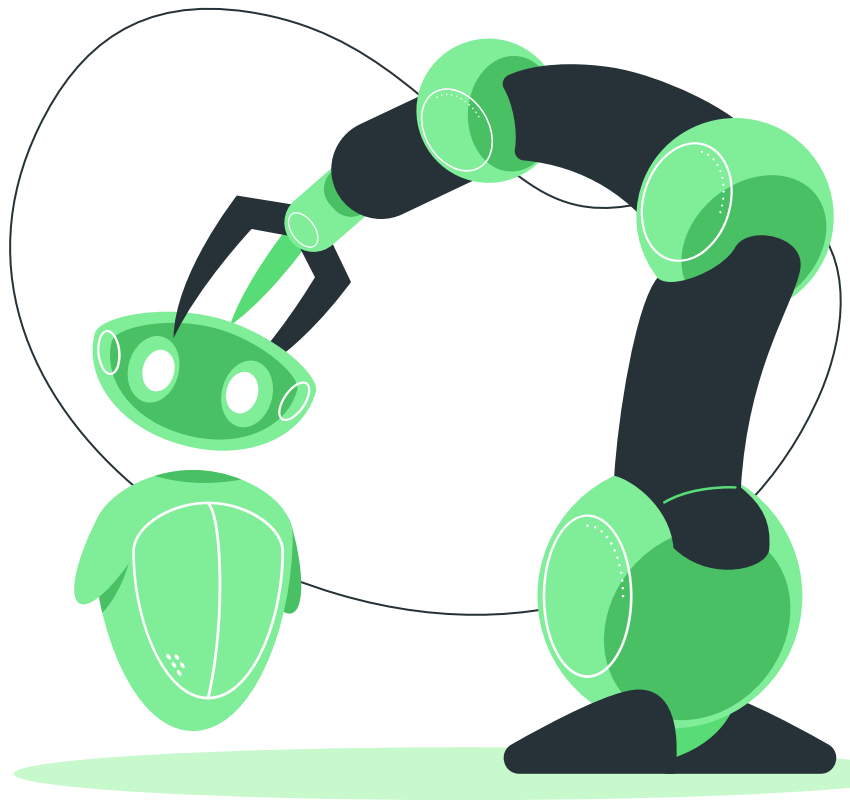


Freeways
free software club

STM32 Workshop

Session 2

By Moktar SELLAMI



Plan

1

Quiz

2

Setup ST account

3

**STM32Cube IDE
Installation**

4

HAL

5

HAL APIs

6

Blinking an LED

7

Reading Button state

Quiz



Quiz

1. What does GPIO stand for?
 - A) General Processor Input/Output
 - B) General Purpose Input/Output
 - C) Generic Port Input/Output
 - D) General Programmable Input/Output

Quiz

2. On an STM32 microcontroller, how are GPIO pins grouped?

- A) Into clusters called PORTS (e.g., GPIOA, GPIOB)
- B) Into clusters called BANKS
- C) Into clusters called MODULES
- D) Into clusters called UNIT

Quiz

3. If an internal green LED is connected to "PD12" meaning:

- A) It is connected to Peripheral Device 12.
- B) It is connected to Pin Diagram 12
- C) It is connected to Power Domain 12.
- D) It is connected to Port D, Pin 12..

Quiz

4. Which of the following is NOT one of the four main GPIO modes on an STM32?

- A) Input
- B) Output
- C) Analog
- D) Serial
- E) Alternate Function

Quiz

5. In a GPIO output configuration, what is a key characteristic of the "Push-Pull" mode?

- A) It can only sink current, connecting the pin to ground for a logic low.
- B) It uses an external resistor to pull the line high when inactive.
- C) It can actively drive the pin both high (to VCC) and low (to GND).
- D) It is used exclusively for analog signals like audio.

Quiz

6. Why is an internal "Pull-Up" resistor often used with a GPIO pin configured as an input?

- A) To increase the pin's output current for driving an LED.
- B) To set a known default logic state (high) when no external signal is present.
- C) To convert the pin into an analog input for a sensor.
- D) To protect the pin from electrostatic discharge (ESD).

Quiz

7. In the context of STM32's HAL, which functions would you use to change the state of an LED ?

- A) HAL_GPIO_ReadPin()
- B) HAL_GPIO_WritePin()
- C) HAL_GPIO_TogglePin()
- D) HAL_GPIO_LockPin()

Quiz

Answers:

1 - B

4 - D

7 - B & C

2 - A

5 - C

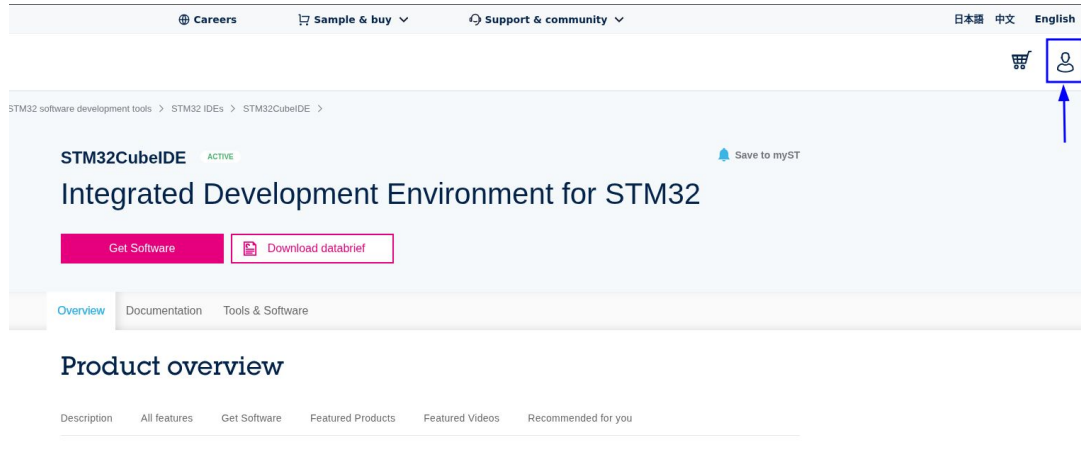
3 - D

6 - B

Setting up the Environment: STM32 CubeIDE

0- You need a lot of patience.

1- Go this [link](#) and create a account, you will get an email verification.



Setting up the Environment: STM32 CubeIDE

2- Create the account, fill in the necessary information

Fill in only the necessary info that are indicated with *

Already registered?

Enter your e-mail address and password to login your myST user.

E-mail address

Password

☐ Remember me on this device. ⓘ

Login

[Forgot password?](#)

New user?

myST brings you a set of personalized features:

- Participate to ST Events
- Stay informed with ST eNewsletters
- Get help with ST Online Support
- Discuss on the ST Community
- Benefit from our Online Design Tools
- Download Software
- Order free samples
- Manage your weekly product updates
- Buy ST Products & Tools

Create Account

Setting up the Environment: STM32 CubeIDE

3- After some time you will receive an email of verification, verify it and your St account should be created.

Already registered?

Enter your e-mail address and password to login your myST user.

E-mail address

Password

☐ Remember me on this device. 

Login

[Forgot password?](#)

New user?

myST brings you a set of personalized features:






- Participate to ST Events
- Stay informed with ST eNewsletters
- Get help with ST Online Support
- Discuss on the ST Community
- Benefit from our Online Design Tools
- Download Software
- Order free samples
- Manage your weekly product updates
- Buy ST Products & Tools

Create Account

Setting up the Environment: STM32 CubeIDE

4- Go this [link](#) again and download the Latest STM32 CubeIDE version

Get Software

Part Number	General Description	Latest version	Download	All versions
+ STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	1.19.0	Get latest	Select version  ubuntu
+ STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	1.19.0	Get latest	Select version 
+ STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	1.19.0	Get latest	Select version 
+ STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	1.19.0	Get latest	Select version 
+ STM32CubeIDE-Win	STM32CubeIDE Windows Installer	1.19.0	Get latest	Select version  windows

STMicroelectronics recommends always keeping your software up to date

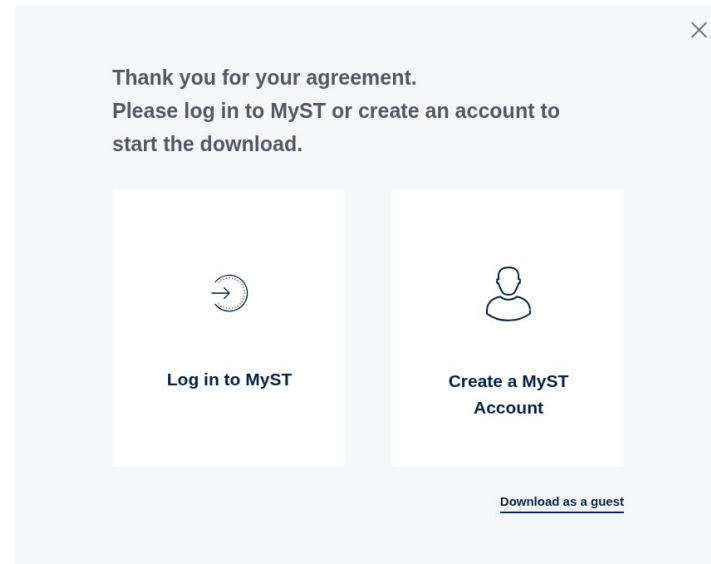
Setting up the Environment: STM32 CubeIDE

5- Accept to the Licence agreement

6- It will prompt you to Login or use a guest choose Log in to MyST

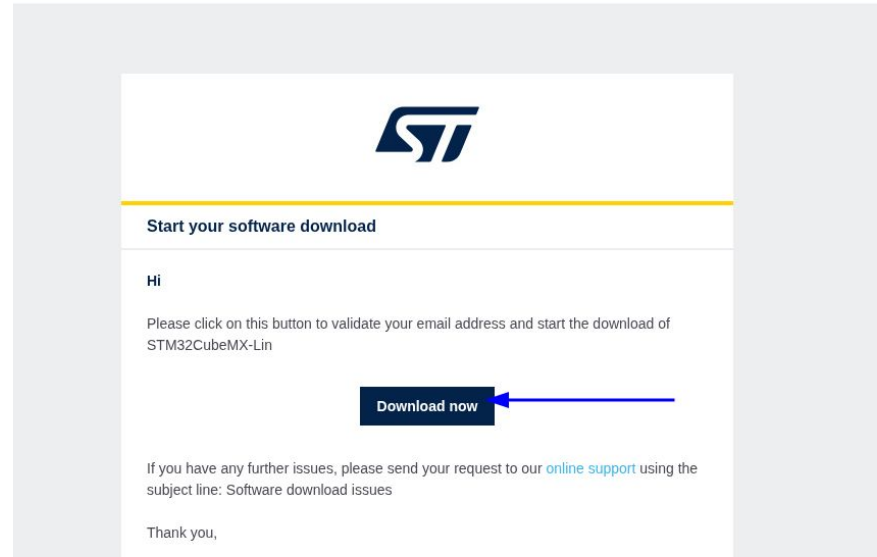
Note:

We create an account
because, CubeIDE requires it
sometimes



Setting up the Environment: STM32 CubeIDE

7- An Email should be sent to you, give it some time .

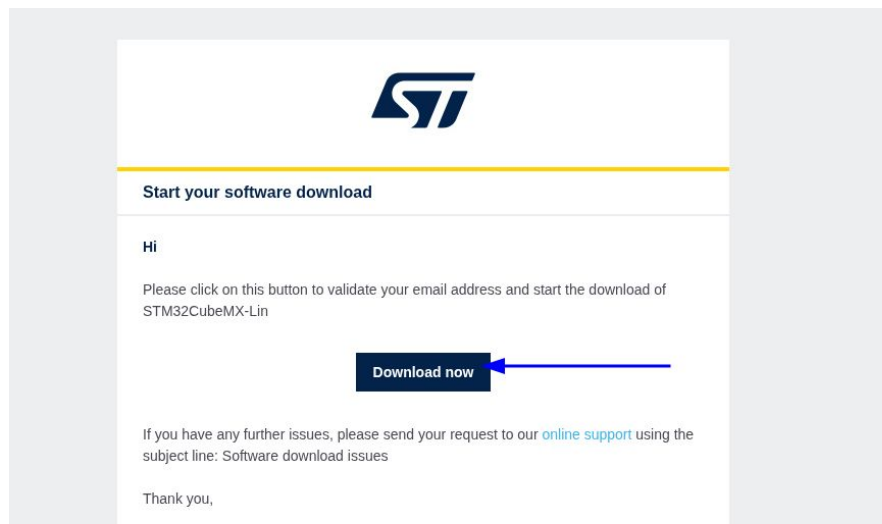


Setting up the Environment: STM32 CubeIDE

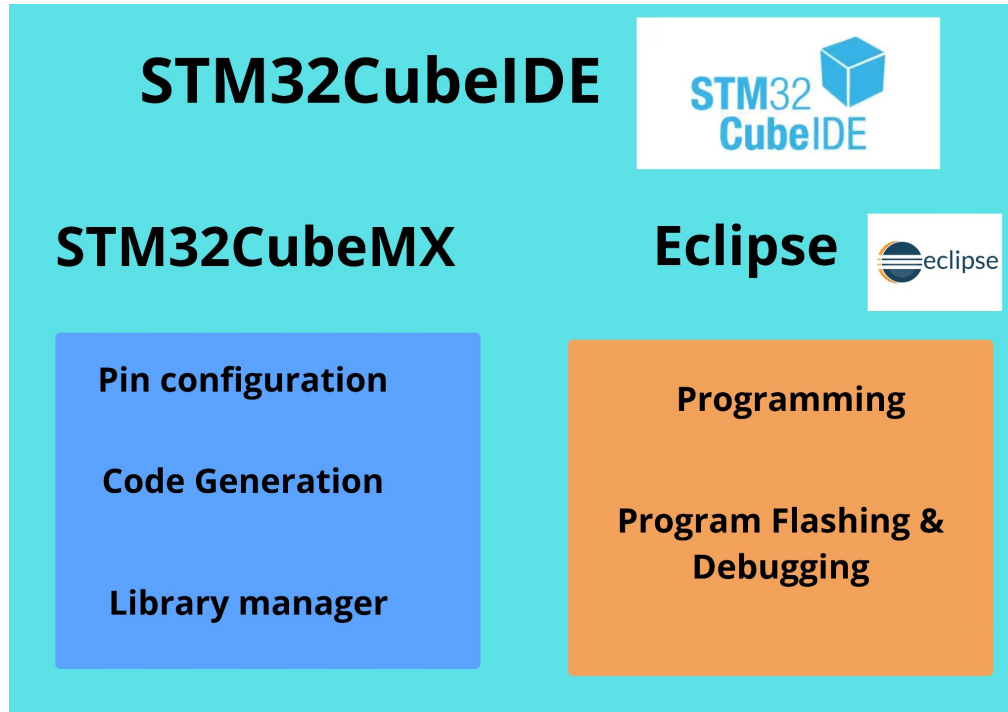
8- it will open a new session on you browser and hopefully, is will start downloading

Note:

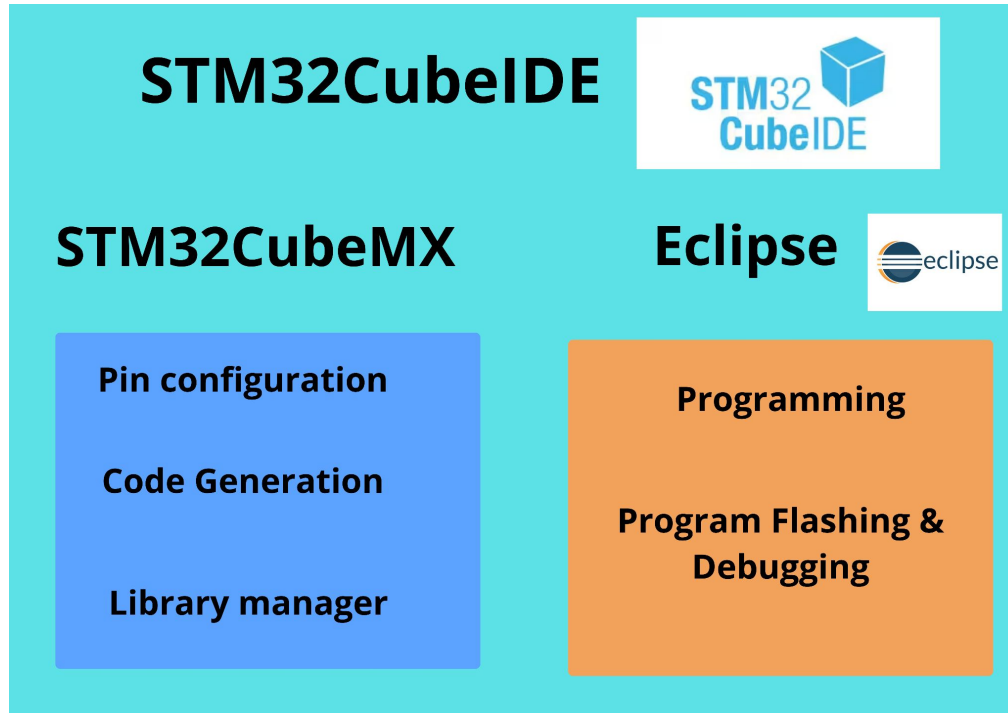
In case it didn't start, restart the downloading process.



STM32Cube IDE



STM32Cube IDE



STM32Cube IDE: new project

workshop - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help Hello kakashi

Information Center X

STM32CubeIDE Home

Start a project

- Start new STM32 project
- Start new project from STM32CubeMX file (.ioc)
- Import project
- Import STM32Cube example

Data collection information notice

Welcome to STM32CubeIDE

What's new

STM32CubeIDE

GCC12, the default compiler

- Bugfixes and overall improvements of performance and code size optimizations
- Added support of non-ASCII characters on Windows target

Toolchain manager still allows using any GCC version of your choice

Quick links

- STM32CubeIDE resource portal on wiki
- STM32CubeIDE manuals

Support & Community

- ST Home
- ST Community
- ST Longevity Commitment

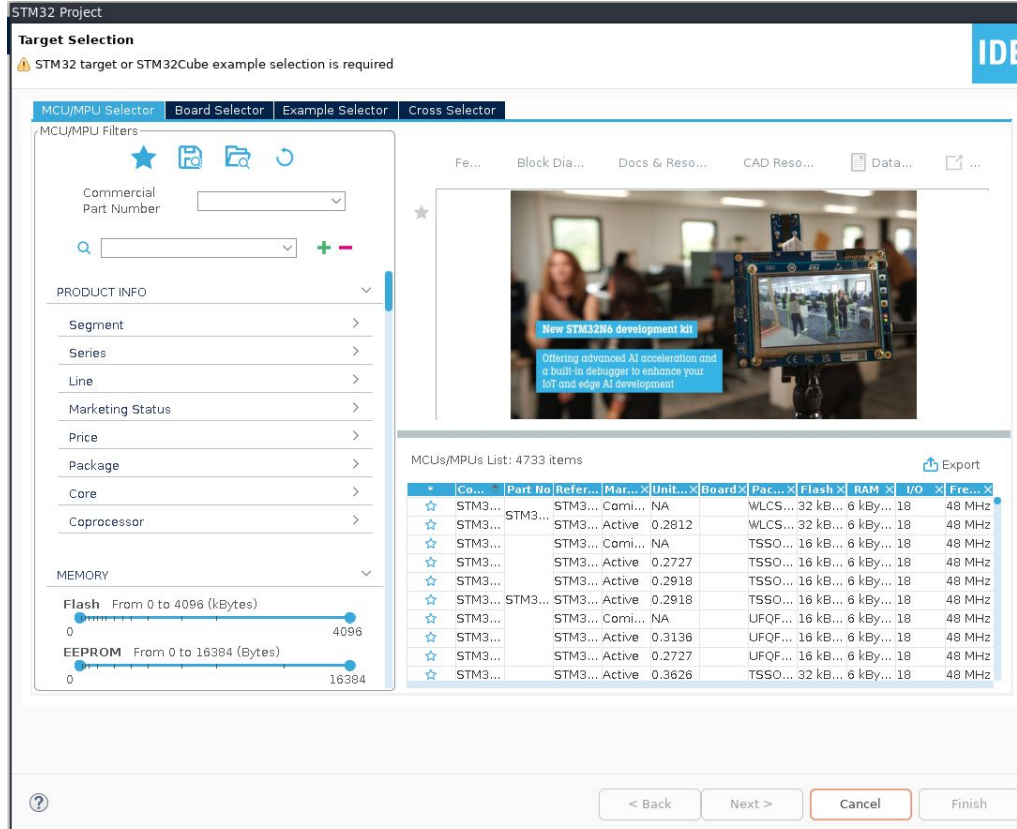
Standalone STM32 Tools

- STM32CubeMX
- STM32CubeMonitor
- STM32CubeMon-Pwr
- STM32CubeMon-RF
- STM32CubeMon-UCPD
- STM32CubeProg

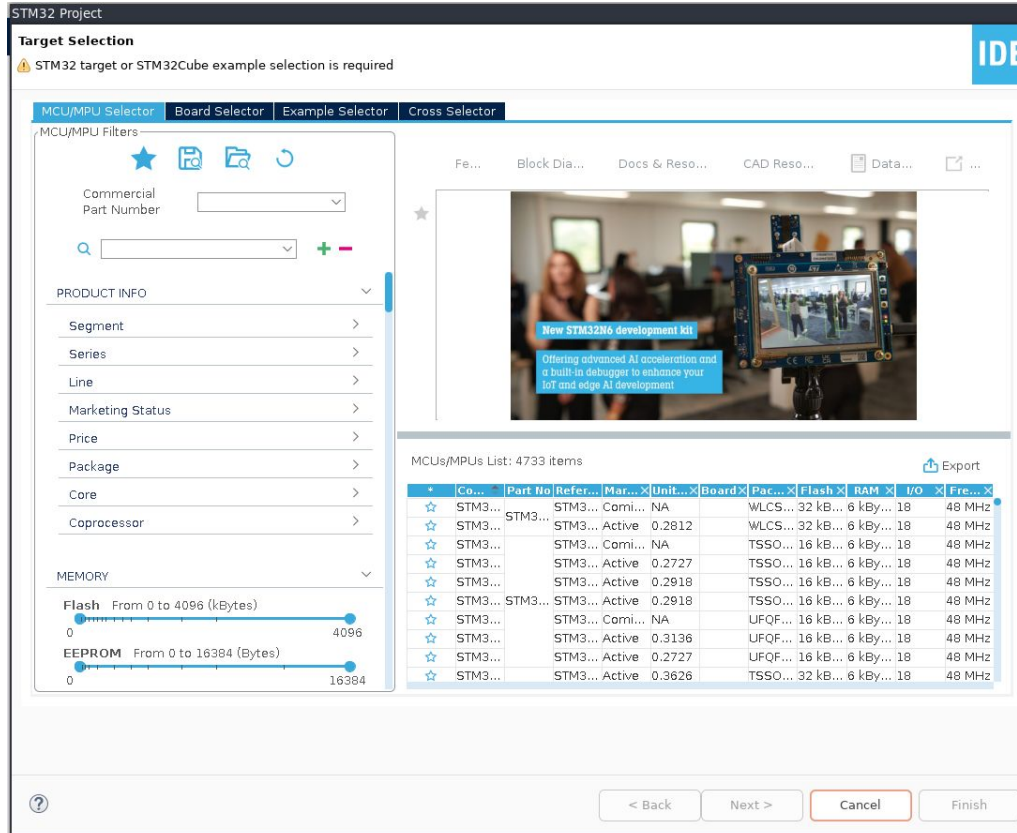
Application Tools

- eDesignSuite
- AlgoBuilder
- ST-MC-Suite

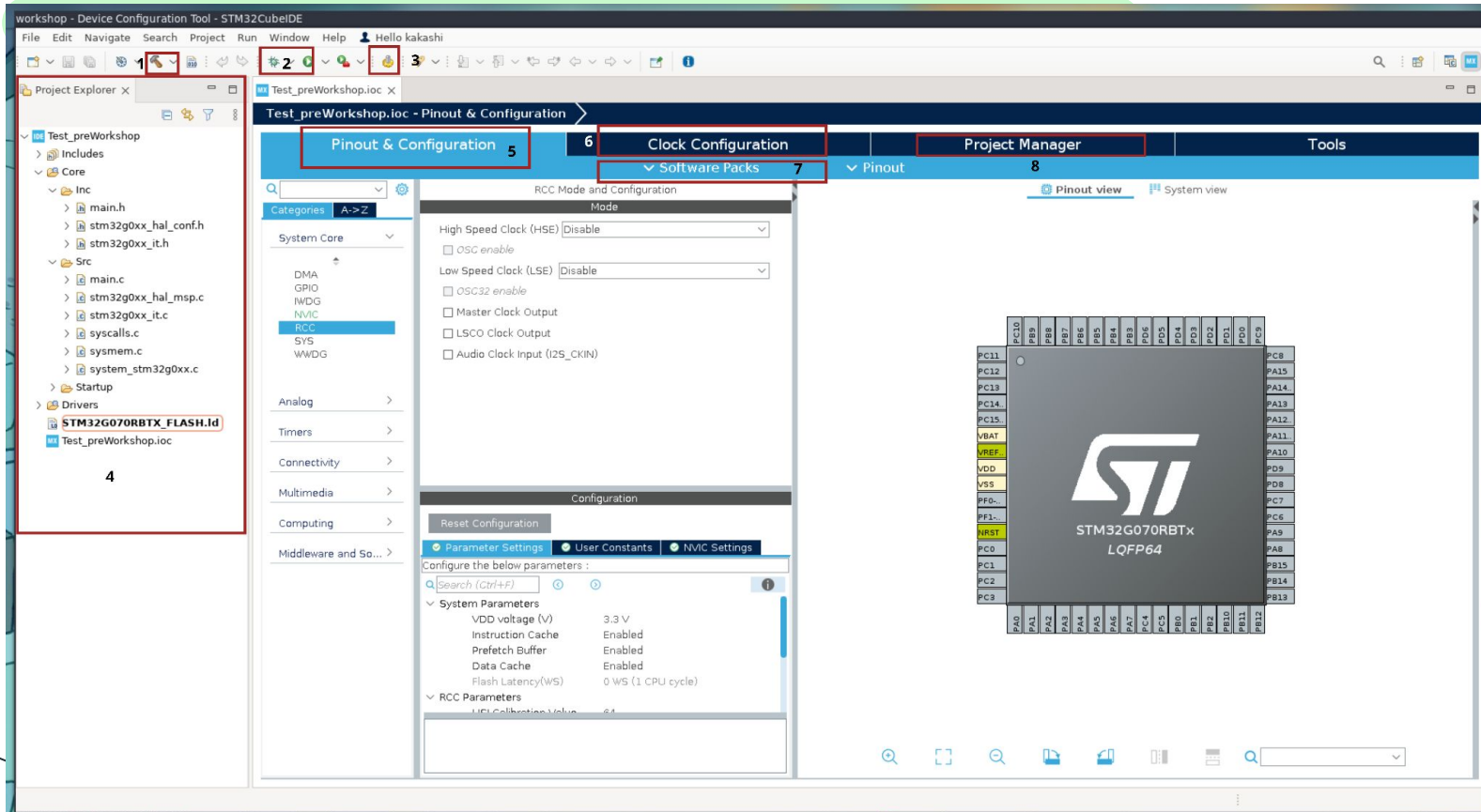
STM32Cube IDE: select target



STM32Cube IDE: Creating the project

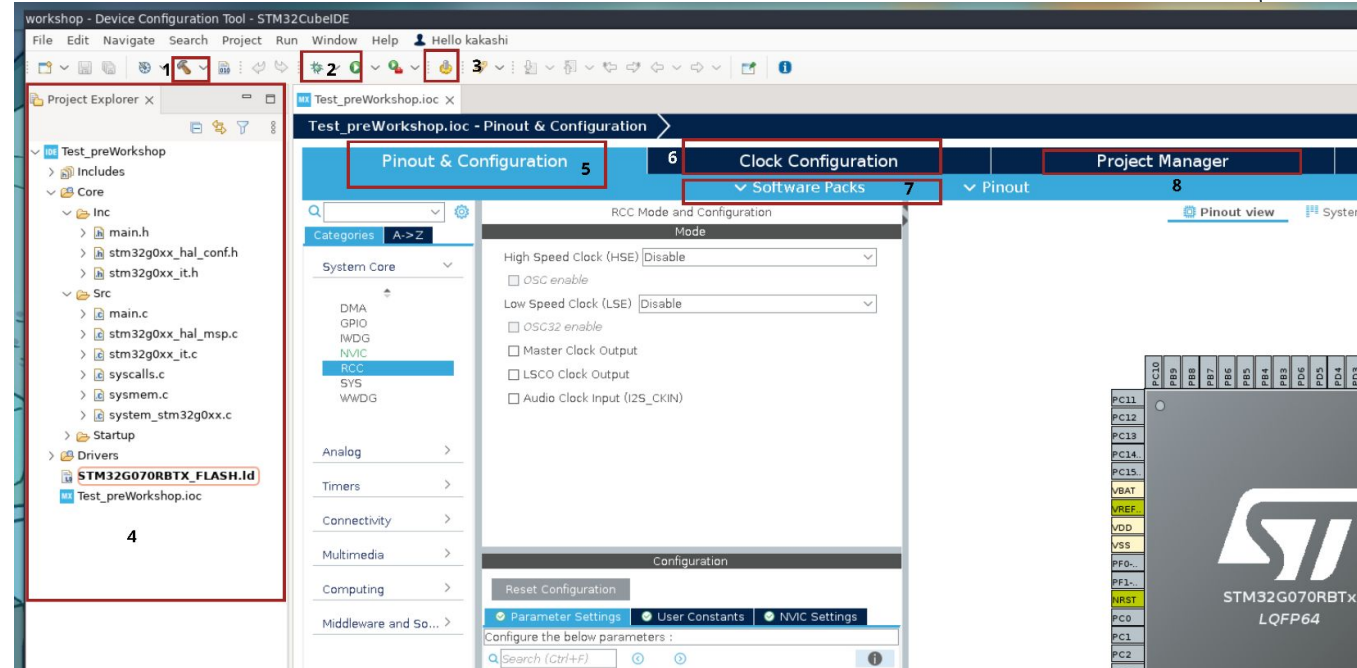


STM32Cube IDE: Creating the project



STM32Cube IDE: Creating the project

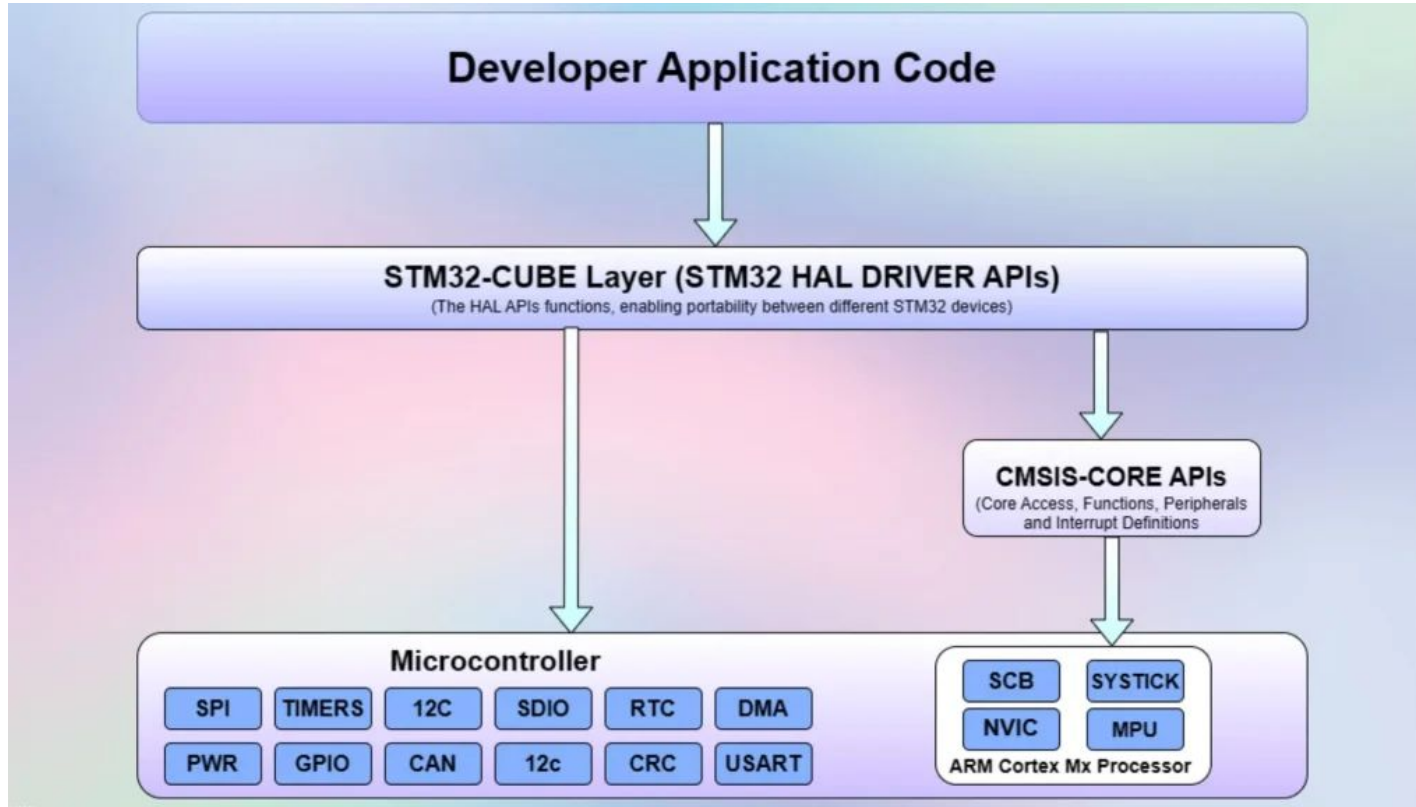
- 1- Build
- 2- Run & debug
- 3- Generate code
- 4- Project structure
- 5- Pinout config
- 6- clock config
- 7- Software packs
- 8- Project manager



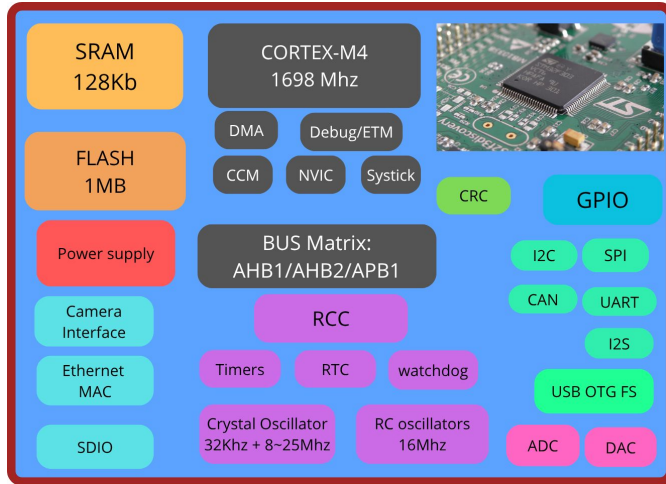
HAL

HAL is a software layer provided by STMicroelectronics that abstracts the hardware details of STM32 microcontrollers. It provides a standardized set of functions to interact with peripherals (GPIO, UART, I2C, etc.) regardless of the specific STM32 chip you're using.

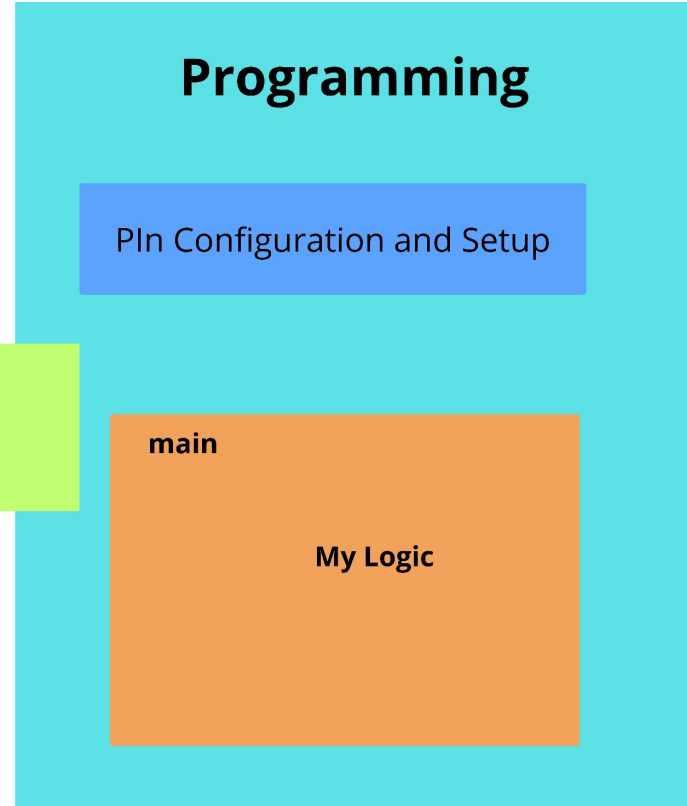
HAL



Programming: HAL



HAL APIs



Programming: Pin Configuration

Enable BUS Clock for GPIO

GPIO Configuration

Main Program

HAL:

GPIO InitStruct

```

18 /**.
17  * @brief GPIO Init structure definition..
16  */.
15 typedef struct
14 {
13     uint32_t Pin;        /*!< Specifies the GPIO pins to be configured.
12                          This parameter can be any value of @ref GPIO_pins_define */
11
10     uint32_t Mode;       /*!< Specifies the operating mode for the selected pins.
9                          This parameter can be a value of @ref GPIO_mode_define */
8
7     uint32_t Pull;       /*!< Specifies the Pull-up or Pull-Down activation for the selected pins.
6                          This parameter can be a value of @ref GPIO_pull_define */
5
4     uint32_t Speed;      /*!< Specifies the speed for the selected pins.
3                          This parameter can be a value of @ref GPIO_speed_define */
2
1     uint32_t Alternate;  /*!< Peripheral to be connected to the selected pins..
61  This parameter can be a value of @ref GPIO_Alternate_function_selection */
1 }GPIO_InitTypeDef;
2

```

HAL:

GPIO Mode

```

19
18 /** @defgroup GPIO_mode_define GPIO mode define
17  * @brief GPIO Configuration Mode
16  *      Elements values convention: 0x00WX00YZ
15  *      - W : EXTI trigger detection on 3 bits
14  *      - X : EXTI mode (IT or Event) on 2 bits
13  *      - Y : Output type (Push Pull or Open Drain) on 1 bit
12  *      - Z : GPIO mode (Input, Output, Alternate or Analog) on 2 bits
11  * @{
10  */
9  #define GPIO_MODE_INPUT          MODE_INPUT
8  #define GPIO_MODE_OUTPUT_PP      (MODE_OUTPUT | OUTPUT_PP)
7  #define GPIO_MODE_OUTPUT_OD      (MODE_OUTPUT | OUTPUT_OD)
6  #define GPIO_MODE_AF_PP          (MODE_AF | OUTPUT_PP)
5  #define GPIO_MODE_AF_OD          (MODE_AF | OUTPUT_OD)
4
3  #define GPIO_MODE_ANALOG         MODE_ANALOG
2  ....

```

HAL:

GPIO Speed

```

22  */
21
20 /** @defgroup GPIO_speed_define GPIO speed define
19  * @brief GPIO Output Maximum frequency
18  * @{
17  */
16 #define GPIO_SPEED_FREQ_LOW          0x00000000U
15 #define GPIO_SPEED_FREQ_MEDIUM      0x00000001U
14 #define GPIO_SPEED_FREQ_HIGH       0x00000002U
13 #define GPIO_SPEED_FREQ_VERY_HIGH  0x00000003U
12 /**
11  * @}
10  */
9

```


HAL:

GPIO Pin state

```

15 /**.
14  * @brief  GPIO Bit SET and Bit RESET enumeration.
13  */
12 typedef enum
11 {
10     GPIO_PIN_RESET = 0,
9     GPIO_PIN_SET
8 }GPIO_PinState;

```

HAL GPIO APIs:

HAL function

```

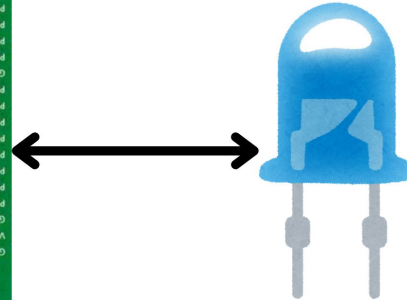
10
9 /** @addtogroup GPIO_Exported_Functions_Group2
8  * @{
7  */
6 /* IO operation functions *****/
5 GPIO_PinState HAL_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);
4 void HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);
3 void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);

4
3 /** @addtogroup GPIO_Exported_Functions_Group1
2  * @{
1  */
223 /* Initialization and de-initialization functions *****/
1 void HAL_GPIO_Init(GPIO_TypeDef *GPIOx, GPIO_InitTypeDef *GPIO_Init);
2 void HAL_GPIO_DeInit(GPIO_TypeDef *GPIOx, uint32_t GPIO_Pin);
3 /**
4  * @}
5  */

```

GPIO output STM32 and HAL: blinking an LED:

Toggling an LED



STM32F407

GPIO
PIN12

GPIO Input STM32 and HAL: Button

Reading button state



STM32F407

GPIOA
PIN0

Freeways
free software club

**Thank
You**

