

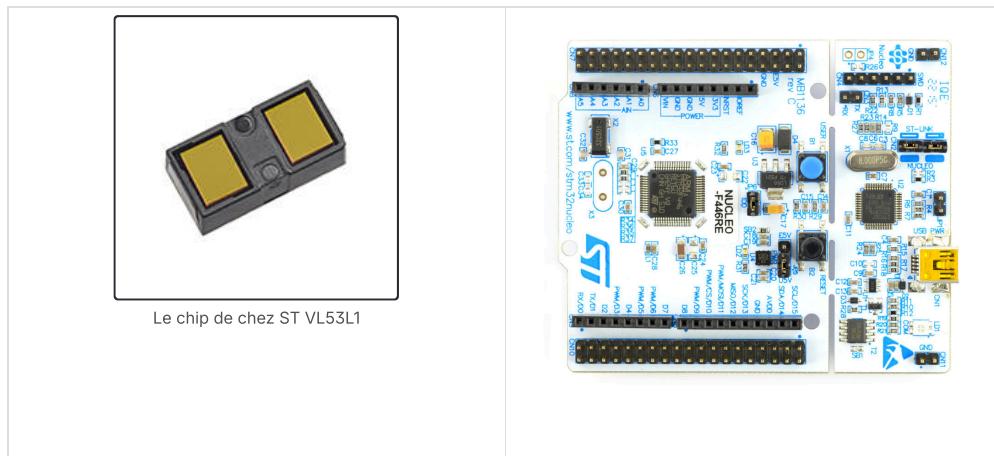


## STM32 Tutoriel

L'objectif de ce projet est de mettre en œuvre un **capteur de distance Time-of-Flight (ToF) VL53L1** en utilisant une carte de développement **NUCLEO-STM32F446RE**.

Le principe est simple : le capteur envoie un faisceau lumineux infrarouge, mesure le temps de vol du signal réfléchi sur un objet, et en déduit la distance avec une grande précision.

[ la carte NUCLEO-STM32F446RE et le capteur TOF VL53L1 ]



### 1- La Carte NUCLEO-STM32446RE:

Les **STM32** sont une famille de microcontrôleurs développés par **STMicroelectronics**.

Ils sont basés sur un **cœur ARM Cortex-M**, qui est une architecture très répandue dans l'embarqué grâce à sa puissance et sa faible consommation d'énergie.

Ces microcontrôleurs existent dans une large gamme :

- des modèles très simples et économiques, pour des projets basiques,
- jusqu'à des modèles très performants, capables de gérer des applications complexes en temps réel.

Les STM32 sont largement utilisés parce qu'ils offrent :

une grande variété de périphériques intégrés (I2C, SPI, UART, ADC, timers...).

### a. Documentation de la carte **NUCLEO-STM32446RE**

</doc/nucleo-c031c6.pdf>



La carte **NUCLEO-F446RE** est une plateforme qui embarque un chip de la famille **STM32F4** de STMicroelectronics. C'est un microcontrôleur basé sur un **œur ARM Cortex-M4 cadencé jusqu'à 180 MHz**, avec unité de calcul en virgule flottante (FPU).

Quelques points clés :

- **Mémoire** : jusqu'à 512 Ko de Flash et 128 Ko de RAM, suffisant pour des applications complexes.
- **Périphériques intégrés** :
  - Interfaces de communication multiples : **I2C, SPI, USART/UART, CAN, USB**
  - Convertisseurs **ADC 12 bits** rapides pour lire des capteurs analogiques
  - **Timers avancés** pour PWM, capture/comparaison, etc.
- **Performance** : c'est un microcontrôleur milieu-haut de gamme, capable de gérer à la fois des calculs rapides et plusieurs périphériques en parallèle.

### b. Documentation du chip STM32F446

</doc/stm32f446re.pdf>

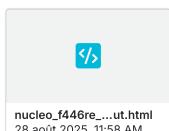


### c. les entrées sorties et le câblage:

Outil pour rechercher par fonction / Pinout

[✳ NUCLEO-F446RE Pinout Table | Claude](#)

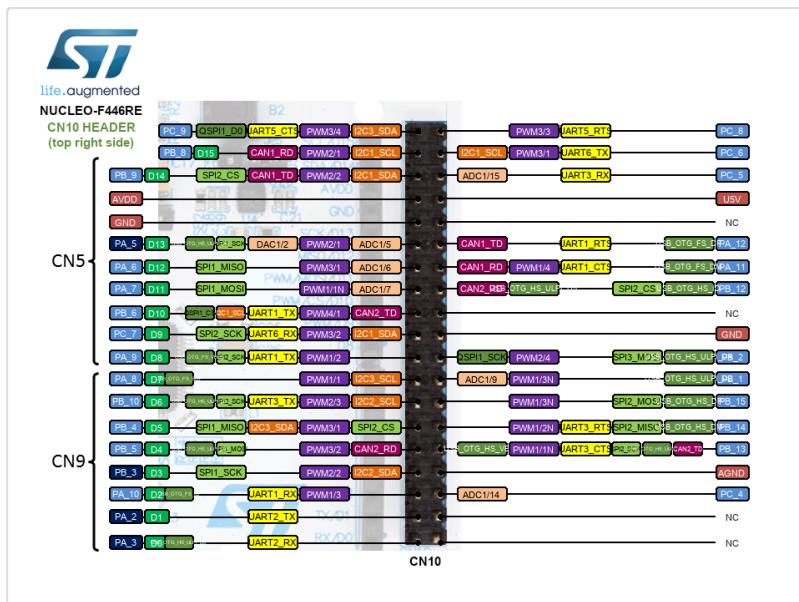
[fichier code source]

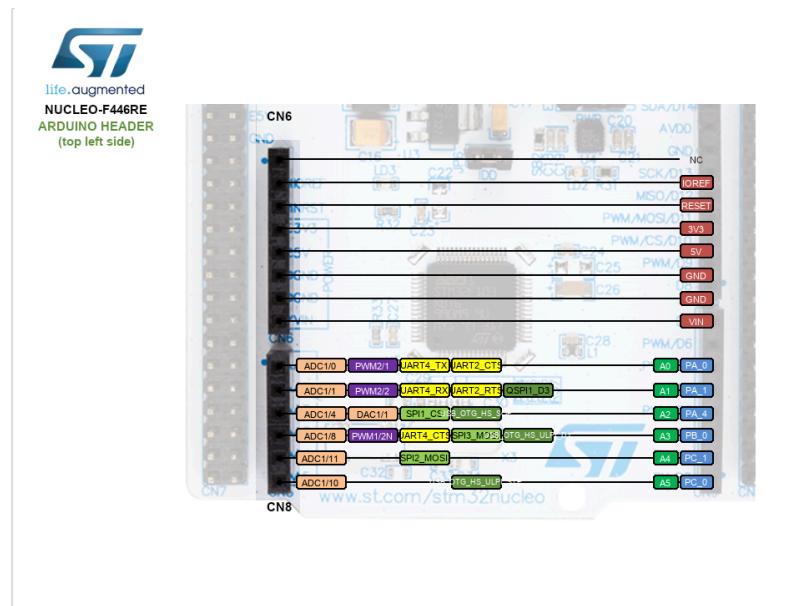


# Nucleo pinout

## Pins Legend

Labels usable in code	
<b>PX_Y</b>	MCU pin without conflict
<b>PX_Y</b>	MCU pin connected to other components See <a href="#">PeripheralPins.c</a> (link below) for more information
Labels not usable in code (for information only)	
<b>XXX</b>	Serial pins (USART/UART)
<b>XXX</b>	SPI pins
<b>XXX</b>	I2C pins
<b>XXX</b>	PWMOut pins (TIMER n/c[N]) n = Timer number c = Channel N = Inverted channel
<b>XXX</b>	Arduino connector names (A0, D1, ...)
<b>XXX</b>	LEDs and Buttons (LED_1, USER_BUTTON, ...)
<b>XXX</b>	AnalogIn (ADC) and AnalogOut pins (DAC)
<b>XXX</b>	CAN pins
<b>XXX</b>	Power and control pins (3V3, GND, RESET, ...)





## 2 - Utilisation du capteur TOF VL53L1 avec la carte STM32:

Le **VL53L1** est un capteur de distance de type **Time-of-Flight (ToF)** développé par STMicroelectronics ( mesure en mm avec précision).

Il envoie des impulsions laser infrarouges et mesure le temps que met la lumière à revenir, ce qui lui permet de calculer la distance avec une précision de l'ordre du millimètre.

Quelques points clés :

- **Plage de mesure** : jusqu'à environ **4 mètres**.
- **Communication** : via le bus **I<sup>2</sup>C**.
- **Précision** : fiable et stable, même avec des objets peu réfléchissants.
- **Applications typiques** : détection de présence, robotique (éviter les obstacles), interfaces gestuelles, objets connectés.

👉 En résumé, c'est un petit capteur compact et précis, idéal pour ajouter la **perception de distance** dans un projet embarqué.

### a. Documentation de l'utilisation du capteur de distance:

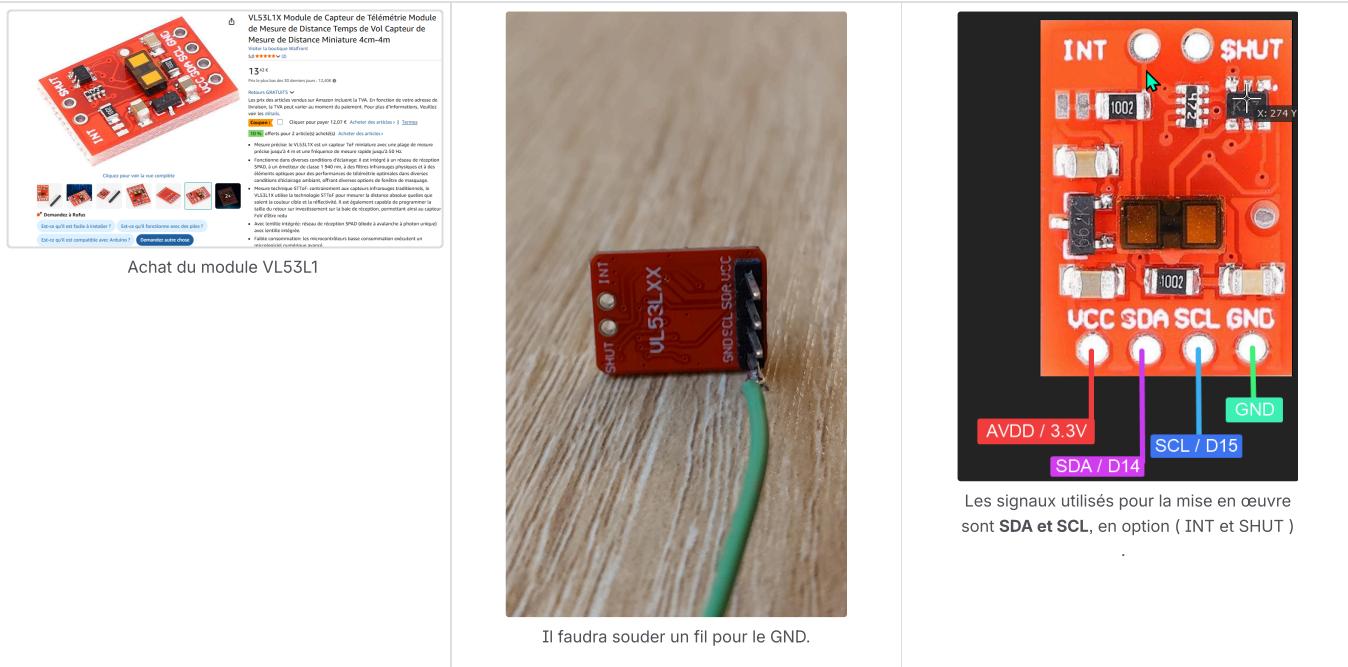
[/doc/um2978-how-to-use-the-vl53l1-with-stmicroelectronics-xcubetof1-timeofflight-sensor](https://www.st.com/resource/um/um2978-how-to-use-the-vl53l1-with-stmicroelectronics-xcubetof1-timeofflight-sensor)



## 3. Objectif du tutoriel:

👉 Montrer comment connecter le VL53L1 à un STM32 et récupérer des mesures de distance.

Pour faciliter la mise en œuvre, j'ai choisi d'acheter un **module déjà monté** basé sur le capteur **VL53L1**. Cela m'évite d'avoir à souder directement le composant, et le module expose toutes les broches nécessaires (VCC, GND, SDA, SCL, etc.), ce qui rend le câblage beaucoup plus simple avec la carte NUCLEO.



### 👉 Câblage du VL53L1 sur la carte NUCLEO-F446RE

Le capteur **VL53L1** communique en **I<sup>2</sup>C**, ce qui signifie qu'il utilise seulement **deux fils de données** (SDA et SCL), en plus de l'alimentation.

Les connexions de base sont donc :

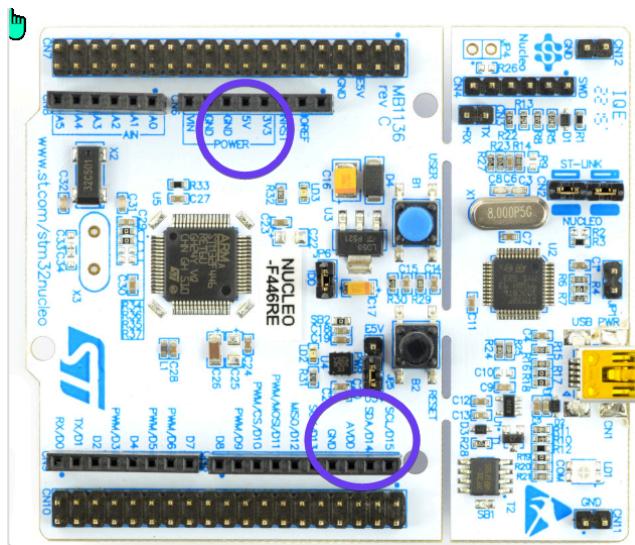
- **VCC** → **3.3V** de la carte NUCLEO
- **GND** → **GND** de la NUCLEO
- **SDA** → **PB9** (I<sup>2</sup>C1 SDA sur la NUCLEO-F446RE)
- **SCL** → **PB8** (I<sup>2</sup>C1 SCL sur la NUCLEO-F446RE)

👉 Le module VL53L1 dispose aussi des broches :

- **XSHUT** : permet de mettre le capteur en veille ou de le réinitialiser (peut être relié à un GPIO du STM32 si besoin).
- **GPIO1 (INT)** : broche d'interruption pour signaler qu'une mesure est prête (optionnelle pour les démos simples).

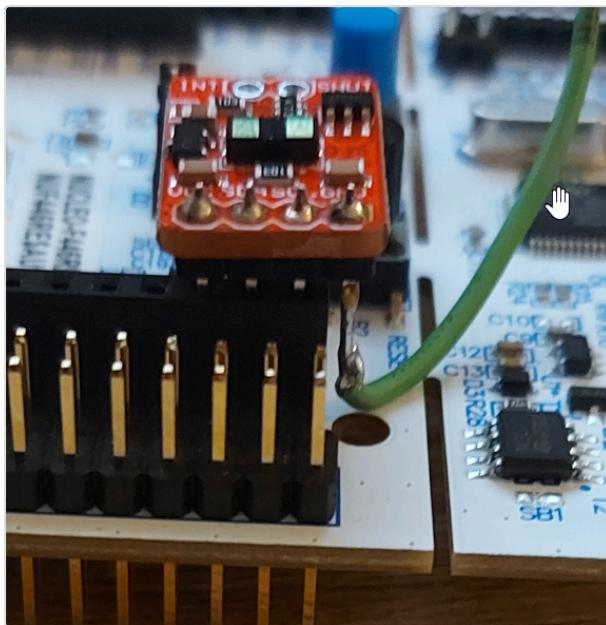


Les signaux SDA et SCL (les pins PB8/PB9 de la NUCLEO) sont sur le connecteur CN5



Les connecteurs pour l'alimentation, et les signaux I2C

Je vais utiliser pour le **+3.3V** le signal **AVDD** , plus simple pour le câblage du module VL53L1, et le fil ( en vert ) pour le relier au GND de la carte NUCLEO.



Le module VL53L1 sur la carte NUCLEO.

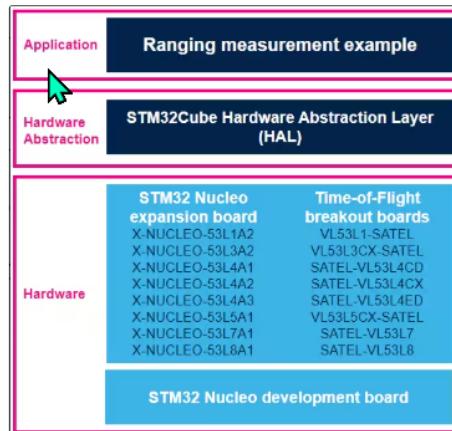


La carte et le module

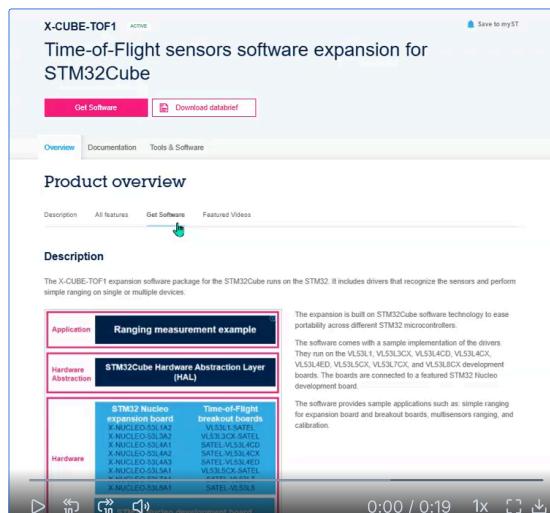
#### 4 – XCUBE-TOF1 de chez ST:

Le pack logiciel X-CUBE-TOF1 fournit directement les **drivers et exemples de code** pour le capteur VL53L1. Il permet donc de **faciliter la mise en œuvre** du module ToF avec un STM32, sans avoir à réécrire toutes les fonctions de communication et de calcul des distances.

X-CUBE-TOF1 | Product - STMicroelectronics



XCUBE-TOF1 permet de gérer plusieurs modules hardware, les drivers et exemples prêts à l'emploi.



XCUBE-TOF1

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/driver\\_xcube-tof1.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/driver_xcube-tof1.mp4)

#### 5 – Mise en place du Software STM32CubeMX :

**STM32CubeMX** est un outil qui permet de **configurer facilement le microcontrôleur STM32** et de générer automatiquement le code de base.

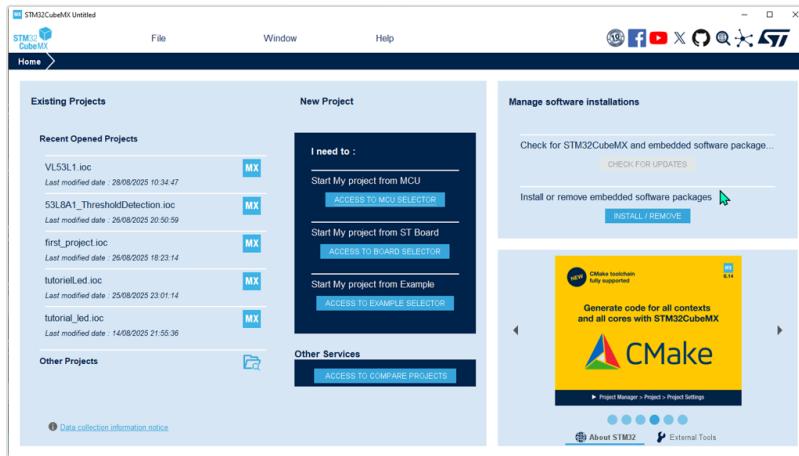
Il simplifie le développement, car on n'a pas besoin d'écrire manuellement toute l'initialisation des périphériques (I<sup>2</sup>C, UART, timers, etc.), ce qui fait gagner beaucoup de temps et réduit les erreurs.



Le logiciel  
STM32CubeMx

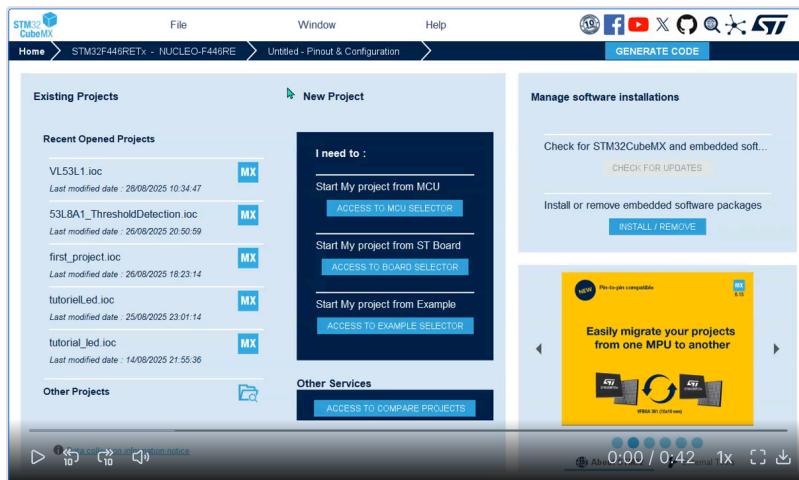
#### a. Configuration initiale dans STM32CubeMX

- Ouvrir STM32CubeMX.



Lancement de STM32CubeMx

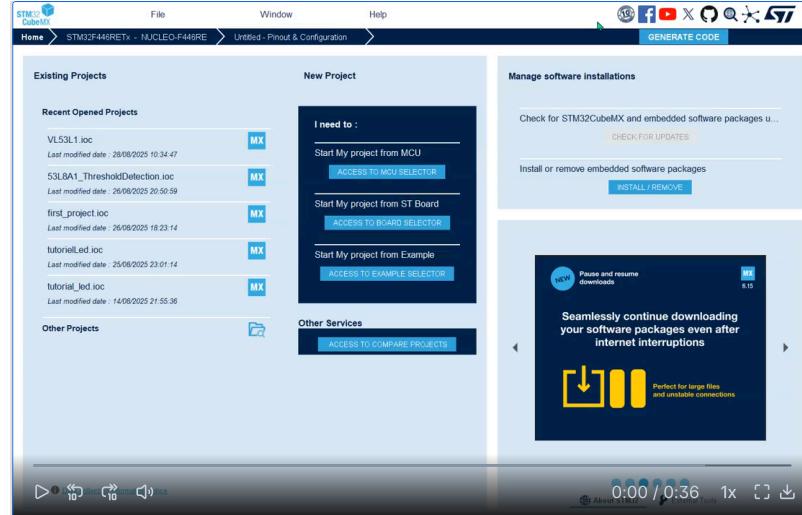
Nous allons dans un premier temps **installer le package XCUBE-TOF1**



installation du Package XCUBE-TOF1

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/driver\\_xcube-tof1.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/driver_xcube-tof1.mp4)

- Sélectionner la carte NUCLEO-STM32F446RE.

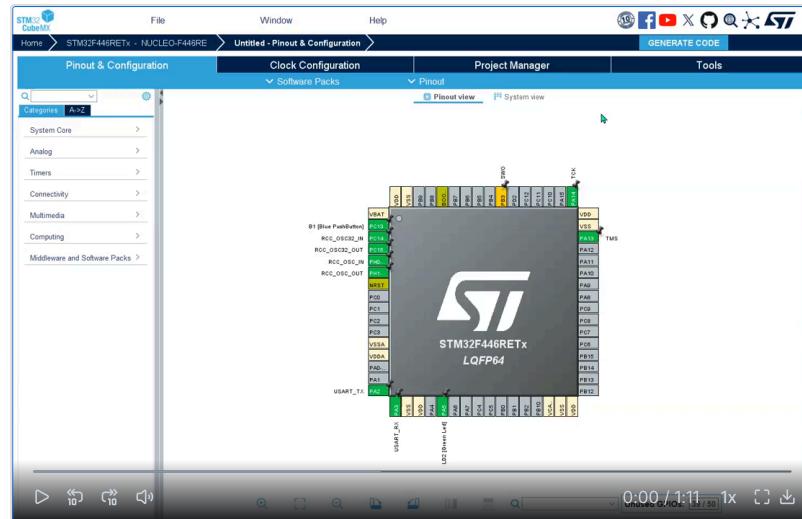


Sélection de la carte NUCLEO-F446RE

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/select\\_board.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/select_board.mp4)

Dans STM32CubeMX, il suffit d'aller dans l'onglet “Software Packs” , select components puis de sélectionner et configurer le composant X-CUBE-TOF1.

Cela permet d'intégrer directement le **driver et les exemples pour le capteur VL53L1** dans le projet, et de configurer facilement ses paramètres depuis l'interface graphique.



Sélection et configuration de XCUBE-TOF1

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/select\\_components.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/select_components.mp4)

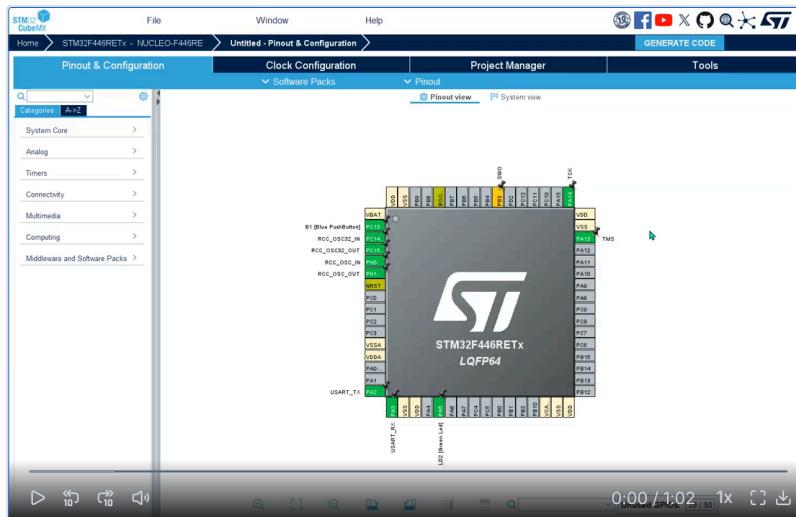
4. Click on "X-CUBE-TOF1" > select "53L1A2 Board Extension" > select "VL53L1CB\_SimpleRanging" > click "OK".

**Figure 59. VL53L1CB\_SimpleRanging**

STMicroelectronics.X-CUBE-TOF1	3.0.0	
Board Extension 53L1A2	1.0.0	<input type="checkbox"/>
Board Extension 53L3A2	2.0.0	<input type="checkbox"/>
Board Extension 53L5A1	1.0.0	<input type="checkbox"/>
Board Part Ranging / VL53L3CX	2.0.0	<input type="checkbox"/>
Board Part Ranging / VL53L5CX	1.0.0	<input type="checkbox"/>
Board Part Ranging / VL53L1CB	1.0.0	<input checked="" type="checkbox"/>
Device TOF1_Applications	1.0.0	<input checked="" type="checkbox"/>
Application	VL53L1CB_SimpleRanging	<input type="checkbox"/>
Board Support STM32Cube_Custom_BSP_Di	1.0.0	<input type="checkbox"/>
Custom / RANGING_SENSOR		<input checked="" type="checkbox"/>

installation du package XCUBE-TOF1 et configuration

- Configurer et activer le périphérique I<sup>2</sup>C1 (pour communiquer avec le VL53L1).

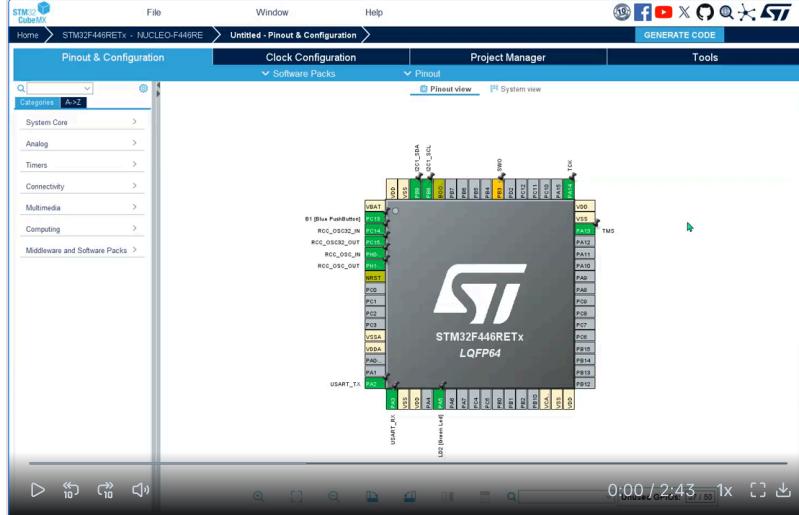


Configuration Pinout view , bus I<sup>2</sup>C

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/pinout\\_configuration.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/pinout_configuration.mp4)

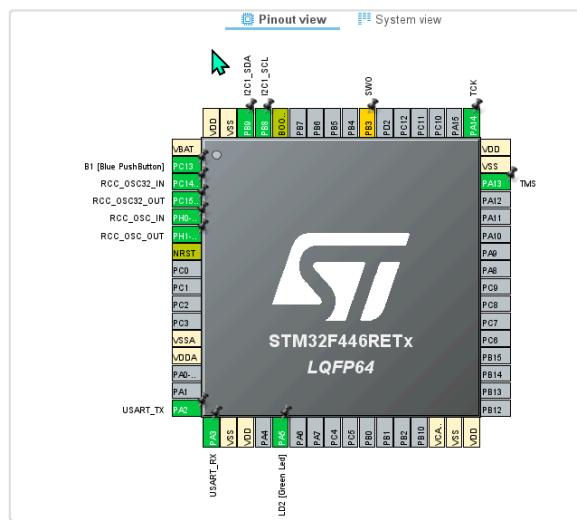
Dans **STM32CubeMX**, la section **Middleware et Software Pack** permet de **configurer facilement les composants logiciels** comme le **X-CUBE-TOF1**.

On peut activer le package, ajuster ses paramètres (par exemple la fréquence I<sup>2</sup>C, le mode de fonctionnement du capteur, etc.) et **préparer automatiquement le projet pour CubeIDE** sans écrire manuellement le code de gestion du capteur.



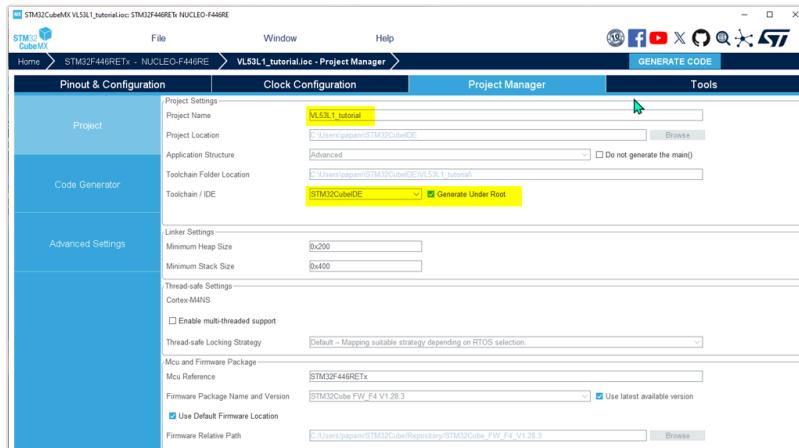
Middleware et software Packs pour XCUBE-TOF1 et génération du code

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/Middleware\\_software\\_pack.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/Middleware_software_pack.mp4)



Pinout view configuré

## 6 - Génération du code et ouverture dans STM32CubeIDE

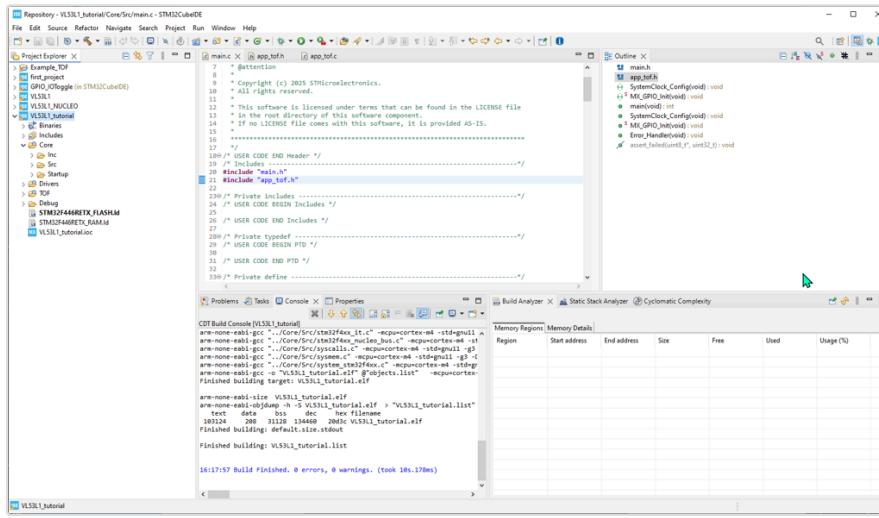


La configuration du projet

- Cliquer sur GENERATE CODE.

- Ouvrir automatiquement le projet dans **STM32CubeIDE**.

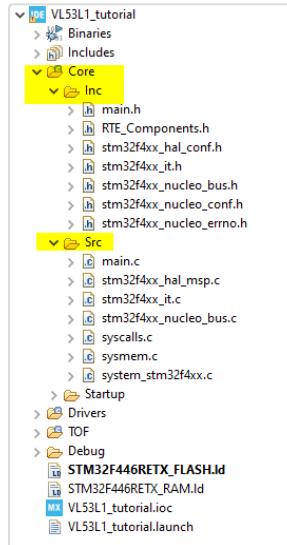
## 7 – Le code dans STM32CubeIDE :



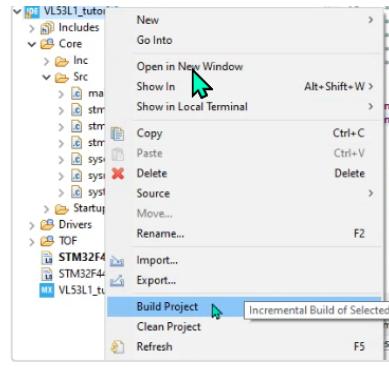
Le code dans STM32CubeIDE

- Vérifier la structure du projet :

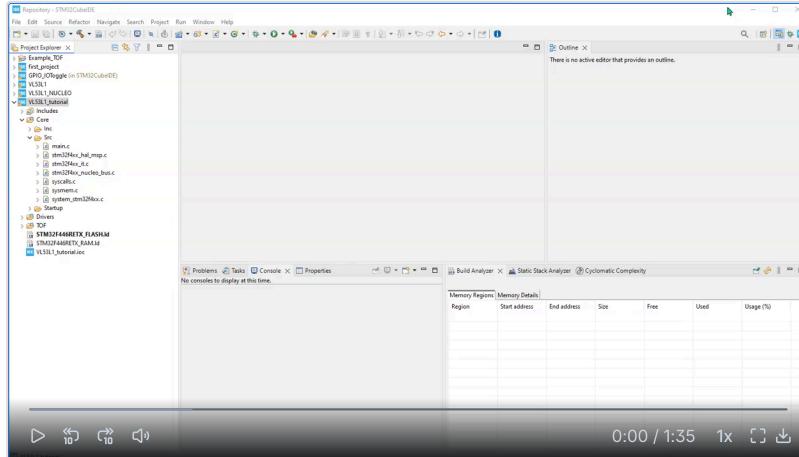
- Drivers STM32
- Middleware VL53L1
- Dossier `Core/Src` et `Core/Inc`



Le code et la structure Dossier `Core/Src` et `Core/Inc`

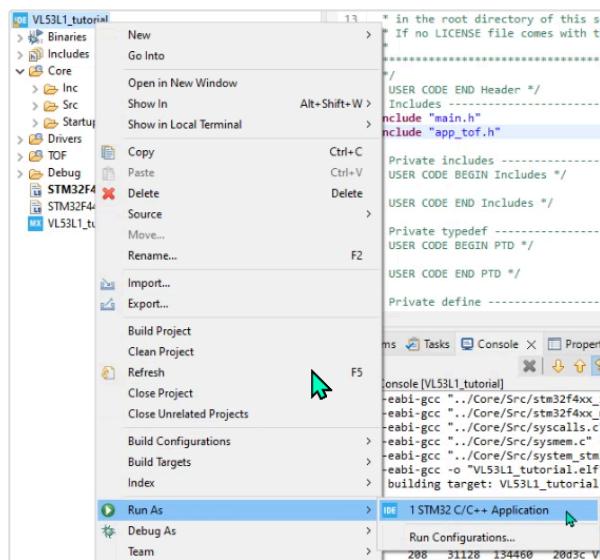


Build Project

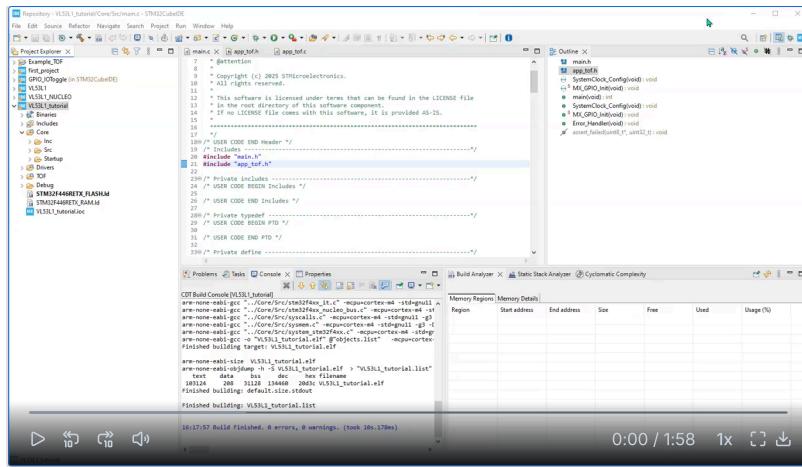


Le code , les fichiers

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/stm32cubeide\\_code\\_build.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/stm32cubeide_code_build.mp4)



run Application

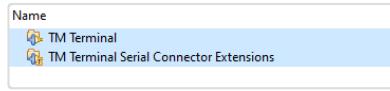


run Application, le terminal

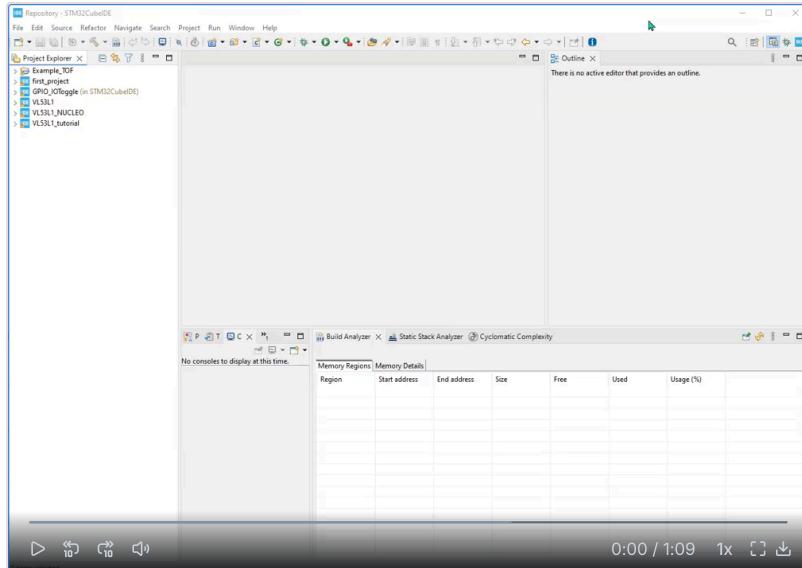
[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/stm32cubeide\\_runApplication.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/stm32cubeide_runApplication.mp4)

L'installation du **Terminal Série** dans CubeIDE permet de **visualiser en temps réel les données envoyées par le STM32**.

C'est très pratique pour **suivre les mesures du capteur VL53L1**, vérifier que le code fonctionne correctement, et observer les variations de distance directement sur l'ordinateur, sans avoir à ajouter d'affichage externe.



Ajout du Terminal , Terminal Serial



ajouter le terminal

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/stm32cubeide\\_ajout\\_terminal.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/stm32cubeide_ajout_terminal.mp4)

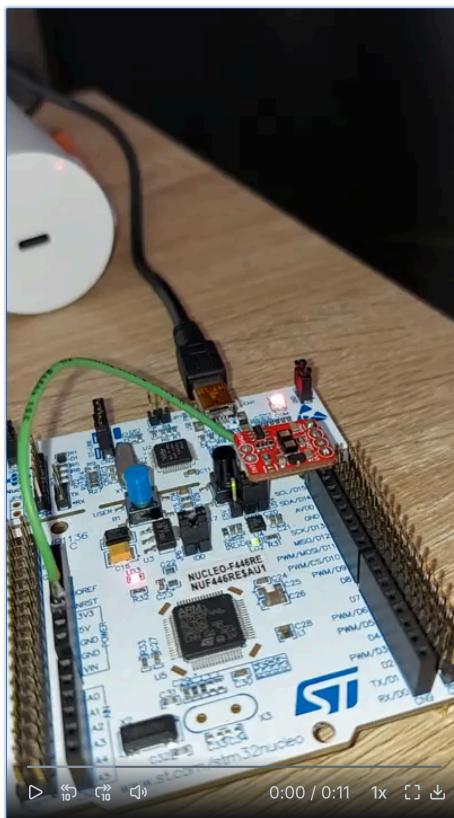
- Ouvrir le Terminal Série intégré à CubeIDE
- Modifier le paramétrage (baudrate 115200, 8N1).

## 8 - Test du capteur (lecture des distances)

Après avoir flashé le code dans la carte **NUCLEO-STM32F446RE**, il peut arriver que le **Terminal Série** n'affiche pas immédiatement les données du capteur.

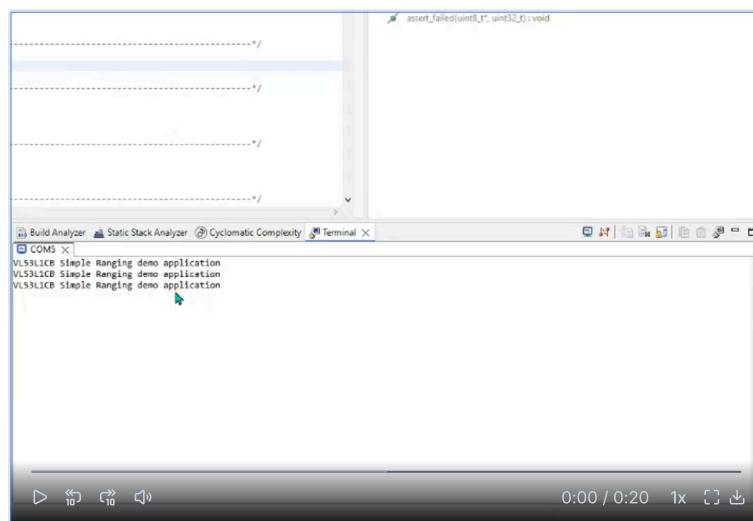
Dans ce cas, il suffit de **débrancher puis rebrancher le câble USB** de la carte.

Cela permet de réinitialiser la connexion série et d'obtenir correctement l'affichage des mesures envoyées par le microcontrôleur.



Test du programme et mesure de la distance

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/carte\\_main\\_detection.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/carte_main_detection.mp4)



Exécution du code

[https://github.com/esaid/STM32\\_Capteur\\_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos\\_videos/stm32cubeide\\_execution.mp4](https://github.com/esaid/STM32_Capteur_TOF/raw/fa6b7ad4d4fcdfbc1bb1fe87e3ccfab085219cdf/photos_videos/stm32cubeide_execution.mp4)

```

Build Analyzer Static Stack Analyzer Cyclomatic Complexity Terminal X
COM5 X

Targets = 1
|---> Status = 0, Distance = 65 mm , Ambient = 13.67 kcps/spad, Signal = 9500.00 kcps/spad
Targets = 1
|---> Status = 0, Distance = 59 mm , Ambient = 15.62 kcps/spad, Signal = 10921.87 kcps/spad
Targets = 1
|---> Status = 0, Distance = 61 mm , Ambient = 15.62 kcps/spad, Signal = 12242.18 kcps/spad
Targets = 1
|---> Status = 0, Distance = 59 mm , Ambient = 15.62 kcps/spad, Signal = 13098.95 kcps/spad
Targets = 1
|---> Status = 0, Distance = 60 mm , Ambient = 13.02 kcps/spad, Signal = 11979.16 kcps/spad
Targets = 1
|---> Status = 0, Distance = 59 mm , Ambient = 13.02 kcps/spad, Signal = 12666.66 kcps/spad
Targets = 1
|---> Status = 0, Distance = 59 mm , Ambient = 13.02 kcps/spad, Signal = 12192.70 kcps/spad

```

Mesure de la distance avec le capteur

- Avec CubeMX → configuration rapide des périphériques
- Avec CubeIDE → intégration du driver VL53L1 (X-CUBE-TOF1)
- Avec UART → affichage facile des mesures
- Transition vers la suite : applications possibles (robotique, détection de présence, etc.).



#### Conclusion

Dans ce projet, nous avons vu comment **utiliser un capteur VL53L1 avec une carte STM32F446** :

- Grâce à **STM32CubeMX**, la configuration du microcontrôleur et des périphériques est rapide et intuitive.
- Le **package X-CUBE-TOF1** simplifie l'intégration du capteur en fournissant drivers et exemples prêts à l'emploi.
- Le **Terminal Série** permet de visualiser les mesures en temps réel et de vérifier le bon fonctionnement du montage.

En combinant ces outils, on peut **mettre en œuvre un capteur de distance ToF de façon simple et efficace**, et ainsi ouvrir la porte à de nombreuses applications : robotique, domotique, détection de présence, ou interfaces interactives.