

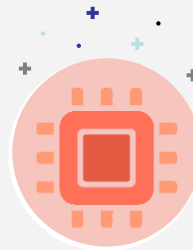


In The Name of God

Faculty of Electrical Engineering
Microprocessor Laboratory



V1 (@2025)



Microprocessor LAB

Lecture 2:

GPIO Peripherals

By: M.Razeghizadeh

Start now!

• به طور کلی می‌توان گفت هر میکروکنترلر از چند بخش اصلی تشکیل شده است:

✓ هسته‌ی پردازنده (CPU):

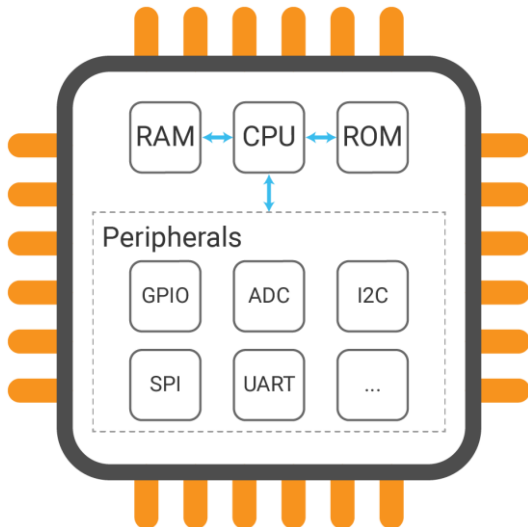
مسئول اجرای دستورات و پردازش داده‌هاست.

✓ حافظه‌ها (Memory):

شامل Flash برای ذخیره‌ی برنامه، SRAM برای داده‌ها و گاهی EEPROM برای داده‌های ماندگار.

✓ پریفرال‌ها (Peripherals):

واحدهای جانبی برای انجام وظایف خاص مانند تایمر، ورودی/خروجی، مبدل آنالوگ به دیجیتال و ...

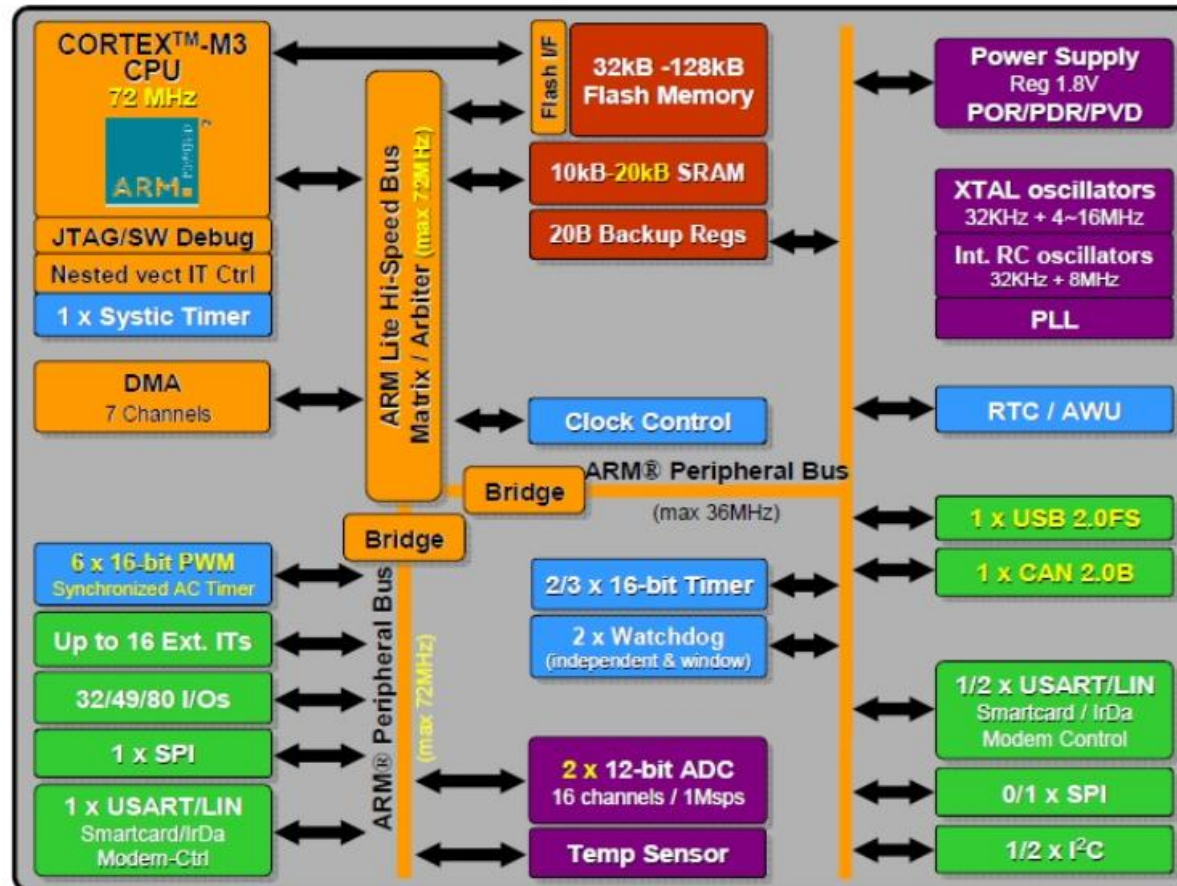


• **پریفرال (Peripheral)** به هر واحد سخت‌افزاری جانبی گفته می‌شود که

داخل تراشه‌ی میکروکنترلر قرار دارد و برای انجام یک وظیفه‌ی خاص طراحی شده است.

• CPU فقط محاسبات و اجرای دستورات را برعهده دارد، اما پریفرال‌ها امکانات ورودی/خروجی، ارتباطی و کنترلی را فراهم می‌کنند تا میکروکنترلر بتواند با محیط واقعی کار کند.

بلوک دیاگرام STM32F103



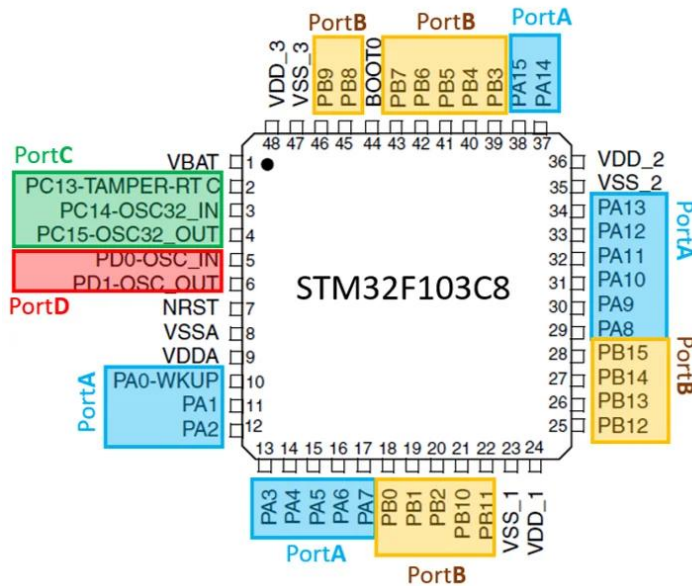
GPIO

- پریفرال‌های (General Purpose Input/Output) GPIO همانند سایر پریفرال‌های میکروکنترلر، مدارهای الکترونیکی مستقلی هستند که عملکرد آن‌ها از طریق مقادیری که در رجیسترهای مربوط به آن‌ها نوشته می‌شود، تنظیم می‌گردد. این رجیسترها در واقع خانه‌هایی از حافظه هستند که با تغییر محتوای آن‌ها می‌توان رفتار پریفرال را کنترل کرد.

- از پریفرال GPIO برای **تعیین جهت پایه‌ها (ورودی یا خروجی)**، **خواندن وضعیت پایه‌های ورودی و تعیین مقدار منطقی پایه‌های خروجی** استفاده می‌شود.

- پریفرال GPIO وظیفه‌ی مدیریت پایه‌های ورودی و خروجی عمومی میکروکنترلر را بر عهده دارد. هر پریفرال GPIO از چند پورت (Port) تشکیل شده است و هر پورت می‌تواند **حداکثر ۱۶ پایه (Pin 0 تا Pin 15)** را کنترل کند.

- در میکروکنترلر STM32F103C8 با بسته‌بندی ۴۸ پایه‌ای، چهار پورت به نام‌های Port A، Port B، Port C و Port D وجود دارد که هر یک توسط پریفرال GPIO مدیریت می‌شوند.





GPIO

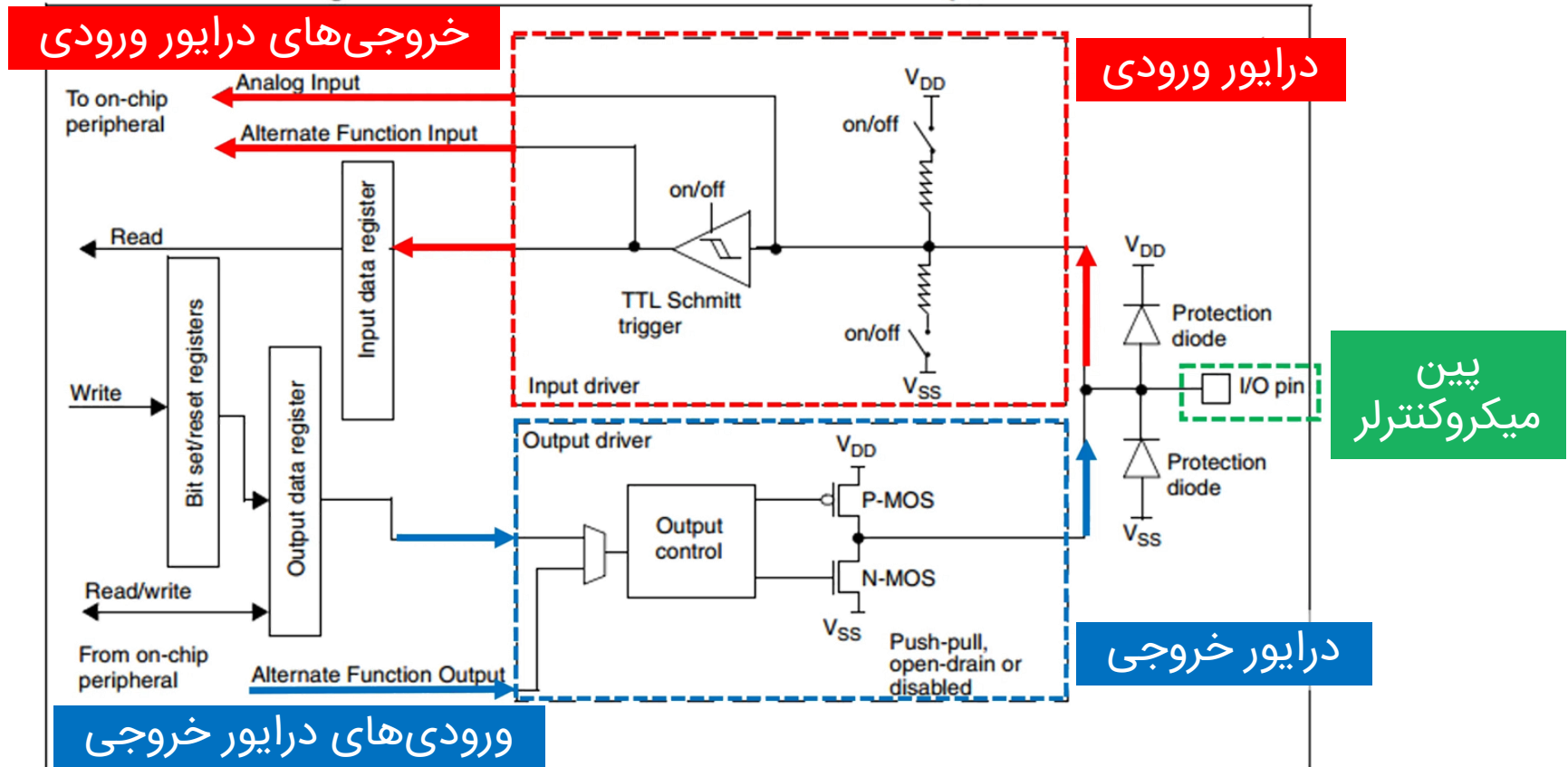
- هر پین GPIO می‌تواند در یکی از ۱۶ حالت کاری زیر تنظیم شود:

نوع عملکرد	حالت کاری	توضیح کوتاه
ورودی	Floating	بدون مقاومت داخلی
	Analog	برای ADC
	Pull-up	مقاومت داخلی به VCC
	Pull-down	مقاومت داخلی به GND
خروجی General Purpose	GP Push-Pull (2/10/50MHz)	خروجی عادی، توان بالا
	GP Open-Drain (2/10/50MHz)	نیاز به مقاومت pull-up خارجی
خروجی Alternate Function	AF Push-Pull (2/10/50MHz)	برای عملکردهایی مثل UART، PWM، SPI
	AF Open-Drain (2/10/50MHz)	عملکرد خاص با درایو باز

ساختار داخلی هر پین در STM32

- در پشت هر پین میکروکنترلر (که بخشی از یک پورت GPIO است) یک مدار کنترلی وجود دارد که تعیین می‌کند پین در حالت ورودی یا خروجی چگونه عمل کند.
- در STM32F103C8، از میان ۴۸ پین، ۳۷ پین دارای این ساختار هستند.

Figure 13. Basic structure of a standard I/O port bit





ساختار داخلی هر پین در STM32

حالت ورودی (Input Mode):

در این حالت از درایور ورودی استفاده می‌شود و عملکرد پین می‌تواند یکی از این سه نوع باشد:

- ورودی آنالوگ (برای ADC و مدارهای آنالوگ)
- ورودی دیجیتال معمولی (General Purpose < مقدار در رجیستر IDR ذخیره می‌شود)
- ورودی تابع جایگزین (Alternate Function < سیگنال به پریفرال‌هایی مثل USART یا SPI می‌رود)

حالت خروجی (Output Mode):

در این حالت از درایور خروجی استفاده می‌شود و عملکرد پین می‌تواند یکی از دو نوع باشد:

- خروجی دیجیتال معمولی (General Purpose < مقدار از رجیستر ODR تعیین می‌شود)
- خروجی تابع جایگزین (Alternate Function < کنترل توسط پریفرال‌های داخلی انجام می‌شود)

نکته: Alternate Function یعنی پین در حالت دیجیتال است اما مستقیماً توسط یک پریفرال مثلاً UART، SPI یا

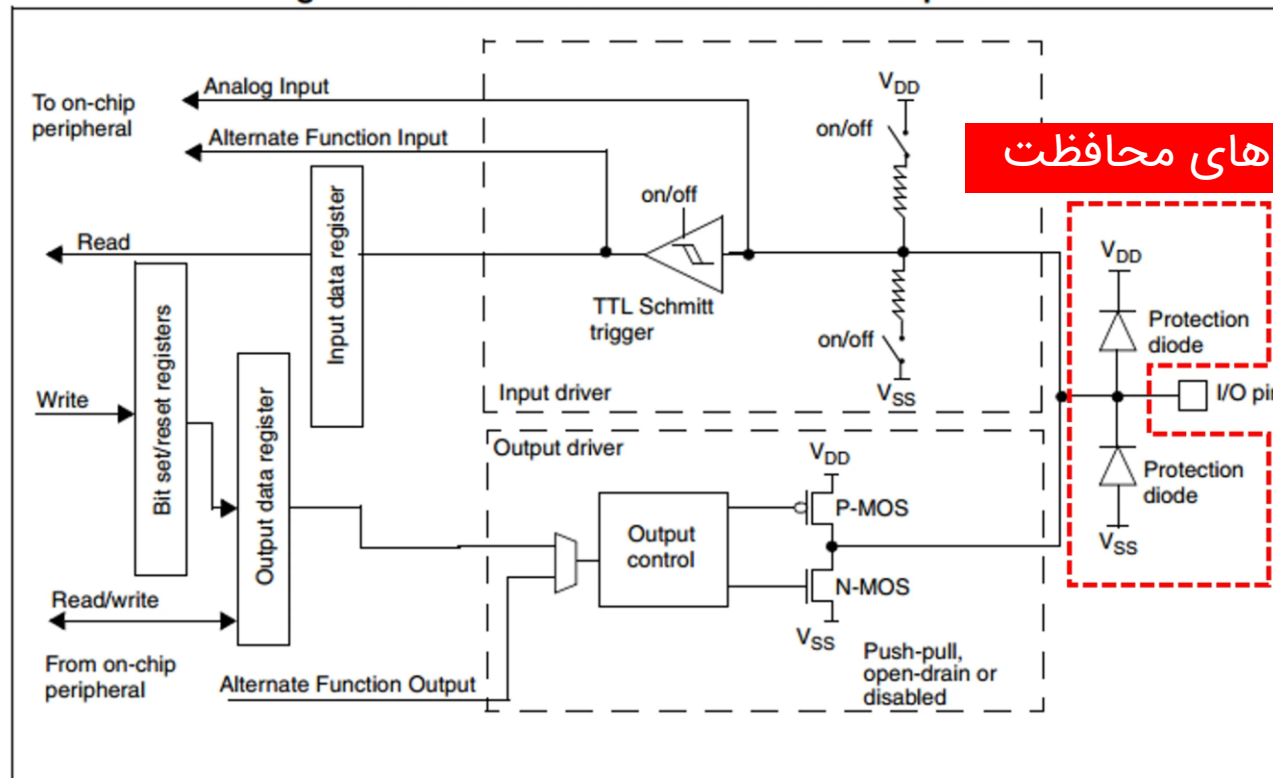
TIMER و ... کنترل می‌شود.

دیودهای محافظت در پرفرال GPIO

همان طور که در بلوک دیاگرام پرفرال GPIO مشاهده می شود، پشت هر پین دو دیود محافظت قرار دارد:

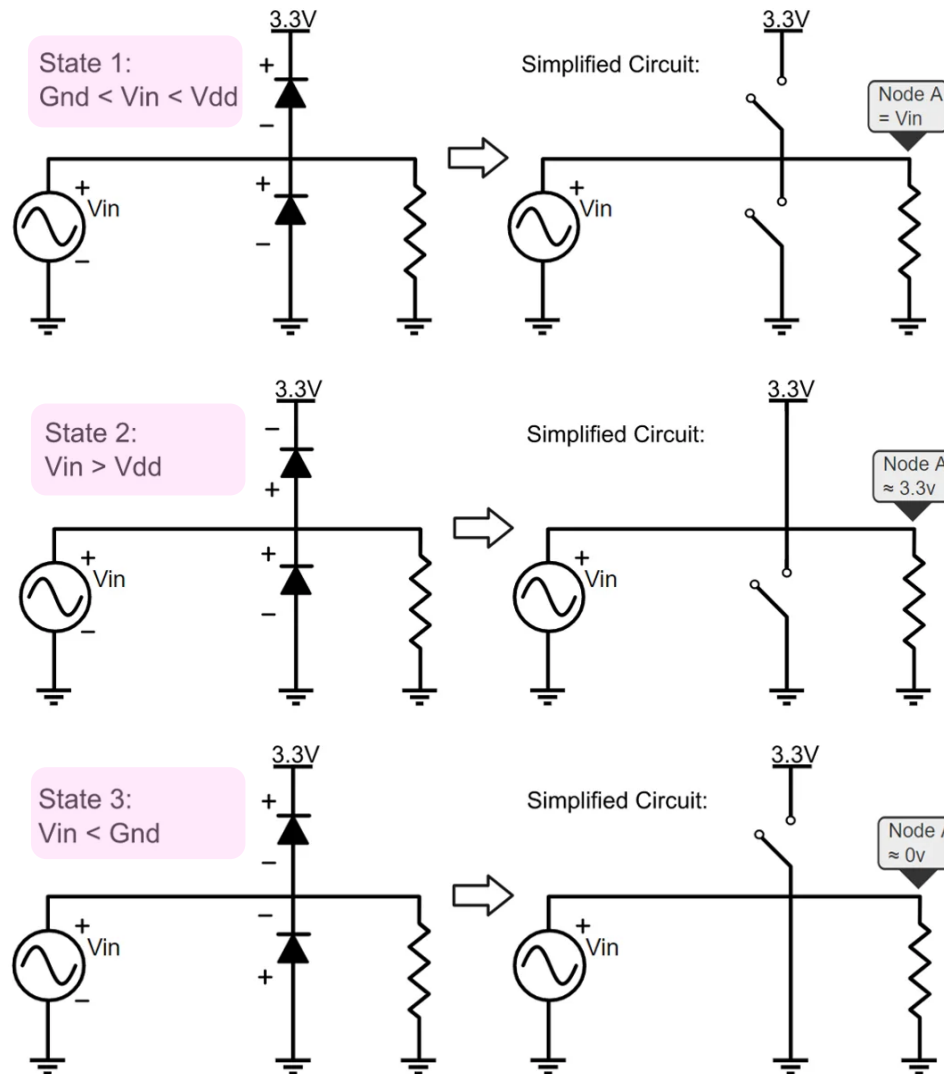
- یک دیود، پین را به زمین (GND) متصل می کند. <<<<<< اجازه نمی دهد ولتاژ روی پین از $GND - 0.7$ ولت کمتر شود.
- یک دیود، پین را به تغذیه (VDD) متصل می کند. <<<<<< اجازه نمی دهد ولتاژ روی پین از $VDD + 0.7$ ولت بیشتر شود.

Figure 13. Basic structure of a standard I/O port bit





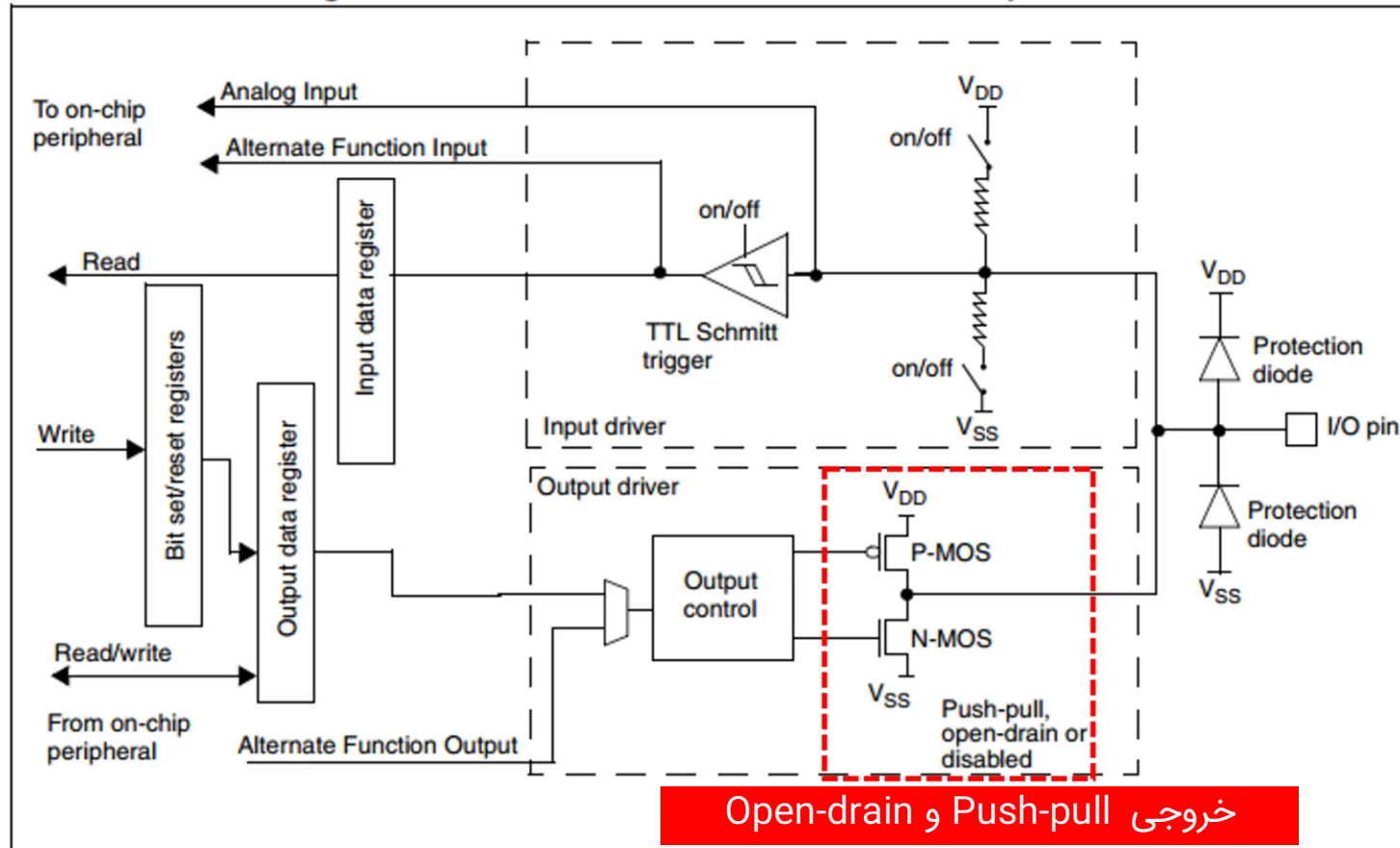
دیودهای محافظت در پرفرال GPIO



دراپور خروجی در پرفرال GPIO

- پین خروجی میکروکنترلر میتواند در حالت Push-pull یا Open-drain تنظیم شود.

Figure 13. Basic structure of a standard I/O port bit



دراپور خروجی در پرفرال GPIO: خروجی Open-Drain

در این حالت، فقط ترانزیستور پایین (اتصال به GND) وجود دارد و مسیر بالا (به VDD) باز است.

برای داشتن سطح منطقی ۱، باید از مقاومت Pull-up استفاده شود.

✓ ویژگی‌ها:

✓ خروجی فقط می‌تواند ۰ منطقی را فعالانه تولید کند (۱ توسط مقاومت پول‌آپ ایجاد می‌شود).

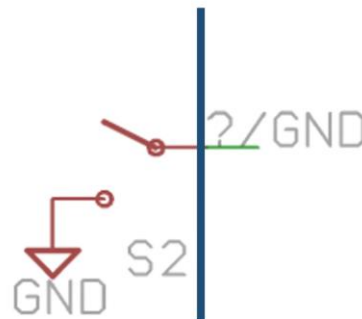
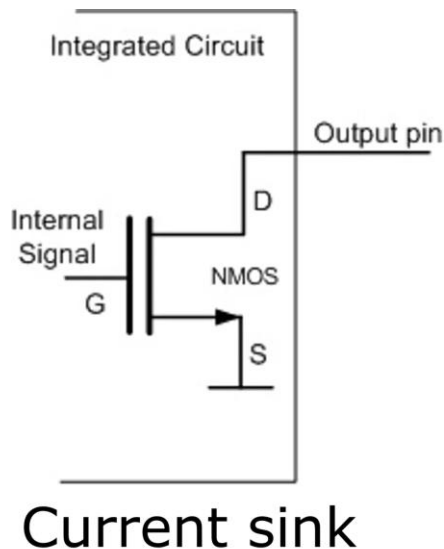
✓ چند پین می‌توانند به صورت اشتراکی (Wired-OR یا Wired-AND) روی یک خط متصل شوند (مثل در I²C).

✓ مناسب برای ارتباط‌های دوطرفه با خطوط مشترک است.

✗ محدودیت:

✓ سرعت سوئیچ کمتر از Push-Pull است.

✓ نیاز به مقاومت Pull-up خارجی یا داخلی دارد.



دراپور خروجی در پرفرمانس GPIO: خروجی Push-Pull

در این حالت، هر پین خروجی به دو ترانزیستور (یکی بالا و یکی پایین) متصل است:

✓ ترانزیستور بالا، پین را به VDD (منطق ۱) متصل می‌کند.

✓ ترانزیستور پایین، پین را به GND (منطق ۰) متصل می‌کند.

✓ ویژگی‌ها:

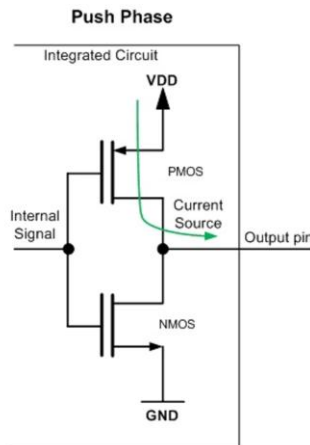
✓ خروجی می‌تواند هم ۰ و هم ۱ منطقی را فعالانه دراپور کند.

✓ سرعت سوئیچ زیاد است (مناسب برای سیگنال‌های سریع مثل SPI).

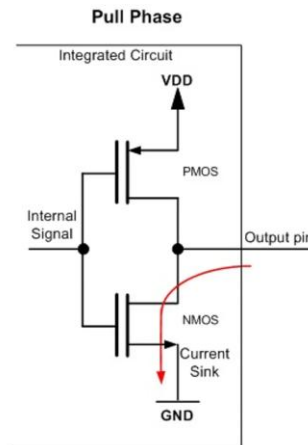
✓ نیازی به مقاومت پول‌آپ ندارد.

✗ محدودیت:

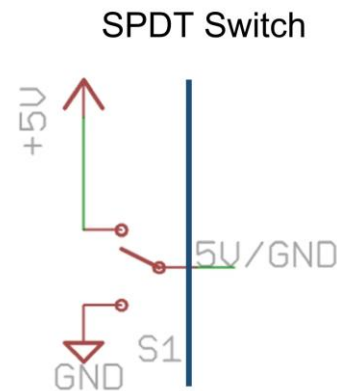
✓ نمی‌توان چند خروجی Push-Pull را مستقیماً به هم متصل کرد (باعث اتصال کوتاه می‌شود).



Current source



Current sink

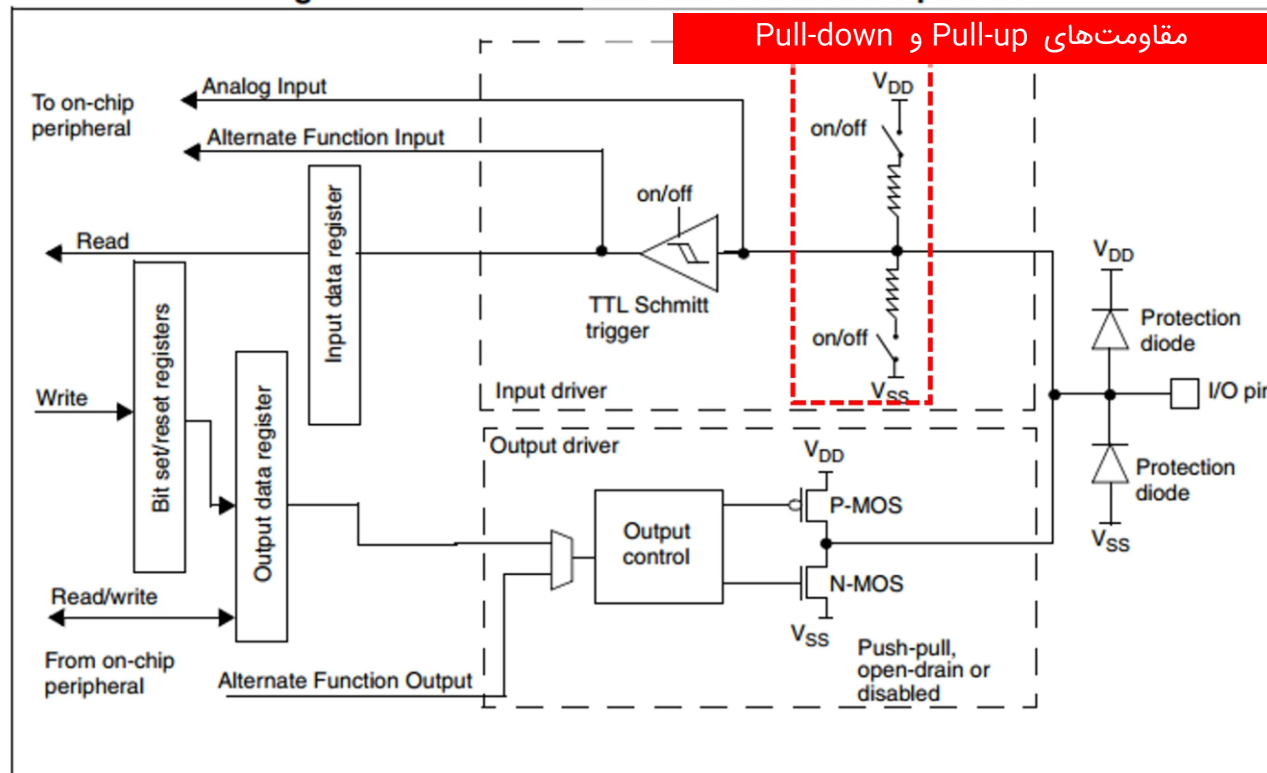


Pull-up / Pull-down مقاومت‌های Floating

در درایور ورودی هر پین، دو مقاومت داخلی Pull-up و Pull-down وجود دارد که می‌توان با تنظیم رجیسترها، یکی از آن‌ها را فعال کرد.

این مقاومت‌ها برای جلوگیری از شناور ماندن پین (Floating) استفاده می‌شوند.

Figure 13. Basic structure of a standard I/O port bit



Pull-up / Pull-down و مقاومت‌های Floating

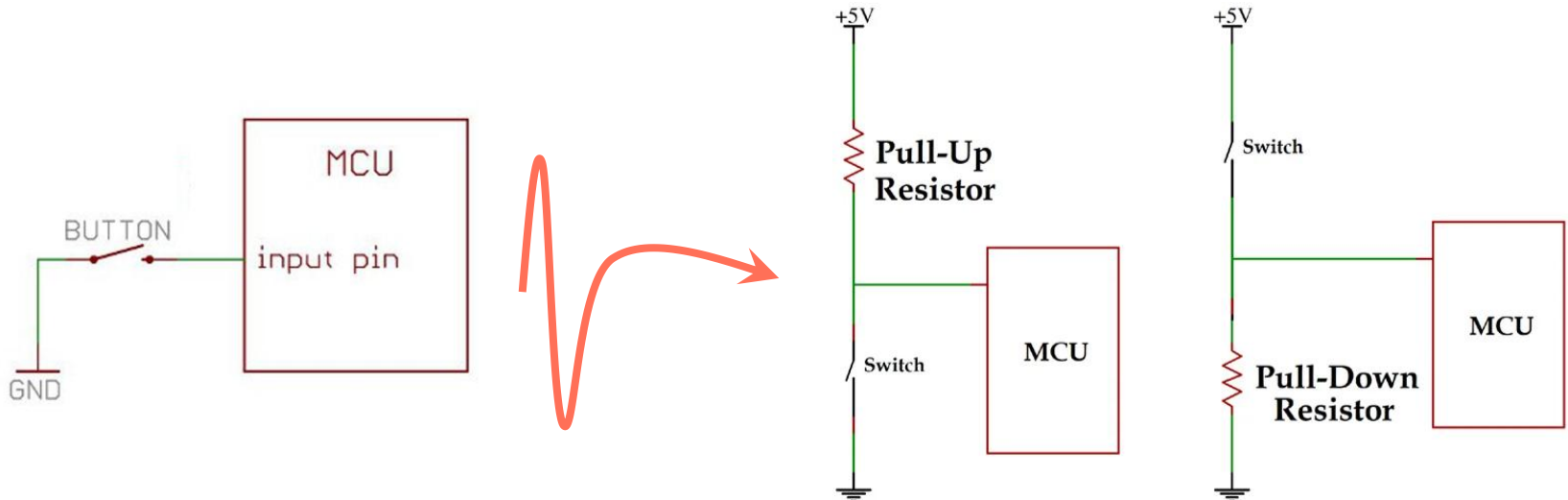
پین Float (شناور)

اگر پین میکروکنترلر در **حالت ورودی** باشد و به هیچ منبعی متصل نباشد (مثلاً قبل از فشردن کلید)، ولتاژ آن نامشخص است و پین در وضعیت شناور (Floating) یا امپدانس بالا (High Impedance) قرار دارد. در این حالت، مقدار خوانده‌شده از رجیستر ورودی ممکن است به صورت تصادفی ۰ یا ۱ باشد.

راه‌حل: (برای جلوگیری از این وضعیت)

با **مقاومت Pull-up**، پین در حالت عادی به VDD وصل می‌شود < مقدار پیش‌فرض ۱ منطقی.

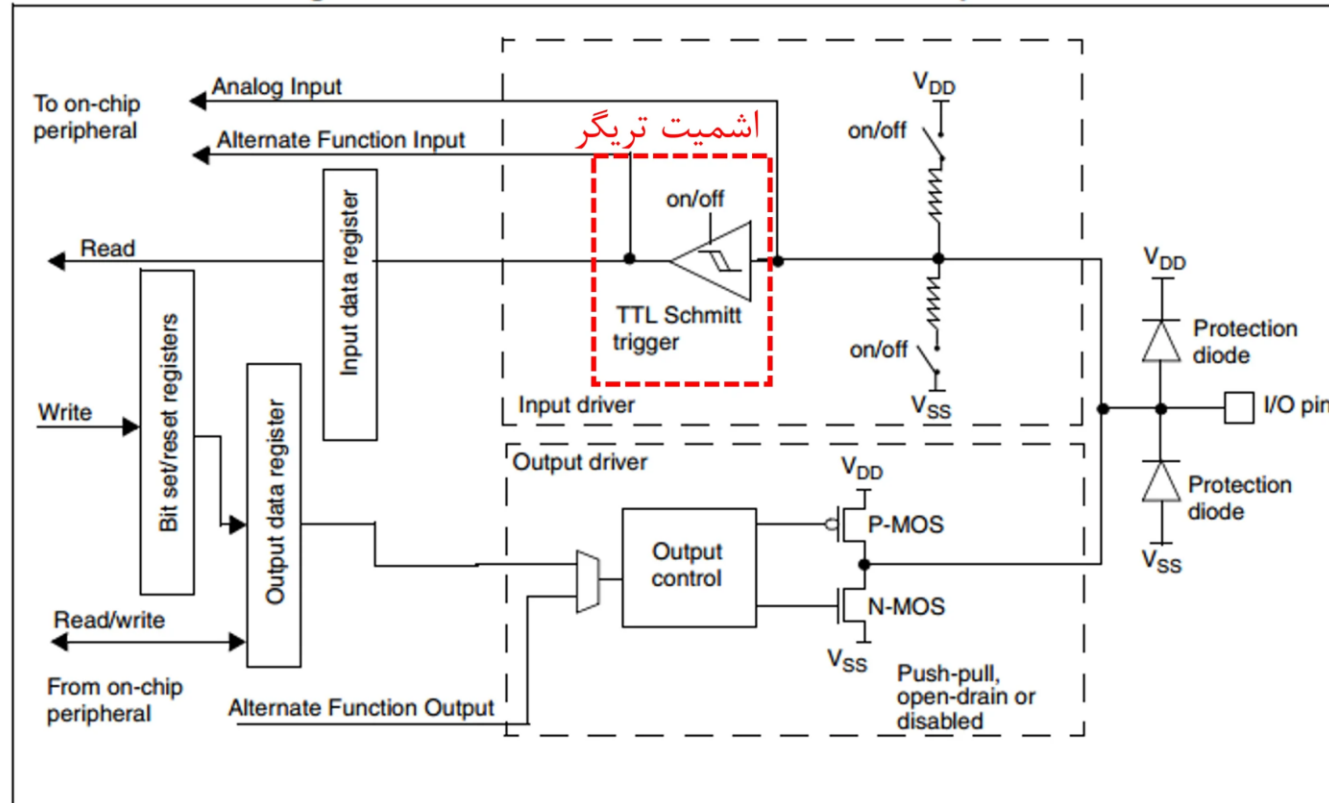
با **مقاومت Pull-down**، پین در حالت عادی به GND وصل می‌شود < مقدار پیش‌فرض ۰ منطقی.



اشمیت تریگر در درایور ورودی پرفیال GPIO

در درایور ورودی پرفیال GPIO، یک مدار اشمیت تریگر (Schmitt Trigger) قرار دارد که دارای یک ورودی و یک خروجی است.

Figure 13. Basic structure of a standard I/O port bit



اشمیت تریگر در درایور ورودی پرفیروال GPIO

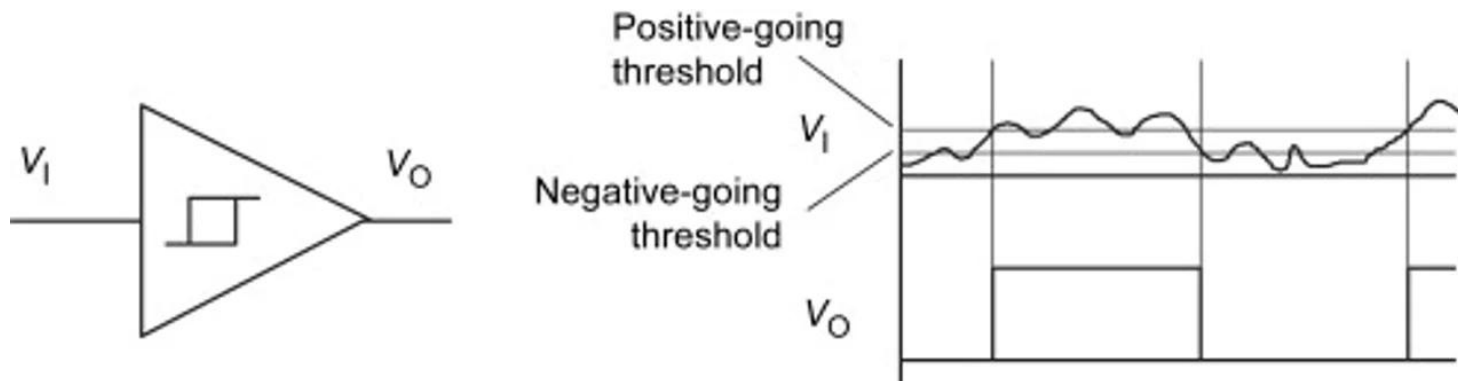
این مدار دو آستانه ولتاژ دارد:

✓ اگر ولتاژ ورودی کمتر از حد پایین باشد < خروجی صفر منطقی (GND) می‌شود.

✓ اگر ولتاژ ورودی بیشتر از حد بالا باشد < خروجی یک منطقی (VDD) می‌شود.

کاربرد:

اشمیت تریگر با ایجاد هیستریزس ولتاژی، از تغییرات ناگهانی ناشی از نویز جلوگیری می‌کند و باعث می‌شود پین ورودی فقط در صورت تغییر واقعی سیگنال، وضعیت خود را عوض کند. این مدار برای نویز گیری از ولتاژ روی پین ورودی هست تا یک ورودی دیجیتال ایده آل داشته باشیم.

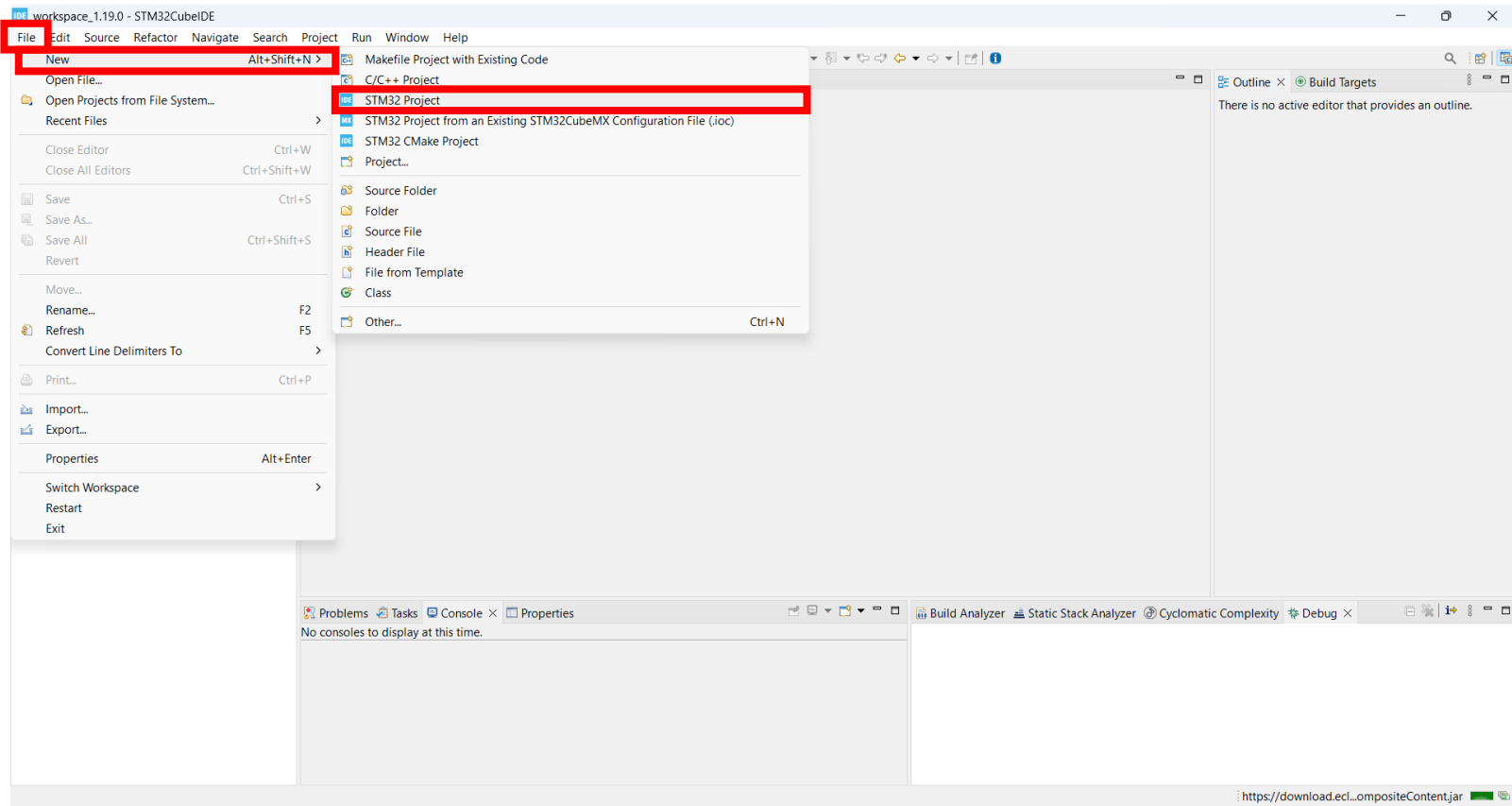


پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE



پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

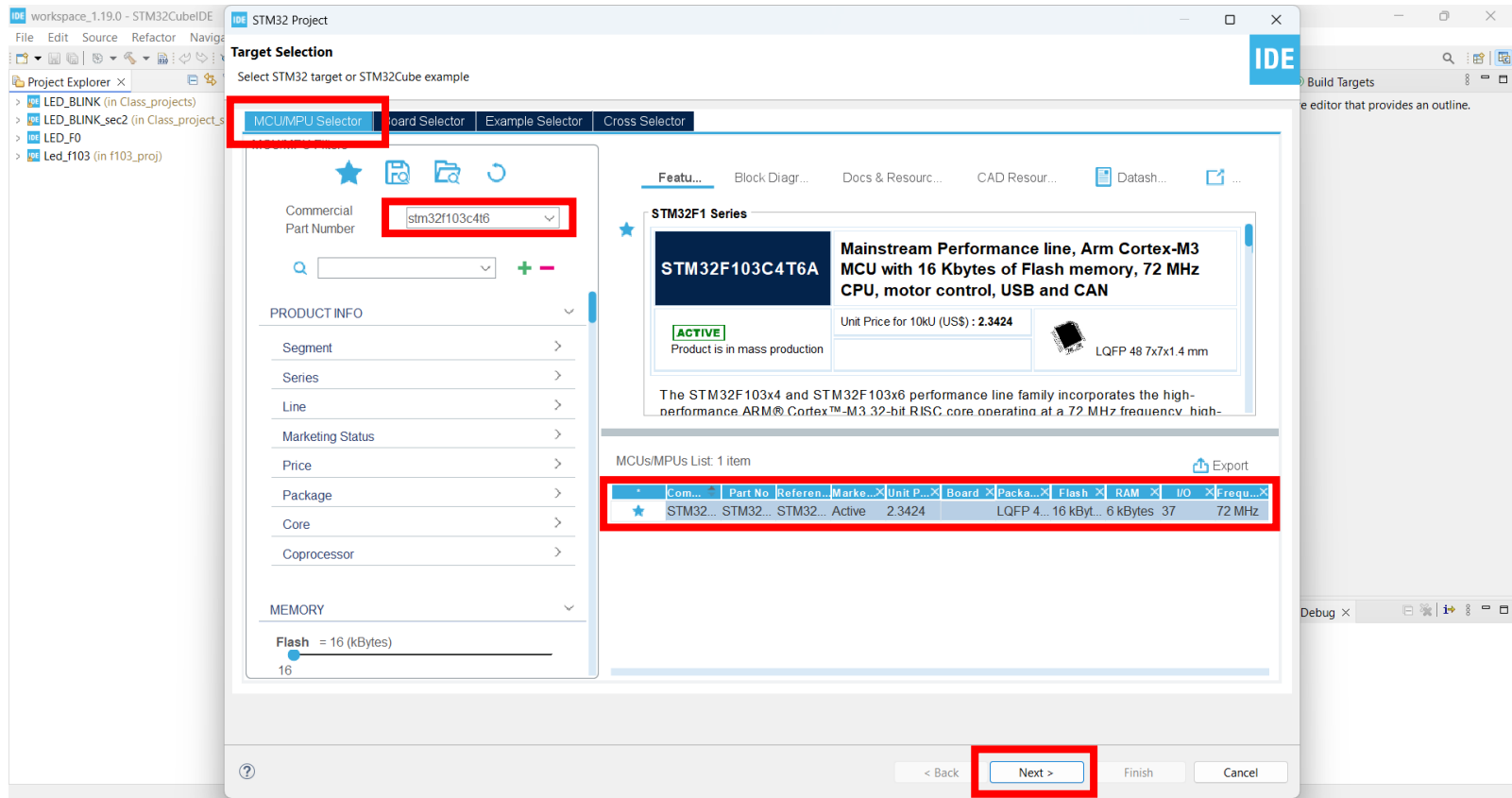
ابتدا در نرم افزار STM32CubeIDE از منو File گزینه New Project را انتخاب کنید.





پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

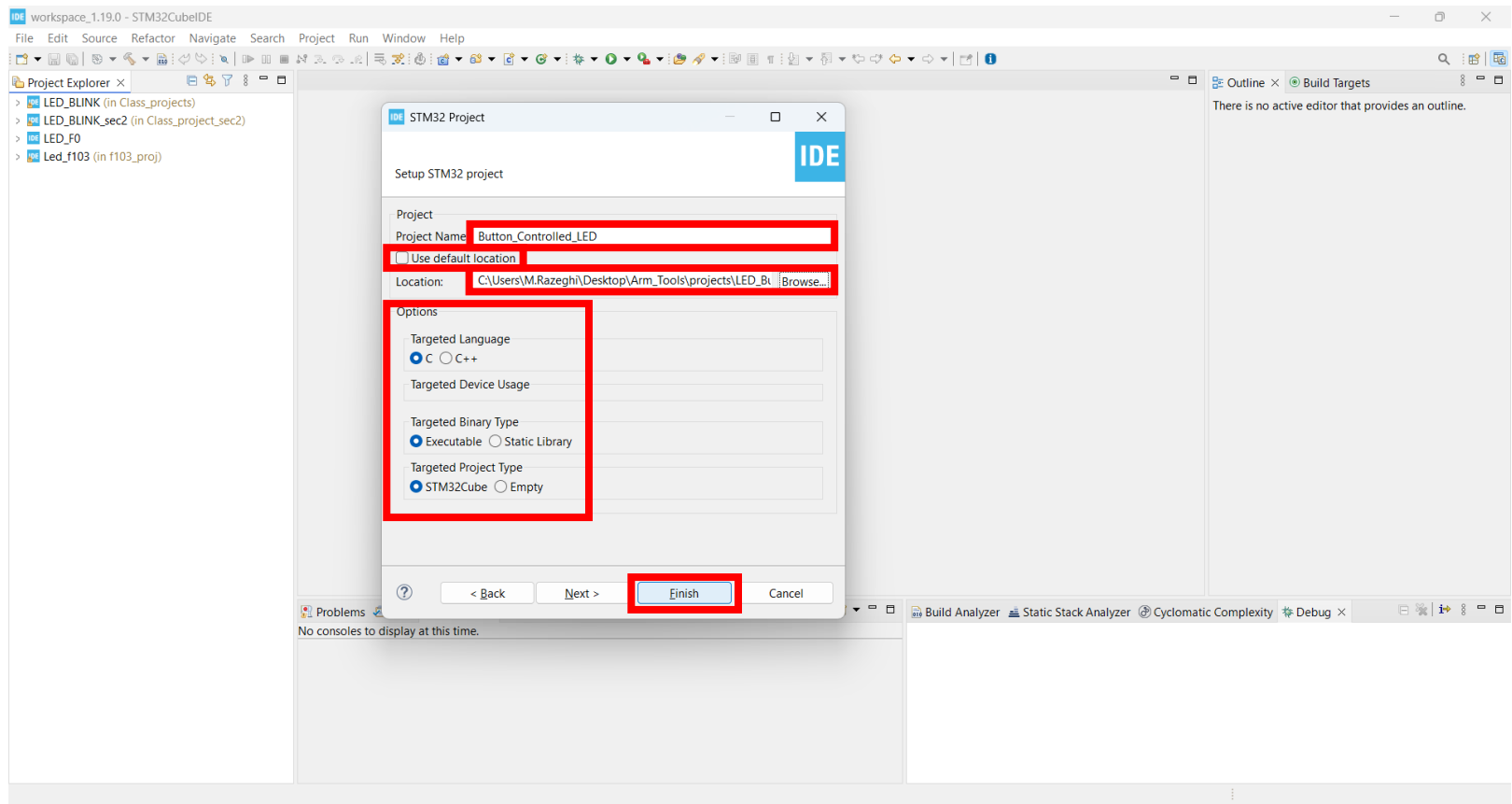
سپس در پنجره باز شده در قسمت Part Number Search نام میکروکنترلر مورد نظر خود را انتخاب کنید (برای مثال STM32F103C4) و بر روی Next کلیک کنید.





پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

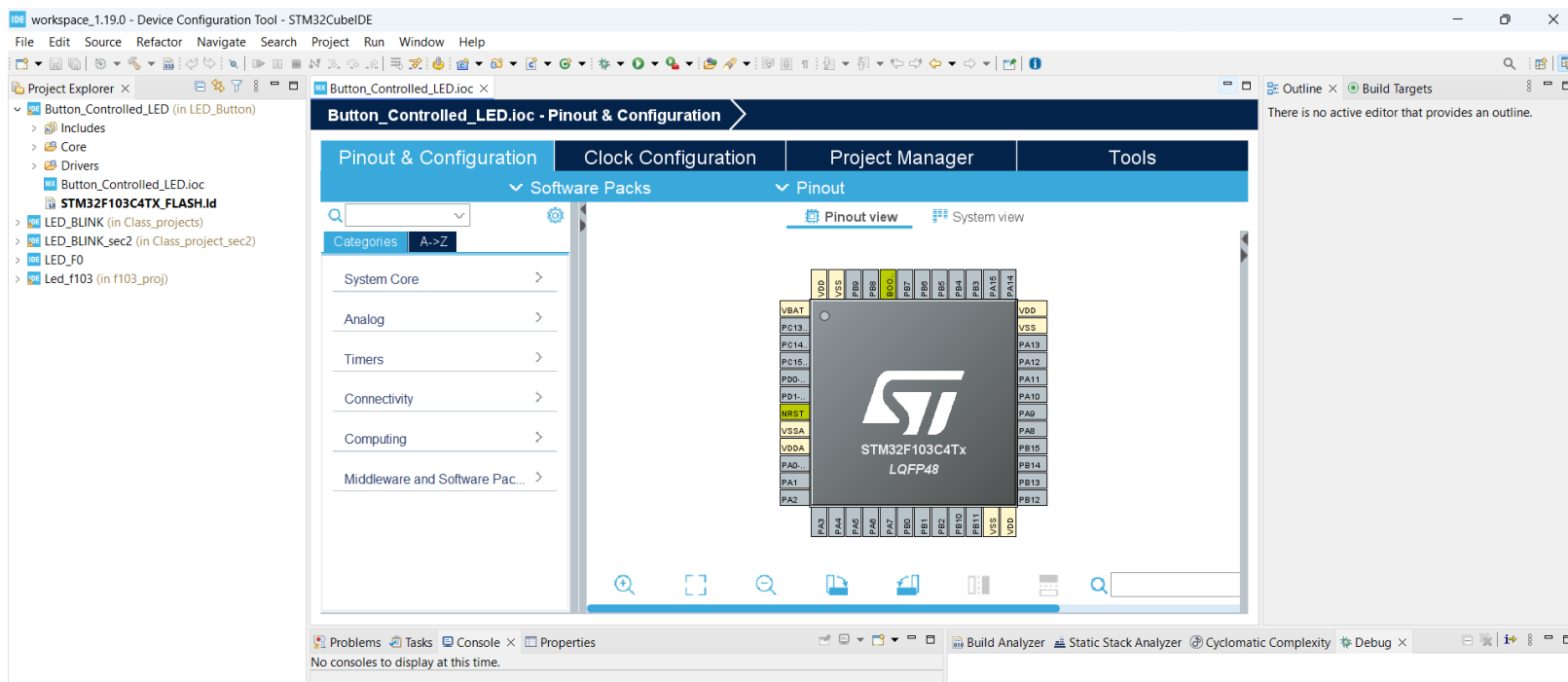
سپس در پنجره باز شده در قسمت Project Name، نام پروژه و در صورت لزوم مکان ذخیره پروژه و تنظیمات مدنظر را مطابق تصویر را وارد کرده و نهایتاً روی Finish کلیک کنید.





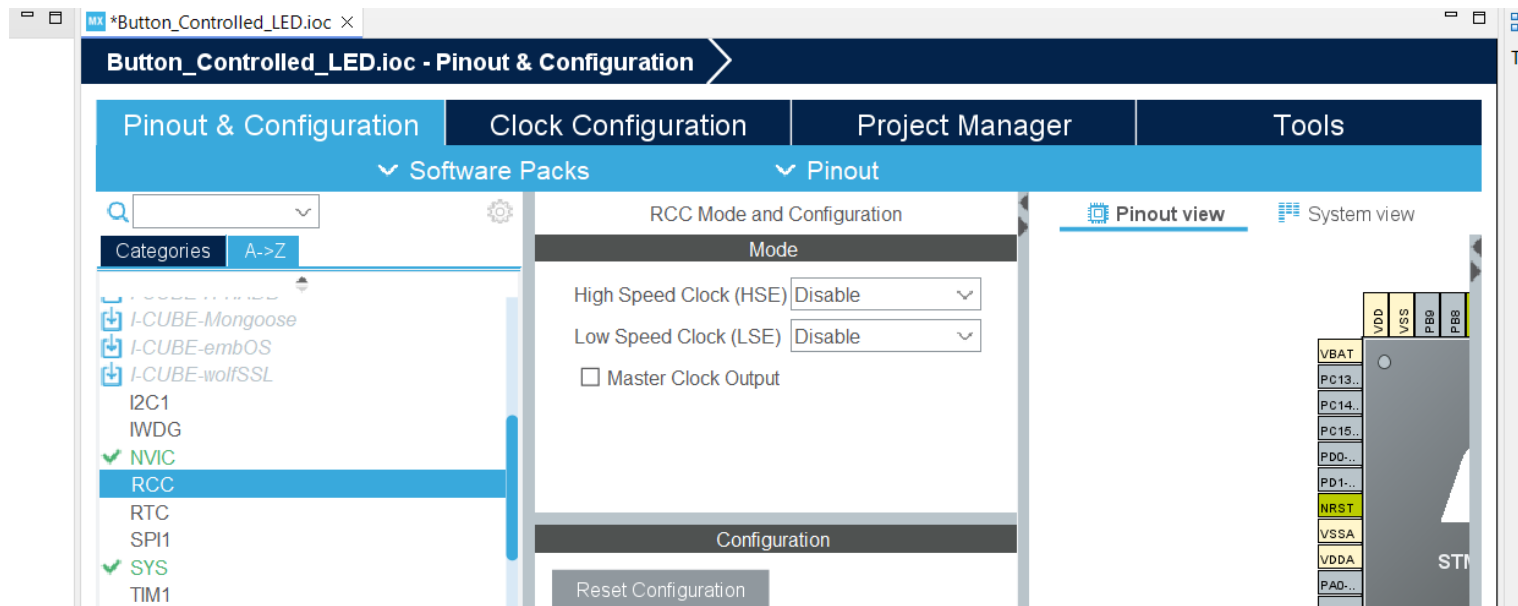
پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

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.



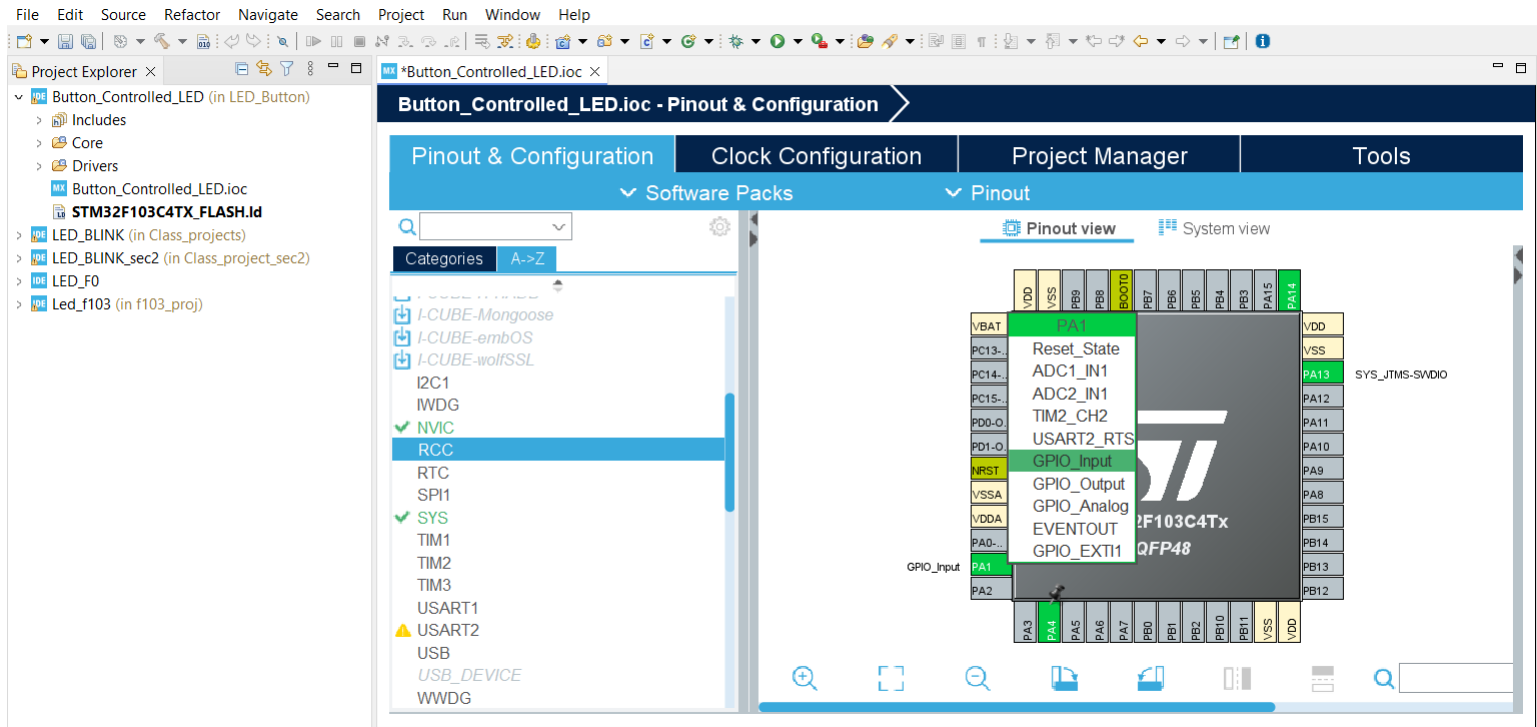
پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

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.

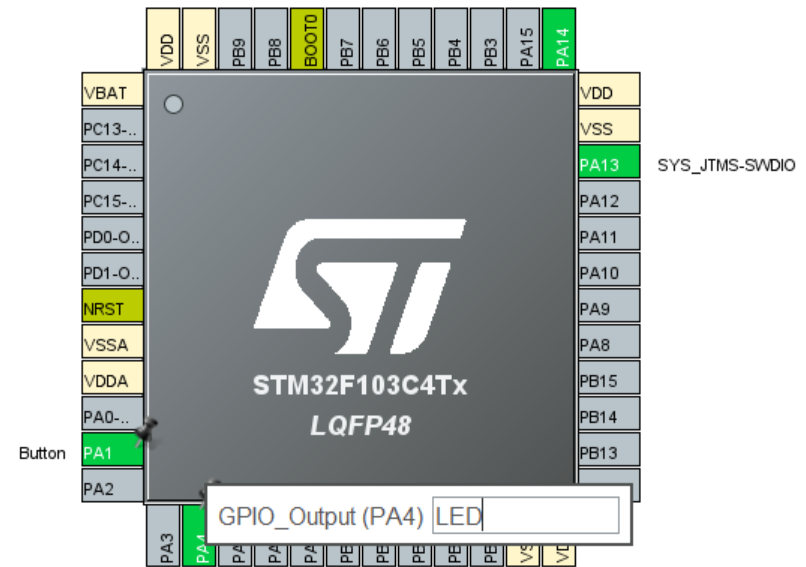


پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

با توجه به میکروکنترلی که انتخاب شده، تعداد مشخصی GPIO در اختیار شما می‌باشد که می‌توانید این GPIOها را به صورت ورودی و یا خروجی تعریف کنید. برای تعریف GPIO بصورت ورودی و یا خروجی بایستی بر روی یکی از پایه های میکروکنترلر کلیک کنید تا گزینه های قابل تنظیم GPIO را مشاهده نمایید. سپس با کلیک بر روی **GPIO_Output** پایه مورد نظر خروجی و با انتخاب **GPIO_Input** پایه بصورت ورودی تعریف می‌شود.



بهرتر است تا برای هر پایه ای که استفاده می‌کنید یک Label انتخاب کنید. برای انتخاب Label بایستی بر روی پایه ای که استفاده نموده اید راست کلیک کنید و گزینه Enter User Label را انتخاب کنید و سپس نام مورد نظر را وارد نمایید و Enter را بزنید.





پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

در این قسمت می‌توانید تنظیمات مربوط به 16 حالت نامبرده برای GPIO را با استفاده از گزینه‌هایی همچون GPIO pull-up/pull-down، Push-pull یا Open-drain، سطح اولیه خروجی و ... را مشخص نمایید.

PA1 Configuration :

GPIO mode: Input mode

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

User Label: Button

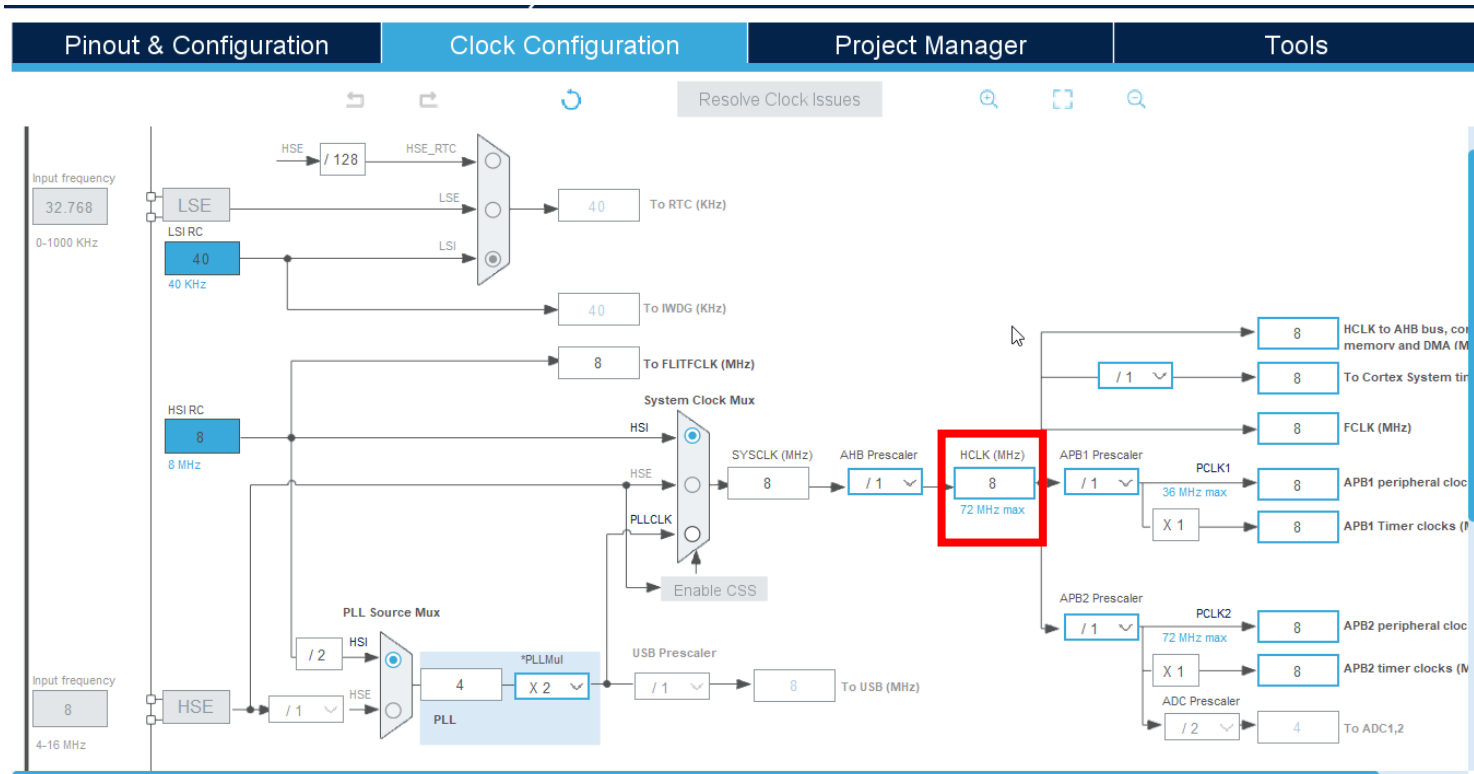
Pin Name	Signal on Pin	GPIO output	GPIO mode	GPIO Pull-up	Maximum ou...	User Label	Modified
PA1	n/a	n/a	Input mode	No pull-up an...	n/a	Button	✓
PA4	n/a	Low	Output Push ...	No pull-up an...	Low	LED	✓

GPIO output level	Low
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low



پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

پس از انجام تنظیمات مورد نظر، وارد بخش Clock Configuration شده تا تنظیمات مربوط به فرکانس کاری هر بخش میکروکنترلر را انجام دهیم. در این آموزش قصد داریم تا Clock میکروکنترلر را بر روی 8 مگاهرتز تنظیم کنیم. بنابراین بر روی بخش HCLK کلیک کرده و عدد 8 را وارد می‌کنیم و سپس Enter را وارد می‌کنیم. نرم افزار بصورت خودکار فرکانس را بر روی 8 مگاهرتز بر مبنای منبع کلاک داخلی میکرو تنظیم می‌کند.





پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

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

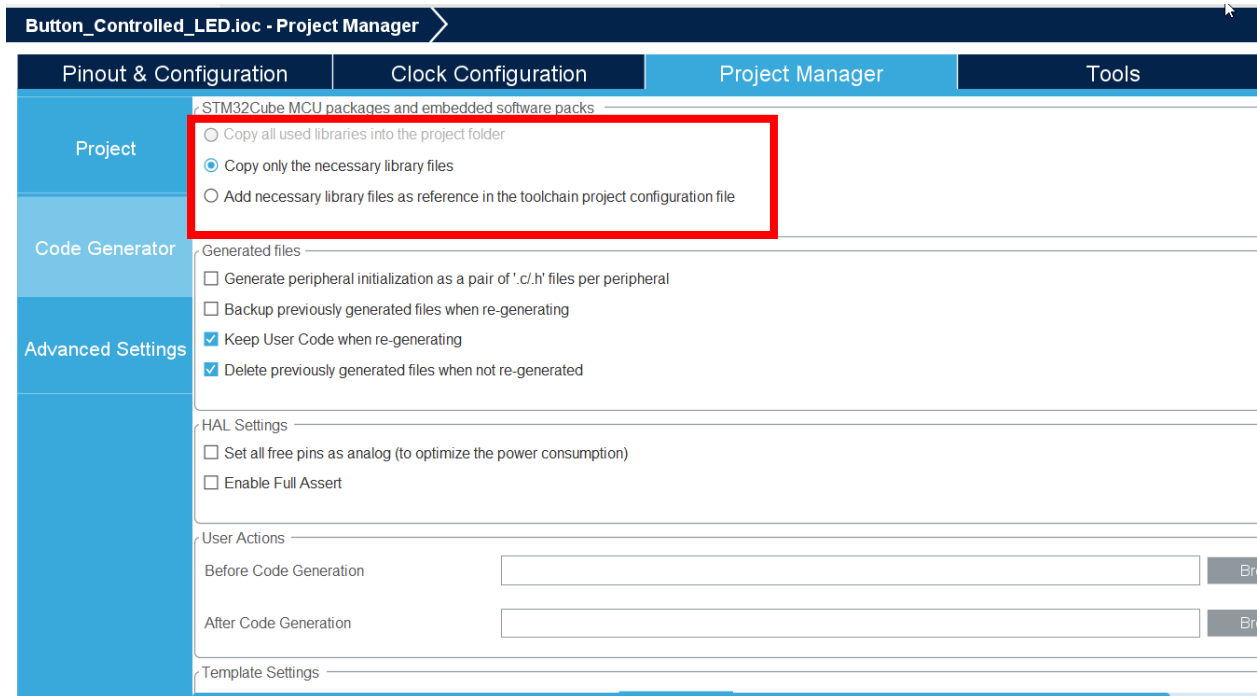
Button_Controlled_LED.ioc - Project Manager

Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project	Project Settings Project Name Button_Controlled_LED Project Location C:\Users\M.Razeghi\Desktop\Arm_Tools\projects\LED_Button Bro...		
Code Generator	Application Structure Advanced Do not generate the mai Toolchain Folder Location C:\Users\M.Razeghi\Desktop\Arm_Tools\projects\LED_Button\ 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 vers		



پیکربندی اولیه GPIO در نرم افزار STM32CubeIDE

وارد بخش Code Generator شده و بخش STM32Cube MCU package and embedded software pack را بروی 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.

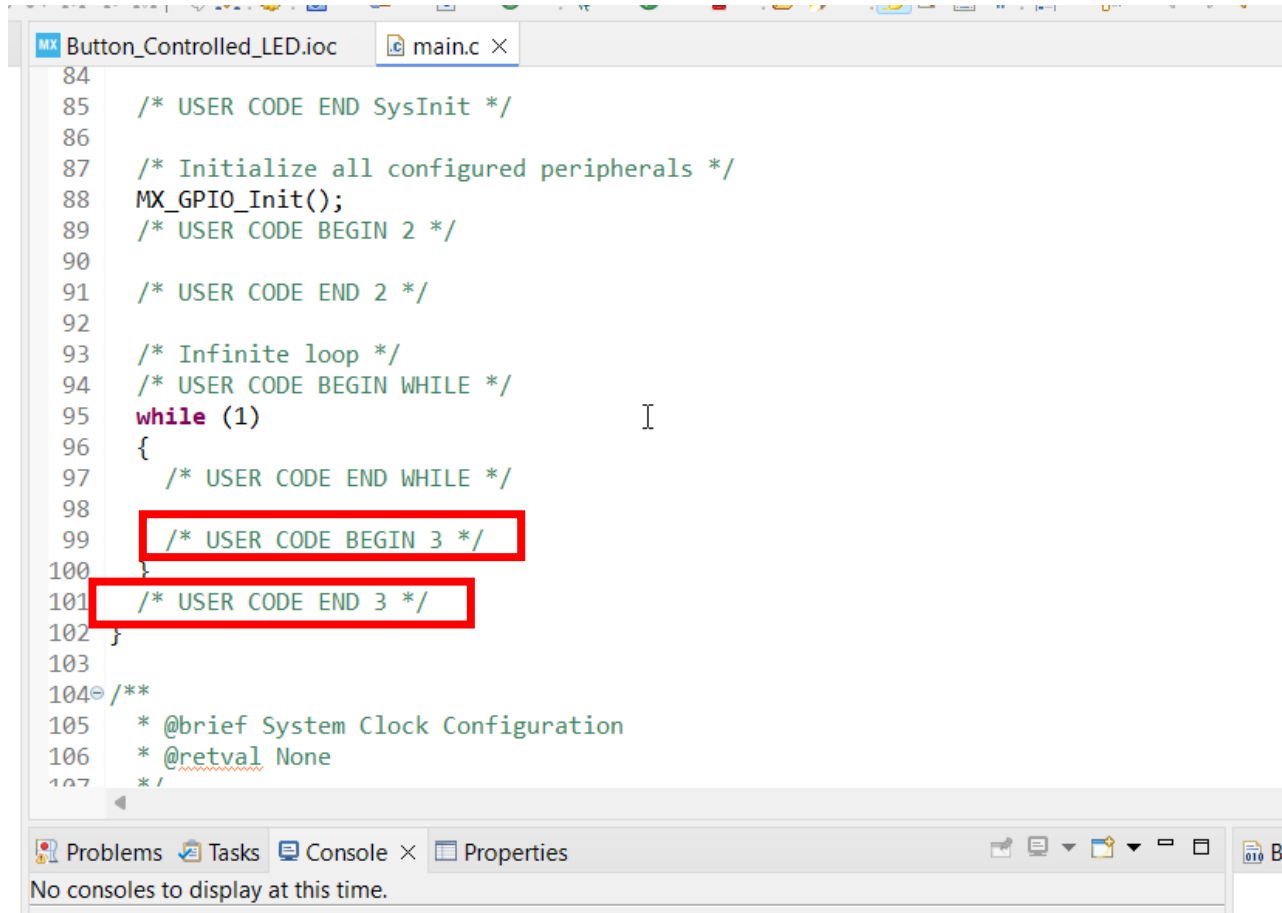
The screenshot displays the STM32CubeIDE workspace. The title bar indicates the project is 'workspace_1.19.0 - Button_Controlled_LED/Core/Src/main.c - STM32CubeIDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Project Explorer on the left shows the project structure with 'Button_Controlled_LED (in LED_Button)' expanded. Under 'Core', the 'Src' folder is expanded, and 'main.c' is selected. The main editor window shows the content of 'main.c', which is a header file for the user code. The code includes a copyright notice for 2025 STMicroelectronics and a license disclaimer. The code is as follows:

```
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 /* Private includes -----*/
22 /* USER CODE BEGIN Includes */
```



شروع کد نویسی در نرم افزار STM32CubeIDE

حال در بین بخش هایی که با USER CODE BEGIN و USER CODE END مشخص شده، کدهای خود را بنویسید.



```
84
85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_GPIO_Init();
89  /* USER CODE BEGIN 2 */
90
91  /* USER CODE END 2 */
92
93  /* Infinite loop */
94  /* USER CODE BEGIN WHILE */
95  while (1)
96  {
97      /* USER CODE END WHILE */
98
99      /* USER CODE BEGIN 3 */
100
101      /* USER CODE END 3 */
102  }
103
104 /**
105  * @brief System Clock Configuration
106  * @retval None
107  */
```

STM32CubeIDE در نرم افزار

توابع مربوط به GPIO عبارتند از:

- 1] `HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);`
- 2] `HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);`
- 3] `HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);`

همچنین تابع مربوط به ایجاد تاخیر عبارت است از:

`HAL_Delay(uint32_t Delay)`

شروع کد نویسی در نرم افزار STM32CubeIDE

```
1] HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
```

- ✓ همانطور که از اسم تابع مشخص می باشد، وظیفه اش خواندن وضعیت یک پین در GPIO هست و با توجه به اینکه مقدار برگشتی از نوع GPIO_PinSate هست خروجی تابع از دو حال خارج نیست: GPIO_PIN_RESET معادل ۰ و GPIO_PIN_SET معادل ۱.
- ✓ آرگومان اول GPIOx نام پورته که قرار است مقدار دهی شود که بسته به میکروپی که استفاده می کنید می تواند از **GPIOA** تا **GPIOG** متغیر باشد.
- ✓ آرگومان دوم شماره پین مربوط به GPIO است. که می تواند مقداری بین **GPIO_PIN_0** الی **GPIO_PIN_15** باشد.



شروع کد نویسی در نرم افزار STM32CubeIDE

```
2] HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);
```

✓ همانطور که از اسم تابع مشخص می‌باشد، وظیفه این تابع مقدار دهی یا نوشتن در GPIO است و چون نوعش void می‌باشد مقداری را بر نمی‌گرداند.

✓ آرگومان اول GPIOx نام پورتهی که قرار است مقدار دهی شود که بسته به میکروبی که استفاده می‌کنید می‌تواند از GPIOA تا GPIOG متغیر باشد.

✓ آرگومان دوم شماره پین مربوط به GPIO است. که می‌تواند مقداری بین GPIO_PIN_0 الی GPIO_PIN_15 و یا GPIO_PIN_ALL (نوشتن بر روی همه پینها بطور همزمان) باشد.

✓ PinState یک نوع شمارشی یا enum است که میتونه مقدار GPIO_PIN_RESET یا صفر و مقدار GPIO_PIN_SET معادل ۱ را داشته باشد.



شروع کد نویسی در نرم افزار STM32CubeIDE

```
3] HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);
```

✓ این تابع مقداری را باز نمی گرداند، بلکه تنها وضعیت پین مربوط به یک GPIO را ، صرف نظر از مقدار فعلی آن ، معکوس می کند.

✓ آرگومان اول GPIOx نام پورتهی که قرار است مقدار دهی شود که بسته به میکروبی که استفاده می کنید می تواند از **GPIOA** تا **GPIOG** متغیر باشد.

✓ آرگومان دوم شماره پین مربوط به GPIO است. که می تواند مقداری بین **GPIO_PIN_0** الی **GPIO_PIN_15** باشد.

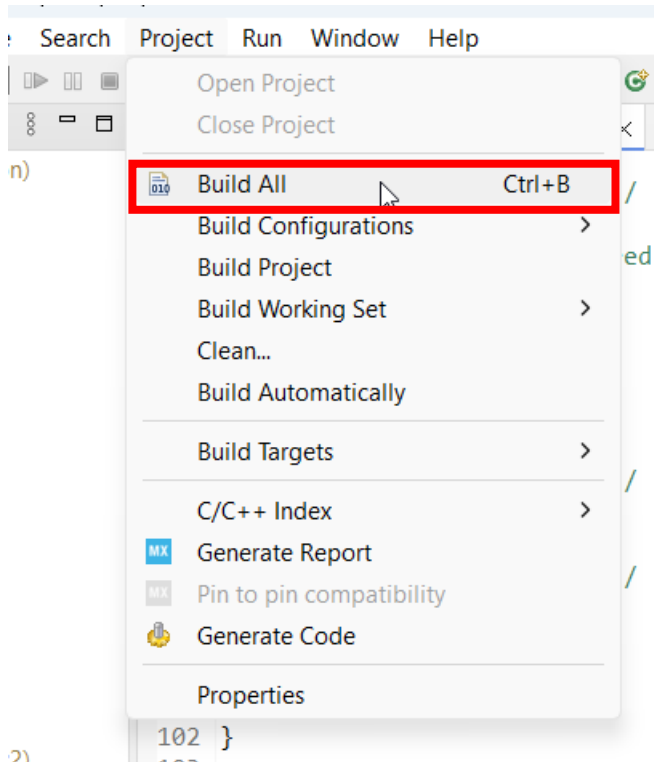
شروع کد نویسی در نرم افزار STM32CubeIDE

```
HAL_Delay(uint32_t Delay)
```

- ✓ یکی از توابع کتابخانه‌ی HAL در محیط STM32Cube است که برای ایجاد تأخیر زمانی به کار می‌رود.
- ✓ ورودی تابع، یک عدد از نوع uint32_t است که مقدار آن مدت تأخیر را بر حسب میلی‌ثانیه مشخص می‌کند.
- ✓ این تابع معمولاً با استفاده از تایمر سیستم (SysTick Timer) پیاده‌سازی شده و تا پایان مدت مشخص شده، اجرای برنامه را متوقف می‌کند.

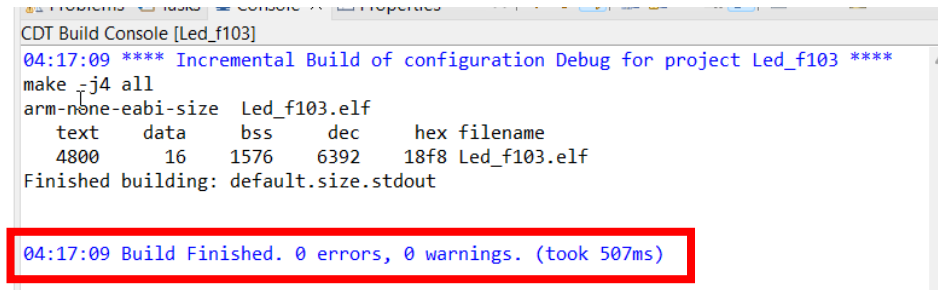


شروع کد نویسی در نرم افزار STM32CubeIDE



✓ نهایتا 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