

BE « Réalisations Systèmes » BE Microcontrôleur STM32

ARM architecture

ARM/CORTEX processeurs

STM32 microcontrôleur

NUCLEO cartes de développement

EDI Environnement de Développement Intégré

BE Réalisations Systèmes BE « Microcontrôleur STM32 »

Toute la documentation est disponible sur mon site web perso :
<http://thierryperisse.free.fr/>
 Onglet : Cours/TDs/TPs → [NUCLEOSTM32](#).

M1 SME (2 groupes) (G1 et G2) (alternant G2 et non alternant G1)
 BE 32h (soit 8 séances de 4h)
 Horaires : Matin 7h45-11h45 (4h) et Après midi 13h-17h (4h)
 Lieu : Salle G45 Bât3A (voire en H0 si pb)

Sauvegarde et mise à jour de l'avancement du BE :

sur PC **STM32_2024_Gxy**
 → mes documents
 → **STM32_2024_Gxy** x n° du groupe et y n° du binôme

Gestion du projet (**Github**) **STM32_2024_Gxy**

BE Microcontrôleur STM32

Piste pour l'organisation du répertoire **STM32_2024_Gxy** x n° du groupe et y n° du binôme

STM32_2024_Gxy	/BE	/NUCLEOL152	/CubeIDE	/ noms_projets ...
	/TPsBase	/NUCLEOL152	/CubeIDE	/ noms_projets ...
	/Docs	/cartes		
		/capteurs	/capteur1 ...	

BE Microcontrôleur STM32

Timing approximatif du BE (32h) :

1. Introduction au microcontrôleurs STM32 : (~2h30)
2. Réalisation des Projets de base : (~16h)
3. Réalisation d'un BE utilisant l'IDE STM32CubeIDE : (~16h)

1- Introduction au microcontrôleurs STM32

- ✓ Présentations des documents STM32 utilisés : (~ 30mn)
 - Présentation **microcontrôleurs ARM Cortex STM32**
 - Présentation **cartes de développement NUCLEO**
 - Présentation **Environnements de Développement Intégré IDE**
- ✓ Réalisations mini projets : (~ 2h)
 - Mise en œuvre Projets simples sur l'IDE « STM32CubeIDE »



BE Microcontrôleur STM32

2- Réalisation des Projets de base : (~16h)

- STM32 + Capteur + Afficheur LCD I2C
- Analyser une fonction existante « Commenter » où
- Créer une nouvelle fonction « Commenter ».
- **Notation Projet de base :**
 - *Etude doc constructeur* *Datasheet (Nucléo, capteurs, ...)*
 - *Câblage* *Fritzing (logiciel gratuit)*
 - *Visualisations oscilloscope* *Trame, bus, ... (// avec les docs)*
 - *Validations expérimentales* *(Projets de base complet)*
 - *Gestion du projet* *(Github)*
 - *Rapport (pdf, docx, Notion, ...)* *(~ 10 pages + annexes)*



BE Microcontrôleur STM32

3- Réalisation d'un BE utilisant l'IDE STM32CubeIDE : (~16h)

- Choix d'un thème de BE (en fonction des cartes et composants disponibles et en accord avec l'encadrant)
- Choix d'un cahier des charges (en accord avec l'encadrant)
- *Analyse fonctionnelle (décomposition en fonctions principales (FPx), (FSx)).*
- *Algorithme / Programme / Commentaires / ...*
- Test sur NUCLEO
- **Notation BE :**
 - *Etude doc constructeur* *Datasheet (Nucléo, capteurs, ...)*
 - *Câblage* *Fritzing (logiciel gratuit)*
 - *Visualisations oscilloscope* *Trame, bus, ... (// avec les docs)*
 - *Validations expérimentales* *(BE)*
 - *Gestion du BE* *(Github)*
 - *Rapport (pdf, docx, Notion, ...)* *(~ 20 pages + annexes)*
 - *Diapos projets (pptx)* *(environ 10)*
 - *Vidéo projets* *(< 2mn)*



1- Introduction aux microcontrôleur STM32

1 - Les microcontrôleurs ARM Cortex STM32 32 bits :

- Architecture ARM :
- Processeurs ARM-CORTEX :
- Microcontrôleurs STM32 :

2 - Les Cartes Nucléo STM32

3 - Cartes NUCLEO :

- 3.1 Cartes NUCLEO disponibles
- 3.2 Carte STM32L152RE (caractéristiques et structure interne)

4 - Outils de développement logiciel STM32

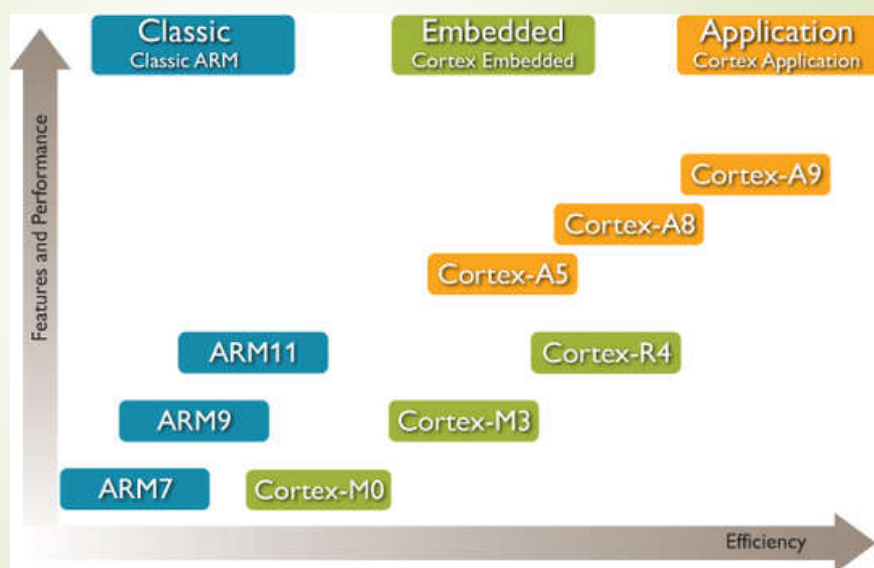
- 4.1 - EDI STM32CubeIDE
- 4.2 - STM32CubeMX
- 4.3 - STM32 Cube

5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE :

- 5.1 - Projets simple sous STM32CubeIDE : **LED**
- 5.2 - Projets simple sous STM32CubeIDE : **BP_LED**
- 5.3 - Projet simple avec interruption sous STM32CubeIDE **BP_LED_Interrupt**

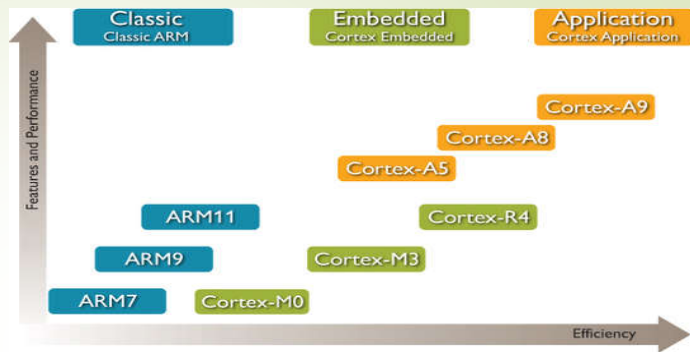
8

Processeurs ARM-CORTEX



9

Processeurs ARM-CORTEX



Processeurs ARM dominant dans l'électronique embarquée (téléphones, tablettes, ...).

Points forts : Architecture simple et Faible consommation .

Les processeurs développés par la société ARM ne sont pas des composants autonomes mais seulement des IP (Intellectual Properties, *Propriétés Intellectuelles*).

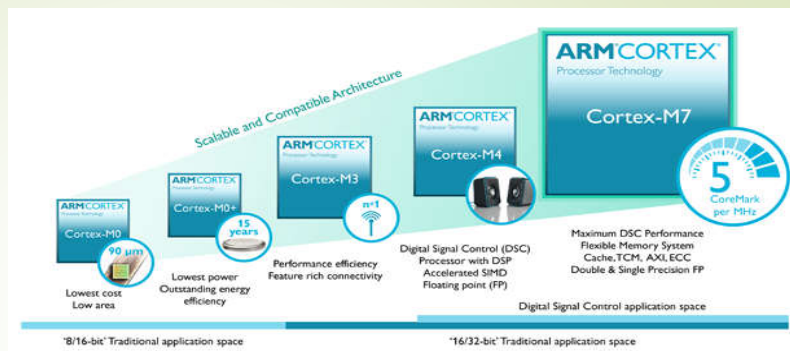
Les SoC CORTEX sont équipés de processeurs ARM

Le Cortex-M3 est un bon compromis entre Efficacité / Performances / Caractéristiques

Ce sont les fabricants, tels que STMicroelectronics, qui réalisent des composants autonomes tels que les microcontrôleurs STM32 en ajoutant au processeur ARM-Cortex-M de la mémoire et des périphériques.

10

Processeurs ARM-CORTEX



Les SoC CORTEX (systèmes on chip CPU + GPU + DSP + mémoires + périphériques) sont équipés de processeurs ARM

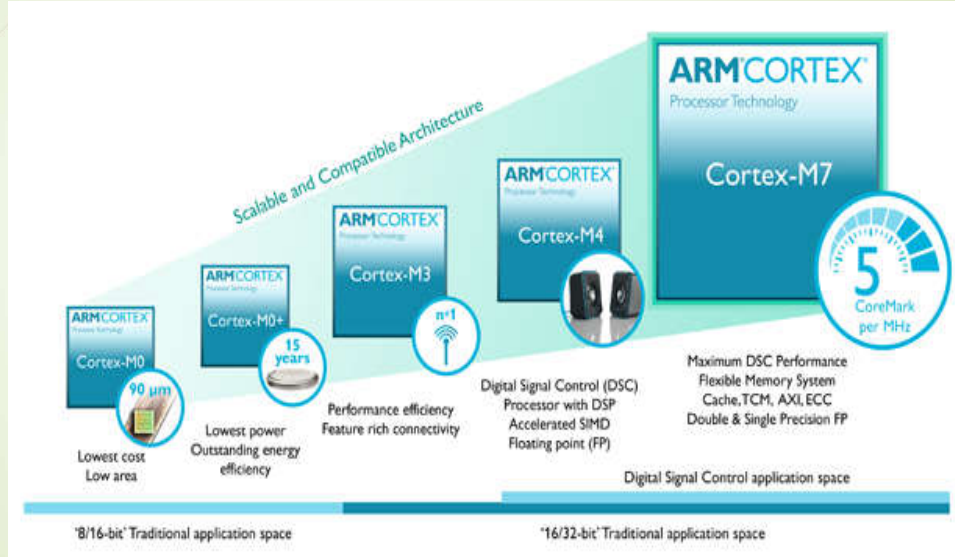
- CORTEX A, pour Application (plutôt pour l'informatique)
- CORTEX R, pour Real-Time (pour les applications temps réel)
- **CORTEX M, pour eMbedded (eMbarqué) (pour les applications embarquées)**

Cortex-M est un cœur complet devenu un standard dans le domaine des processeurs. Ils se déclinent en différents modèles : Cortex-M0/M0+ M1 M3 M4 et plus récemment le M7.

TP1

11

Processeurs ARM-CORTEX



12

Microcontrôleurs STM32 ARM-CORTEX



STM32 32-bit ARM Cortex MCUs

Category	Model	CoreMark	Freq	DMIPS
High performance	STM32 F2	398	120 MHz	150 DMIPS
	STM32 F4	608	180 MHz	225 DMIPS
	STM32 F7	1 062	210 MHz	462 DMIPS
Mainstream	STM32 G0	142	64 MHz	56 DMIPS*
	STM32 F0	106	48 MHz	38 DMIPS
	STM32 F1	177	72 MHz	61 DMIPS
	STM32 F3	245	72 MHz	90 DMIPS*
	STM32 L0	75	32 MHz	26 DMIPS
	STM32 L1	93	32 MHz	33 DMIPS
Ultra-low-power	STM32 L4	273	80 MHz	100 DMIPS
	STM32 L4+	408	120 MHz	150 DMIPS
	STM32 L5	427	110 MHz	165 DMIPS
	STM32 WB	216	64 MHz	80 DMIPS
	STM32 WB	216	32 MHz	40 DMIPS

Legend: *from CCM-SRAM ● Cortex-M0+ Radio Co-processor

STM32 Ecosystem

- Software tools: STM32CubeMX, Integrated Development Environments (IDE), STM Studio, Monitoring tool
- Embedded software: STM32Cube MCU Packages, STM32Cube Expansion Packages
- Hardware tools: STM32 Nucleo development boards, Discovery kits, Evaluation boards, STM32 Nucleo expansion boards, ST-LINK in-circuit debugger/programmer

Join the STM32 Community! community.st.com/stm32

Diapositive 11

TP1 Thierry PERISSE; 20/01/2019

13

Applications de hautes performances

Applications de moyennes dimensions

Applications de très basses consommations

Applications sans fils

High Performance

STM32F2 398 CoreMark 120 MHz Cortex-M3	STM32F4 608 CoreMark 180 MHz Cortex-M4	STM32F7 1082 CoreMark 216 MHz Cortex-M7	STM32H7 Up to 3224 CoreMark Up to 550 MHz Cortex-M7 240 MHz Cortex-M4
---	---	--	--

Mainstream

STM32G0 142 CoreMark 64 MHz Cortex-M0+	STM32G4 569 CoreMark 170 MHz Cortex-M4
STM32G0 114 CoreMark 48 MHz Cortex-M0+	STM32F0 106 CoreMark 48 MHz Cortex-M0
STM32F1 177 CoreMark 72 MHz Cortex-M3	STM32F3 245 CoreMark 72 MHz Cortex-M4

● Optimized for mixed-signal applications

Ultra-low-power

STM32L0 75 CoreMark 32 MHz Cortex-M0+	STM32L1 93 CoreMark 32 MHz Cortex-M3	STM32L4 273 CoreMark 80 MHz Cortex-M4	STM32L5 443 CoreMark 110 MHz Cortex-M33
--	---	--	--

Wireless

STM32WL 162 CoreMark 48 MHz Cortex-M4 48 MHz Cortex-M0+	STM32WB 216 CoreMark 64 MHz Cortex-M4 32 MHz Cortex-M0+
---	---

● Cortex-M0+
Radio co-processor

14

STM32 High performance MCUs 32-bit Arm® Cortex®-M

- STM32H7**
- STM32F7**
- STM32F4**
- STM32F2**
- STM32F4 Access**

- Arm® Cortex®-M7 + Arm® Cortex®-M4 FPU at 480 MHz – 1327 DMIPS and up to 550 MHz - 1177 DMIPS on single core Arm® Cortex®-M7
- From 512 Kbytes to 2 Mbytes of Flash memory
- Very high performance with embedded Flash memory and external memories (including Dual-mode Quad-SPI interface and Dual Octo-SPI interface)

- Arm® Cortex®-M7 + FPU at 216 MHz – 462 DMIPS
- From 256 Kbytes to 2 Mbytes of Flash memory
- Very high performance with embedded Flash memory and external memories (including Dual Octo-SPI interface)

- Arm® Cortex®-M4 + FPU up to 180 MHz – 225 DMIPS
- From 256 Kbytes to 2 Mbytes of Flash memory
- Foundation or Advanced lines: Select the right F4 for your needs

- Arm® Cortex®-M3 at 120 MHz – 150 DMIPS
- From 128 Kbytes to 1 Mbyte of Flash memory
- Foundation lines for performance and connectivity

- The entry-level microcontrollers of the STM32F4 series!
- From 64 Kbytes to 1.5 Mbytes of Flash memory
- Cost-effective and outstanding power efficiency (Dynamic Efficiency™)
- Batch Acquisition Mode (BAM) for Always-on data collection and dynamic power scaling

Series	CoreMark score	DMIPS	Frequency (MHz)
STM32F4 Access lines	339	~125	~120
Foundation STM32F2 Series	398	~150	~120
Foundation STM32F4 Series	608	~225	~180
Advanced STM32F4 Series	608	~225	~180
Value line STM32F7 Series	1082	~462	~216
Advanced STM32F7 Series	1082	~462	~216
Foundation STM32H7 Series	3224	~1327	~480
Value line STM32H7 Series	3224	~1327	~480

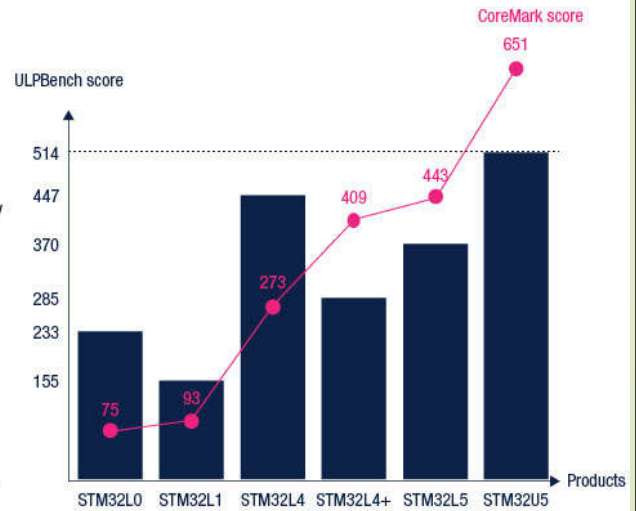
15



STM32 Ultra-low power MCUs 32-bit Arm® Cortex®-M



STM32U5	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M33 + FPU at 160 MHz From 1 to 2 Mbytes of Flash memory Lowest power mode + RAM + RTC: 0.35 µA
STM32L5	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M33 + FPU at 110 MHz From 256 to 512 Kbytes of Flash memory Lowest power mode + RAM + RTC: 0.35 µA
STM32L4+	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M4 + FPU at 120 MHz From 512 Kbytes up to 2 Mbytes of Flash memory Lowest power mode + RAM + RTC: 0.39 µA
STM32L4	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M4 + FPU at 80 MHz From 64 Kbytes to 1 Mbyte of Flash memory Lowest power mode + RAM + RTC: 0.34 µA
STM32L1	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M3 at 32 MHz From 32 to 512 Kbytes of Flash memory Lowest power mode + RAM + RTC: 1.2 µA
STM32L0	<ul style="list-style-type: none"> 32-bit Arm® Cortex®-M0+ at 32 MHz From 8 to 192 Kbytes of Flash memory Lowest power mode + RAM + RTC: 0.67 µA
STM8L	<ul style="list-style-type: none"> 8-bit STM8 core at 16 MHz From 2 to 64 Kbytes of Flash memory Lowest Halt mode: 0.3 µA



16

Cartes NUCLEO

Figure 1. Nucleo development tools

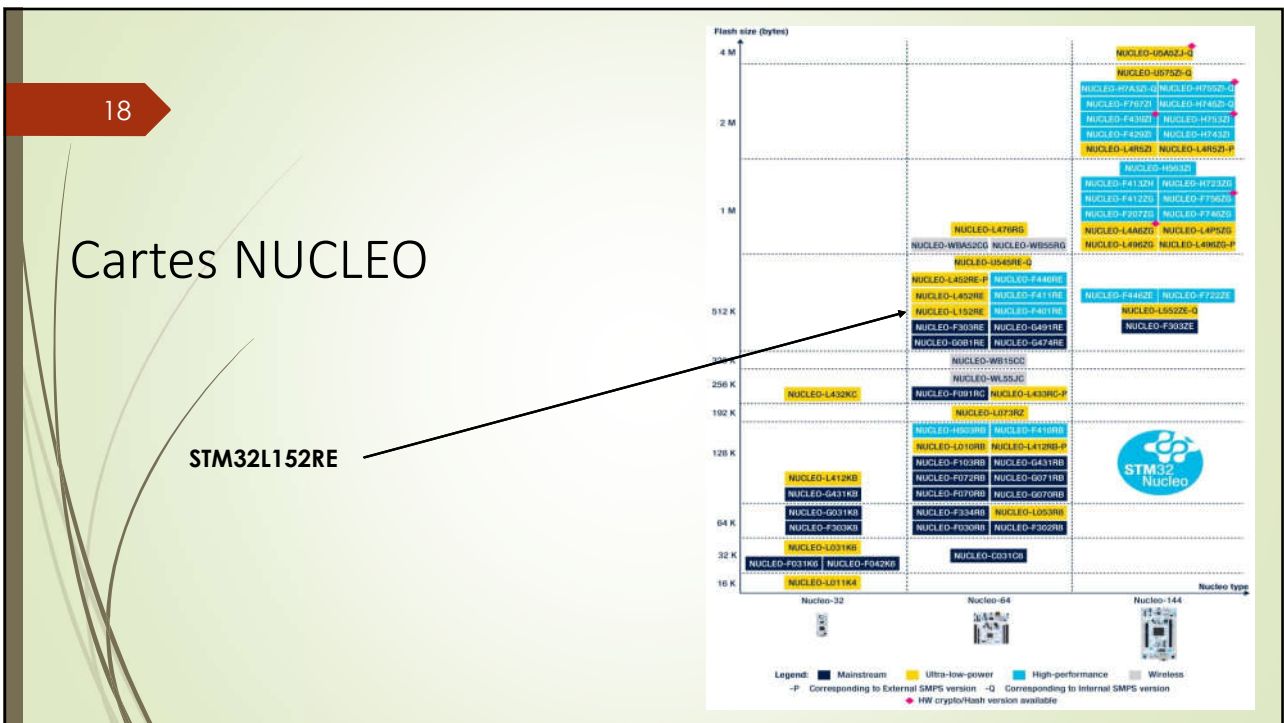
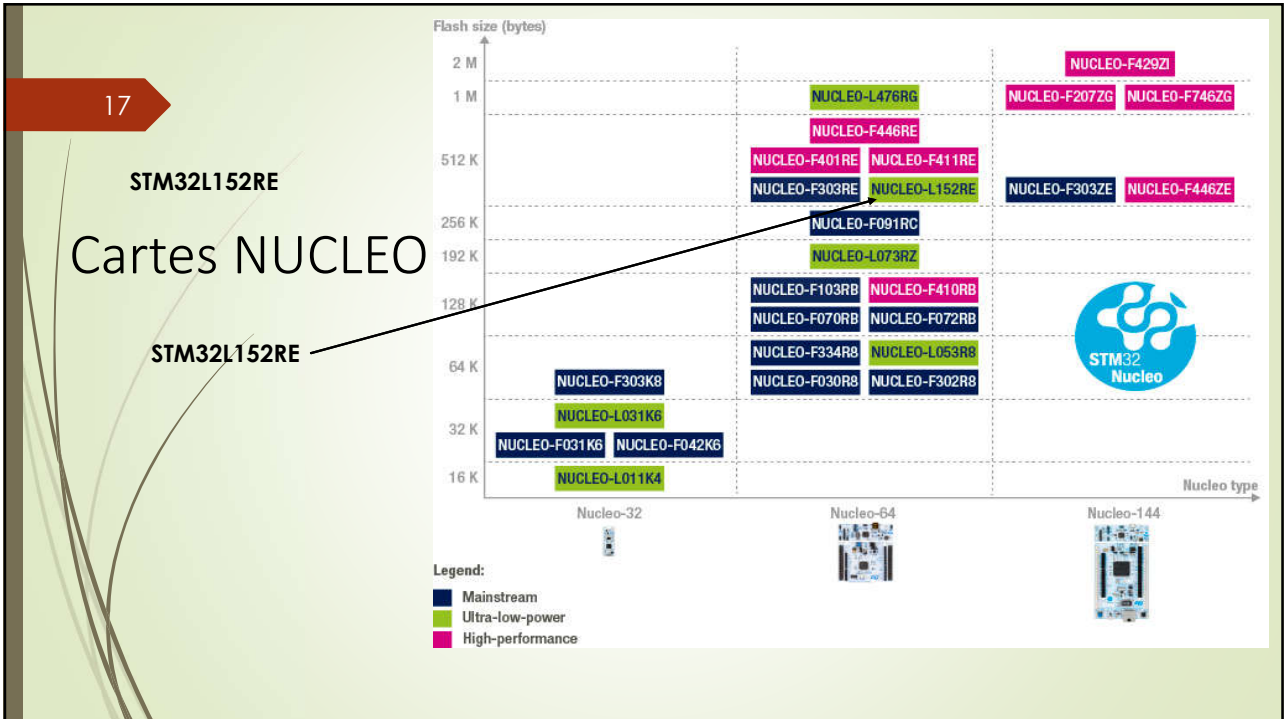


NUCLEO 32, 64, 144

Tous les microcontrôleurs STM32 ont été implantés sur des cartes NUCLEO

Cartes NUCLEO-64 avec connectique ARDUINO et ST

Cartes NUCLEO-32 pour le prototypage sur support DIP



19

Cartes NUCLEO disponibles

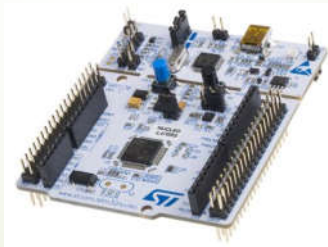


Nucleo-144 STM32F303ZE
STM32 with LQFP144 package

Informations RS

- ⊙ Code commande RS 917-3762
- ⊙ Prix HT 17€
- ⊙ Référence fabricant NUCLEO-F303ZE
- ⊙ Fabricant [STMicroelectronics](#)

Micro USB B mâle vers USB A mâle



Nucleo-64 STM32L476RG
STM32 with LQFP64 package

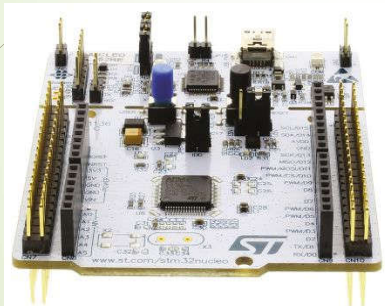
Informations RS

- ⊙ Code commande RS 896-7721
- ⊙ Prix HT 12€
- ⊙ Référence fabricant NUCLEO-L476RG
- ⊙ Fabricant [STMicroelectronics](#)

Mini USB B mâle vers USB A mâle

20

Carte NUCLEO-64 STM32L152RE



Nucleo-64 STM32L152RE
STM32 with LQFP64 package

Informations RS

- ⊙ Code commande RS 802-9428
- ⊙ Prix HT 12€
- ⊙ Référence fabricant NUCLEO-L152RE
- ⊙ Fabricant [STMicroelectronics](#)

Mini USB B mâle vers USB A mâle

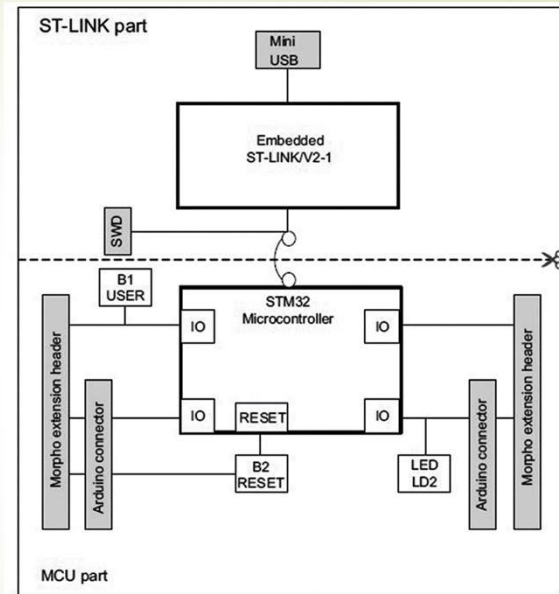
- Connecteurs Arduino Uno. (Tous les shields sont compatibles)
- Connecteurs STMicroelectronics Morpho
- ST-LINK/V2-1 debugger/programmer intégré
- Une LED utilisateur
- Deux boutons poussoirs B1 USER et B2 RESET
- USB supportant trois modes :
 - Virtual Com port
 - Memoire USB
 - Debug port
- Adapté à tous les EDI ex : IAR™, Keil®, AC6-GCC, TrueSTUDIO, STM32CubeIDE ...

21

Schéma bloc Cartes NUCLEO 64

2 Parties :

- ■ la partie microcontrôle, « **MCU part** », sur laquelle il est facile de reconnaître les Boutons poussoirs B1 pour l'utilisateur et B2 pour l'initialisation, la LED LD2 pour l'utilisateur, les connecteurs Arduino et STMorpho, ainsi que le microcontrôleur STM32, situé au centre ;
- ■ la partie programmeur-débugueur de type ST-LINK/V2-1, nommé « **ST-LINK part** ». Il s'agit également d'un STM32 mais de plus petite taille que le microcontrôleur principal.



22

Vue dessus Cartes NUCLEO 64

2 connecteurs :

- **Arduino compatible headers** (CN6, 8, 5, 9)
- **Morpho headers** (CN7, 10)

Un programmeur-linker (générateurs de liens) de type ST-LINK

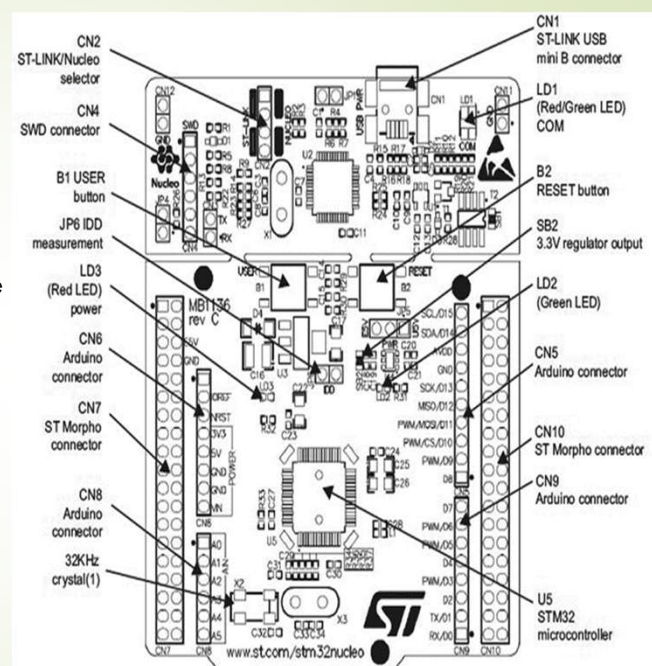
3 LEDs :

- LD1 pour indiquer que le port USB est opérationnel ;
- LD2 qui peut être utilisée dans l'application ;
- LD3 qui indique la présence de l'alimentation.

2 BPs :

- B1 pour l'application si nécessaire ;
- B2 pour initialiser la carte.

Les cartes Nucleo-64 contiennent un oscillateur externe à 32,768 kHz utilisé par le STM32.



23

STM32L152RE

Ultra-low-power
 Arm Cortex-M3 MCU
 with 512 Kbytes of
 Flash memory,
 32 MHz CPU,
 USB,
 2xOp-amp

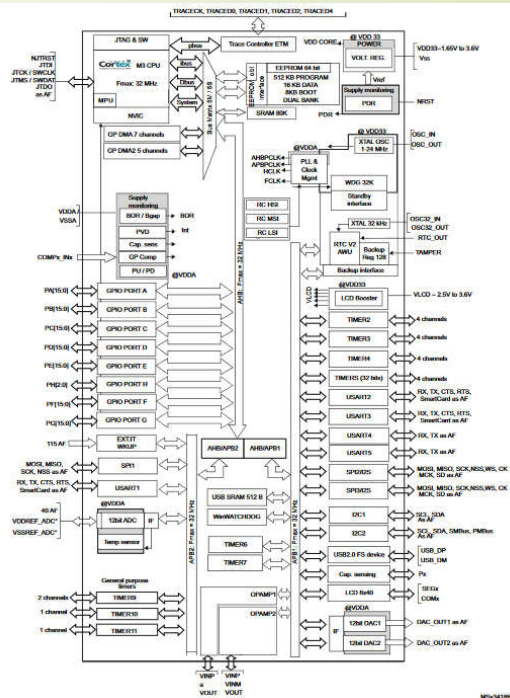
<p>System</p> <ul style="list-style-type: none"> Power supply 1.8 V internal regulator POR/PDR/PVD/BOR Xtal oscillators 32 kHz + 1 ~24 MHz Internal RC oscillators 37 kHz + 16 MHz Internal multispeed ULP RC oscillator 64 kHz to 4 MHz PLL Clock control RTC/AWU 2x watchdogs (independent and window) 37/51/83/109/115 I/Os Cyclic redundancy check (CRC) Voltage scaling 3 modes Touch sensing 	<p>ARM Cortex-M3 CPU 32 MHz</p> <ul style="list-style-type: none"> Nested vector interrupt controller (NVIC) JTAG/SW debug Embedded Trace Macrocell (ETM) Memory protection unit (MPU) <p>AHB bus matrix</p> <ul style="list-style-type: none"> 12-channel DMA 512-Kbyte Flash memory / dual bank 80-Kbyte SRAM 128-byte backup data 16-Kbyte EEPROM Boot ROM 	<p>Analog</p> <ul style="list-style-type: none"> 2x 12-bit DAC 12-bit ADC SAR Up to 40 channels 2x comparators 2x op-amps Temperature sensor <p>Connectivity</p> <ul style="list-style-type: none"> USB 2.0 FS 5x USART 3x SPI 2x I²C <p>Control</p> <ul style="list-style-type: none"> 6x 16-bit timer 1x 32-bit timer 1x 16-bit basic timer <p>Display</p> <ul style="list-style-type: none"> LCD driver (8x40)
---	---	---

24

Ultra-low-power STM32L151xE and STM32L152xE block diagram

STM32L152RE

Ultra-low-power
 Arm Cortex-M3 MCU
 with 512 Kbytes of
 Flash memory,
 32 MHz CPU,
 USB,
 2xOp-amp



25

Outils de développement logiciel STM32

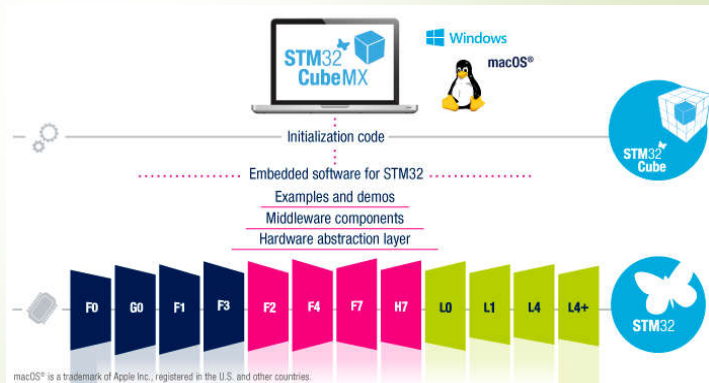
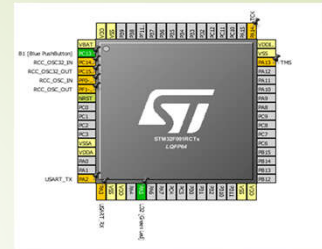
La famille de microcontrôleurs STM32 et microprocesseur basés sur le cœur Arm® Cortex® 32 bits est prise en charge par un ensemble complet d'outils logiciels



26

STM32CubeMX

STM32CubeMX est un outil graphique utilisé pour configurer n'importe quel appareil STM32. Cette interface utilisateur graphique facile à utiliser génère le code C d'initialisation pour les cœurs Cortex-M et génère l'arborescence du projet dans STM32CubeIDE.



27

STM32CubeIDE

STM32CubeIDE est un **environnement de développement intégré**. Basé sur des solutions open source comme **Eclipse** ou la chaîne d'outils GNU C/C++, cet IDE inclut des **fonctionnalités de rapport de compilation** et des **fonctionnalités de débogage avancées**. Il intègre également des fonctionnalités supplémentaires présentes dans d'autres outils de l'écosystème, telles que l'initialisation matérielle et logicielle et la génération de code à partir de STM32CubeMX.



28

STM32CubeProgrammer

STM32CubeProgrammer fournit un environnement facile à utiliser et efficace pour lire, écrire et vérifier la mémoire de l'appareil via l'interface de débogage (JTAG et SWD) et l'interface du chargeur de démarrage (UART et USB).



29

STM32CubeMonitor

Les outils de visualisation de données STM32CubeMonitor permettent de **surveiller et de diagnostiquer les applications STM32** lors de l'exécution. En plus des versions spécialisées, **l'outil polyvalent STM32CubeMonitor permet de visualiser le comportement de l'application sur les périphériques hôtes sous divers facteurs de forme tels que les PC, les tablettes ou les smartphones avec acquisition de données à distance.**

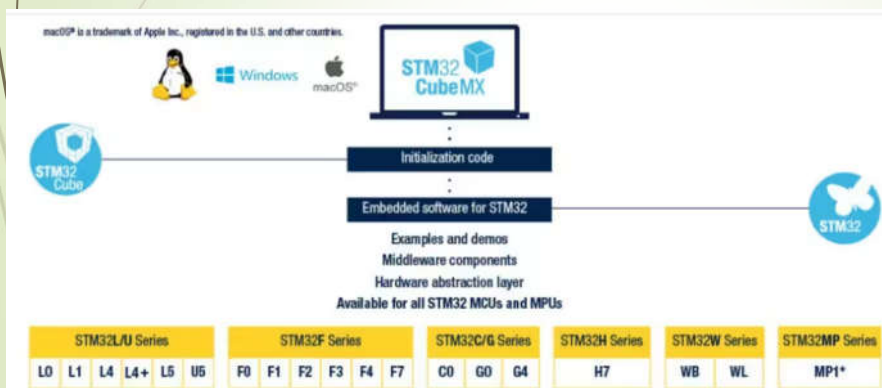


30

Packages MCU et MPU STM32Cube

Pour chaque série de MCU STM32 qui incluent :

- La couche d'abstraction matérielle (HAL) permettant la portabilité entre différents appareils STM32 via des appels d'API standardisés
- Des API de couche inférieure (LL) et un ensemble d'API léger, optimisé et orienté expert conçu à la fois pour les performances et l'efficacité de l'exécution
- Une collection de composants middleware, y compris RTOS, bibliothèque USB, système de fichiers, pile TCP/IP, bibliothèque de détection tactile ou bibliothèque graphique (selon la série STM32)
- Piles RF telles que Bluetooth® Low Energy 5.3, OpenThread, Zigbee 3.0, LoRaWAN® et Sigfox, spécifiques à chaque série sans fil STM32



STM32Cube est un ensemble de logiciels (packages) associés aux microcontrôleurs STM32.

Ex de package : **STM32CubeL1**
pour notre microcontrôleur
STM32L152RE équipant notre carte
NUCLEO64_STM32L152RET6.

5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les ports d'E/S (GPIO) (General Purpose Input Output) (entrée sortie à usage général) :

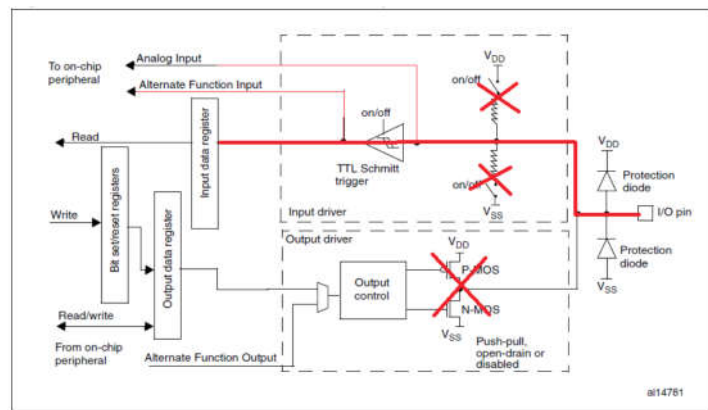
Le GPIO est un périphérique très particulier en ce sens que :

- Il est **l'interface entre l'intérieur du µC et l'extérieur** : il est donc nécessaire de configurer chaque E/S en respectant les contraintes électriques extérieures sans quoi non seulement le comportement souhaité ne sera pas le bon, mais surtout, l'E/S considérée peut être détruite.
- La famille STM32 dispose de nombreuses broches qui peuvent être configurées en GPIO (General Purpose Input Output, **entrée-sortie à usage général**) ou en alternate function (**fonction alternative**).
- **Les différentes configurations d'un GPIO :**
 - Entrée : floating input, analog input, pull up, pull down.
 - Sortie : Push-Pull, OpenDrain, Alternate PushPull, Alternate Output OpenDrain.
- **La programmation du GPIO se fait au travers de plusieurs registres :**
 - Les registres de configuration CRL et CRH permettent de configurer la I/O,
 - Les registres d'utilisation IDR, ODR, BSR et BSRR permettent quant à eux de fixer le niveau de chaque IO, ou de lire l'état logique de chaque IO.

5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

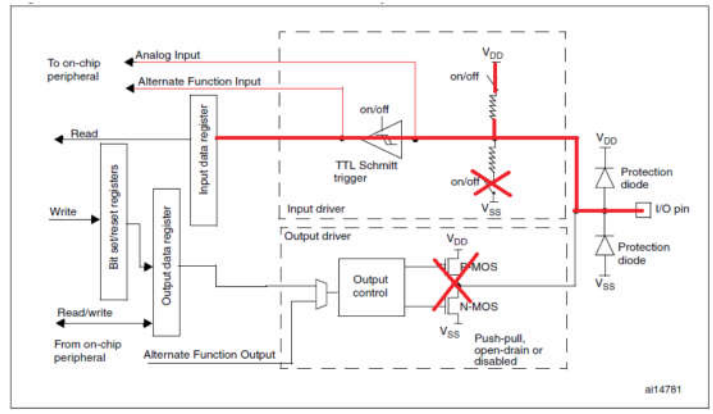
Floating input : de la pin x vers IDR.x, ou encore vers un périphérique (Timer, USART ...)



5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

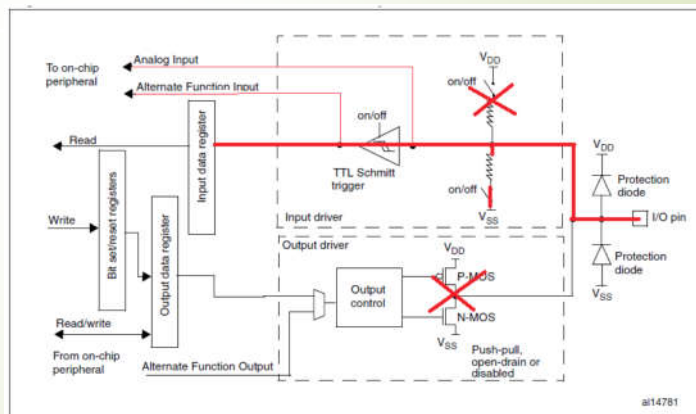
Input Pull Up : de la pin x vers IDR.x, ou encore vers un périphérique (Timer, USART ...)



5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

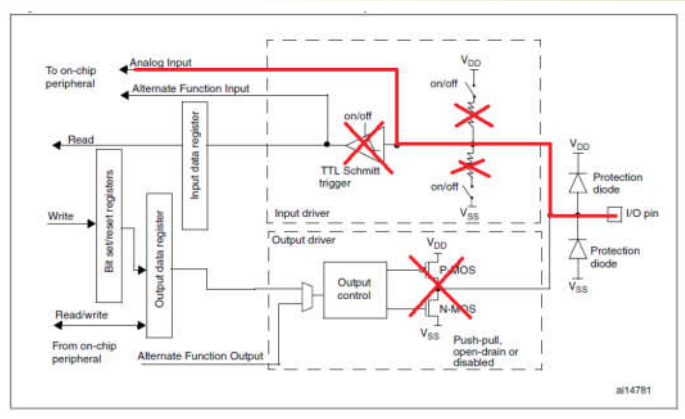
Input Pull Down : de la pin x vers IDR.x, ou encore vers un périphérique (Timer, USART ...)



5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

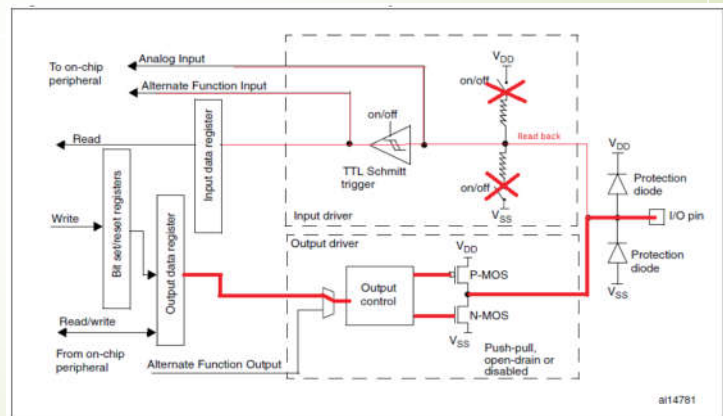
Analog Input : de la pin x vers l'ADC1 ou l'ADC2.



5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

Output Push-pull : de ODR.x à la pin x , forçage « fort » de la pin x à 3.3V ou 0V.



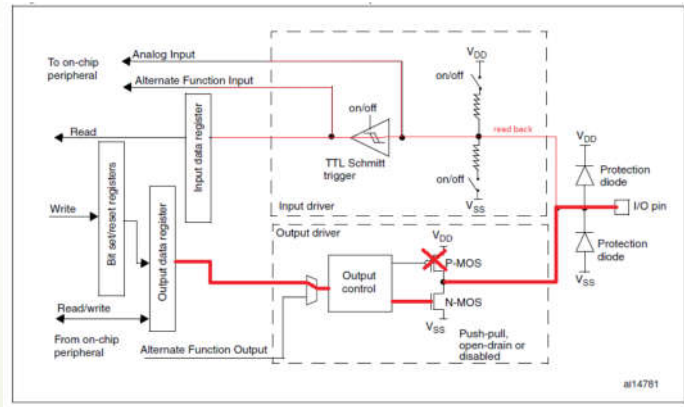
BE Microcontrôleur STM32

37

5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

Output Open Drain : de ODR.x à la pin x , forçage « fort » de la pin x à 0V uniquement.



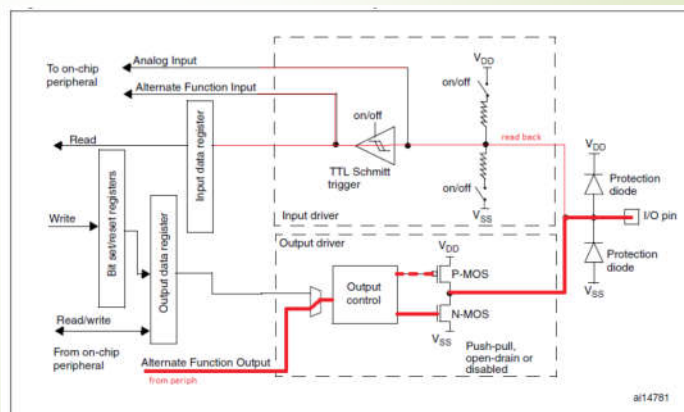
BE Microcontrôleur STM32

38

5 - Exemples de projets simples autour de GPIO utilisant l'EDI CubeIDE

Les différentes configurations d'un GPIO :

Alternate Output Push-pull ou Open Drain : de puis un périphérique (Timer, USART...) vers la pin x. On trouvera les deux options , Push-pull ou Open Drain (deux modes).



5.1 - Projet simple avec STM32CubeIDE

- Projet 1 :** Clignotement LED : « LED »
- Cahier des charges :** Un appui sur le bouton-poussoir RESET B2 de la carte Nucleo-L152RE initialise le programme. La LED LD2 clignote avec une période de 1 seconde.
- Choix de la carte :** NUCLEO64-STM32L152RE
- Initialisation :** Utilisation de STM32CubeMX (configurateur graphique) pour générer le code d'initialisation du programme.
- EDI :** Mise au point du programme avec STM32CubeIDE.
- Test sur la carte NUCLEO-L152RE**

5.2 - Projet simple avec STM32CubeIDE

- Projet 2 :** Contrôleur LED par GPIO : « BP_LED »
- Cahier des charges :** Un appui sur le bouton-poussoir RESET B2 de la carte Nucleo-L152RE initialise le programme. La LED LD2 clignote lorsque l'utilisateur maintient le bouton poussoir B1 enfoncé
- Choix de la carte :** NUCLEO64-STM32L152RE
- Initialisation :** Utilisation de STM32CubeMX (configurateur graphique) pour générer le code d'initialisation du programme.
- EDI :** Mise au point du programme avec STM32CubeIDE.
- Test sur la carte NUCLEO-L152RE**

5.3 - Projet simple GPIO avec interruption avec STM32CubeIDE

- Projet 3 :** GPIO sous interruptions : « BP-LED-Interruption »
- Cahier des charges :** Un appui sur le bouton poussoir RESET B2 de la carte Nucleo-L152RE initialise le programme.
La LED LD2 change d'état lorsque le bouton poussoir utilisateur B1 est relâché ou si un front descendant apparaît sur PA8 ou PA9.
- Choix de la carte :** NUCLEO64-STM32L152RE
- Initialisation :** Utilisation de STM32CubeMX (configurateur graphique) pour générer le code d'initialisation du programme.
- EDI :** Mise au point du programme avec STM32CubeIDE.
- Test sur la carte NUCLEO-L152RE**