Flasher
Getting started

Document: UM08022
Software Version: 9.56
Revision:
Date: July 1, 2026

A product of SEGGER Microcontroller GmbH

www.segger.com

Table of contents

Chapter 1

Introduction

Thank you for choosing Flasher as your in-system programmer.

This manual presents a quick start guide for Flasher and the associated Flasher Software and Documentation Pack, supported on Windows, Linux and macOS.

SEGGER's motto is "It simply works!", which also applies to getting started with in-system programmers and the related software.

To access the complete online Flasher User Guide, please click on the link below.

- *Flasher online User Guide*

Please also consider these helpful online resources:

- *Detailed Flasher model feature comparison*
- *Flasher home page on the SEGGER Knowledge Base*

> **Note**
>
> This guide focuses on the Flasher Pro, Flasher Compact, and Flasher Portable Plus models. The Flasher ATE2, Flasher Hub-4, and Flasher Hub-12 are out of scope for this guide.

Chapter 2

Features

SEGGER Flashers are packed with features. This chapter provides an overview.

2.1 Ultra-fast programming speeds

SEGGER's ultra-fast flash programming algorithms allow for the fastest flash writes in the industry. Flasher in-system programmers can flash at speeds reaching 1 Megabyte per second. SEGGER programmers aren't limited to internal flash; they can also flash external serial NOR flash memory, either through the target's SPI interface, or through direct SPI communication at up to 50 MHz.

- *More on flash programming speed*

2.2 Single-unit / gang programming

Depending on production volume and workflow, Flasher users can choose between single-unit and gang programming solutions optimized for their specific setup.

Single-unit in-system programming devices (e.g. Flasher Pro, Flasher Compact, or Flasher Portable Plus) are designed to program individual microcontrollers, SoCs, or memory components with specific firmware, making them ideal for production environments or in-field updates.

- *Single-unit programming solutions*

SEGGER also offers several gang programming solutions for high-volume in-system programming, such as the Flasher Hub-4 and the Flasher Hub-12 (in combination with Flasher Compacts), or the extremely space-saving Flasher ATE2.

- *Gang programming solutions*

2.3 Multiple operating modes

Each Flasher single-unit programmer is designed to operate in **standalone mode**, i.e. independently, without the need for a connected PC. This capability ensures that programming tasks are performed efficiently and reliably in any environment. The on-board memory on the Flashers allows for storing multiple firmware images.

The **PC-based mode** can be used in order to get a Flasher ready for programming. In this operating mode, it is connected to a PC via USB/Ethernet and controlled by a PC application (e.g. J-Flash).

File access mode can be entered if the Flasher "PROG" button is kept pressed for at least 2 seconds while connecting the Flasher to a host PC via USB. In this mode, the Flasher enumerates as a mass storage device (like an USB Stick) with the host PC. This allows for configuration and data files to be manually placed on the Flasher, and for reading out the Flasher log file.

2.4 Production programming features

Flashers provide several advanced features that are very useful for production programming.

For example, the Flasher supports storing multiple projects on its file system, making it possible to switch between projects purely in standalone mode without the need to update any files or connect the Flasher to a host PC. Each project consists of one or more jobs, which means that the Flasher supports **batch programming**.

Flasher devices also allow programming of variable information, such as serial numbers, Ethernet hardware addresses (MAC), digital signatures, and license keys. These options can be customized for each device by applying **patch data** to the original firmware.

2.5 Easy integration

Flashers are very flexible and easy to integrate into various programming environments. In PC-based mode, Flashers connect to a host computer running Windows, Linux, or macOS using the USB or Ethernet interface (depending on the Flasher model). In standalone mode, Flashers can be controlled via the "PROG" button, or remotely (via handshake control or ASCII interface). The ASCII Interface is available via USB (VCOM), IP (Telnet), RS232, or the Flasher PC software. All Flashers have a 20-pin target connector und support a large variety of target interfaces.

2.6 Extensive core / architecture support

SEGGER Flasher programmers support devices with a wide variety of cores / architectures, including 32-bit and 64-bit Arm Cortex-A and Cortex-R, Cortex-M, legacy ARM7/9/11, Renesas RX, 32-bit and 64-bit RISC-V, Silicon Labs 8051, Microchip PIC32Mx (MIPS), and more.

2.7 Extensive device support

The list of manufacturers, families, devices, and SoCs supported by the Flasher includes tens of thousands of devices in hundreds of device families. Furthermore, Flashers can program on-chip memory (on an MCU/SoC), as well as external serial or parallel NOR Flash. External serial NOR Flash can either be programmed indirectly (via the MCU/SoC the memory is connected to), or directly (without MCU/SoC in between).

- *List of supported devices*

2.8 Authorized Flashing

Authorized Flashing is a security feature for SEGGER Flasher in-system programmers that helps product owners protect their IP. Authorized Flashing employs a hidden partition (a secure area) on the Flasher that ensures that private information remains protected. Additionally, it enables users to determine the number of programming cycles in order to limit the number of devices that can be programmed.

- *More about Authorized Flashing*

2.9 Comprehensive software package

The Flasher Software and Documentation Package includes a variety of tools that extend the capabilities of the Flasher. Almost all Flasher tools have multi-platform support and run on Windows, Linux and macOS. All SEGGER programmers include free software and firmware updates.

- *Download the Flasher SW and Documentation Package*

2.10 Flexibility and extendibility

The Flasher SDK lets developers create Flasher Apps that run directly on SEGGER Flasher programmers. These apps can handle programming, system tests, and diagnostics to ensure efficient workflows and high product quality. The Flasher SDK also enables users to add support for new or unsupported devices via Flasher Device Packs.

- *More on the Flasher SDK*

Chapter 3

First steps

This chapter describes how to get up and running with Flasher.

3.1 Downloading and installing the Flasher Software and Documentation Pack

The Flasher Software and Documentation Pack is required for proper operation of the Flasher. It can be downloaded from the following web page:

- *Flasher Software and Documentation Pack*

3.1.1 Windows

Download the installer for the x86, x64 or arm64. Double-click on the downloaded file to begin the installation process, and follow the instructions provided.

By default, the Flasher SW Pack will be installed into the `Program Files\SEGGER\Flasher_Vxxx` folder. It can also be accessed via the Windows Start menu.

3.1.2 MacOS

Download the installer for the x64 or the Apple silicon (or download the universal installer). Double-click on the downloaded file to begin the installation process, and follow the instructions provided. The Flasher SW Pack will be installed into the `Applications/SEGGER/Flasher_Vxxx` folder.

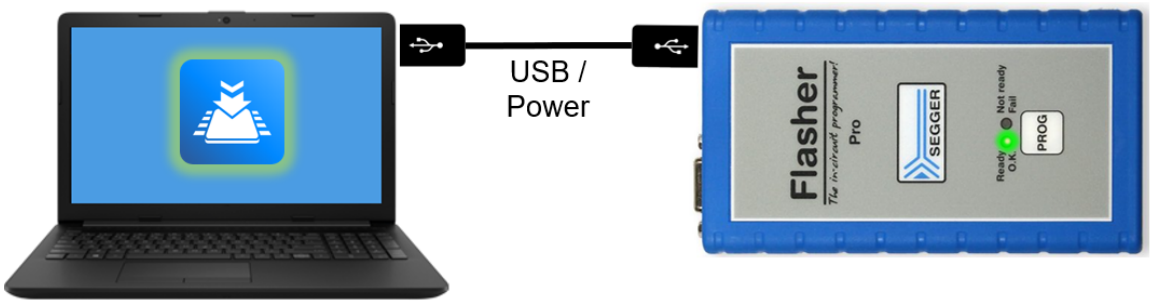
3.1.3 Linux

The Linux version of the Flasher Software Package is available for x86, x64, arm32 & arm64. The package is available as either DEB, RPM or TGZ files. It is entirely self-contained, requiring no external libraries, and as such can be used on every Linux distribution. Download the desired / appropriate file and unpack the downloaded archive to a desired location, such as `/opt/SEGGER/Flasher`, or `/usr/local/SEGGER/Flasher`.

3.2 Connecting to the Flasher via USB

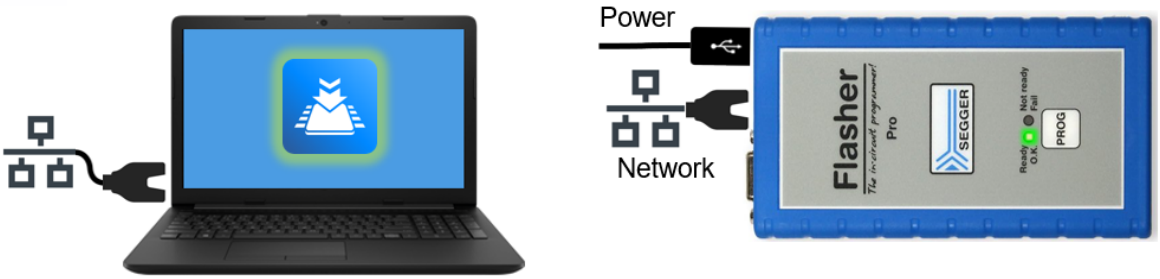
You can connect the Flasher to your host computer using the included USB cable. This will also power the programmer.

On the Flasher Pro (shown below) and Flasher Compact, the green `Ready / OK` LED will light up and slightly flicker once a second to indicate proper USB enumeration. On the Flasher Portable Plus, the display will indicate that the Flasher is ready.



3.3 Connecting to the Flasher via Ethernet

If you have a Flasher Pro, you can also connect to this Flasher model via Ethernet. You still need to connect the Flasher to a 5V power source via the USB connector. This will light up the green Ready / OK LED. Then connect the Flasher Pro to your local network via the Ethernet interface.



The Flasher Pro will obtain an IP address via DHCP, after which the green Ready / OK LED will slightly flicker once a second. You can check the IP address assigned to the Flasher Pro using the Flasher Configurator utility (also see further below).

> **Note**
>
> Your computer and the Flasher need to be on the same subnet.

3.4 Registering the Flasher

It is highly recommended to register the Flasher. Registration is quick, easy, and has the following benefits:

- Verify that your Flasher is an authentic SEGGER product
- Receive up-to-date information on software and product updates
- Gain access to the SEGGER technical support system
- *Knowledgebase article for instructions on how to register*
- *Knowledgebase article for information on technical support for SEGGER programmers*

3.5 Verifying the connection to the Flasher

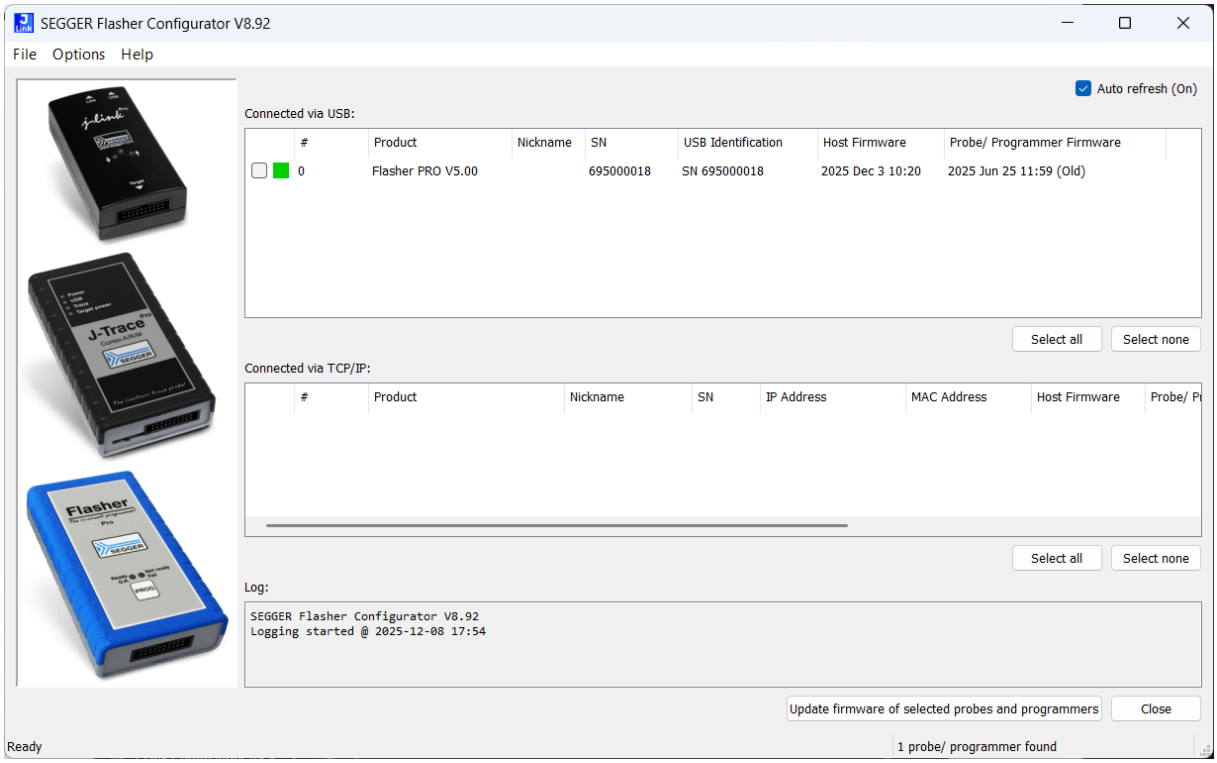
You can use the Flasher Configurator utility to verify that the connection to the Flasher is working properly. Flasher Configurator is part of the Flasher SW Pack you installed earlier.

To start Flasher Configurator under Windows, access the `Start` menu and select Flasher Configurator from the list of available software utilities in the `SEGGER - Flasher Vx.xx` folder.

When using macOS, navigate to the `/Applications/SEGGER/Flasher_Vxxx` folder and double-click `FlasherConfigExe`.

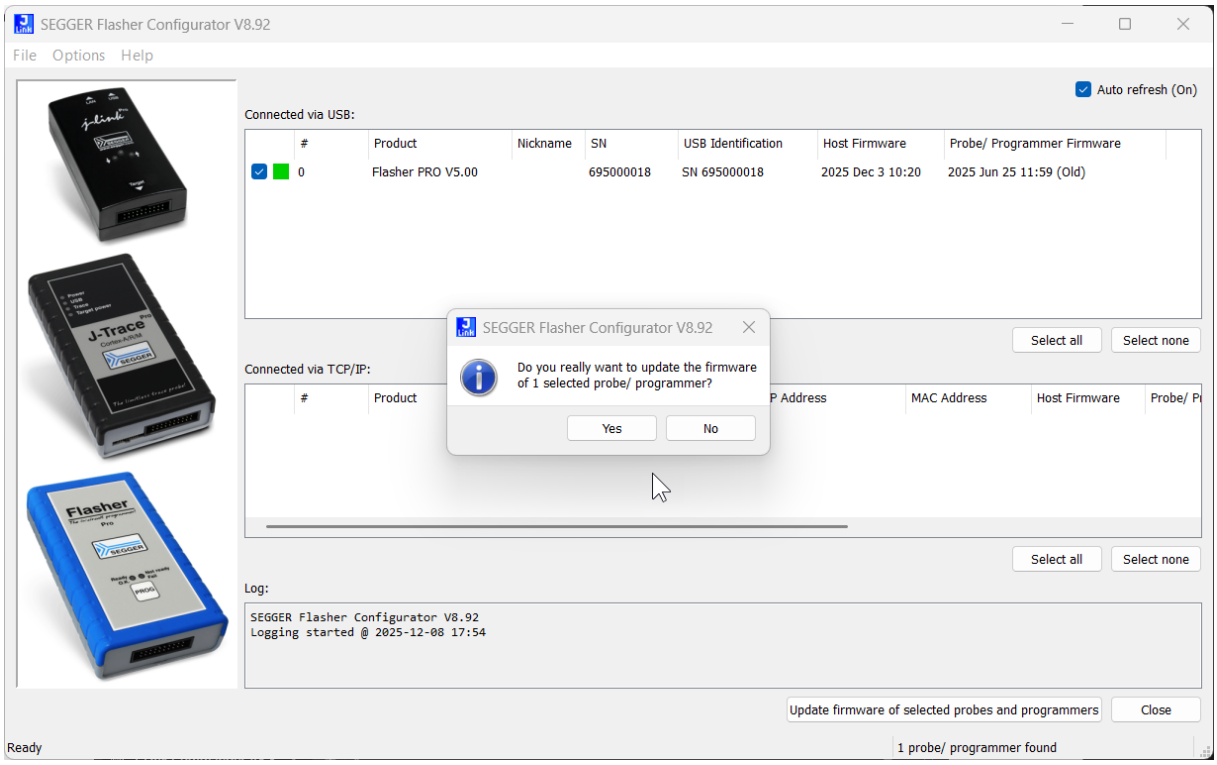
Under Linux, open a terminal and navigate to the Flasher SW installation directory using the cd command, e.g. `cd /opt/SEGGER/Flasher`. Then run the Flasher Configuratot executable: `./FlasherConfigExe`.

Flasher Configurator detects the Flasher (connected via USB in this example) and provides information about the connected programmer:

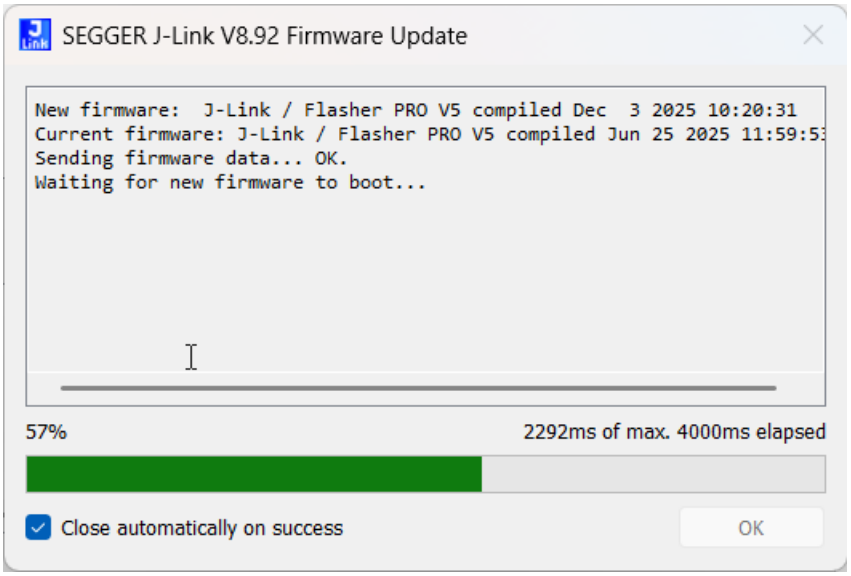


If Flasher Configurator detects that a firmware update is available for the Flasher, it will indicate this by marking the programmer firmware version as "Old" (as shown in the screenshot above).

In this case, you can update the Flasher firmware by selecting the Flasher in the list via the checkbox and clicking on `Update firmware`.

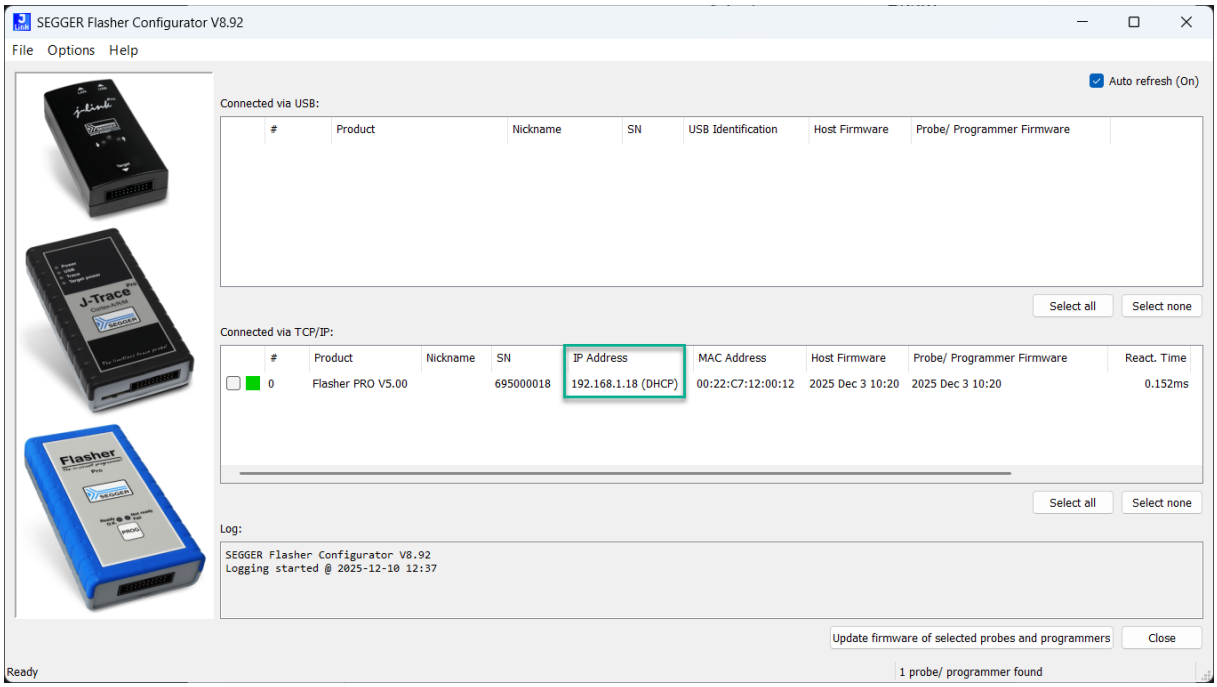


Click Yes to allow Flasher Configurator to update the firmware. You will see the firmware update process:



Flasher Configurator will confirm the successful firmware update in its log window.

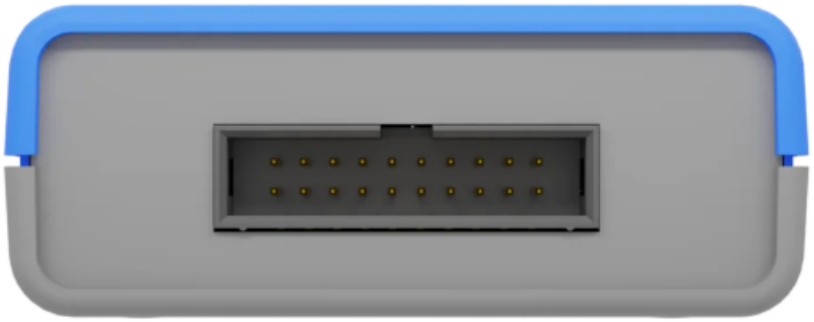
If you have a Flasher Pro and you have connected it to your local network via Ethernet, Flasher Configurator will allow you to verify that connection and see the IP address that was assigned to the Flasher.



Close down Flasher Configurator after you're done with it, or at least turn off the auto-refresh feature. Since Flasher Configurator constantly looks for SEGGER probes and programmers connected via USB or Ethernet if the auto-refresh is enabled, it might interfere with the operation of these probes / programmers at the same time.

3.6 Flasher target connector

The Flasher's 20-pin, 0.1"-pitch target connector is shown in the image below:



The pinout for the connector is flexible and depends on the target interface used.

> **Note**
>
> U-Flash provides wiring diagrams for the selected device / target interface. These wiring diagrams are also visible in the supported device list on the web at https://www.segger.com/supported-devices/.

The Flasher ships with a matching 20-pin ribbon cable. Whether or not you will need a separate programming adapter depends on the type of programming header on the target board.

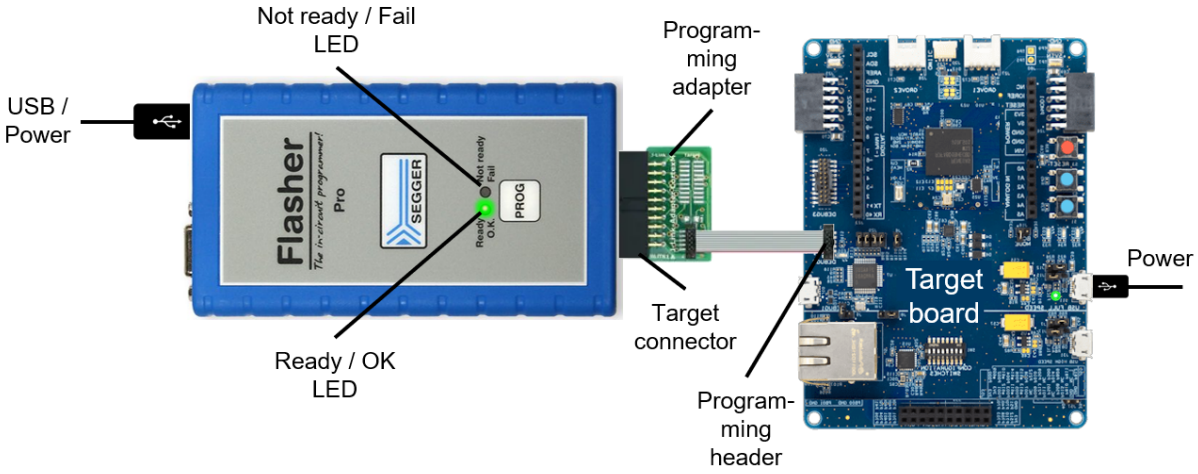
You can see the extensive list of programming adapters available from SEGGER here (also applies to J-Link debug probes):

- *List of available programming adapters*

Of course you can also create your own custom programming adapter to fit your exact needs.

3.7 Connecting the Flasher to a target board

Below is an example for connecting the Flasher to a target board:



Connect the programming adapter (or the 20-pin ribbon cable, if possible) to the Flasher target connector as well as the programming header on the target board as shown in the image above. Watch out for the correct orientation. The colored wire of the ribbon cable needs to connect to pin 1 of the programming header. Also make sure the connector covers both rows of the programming header.

Then provide power to the target board.

> **Note**
>
> It is also possible to power your target board via the Flasher programmer. You can enable this feature when you set up your project in the U-Flash or J-Flash utilities. Several power-related target adapters are available from SEGGER.

3.8 Creating a Flasher project

SEGGER U-Flash is the primary software utility to use in combination with SEGGER Flasher programmers.

- *SEGGER U-Flash web page*

U-Flash provides an interface for configuring Flasher production programmers for use. Within the interface, users can select single or batch projects (i.e., the programming of a single device/memory region or multiple devices/memory regions) and access all necessary data files for programming and verification.

U-Flash is part of the Flasher SW and Documentation pack.

- *Flasher SW and Documentation Pack*

Please refer to the U-Flash User Guide for information on how to set up a Flasher project with U-Flash.

- *SEGGER U-Flash User Guide*

The steps involved in creating a Flasher project in U-Flash are:

- Selecting a device
- Adding a data file
- Configuring common and device-specific options
- Selecting a Flasher
- Downloading the project files to the Flasher

> **Note**
>
> Depending on the target device and the use case, it is also possible to use the SEGGER J-Flash or J-Flash SPI utilities to create projects for a Flasher programmer; however, U-Flash is the primary utility recommended to be used with Flasher.

> **Note**
>
> For information on how to manage multiple projects on a Flasher, please see here:
>
> - *Flasher Project Management*

3.9 Programming a target device

There are a variety of ways to trigger a programming operation on a SEGGER Flasher. Below is an overview, along with the associated options for connecting to the Flasher from a host computer, and information on whether or not the programming operation can be triggered manually and/or in an automated fashion.

		Options for triggering a programming operation	Can the programming operation be triggered manually?	Can the programming operation be triggered programmatically?
Options for connecting to the Flasher from a host computer	No connection to a host computer	via the PROG button	Yes	No
	USB	via virtual COM port	Yes (via a serial terminal program)	Yes (via a script / an application that establishes a VCOM connection to the Flasher)
		via the SEGGER U-Flash utility	Yes (via the U-Flash GUI)	Yes (via a script / an application that utilizes the U-Flash command line interface)
	Ethernet [1)]	via Telnet [1)]	Yes (via a Telnet client)	Yes (via a script / an application that establishes a Telnet connection to the Flasher)
		via the SEGGER U-Flash utility	Yes (via the U-Flash GUI)	Yes (via a script / an application that utilizes the U-Flash command line interface)
	UART / RS232 [1)]	via Serial Port [1)]	Yes (via a serial terminal program)	Yes (via a script / an application that establishes a serial connection to the Flasher)
	Handshake signals via 9-pin SUB-D connector [1)]	via the handshake protocol [1)]	No	Yes (via a script or an application that establishes a handshake connection to the Flasher)

[1)]: Option not available on Flasher Compact / Flasher Portable Plus

> **Note**
>
> Since the Flasher Portable Plus is a hand-held service programmer, the primary way of programming a target device with this model is by pressing the PROG button.

Information about how to control the Flasher remotely can be found here:

- *Flasher Remote Control*

Information about how to automate programming with the Flasher can be found here:

- *Automating Flasher production setups*

Congratulations, you are now up and running with the SEGGER Flasher!

Chapter 4

Working with Flasher

The previous chapter provided you with a good foundation of how to get up and running with the Flasher. This chapter contains some further hints and tools on how to work with the Flasher.

Furthermore, the **Flasher Software Package** includes a large number of additional useful tools and utilities for working with Flasher. Please see chapter *Software Package Overview* on page 19 for details.

4.1 Flasher User Guide - Working with Flasher

The Flasher User Guide on the SEGGER Knowledge Base contains important information on how to work with the Flasher. This chapter is particularly helpful in working with the Flasher:

- *Flasher User Guide - Working with Flasher*

4.2 Flasher BitStreamer

Flasher BitStreamer is SEGGER's software tool for easy and reliable in-system programming of FPGAs and CPLDs using any Flasher production programmer. It converts industry-standard SVF and STAPL files into ready-to-run packages that can be deployed directly to a Flasher or exported as an archive for the Flasher Deployer, within seconds and without complex setup or scripting.

With Flasher BitStreamer, SEGGER Flashers can program microcontrollers, memories, and programmable-logic devices from various vendors via the standard 4-pin JTAG interface, all using the same reliable hardware platform.

- *Flasher BitStreamer web page*
- *Flasher BitStreamer User Guide*

4.3 Flasher SDK

The Flasher SDK lets developers create Flasher Apps that run directly on SEGGER Flasher programmers. These Apps handle programming, system tests, and diagnostics to ensure efficient workflows and high product quality.

The Flasher SDK also enables users to add support for new or unsupported devices via Flasher Device Packs.

- *Flasher SDK web page*

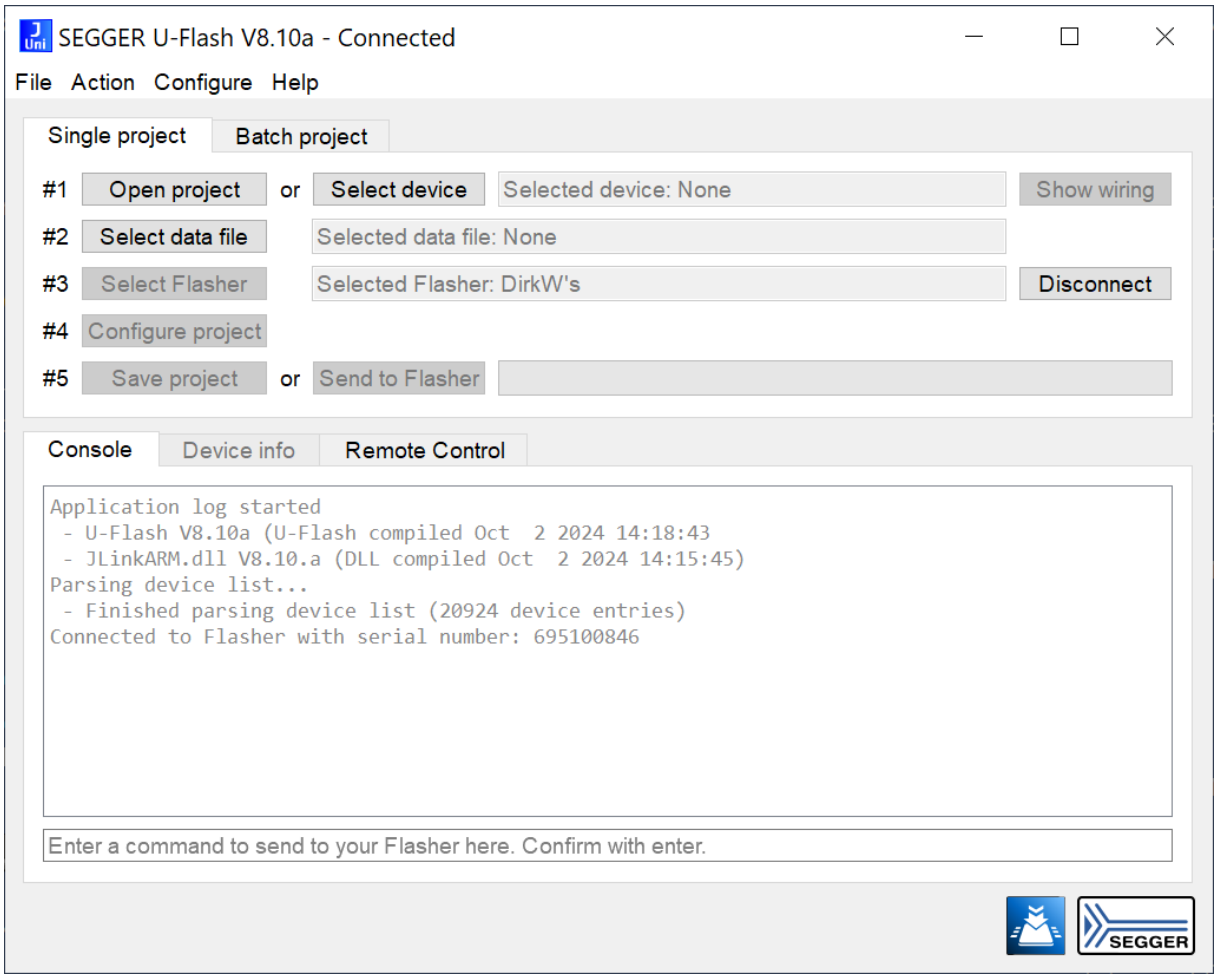
Chapter 5

Software Package Overview

This chapter covers the contents of the Flasher Software Pack, an extensive array of software to fully complement the Flasher.

5.1 U-Flash / U-Flash CL

U-Flash (formerly known as Universal Flash Loader Configurator) is a multi-platform (Windows, Linux and macOS) software utility to prepare SEGGER Flashers for standalone programming of more than 20,000 programmable devices via a variety of target interfaces. It includes a device database and an intuitive user interface for configuration. The configuration and data files required for programming can be directly downloaded via USB or Ethernet to a connected SEGGER Flasher. U-Flash allows saving the configuration settings into a project file for later use, so the image file can easily be updated if needed. It is also possible to trigger programming cycles directly from within U-Flash.



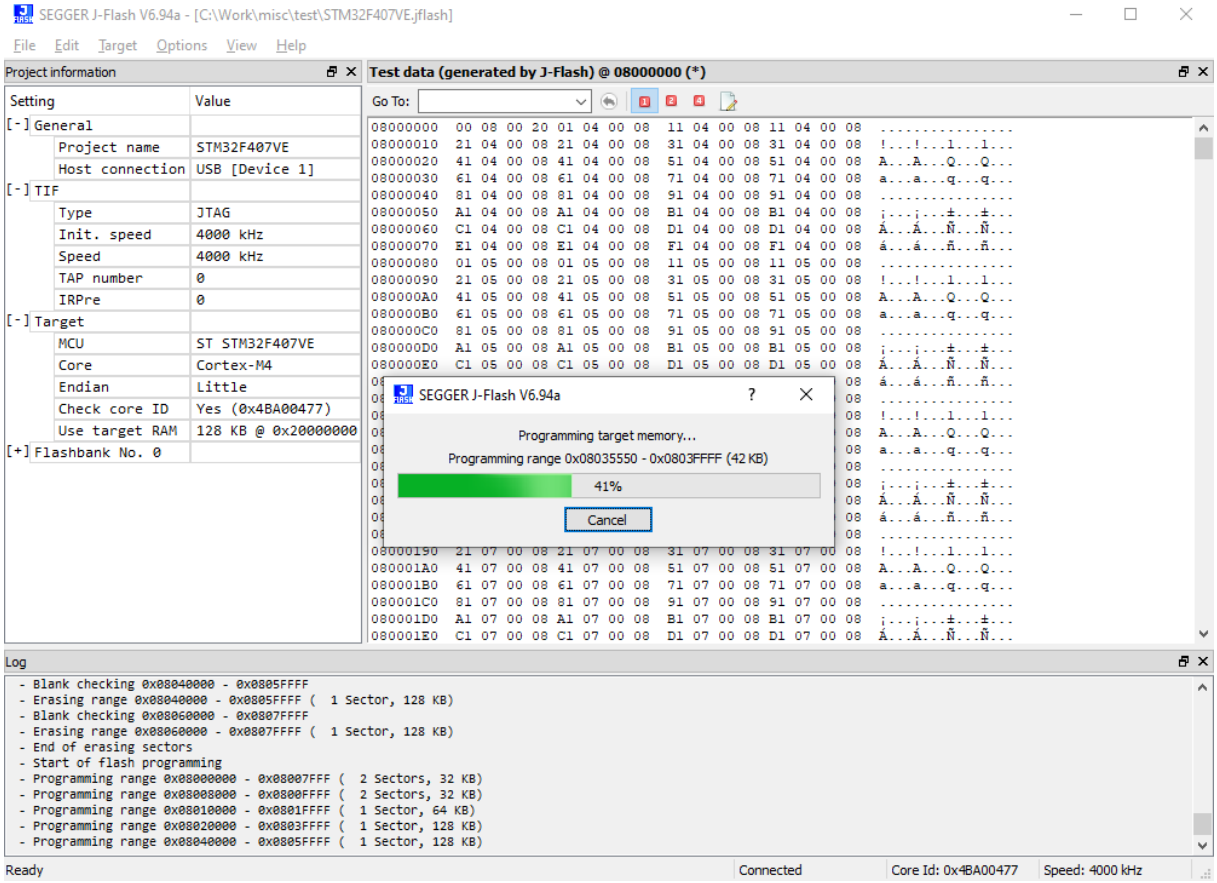
U-Flash CL is the command-line version of U-Flash. The U-Flash command line interface allows users to configure and control Flashers from a terminal or from within a script. It can be used to:

- Download projects to a Flasher
- Trigger standalone mode programming
- Automate production workflows
- Integrate U-Flash into CI/CD pipelines Also see:
- *U-Flash product page*
- *U-Flash User Guide*

5.2 J-Flash

J-Flash is a multi-platform (Windows, Linux and macOS) application used to program the internal flash of the MCU, as well as external SPI flash connected to the MCU, via a SEGGER Flasher, J-Link, or J-Trace.

J-Flash comes with sample projects that run out-of-the-box for most popular microcontrollers and evaluation boards. J-Flash can be controlled via GUI or via command line, which also makes it possible to use J-Flash for production purposes.

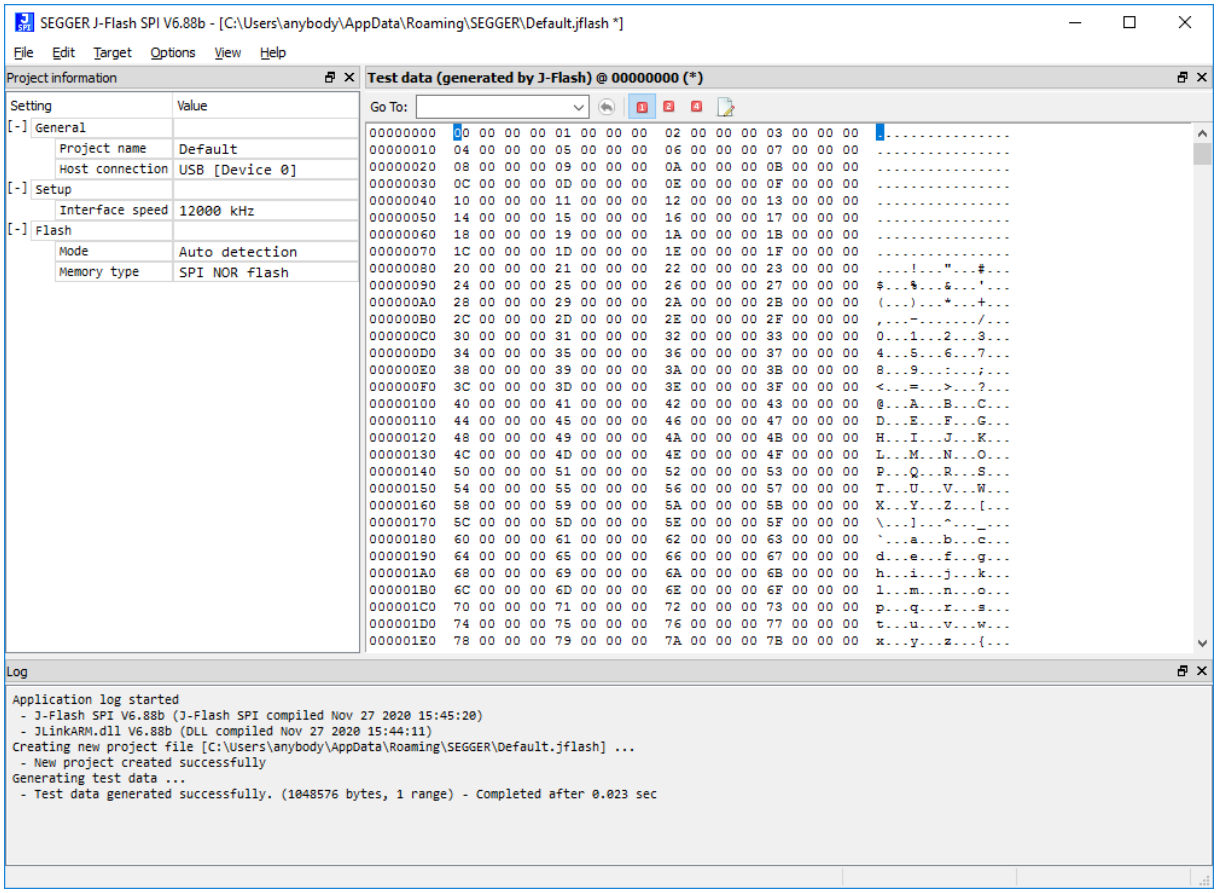


- *J-Flash product page*
- *J-Flash documentation*
- *J-Flash introductory video*

Please note that the use of J-Flash requires a license. This license is included with the purchase of any Flasher model.

5.3 J-Flash SPI / J-Flash SPI CL

J-Flash SPI is a flash programming software for Windows, Linux or macOS. It allows direct programming of SPI flashes via a SEGGER Flasher, J-Link, or J-Trace, without going through an MCU (as is the case with J-Flash). J-Flash SPI has an intuitive user interface and makes programming flash devices convenient. J-Flash SPI is able to auto-detect and program all kinds of SPI flashes, even if the MCU they are connected to is not supported by Flasher / J-Link / J-Trace. This is because J-Flash SPI communicates directly with the SPI flash, bypassing all other components of the hardware. Because the SPI flash is programmed in-circuit by J-Flash SPI without any MCU involvement, it programs at the fastest speed possible.



J-Flash SPI CL is a command-line-only version of the J-Flash SPI programming tool. Except for the missing GUI, J-Flash SPI CL is identical to the GUI version. The commands used to configure / control J-Flash SPI CL are exactly the same as for the command line interface of the J-Flash SPI GUI version.

Both J-Flash SPI and J-Flash SPI CL are part of the Flasher Software and Documentation Pack available from segger.com.

- *J-Flash SPI product page*
- *J-Flash SPI documentation*

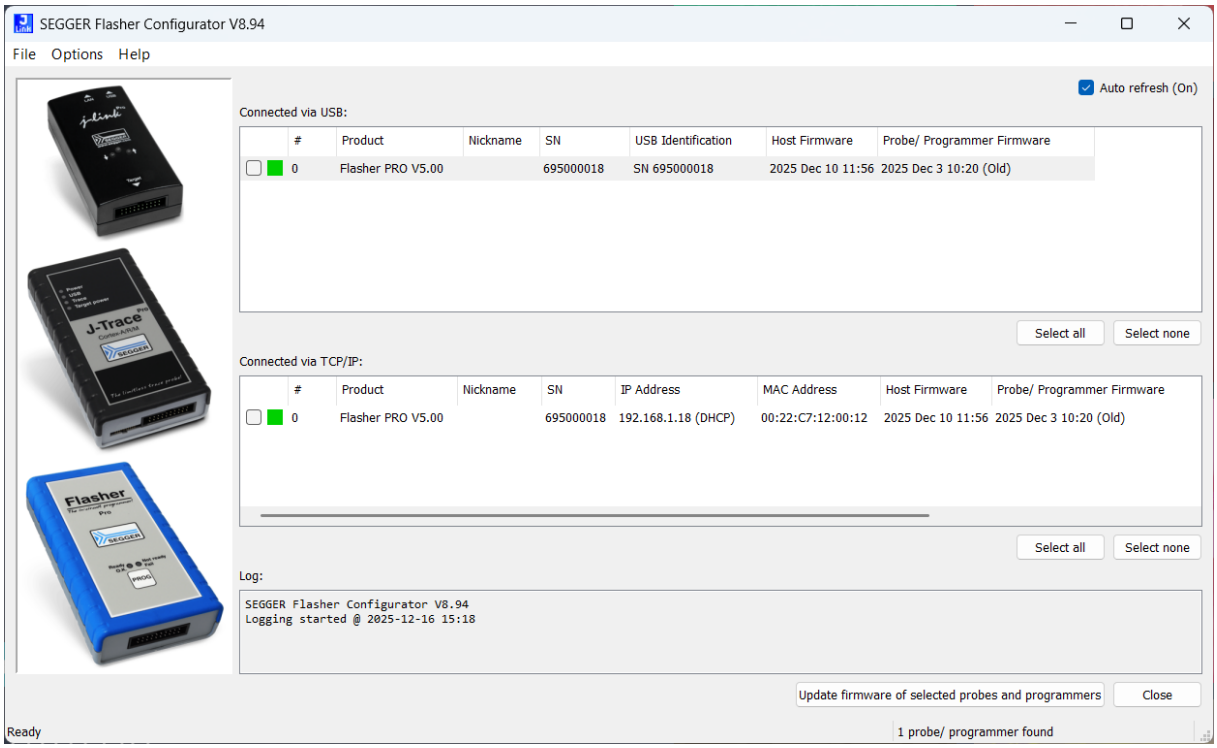
Please note that the use of J-Flash SPI / J-Flash SPI CL requires a license. This license is included with the purchase of any Flasher model.

5.4 Flasher Configurator

Flasher Configurator is a multi-platform (Windows / Linux / macOS) GUI application to configure certain features of a SEGGER Flasher. It can also be used to configure J-Link debug probes / J-Trace trace probes.

The following Flasher settings can be changed via Flasher Configurator:

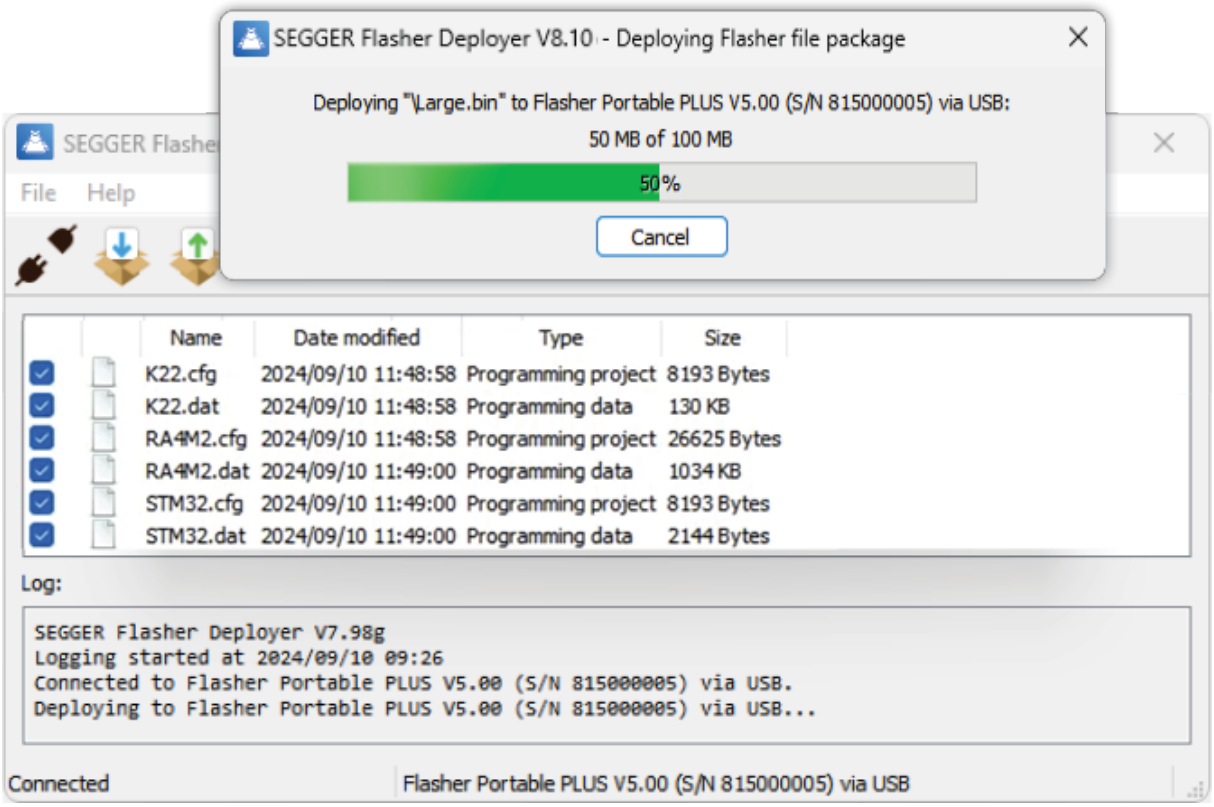
- Updating the firmware of one or multiple SEGGER Flasher programmers connected via USB or TCP/IP
- Configuring of the IP settings (use DHCP, IP address, subnet mask, …) of a Flasher supporting the Ethernet interface
- Configuring the target reference voltage (automatic / fixed)
- Enabling / disabling a secure partition (for Authorized Flashing)
- Adding a nickname to a SEGGER Flasher



- *J-Link / Flasher Configurator product page*
- *J-Link / Flasher Configurator documentation*

5.5 Flasher Deployer

Flasher Deployer is an easy-to-use GUI tool designed to simplify file management and deployment across multiple flash programmers. With its versatility, it enables users to create Flasher Setup Packages from any given Flasher, making it easier than ever to replicate an existing, working setup for distribution to large fleets of Flashers for production lines and service teams. This makes it an ideal solution for situations where multiple programmers need to be configured with the same target firmware image(s) at the same time. By allowing an entire fleet of flash programming devices â whether in production lines or in the field â to be deployed at the same time, the tool helps eliminate the risk of having mixed operation modes across distributed devices.



- *Flasher Deployer product page*
- *Flasher Deployer documentation*

5.6 Flasher Control

Flasher Control is a command line utility that allows accessing the Flasher terminal via USB or IP, rather than COM port, RS232 or Telnet.

Flasher Control is primarily intended for use with Flasher Compact in solitary mode (without Flasher Hub), or generally for troubleshooting when the terminal of the Flasher is not available via other channels. It also contains a rudimentary terminal mode, which makes the use of a special terminal program unnecessary in simple cases.

Several targets can be programmed simultaneously and in parallel by starting multiple instances of FlasherControl, each connected to a different Flasher.

```
C:\Work_Repo\JLinkARM\Output\Release_x64>FlasherControl.exe -I
>#read
#ACK
#STATUS:INITIALIZING
#STATUS:CONNECTING
#INFO:Verbosity: extreme
#INFO:20230413 P12F50X.PEX
#INFO:Warning! No MOVLW instr  00000F83
#INFO:OSCCALValBackup:         00000000
#STATUS:READING
#INFO:**********************************
#INFO:*** Read Program Area         ***
#INFO:Read        @:  00000000
#INFO:Read      Len:           00000200
#D#DATA:S3250000030081018201830184018501860187018801890 18A018B018C018D018E018F0190013F
#DATA:S3250000032091019201930194019501960197019801990 19A019B019C019D019E019F01A0011F
#DATA:S325000003401 01A201A301A401A501A601A701A801A901 AA01AB01AC01AD01AE01AF01B001FF
#DATA:S325000003601 01B201B301B401B501B601B701B801B901 BA01BB01BC01BD01BE01BF01C001DF
#DATA:S325000003801 01C201C301C401C501C601C701C801C901 CA01CB01CC01CD01CE01CF01D001BF
#DATA:S3250000003A0D101D201D301D401D501D601D701D801D901DA01DB01DC01DD01DE01DF01E0019F
#DATA:S3250000003C0E101E201E301E401E501E601E701E801E901EA01EB01EC01ED01EE01EF01F0017F
#DATA:S3250000003E0F101F201F301F401F501F601F701F801F901FA01FB01FC01FD01FE01000A830FC4
#INFO:**********************************
#INFO:*** Read UserId Area          ***
#INFO:Read        @:  00000200
#INFO:Read      Len:           00000004
#DATA:S30D00000400FA0FFF0FFA0FFF0FC0
#INFO:**********************************
#INFO:***          DeInit          ***
#INFO:--- No config words written   ---
#OK (Total 0.220s)
```

- *Flasher Control documentation*

5.7 Flasher App Runner

Flasher App Runner is a command line application designed to load and execute Flasher Apps on a Flasher.

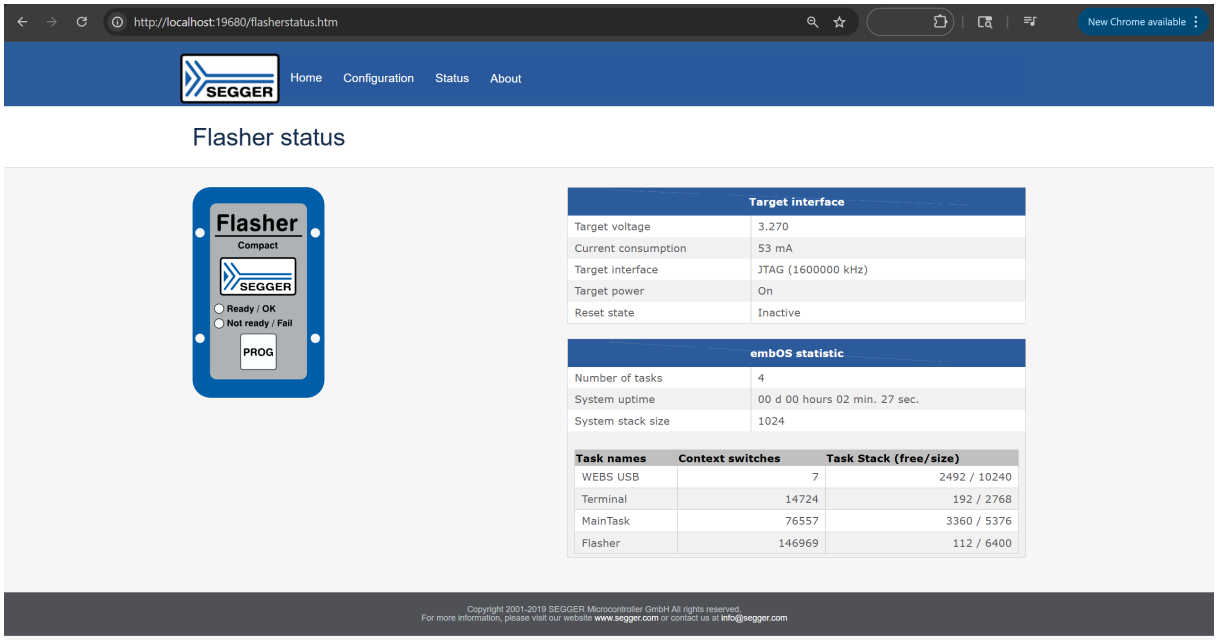
```
C:> flasherrun DemoHelloWorld.pex
SEGGER Flasher App Runner Utility V8.90 (Compiled Nov 26 2025 17:52:51)
Copyright (c) 2025-2025 SEGGER Microcontroller GmbH    www.segger.com
DLL version V8.90, compiled Nov 26 2025 17:29:40

Connecting to Flasher via USB...O.K.
Firmware: J-Link / Flasher Compact V7 compiled Nov 26 2025 15:58:58
Hardware version: V7.00
Flasher uptime (since boot): 0d 00h 02m 39s
S/N: xxxxxxxxxx
License(s): JFlash, GDB
USB speed mode: High speed (480 MBit/s)
VTref=0.000V
Successfully loaded '\Work\SEGGER\emApps\Apps\Src\DemoHelloWorld.pex'.
Executing app... (press <Enter> to cancel)
  Hello, world!

App execution finished with return value 0 after 0.031ms.

C:> flasherrun DemoDhrystone_100k.pex
SEGGER Flasher App Runner Utility V8.90 (Compiled Nov 26 2025 17:52:51)
Copyright (c) 2025-2025 SEGGER Microcontroller GmbH    www.segger.com
DLL version V8.90, compiled Nov 26 2025 17:29:40

Connecting to Flasher via USB...O.K.
Firmware: J-Link / Flasher Compact V7 compiled Nov 26 2025 15:58:58
Hardware version: V7.00
Flasher uptime (since boot): 0d 00h 00m 49s
S/N: xxxxxxxxxx
License(s): JFlash, GDB
USB speed mode: High speed (480 MBit/s)
VTref=0.000V
Successfully loaded '\Work\SEGGER\emApps\Apps\Src\DemoDhrystone_100k.pex'.
Executing app... (press <Enter> to cancel)
  30969 Dhrystones/s, 17 DMIPS

App execution finished with return value 0 after 3229.743ms.

C:> _
```

- *Flasher App Runner documentation*

5.8 Flasher USB Web Server

Flasher USB Web Server enables a Flasher to provide its web interface, accessible via any browser, over USB, without the need to be connected to a network via Ethernet or Wi-Fi.

Flasher USB Web Server is powered by SEGGER's emUSB-Web.

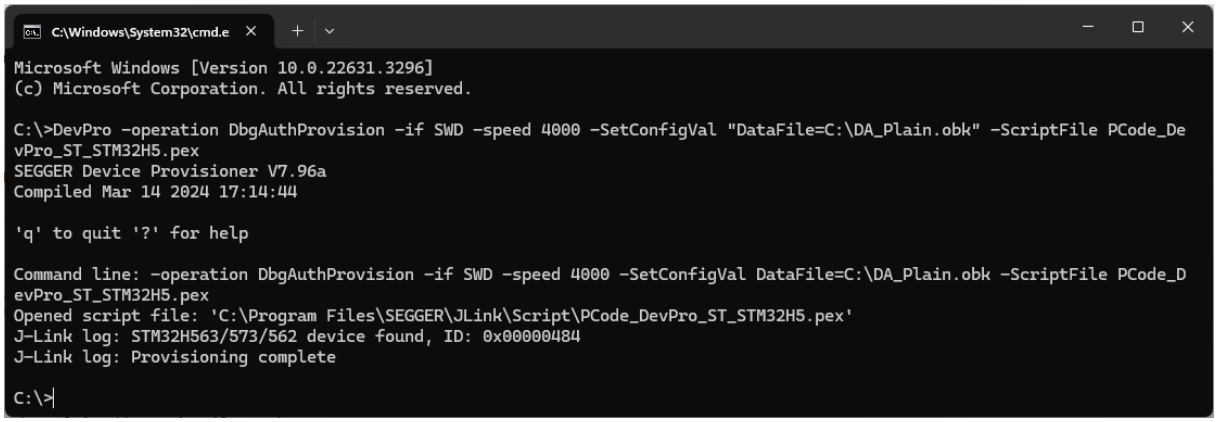


- *Flasher USB Web Server documentation*

5.9 Device Provisioner

Device Provisioner is a command line tool for Flasher (and J-Link / J-Trace), ensuring devices are properly set up and configured for use. The provisioning process includes tasks such as initializing hardware, installing software, configuring settings, and sometimes associating the device with a specific user or network.

Created to seamlessly integrate into automation environments, Device Provisioner executes commands from scripts written in C, which can be provided by SEGGER, silicon vendors, or created by users. These scripts can be executed on Flasher programmers while connected to a host PC. To protect intellectual property, script files can be distributed in source code or pre-compiled form.



- *Device Provisioner product page*
- *Device Provisioner documentation*

Chapter 6

Summary

We hope this guide provided you with a solid understanding of how to get started and how to work with the Flasher programmers from SEGGER. Below are the most commonly accessed pages for further reading.

- *Complete Flasher online User Guide*
- *How to register a Flasher*
- *Flasher SW and Documentation Package*
- *Flasher model feature comparison*
- *List of devices supported by Flasher*