Firmware Options and Upgrade Methods

RM126x LoRaWAN Module

Version 2.0

Revision History

Version	Date	Notes	Contributors	Approver
1.0	15 August 2023	Initial Release	Erik Lins	Senthooran Ragavan
1.1	22 August 2023	Removed mentioning of separate bootloader images	Erik Lins	Senthooran Ragavan
2.0	22 Apr 2025	Rebranded to Ezurio	Sue White	Dave Drogowski

Contents

1 Introduction

The RM126x LoRaWAN modules come with a bootloader and AT Interface application preloaded. The AT Interface exposes AT commands for LoRaWAN functions over the module's UART interface. Even though the module is ready-to-use, at some point customers might want to upgrade the application firmware to a later version.

This firmware update user guide shows and explains all available firmware options and upgrade methods for our RM126x LoRaWAN modules. For demonstration purposes one RM126x development board will be flashed and programmed via:

- **Serial DFU (UART) using uart_dfu application** *Recommended for power users and production use.*
- **Simplicity Studio 5 & Simplicity Commander (SWD)** *Recommended for developer and production use.*
- **SEGGER J-Link Commander (SWD)** *Recommended for developer and production use.*

Note(s) This document does not cover any C code development guidance. For more information, refer to the induvial User Guide documents available under https://www.ezurio.com/wireless-modules/lorawan-modules-solutions/rm126x-ultra-low-power-lorawan-a-b-c-module (RM126x).

2 Requirements

This firmware update user guide is applicable to the following part number(s):

RM126x Development Kits

- (453-00140-K1) RM1261 development board with external antenna (868MHz frequency band, 15dBm output power)
- (453-00139-K1) RM1262 development board with external antenna (915MHz frequency band, 22dBm output power)

RM126x Modules

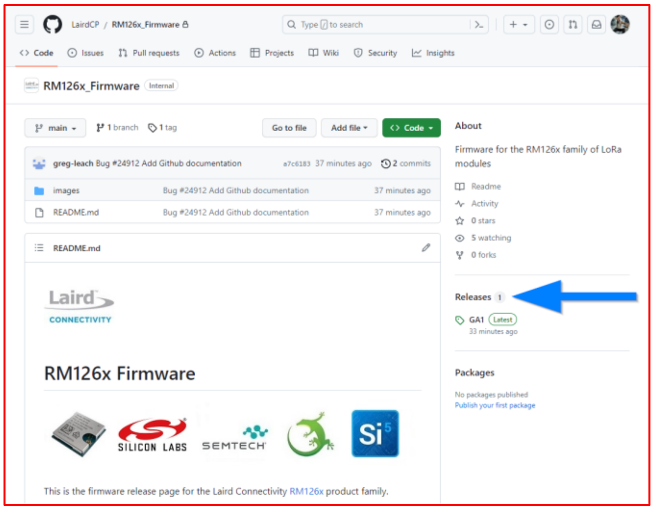
- (453-00140R/C) RM1261 module with external antenna (868MHz frequency band)
- (453-00139R/C) RM1262 module with external antenna (915MHz frequency band)

Software Tools and Utilities

- uart_dfu utility software, available for Windows → https://github.com/LairdCP/RM126x_Firmware
- (Optional) Simplicity Studio 5 for Windows, Linux or Mac → https://www.silabs.com/developers/simplicity-studio.
- J-Link Software and Documentation Pack v7.88n or later → https://www.segger.com/downloads/J-Link/.
- iOS or Android Smartphone with Bluetooth Low Energy support running latest OS version is preferred.
- Simplicity Commander v1.15.3 or later. Available for Windows, Linux, and Mac.
- Silicon Labs EFR Connect Mobile App for iOS / Android v2.7.1 or later.

3 RM126x GA Release Versions

You can obtain the latest available RM126x firmware images at https://github.com/LairdCP/RM126x_Firmware. Select the latest available release as shown below or click on **Release** to see a list of all published firmware image releases. To download a specific firmware release, tap **> Assets** and download the individual firmware image files for the RM126x.

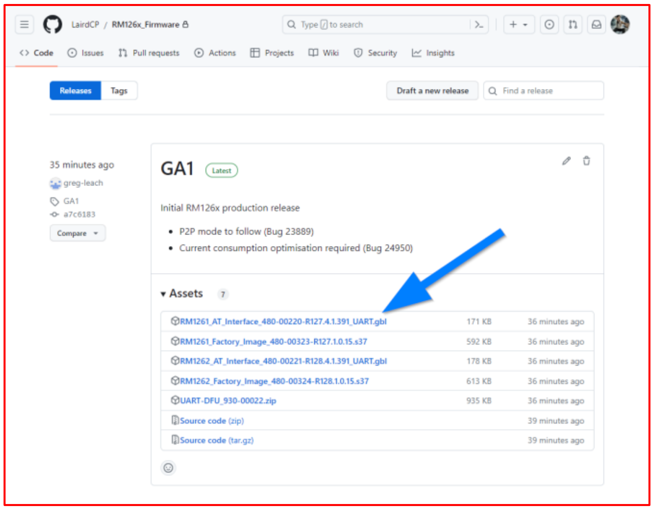


greg-leach Bug #24912 Add Github documentation	a7c6183 37 minutes ago	2 commits
images	Bug #24912 Add Github documentation	37 minutes ago
README.md	Bug #24912 Add Github documentation	37 minutes ago

Firmware for the RM126x family of LoRa modules

RM126x Firmware

This is the firmware release page for the Laird Connectivity RM126x product family.



Initial RM126x production release

- P2P mode to follow (Bug 23889)
- Current consumption optimisation required (Bug 24950)

RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl	171 KB	36 minutes ago
RM1261_Factory_Image_480-00323-R127.1.0.15.s37	592 KB	36 minutes ago
RM1262_AT_Interface_480-00221-R128.4.1.391_UART.gbl	178 KB	36 minutes ago
RM1262_Factory_Image_480-00324-R128.1.0.15.s37	613 KB	36 minutes ago
UART-DFU_930-00022.zip	935 KB	36 minutes ago
Source code (zip)		39 minutes ago
Source code (tar.gz)		39 minutes ago

Note: Older firmware images are unsupported by Ezurio and may contain issues or bugs that have been fixed in newer versions. Ezurio strongly recommends always using the latest firmware images. Use of these unsupported firmware images is at your own risk.

4 Bootloader and Firmware Options

4.1 Overview

The bootloader is the first piece of code which gets executed on modules reset and is required for operation of all other applications. It should be the first software part programmed to the target device.

All RM126x modules and DVKs ship with the bootloader and AT Interface application preloaded. While it's possible to use the modules/DVK as is, at some point it might be beneficial to upgrade the module firmware to the latest release provided by Ezurio since those might include bug fixes as well as additional functionality.

It is also possible to create a custom RM126x bootloader project using the Simplicity Studio 5 IDE and the "Internal Storage Bootloader (single image on 512kB device)" example. This e.g. allows you to modify the BOOT pin definition or to remove the GPIO activation for the BOOT pin if needed.

Table 1: BOOT pin for the RM126x Series

Bootloader Type and Config	RM126x BOOT Pin	Pin #	Comments
Ezurio	PC06/BOOT	19	Input with active low logic. Operating mode selected by BOOT pin status (at power up): If externally held Low (0V), enters BOOT mode; If High (VDD), enter application (default).

4.2 Simplicity Studio Generated Firmware Files

Please find below an overview about all firmware images which can be generated via Simplicity Studio.

For the SWD Firmware Upgrade Method Only

[filename].s37 → This is an application image in Motorola s-record format. Can be used with the SWD firmware upgrade method only.

Other formats such as [filename].bin and [filename].hex are currently not covered in this RM126x Firmware Options and Upgrading Methods User Guide document.

For the Serial DFU (UART) Firmware Upgrade Method Only

[filename]_UART.gbl → Combined application and Silicon Labs Apploader image, can be used with the Serial DFU (UART) firmware upgrade method only.

4.3 Ezurio-Provided Firmware Files

Please find below an overview about all firmware images which are provided by Ezurio files originating from Ezurio are suffixed with a code of the format **V-W.X.Y.Z**, with the parts having the following meaning.

- V: This is the Ezurio part number for the firmware image
- W: This is the module variant identifying code for the RM126x hardware,
- X: This is the major revision number of the Silicon Labs SDK in use
- Y: This is the major revision number of the firmware image
- Z: This is the minor revision number of the firmware image

Module Variants (W) for the RM126x:

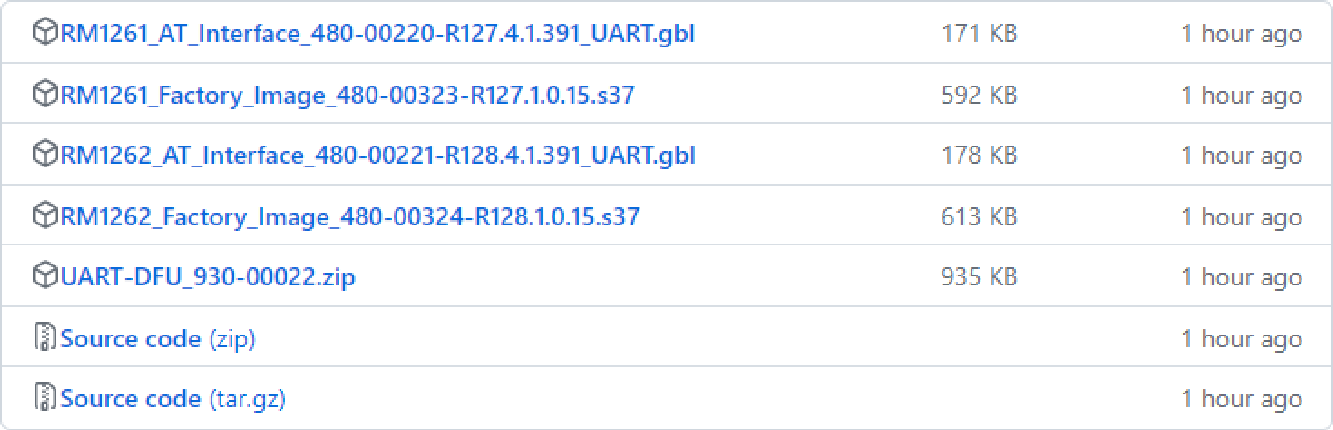
- RM1261: +15dBm TX power (EU863-868, IN865-867, AS923 / JP&TW)
- RM1262: +22dBm TX power (US902-928, AU915-928, AS923 / AU&NZ)

Table 2: Ezurio Code Format for RM126x

Example	RM1261	RM1262
V . . . ≙ . . .	480-00220	480-00221
W . . . ≙ . . .	R127	R128
X . . . ≙ . . .	4	4
Y . . . ≙ . . .	1	1
Z . . . ≙ . . .	391	391

Please note these details are omitted from the table below for clarity.

Image name	Equivalent to	Upgrade Method		Details
		SWD	SERIAL	
RM1261_AT_Interface…gbl	[filename]_UART.gbl	N	Y	RM1261 AT Interface
RM1261_Factory_Image…s36	[filename].s37	Y	N	
RM1262_AT_Interface…gbl	[filename]_UART.gbl	N	Y	RM1262 AT Interface
RM1262_Factory_Image…s36	[filename].s37	Y	N	



RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl	171 KB	1 hour ago
RM1261_Factory_Image_480-00323-R127.1.0.15.s37	592 KB	1 hour ago
RM1262_AT_Interface_480-00221-R128.4.1.391_UART.gbl	178 KB	1 hour ago
RM1262_Factory_Image_480-00324-R128.1.0.15.s37	613 KB	1 hour ago
UART-DFU_930-00022.zip	935 KB	1 hour ago
Source code (zip)		1 hour ago
Source code (tar.gz)		1 hour ago

Figure 1: GitHub Snapshot of all RM126x firmware files.

The UART-DFU_930-00022.zip includes the Windows executable for bootloader firmware upgrade over the UART.

5 Hardware Preparation

To prepare the hardware, do the following:

1. Download and extract the latest available RM126x GA release locally for further use and programming purposes.
2. Connect your RM126x DVK to your PC via the included type A to micro type B USB cable, and make sure that the blue “J-Link” LED is active.
3. Do not forget to mount and attach the included external antenna to your RM126x development board

Note: The RM126x development board will show up and identify as a **JLink CDC UART Port (SEGGER)** device in the Windows device manager under “Ports (COM & LPT)”. If the driver is not installed automatically when the DVK is plugged into your Windows PC, then please manually install Simplicity Studio 5 or the latest version of the J-Link Software and Documentation Pack.

6 Firmware Upgrade Methods

6.1 Serial DFU (UART) Using uart_dfu Application

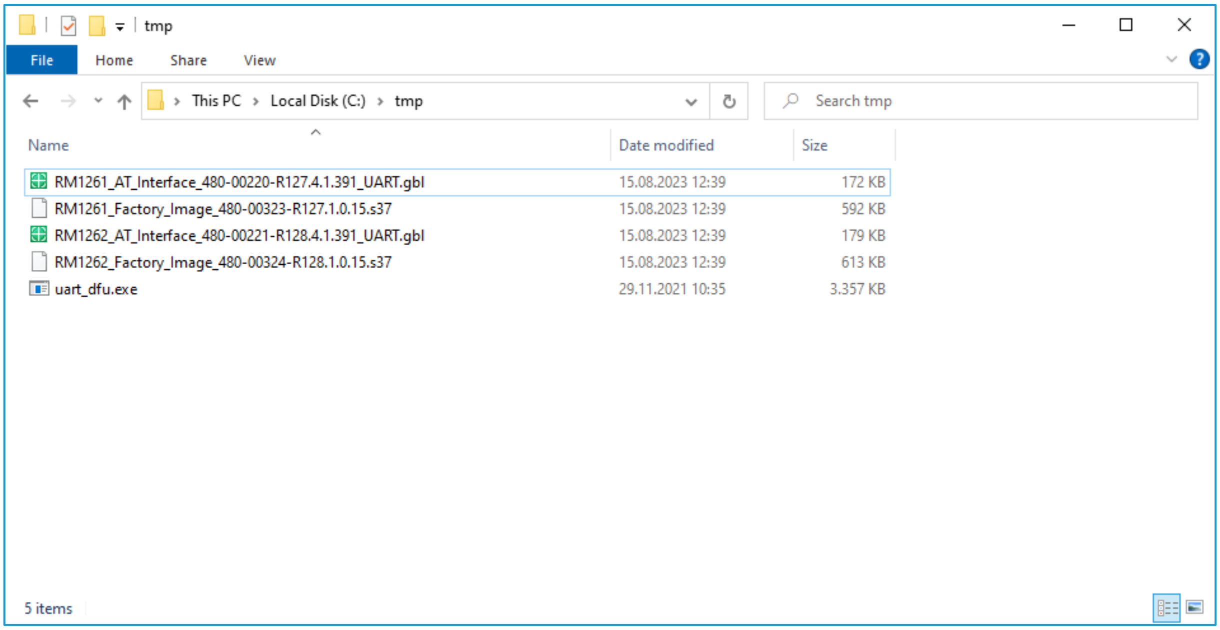
The Serial DFU (UART) firmware upgrade method allows the firmware to be updated via the serial interface using a pre-compiled application called **uart_dfu.** This is a common and powerful way to update firmware for our RM126x modules.

For this firmware upgrade method, it is mandatory to trigger the **BOOT** pin. The BOOT pin is an active-low digital input to force the module into DFU bootloader state upon device reset. It is used to determine when execution of the bootloader is required. Upon reset, execution of the bootloader begins. The state of the BOOT pin is read immediately upon start-up of the bootloader. If **LOW**, execution of the bootloader continues, facilitating firmware update via the UART. If the BOOT pin is **HIGH**, the bootloader will stop execution and pass control to the main application firmware.

1. Download and extract the **Serial DFU (UART) Utility** (uart_dfu) provided by the Ezurio team from the RM126x firmware releases GitHub page and copy the latest UART firmware image file(s) in the folder of the **uart_dfu.exe**. Make sure that the file names end with the “.gbl” suffix and format as shown below.

   Depending on your Simplicity Studio 5 installation / configuration the source code and Makefile for the original version can be found in the Gecko SDK (GSDK). Please make sure that the latest version for the GSDK is installed in this case. In Windows, the application can be built using, for example, MinGW or Cygwin. Under Linux or Mac the program can be compiled with the GCC toolchain. Please refer to this FAQ for build instructions under Windows and/or Linux.

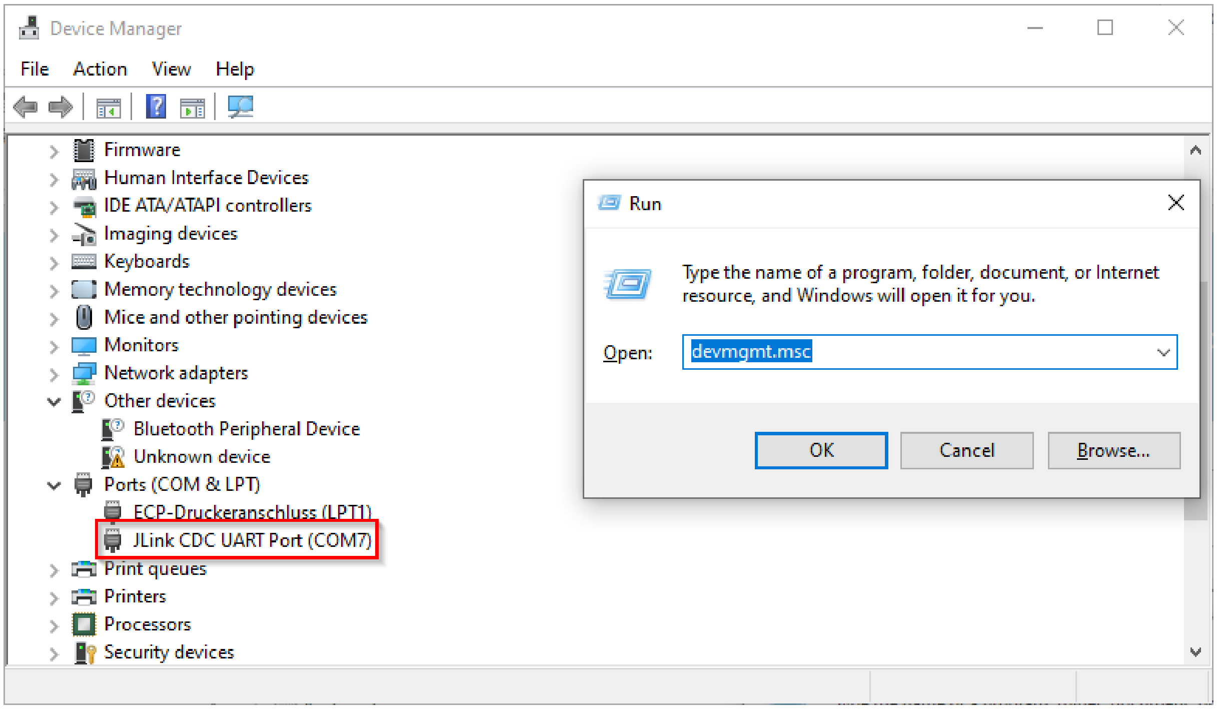
   - C:\SiliconLabs\SimplicityStudio\v5\developer\sdks\gecko_sdk_suite\<VERSION>\app\bluetooth\example_host\uart_dfu
   - C:\Users\<USERNAME>\SimplicityStudio\SDKs\gecko_sdk\app\bluetooth\example_host\bt_host_uart_dfu.



Name	Date modified	Size
RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl	15.08.2023 12:39	172 KB
RM1261_Factory_Image_480-00323-R127.1.0.15.s37	15.08.2023 12:39	592 KB
RM1262_AT_Interface_480-00221-R128.4.1.391_UART.gbl	15.08.2023 12:39	179 KB
RM1262_Factory_Image_480-00324-R128.1.0.15.s37	15.08.2023 12:39	613 KB
uart_dfu.exe	29.11.2021 10:35	3.357 KB

Figure 2: Serial DFU (UART) Utility and Firmware Image Files

2. Before continuing, we need to verify the **COM Port** for our RM126x Series – Development Kit via the **Windows Device Manager**. Press **Windows+R** which will open the **Run** window. Enter **devmgmt.msc** and then click **OK**. Find the COM Port for the **JLink CDC UART Port Device** in the **Ports (COM & LPT)** tab which is surrounded in brackets. In our example it is *COM5*.

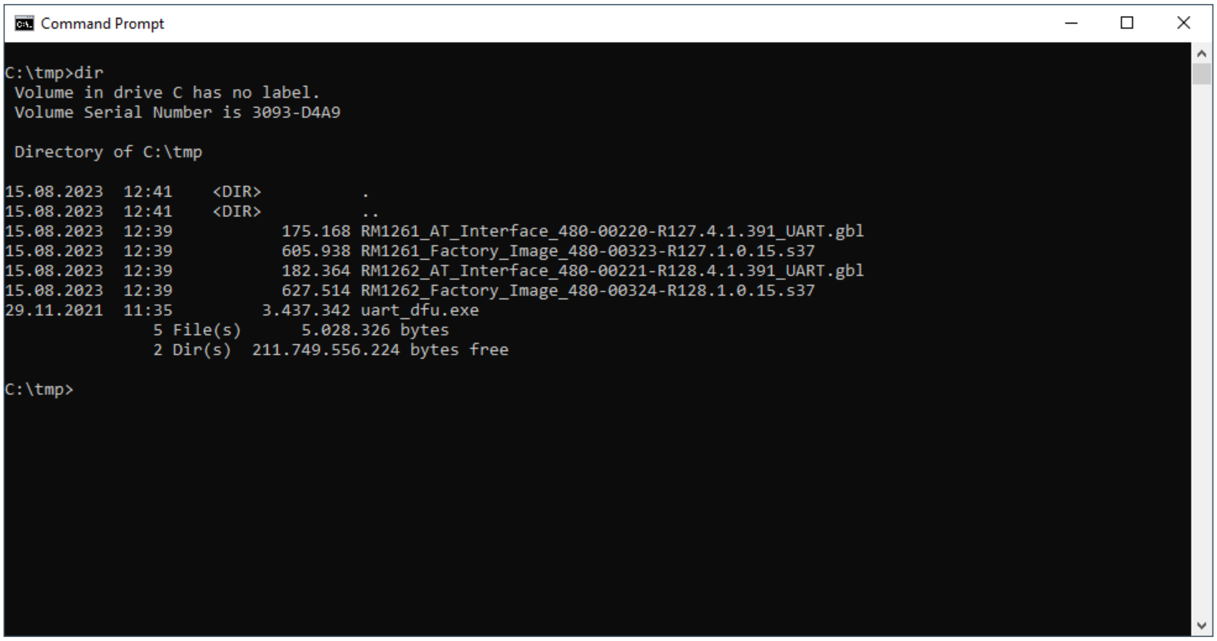


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: devmgmt.msc

Figure 3: Windows Device Manager and Run Window.

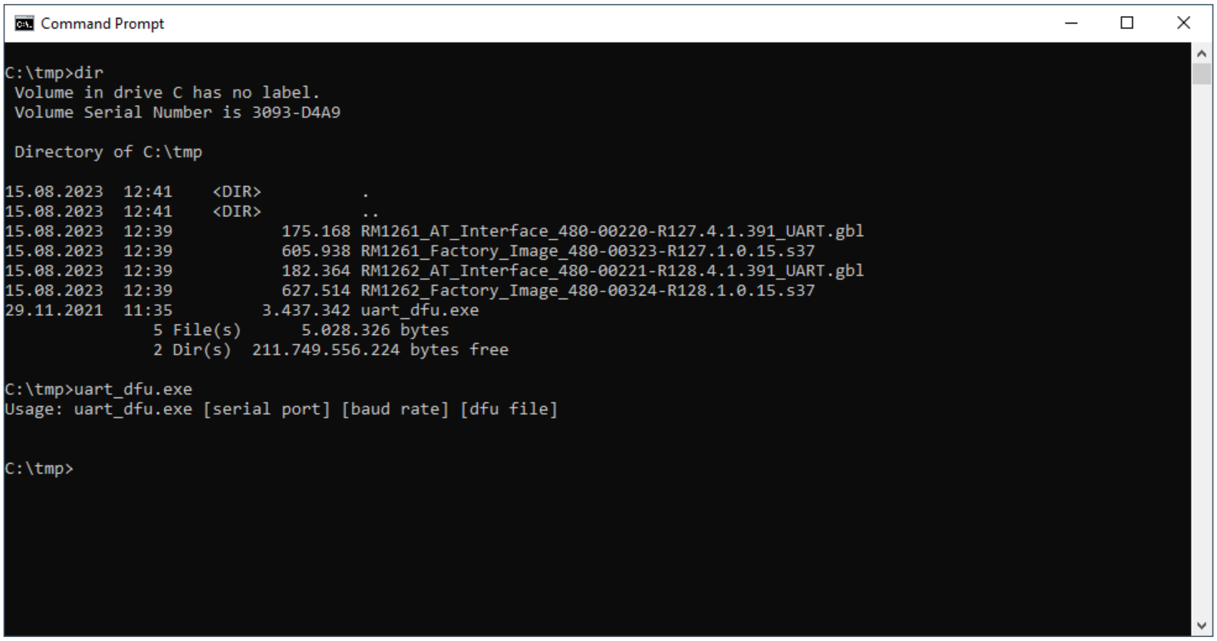
3. Open the **File Explorer** in Windows and navigate to the uart_dfu folder which contains all UART firmware image files as well as the uart_dfu.exe. Click in the address bar or alternatively press **Alt+D**. Once the address bar is highlighted type **cmd** in the bar and press **Enter**. This will automatically set up a cmd window for the uart_dfu folder with the correct paths.



```
C:\tmp>dir
 Volume in drive C has no label.
 Volume Serial Number is 3093-D4A9

 Directory of C:\tmp

15.08.2023  12:41    <DIR>          .
15.08.2023  12:41    <DIR>          ..
15.08.2023  12:39           175.168 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
15.08.2023  12:39           605.938 RM1261_Factory_Image_480-00323-R127.1.0.15.s37
15.08.2023  12:39           182.364 RM1262_AT_Interface_480-00221-R128.4.1.391_UART.gbl
15.08.2023  12:39           627.514 RM1262_Factory_Image_480-00324-R128.1.0.15.s37
29.11.2021  11:35         3.437.342 uart_dfu.exe
               5 File(s)      5.028.326 bytes
               2 Dir(s)  211.749.556.224 bytes free

C:\tmp>
```

Figure 4: Windows CMD Window

4. Enter **uart_dfu.exe** without any arguments to test if the application works on your system. This will print the required input parameters for the application as shown in Figure 5.



```
C:\tmp>dir
 Volume in drive C has no label.
 Volume Serial Number is 3093-D4A9

 Directory of C:\tmp

15.08.2023  12:41    <DIR>          .
15.08.2023  12:41    <DIR>          ..
15.08.2023  12:39           175.168 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
15.08.2023  12:39           605.938 RM1261_Factory_Image_480-00323-R127.1.0.15.s37
15.08.2023  12:39           182.364 RM1262_AT_Interface_480-00221-R128.4.1.391_UART.gbl
15.08.2023  12:39           627.514 RM1262_Factory_Image_480-00324-R128.1.0.15.s37
29.11.2021  11:35         3.437.342 uart_dfu.exe
               5 File(s)      5.028.326 bytes
               2 Dir(s)  211.749.556.224 bytes free

C:\tmp>uart_dfu.exe
Usage: uart_dfu.exe [serial port] [baud rate] [dfu file]


C:\tmp>
```

Figure 5: Windows CMD showing the uart.exe Usage Information.

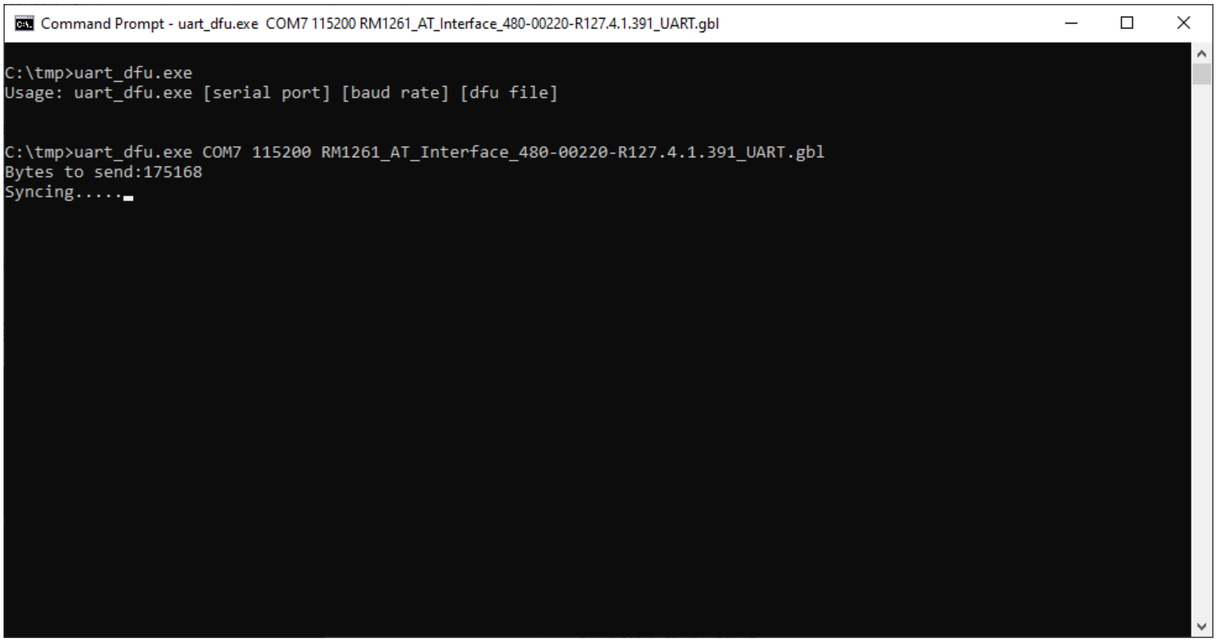
The input parameters are described as follows.

- `[serial_port]`: This is the COM port where the RM126x DVK being updated resides, of the form COMx, where x is the COM port number.
- `[baud rate]`: This is the baud rate of the COM port in decimal numeric format. Default value is 115200.
- `[dfu file]`: This is the path and filename where the update file resides. Note that if the update file resides in the same directory location as the uart_dfu.exe application, only the filename is required.

For an RM126x DVK residing on **COM7** with **115200** baud, and an upgrade filename **RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl**, with the Ezurio provided bootloader, the following command would be used to start the firmware upgrade via Serial DFU (UART), as shown in Figure 6.

```
uart_dfu.exe COM7 115200 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
```

Note in this case the upgrade file resides in the same location as the uart_dfu.exe application. If the upgrade file resides in a different location, the full path and file name should be provided within quotation characters.

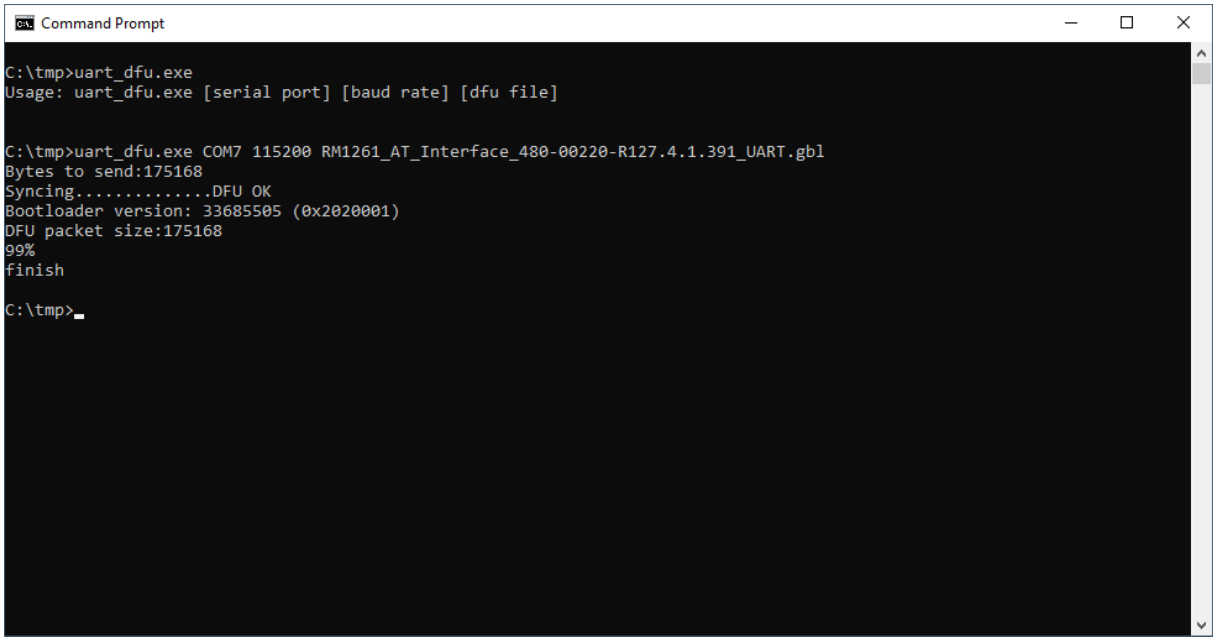


Command Prompt - uart_dfu.exe COM7 115200 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl

```
C:\tmp>uart_dfu.exe
Usage: uart_dfu.exe [serial port] [baud rate] [dfu file]


C:\tmp>uart_dfu.exe COM7 115200 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
Bytes to send:175168
Syncing.....
```

Figure 6: Windows CMD showing that the Firmware Upgrade via uart.exe was initiated and started.

5. Wait until you see the *Syncing…* message. Now press both **BTN0** and **RST** buttons on the RM126x Series – Development Kit simultaneously for at least 3 seconds. After this period, release the **RST** button only while continuing to hold **BTN0**. When you see a progress bar and a similar output as below, release **BTN0**.



```
C:\tmp>uart_dfu.exe
Usage: uart_dfu.exe [serial port] [baud rate] [dfu file]


C:\tmp>uart_dfu.exe COM7 115200 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
Bytes to send:175168
Syncing.............DFU OK
Bootloader version: 33685505 (0x2020001)
DFU packet size:175168
99%
finish

C:\tmp>_
```

Figure 7: Windows CMD showing that the Firmware Upgrade via uart.exe successfully completed.

6. Please make sure to reset or power cycle the RM126x Series – Development Kit (by pressing the **RST** button) once you see the *finish* message, meaning that the update completed successfully.

6.1.1 RM126x Example – uart_dfu.exe with User Commands and Output

User entered commands in this example are **highlighted**:

```
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

c:\tmp>uart_dfu.exe
Usage: uart_dfu.exe [serial port] [baud rate] [dfu file]


c:\tmp>uart_dfu.exe COM7 115200 RM1261_AT_Interface_480-00220-R127.4.1.391_UART.gbl
Bytes to send:175168
Syncing....DFU OK
Bootloader version: 33685505 (0x2020001)
DFU packet size:175168
99%
finish

c:\tmp>
```

6.2 Simplicity Studio 5 and Simplicity Commander (SWD)

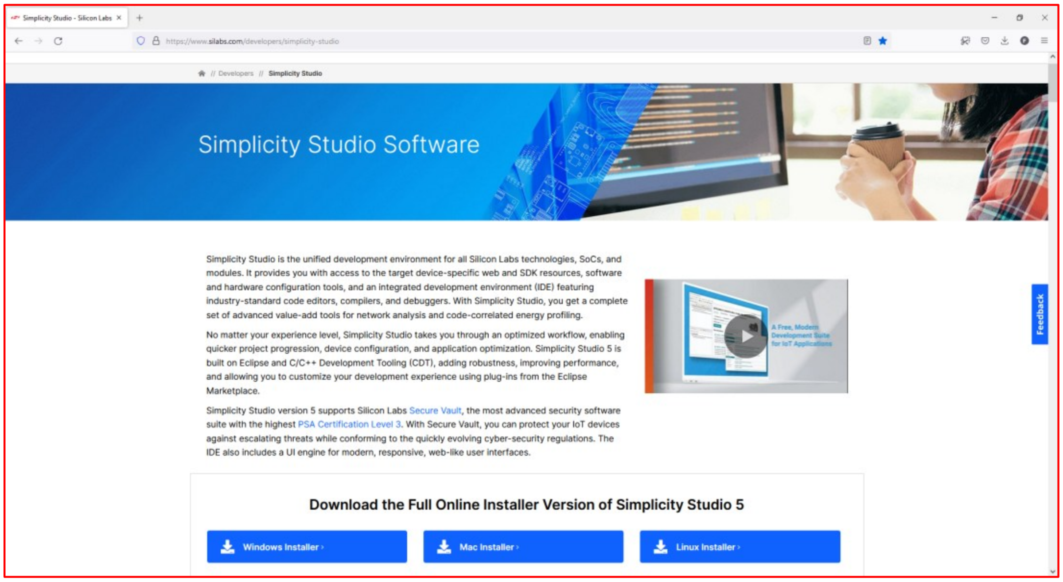
Simplicity Studio 5 is the recommended Integrated Development Environment (IDE) for our RM126x Series, developed and maintained by Silicon Labs. It simplifies IoT development by giving developers everything they need. Simplicity Commander is part of the Simplicity Studio 5 and a single, all-purpose tool that can be used in a production environment allowing to program our modules via a Graphical User Interface (GUI).

The primary audience for this firmware upgrade method are software developers who develop their own C code application using the Silabs SDK. You can find additional usage information for both Simplicity Studio 5 and Simplicity Commander in the latest Simplicity Studio 5 User's Guide and UG162: Simplicity Commander Reference Guide.

Note: SWCLK and SWDIO pins are mandatory for this firmware upgrade method. Please verify your existing hardware design against them and make sure that PA01 (SWCLK) / PA02 (SWDIO) pins are exposed and/or available for use. Please refer to our respective RM126x datasheets or DVK schematics for more information.

1. Before starting, please go to https://community.silabs.com/SL_CommunitiesSelfReg and register for a Silicon Labs account. Remember to confirm your account registration via email. This is important because it will grant you access to all Simplicity Studio 5 features as well as to the Simplicity Commander and Software Development Toolkit (SDK).

2. Next, download and install Simplicity Studio 5 which is available under https://www.silabs.com/developers/simplicity-studio for Windows, Linux, or Mac.


Simplicity Studio - Silicon Labs
https://www.silabs.com/developers/simplicity-studio
// Developers // Simplicity Studio
Simplicity Studio Software
A Free, Modern Development Suite for IoT Applications
Feedback

Simplicity Studio is the unified development environment for all Silicon Labs technologies, SoCs, and modules. It provides you with access to the target device-specific web and SDK resources, software and hardware configuration tools, and an integrated development environment (IDE) featuring industry-standard code editors, compilers, and debuggers. With Simplicity Studio, you get a complete set of advanced value-add tools for network analysis and code-correlated energy profiling.

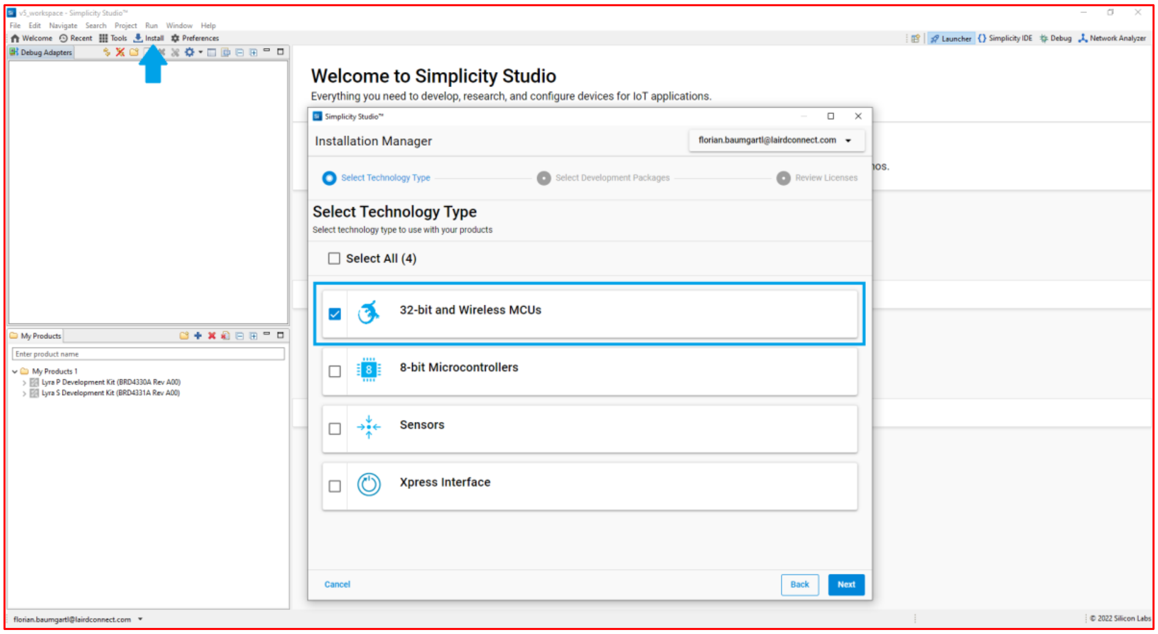
No matter your experience level, Simplicity Studio takes you through an optimized workflow, enabling quicker project progression, device configuration, and application optimization. Simplicity Studio 5 is built on Eclipse and C/C++ Development Tooling (CDT), adding robustness, improving performance, and allowing you to customize your development experience using plug-ins from the Eclipse Marketplace.

Simplicity Studio version 5 supports Silicon Labs Secure Vault, the most advanced security software suite with the highest PSA Certification Level 3. With Secure Vault, you can protect your IoT devices against escalating threats while conforming to the quickly evolving cyber-security regulations. The IDE also includes a UI engine for modern, responsive, web-like user interfaces.

Download the Full Online Installer Version of Simplicity Studio 5

Figure 8: Download Simplicity Studio 5 for Windows Linux, or Mac

3. After the Simplicity Studio 5 installation is finished, start Simplicity Studio 5 and enter your login credentials when prompted. Apply all available updates. Follow the on-screen prompts. This is highly recommended and may take a few minutes. Please also make sure that the *Gecko SDK – 32-bit and Wireless MCUs* technology / software component is installed. If needed, the latest available version can be downloaded through the Installation Manager in Simplicity Studio at any time as shown below.



Welcome to Simplicity Studio

Everything you need to develop, research, and configure devices for IoT applications.

Installation Manager

Select Technology Type

Select technology type to use with your products

☐ Select All (4)

☑ 32-bit and Wireless MCUs

☐ 8-bit Microcontrollers

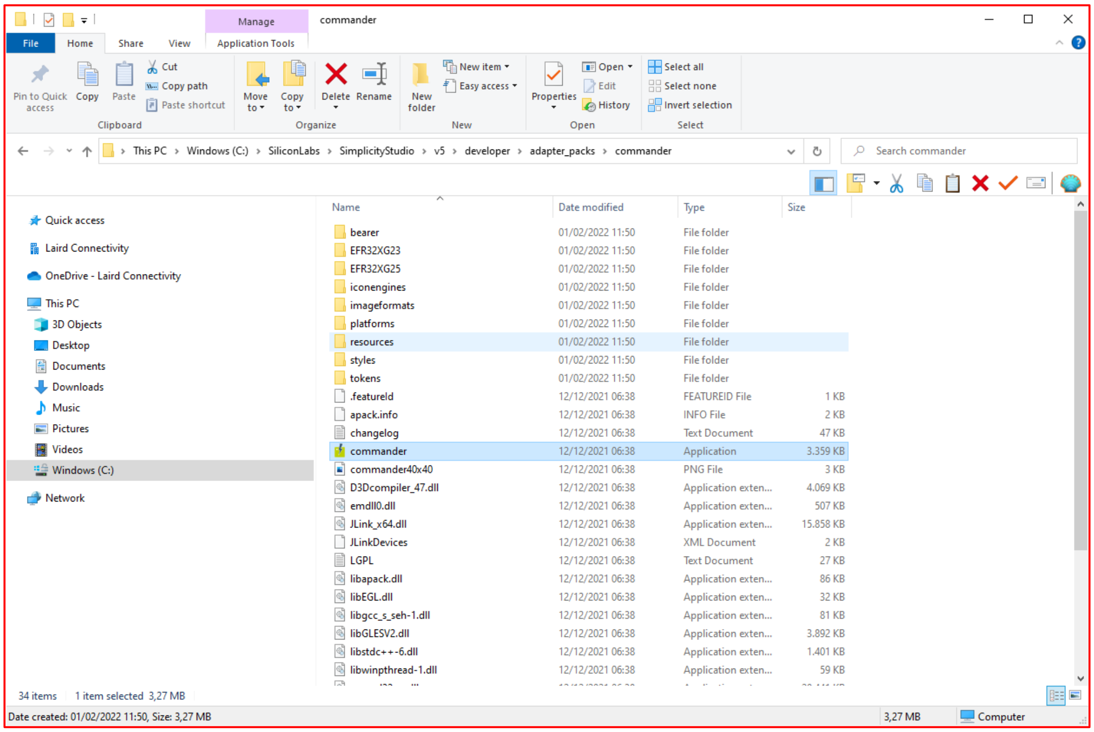
☐ Sensors

☐ Xpress Interface

v5_workspace - Simplicity Studio™
File Edit Navigate Search Project Run Window Help
Welcome Recent Tools Install Preferences
Launcher Simplicity IDE Debug Network Analyzer
Debug Adapters
My Products
Enter product name
My Products 1
Lyra P Development Kit (BRD4330A Rev A00)
Lyra S Development Kit (BRD4331A Rev A00)
Simplicity Studio™
florian.baumgartl@lairdconnect.com
Select Technology Type
Select Development Packages
Review Licenses
Cancel
Back
Next
florian.baumgartl@lairdconnect.com
© 2022 Silicon Labs

Figure 9: Simplicity Studio 5 Installation Manager

4. Now open your **File Explorer** in Windows and navigate to the Simplicity Commander folder. In our example it is *C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\commander*. Your installation path may be different.



This PC › Windows (C:) › SiliconLabs › SimplicityStudio › v5 › developer › adapter_packs › commander

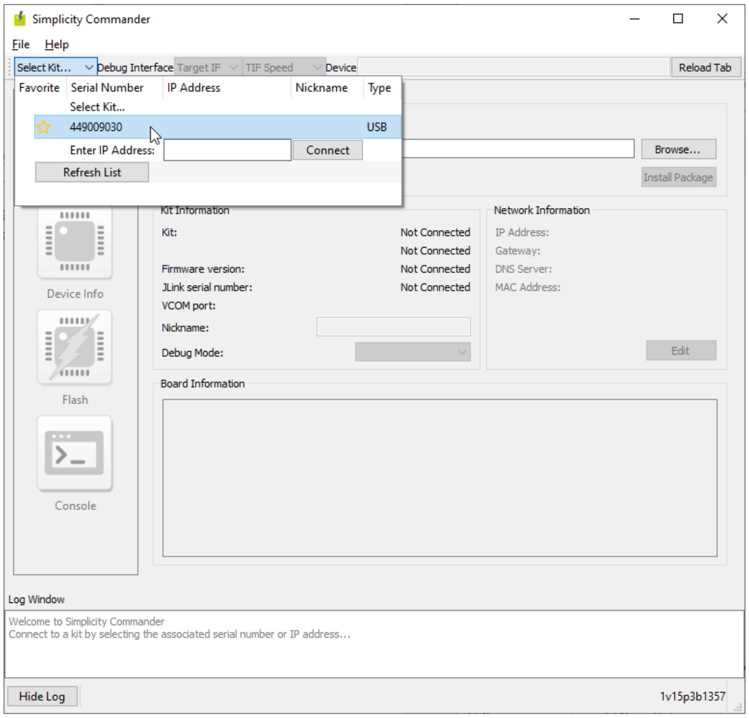
Name	Date modified	Type	Size
bearer	01/02/2022 11:50	File folder	
EFR32XG23	01/02/2022 11:50	File folder	
EFR32XG25	01/02/2022 11:50	File folder	
iconengines	01/02/2022 11:50	File folder	
imageformats	01/02/2022 11:50	File folder	
platforms	01/02/2022 11:50	File folder	
resources	01/02/2022 11:50	File folder	
styles	01/02/2022 11:50	File folder	
tokens	01/02/2022 11:50	File folder	
.featureId	12/12/2021 06:38	FEATUREID File	1 KB
apack.info	12/12/2021 06:38	INFO File	2 KB
changelog	12/12/2021 06:38	Text Document	47 KB
commander	12/12/2021 06:38	Application	3.359 KB
commander40x40	12/12/2021 06:38	PNG File	3 KB
D3Dcompiler_47.dll	12/12/2021 06:38	Application exten...	4.069 KB
emdll0.dll	12/12/2021 06:38	Application exten...	507 KB
JLink_x64.dll	12/12/2021 06:38	Application exten...	15.858 KB
JLinkDevices	12/12/2021 06:38	XML Document	2 KB
LGPL	12/12/2021 06:38	Text Document	27 KB
libapack.dll	12/12/2021 06:38	Application exten...	86 KB
libEGL.dll	12/12/2021 06:38	Application exten...	32 KB
libgcc_s_seh-1.dll	12/12/2021 06:38	Application exten...	81 KB
libGLESV2.dll	12/12/2021 06:38	Application exten...	3.892 KB
libstdc++-6.dll	12/12/2021 06:38	Application exten...	1.401 KB
libwinpthread-1.dll	12/12/2021 06:38	Application exten...	59 KB

34 items 1 item selected 3,27 MB

Date created: 01/02/2022 11:50, Size: 3,27 MB

Figure 10: File Explorer showing the installation folder for the Simplicity Commander

5. Find the **commander.exe** and run it. Go to **Select Kit...** and choose your RM126x Series – Development Kit. In our example it has the **Serial Number** *449000010*. Only one RM126x DVK should be connected to be correctly identified.



Favorite	Serial Number	IP Address	Nickname	Type
	Select Kit...			
☆	449009030			USB

Enter IP Address: Connect

Refresh List

Kit Information

Kit: Not Connected

Not Connected

Firmware version: Not Connected

JLink serial number: Not Connected

VCOM port:

Nickname:

Debug Mode:

Network Information

IP Address:

Gateway:

DNS Server:

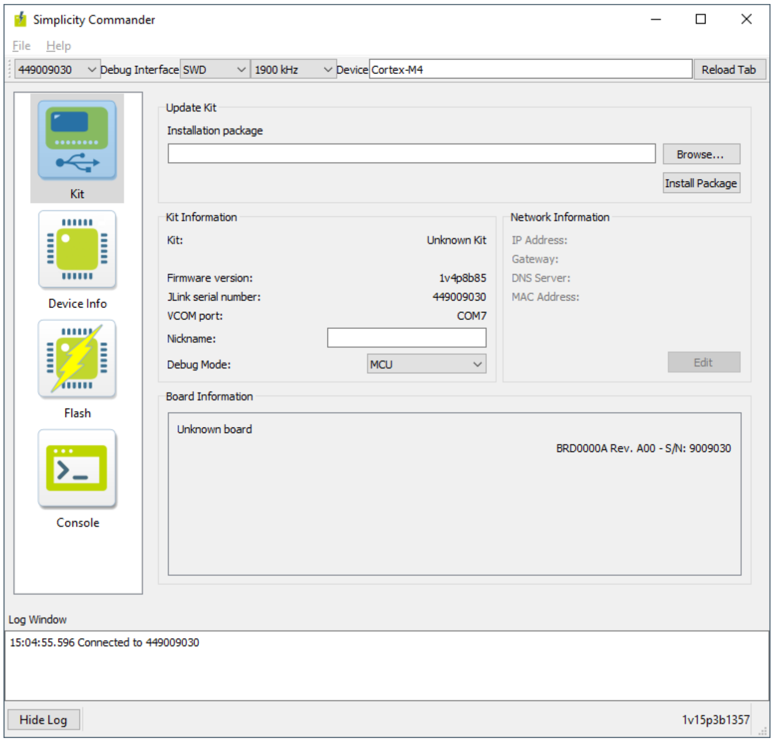
MAC Address:

Log Window

```
Welcome to Simplicity Commander
Connect to a kit by selecting the associated serial number or IP address...
```

Figure 11: Selecting RM126x DVK in Simplicity Commander

6. Once connected, you should see a *Connected to 449009030* message in the Log Window area. Also make sure that the **Debug Mode** is set to **MCU** as shown below.



Simplicity Commander

449009030 | Debug Interface: SWD | 1900 kHz | Device: Cortex-M4

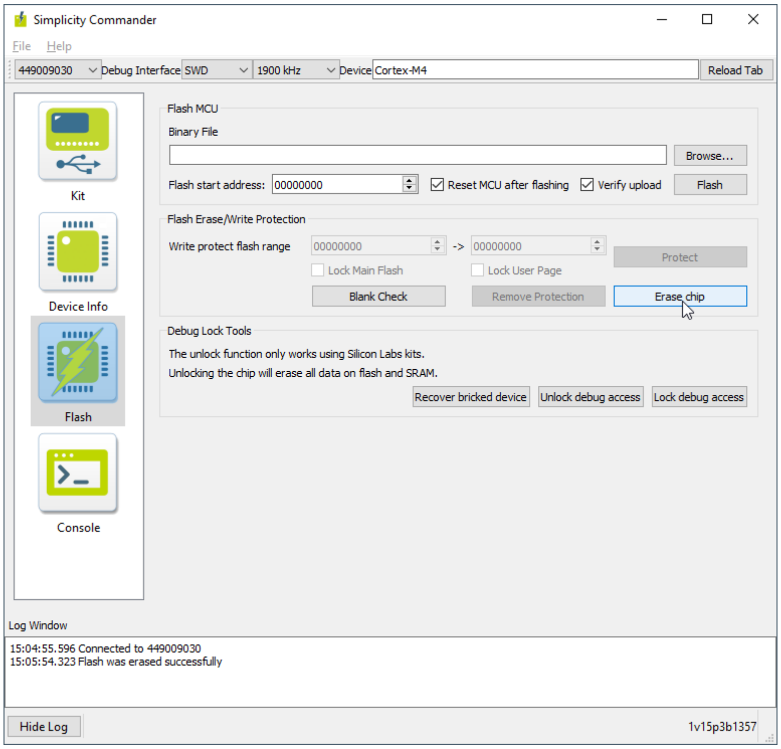
Kit Information
Kit: Unknown Kit
Firmware version: 1v4p8b85
JLink serial number: 449009030
VCOM port: COM7
Nickname:
Debug Mode: MCU

Board Information
Unknown board
BRD0000A Rev. A00 - S/N: 9009030

Log Window
```
15:04:55.596 Connected to 449009030
```
1v15p3b1357

Figure 12: RM126x DVK is connected in Simplicity Commander

7. Next, navigate to the **Flash** page and click the **Erase Chip** button. You should see *Flash was erased successfully message* in the Log Window area. Please leave all other settings unchanged.



Simplicity Commander

449009030 | Debug Interface: SWD | 1900 kHz | Device: Cortex-M4

Flash MCU
Binary File:
Flash start address: 00000000 ☑ Reset MCU after flashing ☑ Verify upload

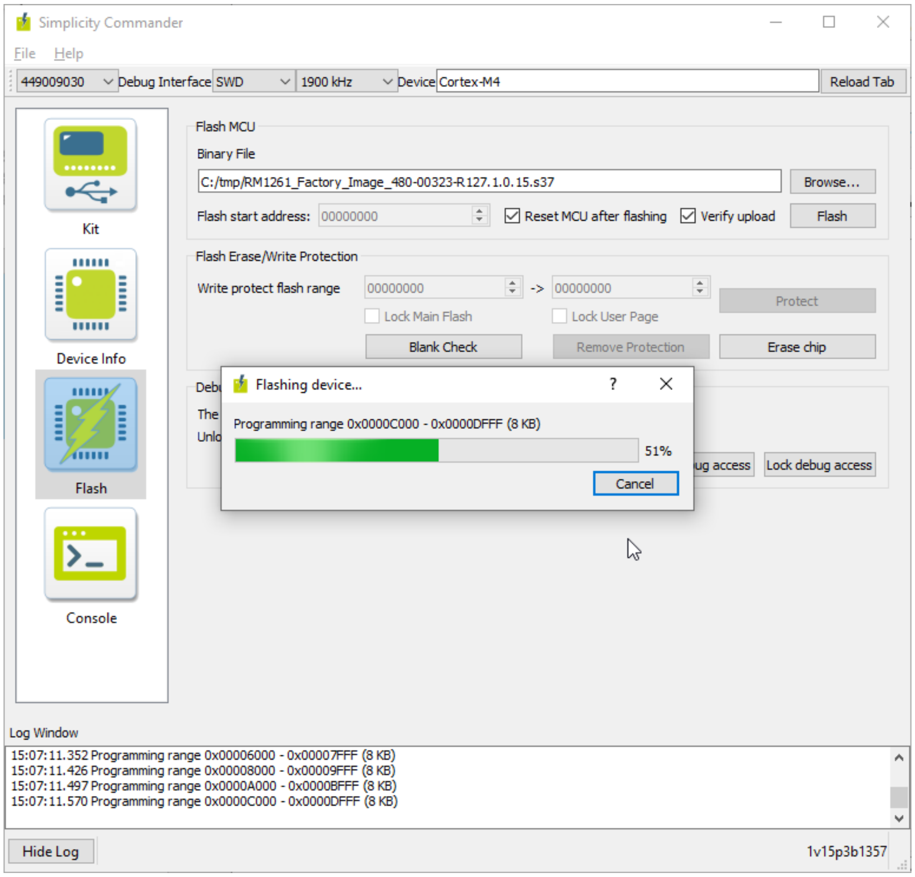
Flash Erase/Write Protection
Write protect flash range: 00000000 -> 00000000
□ Lock Main Flash □ Lock User Page

Debug Lock Tools
The unlock function only works using Silicon Labs kits.
Unlocking the chip will erase all data on flash and SRAM.

Log Window
```
15:04:55.596 Connected to 449009030
15:05:54.323 Flash was erased successfully
```
1v15p3b1357

Figure 13: Flash Erase in Simplicity Commander

8. Go to **Browse** and select the factory image. Make sure to change your factory image file path accordingly if needed. In our example it is *C:\tmp\RM1261_Factory_Image_480-00323-R127.1.0.15.s37*. Press the **Flash** Button. This usually does not take longer than 3 seconds.



449009030 Debug Interface SWD 1900 kHz Device Cortex-M4

Binary File: C:/tmp/RM1261_Factory_Image_480-00323-R127.1.0.15.s37

Flash start address: 00000000 ☑ Reset MCU after flashing ☑ Verify upload

Write protect flash range 00000000 -> 00000000

□ Lock Main Flash □ Lock User Page

Programming range 0x0000C000 - 0x0000DFFF (8 KB) 51%

```
15:07:11.352 Programming range 0x00006000 - 0x00007FFF (8 KB)
15:07:11.426 Programming range 0x00008000 - 0x00009FFF (8 KB)
15:07:11.497 Programming range 0x0000A000 - 0x0000BFFF (8 KB)
15:07:11.570 Programming range 0x0000C000 - 0x0000DFFF (8 KB)
```

Figure 14: Flashing factory image in Simplicity Commander ...

9. Please make sure to reset or power cycle the RM126x Series – Development Kit (by pressing the **RST** button) once programming of the factory image completed successfully.

6.2.1 RM126x Example – Simplicity Commander (Log Window) Output

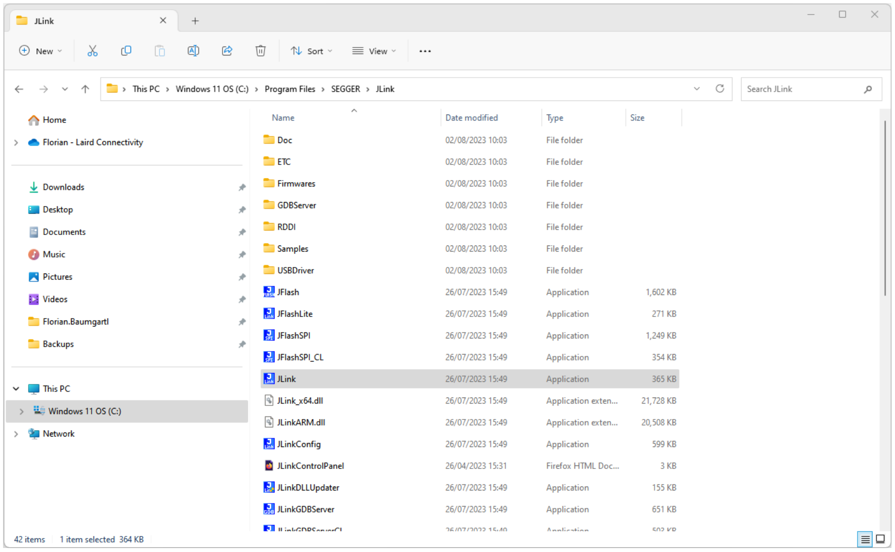
```
15:04:55.596 Connected to 449009030
15:05:54.323 Flash was erased successfully
15:07:10.813 Writing 204800 bytes starting at address 0x00000000
15:07:10.814 Writing 8192 bytes starting at address 0x0007e000
15:07:11.025 Comparing range 0x00000000 - 0x0001FFFF (128 KB)
15:07:11.043 Comparing range 0x00020000 - 0x00031FFF (72 KB)
15:07:11.057 Comparing range 0x0007E000 - 0x0007FFFF (8 KB)
15:07:11.063 Programming range 0x00000000 - 0x00001FFF (8 KB)
15:07:11.207 Programming range 0x00002000 - 0x00003FFF (8 KB)
15:07:11.279 Programming range 0x00004000 - 0x00005FFF (8 KB)
15:07:11.352 Programming range 0x00006000 - 0x00007FFF (8 KB)
15:07:11.426 Programming range 0x00008000 - 0x00009FFF (8 KB)
15:07:11.497 Programming range 0x0000A000 - 0x0000BFFF (8 KB)
15:07:11.570 Programming range 0x0000C000 - 0x0000DFFF (8 KB)
15:07:11.644 Programming range 0x0000E000 - 0x0000FFFF (8 KB)
15:07:11.716 Programming range 0x00010000 - 0x00011FFF (8 KB)
15:07:11.789 Programming range 0x00012000 - 0x00013FFF (8 KB)
15:07:11.862 Programming range 0x00014000 - 0x00015FFF (8 KB)
15:07:11.935 Programming range 0x00016000 - 0x00017FFF (8 KB)
15:07:12.007 Programming range 0x00018000 - 0x00019FFF (8 KB)
15:07:12.081 Programming range 0x0001A000 - 0x0001BFFF (8 KB)
15:07:12.154 Programming range 0x0001C000 - 0x0001DFFF (8 KB)
15:07:12.227 Programming range 0x0001E000 - 0x0001FFFF (8 KB)
15:07:12.298 Programming range 0x00020000 - 0x00021FFF (8 KB)
15:07:12.371 Programming range 0x00022000 - 0x00023FFF (8 KB)
15:07:12.444 Programming range 0x00024000 - 0x00025FFF (8 KB)
15:07:12.517 Programming range 0x00026000 - 0x00027FFF (8 KB)
15:07:12.590 Programming range 0x00028000 - 0x00029FFF (8 KB)
15:07:12.661 Programming range 0x0002A000 - 0x0002BFFF (8 KB)
15:07:12.734 Programming range 0x0002C000 - 0x0002DFFF (8 KB)
15:07:12.807 Programming range 0x0002E000 - 0x0002FFFF (8 KB)
15:07:12.880 Programming range 0x00030000 - 0x00031FFF (8 KB)
15:07:12.954 Programming range 0x0007E000 - 0x0007FFFF (8 KB)
15:07:13.175 Flashing completed successfully!
```

6.3 SEGGER J-Link Commander (SWD)

J-Link Commander is a tool provided by SEGGER that interfaces with devices attached to compatible J-Link debugger like our RM126x Series – Development Kits or any other compatible J-Link debuggers. It for example allows you to erase and flash our AT Interface application or full factory images in an easy way through a command-line interface (CLI). It perfectly fits and suits for all software engineers, hardware engineers, and release engineers who are familiar with programming system-on-chips (SoCs) and modules. This firmware upgrade method is also recommended for module recovery.

Note: SWCLK and SWDIO pins are mandatory for this firmware upgrade method. Please verify your existing hardware design against them and make sure that PA01 (SWCLK) / PA02 (SWDIO) pins are exposed and/or available for use. Please refer to our respective RM126x datasheets or DVK schematics for more information.

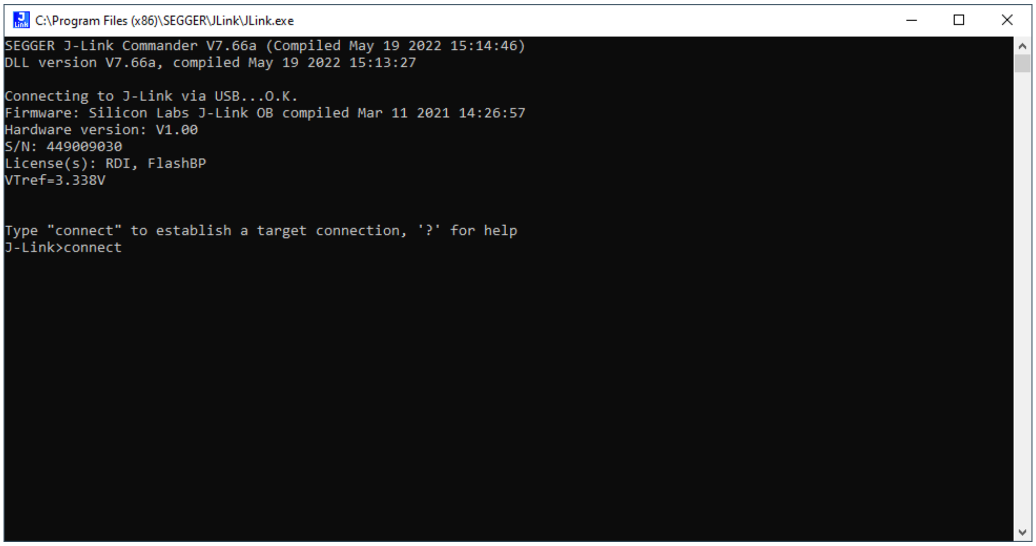
1. Please download and install the latest version of the **J-Link Software and Documentation Pack** which is available under https://www.segger.com/downloads/J-Link/. It's highly recommended not to work with any beta version(s). Further details, resources and instructions about this tool can be found here.
2. Navigate to the default **C:\Program Files\SEGGER\JLink** folder and start the **SEGGER J-Link Commander** by running the **JLink.exe** application. Depending on your installation, paths may differ here.



Name	Date modified	Type	Size
Doc	02/08/2023 10:03	File folder	
ETC	02/08/2023 10:03	File folder	
Firmwares	02/08/2023 10:03	File folder	
GDBServer	02/08/2023 10:03	File folder	
RDDI	02/08/2023 10:03	File folder	
Samples	02/08/2023 10:03	File folder	
USBDriver	02/08/2023 10:03	File folder	
JFlash	26/07/2023 15:49	Application	1,602 KB
JFlashLite	26/07/2023 15:49	Application	271 KB
JFlashSPI	26/07/2023 15:49	Application	1,249 KB
JFlashSPI_CL	26/07/2023 15:49	Application	354 KB
JLink	26/07/2023 15:49	Application	365 KB
JLink_x64.dll	26/07/2023 15:49	Application exten...	21,728 KB
JLinkARM.dll	26/07/2023 15:49	Application exten...	20,508 KB
JLinkConfig	26/07/2023 15:49	Application	599 KB
JLinkControlPanel	26/04/2023 15:31	Firefox HTML Doc...	3 KB
JLinkDLLUpdater	26/07/2023 15:49	Application	155 KB
JLinkGDBServer	26/07/2023 15:49	Application	651 KB

Figure 15: SEGGER J-Link Software and Documentation Pack after installation

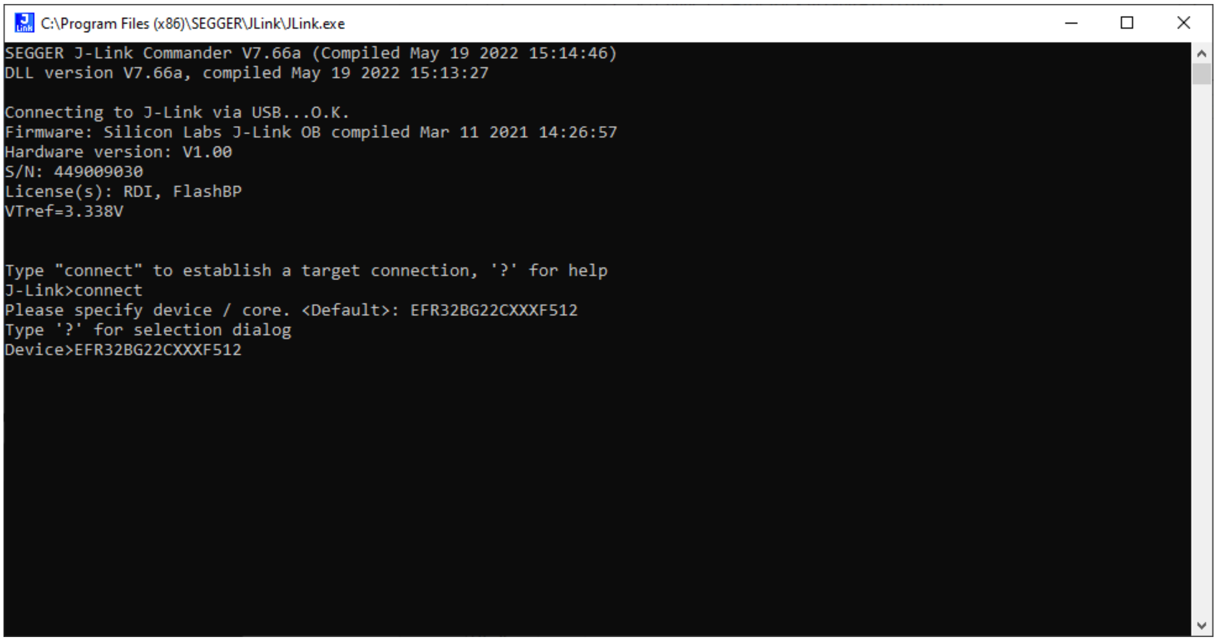
3. Start by connecting to our RM126x Series – Development Kit. This is very easy, simply enter **connect** in the *J-Link>* line.



```
SEGGER J-Link Commander V7.66a (Compiled May 19 2022 15:14:46)
DLL version V7.66a, compiled May 19 2022 15:13:27

Connecting to J-Link via USB...O.K.
Firmware: Silicon Labs J-Link OB compiled Mar 11 2021 14:26:57
Hardware version: V1.00
S/N: 449009030
License(s): RDI, FlashBP
VTref=3.338V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
```

Figure 16: J-Link Commander & RM126x DVK connection establishment.

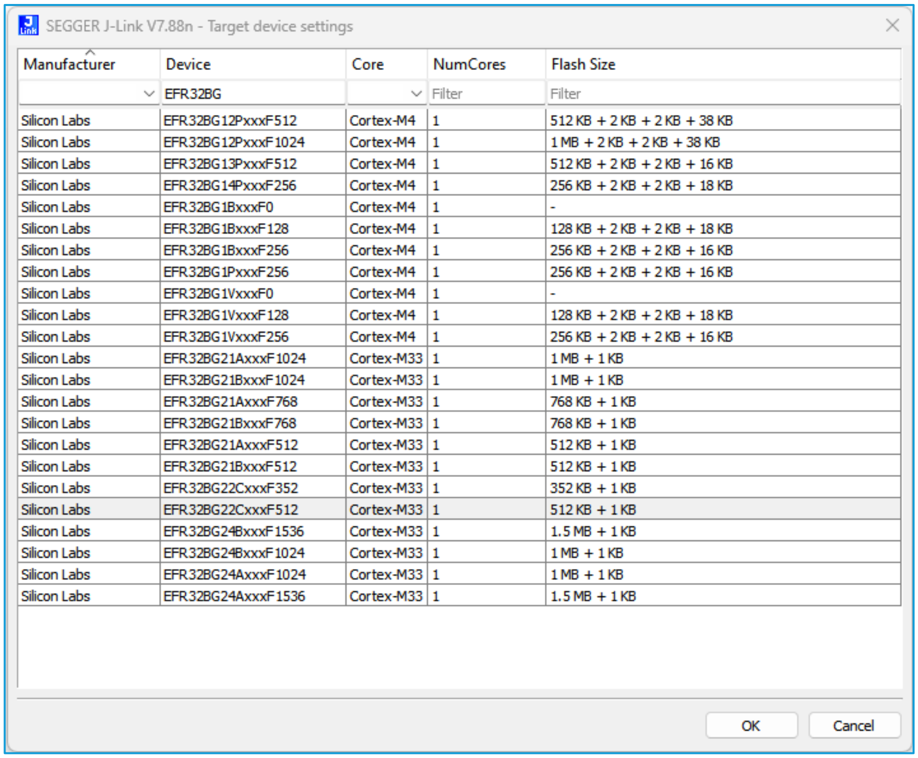
4. J-Link Commander then prompts for a **target device / core**. Only one RM126x development board should be connected to be correctly identified. Enter **EFR32BG22CXXXF512** in the *Device>* line or enter **?** which will open an additional specification dialog with a complete device list:



C:\Program Files (x86)\SEGGER\JLink\JLink.exe

```
SEGGER J-Link Commander V7.66a (Compiled May 19 2022 15:14:46)
DLL version V7.66a, compiled May 19 2022 15:13:27

Connecting to J-Link via USB...O.K.
Firmware: Silicon Labs J-Link OB compiled Mar 11 2021 14:26:57
Hardware version: V1.00
S/N: 449009030
License(s): RDI, FlashBP
VTref=3.338V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: EFR32BG22CXXXF512
Type '?' for selection dialog
Device>EFR32BG22CXXXF512
```

Figure 17: J-Link Commander is asking for target device / core.

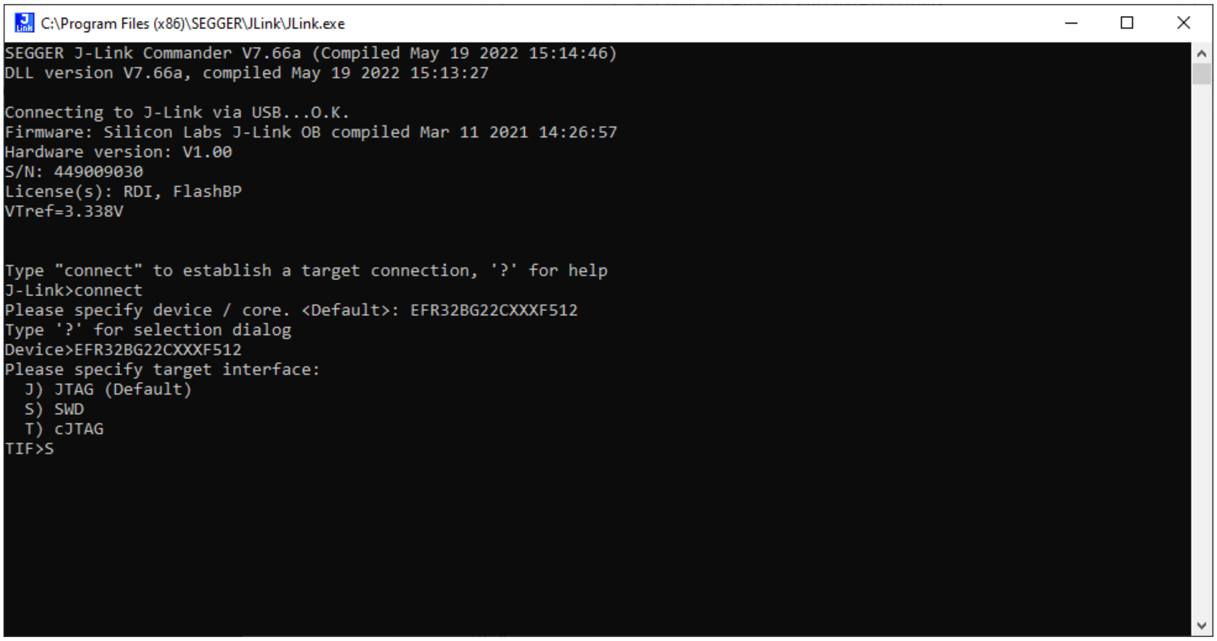


SEGGER J-Link V7.88n - Target device settings

Manufacturer	Device	Core	NumCores	Flash Size
	EFR32BG		Filter	Filter
Silicon Labs	EFR32BG12PxxxF512	Cortex-M4	1	512 KB + 2 KB + 2 KB + 38 KB
Silicon Labs	EFR32BG12PxxxF1024	Cortex-M4	1	1 MB + 2 KB + 2 KB + 38 KB
Silicon Labs	EFR32BG13PxxxF512	Cortex-M4	1	512 KB + 2 KB + 2 KB + 16 KB
Silicon Labs	EFR32BG14PxxxF256	Cortex-M4	1	256 KB + 2 KB + 2 KB + 18 KB
Silicon Labs	EFR32BG1BxxxF0	Cortex-M4	1	-
Silicon Labs	EFR32BG1BxxxF128	Cortex-M4	1	128 KB + 2 KB + 2 KB + 18 KB
Silicon Labs	EFR32BG1BxxxF256	Cortex-M4	1	256 KB + 2 KB + 2 KB + 16 KB
Silicon Labs	EFR32BG1PxxxF256	Cortex-M4	1	256 KB + 2 KB + 2 KB + 16 KB
Silicon Labs	EFR32BG1VxxxF0	Cortex-M4	1	-
Silicon Labs	EFR32BG1VxxxF128	Cortex-M4	1	128 KB + 2 KB + 2 KB + 18 KB
Silicon Labs	EFR32BG1VxxxF256	Cortex-M4	1	256 KB + 2 KB + 2 KB + 16 KB
Silicon Labs	EFR32BG21AxxxF1024	Cortex-M33	1	1 MB + 1 KB
Silicon Labs	EFR32BG21BxxxF1024	Cortex-M33	1	1 MB + 1 KB
Silicon Labs	EFR32BG21AxxxF768	Cortex-M33	1	768 KB + 1 KB
Silicon Labs	EFR32BG21BxxxF768	Cortex-M33	1	768 KB + 1 KB
Silicon Labs	EFR32BG21AxxxF512	Cortex-M33	1	512 KB + 1 KB
Silicon Labs	EFR32BG21BxxxF512	Cortex-M33	1	512 KB + 1 KB
Silicon Labs	EFR32BG22CxxxF352	Cortex-M33	1	352 KB + 1 KB
Silicon Labs	EFR32BG22CxxxF512	Cortex-M33	1	512 KB + 1 KB
Silicon Labs	EFR32BG24BxxxF1536	Cortex-M33	1	1.5 MB + 1 KB
Silicon Labs	EFR32BG24BxxxF1024	Cortex-M33	1	1 MB + 1 KB
Silicon Labs	EFR32BG24AxxxF1024	Cortex-M33	1	1 MB + 1 KB
Silicon Labs	EFR32BG24AxxxF1536	Cortex-M33	1	1.5 MB + 1 KB

Figure 18: J-Link Commander full device list window when entering ? command.

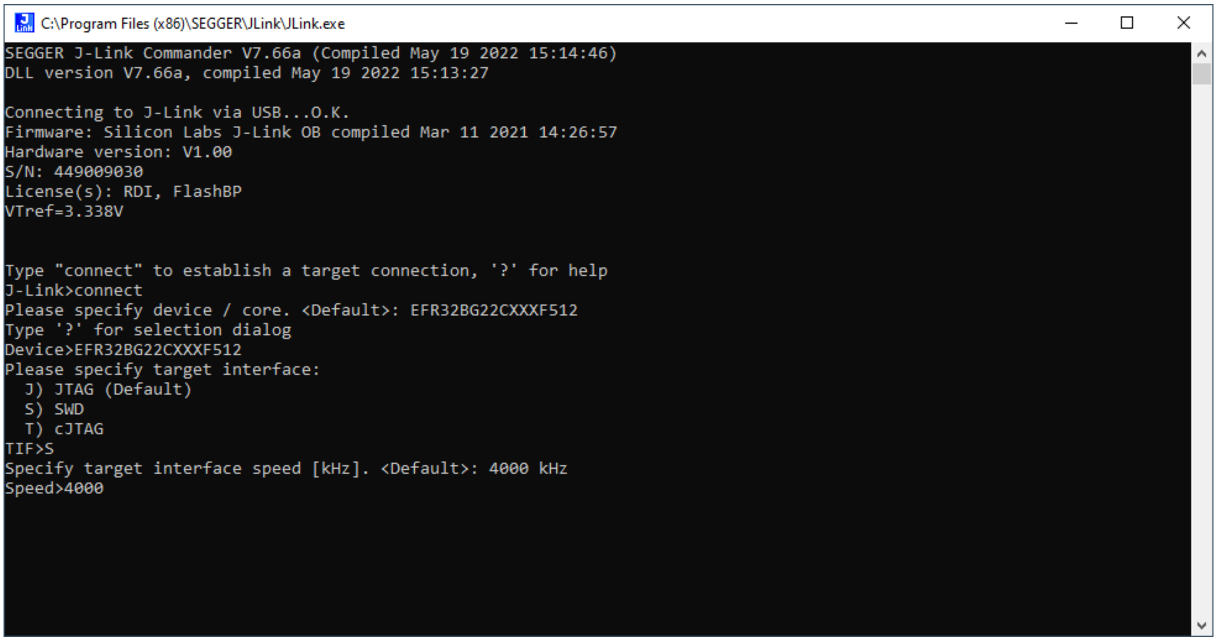
5. Next, J-Link Commander will ask for the correct **target interface**. Enter **S** in the *TIF>* line.



```
SEGGER J-Link Commander V7.66a (Compiled May 19 2022 15:14:46)
DLL version V7.66a, compiled May 19 2022 15:13:27

Connecting to J-Link via USB...O.K.
Firmware: Silicon Labs J-Link OB compiled Mar 11 2021 14:26:57
Hardware version: V1.00
S/N: 449009030
License(s): RDI, FlashBP
VTref=3.338V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: EFR32BG22CXXXF512
Type '?' for selection dialog
Device>EFR32BG22CXXXF512
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>S
```

Figure 19: SEGGER J-Link Commander asking for target interface.

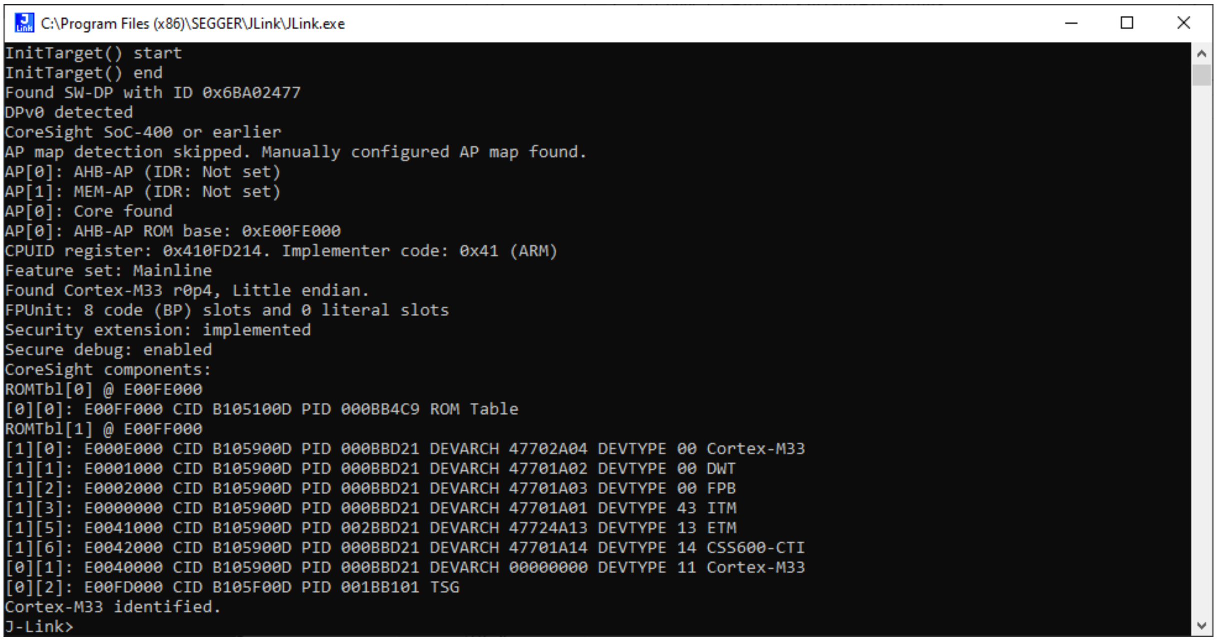
6. J-Link Commander also needs the correct **target interface speed** to proceed. The default value of *4000 kHz* can be selected here. Simply hit **Enter** or manual enter 4000 in the *Speed>* line to confirm the selection.



```
SEGGER J-Link Commander V7.66a (Compiled May 19 2022 15:14:46)
DLL version V7.66a, compiled May 19 2022 15:13:27

Connecting to J-Link via USB...O.K.
Firmware: Silicon Labs J-Link OB compiled Mar 11 2021 14:26:57
Hardware version: V1.00
S/N: 449009030
License(s): RDI, FlashBP
VTref=3.338V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: EFR32BG22CXXXF512
Type '?' for selection dialog
Device>EFR32BG22CXXXF512
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>4000
```

Figure 20: SEGGER J-Link Commander asking for target interface speed.

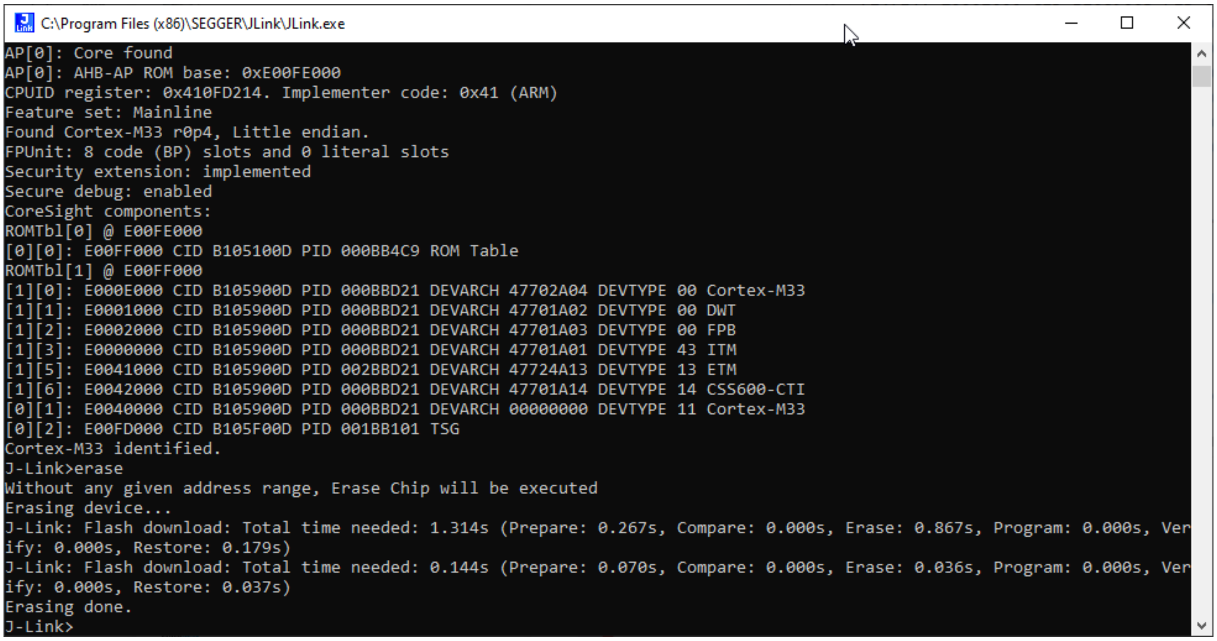
7. Check for a *Cortex-M33 identified.* message indicating a successful SWD connection to the RM126x DVK:



```
InitTarget() start
InitTarget() end
Found SW-DP with ID 0x6BA02477
DPv0 detected
CoreSight SoC-400 or earlier
AP map detection skipped. Manually configured AP map found.
AP[0]: AHB-AP (IDR: Not set)
AP[1]: MEM-AP (IDR: Not set)
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FE000
CPUID register: 0x410FD214. Implementer code: 0x41 (ARM)
Feature set: Mainline
Found Cortex-M33 r0p4, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
Security extension: implemented
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FE000
[0][0]: E00FF000 CID B105100D PID 000BB4C9 ROM Table
ROMTbl[1] @ E00FF000
[1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[1][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[1][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[1][5]: E0041000 CID B105900D PID 002BBD21 DEVARCH 47724A13 DEVTYPE 13 ETM
[1][6]: E0042000 CID B105900D PID 000BBD21 DEVARCH 47701A14 DEVTYPE 14 CSS600-CTI
[0][1]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33
[0][2]: E00FD000 CID B105F00D PID 001BB101 TSG
Cortex-M33 identified.
J-Link>
```

Figure 21: SEGGER J-Link Commander successfully connected to the RM126x DVK.

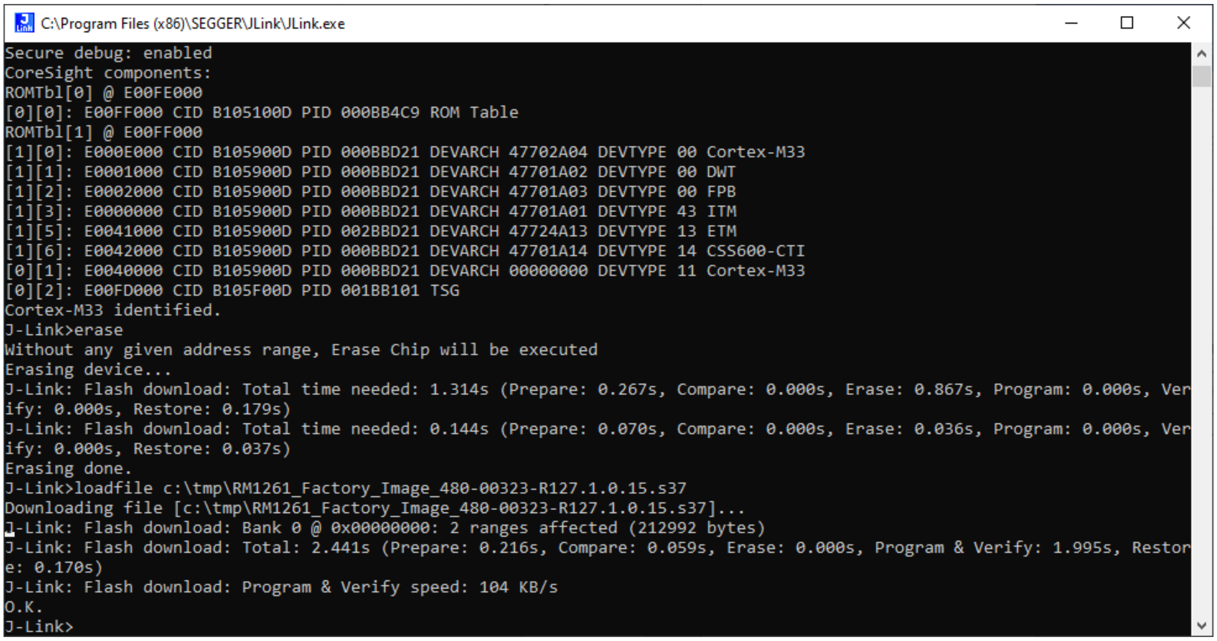
8. It's usually highly recommended to first erase all flash sectors with the **erase** command:



```
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FE000
CPUID register: 0x410FD214. Implementer code: 0x41 (ARM)
Feature set: Mainline
Found Cortex-M33 r0p4, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
Security extension: implemented
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FE000
[0][0]: E00FF000 CID B105100D PID 000BB4C9 ROM Table
ROMTbl[1] @ E00FF000
[1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[1][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[1][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[1][5]: E0041000 CID B105900D PID 002BBD21 DEVARCH 47724A13 DEVTYPE 13 ETM
[1][6]: E0042000 CID B105900D PID 000BBD21 DEVARCH 47701A14 DEVTYPE 14 CSS600-CTI
[0][1]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33
[0][2]: E00FD000 CID B105F00D PID 001BB101 TSG
Cortex-M33 identified.
J-Link>erase
Without any given address range, Erase Chip will be executed
Erasing device...
J-Link: Flash download: Total time needed: 1.314s (Prepare: 0.267s, Compare: 0.000s, Erase: 0.867s, Program: 0.000s, Ver
ify: 0.000s, Restore: 0.179s)
J-Link: Flash download: Total time needed: 0.144s (Prepare: 0.070s, Compare: 0.000s, Erase: 0.036s, Program: 0.000s, Ver
ify: 0.000s, Restore: 0.037s)
Erasing done.
J-Link>
```

Figure 22: Erasing flash memory via SEGGER J-Link Commander ...

9. Start programming the factory image by using the **loadfile ‹filepath›** command. Please make sure to change the *‹filepath›* accordingly. In our example we use the **loadfile C:\tmp\ RM1261_Factory_Image_480-00323-R127.1.0.15.s37** command.



```
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FE000
[0][0]: E00FF000 CID B105100D PID 000BB4C9 ROM Table
ROMTbl[1] @ E00FF000
[1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[1][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[1][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[1][5]: E0041000 CID B105900D PID 002BBD21 DEVARCH 47724A13 DEVTYPE 13 ETM
[1][6]: E0042000 CID B105900D PID 000BBD21 DEVARCH 47701A14 DEVTYPE 14 CSS600-CTI
[0][1]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33
[0][2]: E00FD000 CID B105F00D PID 001BB101 TSG
Cortex-M33 identified.
J-Link>erase
Without any given address range, Erase Chip will be executed
Erasing device...
J-Link: Flash download: Total time needed: 1.314s (Prepare: 0.267s, Compare: 0.000s, Erase: 0.867s, Program: 0.000s, Ver
ify: 0.000s, Restore: 0.179s)
J-Link: Flash download: Total time needed: 0.144s (Prepare: 0.070s, Compare: 0.000s, Erase: 0.036s, Program: 0.000s, Ver
ify: 0.000s, Restore: 0.037s)
Erasing done.
J-Link>loadfile c:\tmp\RM1261_Factory_Image_480-00323-R127.1.0.15.s37
Downloading file [c:\tmp\RM1261_Factory_Image_480-00323-R127.1.0.15.s37]...
J-Link: Flash download: Bank 0 @ 0x00000000: 2 ranges affected (212992 bytes)
J-Link: Flash download: Total: 2.441s (Prepare: 0.216s, Compare: 0.059s, Erase: 0.000s, Program & Verify: 1.995s, Restor
e: 0.170s)
J-Link: Flash download: Program & Verify speed: 104 KB/s
O.K.
J-Link>
```

Figure 23: Programming the factory image via SEGGER J-Link Commander ...

10. Please make sure to reset or power cycle the RM126x Series – Development Kit (by pressing the **RST** button) once the programming command responded with a *O.K.* message, meaning that the programming completed successfully.

6.3.1 RM126x Example – J-Link Commander with User Commands and Output

User entered commands in this example are **highlighted**:

```
SEGGER J-Link Commander V7.66a (Compiled May 19 2022 15:14:46)
DLL version V7.66a, compiled May 19 2022 15:13:27

Connecting to J-Link via USB...O.K.
Firmware: Silicon Labs J-Link OB compiled Mar 11 2021 14:26:57
Hardware version: V1.00
S/N: 449009016
License(s): RDI, FlashBP
VTref=3.335V

Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: EFR32BG22CXXXF512
Type '?' for selection dialog
Device>
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>s
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>4000
Device "EFR32BG22CXXXF512" selected.

Connecting to target via SWD
InitTarget() start
InitTarget() end
Found SW-DP with ID 0x6BA02477
DPv0 detected
CoreSight SoC-400 or earlier
AP map detection skipped. Manually configured AP map found.
AP[0]: AHB-AP (IDR: Not set)
AP[1]: MEM-AP (IDR: Not set)
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FE000
CPUID register: 0x410FD214. Implementer code: 0x41 (ARM)
Feature set: Mainline
Found Cortex-M33 r0p4, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
Security extension: implemented
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FE000
[0][0]: E00FF000 CID B105100D PID 000BB4C9 ROM Table
ROMTbl[1] @ E00FF000
[1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[1][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[1][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[1][5]: E0041000 CID B105900D PID 002BBD21 DEVARCH 47724A13 DEVTYPE 13 ETM
[1][6]: E0042000 CID B105900D PID 000BBD21 DEVARCH 47701A14 DEVTYPE 14 CSS600-CTI
[0][1]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33
[0][2]: E00FD000 CID B105F00D PID 001BB101 TSG
Cortex-M33 identified.
J-Link>erase
Without any given address range, Erase Chip will be executed
Erasing device...
J-Link: Flash download: Total time needed: 1.314s (Prepare: 0.267s, Compare: 0.000s, Erase: 0.867s, Program:
0.000s, Verify: 0.000s, Restore: 0.179s)
J-Link: Flash download: Total time needed: 0.144s (Prepare: 0.070s, Compare: 0.000s, Erase: 0.036s, Program:
0.000s, Verify: 0.000s, Restore: 0.037s)
Erasing done.
J-Link>loadfile c:\tmp\RM1261_Factory_Image_480-00323-R127.1.0.15.s37
Downloading file [c:\tmp\RM1261_Factory_Image_480-00323-R127.1.0.15.s37]...
J-Link: Flash download: Bank 0 @ 0x00000000: 2 ranges affected (212992 bytes)
J-Link: Flash download: Total: 2.441s (Prepare: 0.216s, Compare: 0.059s, Erase: 0.000s, Program & Verify:
1.995s, Restore: 0.170s)
J-Link: Flash download: Program & Verify speed: 104 KB/s
O.K.
J-Link>
```

Further information and resources (i*ncluding Firmware, Certificates, Reports, Software, Application Notes, User Guide and more*) relating to the RM126x Series module is available on our product page of the Ezurio website under: https://www.ezurio.com/wireless-modules/lorawan-modules-solutions/rm126x-ultra-low-power-lorawan-a-b-c-module.

7 References

Ref	Details
[A]	RM126x Firmware GitHub https://github.com/LairdCP/RM126x_Firmware
[B]	RM126x Firmware GitHub (Releases) https://github.com/LairdCP/RM126x_Firmware/releases

8 Additional Information

Please contact your local sales representative or our support team for further assistance:

Headquarters	Ezurio 50 S. Main St. Suite 1100 Akron, OH 44308 USA
Website	http://www.ezurio.com
Technical Support	http://www.ezurio.com/resources/support
Sales Contact	http://www.ezurio.com/contact

Note: Information contained in this document is subject to change.

Ezurio's products are subject to standard Terms & Conditions.