

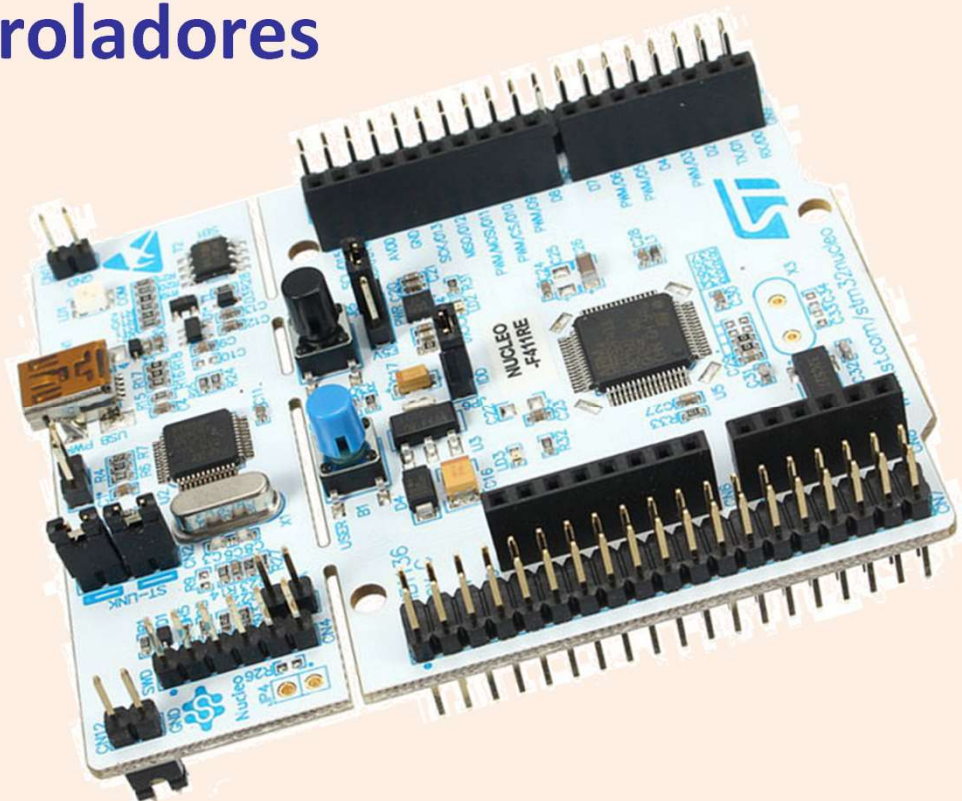
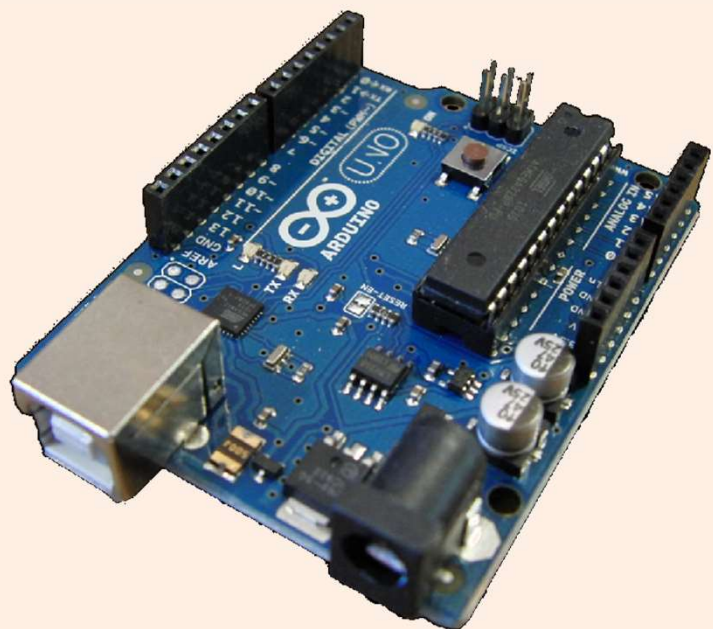
Competências Transferíveis

Microcontroladores e Interação com Sensores e Atuadores

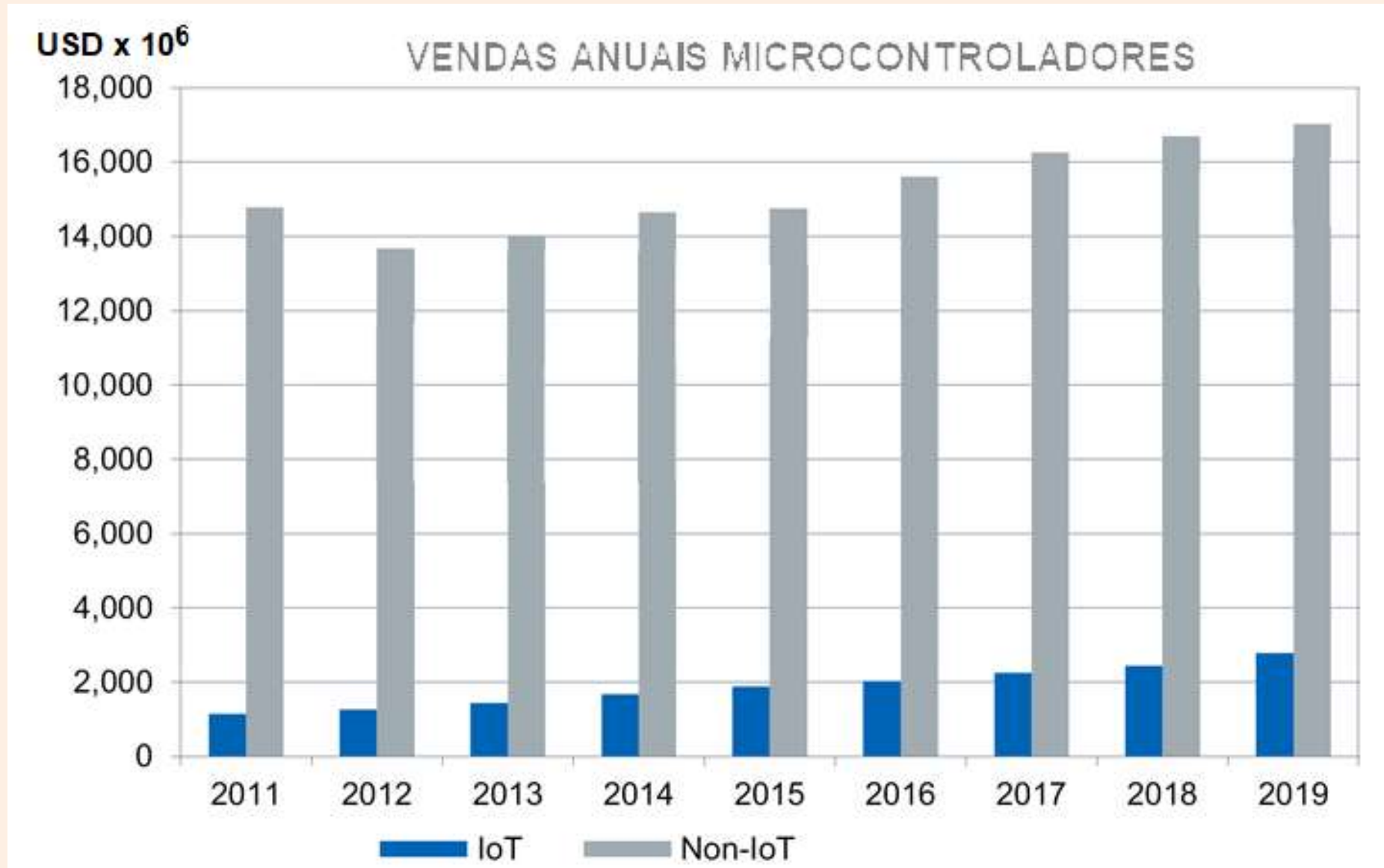
V3_2

Rui Escadas Martins

Introdução aos Microcontroladores



Introdução aos Microcontroladores

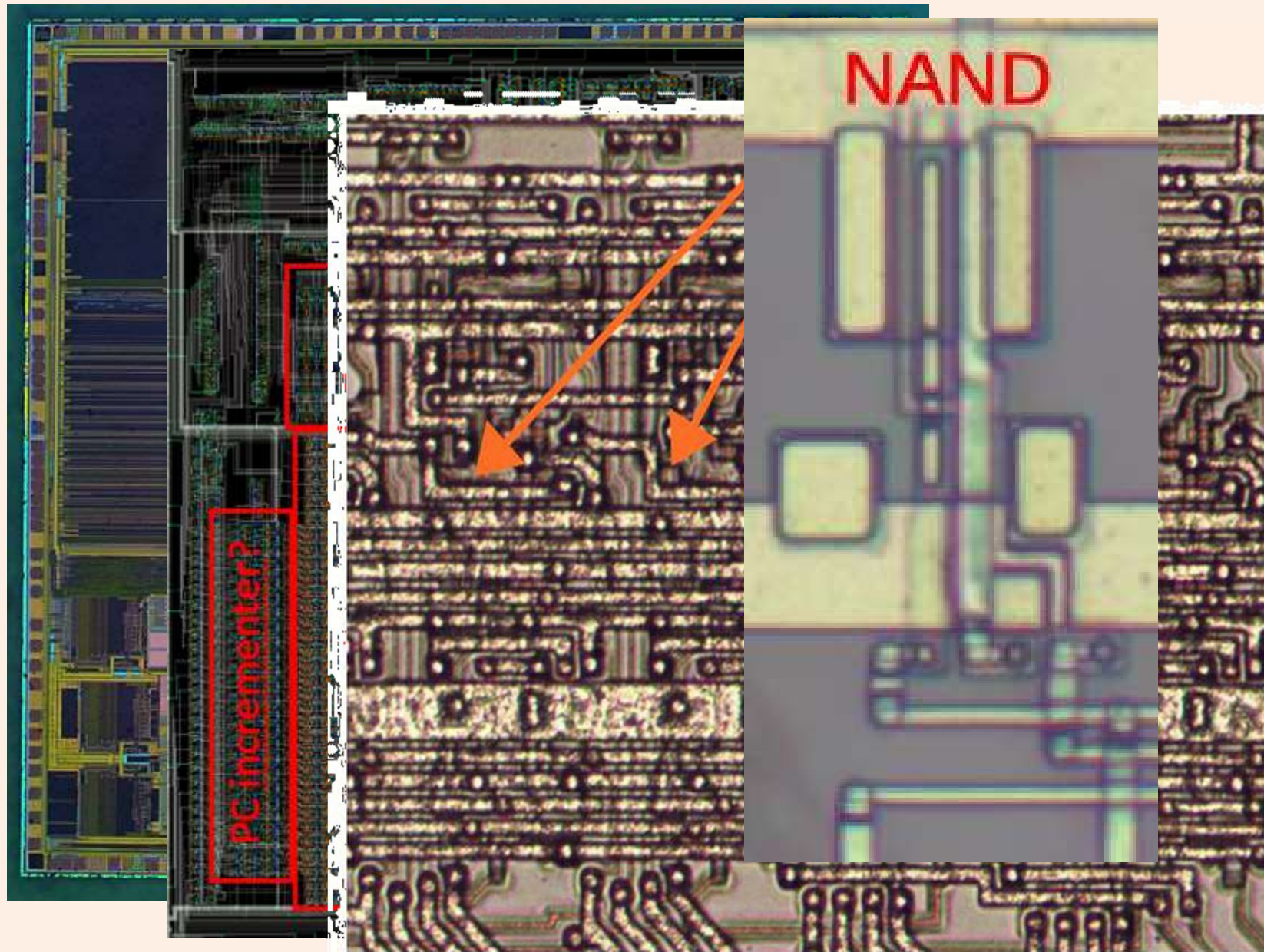


Introdução aos Microcontroladores

Um microcontrolador (MCU) é um pequeno computador implementado num único Circuito Integrado tipicamente baseado num CPU (Central Processing Unit) relativamente simples, integrando um razoável número de periféricos, como:

- Memória RAM;
- Memória Flash;
- I/Os programáveis;
- Timers e contadores;
- A/D – Conversores Analógico/Digitais;
- D/A – Conversores Digital/Analógicos;
- Etc...

Introdução aos Microcontroladores



Como escolher um microcontrolador?

Características mais importantes:

- Arquitectura: nº de bits, ARM, RISC V, MIPS, ...
- Potência consumida + Clock freq./Desempenho
- Nº de I/O pins
- Tamanho memória programa
- Tamanho memória RAM
- Possibilidades Analógicas:
 - Nº de ADCs + resolução + tempo de conversão
 - Nº de DACs + resolução + tempo de conversão
 - Nº de comparadores analógicos, opamps, etc...
- Nº de Timers + resolução
- Funções especiais

Como escolher um microcontrolador?

Características mais importantes:

Arquitectura: nº de bits, ARM, RISC V, MIPS, ...

- 8-bits é excelente para pequenas aplicações;
- 16-bit ok para aplicações mais complicadas;
- 32-bit para aplicações exigentes;
- ARM é a dominante para 32 bits.

Como escolher um microcontrolador?

Características mais importantes:

Potência consumida + Clock freq./Desempenho

- Mais clock -> Melhor desempenho;
- Mais clock -> Maior consumo;
- Maior consumo -> Menor duração da bateria.

Como escolher um microcontrolador?

Características mais importantes:

Nº de I/O pins

- Mais pinos -> Mais possibilidades de ligar periféricos;
- Mais pinos -> packages maiores -> maior custo;
- Mais pinos nem sempre implica maior desempenho.

Como escolher um microcontrolador?

Características mais importantes:

Tamanho memória programa (memória não volátil - actualmente quase sempre do tipo FLASH)

- Mais memória -> Permite maiores programas (o que por sua vez permite algoritmos mais complexos ou que usem mais dados fixos);
- Mais memória nem sempre implica maior desempenho.

Como escolher um microcontrolador?

Características mais importantes:

Tamanho memória RAM (volátil)

- Mais memória RAM-> Melhor desempenho em programs mais complexos;
- Mais memória permite mais fácil processamento de grande quantidade de dados.

Como escolher um microcontrolador?

Características mais importantes:

Possibilidades Analógicas

- Muito importante no interface e interacção com o mundo;
- A maioria das grandezas a medir são analógicas e precisam de ser convertidas em representação digital para processamento por um computador (é semelhante para a actuação);
- Nº ADCs, sua resolução e canais de entrada;
- Nº de DACs e sua resolução.

Como escolher um microcontrolador?

Características mais importantes:

Nº de Timers + resolução

- Muito importante para sistemas que são “disciplinados” por tempo;
- Muito importante para contagem muito precisa de tempo e eventos.

Como escolher um microcontrolador?

Características mais importantes:

Funções especiais

- Há muitas funções especiais.
- Por exemplo: comunicações:
 - **USB**
 - **U(S)ART**
 - **CAN**
 - **LIN**
 - **BLE**
 - **WiFi**
 - **Etc...**

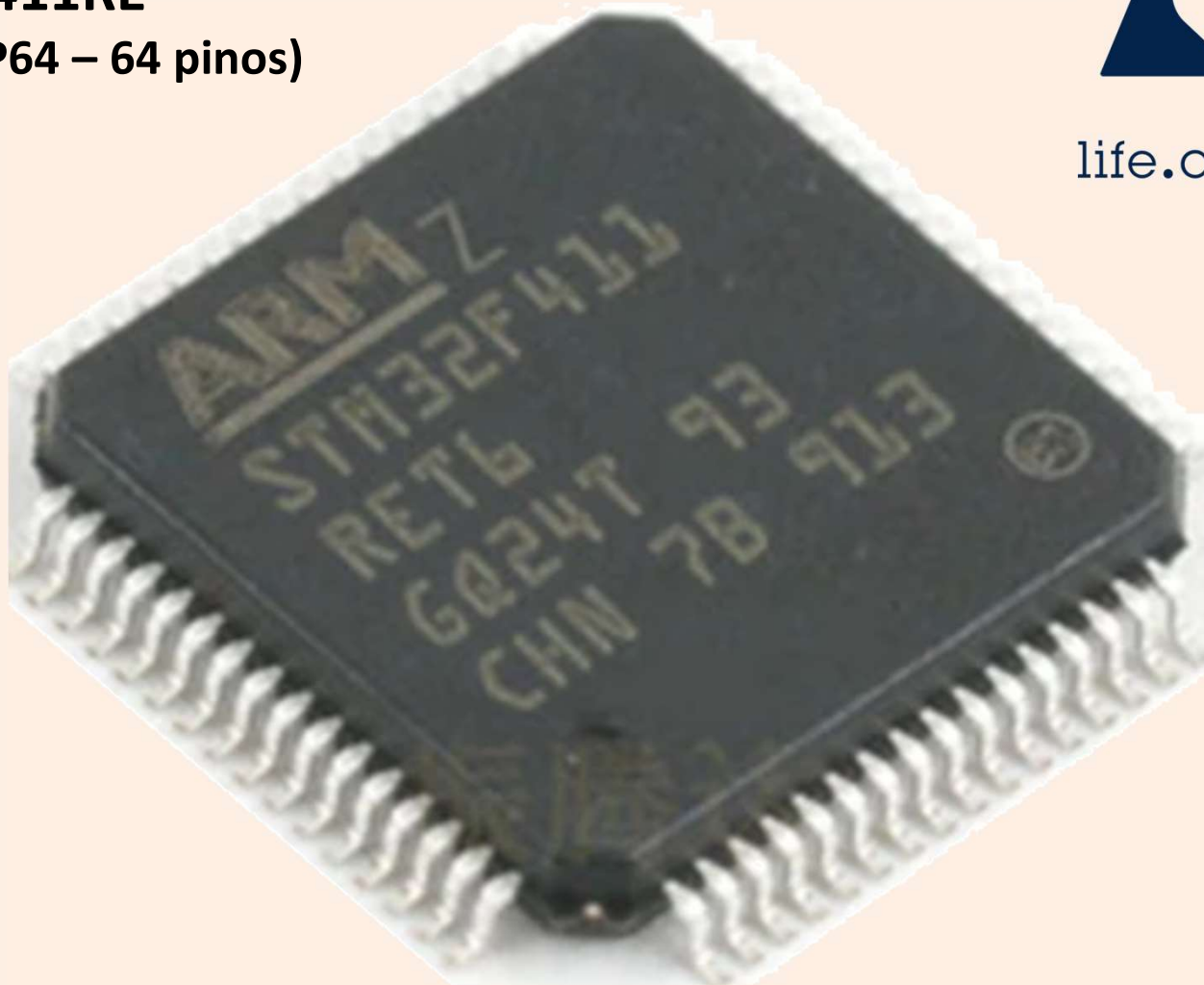
Introdução aos Microcontroladores

Chip:

STM32F411RE
(LQFP64 – 64 pinos)



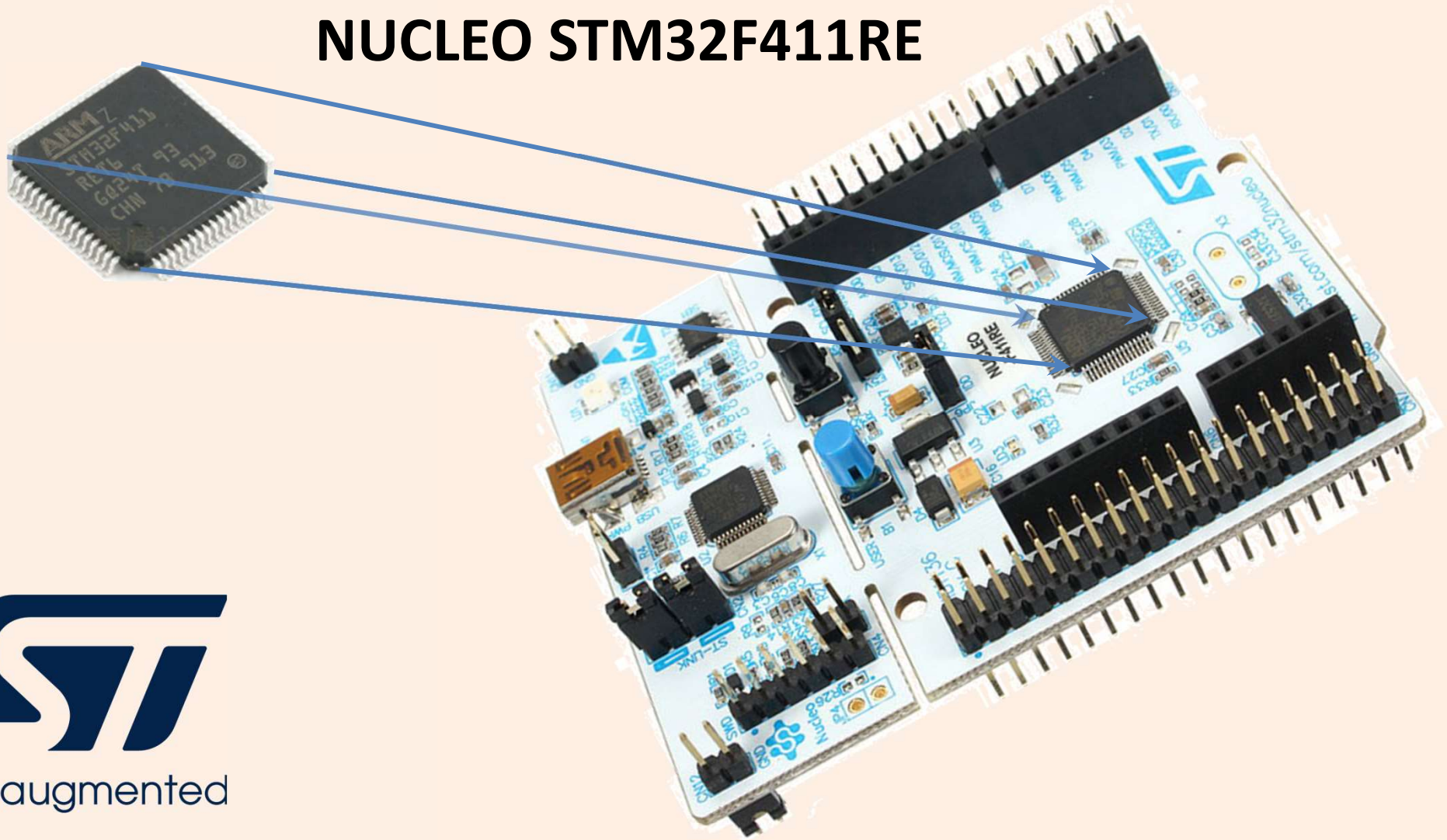
life.augmented



Introdução aos Microcontroladores

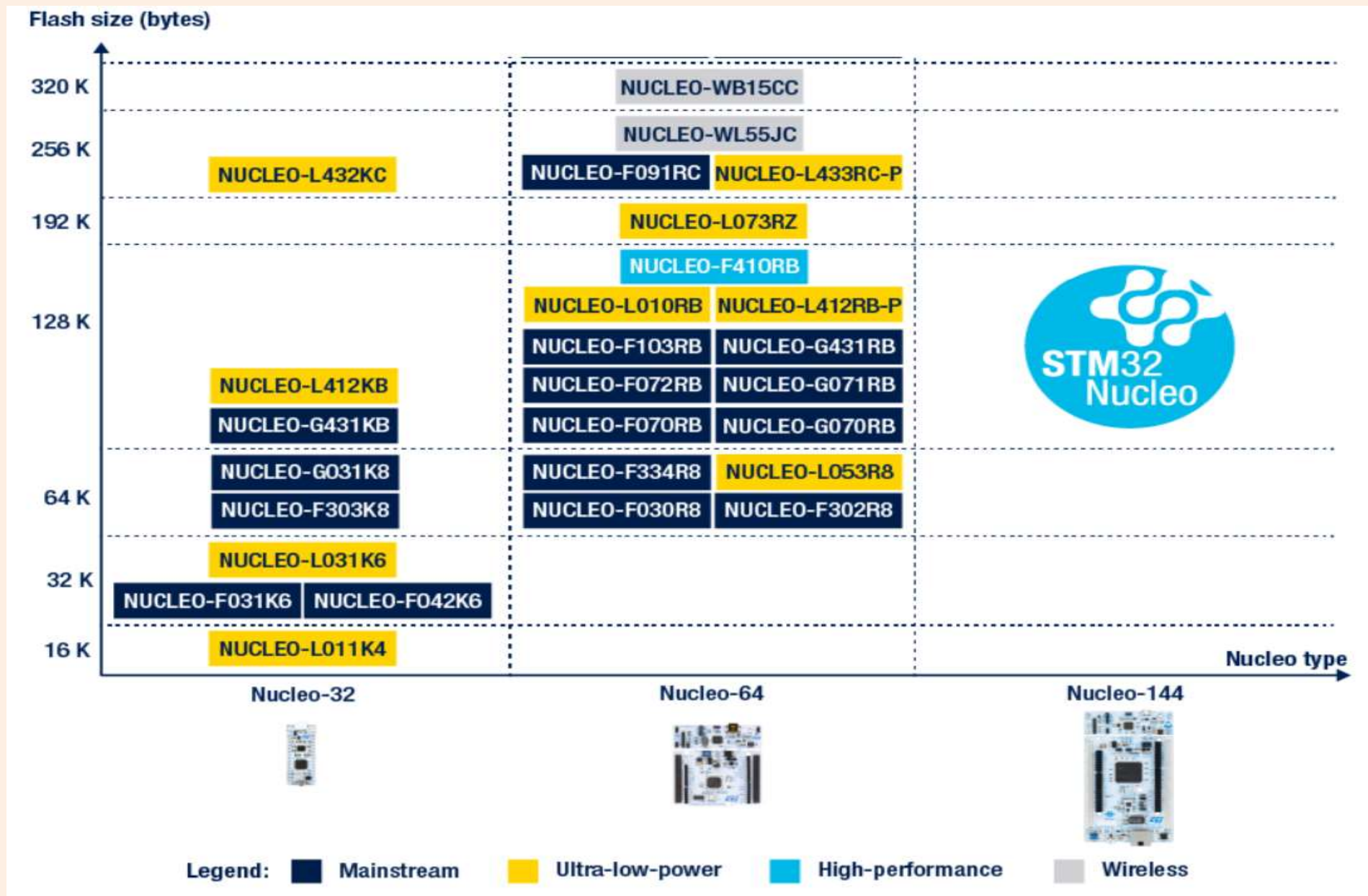
Placa desenvolvimento:

NUCLEO STM32F411RE

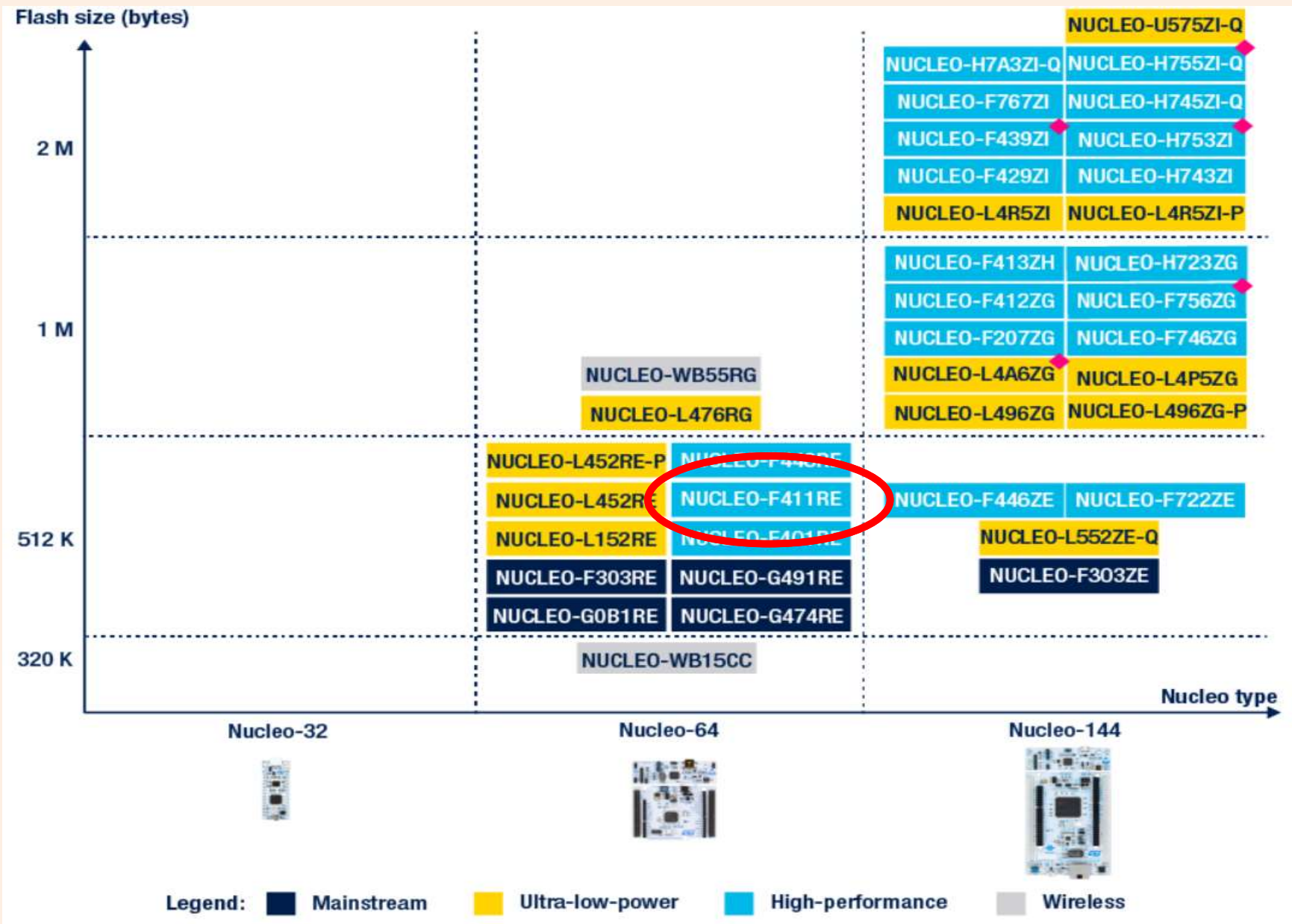


life.augmented

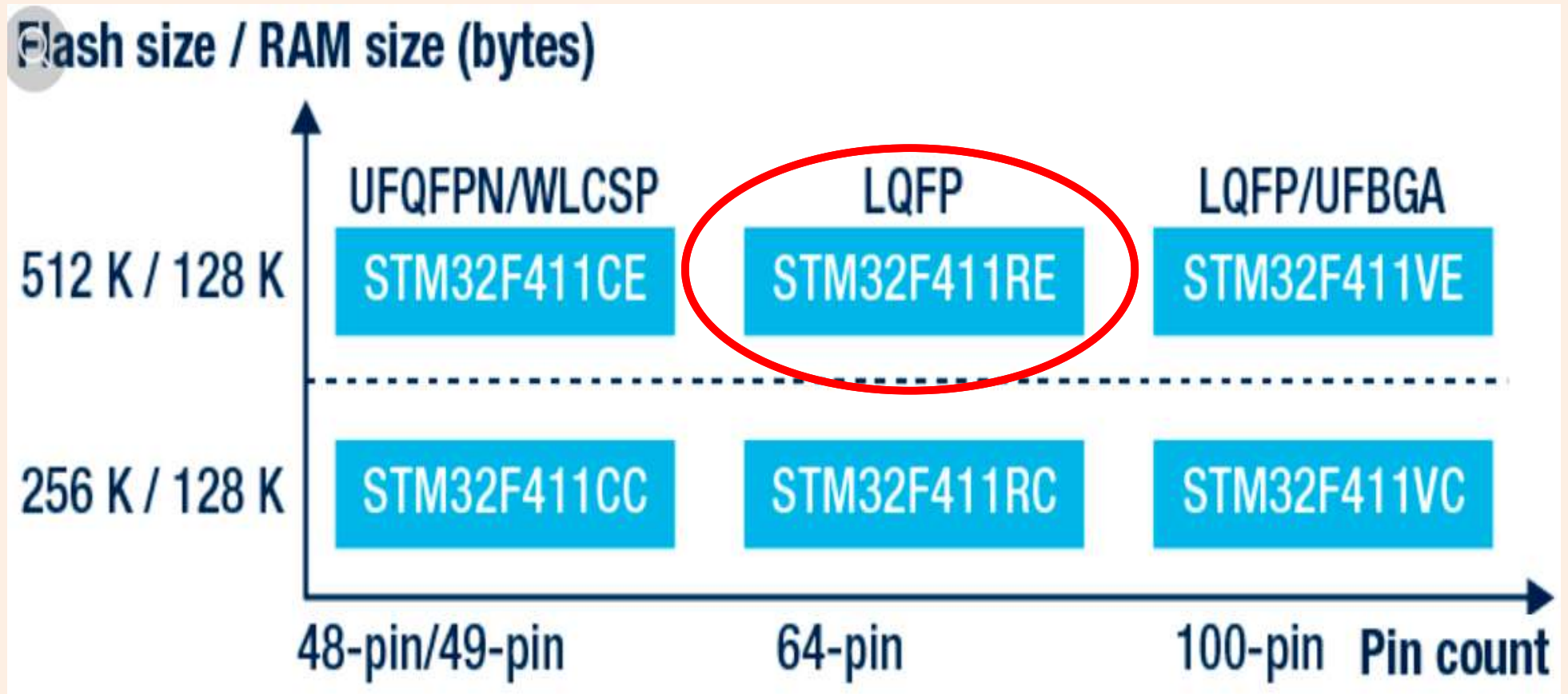
Introdução aos Microcontroladores



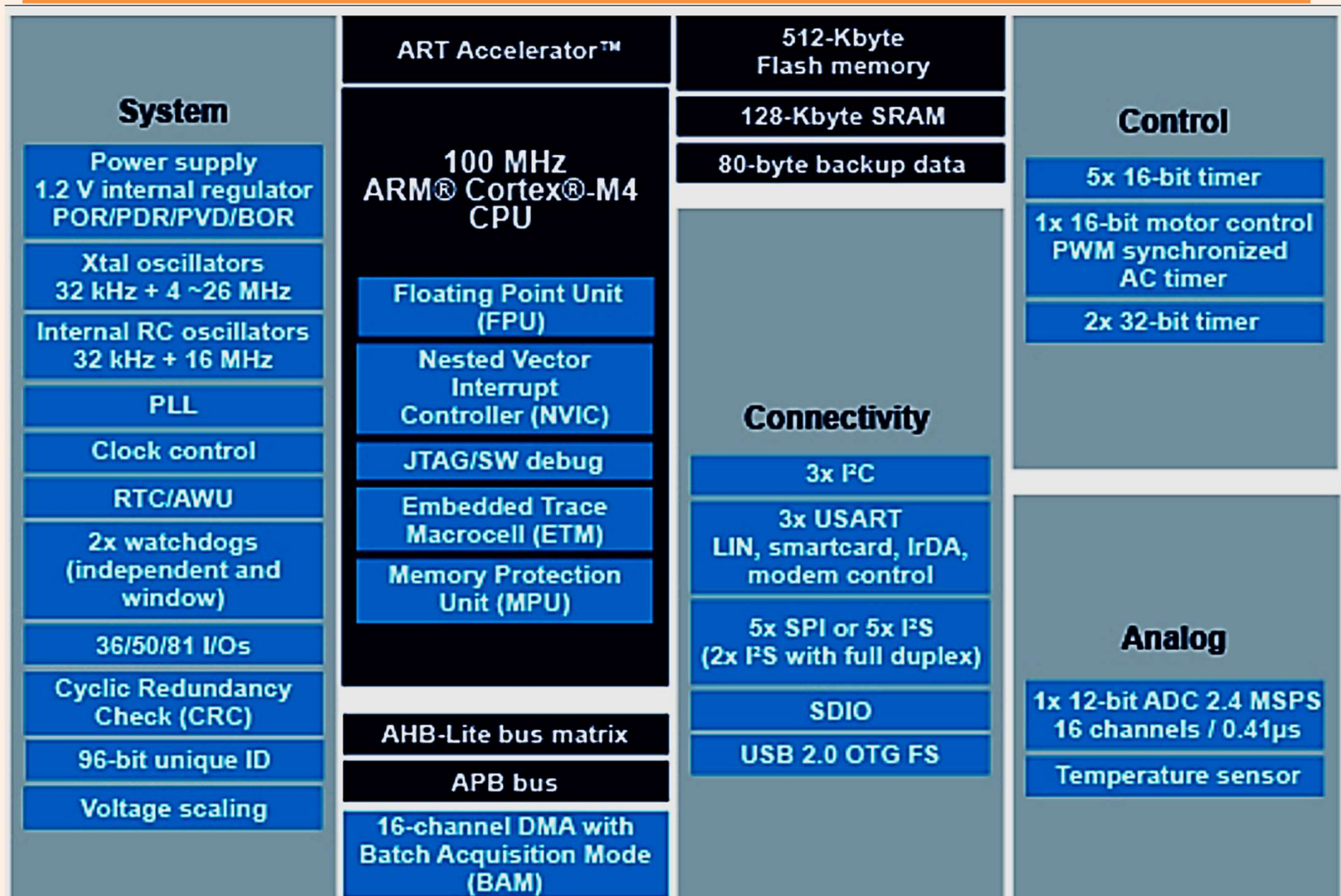
Introdução aos Microcontroladores



Introdução aos Microcontroladores

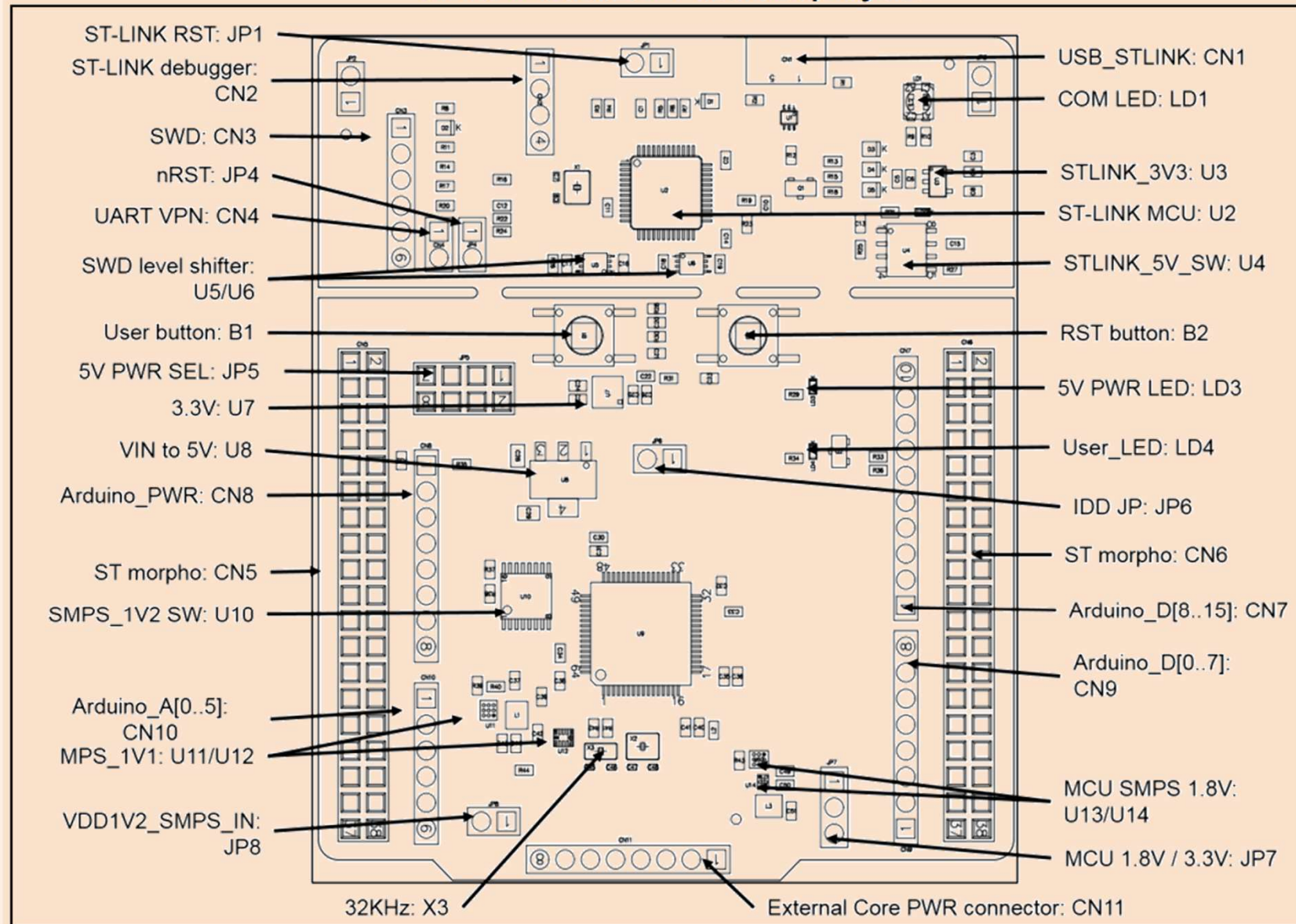


Introdução aos Microcontroladores



Introdução aos Microcontroladores

STM32 Nucleo-64-P board top layout



Get Started

Procedimento:

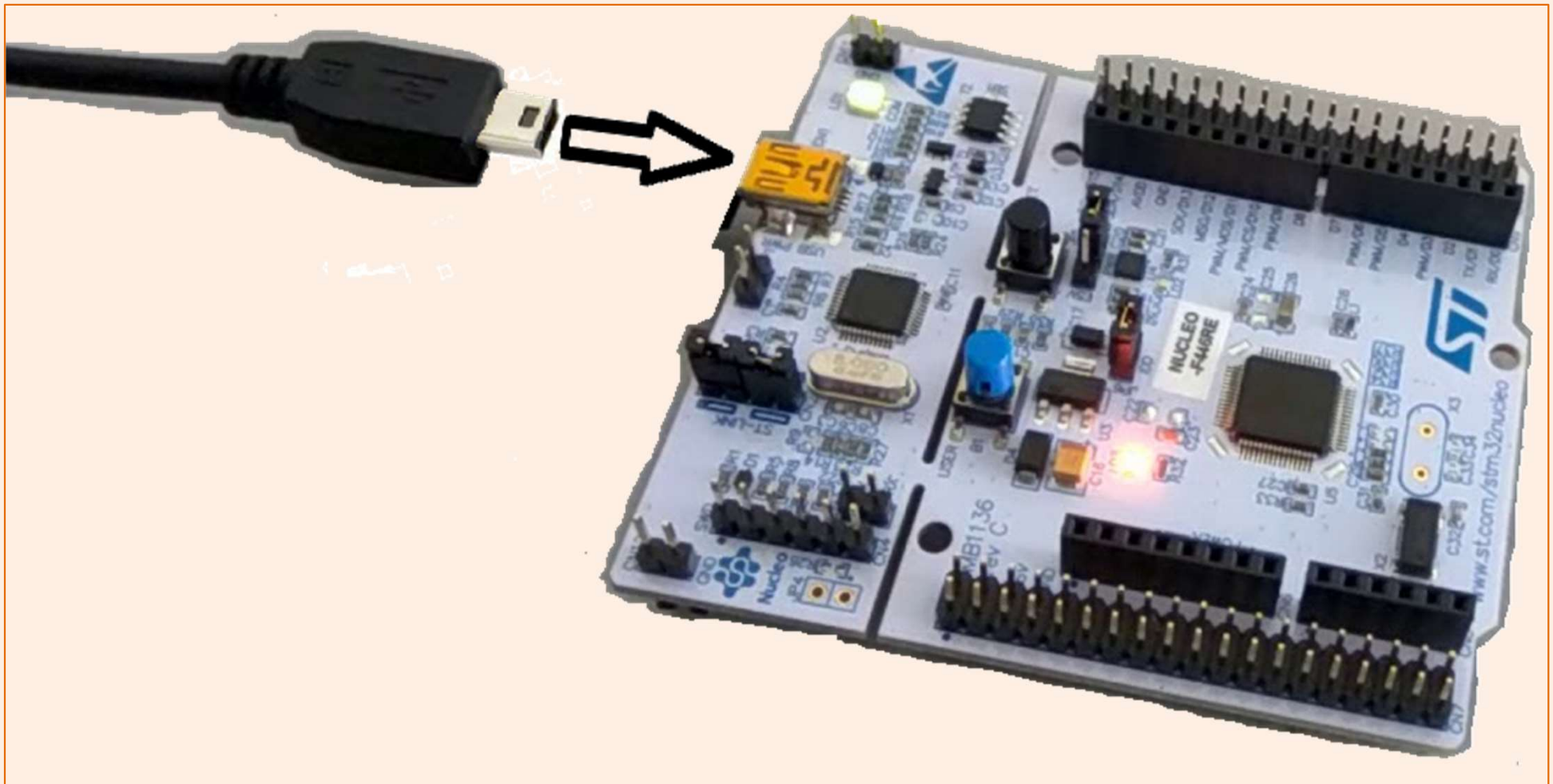
É só ligar a placa a um PC através de um cabo USB – Mini-USB



Get Started

Procedimento:

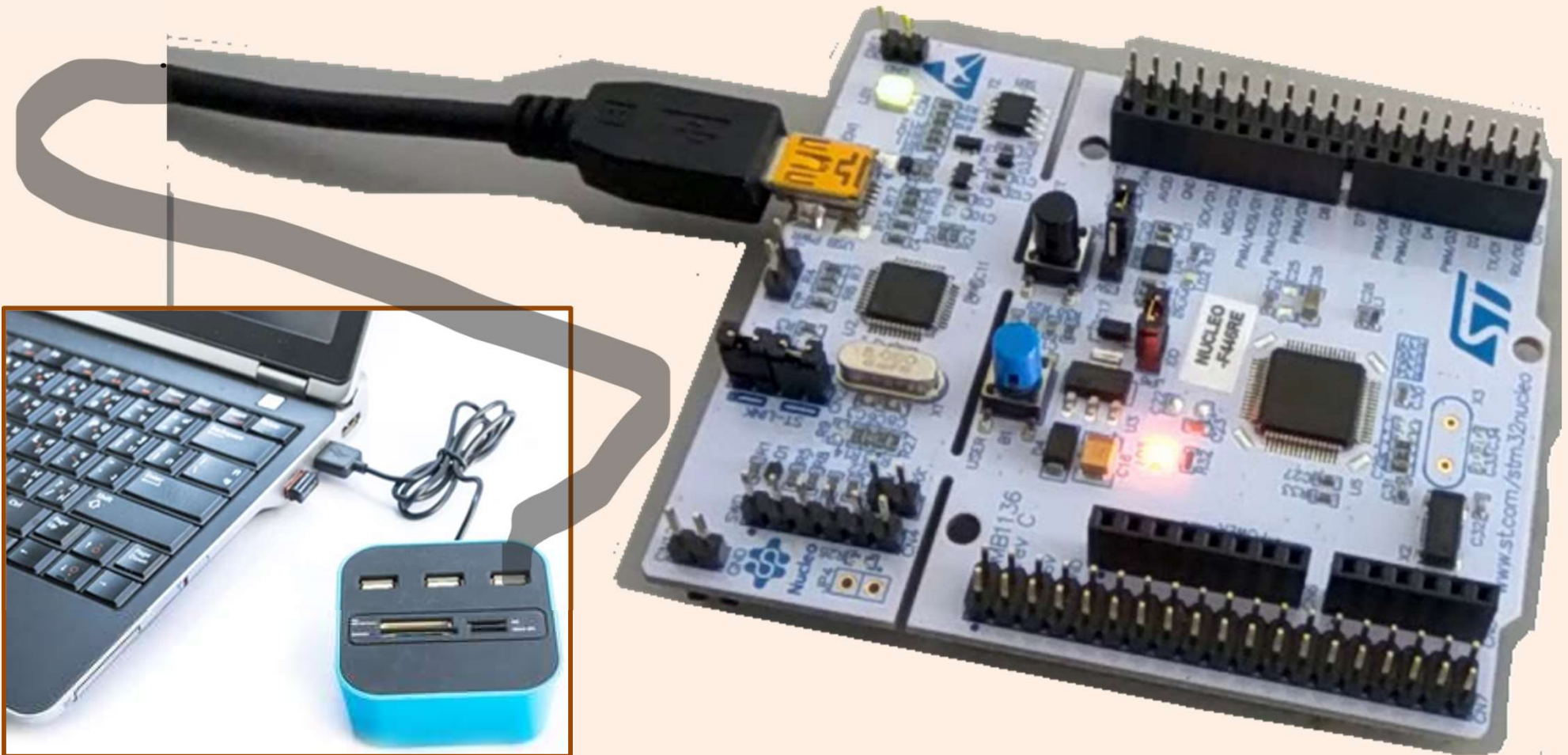
É só ligar a placa a um PC através de um cabo USB – Mini-USB



Get Started

Procedimento:

É só ligar a placa a um PC (se tiverem um Hub ficam mais protegidos contra acidentes)



Get Started

Procedimento:

Arrancar o IDE

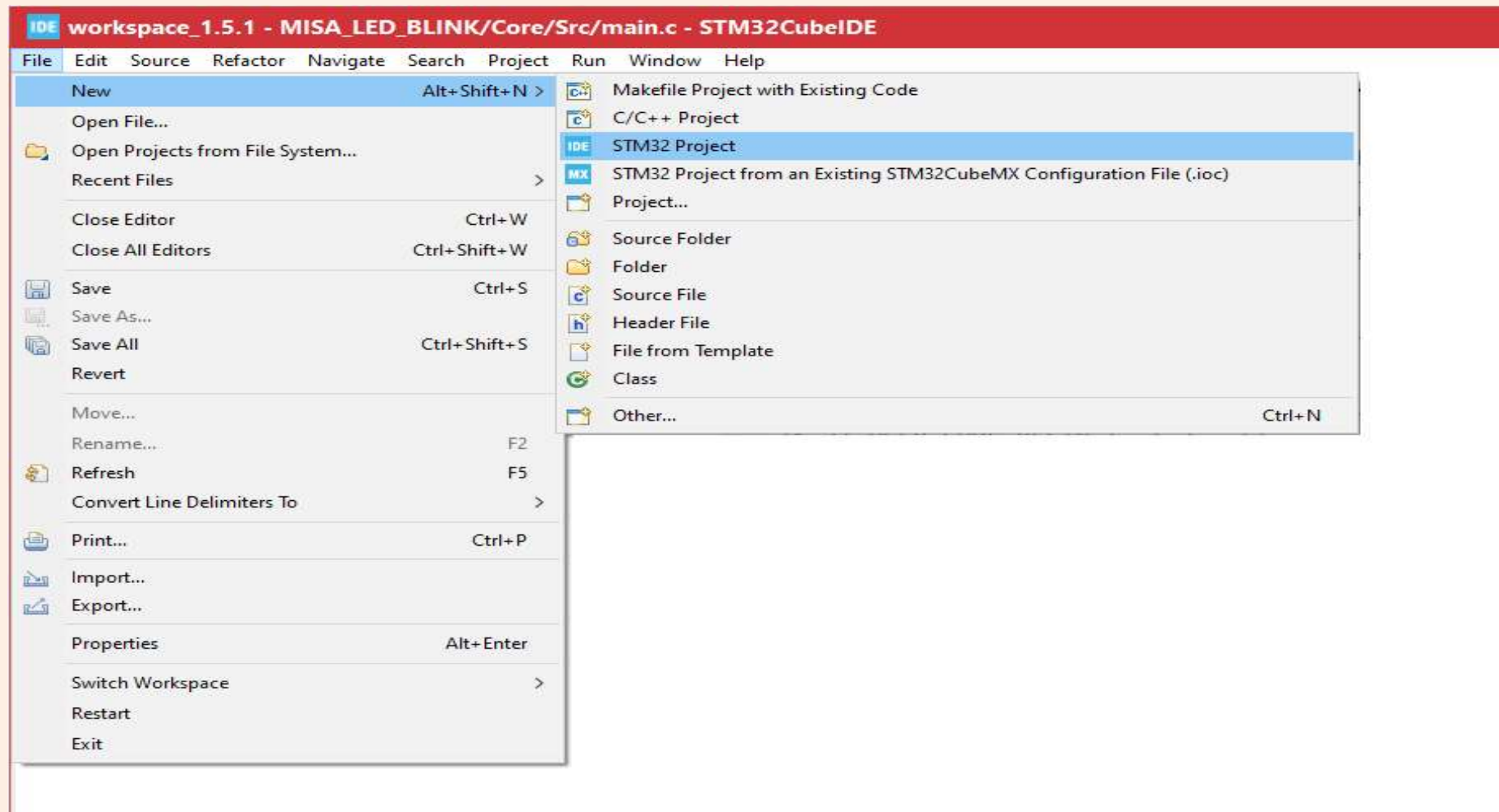
.



Get Started

Procedimento:

Criar um Projecto tipo “STM32”



Get Started

Procedimento:

Selecionar a placa: NUCLEO-F411RE

The screenshot displays the STM32F4 Board Selector interface. The top navigation bar includes tabs for MCU/MPU Selector, Board Selector (active), Example Selector, and Cross Selector. The left sidebar contains filters for Commercial Part Number, Vendor, Type, MCU/MPU Series, Other, and Peripheral. The main content area shows the STM32F4 Series with the NUCLEO-F411RE board selected. Below this, a table lists 158 boards, with the NUCLEO-F411RE board highlighted. The table columns are Overview, Commercial Part Number, Type, Marketing Status, Unit Price (US\$), and Mounted Device.

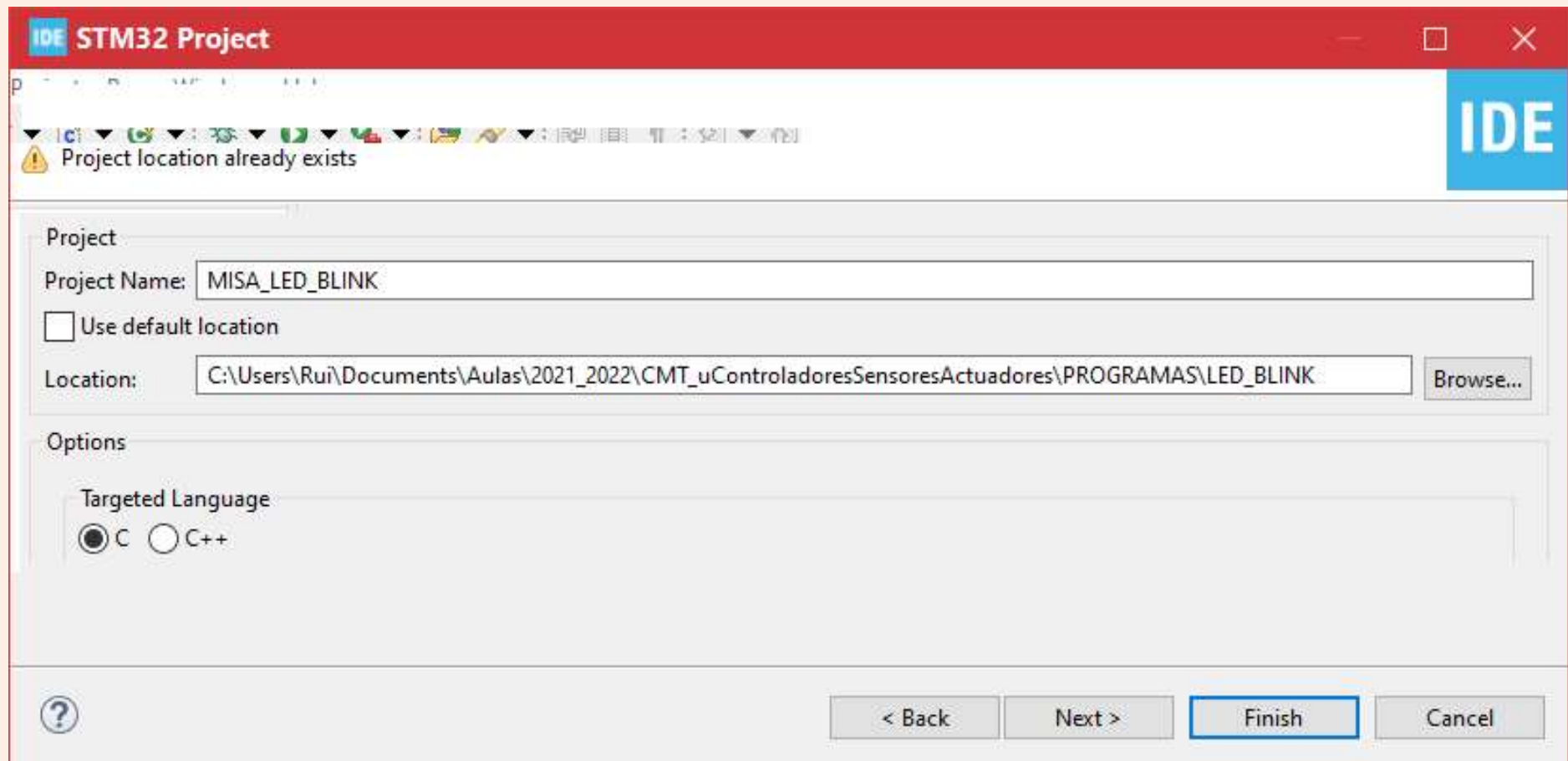
	Overview	Commercial Part ...	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		NUCLEO-F334R8	Nucleo-64	Active	10.32	STM32F334R8Tx
☆		NUCLEO-F401RE	Nucleo-64	Active	13.0	STM32F401RETx
☆		NUCLEO-F410RB	Nucleo-64	Active	13.0	STM32F410RBTx
☆		NUCLEO-F411RE	Nucleo-64	Active	13.0	STM32F411RETx
☆		NUCLEO-F412ZG	Nucleo-144	Active	19.0	STM32F412ZGTX
☆		NUCLEO-F413ZH	Nucleo-144	Active	19.0	STM32F413ZHTx

Navigation buttons at the bottom: Next >, Finish, Cancel.

Get Started

Procedimento:

Selecionar nome e local destino

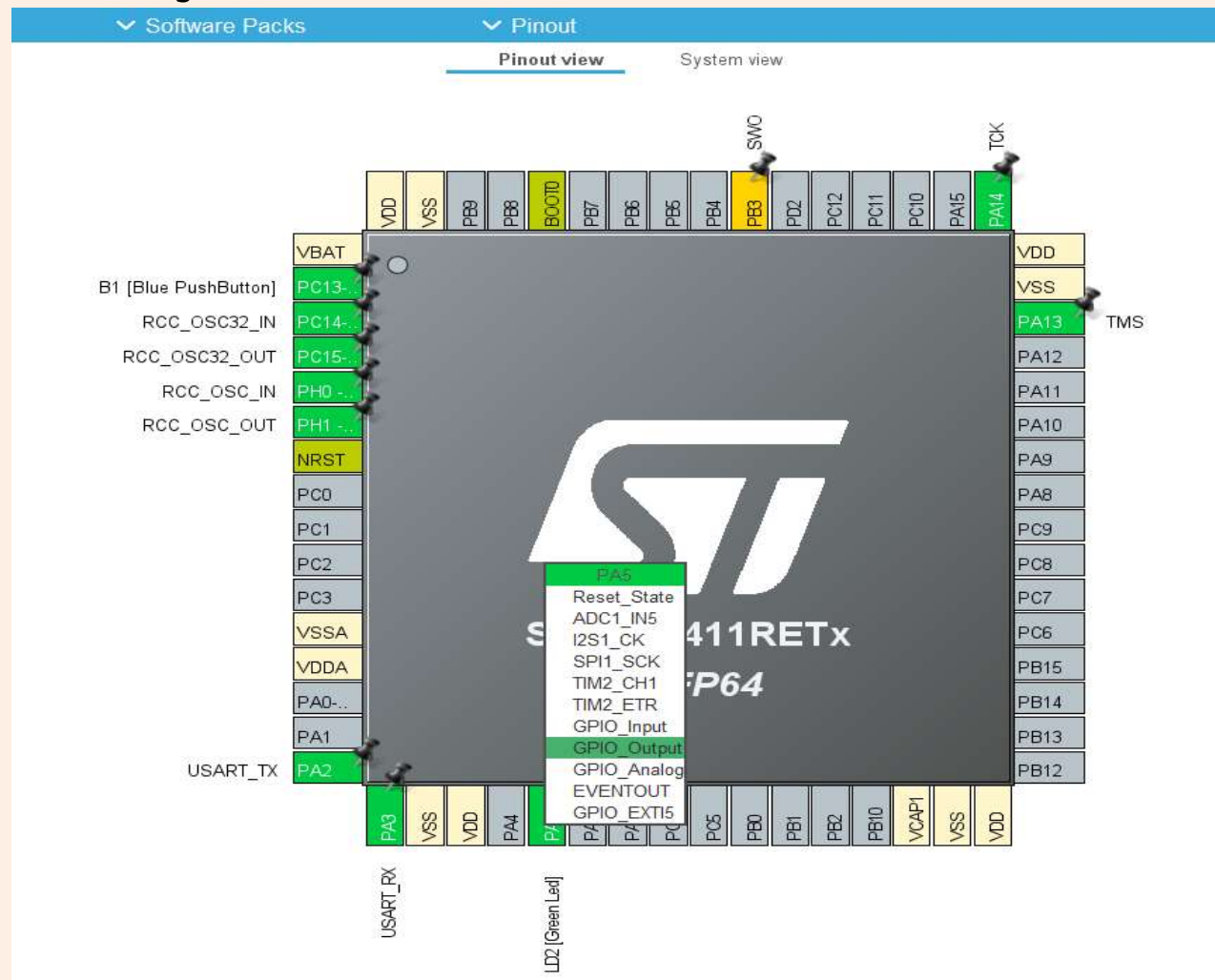


The screenshot shows the 'STM32 Project' dialog box in the IDE. The window title is 'IDE STM32 Project'. A warning icon and text 'Project location already exists' are visible in the top left. The 'Project' section contains a 'Project Name' field with the text 'MISA_LED_BLINK'. Below it is a checkbox labeled 'Use default location' which is unchecked. The 'Location' field contains the path 'C:\Users\Rui\Documents\Aulas\2021_2022\CMT_uControladoresSensoresActuadores\PROGRAMAS\LED_BLINK', followed by a 'Browse...' button. The 'Options' section has a 'Targeted Language' section with two radio buttons: 'C' (selected) and 'C++'. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

Get Started

Procedimento:

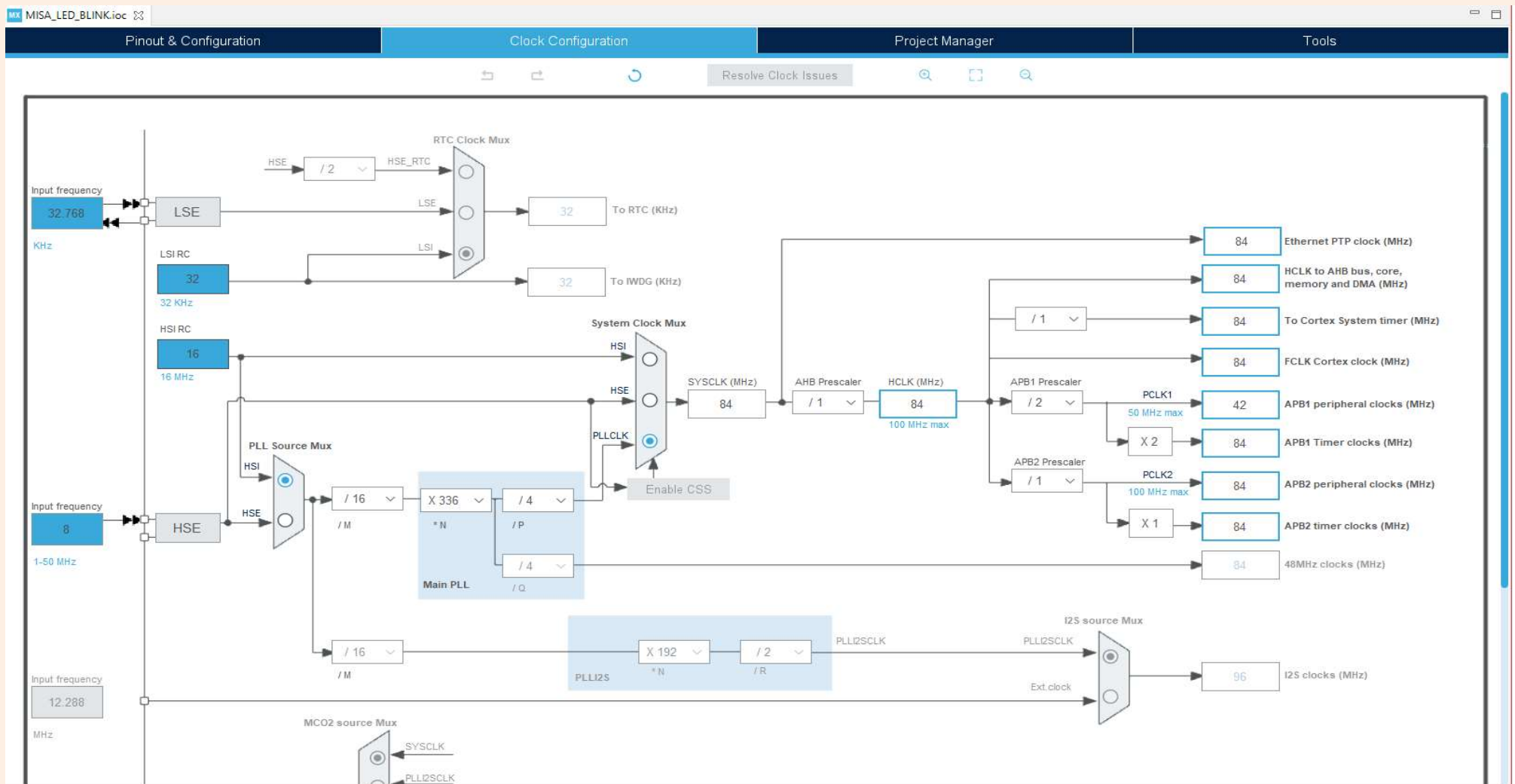
Selecionar pinos e funções



Get Started

Procedimento:

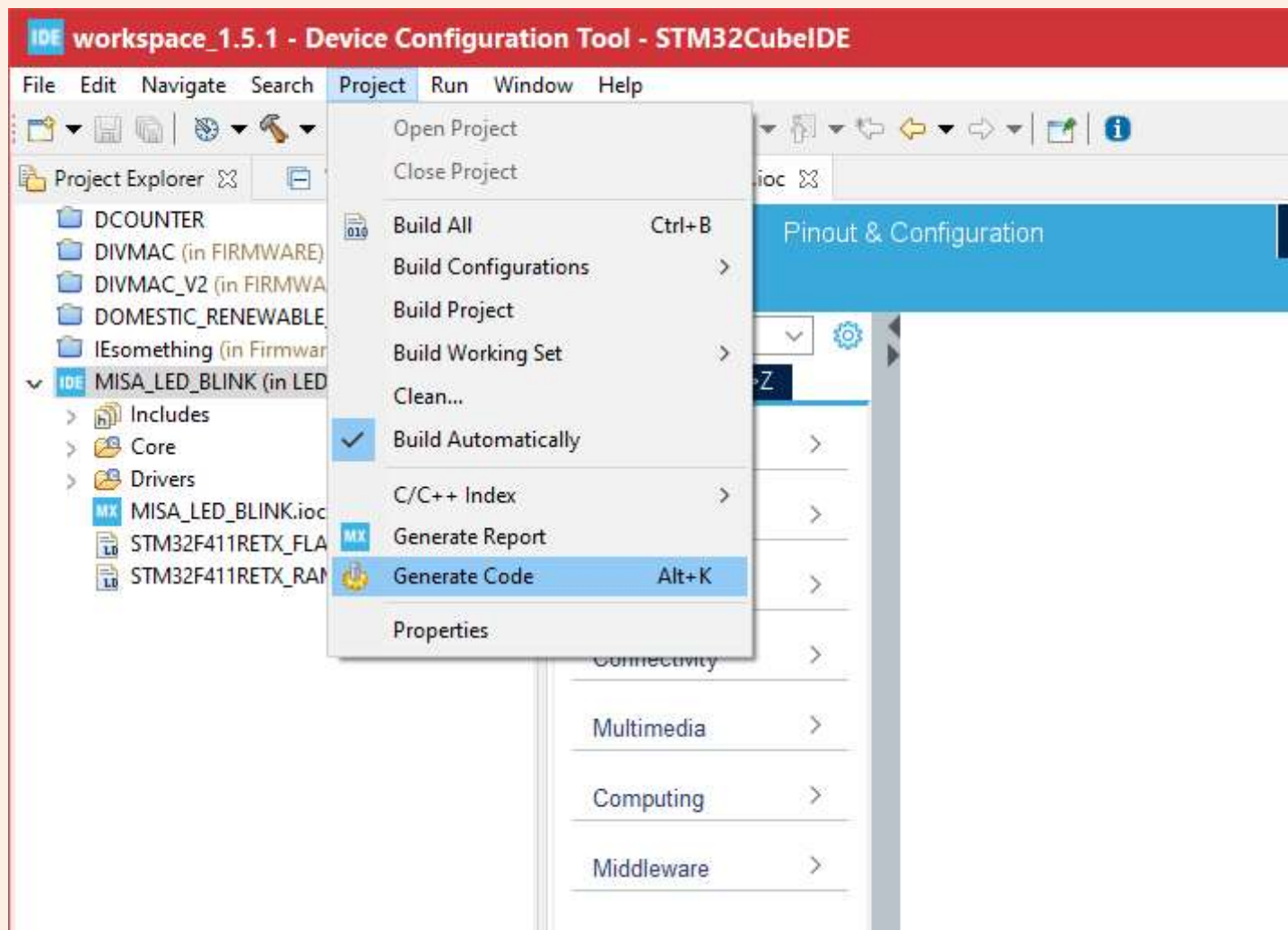
Selecionar oscilador e frequência de relógio



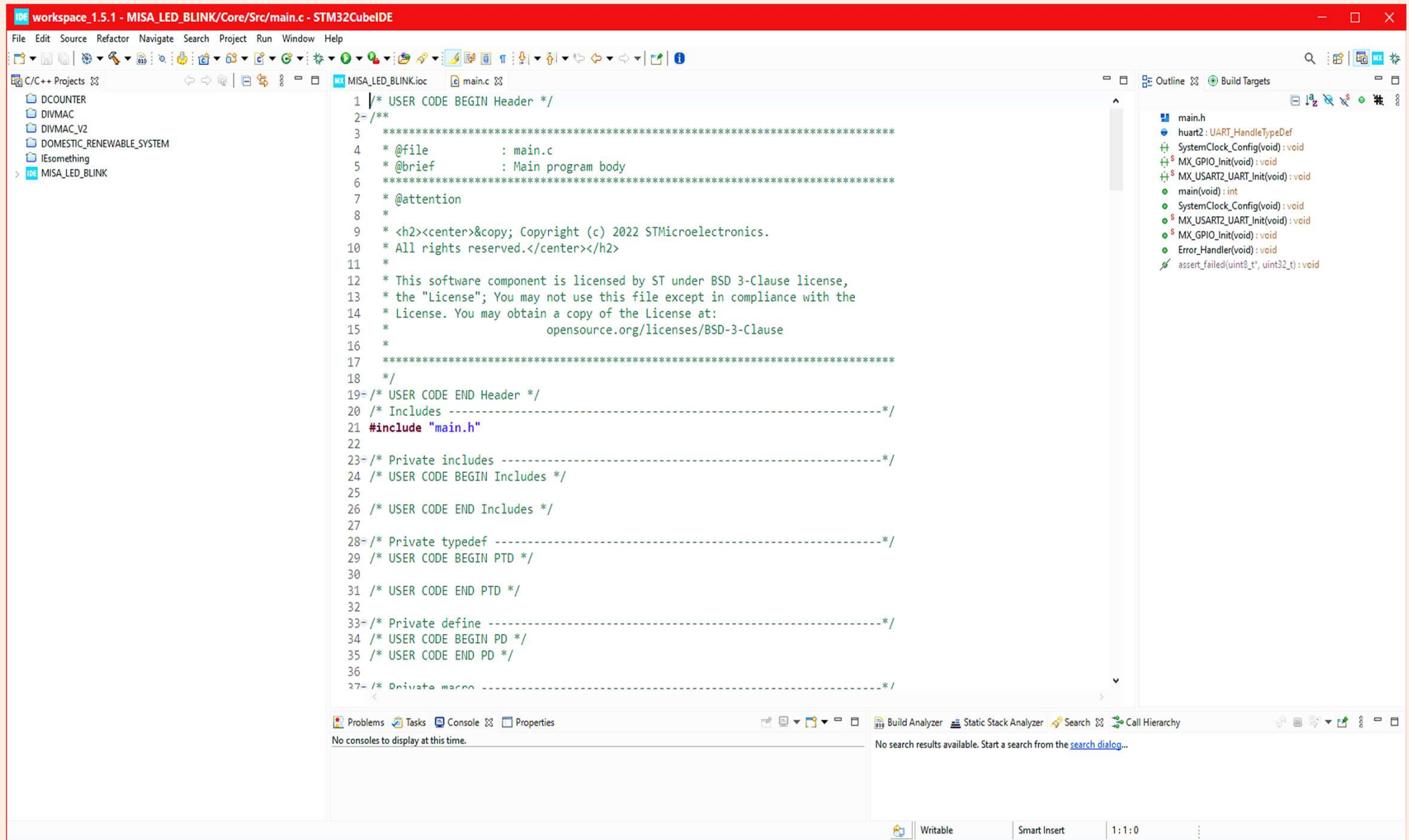
Get Started

Procedimento:

Gerar Código (automaticamente)



STM32CubeIDE



STM32CubeIDE

E agora é só escrever o Código...

```
82  SystemClock_Config();
83
84  /* USER CODE BEGIN SysInit */
85
86  /* USER CODE END SysInit */
87
88  /* Initialize all configured peripherals */
89  MX_GPIO_Init();
90  MX_USART2_UART_Init();
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      //HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); // Força o pino ao estado "1"
100     //HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); // Força o pino ao estado "0"
101     HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin); // Muda o estado do pino
102     HAL_Delay(1000); //Espera sem fazer nada durante 1000 milisegundos
103
104     /* USER CODE END WHILE */
105
106     /* USER CODE BEGIN 3 */
107 }
108 /* USER CODE END 3 */
109 }
110
111 /**
112  * @brief System Clock Configuration
113  * @retval None
114  */
115 void SystemClock_Config(void)
116 {
117     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
118     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
```

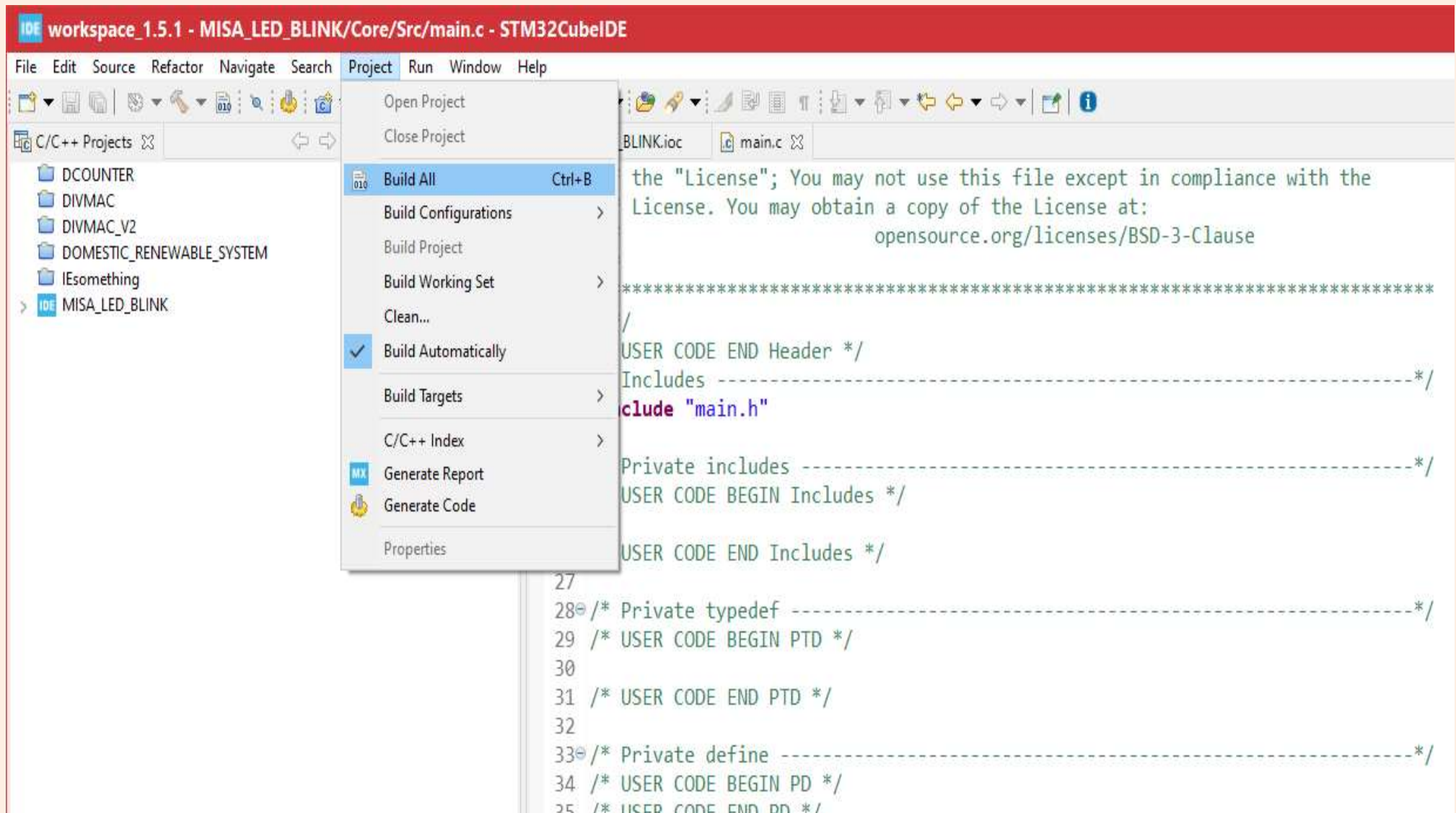
STM32CubeIDE

E agora é só escrever o Código...

```
82  SystemClock_Config();
83
84  /* USER CODE BEGIN SysInit */
85
86  /* USER CODE END SysInit */
87
88  /* Initialize all configured peripherals */
89  MX_GPIO_Init();
90  MX_USART2_UART_Init();
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
95  /* Infinite loop */
96  /* USER CODE BEGIN WHILE */
97  while (1)
98  {
99      //HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); // Força o pino ao estado "1"
100     //HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); // Força o pino ao estado "0"
101     HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin); // Muda o estado do pino
102     HAL_Delay(1000); //Espera sem fazer nada durante 1000 milisegundos
103
104     /* USER CODE END WHILE */
105
106     /* USER CODE BEGIN 3 */
107 }
108 /* USER CODE END 3 */
109 }
110
111 /**
112  * @brief System Clock Configuration
113  * @retval None
114  */
115 void SystemClock_Config(void)
116 {
117     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
118     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
```

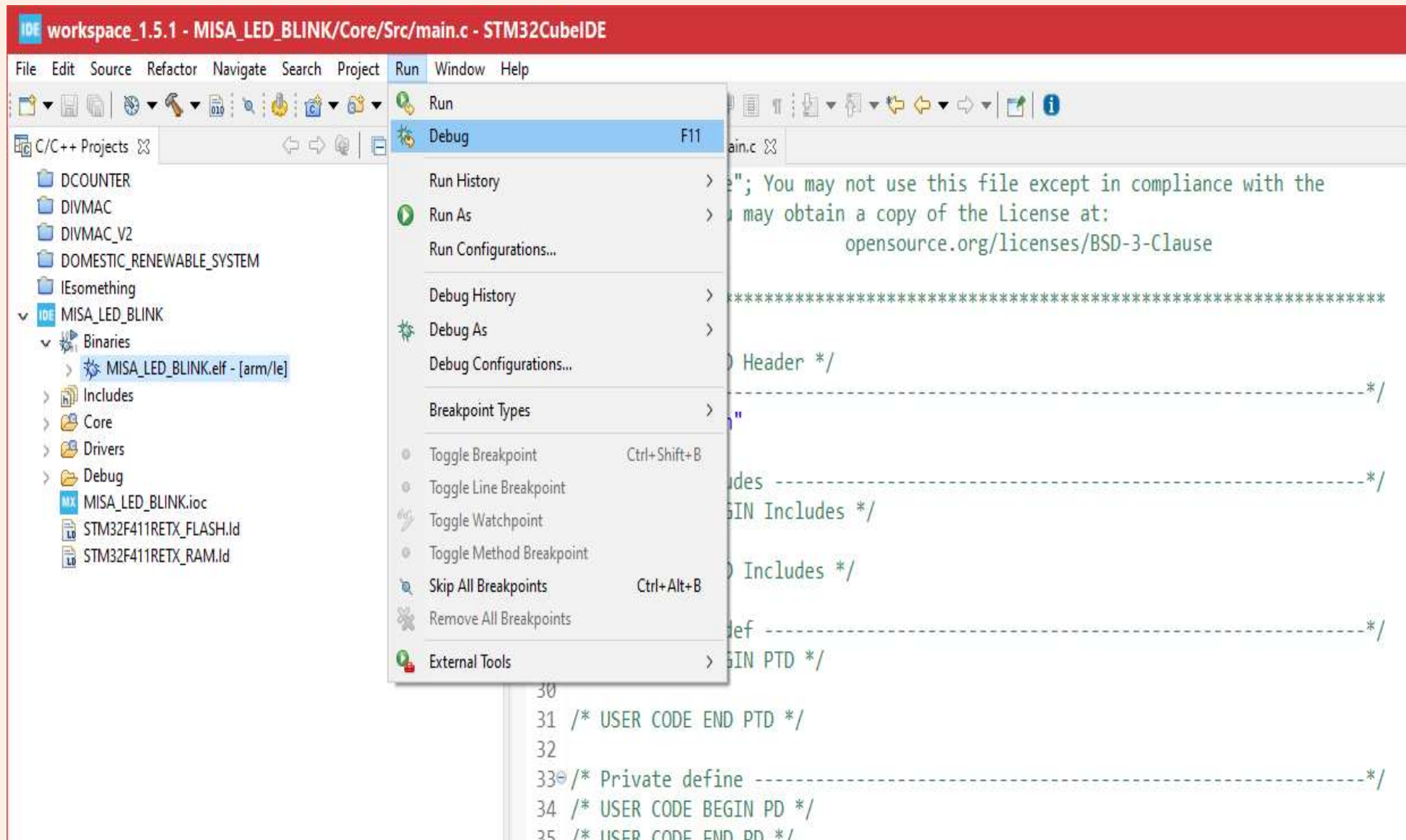

STM32CubeIDE

Compilar o Código e Criar executável (selecionar **Build All**)



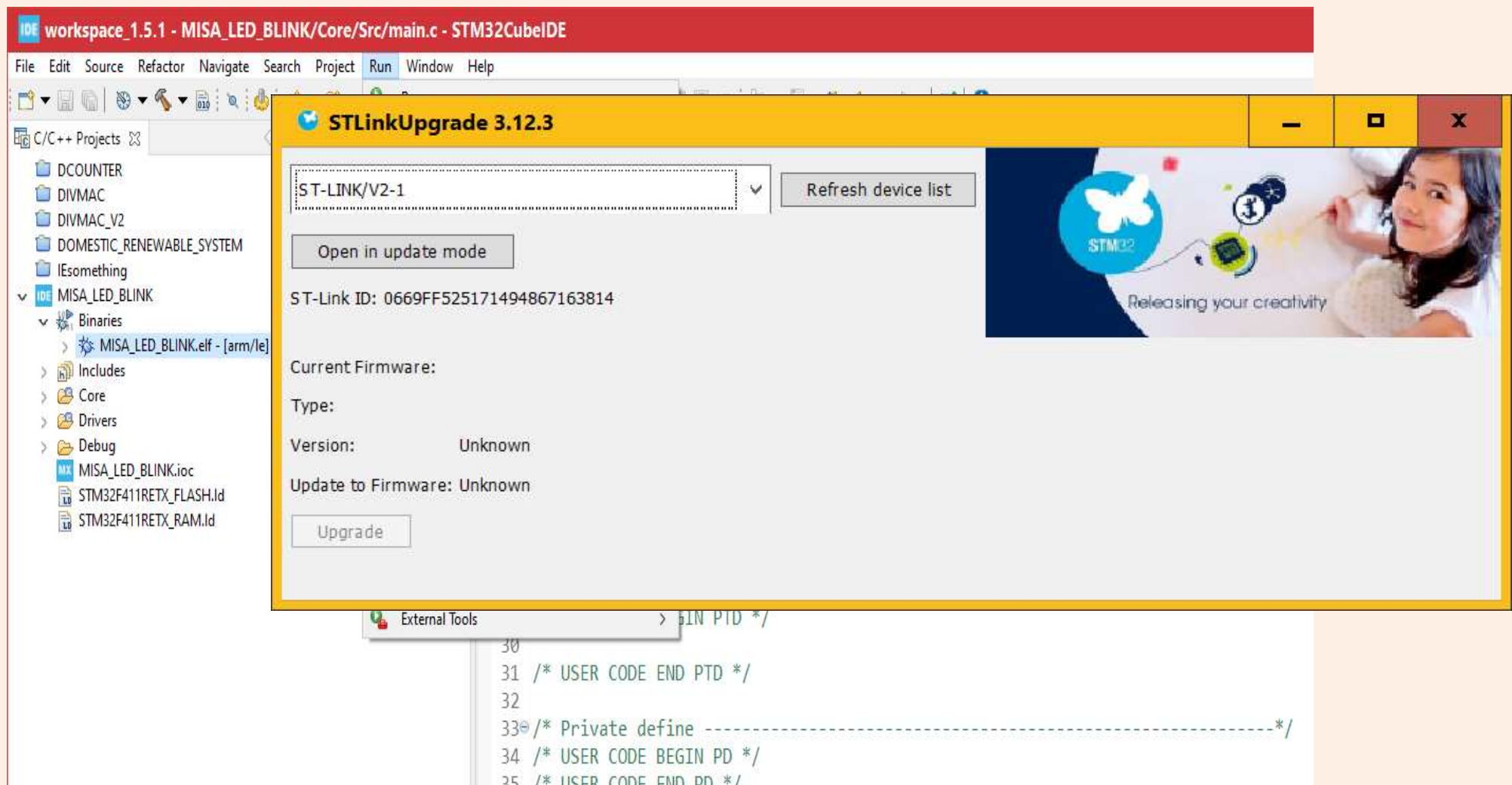
STM32CubeIDE

Para enviar o executável para o microcontrolador fazer **Debug**.



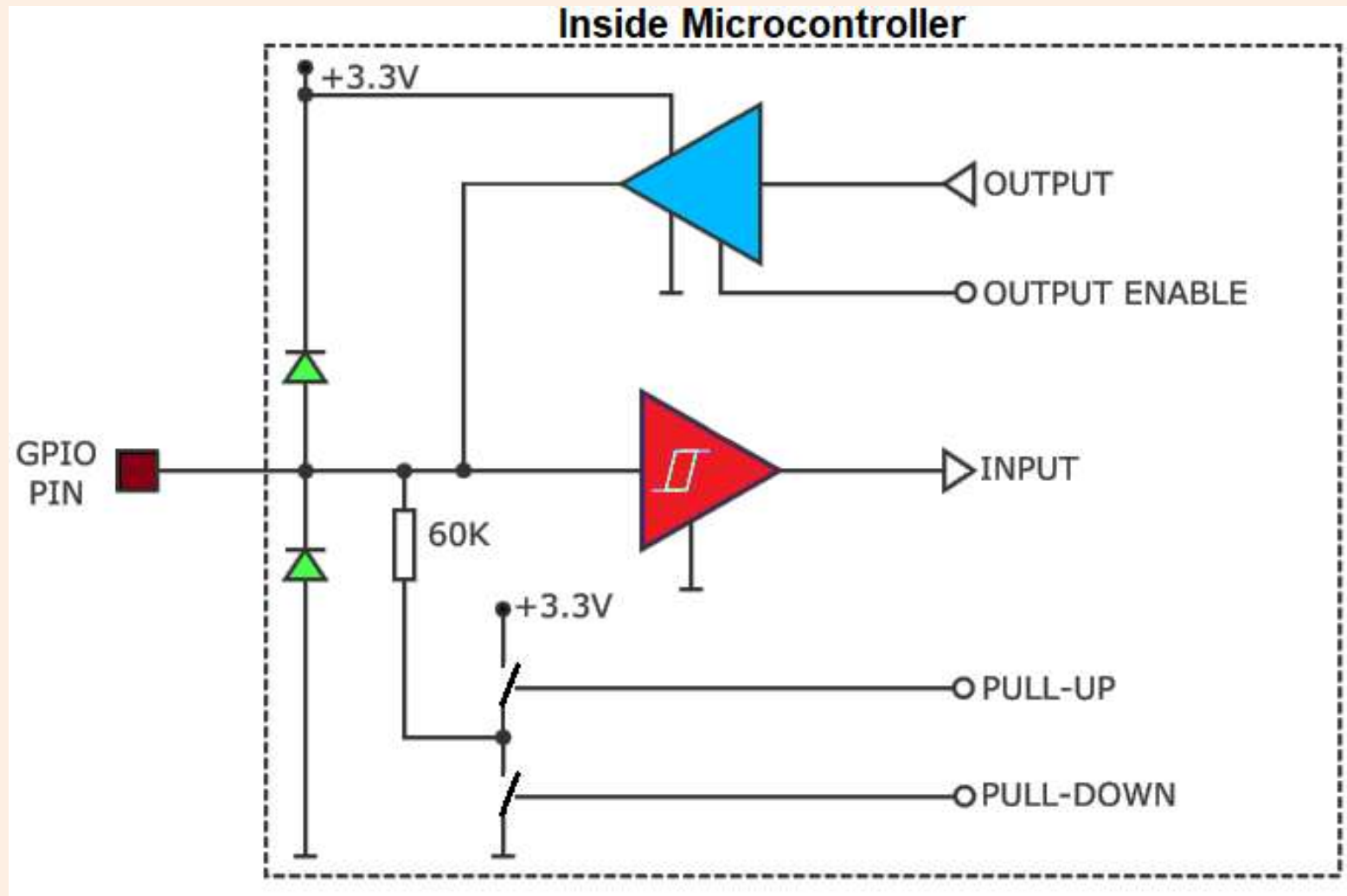
STM32CubeIDE

Pode acontecer que o firmware da placa de desenvolvimento já seja antigo, pelo que possa necessitar de ser actualizado. Se aparecer a janela da figura, carreguem em “Open in update mode” e depois no botão “Update”



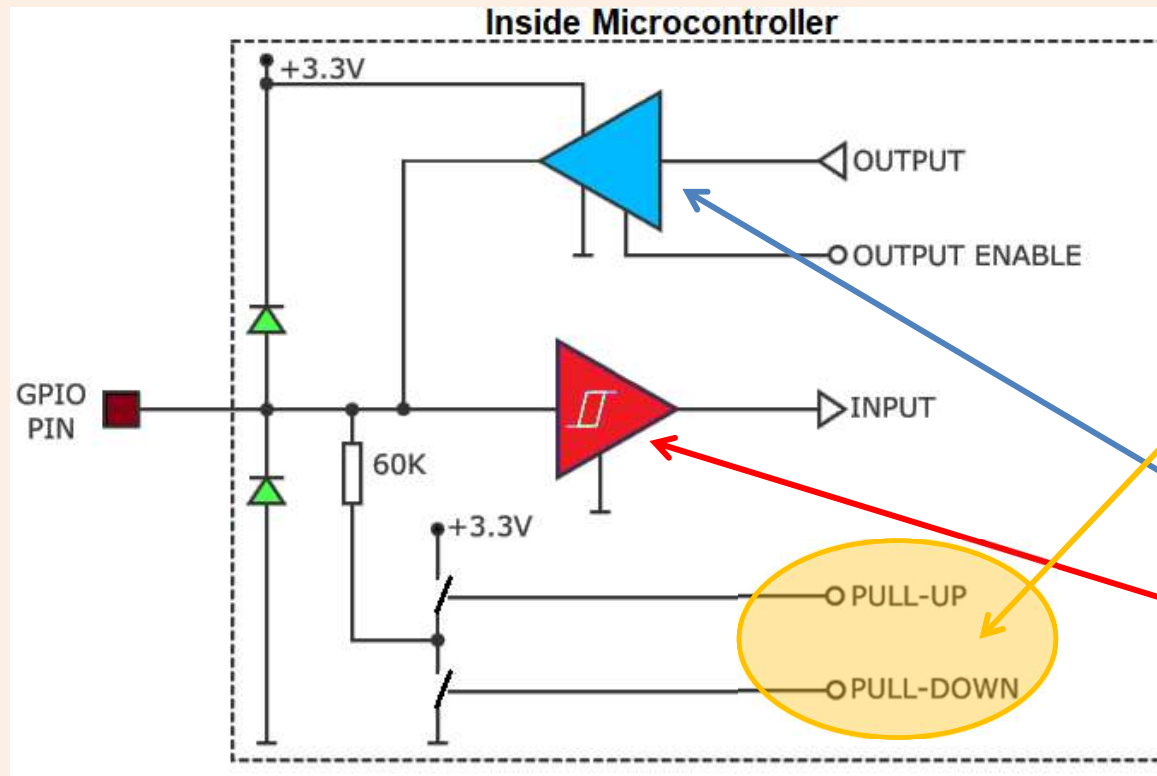
STM32CubeIDE

GPIO pin – General Purpose Input/Output pin



STM32CubeIDE

GPIO pin – General Purpose Input/Output pin



Pull-Up/Pull-Down control

```
HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);  
  
blue_button_state= HAL_GPIO_ReadPin(BLUE_BUTTON_GPIO_Port, BLUE_BUTTON_Pin);
```