

User manual

How to use STMicroelectronics' X-CUBE-TOF1 Time-of-Flight sensor software packages for STM32CubeMX

Introduction

The X-CUBE-TOF1 expansion software package for STM32Cube runs on the STM32. It includes drivers that recognize the sensors, and it performs simple ranging on single or multiple devices.

The expansion is built on STM32Cube software technology to ease portability across different STM32 microcontrollers.

The software comes with a sample implementation of the drivers, running on different Time-of-Flight sensor evaluation boards, connected to a featured STM32 Nucleo development board.

This user manual gives an overview of the use of the STM32CubeMX with the following Time-of-Flight ranging sensors: VL53L4CX, VL53L4CD, VL53L7CX, and VL53L8CX. For further information on the Time-of-Flight sensors supported by the X-CUBE-TOF1, please refer to the software page on www.st.com.

The evaluation boards supported by the X-CUBE-TOF1 expansion software package include:

- X-NUCLEO expansion board
- SATEL breakout board

The X-CUBE-TOF1 software provides some sample applications for the Time-of-Flight sensors listed above.

Visit the STM32Cube ecosystem web page on www.st.com for further information.



1 Acronyms and abbreviations

Acronym	Definition	
API	application programming interface	
BSP	board support package	
HAL	hardware abstraction layer	
l²C	inter-integrated circuit	
IDE	integrated development environment	
MCU	microcontroller unit	
NVIC	nested vector interrupt control	
PCB	printed circuit board	
SDK	software development kit	
ToF	Time-of-Flight sensor	
USB	universal serial bus	



2 X-CUBE-TOF1 software expansion for STM32Cube

2.1 Overview

The X-CUBE-TOF1 software package expands the STM32Cube functionality. The key features are:

- Complete software to build applications using the evaluation boards listed in the "Introduction".
- Several application examples to show the innovative technology for the accurate distance ranging capability.
- Sample application to transmit real-time sensor data to a PC.
- Precompiled binaries available on all evaluation boards (listed in the "Introduction") connected to a NUCLEO-F401RE or NUCLEO-L476RG development board.
- Package compatible with STM32CubeMX, can be downloaded from, and installed directly into, STM32CubeMX.
- Easy portability across different MCU families, thanks to the STM32Cube.
- Free, user-friendly license terms.

2.2 Architecture

This software is a fully compliant expansion of the STM32Cube. It enables the development of applications using Time-of-Flight sensors.

The software is based on the hardware abstraction layer for the STM32 microcontroller, STM32CubeHAL. The package extends the STM32Cube by providing a board support package (BSP) for the sensor expansion board, and a sample application for serial communication with a PC.

The software layers used by the application software to access the sensor expansion board are:

- The STM32Cube HAL driver layer. It provides a simple, generic, and multi-instance set of APIs (application programming interfaces) to interact with the upper layers (application, libraries, and stacks). It includes generic and extension APIs and is based on a generic architecture. This allows the layers that are built on it (such as the middleware layer) to implement their functionalities without dependence on the specific hardware configuration of a given microcontroller unit (MCU). This structure improves library code reusability and guarantees high portability across other devices.
- The BSP layer. It provides supporting software for the peripherals on the STM32 Nucleo board, except for the MCU. It has a set of APIs to provide a programming interface for certain board-specific peripherals (for example, the LED and the user button). The BSP layer allows identification of the specific board version. For the sensor expansion board, it provides the programming interface for various Time-of-Flight sensors, and support for initializing and reading sensor data.



Figure 1. X-CUBE-TOF1 software architecture

2.3 Folder structure

Figure 2. X-CUBE-TOF1 package folder structure



The following folders are included in the software package:

- The Documentation folder contains the current user manual and detailed documentation regarding the software components and APIs. In the .chm file, you can find a list of pin resources used for the XNUCLEO. It is updated when a new board is released.
- The Drivers folder contains:
 - HAL drivers
 - Board-specific drivers for each supported board or hardware platform. This includes drivers for the on-board components and the CMSIS layer (which is a vendor-independent hardware abstraction layer for the Cortex®-M processor series).
- The Projects folder contains several examples and applications for NUCLEO-L476RG and NUCLEO-F401RE platforms. They show the use of sensor APIs provided with three development environments:
 - IAR Embedded Workbench for Arm
 - MDK-ARM® microcontroller development kit
 - STM32CubeIDE
- The STM32CubeMX folder contains all the templates used by the CubeMX ToF pack.

2.4 APIs

Detailed technical information about the APIs available to the user can be found in the compiled HTML file "X-CUBE-TOF1.chm". This file is in the Documentation folder of the software package. All the functions and parameters are fully described.



3 System setup guidelines

3.1 Hardware description

In this section, the VL53L7CX hardware is used as an example. For each sensor listed in the introduction, the same hardware is available.

3.1.1 STM32 Nucleo

STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

The Arduino® connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform. They have a wide range of specialized expansion boards to choose from. The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library, together with various packaged software examples for different IDEs:

- IAR EWARM®
- Keil® MDK-ARM®
- STM32CubeIDE
- Mbed[™] and GCC/LLVM Arm

All STM32 Nucleo users have free access to the Mbed[™] online resources (compiler, C/C++ SDK, and developer community) at www.mbed.org to easily build complete applications.



Figure 3. STM32 Nucleo board

Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo.



3.1.2 X-NUCLEO expansion boards

The X-NUCLEO expansion boards are for use with any Nucleo 64 development board. Figure 4. Example of an X-NUCLEO-53L7A1 expansion board with a cover glass is an example of the X-NUCLEO-53L7A1 expansion board, with a cover glass and spacers. They provide a complete evaluation kit allowing anyone to learn, evaluate, and develop their applications using the Time-of-Flight ranging sensor.

The X-NUCLEO expansion boards are delivered with a cover glass holder. It contains three different spacers of 0.25, 0.5, and 1 mm height. They can be fitted below the cover glass to simulate various air gaps.

The X-NUCLEO expansion boards are compatible with the STM32 Nucleo board family, and with the Arduino® UNO R3 connector layout.

Several STMicroelectronics' X-NUCLEO expansion boards can be superposed through the Arduino connectors. It allows the development of applications with Bluetooth or Wi-Fi interfaces.



Figure 4. Example of an X-NUCLEO-53L7A1 expansion board with a cover glass



3.1.3 SATEL breakout boards

The SATEL breakout boards can be used for easy integration into customer devices. Thanks to the voltage regulator, the breakout boards can be used in any application in the supply range 2.8 V to 5 V.

The PCB section supporting the module is perforated so that developers can break off the mini-PCB for use in a 2V8 or 3V3 supply application using flying leads. This makes it easier to integrate the SATEL breakout boards into development and evaluation devices due to their small form factor.



Figure 5. Example of a SATEL-VL53L7CX breakout board

3.2 Software description

The following software components are required to establish a suitable development environment for creating applications for the STM32 Nucleo equipped with the sensor expansion board:

- X-CUBE-TOF1: an STM32Cube expansion for sensor application development. The X-CUBE-TOF1 firmware and associated documentation is available on www.st.com.
- Development tool-chain and compiler: The STM32Cube expansion software supports the following three environments:
 - IAR Embedded Workbench for Arm(EWARM) toolchain + STLINK
 - RealView microcontroller development kit (MDK-ARM-STR) toolchain + STLINK
 - STM32CubeIDE for STM32 + STLINK



3.3 Hardware setup

The following hardware components are required:

- 1. One STM32 Nucleo development platform (suggested order code: NUCLEO-F401RE or NUCLEO-L476RG)
- 2. An X-NUCLEO expansion board or a SATEL breakout board
- 3. One USB type A to mini-B USB cable to connect the STM32 Nucleo to a PC

3.3.1 Setup using an STM32 Nucleo and an X-NUCLEO expansion board

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. Developers can download the relevant version of the ST-LINK/V2-1 USB driver by searching STSW-LINK008 or STSW-LINK009 (depending on your version of Windows) on www.st.com.

The X-NUCLEO expansion boards can be easily connected to the STM32 Nucleo board through the Arduino UNO R3 extension connector. They can also interface with the external STM32 microcontroller (on the STM32 Nucleo) via the inter-integrated circuit (I²C) transport layer.



Figure 6. Sensor expansion board plugged to STM32 Nucleo board



3.3.2 Setup using an STM32 Nucleo board and a SATEL breakout board

The SATEL breakout boards can be connected to the STM32 Nucleo board through flying wires. They can interface with the external STM32 microcontroller on the STM32 Nucleo board via the I²C transport layer. Figure 7. Example of the connection between an STM32 Nucleo board and a SATEL shows the connection between the SATEL-53L7CX and the STM32 Nucleo board.

Refer to the Time-of-Flight application notes on st.com, which explain how to connect each product SATEL to the Nucleo board.



Figure 7. Example of the connection between an STM32 Nucleo board and a SATEL



4 Sample application examples

The X-CUBE-TOF1 package comes with several examples for each sensor. For each example, you can either:

- use the precompiled file delivered in the binary folder, or
- configure the project to build the binary file used to configure the system.

This section describes two ways of running a sampling application using the evaluation boards and a STM32F401RE Nucleo board.

The same method can be used for each sensor listed in the introduction.

Note that information concerning the examples is in the readme.html file, which is delivered with the examples.

Figure 8. SimpleRanging folder architecture





4.1 SimpleRanging singular form using an STM32 Nucleo board and an X-Nucleo expansion board

4.1.1 Loading a prebuilt binary file

Figure 9. NUCLEO-F401RE folder architecture with an X-NUCLEO expansion board shows the folder architecture for the X-CUBE-TOF1 package examples. It is based on version 3.3.0 of the X-CUBE-TOF1, but can be used for all versions.



Figure 9. NUCLEO-F401RE folder architecture with an X-NUCLEO expansion board

The binary file is placed in the "Binary" folder (see Figure 9. NUCLEO-F401RE folder architecture with an X-NUCLEO expansion board).

Follow the steps below to run a SimpleRanging on a VL53L7CX sensor using the precompiled file:

- 1. Set up the hardware (see Section 3.3.1 Setup using an STM32 Nucleo and an X-NUCLEO expansion board).
- 2. Flash the NUCLEO-F401RE board with the prebuilt binary by dragging and dropping the binary file onto the NODE drive. You can call the disk name "NOD" or "NODE" depending on the Nucleo board variant.

Figure 10. NODE drive





3. Open a Tera Term and configure the settings.

Port:	COM11 ~	New setting
Speed:	460800 ~	lien ootang
Data:	8 bit ~	Cancel
Parity:	none ~	
Stop bits:	1 bit ~	Help
Flow control:	none ~	
evice Friendly N	msec/char 0 ame: STMicroelectr D: USB\VID_0483&P	msec/line onics STLink Virtual COM Pr ID_374B&MI_02\6&2377F314

Note that speed is defined in the stm32f4xx_nucleo.c file as "Init.BaudRate = 460800". Speed is also defined in the readme.html file in the examples directory (Figure 8. SimpleRanging folder architecture).

Figure 12. Tera Term terminal setup

Tera Term: Terminal setup		×
Terminal size 80 × 24 ✓ Term size = win size	New-line Receive: AUTO ~ Transmit: CR ~	OK Cancel
Auto window resize Terminal ID: VT100 ~ Answerback:	□ Local echo □ Auto switch (VT<->1	Help TEK)
Coding (receive) UTF-8 v	Coding (transmit) UTF-8 ~	
locale: american		



4. Wave your hand in front of the sensor to display the ranging data on the serial terminal, as shown below.

Figure 13. VL53L7CX ranging results

80	COMI	1 - Tera	Term VT									
File	Edit	Setup	Contro	Window	w Hel	P						
1	701 0	6	Pangi	i dana	7 :	instia 3	1	5 :	2	1		
551	17HI 0	Imple	nangi	ng aeno	appi	164610	-					
Jse 'i 'c Cel	the ; : c ; : e ; : c 11 For	follo hange nable lear nat :	wing k resol signa screen	eys to ution l and a	contr mbien	ol app t	licatio	n				
_	Signa	Dista 1 [kc	ps/spa	n]: d]: A	nbien	t [kcp	Status s/spad]					
	407 0		Ø 3	1924 6	-	Ø 3	1919 9	-	0 4	1982 8	:	83
	1818	:	0	1936	:	0	1940	:	0	1989	:	0
_	2	:	4 :	?	:	4 :	10	:	3 :	9	:	2
	1876 3	:	Ø 6	1959 8		0 10	1938 8	:	Ø 5	1910 8	:	Ø 3
	1982	:	0	1943	:	0	1912	:	03	1981	:	02
-												

5. Use the same method for the different X-CUBE-TOF1 package examples present in the "Examples" folder.



4.1.2

Configuring the STM32CubeMX project

The following two examples are presented:

- SimpleRanging
- ThresholdDetection

4.1.2.1 How to generate the SimpleRanging example with the STM32CubeMX

1. Create a new STM32 project.



Figure 14. Create a new STM32 project

- 2. Open STM32CubeIDE and follow the instructions to create a workspace area.
- 3. Select the NUCLEO-F401RE board using the "Board selector" field.

Figure 15. Choose the NUCLEO-F401RE board





4. Give the project a name and a location.

Figure 16. Give the project name	
IDE STM32 Project	×
Setup STM32 project	IDE
Project	
Project Name: F401_VL53L7CX	
Use default location	
Location: C:/Users/bigota/Downloads/L7CX-workspace Br	owse
Options Targeted Language ● C ○ C++	
Targeted Binary Type	
Targeted Project Type	
? < Back Next > Finish Ca	ncel

Click "Finish" and choose the "default configuration" when it pops up. The purpose is now to choose the right software pack.

5. Click on "Middleware and Software Packs".

F401_VL53L5CX.ioc ×					
F401_VL53L5CX.ioc - Pinout & Configuration					
Pinout & Configuration					
	✓ Soft				
Q ~	0				
Categories A->Z					
System Core	>				
Analog	>				
Timers	>				
Connectivity	>				
Multimedia	>				
Computing	>				
Middleware and Software Packs	>				

Figure 17. Middleware and software packs selection



6. Click on "X-CUBE-TOF1". Choose the "53L7A1 board", then the "53L7A1_SimpleRanging" application, then verify the green ticks.

Software Packs Component Selector							
Pack / Bundle / Component	Stat	us	Version	Selection			
STMicroelectronics.X-CUBE-TOF1	\odot		3.3.0 ~				
✓ Device TOF1_Applications	\odot		1.0.0				
Application	\odot		1.0.0 🤇	53L7A1_SimpleRanging			
> Board Support STM32Cube_Custom_BSP_		l i	1.0.0				
Board Extension 53L3A2		2	2.0.3				
Board Extension 53L5A1			1.0.5				
Board Extension 53L7A1	\odot		1.0.1	(☑)			
Board Extension 53L8A1	$\mathbf{\nabla}$		1.0.0				
Board Extension 53L1A2			1.0.2				
Board Extension 53L4A2			1.0.2				
Board Extension 53L4A1			1.0.2				
> Board Part Ranging		4	2.0.3				



The next step is to configure the hardware.

7. Configure the GPIOs for the application. Click on "System Core" then "GPIO".

*F401_VL53L5CX.ioc ×							
F401_VL53L5CX.ioc - Pinout & Configuration							
Pinout & Configuration							
Q							
Categories A->Z							
System Core 1	~						
► 1							
GPIO							
NVIC							
✓ RCC							
✓ SYS WWDG							

Figure 19. GPIO selection

For this particular application, configure the I2C_RST, LPn, PWR_EN, and INT pins.



8. Select the "PB3" pin by clicking on it.



Figure 20. PB3 pin selection

9. Make the PB3 a "GPIO_Output".



Figure 21. GPIO_output choice



10. Do the same for PB0, PB4, and PA4.





PB0 and PB4 have to be defined as "GPIO_output". PA4 has to be defined as "GPIO_EXT4". Consequently, add the GPIOs to the GPIO table as shown Figure 23. GPIO table.

Figure 23. GPIO table

🛛 GPIO 🛛 🛇 R	CC 🛛 🛇 SYS 🔄	🛛 USART 🛛 🛇 N	/IC				
Search Signals							
Search (Ctrl+F)						Show only M	odified Pins
Pin Name 🗢	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output	. User Label	Modified
PA4	n/a	n/a	External Interrup	No pull-up and no pull-down	n/a		
PA5	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LD2 [Gree	~
PB0	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low		
PB3	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low		
PB4	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low		
PC13-ANTI_TAMP	n/a	n/a	External Interrup	lo pull-up and no pull-down	n/a	B1 [Blue	~



11. Next, configure the GPIO output level, GPIO mode, GPIO pull-up/pull-down, and the user label. Click on PA4 and give it a "User Label".

Figure 24. PA4 configuration

Pin Name 🗢	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output	. User Label	Modified	
PA4	n/a	n/a	External Interrup	No pull-up and no pull-down	n/a	TOF_INT	V	
PA5	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LD2 [Gree	~	
PB0	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low			
PB3	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low			
PB4	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low			
PC13-ANTI_TAMP	n/a	n/a	External Interrup	No pull-up and no pull-down	n/a	B1 [Blue	~	
PA4 Configuration	:		External latercust M	nde with Dising adap triagge dat	ntion			
GPIO mode		Ľ	External interrupt wi	ode with Kising edge trigger dete	ection			
GPIO Pull-up/Pull	-down	[No pull-up and no pull-down					
User Label		i	TOF_INT					

Do the same for the other GPIOs as per Figure 25. GPIO pin names and configuration. For each GPIO, configure the:

- GPIO output level
- GPIO pull-up/pull-down
- user label

Figure 25. GPIO pin names and configuration

Saarch Signala							
Search (ChinP)	1					Stew only 1	Modified Pin
Pro Name **	Signal of Pin	GPID subst	GPID made	GPI0 PullupPull-town	Waxmum output	User Label	Mothet
PAI	nie	n/a	External Interrup	No pull-up and no pull-down	m'a	TOF_INT	5
PAS	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LD2 Green	5
PBO	n/a	High	Output Push Pull	No pull-up and no pull-down	Low	TOF PWR_	5
PB3	nía	Low	Output Push Pull	Pull-down	Low	TOF UC RST	
PBI	n/a	High	Output Push Pull	Pull-up	Low	TOF_LPn	2
PC13-AMTI TAMP	0/8	n/a	External Interrup	No pull-op and no pull-down	0/8	B1 (Blue Pa	

12. Activate the NVIC interrupt vector as shown below by clicking on "NVIC" and checking "Enabled".

Figure 26. NVIC interrupt activation

Configuration							
Group By Peripherals 1							
📀 SYS		USART					
S GPIO		🥺 Single Mapped Signals		RCC			
NVIC Interr	upt Table		tion Priority	Sub Priority			
EXTI line4 interrupt		(🗹)	()			
EXTI line[15:10] inter	rrupts	2 0	()			



- 13. Next, configure the I²C. Click on PB9 and change it to "I2C1_SDA". Click on PB8 and change it to "I2C1_SCL".
- Figure 27. PB9 and PB8 selection

14. Click on "Connectivity" and then on "I2C1".



Search Signala							
(Search (Chirt?)						Staw only	Modified Piru
Per Name 1	Signal on Pin	GPI0 suppr	GPID made	GPI0 PullupPullionn	Maximum output	User Label	Mothet
PAI	n/a	n/a	External Interrup	No pull-up and no pull-down	m/a	TOF_INT	15
PAS	n/a :	Low	Output Push Pull	No pull-up and no pull-down	Low	D2 Green	5
PBO	nia	High	Output Push Pull	No pull-up and no pull-down	Low	TOF PWR	53
PB3	nia	Low	Output Push Pull	Pull-down	Low	TOF UC RST	
PBI	n/a	High	Output Push Pull	Pull-up	Low	TOF_LPn	
PC13-ANTI TAMP	ala	n/a	External Internal	No pull-op and no pull-down	n/a	B1 (Blue Pa	

15. Enable I2C1 by clicking on "I2C".

Figure 29	9. I2C1 e	nabling
-----------	-----------	---------

Pinout & Configuration	Clo	ock Configuration		Pro
Q. V	•	I2C1 Mode ar M	ed Configuration	
System Core	> 0	C Disable Disable		~
Analog	<u> </u>	SMEus-Alett-mode		
Timers	>	SMDus-teo-wre-interface		
Connectivity	· · · · ·	Contr	Natios	



16. Set SCL and SDA in "Pull-up".

|--|

			I2C1 Mode and Configuratio Mode	on			🛄 Pino	ut view	US System
12C [[2C						~			
2000			Configuration			_			CC1_SDA
Parame Search Size	ter Settings 🛛 👁 Us	er Constants 📔 🥥 M	MC Settings 📔 👁 DMA Settin	igs 🥏 GPIC Settings		_	B1 (Blue PushButton)	BAT O	
Search (Ct	11+F)	0000	-000	00/0 Bull	Show only Mo	dified Pins	RCC_OSC32_OUT RCC_OSC32_OUT RCC_OSC_IN		
P68 P69	I2C1_SCL I2C1_SCL I2C1_SDA	n/a n/a	Alternate Function Open Drain Alternate Function Open Drain	Pull-up Pull-up	Very High Very High		RCC_OSC_OUT	11	

17. Configure the I²C speed in "Fast Mode".

Figure	31.	l ² C	speed	mode	config	uration
--------	-----	------------------	-------	------	--------	---------

		Co	nfiguration		
Reset Configuration					
Parameter Settings	🥝 User Constants	⊗ NVIC Settings	🥝 DMA Settings	🥺 GPIO Settings	
Configure the below parame	eters :				
Q Search (Ctrl+F)) ()				
✓ Master Features					
I2C Speed Mode)		Fast Mode		
I2C Clock Speed	d (Hz)		400000		
Fast Mode Duty	Cycle		Duty cycle Tlow/Thigh = 2		
✓ Slave Features					
Clock No Stretch Mode			Disabled		
Primary Address	s Length selection		7-bit		
Dual Address Ad	Dual Address Acknowledged				
Primary slave ad	Primary slave address				
General Call add	lress detection		Disabled		



18. Next, tell the application where the dedicated pins are. Click on "Middleware and Software Packs", then on "X-CUBE-TOF1", and choose "Platform Settings".

Pinout & Configuration	Clock
	✓ Software
Categories A->Z	
Computing >	Board Extension 53L7A1
Middleware and Software Packs	Device TOF1 Applications
¢ FATES	
FREERTOS	
N I-CUBE-Cesium	
U-CUBE-UNISONRTOS	
U-CUBE-embOS	
U I-CUBE-wolfSSL	
🕒 I-Cube-SoM-uGOAL	
LIBJPEG	
MBEDTLS	
PDM2PCM	
USB_DEVICE	
USB_HOST	
X-CUBE-AI	
X-CUBE-ALGOBUILD	
X-CUBE-ALS	
X-CUBE-AZRTOS-F4	
X-CUBE-BLE1	
X-GUBE-BLEZ	
♦ X-GUBE-BLEMGR	
X-GUBE-DISPLAY	
X-GUDE-EEPRMAT	
♦ X-GUBE-GNSST	
the viewer week	
the victor victor	
A-CODE-SEXSZLPT	
V OLIDE TOPI	

Figure 32. Platform settings

19. Choose "Board Extension" and "Device TOF1 Applications".

Figure 33. Board and application choice

F401_VL53L7CX.ioc × F401_VL53L7CX.ioc - Pinout & Configuration						
Pinout & Cor	nfiguration	Clock Configuration	Project Man			
		✓ Software Packs	✓ Pinout			
Q ~	٢	STMicroelectronics.X-CUBE	-TOF1.3.3.0 Mode and Configuration			
Categories A->Z			Mode			
Multimedia	> (doard Extension 53L7A1				
Computing	>	Device TOF1 Applications				
Middleware and Software Pa	acks 🗸					
\$						
FATES						
I-CUBE-Cesium						
UCUBE-UNISONRTOS						



20. Click on "Platform Settings" and assign pins as per Figure 34. Hardware and software pin assignment.

Figure 3	4. Hardw	are and s	oftware r	oin assi	anment
i igui o o					ginnoin

Configuration						
Reset Configuration						
Platform proposal Application						
Name	IPs or Components		Found Solutions	BSP API		
53L7A1_PWR_EN_C	GPIO:Output	\sim	PB0 [TOF_PWR_EN]	∨ Unknown		
TOF_INT_PIN	GPIO:EXTI	\sim	PA4 [TOF_INT]	√ Unknown		
53L7A1_I2C_RST_C	GPIO:Output	~	PB3 [TOF_I2C_RST]	√ Unknown		
53L7A1_LPn_C	GPIO:Output	\sim	PB4 [TOF_LPn]	√ Unknown		
BSP	•					
Name	IPs or Components	1	Found Solutions	BSP API		
53L7A1 BUS IO driver	12C:12C ~		2C1	✓ BSP_BUS_DRIVER		
BSP BUTTON	GPIO:EXTI ~	E	PC13-ANTI_TAMP [B1 [Blue Pi	ushButton]] V BSP_COMMON_DRIVER		
BSP USART	USART:Asynchronous $$	а I	JSART2	V BSP_COMMON_DRIVER		
				<u> </u>		

21. Verify the green tick.







22. Next, generate and build the application. Click on the generate icon, then the hammer icon.

IDE L7CX-workspace - F401_VL53L7CX/Core/Src/main.c - STM32CubeIDE							
File Edit Source Refactor ² Navigate Search Project Run W	'indow Help _1						
🔁 🖬 🕼 🕲 T 🔦 T 🛍 🕯 🖬 📽 T 🗳 T 🥸 T 🔅	• 🖸 • 💁 • 🛯 🗙 🚯 🎾 🛷 • 💷 🖻						
🖹 Project Explorer 🗙 🔽 🗐 🛱	F401_VL53L7CX.ioc 🗈 main.c 🗙						
✓ IDE F401 VL53L7CX	1 /* USER CODE BEGIN Header */						
> 🔊 Includes	2⊖ /**						
V 🛱 Core	3 *************************************						
	4 * @file : main.c						
	5 * @brief : Main pr						
	6 ************************************						
> .c main.c	7 Wattention						
> .c stm32f4xx_hal_msp.c	9 * Convright (c) 2023 STMicr						
> 🖻 stm32f4xx_it.c	10 * All rights reserved.						
> stm32f4xx_nucleo_bus.c	11 *						
> 🖻 syscalls.c	12 * This software is licensed						
> c sysmem.c	13 * in the root directory of						
> c system stm32f4xx.c	14 * If no LICENSE file comes						
Startun	15 *						
	16 ************************************						
	17 */						
	188 /* USER CODE END Header */						
H4U1_VL53L/CX.IOC	19 /* Includes						
STM32F401RETX_FLASH.Id	20 #include "app tof h"						
STM32E401RETX RAM.Id	zi #inciuue app_col.n						

Figure 36. Generate and build the code

23. Next, run the debugger. Click on the bug.





You may have to upgrade your STLINK. If so:

- Open in upgrade mode
- Click on "Upgrade"
- Close the upgrade window
- Restart the debugger

Figure 38. Upgrade STLINK





24. Start a serial port terminal emulation by adapting the baud rate as per Figure 11. Tera Term serial port setup. The result should look like Figure 39. Results.

Figure 39. Results

💹 COM11 - Tera Term VT File Edit Setup Control Window Help												
53	53L7A1 Simple Ranging demo application											
Use the following keys to control application 'r' : change resolution 's' : enable signal and ambient 'c' : clear screen Call Forwart												
	Distance [mm]: Status											
Signal [kcps/spad] : Ambient [kcps/spad]												
	X	:	××	110 855	:	Ø 2	118 452	:	Ø 5	1900 8	:	Ø 6
	X	:	×××	119 854	:	Ø 5	124 742	:	0 3	1926 5	:	0 4
	2028 3	:	0 7	122 1011	:	Ø 3	127 858	:	0 2	<mark>136</mark> 370	:	0 4
	2001 5	=	0 7	119 877	:	0 4	120 820	:	Ø 2	1958 3	:	Ø 3

The above steps for the VL53L7CX can be used for other sensors. The pin connection is described in the X_CUBE_TOF1 .chm file in the "Documentation" folder.

The above steps for the VL53L7CX can be used for other sensors. The pin connection is described in the X_CUBE_TOF1.chm file in the "Documentation" folder.



4.1.2.2 How to generate the 53L7A1_ThresholdDetection example with the STM32CubeMX

The steps for generating the 53L7A1_ThresholdDetection example with the STM32CubeMX are almost identical to those in Section 4.1.2.1 How to generate the SimpleRanging example with the STM32CubeMX. Steps 6, 20, and 24 are different. They are described below respectively.

Step 6: Choose the "ThresholdDetection Application".

Figure 40. ThresholdDetection application

Software Packs Component Selector Packs							
🗮 🗞 🕕 >							
Pack / Bundle / Component	Status	Version	Selection				
✓ STMicroelectronics.X-CUBE-TOF1	\odot	3.3.0 ~					
✓ Device TOF1_Applications	\odot	1.0.0					
Application	\odot	1.0.0	53L7A1_ThresholdDetection ~				
> Board Support STM32Cube_Custom_BSP_D		1.0.0					
Board Extension 53L3A2		2.0.3					
Board Extension 53L5A1		1.0.5					
Board Extension 53L7A1	\odot	1.0.1					
Board Extension 53L8A1		1.0.0					
Board Extension 53L1A2		1.0.2					
Board Extension 53L4A2		1.0.2					
Board Extension 53L4A1		1.0.2					
> Board Part Ranging		2.0.3					

Step 20: Assign pins.

Configuration							
Reset Configuration	n						
Parameter Setting	s ⊗ Platform Settings						
Platform proposal							
Name	IPs or Components			Found Solutions		BSP API	
53L7A1_PWR_EN_C	GPIO:Output	\sim		PB0 [TOF_PWR_EN]	,	- Unknown	
TOF_INT_PIN	GPIO:EXTI	\sim		PA4 [TOF_INT]		- Unknown	
53L7A1_I2C_RST_C	GPIO:Output	\sim		PB3 [I2C_RST_C]		- Unknown	
53L7A1_LPn_C	GPIO:Output	\sim		PB4 [LPn]	,	Unknown	
BSP							
Name	IPs or Components			Found Solutions	BSP AP	ı 📄	
53L7A1 BUS IO driver	r 12C:12C			✓ II I2C1	V BSP_BUS	DRIVER	
BSP USART	USART:Asynchronous			V USART2	V BSP_COM	MON_DRIVER	
11			_				

Figure 41. Pin assignment for the ThresholdDetection example



• Step 24: Start a serial port terminal emulation with a baud rate as per Figure 11. Tera Term serial port setup. The result should look like Figure 42. Result of the ThresholdDetection example.

Figure 42. Result of the InfesholdDetection exam
--

531	L7A1 T	hresh	old De	tection	dem	appli	cation					
Cel	11 For 370	mat : Dista :	ince [m	m]: 1870		0	Status	:		1895	:	0
	1876	:	0	1934	:	0	1931	:	0	1906	:	Ø
	1970	:	ø	1947	:	Ø	1952	:	Ø	1909	:	0
	1952	:	ø	1956	:	0	1930	:	0	1912	:	0

An interrupt occurs if the target distance (d) = [200 mm and 600 mm] as shown in Figure 43. Interrupt.

Figure 43. Interrupt



Other examples can be found for the different sensors. Do not hesitate to check in the "Examples" folder.



4.2 SimpleRanging examples using an STM32 Nucleo board and a SATEL breakout board

For each Time-of-Flight sensor present in the X-CUBE-TOF1 pack, a simple ranging example is delivered in the "CUSTOM" folder. This example is for use with a NUCLEO-F401RE or a NUCLEO-L476RG as per Figure 44. Customer folder of the SimpleRanging example using a SATEL breakout board.

Figure 44. Customer folder of the SimpleRanging example using a SATEL breakout board



4.2.1 Loading a prebuilt binary file

Use the same steps given in Section 4.1.1 Loading a prebuilt binary file by loading the binary file in the CUSTOM/ <Sensor_Name>_SimpleRanging/Binary folder (see Figure 45. Binary folder for the SimpleRanging example using a SATEL breakout board).



Figure 45. Binary folder for the SimpleRanging example using a SATEL breakout board



4.2.2 Configuring the STM32CubeMX for the SimpleRanging example

The steps for configuring the STM32CubeMX for the SimpleRanging example using an STM32 Nucleo board and a SATEL board are almost identical to those in Section 4.1.2.1 How to generate the SimpleRanging example with the STM32CubeMX. Steps 6 and 19 are different. They are described below respectively.

 Step 6: Choose the right application. It is called <Sensor_Name>_SimpleRanging. For the current example with a VL53L7CX sensor, choose "VL53L7CX_SimpleRanging".

Pack / Bundle / Component	Status	Version	Selection
 STMicroelectronics.X-CUBE-TOF1 	\odot	3.3.0 ~	
Device TOF1_Applications	\oslash	1.0.0	
Application	\odot	1.0.0	VL53L7CX_SimpleRanging
✓ Board Support STM32Cube_Custom_BSP_D	\odot	1.0.0	
Custom / RANGING_SENSOR	\odot	1.0.0	~
Board Extension 53L3A2		2.0.3	
Board Extension 53L5A1		1.0.5	
Board Extension 53L7A1		1.0.1	
Board Extension 53L8A1		1.0.0	
Board Extension 53L1A2		1.0.2	
Board Extension 53L4A2		1.0.2	
Board Extension 53L4A1		1.0.2	
✓ Board Part Ranging	\odot	2.0.3	
VL53L3CX		2.0.3	
VL53L5CX		1.0.4	
VL53L7CX	\odot	1.0.1	V
VL53L8CX		1.0.0	
VL53L1CB		1.0.2	
VL53L4CX		1.0.1	
VL53L4CD		1.0.1	

Figure 46. SimpleRanging application example for the SATEL breakout board

Step 19: Choose "Board Extension" and "Device TOF1 Applications".

Figure 47. Choice of boards and application



Revision history

Table 1. Document revision history

Date	Version	Changes
27-Apr-2023	1	Initial release
21-Jun-2023	2	Section 3.1 Hardware description: added a preamble. Updated Figure 4. Example of an X-NUCLEO-53L7A1 expansion board with a cover glass.



Contents

1	Acro	onyms a	Ind abbreviations	2								
2	X-CUBE-TOF1 software expansion for STM32Cube											
	2.1	Overview										
	2.2	Architecture										
	2.3	Folder structure										
	2.4	APIs .		4								
3	Syst	em seti	ıp guidelines	5								
	3.1	Hardw	are description	5								
		3.1.1	STM32 Nucleo	5								
		3.1.2	X-NUCLEO expansion boards	6								
		3.1.3	SATEL breakout boards	7								
	3.2	Software description										
	3.3	.3 Hardware setup										
		3.3.1	Setup using an STM32 Nucleo and an X-NUCLEO expansion board	8								
		3.3.2	Setup using an STM32 Nucleo board and a SATEL breakout board	9								
4	Sample application examples1											
	4.1	Ranging singular form using an STM32 Nucleo board and an X-Nucleo sion board	11									
		4.1.1	Loading a prebuilt binary file	11								
		4.1.2	Configuring the STM32CubeMX project	14								
	4.2	Simple	Ranging examples using an STM32 Nucleo board and a SATEL breakout board	28								
		4.2.1	Loading a prebuilt binary file	28								
		4.2.2	Configuring the STM32CubeMX for the SimpleRanging example	29								
Rev	ision	history		30								

IMPORTANT NOTICE - READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved