

XM125 Software

User Guide

XM125 Software

User Guide

(())

Author: Acconeer AB

Version:a121-v1.3.0

Acconeer AB October 6, 2023



Contents

1	Acconeer SDK Documentation Overview	3
2	Introduction	4
3	Installing Software Image 3.1 Windows COM port drivers 3.2 Flash Over UART Using STM32CubeProgrammer 3.2.1 Boot the XM125 in bootloader mode 3.2.2 Program the XM125 3.3 Flash Over UART Using stm32loader	5 5 5 5 5 6
4	 4.1 Using a Debugger	7 7 7 7 8 10 10 10
5	Troubleshooting and FAQ 5.1 LTO wrapper fails 5.2 Link errors in sysmem.c	
6	Disclaimer	12

(()

To better understand what SDK document to use, a summary of the documents are shown in the table below.

Name	Description	When to use
	RSS API documentation (html)	L
rss_api	The complete C API documentation.	- RSS application implementation - Understanding RSS API functions
	User guides (PDF)	
	Describes the Acconeer assembly	- Bring-up of HW/SW
A121 Assembly Test	test functionality.	- Production test implementation
A121 Breathing	Describes the functionality of the	- Working with the Breathing
Reference Application	Breathing Reference Application.	Reference Application
A121 Distance Detector	Describes usage and algorithms of the Distance Detector.	- Working with the Distance Detector
A121 SW Integration	Describes how to implement each integration function needed to use the Acconeer sensor.	- SW implementation of custom HW integration
A121 Presence Detector	Describes usage and algorithms of the Presence Detector.	- Working with the Presence Detector
A121 Smart Presence	Describes the functionality of the	- Working with the Smart Presence
Reference Application	Smart Presence Reference Application.	Reference Application
A121 Sparse IQ Service	Describes usage of the Sparse IQ Service.	- Working with the Sparse IQ Service
A121 Tank Level	Describes the functionality of the	- Working with the Tank Level
Reference Application	Tank Level Reference Application.	Reference Application
A121 STM32CubeIDE	Describes the flow of taking an Acconeer SDK and integrate into STM32CubeIDE.	- Using STM32CubeIDE
A121 Raspberry Pi Software	Describes how to develop for Raspberry Pi.	- Working with Raspberry Pi
A121 Ripple	Describes how to develop for	- Working with Ripple
A121 Kipple	Ripple.	on Raspberry Pi
XM125 Software	Describes how to develop for XM125.	- Working with XM125
I2C Distance Detector	Describes the functionality of the	- Working with the
	I2C Distance Detector Application.	I2C Distance Detector Application
I2C Presence Detector	Describes the functionality of the I2C Presence Detector Application.	- Working with the I2C Presence Detector Application
	Handbook (PDF)	
Handbook	Describes different aspects of the Acconeer offer, for example radar principles and how to configure	- To understand the Acconeer sensor - Use case evaluation
	Readme (txt)	
[README	Various target specific information and links	- After SDK download
	1	1

Table 1: S	DK document	overview.
------------	-------------	-----------

2 Introduction

The Acconeer Software Development Kit (SDK) enables customers to develop their own software that can be executed on the module. This enables full control of all the peripherals and to maximize the performance and power consumption for a specific use case.

The SDK comes with a number of example applications that can be used as a starting point when developing your own application. These applications can be downloaded and executed using the methods described in "Installing Software Image" at page 5.

When developing your own application we recommend that you setup a development environment as described in "Setting up a Development Environment" at page 7.

This guide has been verified in Ubuntu 20.04 and Windows with STM32CubeIDE 1.11.2 and STM32CubeMX 6.8.0

3 Installing Software Image

The XM125 uses the STM32L431 MCU which contains a ROM bootloader. The MCU is configured to enable the bootloader during manufacturing.

Another option is to use a SWD debugger, this requires additional hardware which is suitable when developing your own applications.

3.1 Windows COM port drivers

If running on Windows, you might need to install a driver for the USB to UART Bridge. It can be downloaded here.

3.2 Flash Over UART Using STM32CubeProgrammer

Download and install STM32CubeProgrammer.

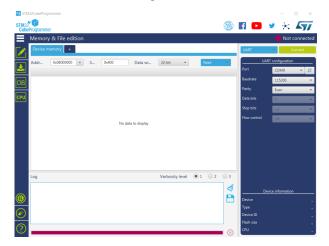
3.2.1 Boot the XM125 in bootloader mode

- 1. Connect the XE125 to your PC with a USB-C cable to the USB connector
- 2. Press and hold the "DFU" button on the board
- 3. Press the "RESET" button (still holding the "DFU" button)
- 4. Release the "RESET" button
- 5. Release the "DFU" button

Your XM125 device is now in "DFU" mode waiting for a software upgrade procedure to be started.

3.2.2 Program the XM125

1. Start the STM32CubeProgrammer



- 2. Select correct port to the right. E.g. COM9.
- 3. Press "Connect" in the upper right corner
- 4. Press The "+" button and the "Open file"



- 5. Browse to and select the binary you like to program, e.g. "example_service.bin"
- 6. Press the "Download" button. The green progress bar in the bottom indicates the progress

Memory & F	le edition						Connected
Device memory	acc_modul	e_server_xm18	i2.bin × 👘			UART	• Disconnec
Addr 0x0	٠	S 0x40	0 D	ata wi 3	2-bit • Downlos		UART configuration
Address	0	4	8	с	ASCII	Port	COM9 -
0x00000000	20009000	080112C1	08004839	0800484B	А́9нкн	audrate	115200
0x00000010	00000000	00000000	00000000	00000000		Parity	Fore
0x00000020	00000000	00000000	00000000	08004845	ЕН	U	
0x00000030	00000000	00000000	0800483F	08003BDD	Ŷ;	Data bits	8
0x00000040	08011311	08011311	080047D9	08011311	ÛG	Stop bits	1.0
0x00000050	08011311	08011311	08003BB1	08011311	±;	Flow control	01
0x0000060	08011311	08003BA1	08003891	08011311			94
0x00000070	08011311	08011311	08011311	08011311			
0x0000080	08011311	08011311	08011311	08011311			
0x00000090	08011311	08011311	08011311	08011311			
0x000000A0	08011311	08011311	08011311	08003959	Y9		
0x000000B0	08011311	08011311	08011311	4C06B510	µ.L		
0x000000C0	28007823	4805D107	D0022800	E0004804	#x.+.Ň.K.+.Đ.H.à	~	
Log				v	erbosity level 💿 1 💿 2	3	
12112122 1 8	gnentiul: a	aaress= uxu,	size= uxic	584	-135.57		
17:15:59 : 5 17:17:15 : R 17:17:15 : N 17:17:15 : N 17:17:15 : 5	mber of seg	ments: 1	nies (acc_modi	ine_server_x		4	Device information
17:17:58 : No 17:17:58 : 0	smory Progra	mming	size= 0x10	conver yet2	2 bio	Device	STM32G07x/STM32
17:17:58 F 17:17:58 S			xm132.bin	20 407_0015		Type	
17:17:58 : A 17:17:58 : A 17:17:58 : E	idness : 0x0	8000000				Device ID	
17:17:58 E	asing memor	nal memory s	ectors [0 5	5]		Flash size	12
12112159 1 0	white the second second	concess:				C011	Cortex-
							Contex-

- 7. Once programming is complete press the "Disconnect" button
- 8. Press the "RESET" button or do a power cycle to start the embedded application

3.3 Flash Over UART Using stm32loader

The stm32loader is a python program. See pypi.org/project/stm32loader/ for more information.

Install it using "pip install stm32loader"

- 1. Set the XM125 into bootloader mode, see above for how to do this
- 2. Program the device with "stm32loader -p /dev/ttyUSB0 -e -w -v example_service.bin". Make sure to specify correct port.
- 3. Press "RESET" or power cycle the device to start the embedded application

4 Setting up a Development Environment

In order to develop your own applications you need to set up a development environment. The XM125 is based on a STM32L431 SoC by STMicroelectronics.

4.1 Using a Debugger

In order to debug your applications it is recommended to use a SWD debugger. We recommend that you use a SEGGER JLink debug probe e.g. J-Link BASE Compact or an ST-LINK debugger.



Figure 1: J-Link Base Compact

The J-Link BASE Compact can be used to set breakpoints and single step the program in an easy way.

4.2 Building From the Command Line

All example applications can be built from the command line using "make".

- 1. Download the STM32Cube MCU Package for STM32L4 series (version 1.17.0) from www.st.com.
- 2. Extract the archive into a folder, e.g. "/home/acconeer/sdk/"
- 3. Download "GCC ARM Embedded 9-2020-q2-update" from developer.arm.com.
- 4. Extract the archive into a folder, e.g. "/home/acconeer/compilers/"
- 5. Download and extract the Acconeer SDK zip file, e.g. "/home/acconeer/xm125/"

The above will compile all example applications which can be downloaded to the target using any of the methods described in "Installing Software Image" at page 5

4.2.1 Download Software Using a J-Link

You can flash the software using a J-Link debugger from the command line. First install the "J-Link Software and Documentation Pack" from www.segger.com.

```
$ make flash_jlink_example_servic
```

4.3 STM32CubeIDE

If you prefer to use an integrated development environment we recommend that you use the STM32CubeIDE together with a SEGGER J-Link debug probe or an ST-Link debugger. The Acconeer SDK for XM125 includes an STM32CubeMX project file, 'xm125.ioc'. From this file it's possible to generate an STM32CubeIDE project directly from the SDK.

- 1. Download the latest version of STM32CubeMX from www.st.com.
- 2. Extract the archive into a temporary folder, e.g. "/home/acconeer/sdk/temp"
- 3. Run the installer for your preferred OS from "/home/acconeer/sdk/temp"
- 4. Download and install the latest version of STM32CubeIDE for your preferred OS from www.st.com.
- 5. Download and extract the Acconeer SDK zip file, e.g. "/home/acconeer/acconeer_xm125/"

- 6. Start STM32CubeMX, then select "File/Load Project..." and browse to the folder where you unpacked the zip file, then select "xm125.ioc" and click on "Open"
- 7. Select the Project Manager tab and change "Toolchain / IDE" to "STM32CubeIDE" and press "GENERATE CODE".
- 8. Select "Open Project" in the dialog to open the newly created project in STM32CubeIDE.

	ऽग	M32CubeMX xm125.ioc*: STM	32L431CBYx		8
STM:20	File Window	Help		i 💿 📑 🕒 🄰 🗄	k (77
Home STM32L431CBYx	xm125.ioc - Project Manager	\rightarrow		GENERATE CODE	
Pinout & Configurat	ion Clock Config	uration	Project Manager	Tools	
Project	Project Settings Project Name Project Location Application Structure	ym125 /home/acconeer Basic		Do not generate the ma	
Code Generator	Toolchain Folder Location	home/acconeer/xm125/	Generate Under Root		
Advanced Settings	Linker Settings Minimum Heap Size Minimum Stack Size	0x200 0x1400			
	Thread-safe Settings Cortex-M4NS				
	Thread-safe Locking Strategy	Default – Mapping suitable str	itegy depending on RTOS selection.		
	Mcu and Firmware Package Mcu Reference	STM32L431CBVx			
	Firmware Package Name and Version	STM32Cube PW_L4 V1:17:1	Migrate to	the latest supported Firmware version	n -
	Use Default Firmware Location				
	Firmware Relative Path	home/fredrik/STM32Cube/Rep	ository/STM32Cube_PW_L4_V1:17:1	Browse	

4.3.1 Configuring Project for Acconeer Software

Now when you have an STM32CubeIDE project you need to modify it to include the Acconeer SDK components.

		wor	kspace - STI	M32CubeIDE		- • •	3
File Edit Source Refactor Navigate	Search Project Ru	un Window	Help				
🗂 = 🖂 🐁 = 🗞 = 🙈 🖬 =	🚳 + 💽 + 🕞 + 🗄	* • 0 •	9 L X	1 🍐 1 🍅 🛷	-	◎ Ⅲ 〒 创 + 利 + や ヴ や + ウ + - 団 ③	
						Q. i 😰 🖪	à
🍐 Project Explorer 🛪 👘 🖽						😁 🗖 📴 Outli 🗙 🛞 Build 😁 🗖	3
Imit23 Imit23 Imit23 Imit24 Imit24						There is no active editor that provides an outline.	
	🖹 Problems 🗙 🧟	Tasks 📮 Cor	nsole 🛅 Pro		8	🚡 Build Analyzer 🗙 🛓 Static Stack Analyzer 📑 🕴 🖱 🗖	3
	0 items					Memory Regions Memory Details	
	Description			Resourc	ce	Region Start address End address Size Fr	24
0 items selected							

Source Files The SDK includes many examples and applications. STM32CubeIDE will try to compile and link all source files in the SDK which will cause "multiple definition" errors when linking. To avoid this you should exclude the source files not needed from the build. Select all source files starting with "applications", "example" and "use_cases" except the file you want to use, right click and select "Resource Configurations \rightarrow Exclude from build".

For building "examples", only the example source file is needed. For "use_cases", the use case source file and "algorithms" are needed.

When building i2c-examples multiple files are needed.

For i2c_distance_detector.c the following files are needed:

- acc_reg_protocol.c
- distance_reg_protocol.c
- distance_reg_protocol_access.c

- i2c_application_system_stm32.c
- i2c_distance_detector.c

For i2c_presence_detector.c the following files are needed:

- acc_reg_protocol.c
- i2c_application_system_stm32.c
- i2c_presence_detector.c
- presence_reg_protocol.c
- presence_reg_protocol_access.c

Header-files You have to manually add the "Inc" folder to the project paths:

- 1. Select your project in the "Project Explorer"
- 2. Go into "Project \rightarrow Properties \rightarrow C/C++ General \rightarrow Paths and Symbols \rightarrow Includes"
- 3. Press "Add..." and then "Workspace..."
- 4. Select the "Inc"-folder in your project

Libraries In order to set the path for the libraries, do the following:

- 1. Select your project in the "Project Explorer"
- 2. Go into "Project \rightarrow Properties \rightarrow C/C++ General \rightarrow Paths and Symbols \rightarrow Library Paths"
- 3. Press "Add..." and then "Workspace..."
- 4. Select the "lib"-folder in your project

	Properties for xm125	0 😣
type filter text	Paths and Symbols	□ ▼ □} ▼ 8
 Resource Builders C/C++ Build 	Configuration: Debug [Active]	nfigurations
 ▼ C/C++ General ▶ Code Analysis 	▲Libraries ●Library Paths ●Source Location ●Ref	erences 🕨
Documentation File Types	acc_rf_certification_test_a121	Add
Formatter Indexer	n [≜] acconeer_a121	Edit
Language Mapping Paths and Symbols		Delete
Preprocessor Inclu CMSIS-SVD Settings		Export
Project References Run/Debug Settings		Move Up
Kun Debug Settings	Using relative paths is ambiguous It can cause unexpected effects.	Move Down
	Restore Defaults	Apply
?	Cancel	ply and Close

Once the path is set, you can add the specific libraries by the following:

- 1. Go into "Project \rightarrow Properties \rightarrow C/C++ General \rightarrow Paths and Symbols \rightarrow Libraries"
- 2. Click "Add..."
- 3. Enter "acconeer_a121"
- 4. Click "OK"

If you want to add the "acc_rf_certification_test_a121" library, simply repeat the procedure above and exchange "acconeer_a121" for "acc_rf_certification_test_a121". Make sure that the additional library is being added before the "acconeer_a121"-library by moving "acconeer_a121" down using the "Move Down" button when "acconeer_a121" is selected.

4.3.2 Project Settings

Select GNU99 as language standard in "Project \rightarrow Properties \rightarrow C/C++ Build \rightarrow Settings \rightarrow Tool Settings \rightarrow MCU GCC Compiler \rightarrow General".

	Properties for xm 125 🔹 🛽 😣
type filter text	Settings Image Configurations Configuration: Debug [Active] Image Configurations Image Configuration: Image Configurations Image Configurations Image Configuration: Image C
 ▶ Resource Builders ▼ C/C++ Build 	Configuration: Debug [Active]
Build Variables Environment Logging Stithos C/C++ General CMSIS-SVD Settings Project References Run/Debug Settings	MCU Toolchuin MCU Settings MCU Settings MCU Settings MCU Settings MCU CCC Assembler Gereral Mound Settings Mo
3	Cancel Apply and Close

4.3.3 Running the Program

Build the software by pressing "Ctrl-B" and then start debugging by right-clicking on the project "xm125 \rightarrow Debug As \rightarrow STM32 Cortex-M C/C++ Application". This will open the "Debug Configurations" dialog and there you can choose which debugger to use, "Debugger \rightarrow Debug Probe", either ST-LINK or SEGGER J-LINK. Click "Debug", this will automatically flash the XM125 and execute the program until the "main()" function.

	Debug	Configurations			
eate, manage, and run configuratio	ns		Ŕ		
2 2 1 X 1 X 1	Name: xm125		(
type filter text	🖹 Main 🏶 Debugger 🕨 Startup 🂱 Source	ce 🔟 Common			
C/C++ Application	Project:				
C/C++ Attach to Application	xm125				
C/C++ Postmortem Debugger	C/C++ Application:				
GDB Hardware Debugging	Debug/xm125.elf	Sear	h Project Browse		
🕏 Launch Group	Build (if required) before launching				
STM32 C/C++ Application	Build Configuration: Select Automatical	llv	•		
STM32 Cortex-A Remote Application		Disable auto build			
	 Enable auto build Use workspace settings 	Configure Workspace Settings			
ter matched 9 of 9 items			Revert Apply		

4.3.4 Debug Output

Debug logs will be outputted on UART2 using a baud rate of 921600.

5 Troubleshooting and FAQ

5.1 LTO wrapper fails

When using STM32CubeIDE for Windows there is a problem with the LTO wrapper. Therefore you need to explicitly disable LTO (link-time optimizations):

- 1. Go to "Project \rightarrow Properties \rightarrow C/C++ Build \rightarrow Settings \rightarrow Tool Settings \rightarrow MCU GCC Linker \rightarrow Miscellaneous \rightarrow Other flags".
- 2. Add "-fno-lto"

Properties for xm125			– o x
type filter text	Settings		← → ⇒ → 8
 > Resource Builders < C/C++ Build Build Variables Environment Logging Settings > C/C++ General CMSIS-SVD Settings 	MCU Toolchain MCU Settings	Build Artifact 📷 Binary Parsers 🧿 Error Parsers ther flags notice	 Manage Configurations ・ ・
Project References Run/Debug Settings	MCU Post build outputs Would outputs Would outputs General Debugging Preprocessor Include paths Miscellaneous Would CC Compiler General Debugging Preprocessor Include paths MCU GCC Dompiler Compiler Debugging Preprocessor Include paths MCU documents MCU documents Include paths MCU documents Include paths MCU documents Include paths MCU documents MCU documents Include paths MCU documents MCU d	19-110	
	[™] Warnings [™] Miscellaneous [™] MCU GCC Linker [™] MCU GCC Linker [™] General [™] Libraries [™] Miscellaneous [™] [™] Miscellaneous [™]	dditional object files	ର ୟ ର ଚ୍ୟା <u>ହା</u>
?			Apply and Close Cancel

5.2 Link errors in sysmem.c

When using STM32CubeIDE for Windows there might be a problem with the tool not finding the linker script which will lead to linker errors in system.c. Therefore you need to change the path to the Linker script:

- 1. Go into "Project \rightarrow Properties \rightarrow C/C++ Build \rightarrow Settings \rightarrow Tool Settings \rightarrow MCU GCC Linker \rightarrow General".
- 2. Press "Linker Script (-T) \rightarrow Browse" and find the file "STM32L431CBYx_FLASH.ld" from the project

Iter text Settings		← → ⇒ →
ource liders 2++ Build Build Variables Environment Logging Settings -+ - General SIS-SV0 Settings jet References VDebug Settings WD boug Settings WD Settings WD Settings WD Settings WD Settings WD Settings WD MCU Settings WD Sett	s 😤 Build Artifact 🗟 Binary Parsers 🥝 Error Parsers Linker Script (-T) C:\Userx\Public\xm125\STM32L431CBYx_FLASH.ld	V Manage Configurations Browse V

6 Disclaimer

The information herein is believed to be correct as of the date issued. Acconeer AB ("Acconeer") will not be responsible for damages of any nature resulting from the use or reliance upon the information contained herein. Acconeer makes no warranties, expressed or implied, of merchantability or fitness for a particular purpose or course of performance or usage of trade. Therefore, it is the user's responsibility to thoroughly test the product in their particular application to determine its performance, efficacy and safety. Users should obtain the latest relevant information before placing orders.

Unless Acconeer has explicitly designated an individual Acconeer product as meeting the requirement of a particular industry standard, Acconeer is not responsible for any failure to meet such industry standard requirements.

Unless explicitly stated herein this document Acconeer has not performed any regulatory conformity test. It is the user's responsibility to assure that necessary regulatory conditions are met and approvals have been obtained when using the product. Regardless of whether the product has passed any conformity test, this document does not constitute any regulatory approval of the user's product or application using Acconeer's product.

Nothing contained herein is to be considered as permission or a recommendation to infringe any patent or any other intellectual property right. No license, express or implied, to any intellectual property right is granted by Acconeer herein.

Acconeer reserves the right to at any time correct, change, amend, enhance, modify, and improve this document and/or Acconeer products without notice.

This document supersedes and replaces all information supplied prior to the publication hereof.

