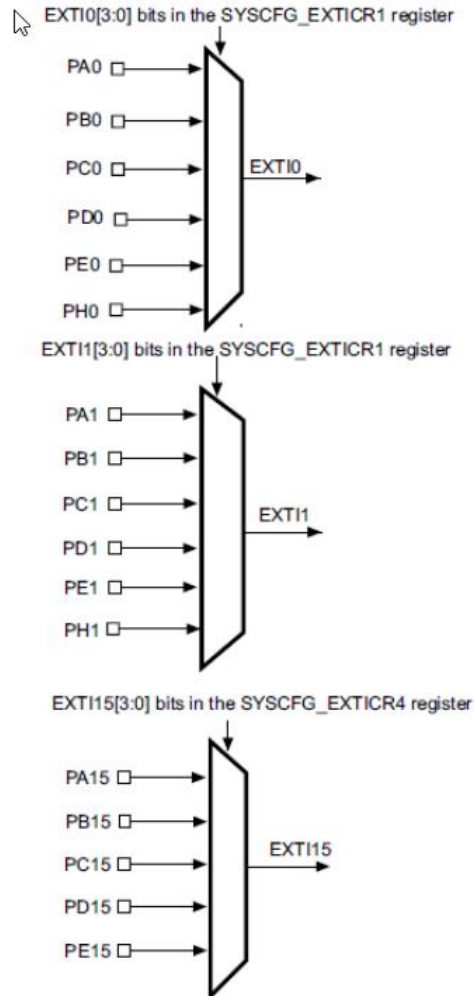
کنترل کننده وقفه های تو در تو (Nested Vector Interrupt Controller)

- وقفه هایی که از درون هسته ی پردازنده ایجاد می شوند، **رویدادهای داخلی (Exceptions)** نام دارند.
- وقفه هایی که از خارج از هسته ی پردازنده ایجاد می شوند، **وقفه های سخت افزاری (Hardware Interrupts)** یا **درخواست های وقفه (Interrupt Requests - IRQ)** نامیده می شوند.



خطوط وقفه خارجی (External Interrupt Lines – EXTI)

Figure 30. External interrupt/event GPIO mapping



• پایه‌های GPIO به خطوط EXTI متصل هستند.

• می‌توان برای هر پایه‌ی GPIO وقفه را فعال کرد.

• چندین پایه می‌توانند یک خط EXTI مشترک داشته باشند.

• پایه 0 از هر پورت به خط EXTI0_IRQ متصل است.

• پایه 1 از هر پورت به خط EXTI1_IRQ متصل است.

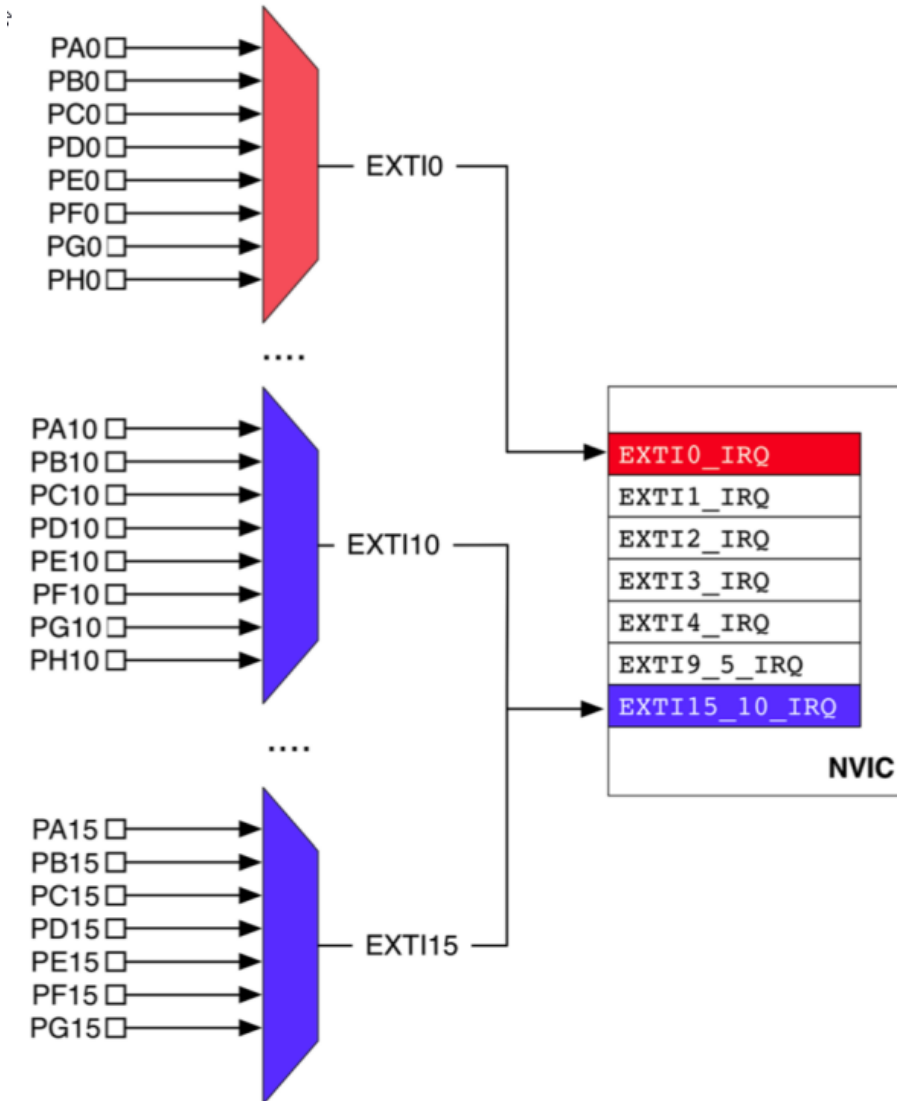
• پایه 2 از هر پورت به خط EXTI2_IRQ متصل است.

• پایه 3 از هر پورت به خط EXTI3_IRQ متصل است.

...

و به همین ترتیب برای سایر پایه‌ها... این بدان معناست که نمی‌توان به‌طور هم‌زمان از پایه‌های PA0 و PB0 به‌عنوان ورودی وقفه استفاده کرد، چون هر دو به یک مالتی‌پلکسر یعنی EXTI0 متصل‌اند. همین محدودیت برای پایه‌هایی مانند PC4 و PB4 نیز وجود دارد.

خطوط وقفه خارجی (EXTI) (External Interrupt Lines – EXTI)



پایه‌های ۱۰ تا ۱۵ همگی درون NVIC از یک IRQ مشترک استفاده می‌کنند، بنابراین همگی توسط یک روال سرویس وقفه (ISR) رسیدگی می‌شوند.

در نتیجه، کد برنامه باید بتواند تشخیص دهد کدامیک از پایه‌های ۱۰ تا ۱۵ باعث ایجاد وقفه شده است. به بیان دیگر، در ISR باید وضعیت خطوط EXTI بررسی شود تا منبع واقعی وقفه (مثلاً PA12 یا PB14) شناسایی گردد.

انواع حالات وقفه‌ها

✓ Disabled (غیرفعال):

وقفه غیرفعال است و پردازنده آن را نادیده می‌گیرد.

✓ Enabled (فعال):

این حالت پیش‌فرض است؛ وقفه فعال بوده و آماده‌ی دریافت سیگنال از منبع خود است.

✓ Pending (در انتظار سرویس):

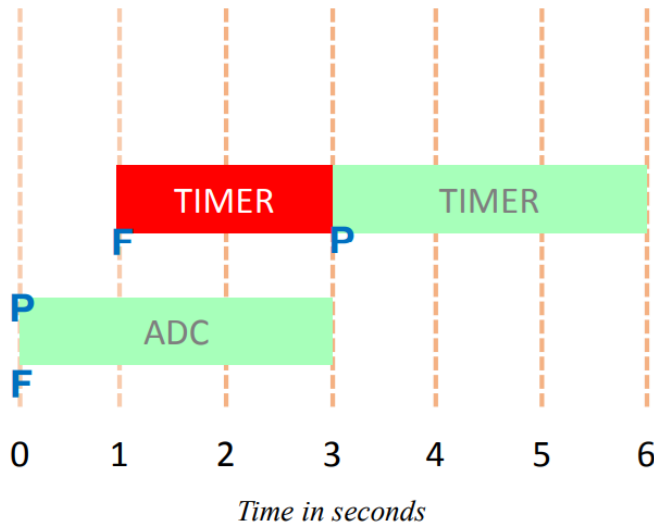
وقفه درخواست داده شده ولی هنوز سرویس‌دهی شروع نشده است.

✓ Active (در حال سرویس‌دهی):

پردازنده در حال اجرای روال سرویس وقفه (ISR) برای آن وقفه است.

تفاوت بین حالت‌های Pending و Active در وقفه‌ها

مثال: فرض کنید دو وقفه داریم: ADC و TIMER که وقفه‌ی ADC اولویت بالاتری نسبت به وقفه تایمر TIMER دارد.



- حالت فعال
- حالت در انتظار
- P پاک شدن حالت در انتظار
- F وقفه فعال شد

به طور خلاصه:

- **Pending:** وقفه درخواست داده شده ولی هنوز نوبتش برای اجرا نرسیده.
- **Active:** وقفه در حال اجراست.

- در زمان $t = 0$ ، وقفه‌ی ADC فعال می‌شود (با علامت F مشخص شده است). چون وقفه‌ی دیگری در حال اجرا نیست، حالت Pending آن بلافاصله پاک شده و به حالت Active (در حال سرویس‌دهی) تغییر می‌کند. که این تغییر با علامت P نشان داده شده است.
- در زمان $t = 1$ ، وقفه‌ی TIMER فعال می‌شود (علامت F). اما چون اولویت آن کمتر از وقفه‌ی ADC است، در حالت Pending باقی می‌ماند و منتظر تمام شدن وقفه‌ی ADC می‌ماند.
- در زمان $t = 3$ ، وقفه‌ی ADC اجرای خود را کامل می‌کند. حال چون وقفه‌ی دیگری با اولویت بالاتر وجود ندارد، وضعیت Pending وقفه‌ی TIMER پاک می‌شود و به حالت Active تغییر می‌کند (علامت P).



اولویتهای وقفه

- اولویتهای اجازه می‌دهند مشخص کنیم کدام وقفه باید زودتر اجرا شود.
- همچنین امکان می‌دهند تعیین کنیم کدام وقفه می‌تواند وقفه‌ی دیگری را قطع کند.

Table 37. Vector table for STM32F411xC/E (continued)

option	priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000 0000
-3	fixed	Reset	Reset	Reset	0x0000 0004
-2	fixed	NMI	NMI	Non maskable interrupt, Clock Security System	0x0000 0008
-1	fixed	HardFault	HardFault	All class of fault	0x0000 000C
0	settable	MemManage	MemManage	Memory management	0x0000 0010
1	settable	BusFault	BusFault	Pre-fetch fault, memory access fault	0x0000 0014
2	settable	UsageFault	UsageFault	Undefined instruction or illegal state	0x0000 0018
-	-	-	-	Reserved	0x0000 001C - 0x0000 002B
3	settable	SVCall	SVCall	System Service call via SWI instruction	0x0000 002C
4	settable	Debug Monitor	Debug Monitor	Debug Monitor	0x0000 0030
-	-	-	-	Reserved	0x0000 0034
5	settable	PendSV	PendSV	Pendable request for system service	0x0000 0038
6	settable	Systick	Systick	System tick timer	0x0000 003C
0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	EXTI16 / PVD	EXTI Line 16 interrupt / PVD through EXTI line detection interrupt	0x0000 0044
2	9	settable	EXTI21 / TAMP_STAMP	EXTI Line 21 interrupt / Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
3	10	settable	EXTI22 / RTC_WKUP	EXTI Line 22 interrupt / RTC Wakeup interrupt through the EXTI line	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068
11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000 006C
12	19	settable	DMA1_Stream1	DMA1 Stream1 global interrupt	0x0000 0070
13	20	settable	DMA1_Stream2	DMA1 Stream2 global interrupt	0x0000 0074
14	21	settable	DMA1_Stream3	DMA1 Stream3 global interrupt	0x0000 0078

بخشی از جدول بردار وقفه برای یک نمونه STM32

- برخی از اولویتهای وقفه توسط ARM تعیین شده و قابل تغییر نیستند،
- برای مثال:
- RESET: اولویت 3-
- NMI: اولویت 2-
- HardFault: اولویت 1-

اعداد کوچکتر = اولویت بالاتر



Peripheral Interrupts in STM32F10x

INT#	IRQ#	Vector location	Acronym	Device
1-15	None	0000 0000 to 0000 003C		CPU Exception (set by ARM) See Table 12-1.
16	0	0000 0040	WWDG	Window Watchdog interrupt
17	1	0000 0044	PVD	PVD through EXTI line detection interrupt
18	2	0000 0048	TAMPER	Tamper interrupt
19	3	0000 004C	RTC	RTC global interrupt
20	4	0000 0050	FLASH	Flash global interrupt
21	5	0000 0054	RCC	RCC global interrupt
22	6	0000 0058	EXTI0	EXTI line 0 interrupt
23	7	0000 005C	EXTI1	EXTI line 1 interrupt
24	8	0000 0060	EXTI2	EXTI line 2 interrupt
25	9	0000 0064	EXTI3	EXTI line 3 interrupt
26	10	0000 0068	EXTI4	EXTI line 4 interrupt
27	11	0000 006C	DMA1_Channel1	DMA1 Channel1 global interrupt
28	12	0000 0070	DMA1_Channel2	DMA1 Channel2 global interrupt
29	13	0000 0074	DMA1_Channel3	DMA1 Channel3 global interrupt
30	14	0000 0078	DMA1_Channel4	DMA1 Channel4 global interrupt
31	15	0000 007C	DMA1_Channel5	DMA1 Channel5 global interrupt
32	16	0000 0080	DMA1_Channel6	DMA1 Channel6 global interrupt
33	17	0000 0084	DMA1_Channel7	DMA1 Channel7 global interrupt
34	18	0000 0088	ADC1_2	ADC1 and ADC2 global interrupt
35	19	0000 008C	CAN1_TX	CAN1 TX interrupts
36	20	0000 0090	CAN1_RX0	CAN1 RX0 interrupt

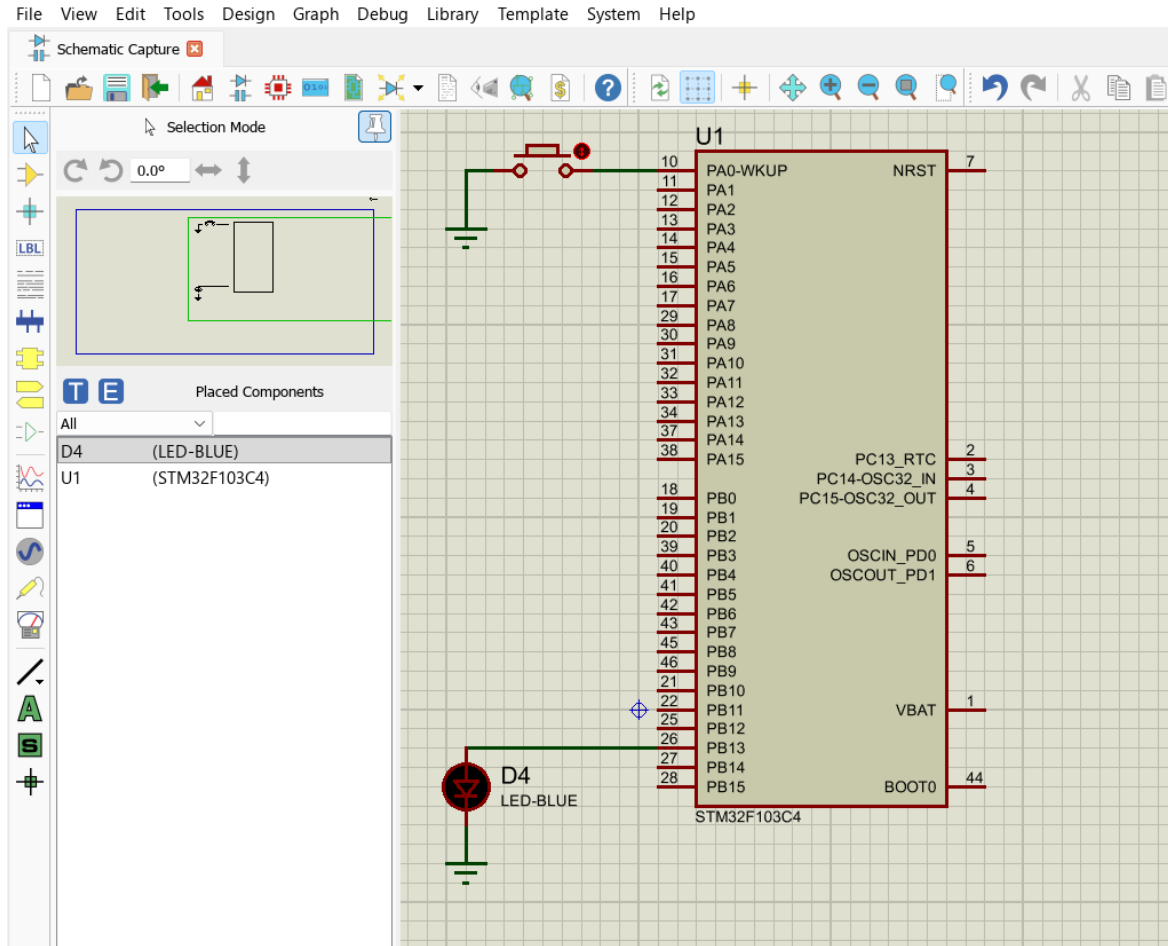


مثال کاربردی

در این مثال، که از میکروکنترلر STM32F103C4 استفاده شده است، مطابق شکل روبرو می خواهیم به کمک یک کلید فشاری متصل به پایه PA0، ال ای دی متصل به PB13 را Toggle نماییم.

فرضیات:

- ✓ تغییر وضعیت کلید به کمک واحد وقفه خارجی (EXTI) شناسایی شود.
- ✓ تغییر وضعیت کلید در هنگام رخداد لبه بالارونده شناسایی شود.

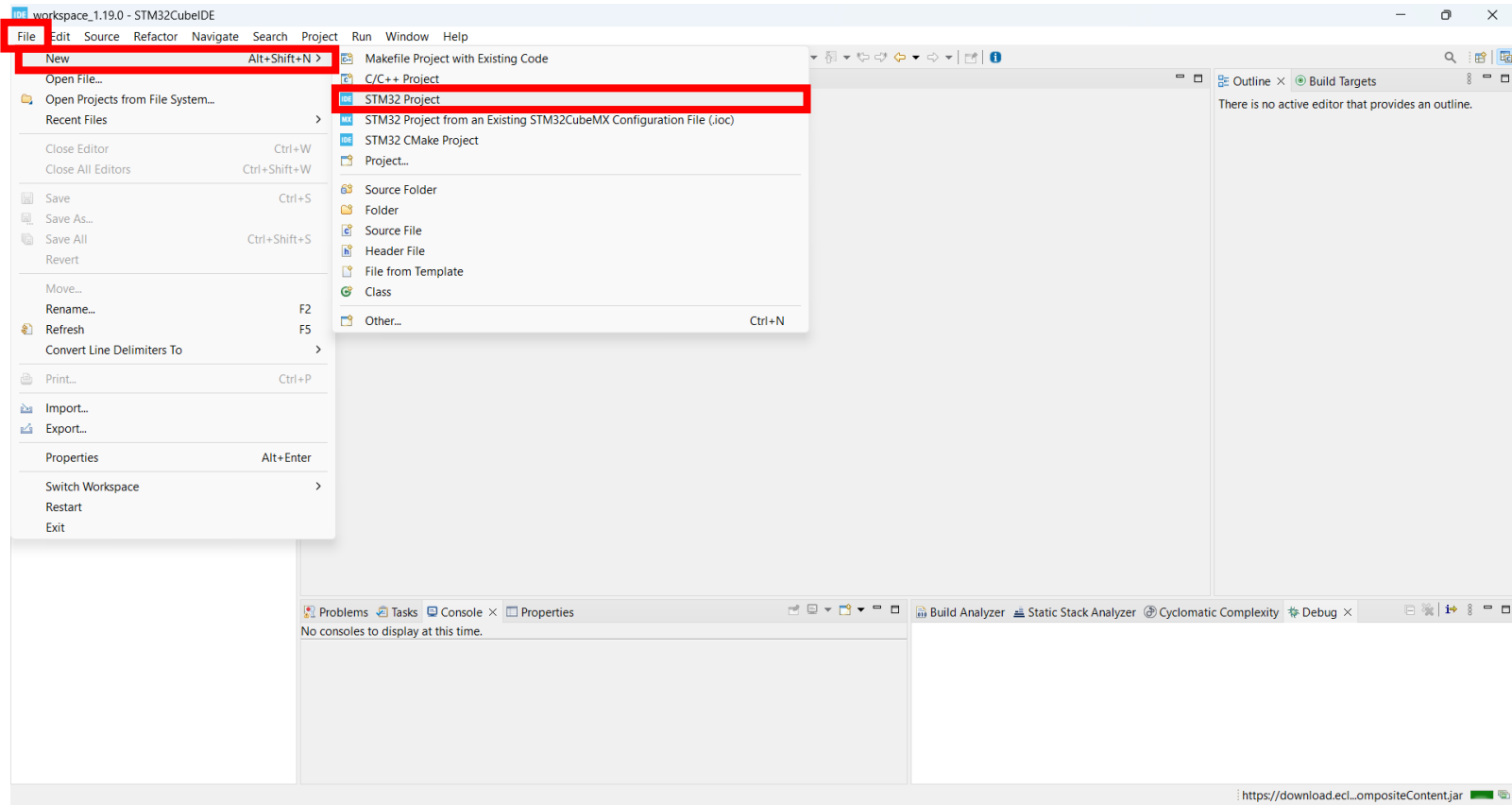


پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

ابتدا در نرم افزار STM32CubeIDE از منو File ، New ، و سپس گزینه New Project را انتخاب کنید.





پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

سپس در پنجره باز شده در قسمت Part Number Search نام میکروکنترلر مورد نظر خود را انتخاب کنید (برای مثال STM32F103C4) و بر روی Next کلیک کنید.

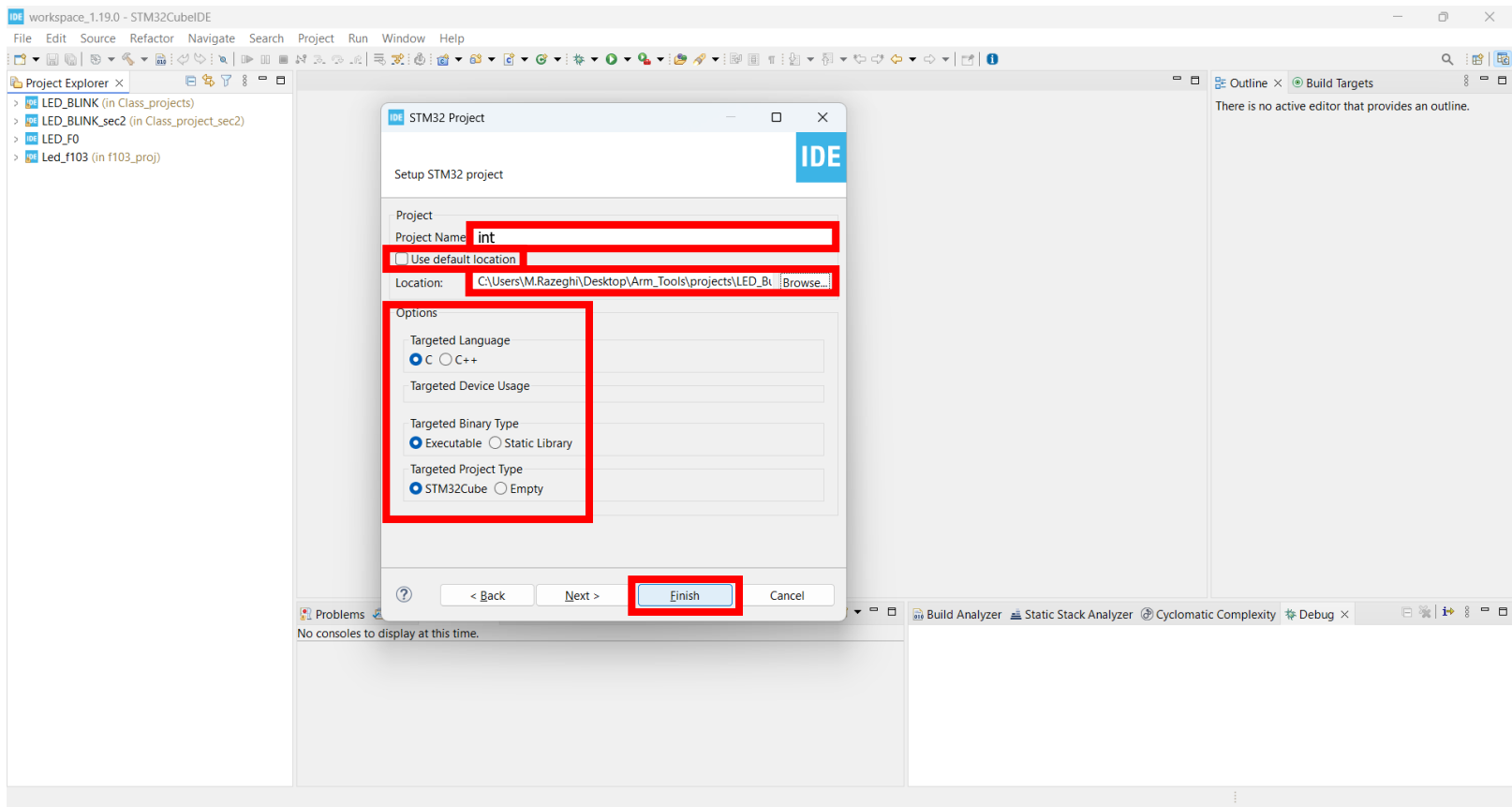
The screenshot shows the STM32CubeIDE Target Selection dialog. The 'MCU/MPU Selector' tab is selected. The 'Commercial Part Number' dropdown is set to 'stm32f103c4t6'. The 'PRODUCT INFO' section shows details for the STM32F103C4T6A, including 'Mainstream Performance line, Arm Cortex-M3 MCU with 16 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN'. The 'MCUs/MPUs List' table shows one item: STM32F103C4T6A. The 'Next >' button is highlighted.

Com...	Part No	Referen...	Marke...	Unit P...	Board	Packa...	Flash	RAM	I/O	Frequ...	
★	STM32...	STM32...	STM32...	Active	2.3424		LQFP 4...	16 kByt...	6 kBytes	37	72 MHz



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

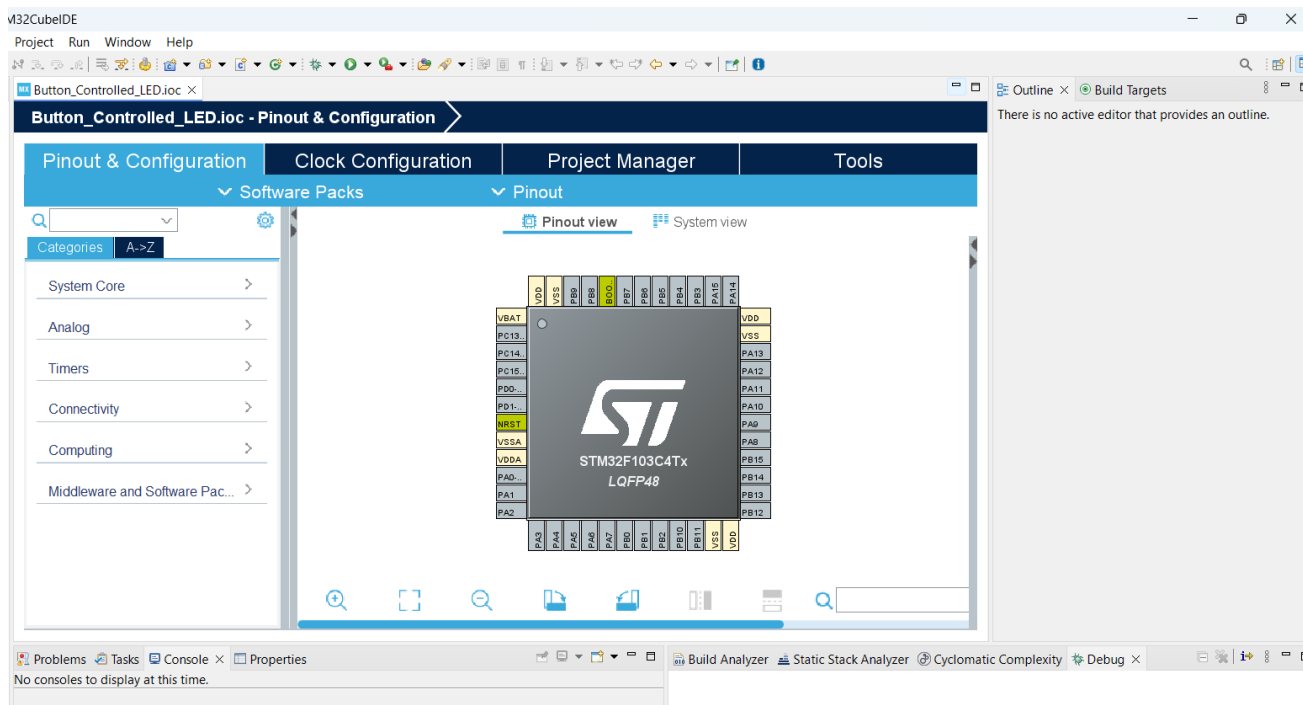
سپس در پنجره باز شده در قسمت Project Name، نام پروژه و در صورت لزوم مکان ذخیره پروژه و تنظیمات مدنظر را مطابق تصویر را وارد کرده و نهایتاً روی Finish کلیک کنید.





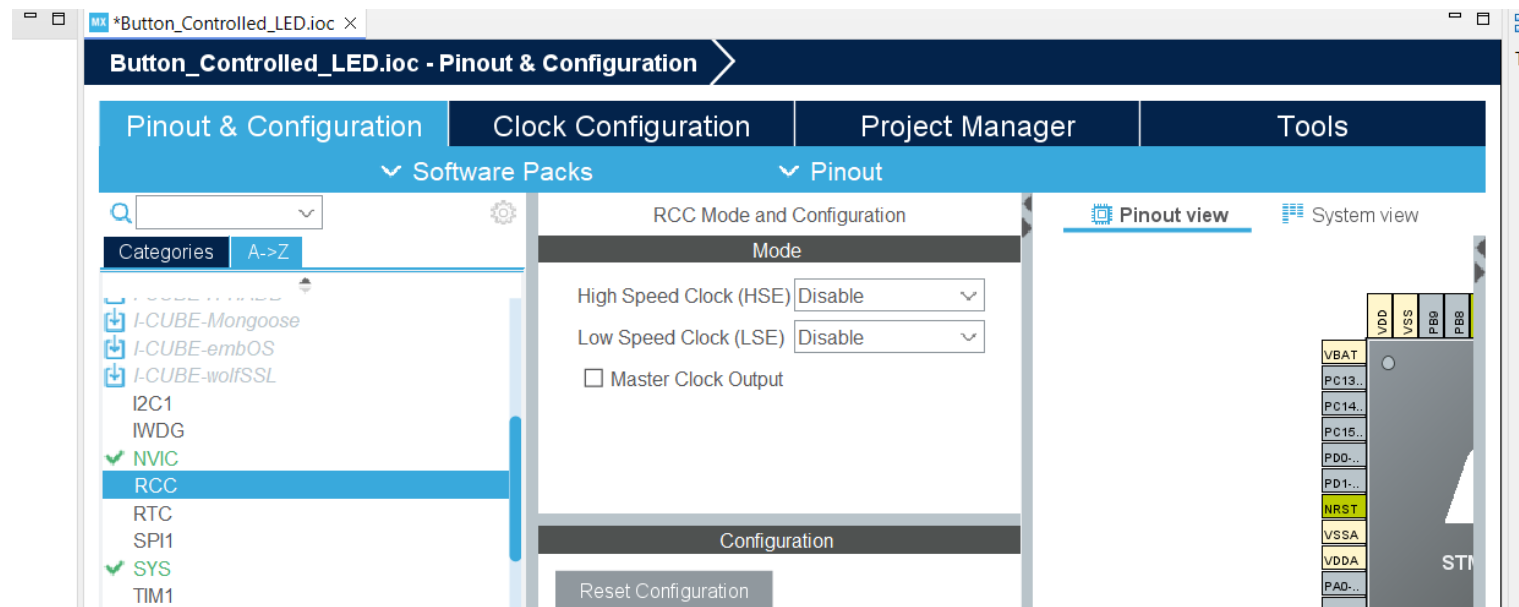
پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

پس از آن پنجره جدیدی باز می‌شود که در آن تصویری گرافیکی از میکرو و GPIOهای آن مشاهده خواهید کرد. در سمت چپ پنجره باز شده نوار ابزاری موجود است که از این قسمت می‌توانید peripheralهای میکروکنترلر را فعال و تنظیم نمایید. در این قسمت گزینه A->Z را انتخاب کنید تا تمامی peripheralهای میکروکنترلر بصورت یک لیست درآید.



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

در این قسمت بایستی دو بخش را تنظیم نمایید که در تمامی پروژه هایی که با STM32 با استفاده از نرم افزار STM32CubeMX انجام می شود مشترک است. ابتدا گزینه SYS را انتخاب نمایید و گزینه Debug را بر روی Serial Wire تنظیم نمایید. سپس وارد بخش RCC شده و منبع کلاک مدنظر را از بین موارد HSE / LSE / منبع داخلی میکرو انتخاب نمایید. در صورتی که HSE یا LSE انتخاب نشوند و Disable باشند، منبع کلاک 8 مگاهرتز داخلی انتخاب خواهد شد.

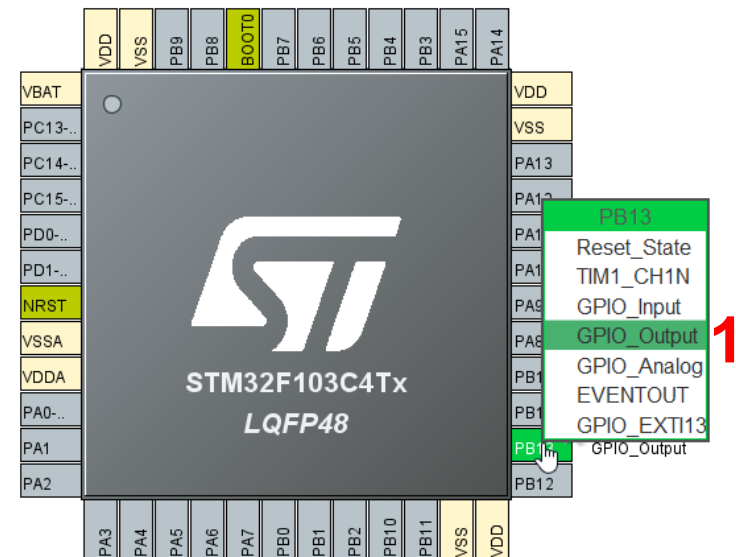
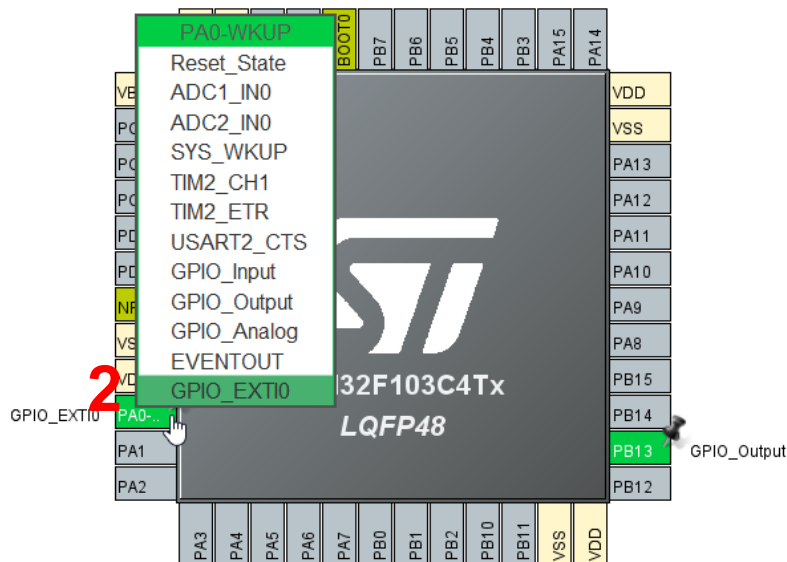


پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

حال با کلیک بر روی هر یک از پین‌ها وظیفه موردنظر را به آن اختصاص دهید. مطابق صورت سوال:

1- پین PB13 را در مد GPIO_Output برای اتصال ال ای دی به آن قرار می‌دهیم.

2- پین PA0 را در مد GPIO_EXTI0 قرار می‌دهیم تا به عنوان یک پایه برای تحریک وقفه خارجی پیکربندی شود.





پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

- بر روی System view کلیک کرده و GPIO را انتخاب نمایید.
- از پنجره configuration بر روی ردیف مربوط به تنظیمات پایه PA0 کلیک کرده تا در زیر نمایان شوند.

int.ioc - Pinout & Configuration

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

Categories | A->Z

- ADC1
- ADC2
- AIROC-Wi-Fi-Bluetooth-STM32
- CAN
- CRC
- DMA
- FATFS
- FP-SNS-MOTENVWB1
- FP-SNS-SMARTAG2
- FP-SNS-STAIOTCFT
- FP-SNS-STBOX1
- FREERTOS
- GPIO**
- I-CUBE-FS-RTOS
- I-CUBE-ITTIADB
- I-CUBE-Mongoose
- I-CUBE-embOS
- I-CUBE-wolfSSL
- I2C1
- IWDG
- NVIC**
- RCC
- RTC
- SPI1
- SYS

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO NVIC

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin	Signal	GPIO o	GPIO	GPIO P	Maximu	User La	Modified
PA0-W...	n/a	n/a	Externa	No pull...	n/a		☐
PB13	n/a	Low	Output ...	No pull...	Low		☐

Select Pins from table to configure them. Multiple selection is Allowed.

Pinout view | **System view** 1

Middleware

System Core | Analog | Timers

DMA

GPIO 2

NVIC

RCC

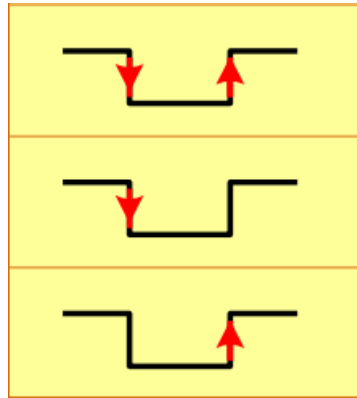
پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

- با توجه به فرضیات سوال: "تغییر وضعیت کلید در هنگام رخداد لبه بالارونده شناسایی شود"، GPIO mode را بر روی **External Interrupt Mode with Rising edge trigger detection** قرار دهید.

Rising/Falling

Falling

Rising



Clock Configuration Project Manager Tool

Software Packs Pinout

GPIO Mode and Configuration

Mode

Configuration

Group By Peripherals

GPIO NVIC

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin Name	Signal on Pin	GPIO output L...	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PA0-WKUP	n/a	n/a	External Interr...	No pull-up an...	n/a		☐
PB13	n/a	Low	Output Push ...	No pull-up an...	Low		☐

PA0-WKUP Configuration :

GPIO mode

GPIO Pull-up/Pull-down

User Label

External Interrupt Mode with Rising edge trigger detection

External Interrupt Mode with Rising edge trigger detection

External Interrupt Mode with Falling edge trigger detection

External Interrupt Mode with Rising/Falling edge trigger detection

External Event Mode with Rising edge trigger detection

External Event Mode with Falling edge trigger detection

External Event Mode with Rising/Falling edge trigger detection

GPIO_MOD



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

- با توجه به شکل ترسیم شده در صورت سوال، کلید باید با مقاومت Pull-up پیکربندی شود تا در صورت فشردن کلید با تغییر وضعیت به سطح 0 وقفه رخ دهد. لذا پارامتر GPIO Pull-up/Pull-down را بر روی Pull-up تنظیم نمایید.

Configuration

Group By Peripherals

GPIO NVIC

Search Signals

Search (Ctrl+F) ☐ Show only Modified Pins

Pin Name	Signal on Pin	GPIO output I...	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PA0-WKUP	n/a	n/a	External Interr...	Pull-up	n/a		☒
PB13	n/a	Low	Output Push ...	No pull-up an...	Low		☐

PA0-WKUP Configuration :

GPIO mode: External Interrupt Mode with Rising edge trigger detection

GPIO Pull-up/Pull-down: Pull-up

User Label: Pull-up

GPIO_PULLUP



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

- بر روی System view کلیک کرده و NVIC را انتخاب نمایید.
- با تیک زدن checkbox مربوط به EXTI Line0 interrupt، وقفه مربوطه را فعال کنید.
- در صورت نیاز به مشخص کردن اولویت وقفه از منوهای موجود در کادر شماره 4 استفاده نمایید. در این مثال چون فقط یک وقفه خارجی تعریف کرده ایم به صورت پیش فرض بر روی بالاترین اولویت موجود یعنی 0 تنظیم می کنیم.

The screenshot displays the STM32CubeIDE interface for configuring the NVIC (Nested Vectored Interrupt Controller). The 'System view' tab is selected in the top right corner (1). The 'NVIC' component is highlighted in the left sidebar (2). The 'NVIC Interrupt Table' is shown, listing various interrupts. The 'EXTI line0 interrupt' is selected (3). The 'Enabled' checkbox for 'EXTI line0 interrupt' is checked (4). The 'Preemption Priority' is set to 0 and the 'Sub Priority' is set to 0.

NVIC Interrupt Table	Enabled	Preemption Priority
Non maskable interrupt	☒	0
Hard fault interrupt	☒	0
Memory management fault	☒	0
Prefetch fault, memory access fault	☒	0
Undefined instruction or illegal state	☒	0
System service call via SWI instruction	☒	0
Debug monitor	☒	0
Pendable request for system service	☒	0
Time base: System tick timer	☒	15
PVD interrupt through EXTI line 16	☐	0
Flash global interrupt	☐	0
RCC global interrupt	☐	0
EXTI line0 interrupt	☒	0



پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

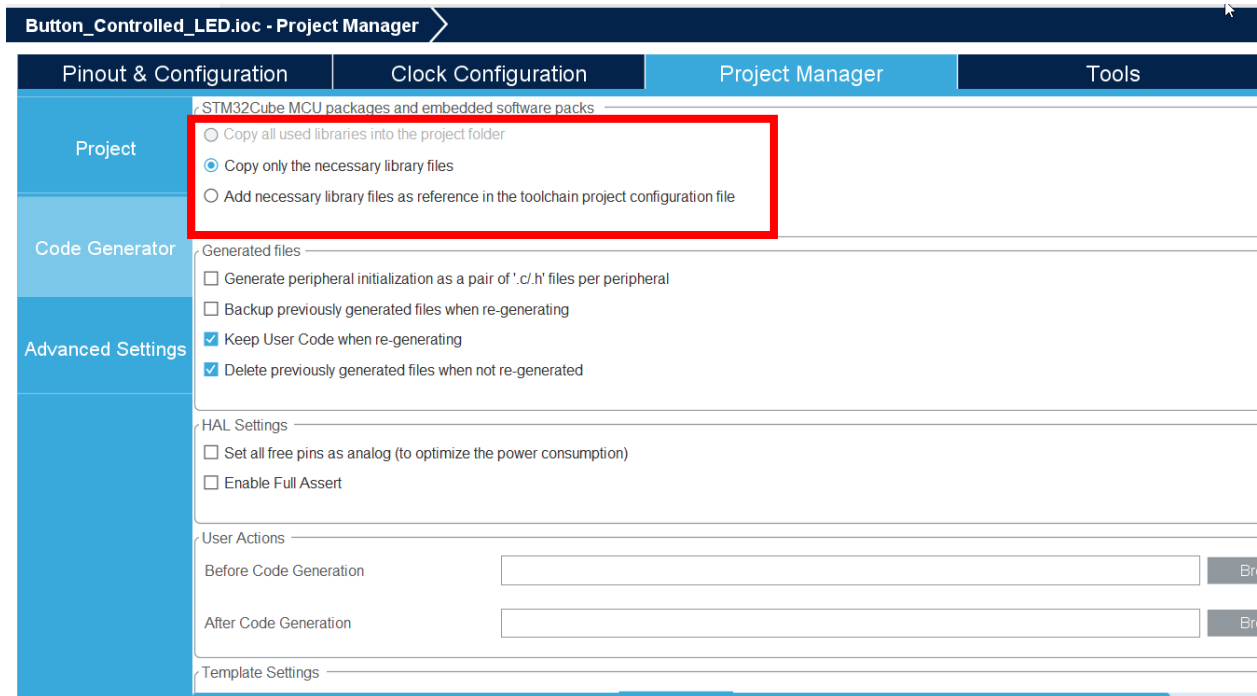
سپس وارد بخش Project Manager شده و تنظیمات اولیه را در صورت نیاز به دلخواه تغییر دهید.

int.ioc - Project Manager

Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project	Project Settings Project Name int Project Location C:\Users\W.Razeghi\STM32CubeIDE\workspace_1.19.0 Browse		
Code Generator	Application Structure Advanced ☐ Do not generate the main() Toolchain Folder Location C:\Users\W.Razeghi\STM32CubeIDE\workspace_1.19.0\int\ Toolchain / IDE STM32CubeIDE ☒ Generate Under Root		
Advanced Settings	Linker Settings Minimum Heap Size 0x200 Minimum Stack Size 0x400 Thread-safe Settings Cortex-M3NS ☐ Enable multi-threaded support Thread-safe Locking Strategy Default – Mapping suitable strategy depending on RTOS selection. Mcu and Firmware Package Mcu Reference STM32F103C4Tx Firmware Package Name and Version STM32Cube FW_F1 V1.8.6 ☒ Use latest available version		

پیکربندی اولیه مسئله در نرم افزار STM32CubeIDE

وارد بخش Code Generator شده و بخش STM32Cube MCU package and embedded software pack را بر روی گزینه Copy only necessary library files تنظیم کنید. تا نرم افزار STM32CubeMX تنها کتابخانه های مورد نیاز بر نامه را به پوشه پروژه اضافه کند. این عمل باعث می شود تا حجم پروژه بسیار کاهش پیدا کند. سپس بر روی Ctrl + S کلیک کنید تا پس از مدت کوتاهی پروژه ساخته شود.



شروع کد نویسی در

نرم افزار STM32CubeIDE

شروع کد نویسی در نرم افزار STM32CubeIDE

پس از تولید کد در مرحله قبل، حال با مراجعه به منوی سمت چپ صفحه از زیر شاخه پروژه ایجاد شده بر روی Core بعد Src و بعد main.c کلیک کنید تا فایل اصلی کد باز شود.

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure. The 'int' folder is selected, and the 'Src' folder is expanded, showing the 'main.c' file. On the right, the main.c file is open, showing the following code:

```

1  /* USER CODE BEGIN Header */
2  /**
3   *
4   * @file      : main.c
5   * @brief     : Main program body
6   *
7   * @attention
8   *
9   * Copyright (c) 2025 STMicroelectronics.
10  * All rights reserved.
11  *
12  * This software is licensed under terms that can be found in the LICENSE file
13  * in the root directory of this software component.
14  * If no LICENSE file comes with this software, it is provided AS-IS.
15  *
16  */
17 /* USER CODE END Header */
18 /* Includes -----*/
19 #include "main.h"
20
21
22 /* Private includes -----*/
23 /* USER CODE BEGIN Includes */
24
25 /* USER CODE END Includes */

```

شروع کد نویسی در نرم افزار STM32CubeIDE

در STM32 وقتی وقفه خارجی (EXTI) از پایه‌ای مثل PA0 یا هر پایه‌ی دیگری رخ می‌دهد (مثلاً با فشردن دکمه)، در لایه HAL، تابعی به نام **HAL_GPIO_EXTI_Callback()** فراخوانی می‌شود. این تابع معمولاً در مسیر زیر قابل دسترسی می‌باشد:

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure. The 'Drivers' folder is expanded, and the 'STM32F1xx_HAL_Driver' folder is selected. The 'stm32f1xx_hal_gpio.c' file is highlighted. On the right, the Code Editor shows the implementation of the **HAL_GPIO_EXTI_Callback** function. The function is defined as a weak function, allowing it to be overridden in a user file. The code includes comments in Persian and C-style comments.

```

540
541 /**
542  * @brief This function handles EXTI interrupt request.
543  * @param GPIO_Pin: Specifies the pins connected EXTI line
544  * @retval None
545  */
546 void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin)
547 {
548     /* EXTI line interrupt detected */
549     if (__HAL_GPIO_EXTI_GET_IT(GPIO_Pin) != 0x00u)
550     {
551         __HAL_GPIO_EXTI_CLEAR_IT(GPIO_Pin);
552         HAL_GPIO_EXTI_Callback(GPIO_Pin);
553     }
554 }
555
556 /**
557  * @brief EXTI line detection callbacks.
558  * @param GPIO_Pin: Specifies the pins connected EXTI line
559  * @retval None
560  */
561 weak void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
562 {
563     /* Prevent unused argument(s) compilation warning */
564     UNUSED(GPIO_Pin);
565     /* NOTE: This function Should not be modified, when the callback is needed,
566      the HAL_GPIO_EXTI_Callback could be implemented in the user file
567     */
568 }

```

شروع کد نویسی در نرم افزار STM32CubeIDE

✓ کلیدواژه **__weak** یعنی:

اگر شما در پروژه خود (main.c)، تابعی با همین نام بنویسید، تابع شما جایگزین این تابع در کتابخانه اصلی می‌شود.

✓ لذا برای جلوگیری از این اتفاق و فقط فراخوانی تابع، عبارت **__weak** را حذف کرده و از باقی تابع به صورت زیر استفاده نمایید.

✓ همچنین توجه شود که این تابع در User Code شماره 4 نوشته شود.

```
/* USER CODE BEGIN 4 */
```

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
```

```
{
```

```
/* Prevent unused argument(s) compilation warning */
```

```
UNUSED(GPIO_Pin);
```

```
/* NOTE: This function should not be modified, when the callback is needed,  
the HAL_GPIO_EXTI_Callback could be implemented in the user file
```

```
*/
```

```
}
```

```
/* USER CODE END 4 */
```



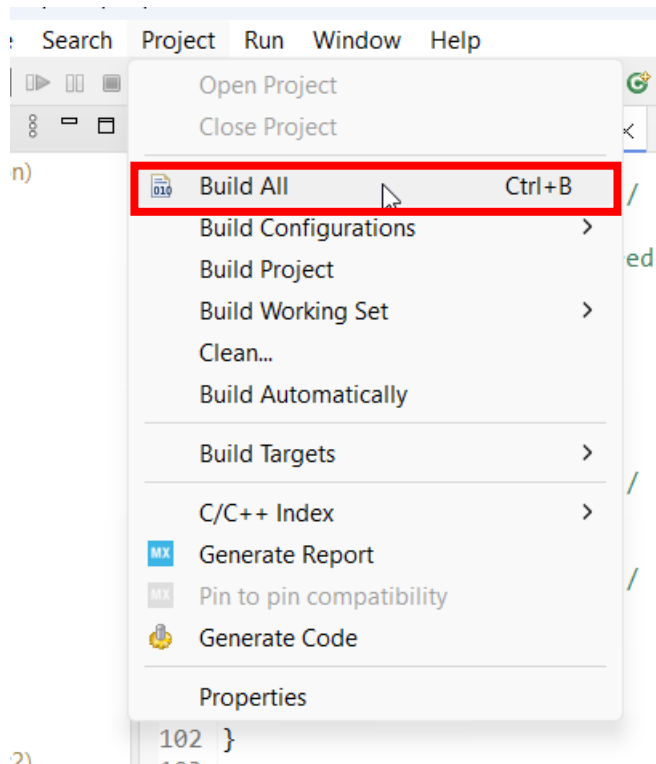

شروع کد نویسی در نرم افزار STM32CubeIDE

✓ حال برای آنکه هر بار کلید فشرده شده و یک وقفه رخ داد، وضعیت ال ای دی Toggle شود، باید تابع Callback را مطابق زیر ویرایش نماییم:

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == GPIO_PIN_0)
    {
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_13);
    }
}
```

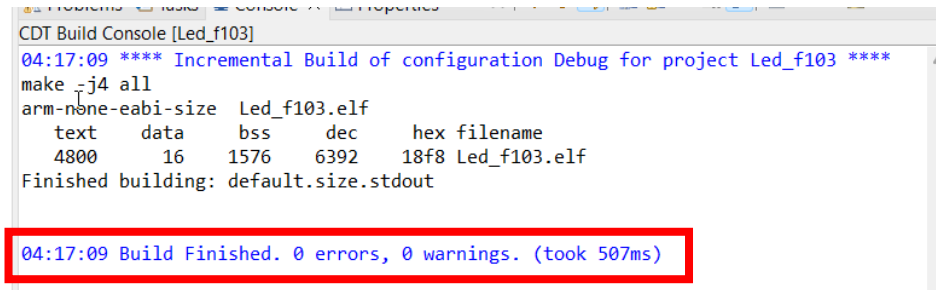


شروع کد نویسی در نرم افزار STM32CubeIDE



✓ نهایتاً پس از نوشتن کد در ادیتور CubeIDE از منوی project بر روی Build All کلیک کنید تا پروژه کامپایل شده و فایل‌های مورد نیاز در خروجی ایجاد شود.

✓ دقت شود که در انتهای کامپایل کد دارای error یا warning نباشد.





razeghizade@gmail.com

Razeghizade.pudica.ir

CREADITS: This presentation was created by M.Razeghizadeh
Please keep this slide for attribution