Flasher Hub
User Guide

Control Module for Parallel Programming

Document: UM08039
Manual Version: 1.08
Revision: 1
Date: November 29, 2022

A product of SEGGER Microcontroller GmbH

www.segger.com

Disclaimer

The information written in this document is assumed to be accurate without guarantee. The information in this manual is subject to change for functional or performance improvements without notice. SEGGER Microcontroller GmbH (SEGGER) assumes no responsibility for any errors or omissions in this document. SEGGER disclaims any warranties or conditions, express, implied or statutory for the fitness of the product for a particular purpose. It is your sole responsibility to evaluate the fitness of the product for any specific use.

Copyright notice

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of SEGGER. The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such a license.

© 2020-2022 SEGGER Microcontroller GmbH, Monheim am Rhein / Germany

Trademarks

Names mentioned in this manual may be trademarks of their respective companies.

Brand and product names are trademarks or registered trademarks of their respective holders.

Contact address

SEGGER Microcontroller GmbH

Ecolab-Allee 5
D-40789 Monheim am Rhein

Germany

Tel. +49-2173-99312-0
Fax. +49-2173-99312-28
E-mail: support@segger.com
Internet: www.segger.com

Manual versions

This manual describes the Flasher Hub, and (in part) the Flasher Compact, which can be connected to the Flasher Hub.

For further information on topics or routines not yet specified, please contact us.

Print date: November 29, 2022

Manual version	Revision	Date	By	Description
1.08	1	221129	JB	Chapter "ASCII command interface" Added new commands.
1.08	0	211129	LG	Updated user guide version number from 1.04 to 1.08. Chapter "Web server" Updated screenshots. Added notes for settings that require "Administrator mode" to be changed. Added section "Flasher Hub web interface configuration page".
1.04	3	211111	LG	Chapter "Working with the Flasher Hub" Added section "Fail-safe mode"
1.04	2	211111	LG	Chapter "Working with the Flasher Hub" Section "Operating modes" Improved phrasing for explanation of operating modes. Added "web interface" as another operating mode.
1.04	1	211026	LG	Chapter "ASCII command interface" Section "Command #AUTO" Removed "#DONE" response from example sequence. Section "Command #AUTO NOPATCH" Removed "#DONE" response from example sequence. Section "Command #AUTO PATCH" Removed "#DONE" response from example sequence. Section "Command #CANCEL" Removed "#DONE" response from example sequence. Section "Command #ERASE" Removed "#DONE" response from example sequence. Section "Command #PROGRAM" Removed "#DONE" response from example sequence. Section "Command #VERIFY" Removed "#DONE" response from example sequence.
1.04	0	211013	LG	Updated screenshots. Chapter "Working with the Flasher Hub" Added section "UART to TCP transceiver". Section "Seting up the IP interface" Improved formatting. Chapter "Serial number handling" Fixed typo. Chapter "Web server" Removed obsolete sub chapters. Added chapter "Status page". Added chapter "Flasher Hub configuration page". Added chapter "Flasher modules configuration page". Added chapter "Programming page". Added chapter "UART page". Chapter "Remote control" Section "Ascii command interface" Added "Command #TERMINAL".
1.02	0	211004	LG	Chapter "FTP server" Section "Access data" Fixed typo.
1.00	3	210610	LG	Updated screenshots. Removed obsolete contents.
1.00	2	210610	LG	Chapter "Serial number handling" updated. Section "Limiting the number of programming cycles" Added missing screenshot. Chapter "Remote control" updated. Section "Command #POWERON" Removed obsolete reference. Section "Command #POWEROFF"

 © 2020-2022 SEGGER Microcontroller GmbH

Manual version	Revision	Date	By	Description
				Removed obsolete reference.
1.00	1	210610	LG	Slightly adjusted structure, wording and updated screenshots
1.00	0	210603	AW	Initial version

© 2020-2022 SEGGER Microcontroller GmbH

About this document

Assumptions

This document assumes that you already have a solid knowledge of the following:

- The software tools used for building your application (assembler, linker, C compiler).
- The C programming language.
- The target processor.
- DOS command line.

If you feel that your knowledge of C is not sufficient, we recommend *The C Programming Language* by Kernighan and Richie (ISBN 0--13--1103628), which describes the standard in C programming and, in newer editions, also covers the ANSI C standard.

How to use this manual

This manual explains all the functions and macros that the product offers. It assumes you have a working knowledge of the C language. Knowledge of assembly programming is not required.

Typographic conventions for syntax

This manual uses the following typographic conventions:

Style	Used for
Body	Body text.
`Keyword`	Text that you enter at the command prompt or that appears on the display (that is system functions, file- or pathnames).
`Parameter`	Parameters in API functions.
`Sample`	Sample code in program examples.
`Sample comment`	Comments in program examples.
Reference	Reference to chapters, sections, tables and figures or other documents.
GUIElement	Buttons, dialog boxes, menu names, menu commands.
Emphasis	Very important sections.

© 2020-2022 SEGGER Microcontroller GmbH

 © 2020-2022 SEGGER Microcontroller GmbH

Table of contents

1 Introduction 10

1.1 Flasher Hub overview 11
1.1.1 Features of the Flasher Hub 11
1.1.2 Working environment 12
1.2 Specifications 14
1.2.1 Specifications for Flasher Hub 14
1.2.1.1 Supported CPU cores 14
1.2.1.2 Supported Target interfaces 14

2 Working with the Flasher Hub 15

2.1 The Flasher Hub 16
2.1.1 Power supply 16
2.1.1.1 Flasher Hub and Flasher Compact power supply 16
2.1.1.2 Target power supply 17
2.2 Setting up the IP interface 18
2.2.1 Connecting for the first time 18
2.3 Operating modes 19
2.3.1 Web interface 19
2.3.2 Remote-controlled mode 19
2.4 LED status indicators 20
2.4.1 LED indicators 20
2.4.2 Flasher Compact LED indicators 20
2.5 Storing configuration and programming data files 21
2.6 UART to TCP transceiver 23
2.7 Log files 24
2.8 Updating the Flasher Hub Firmware 25
2.9 Newline encoding 26
2.10 Fail-safe mode 27

3 Setting up a project for the Flasher Hub 28

3.1 Using J-Flash 29
3.2 Using Universal Flash Loader 33

4 Serial number handling 34

4.1 Serial number programming 35
4.1.1 Serial number settings 35
4.1.2 Continuous Serial numbers 36
4.1.3 Serial number list file 36
4.1.4 Programming process 37

 © 2020-2022 SEGGER Microcontroller GmbH

4.1.5 Example setup 38
4.2 Limiting the number of programming cycles 40
4.2.1 Modified fail/error LED indicator behavior 40

5 Patch data file 41

5.1 Patch file support 42

6 FTP server 43

6.1 FTP server connection 44
6.1.1 Access data 45

7 Web server 46

7.1 Web server features 47
7.1.1 Status page 47
7.1.2 Flasher Hub configuration page 48
7.1.3 Flasher modules configuration page 49
7.1.4 Flasher Hub web interface configuration page 50
7.1.5 Files page (File browser) 51
7.1.6 Programming page 52
7.1.7 UART page 53
7.1.8 Log page 54

8 Remote control 55

8.1 Overview 56
8.2 ASCII command interface 57
8.2.1 Introduction 57
8.2.2 General command and reply message format 57
8.2.3 General usage 57
8.2.4 Settings for ASCII interface via Telnet 57
8.2.5 Commands and replies 57
8.2.5.1 Commands to the Flasher Hub 59
8.2.5.1.1 Command #ASSIGN 59
8.2.5.1.2 Command #AUTO 60
8.2.5.1.3 Command #AUTO NOPATCH 61
8.2.5.1.4 Command #AUTO PATCH 62
8.2.5.1.5 Command #CANCEL 63
8.2.5.1.6 Command #ERASE 64
8.2.5.1.7 Command #FWVERSION 65
8.2.5.1.8 Command #FWVERSIONMOD 66
8.2.5.1.9 Command #IPCONFIG 67
8.2.5.1.10 Command #POWERON 68
8.2.5.1.11 Command #POWEROFF 69
8.2.5.1.12 Command #PROGRAM 70
8.2.5.1.13 Command #PROJECT 71
8.2.5.1.14 Command #PROTVER 72
8.2.5.1.15 Command #RESETIPCONFIG 73
8.2.5.1.16 Command #RESTART 74
8.2.5.1.17 Command #RESULT 75
8.2.5.1.18 Command #SELECT 76
8.2.5.1.19 Command #SELMODULE 77
8.2.5.1.20 Command #SERIAL 78
8.2.5.1.21 Command #SERIALMOD 79
8.2.5.1.22 Command #SETVTREF 80
8.2.5.1.23 Command #START 81
8.2.5.1.24 Command #STATUS 82
8.2.5.1.25 Command #TERMINAL 83
8.2.5.1.26 Command #VERIFY 84
8.2.5.2 Replies from the Flasher Hub 85

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.2.1 #ACK 85
8.2.5.2.2 #NACK 85
8.2.5.2.3 #OK:<module>:<data> 85
8.2.5.2.4 #STATUS:<status> 85
8.2.5.2.5 #RESULT:<module>:<data> 86
8.2.5.2.6 #DONE 86
8.2.5.2.7 #ERRxxx <data> 86

9 Support and FAQs 87

9.1 Contacting support 88
9.2 Frequently Asked Questions 89

10 Mechanics 90

11 Literature and references 91

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 1

Introduction

This chapter provides a short overview about the Flasher Hub and its features.

© 2020-2022 SEGGER Microcontroller GmbH

1.1 Flasher Hub overview

SEGGER's Flasher Hub is a Control Module for parallel programming.

The Flasher Hub is able to control up to 24 Flasher Compact modules serving as individual channels for parallel, high-speed gang-programming. If required, each channel can be configured to program a different device with a different firmware image.

The Flasher Compact modules connected to the Flasher Hub are set up just once per channel configuration using SEGGER's Flasher software package. The software to be used depends on the type of flash chip being programmed. The Flasher Hub can receive commands and send results via telnet conection "stand-alone", without the need of a desktop PC. The combination of a single Flasher Hub and multiple Flasher Compact modules is the perfect solution for high-volume mass production.

The Flasher Hub supports all flash devices and programming interfaces supported by the Flasher Compact. By using Flasher Compacts as the programming modules, the Flasher Hub takes advantage of the extensive list of supported devices and target interfaces, plus the ultra-fast programming speed and reliability of these 'Almost-Anything'-Programmers.

1.1.1 Features of the Flasher Hub

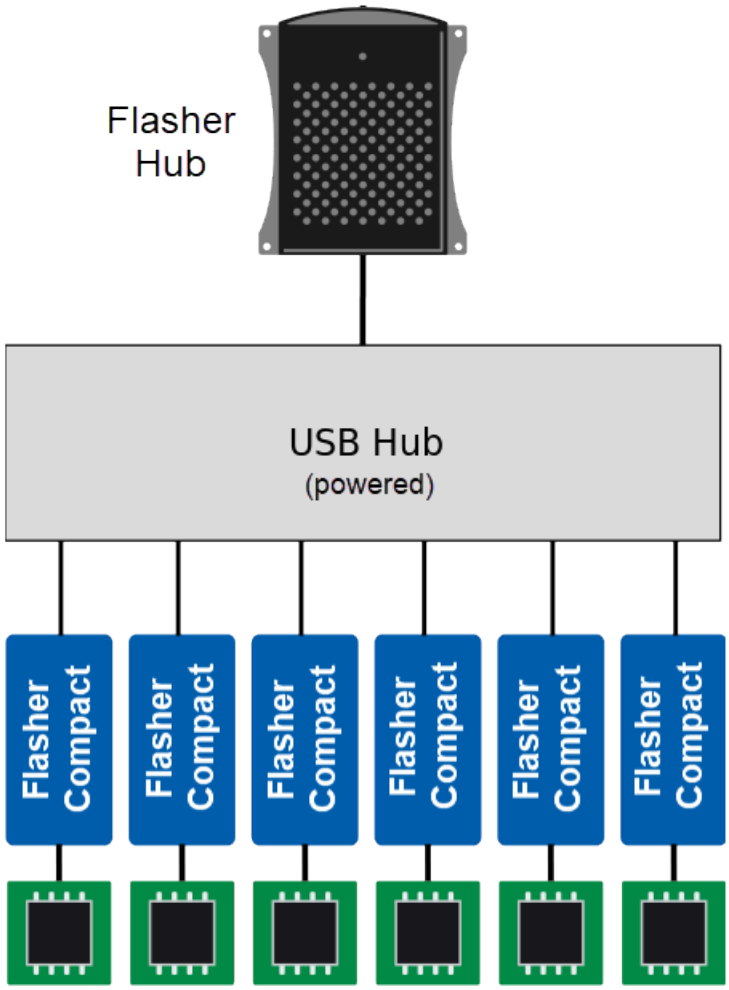
- Stand-alone In-System Programming (ISP) hub
- Scalable solution with up to 24 supported individual parallel programming channels, each with their own programming circuit memory
- Built-in web & FTP servers for easy setup
- Includes Flasher software package

 © 2020-2022 SEGGER Microcontroller GmbH

1.1.2 Working environment

General

The Flasher Hub has been designed to be used in conjunction with automated test equipment (ATE). It is modular and scalable from 1 to 24 individual programming modules, represented by SEGGER Flasher Compacts. The Flasher Compacts are connected to the Flasher Hub via powered USB hubs.



Flasher PC-software (J-Flash / J-Flash SPI)

In order to prepare the firmware / data to be programmed to the targets and download it to the Flasher Compacts, the latest version of the J-Flash / J-Flash SPI software is required. J-Flash / J-Flash SPI are available for Windows, Linux, and macOS, and are part of the Flasher software and documentation package, which can be downloaded from our website: `https://www.segger.com/downloads/flasher/#FlasherSoftwareAndDocumentationPack`

For more information about using J-Flash, please refer to `UM08003_JFlash.pdf` (J-Flash User Guide, available at `https://www.segger.com/downloads/flasher/UM08003`). J-Flash SPI is described in `UM08001_JLink.pdf` (J-Link User Guide, available at `https://www.segger.com/downloads/jlink/UM08001`).

FTP Client

The firmware to be programmed to the targets can also be uploaded to the Flasher Compacts via an FTP connection to the Flasher Hub. For this, an FTP client is required.

 © 2020-2022 SEGGER Microcontroller GmbH

Web Browser

The Flasher Hub's web interface offers a lot of useful status information about the system. It also allows firmware updates to the Flasher Hub and the connected Flasher Compacts, uploading the firmware to be programmed to the targets, configuring the index of the connected Flasher Compact modules, and more. In order to access the Flasher Hub web interface, a web browser is required.

Terminal program

For communication with the Flasher Hub via Telnet, a terminal program like TeraTerm or PuTTY is helpful.

Universal Flash Loader Configurator software

The latest version of the Universal Flash Loader Configurator software can be downloaded from our website at `https://www.segger.com/downloads/flasher`.

This software is only needed if the device you would like to program requires the usage of the Universal Flash Loader.

 © 2020-2022 SEGGER Microcontroller GmbH

1.2 Specifications

1.2.1 Specifications for Flasher Hub

Specifications	
Power supply	USB C-powered, max. 3.0A
USB Interfaces (for connecting powered USB hubs)	2x USB 3.0, 2x USB 2.0
Host Interfaces	FTP, TELNET
Operating Temperature	+0 °C … +50 °C
Storage Temperature	-20 °C … +65 °C
Relative Humidity (non-condensing)	<90% rH
Safety notes	For indoor use only.
Size (without cables or mounting brackets)	60mm x 32mm x 95mm
Weight (without cables)	132g

1.2.1.1 Supported CPU cores

The Flasher Hub itself is CPU-core agnostic. The supported CPU cores are determined by the connected Flasher Compacts. Please see the Flasher Compact User Guide for details.

1.2.1.2 Supported Target interfaces

The Flasher Hub itself is target-interface agnostic. The supported target interfaces are determined by the connected Flasher Compacts. Please see the Flasher Compact User Guide for details.

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 2

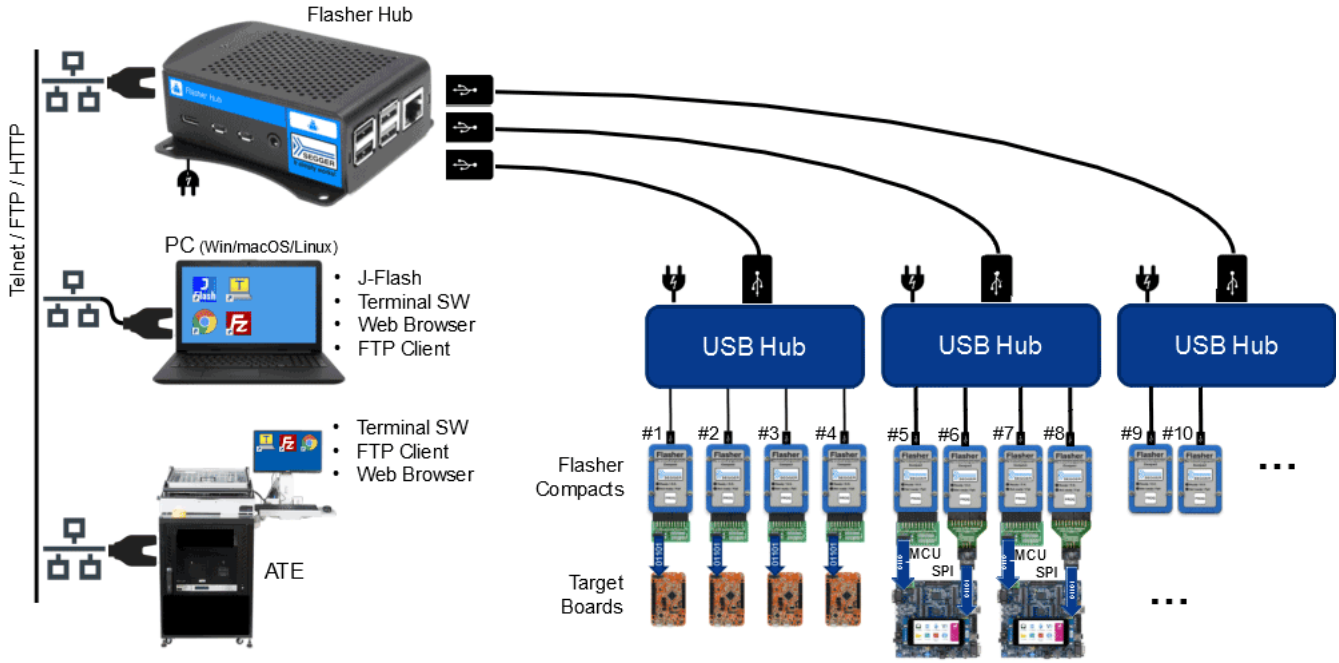
Working with the Flasher Hub

This chapter describes functionality and how to use the Flasher Hub.

© 2020-2022 SEGGER Microcontroller GmbH

2.1 The Flasher Hub

The Flasher Hub makes ultra-fast, high-speed parallel programming (“gang programming”) possible. Via powered USB hubs, up to 24 Flasher Compact modules can be connected to the Flasher Hub. Each Flasher Compact module is connected to a target device.



Via a Telnet connection between the ATE (Automated Test Equipment) and the Flasher Hub, flash programming operations can be triggered and responses from the Flasher Hub can be read and evaluated. This makes the Flasher Hub a perfect fit for small-scale as well as large-scale production environments. The firmware to be programmed into the target devices can be prepared via a computer running Windows, Linux, or macOS and SEGGER’s J-Flash or J-Flash SPI software. The Flasher Compact modules connected to the Flasher Hub can be provisioned with the firmware to be programmed either directly via J-Flash, via an FTP connection to the Flasher Hub, or via the Flasher Hub’s web interface.

2.1.1 Power supply

2.1.1.1 Flasher Hub and Flasher Compact power supply

The Flasher Hub needs to be powered via a 5V USB-C power supply (1.0A or higher; not included).

The Flasher Compacts are powered by the powered USB hubs through which they are connected to the Flasher Hub.

 © 2020-2022 SEGGER Microcontroller GmbH

Current consumption (typical)

Flasher Hub	
5V via USB-C connector	800 mA (Ethernet connected)

Flasher Compact	
5V via powered USB Hub	130 mA (no target power supply)
5V via powered USB Hub	230 mA (target power supply with 100mA)

Note

Don’t connect Flasher Compact modules directly to the Flasher Hub. Always connect Flasher Compact modules to the Flasher Hub via powered USB hubs.

Note

Undefined behavior may occur if the Flasher Hub and/or the Flasher Compacts are not powered sufficiently.

2.1.1.2 Target power supply

The target boards to be programmed can either be powered via their own power supplies, or they can be powered via a 5V / 100mA supply from the Flasher Compacts. Please refer to the Flasher Compact User Guide for details on how to do this.

 © 2020-2022 SEGGER Microcontroller GmbH

2.2 Setting up the IP interface

The Flasher Hub is equipped with an Ethernet interface to communicate with the host system via your network.

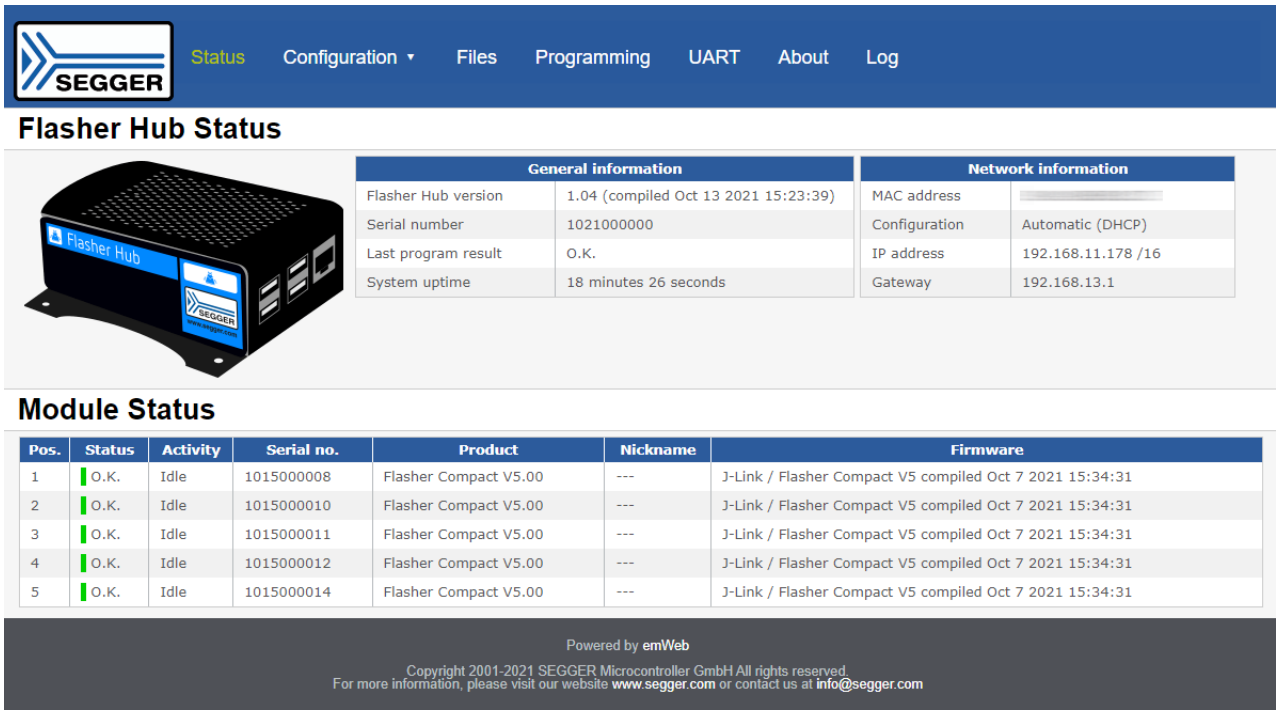
The Flasher Hub also has a built-in web server that provides system status and allows system configuration.

2.2.1 Connecting for the first time

When connecting the Flasher Hub to your network and powering it up for the first time, it attempts to acquire an IP address via DHCP. After the Flasher Hub has finished booting (i.e. after the green LED has stopped flashing), you can connect to the Flasher Hub's web interface by pointing your web browser to

`http://flasherhub-<serial_number>/`
(Make sure to replace <serial_number> with the serial number of your Flasher Hub, which can be found on the housing.)

The "Overview" page served by the Flasher Hub's web server contains the IP address that has been assigned to the Flasher Hub by your network.



Note

Don't connect the Flasher Hub directly to the Ethernet interface of an ATE or host computer. Only connect the Flasher Hub to your network via a router, switch, etc.

 © 2020-2022 SEGGER Microcontroller GmbH

2.3 Operating modes

The Flasher Hub currently can be operated via web interface or via remote-controlled mode. (Handshake mode is planned to be added in the future.)

The Flasher Hub web interface is the recommended way of operating the Flasher Hub for **manual** operation.

In remote-controlled mode, the Flasher Hub receives commands via Telnet connection and reports the results back to the caller. This allows the caller to check whether the desired operation was executed successfully or not. This method of operation was mainly designed for **automation** purposes.

2.3.1 Web interface

The Flasher Hub web interface provides a way to use the Flasher Hub in a production environment and control it manually, i.e. not using automation via scripts or similar. Using the extensive but intuitive graphical user interface of Flasher Hub's web server, programming operations and more can be configured, monitored and controlled with ease.

The web interface is described in more detail in *Web server* on page 46.

2.3.2 Remote-controlled mode

The remote-controlled mode provides a way to integrate the Flasher Hub into a production environment and control it via Ethernet. In this setup, the Flasher Hub provides detailed status information that can be used to verify the success of the programming sequence, optimize the production setup, and identify errors.

In order to use a Flasher Hub based system in remote-controlled mode, the connected Flasher Compact modules need to be configured first. This is described in *Setting up Flasher Hub for remote-controlled mode* on page 28.

 © 2020-2022 SEGGER Microcontroller GmbH

2.4 LED status indicators

The Flasher Hub uses different LEDs as indicated in the following table.

2.4.1 LED indicators

#	Status of LEDs	Meaning
1	GREEN flickering	SD-card access / booting in progress.
2	RED constant	The Flasher Hub is powered.

2.4.2 Flasher Compact LED indicators

LED	Status	Meaning
Ready / O.K.	GREEN short flicker	Module is ready.
Ready / O.K.	GREEN slow blinking	Flashing operation in progress: • Erasing (blinking at 6.25 Hz) • Programming (blinking at 1.67 Hz) • Verifying (blinking at 5 Hz)
Not ready / Fail	RED constant	a) The module is in bootloader mode. b) The most recent operation has failed.

 © 2020-2022 SEGGER Microcontroller GmbH

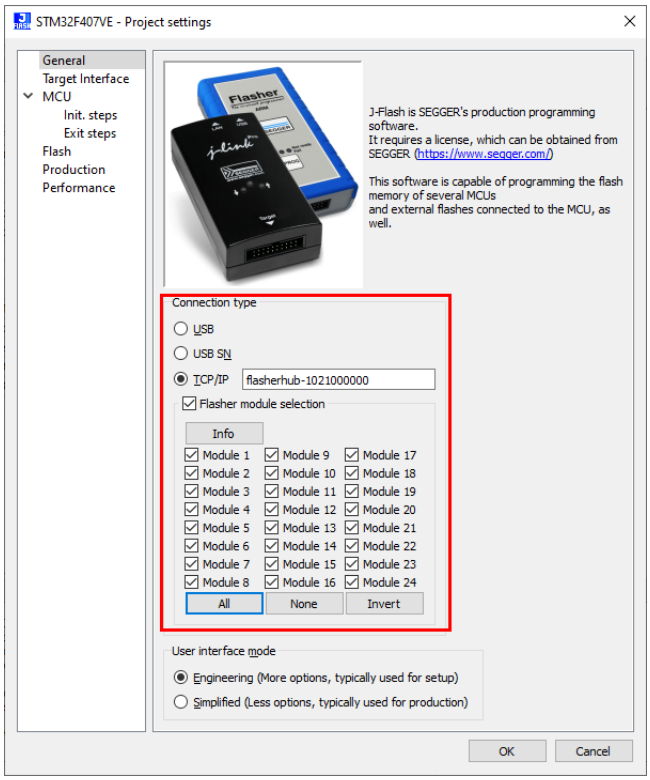
2.5 Storing configuration and programming data files

The configuration and programming data required to program the target devices is stored on the connected Flasher Compact modules. Each Flasher Compact module has approximately 126MB of storage available for data and configuration files.

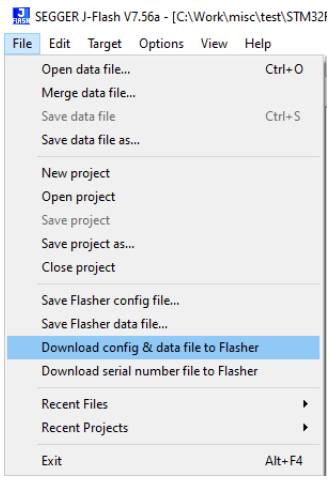
The Flasher Compact modules can be accessed in the following ways:

- via the J-Flash software
- via an FTP client connected to the Flasher Hub's integrated FTP-server
- via the File browser in the Flasher Hub's web interface

When using J-Flash, the connected Flasher Compact modules (Module 1, Module 2, …, Module n) can be selected via tick boxes on the "Options > Project settings > General" screen. Make sure to select TCP/IP as the connection type and enter the Flasher Hub's IP address (or use flasherhub-<serial_number>).



When selecting "File > Download config & data file to Flasher" from the J-Flash menu, the configuration and data file of the current J-Flash project will be downloaded to the selected Flasher Compact modules.

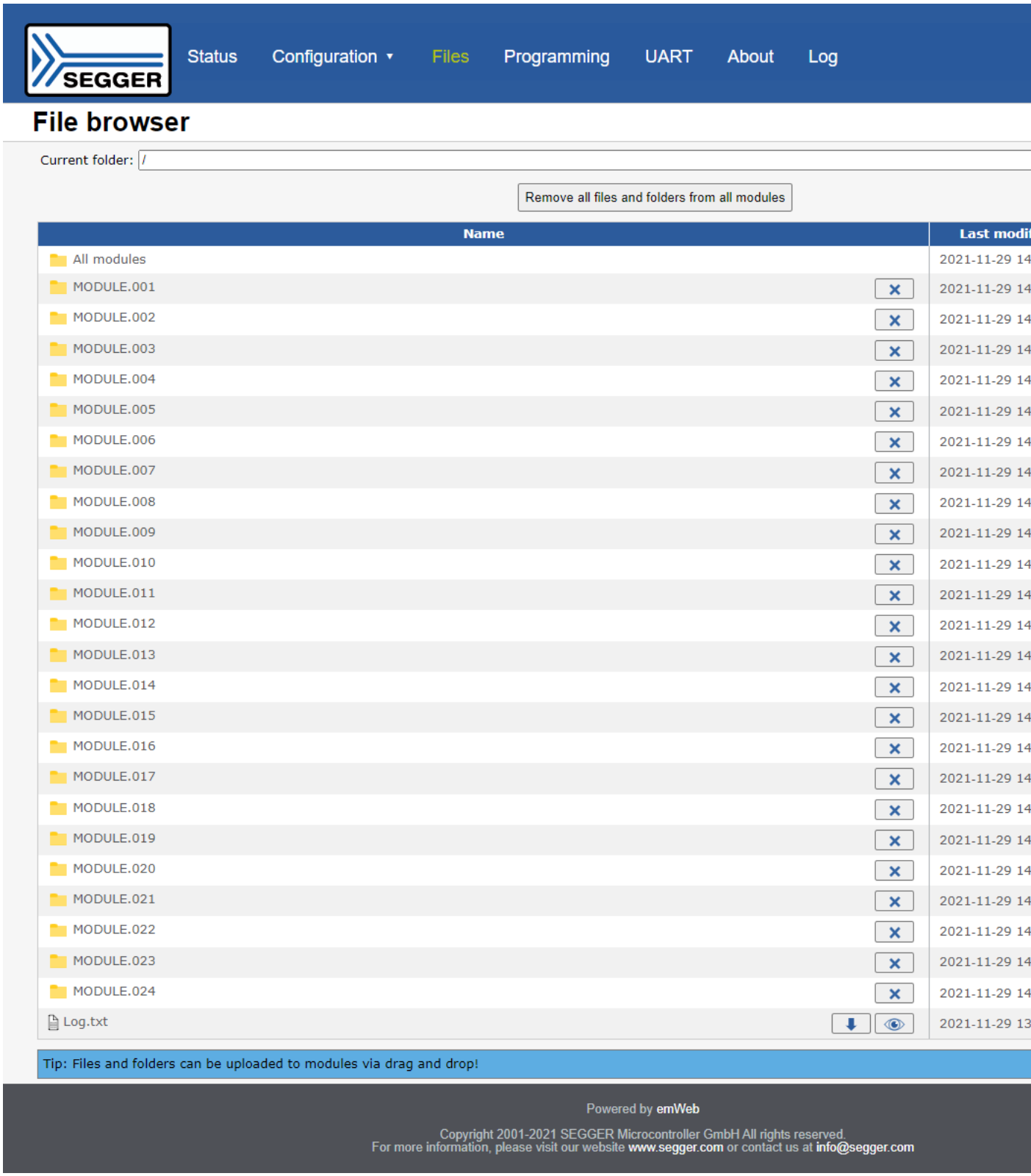


When using an FTP client or the file browser in the Flasher Hub's web interface, the Flasher Compact modules are accessible via subfolders named "MODULE.xxx", with xxx being

 © 2020-2022 SEGGER Microcontroller GmbH

the index of the associated Flasher Compact, e.g. “MODULE.001” for the Flasher Compact module with index 1.

The index of the connected Flasher Compact modules can be changed via the Flasher Hub web interface, on the “Configuration > Flasher” page



2.6 UART to TCP transceiver

The Flasher Hub features an UART to TCP transceiver: Each module is accessible via a connection to a module-specific TCP port.

Module	TCP port
#1	41
#2	42
#3	43
#4	44
#5	45
#6	46
#7	47
#8	48
#9	49
#10	50
…	..
#24	64

If the transceiver is enabled, incoming data on pin 17 on the debug interface is sent to the TCP connection, and incoming data on the TCP connection is sent to pin 5 on the debug interface.

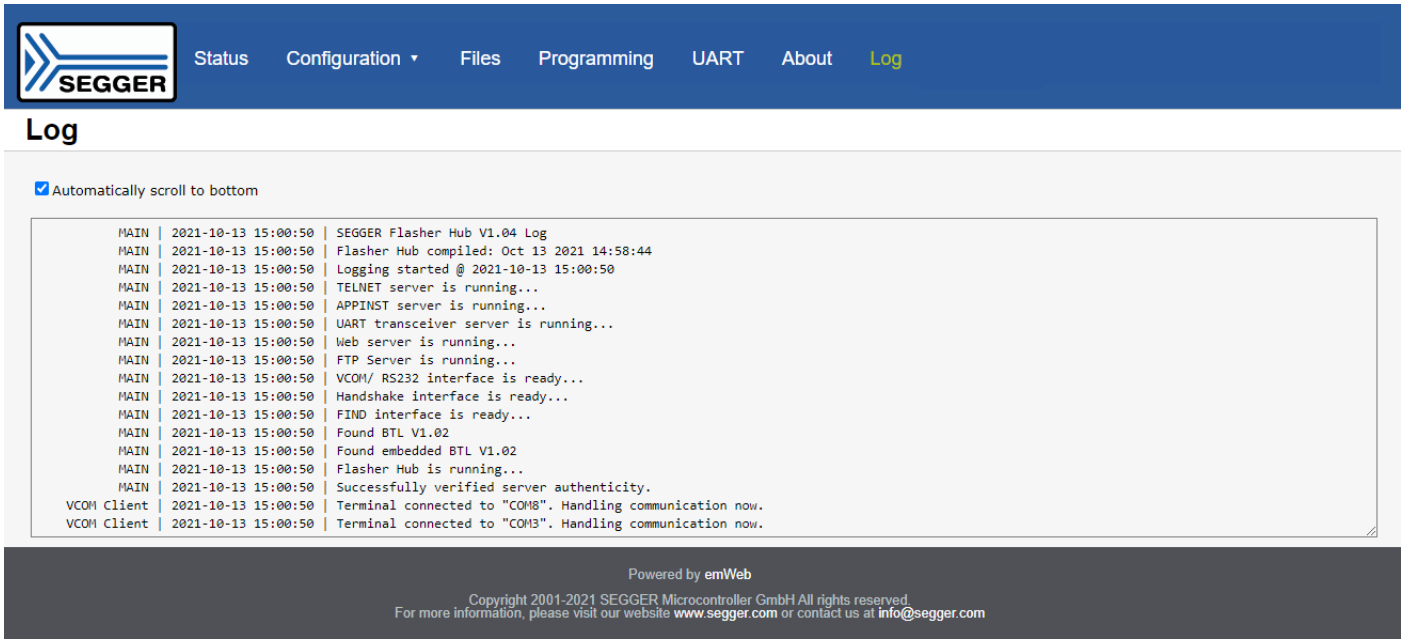
The transceiver can be enabled for one or more modules either via the ASCII Terminal command #TERMINAL or via the web interface "UART" page. For more information, please refer to *the "#TERMINAL" command* on page 83 and *the web interface "UART" page* on page 53.

> **Note**
>
> The transceiver cannot be enabled while the module is programming. Also programming, erasing etc. can not be started if the transceiver mode is active.

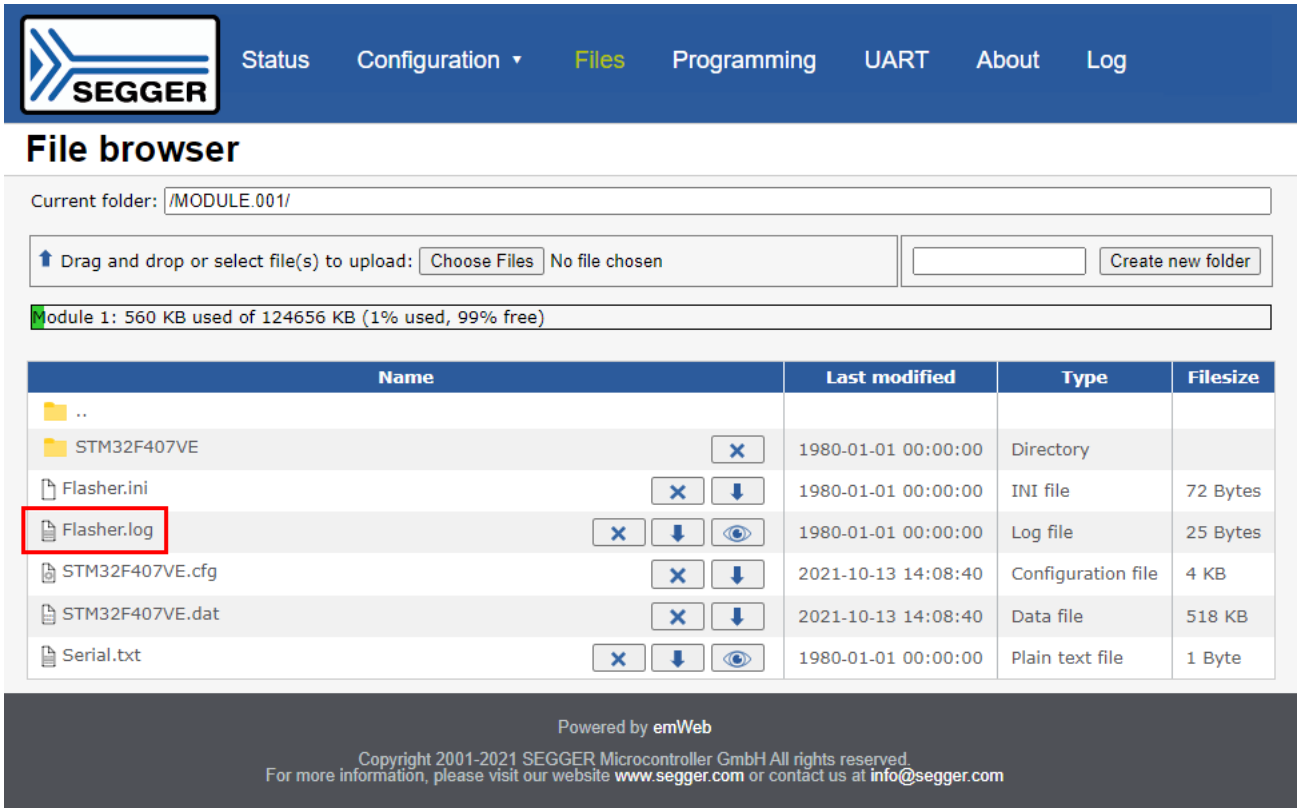
 © 2020-2022 SEGGER Microcontroller GmbH

2.7 Log files

The Flasher Hub keeps a log file if logging is enabled via the web interface on the "Configuration > Server" page. The log itself can be accessed on the "Log" page.



In addition, each Flasher Compact keeps its own log file. These log files are stored in the module folder of the corresponding Flasher Compact and can be downloaded via FTP or the file browser in the Flasher Hub's web interface. For more information on the Flasher Compact log file, please refer to the Flasher User Guide (UM08022).



2.8 Updating the Flasher Hub Firmware

Firmware updates for the Flasher Hub are provided on the Flasher download page `https://www.segger.com/downloads/flasher/`.

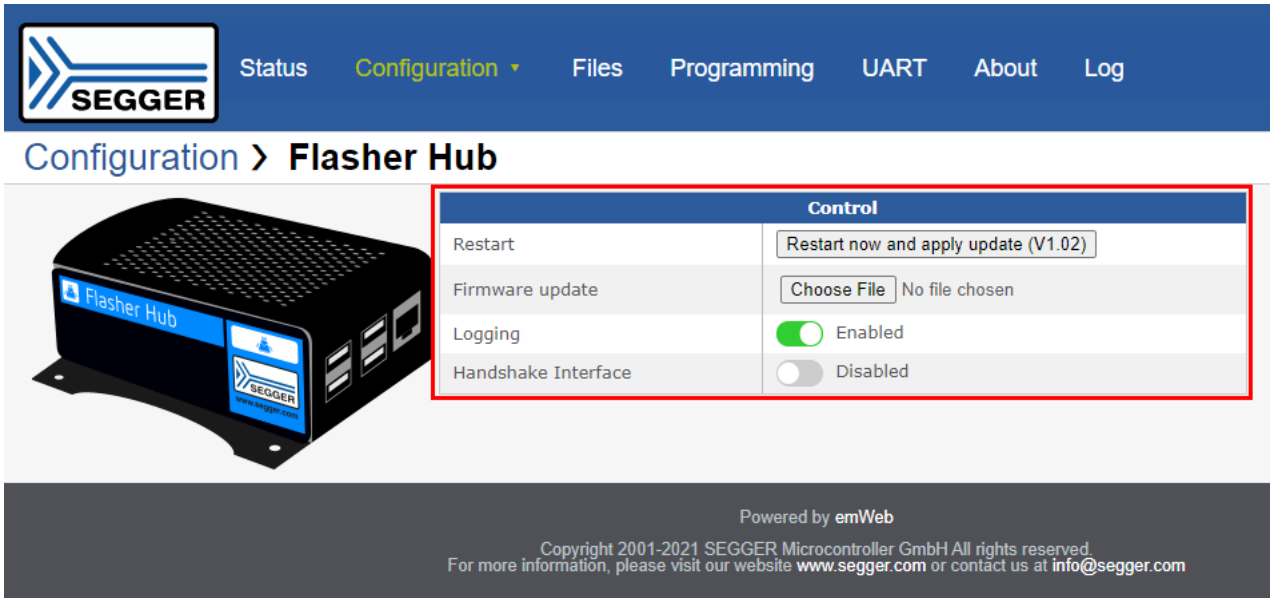
> **Note**
>
> Please note the Flasher Hub cannot be updated via J-Link Commander or J-Link Configurator.

The Flasher Hub update package provides the update file `Firmware.update`. This file needs to be uploaded via FTP to the root directory of the Flasher Hub.

After the upload process, the button in the Control section of the Flasher Hub web interface "Overview" page will say "Restart now and apply update". Click the button and wait for the Flasher Hub to restart.

After the update, a new server version is displayed in the General information section of the "Overview" page.

> **Note**
>
> It is recommended to use the most recent version of the Flasher Hub firmware as newer versions may contain various improvements as well as bugfixes.



2.9 Newline encoding

In general, for all patch files, init files etc., the Flasher Compact modules support both newline encodings:

- Windows: \r\n
- Unix/Mac: \n

All parser functionality etc. are written to be independent from the host operating system.

 © 2020-2022 SEGGER Microcontroller GmbH

2.10 Fail-safe mode

For more information on Flasher Hub’s fail-safe mode, please refer to:
SEGGER Wiki: Flasher Hub - Fail-safe mode

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 3

Setting up a project for the Flasher Hub

In order to set up the Flasher Hub for parallel programming, the connected Flasher Compact modules need to be configured once using either the J-Flash software or using the Universal Flash Loader Configurator software. Both programs are part of SEGGER's Flasher software and documentation package, available for download free of cost at: `https://www.segger.com/downloads/flasher/#FlasherSoftwareAndDocumentationPack`.

© 2020-2022 SEGGER Microcontroller GmbH

3.1 Using J-Flash

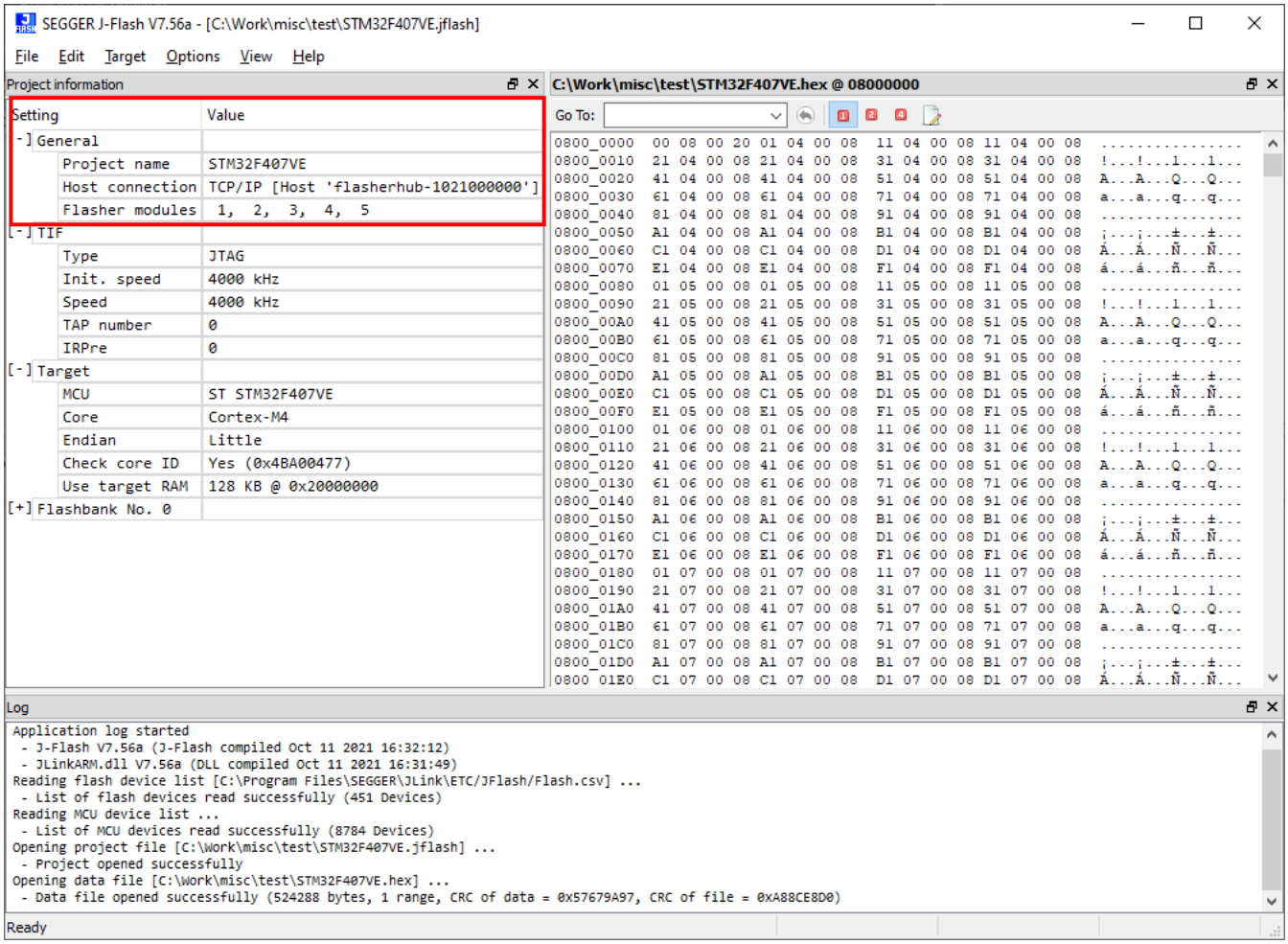
J-Flash is SEGGER's desktop software for production prorgramming and it is available for Windows, macOS and Linux. It can be used to create Flasher configuration and data files for ARM-based target devices. For more information about J-Flash, please refer to the *J-Flash User Guide*.
In order to set up the Flasher Hub for parallel programming, the connected Flasher Compact modules need to be configured once using J-Flash.

After starting J-Flash, open the appropriate J-Flash project for the target device the Flasher Compacts shall be configured for, by selecting **File** -> **Open Project**. If J-Flash does not come with an appropriate sample project for the desired hardware, a new project needs to be created by selecting **File** -> **New Project**.

After the appropriate project has been opened / created, the data file which shall be programmed needs to be loaded, by selecting **File** -> **Open**. Next, click on **Options** -> **Project settings** -> **General** and select **Connection type** to be **TCP/IP**. As the TCP/IP address, enter **flasherhub-<serial_number>**. You can find the serial number of your Flasher Hub on the product housing.

Click **Flasher Hub module selection** and select the Flasher Compact module numbers you would like to configure. Then click **OK**.

After this, J-Flash should look similar to the screenshot below.



 © 2020-2022 SEGGER Microcontroller GmbH

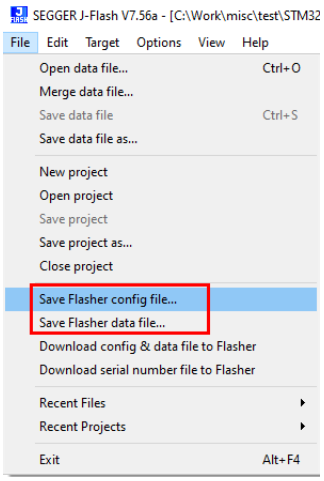
Next, you can transfer the necessary files to the Flasher Compacts using **File** -> **Download config & data file to Flasher**. Check the J-Flash Log window to make sure the file transfer was successful. For further details please check the J-Flash manual.

> **Note**
>
> If J-Flash has trouble connecting to **flasherhub-<serial_number>**, use the actual IP address of your Flasher Hub. You can find it on the "Overview" page of the Flasher Hub's web interface.

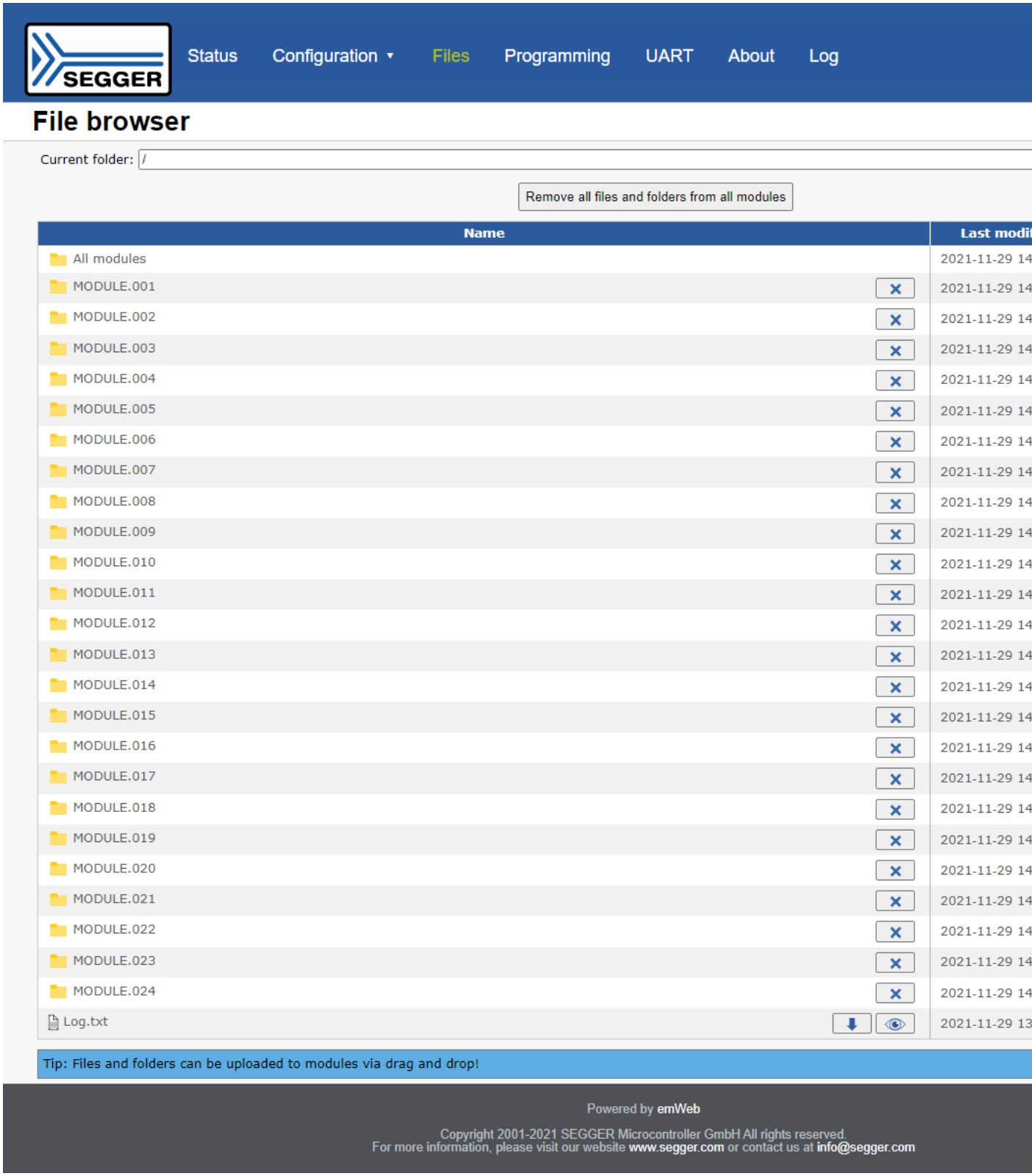
From now on, the Flasher Hub and the attached Flasher Compacts can be used in stand-alone mode (without host PC interaction) for stand-alone programming.

As an alternative to the process described above, you may also save the files to disk and upload them to the Flasher Compacts **via the web inteface**. To do this, follow the instructions below:

1. Save the configuration (project) file to disk by selecting **File** -> **Save Flasher Config File**
2. Save the data (program data) file by selecting **File** -> **Save Flasher Data File**
4. Connect to the Flasher Hub Web interface via a browser.
5. Navigate to the file browser in the "Files" tab
6. Now the configuration file and the data file can be uploaded to the Flasher Hub module folder(s) (MODULE.xxx), i.e. to the connected Flasher Compacts.
   Choosing the **All modules** folder will upload the files to all Flasher Compacts connected to the Flasher Hub. Note: The file(s) can be uploaded by dragging and dropping inside the browser window.



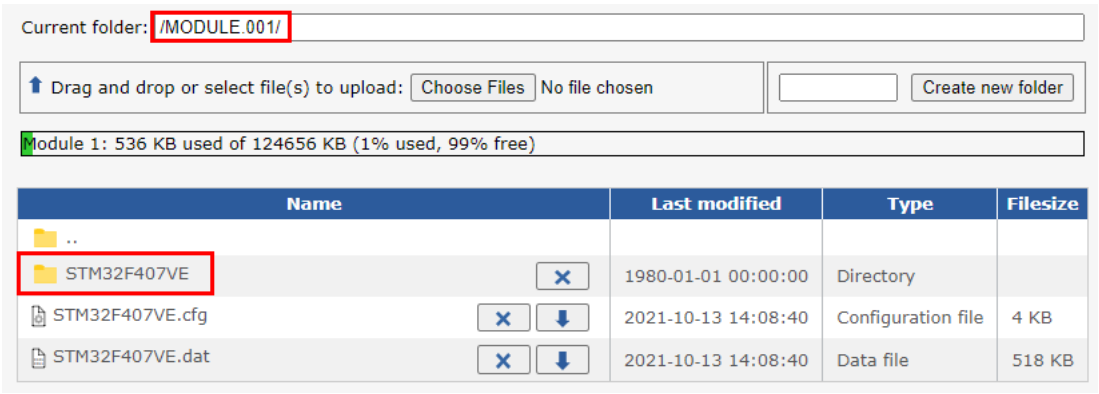
 © 2020-2022 SEGGER Microcontroller GmbH

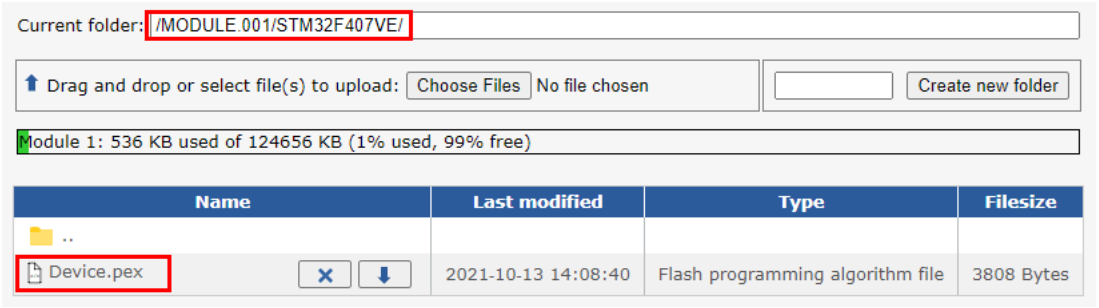


A third option is to upload the configuration and data files to the Flasher Compact modules via FTP as described here: *FTP server* on page 43.

If J-Flash also generates a .pex file (which is a device-specific flash programming algorithm that is required for certain target devices), this also needs to be uploaded to the module folder(s) for programming to be successful. Please note that it must be located in a sub-folder with the same name as your project, e.g. if your project is named **MyProject** with the project files **MyProject.cfg** and **MyProject.dat**, then the subfolder must be named **MyProject**.

 © 2020-2022 SEGGER Microcontroller GmbH





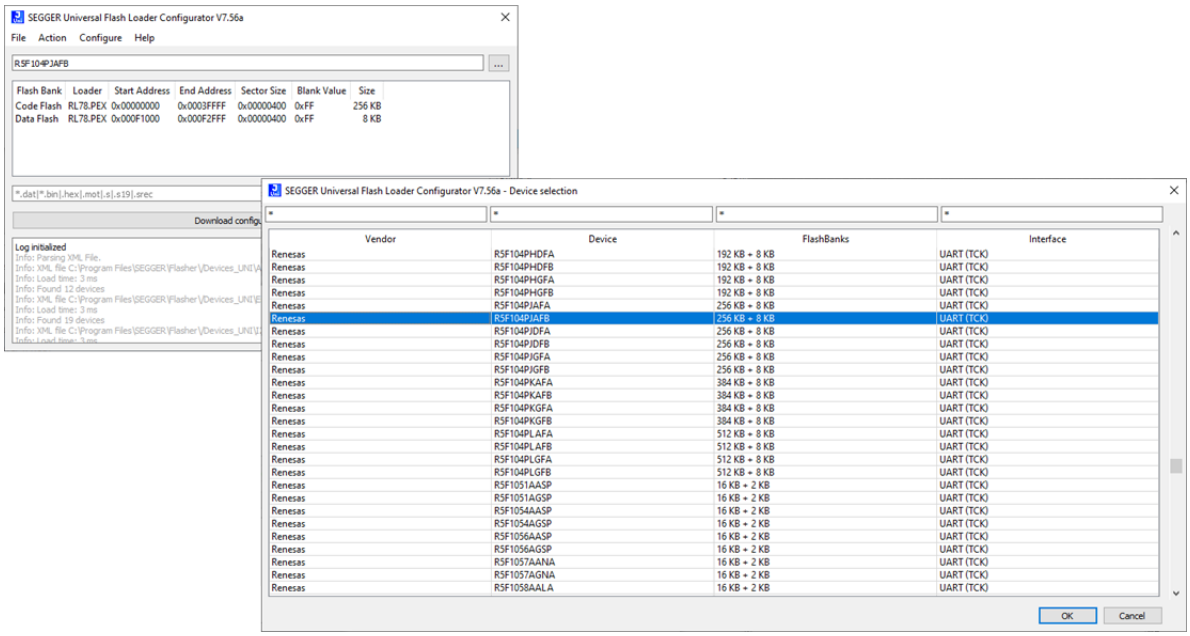
 © 2020-2022 SEGGER Microcontroller GmbH

3.2 Using Universal Flash Loader

The Universal Flash Loader mode can be used to program non ARM-based targets. While configurations generated with J-Flash rely on using the debug interface of the device, configurations using the Universal Flash Loader make use of device or vendor specific programming interfaces and protocols and therefore it is independent of the CPU core.

A Universal Flash Loader configuration can be created using SEGGER's Universal Flash Loader Configurator, available for Windows, macOS and Linux.

For details, please see the Universal Flash Loader User Guide (UM08037) as well as the Flasher User Guide (UM08022).



 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 4

Serial number handling

This chapter describes how to deal with serial numbers in a Flasher Hub based system.

© 2020-2022 SEGGER Microcontroller GmbH

4.1 Serial number programming

A Flasher Hub based system supports programming of serial numbers. In order to use the serial number programming feature, the J-Flash project to be used as well as some files on the Flasher Compacts connected to the Flasher Hub (depending on the configuration) need to be configured first.

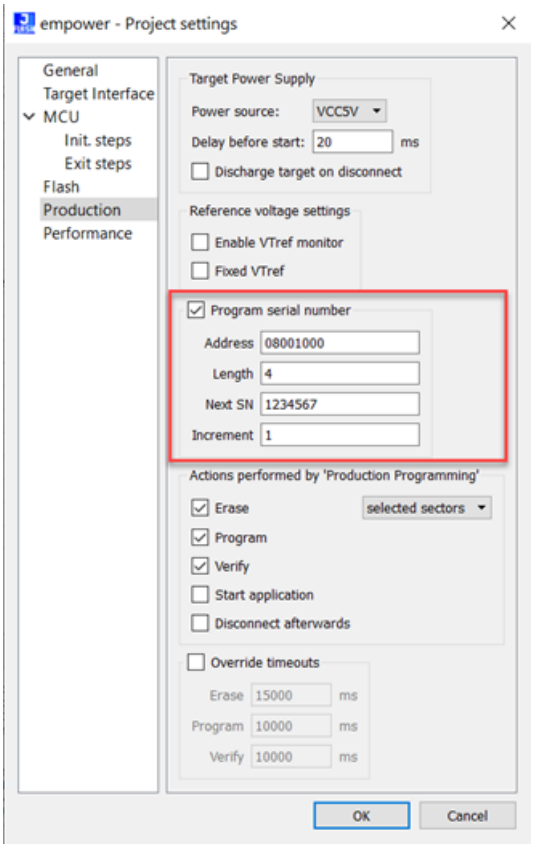
In general, a Flasher Hub based system supports two ways of programming a serial number into the target:

1. Programming continuous serial numbers. Serial number is 1-4 bytes in size. Start serial number, increment, serial number size and address is configured in the J-Flash project.
2. Programming custom serial numbers from a serial number list file. Start line into serial number list file to get next serial number bytes, line increment, serial number size and address is configured in the J-Flash project. Serial number list file needs to be specified and created by user.

Some generic information on how to setup the Flasher Hub based system & the J-Flash project for serial number programming is provided below.

4.1.1 Serial number settings

In order to enable the programming of serial numbers in stand-alone mode, the J-Flash project has to be configured to enable programming a serial number at a specific address. This is done by enabling the **Program serial number** option as shown in the screenshot and table below:



 © 2020-2022 SEGGER Microcontroller GmbH

Setting	Meaning
`Address`	The address the serial number should be programmed at.
`Len`	The length of the serial number (in bytes) that should be programmed. If no serial number list file is given, J-Flash allows to use a 1-4 byte serial number. In case 8 is selected as length, the serial number and its complement are programmed at the given address. In case a serial number list file is given, the Flasher Hub will take the serial number bytes from the list file. If a serial number in the list file does not define all bytes of `Len` , the remaining bytes are filled with 0s. No complements etc. are added to the serial number.
`Next SN`	In case no serial number list file is given, `Next SN` is the next serial number which should be programmed. The serial number is always stored in flash memory in little endian format. In case a serial number list file is given, `Next SN` describes the line of the serial number list file where to read the next serial number bytes from. The Flasher Hub starts counting at line 0, so in order to start serial number programming with the first line of the `SNList.txt`, `Next SN` needs to be set to 0.
`Increment`	Specifies by how much `Next SN` is incremented.

4.1.2 Continuous Serial numbers

The Flasher Hub can generate serial numbers. Therefore the project can be configured to use the serial number feature (see on page 35). The Flasher Hub will use the first serial number for the first programmed device. Then the increment is added to the serial number and this is used for the next programming sequence. The next serial number is stored in the SERIAL.TXT file on each Flasher Compact module. Therefore, the serial number is also power-cycle safe. If the file is missing at start up time, the number 0 is used for the first target.

To avoid duplication of serial numbers with the Flasher Hub using more than one Flasher Compact module, there are two options:

- Use an increment of the number of connected Flasher Compact modules; e.g. if you have 5 Flasher Compacts connected, use an increment of 5 as well as 5 different SERIAL.TXT files at the beginning of production.
- Use different serial number areas; e.g. if you have 5 Flasher Compacts connected, use an increment of 1 as well as 5 different SERIAL.TXT files at the beginning of production. For the first module, use the range from 1 to 1000, for the second 1001 to 2000, and so on.

The SERIAL.TXT file contains the value `Next SN` in ASCII notation, e.g. `1234` if the next serial number is 1234.

> **Note**
>
> The serial number in SERIAL.TXT will also be incremented if serial number programming is disabled, to make sure that for the Flasher Hub logfile there is a reference for which programming cycle passed and which did not. As long as serial number programming has not been enabled in the J-Flash project, the Flasher Hub does not merge any serial number data into the image data to be programmed.

4.1.3 Serial number list file

In order to program custom serial numbers which cannot be covered by the standard serial number scheme provided by J-Flash (e.g. when programming non-continuous serial num-

 © 2020-2022 SEGGER Microcontroller GmbH

bers or having gaps between the serial numbers), a so called serial number list file needs to be created by the user.

The SERIAL.TXT file needs to contain the values for the serial numbers in ASCII notation. Each line in the file must contain one serial number.

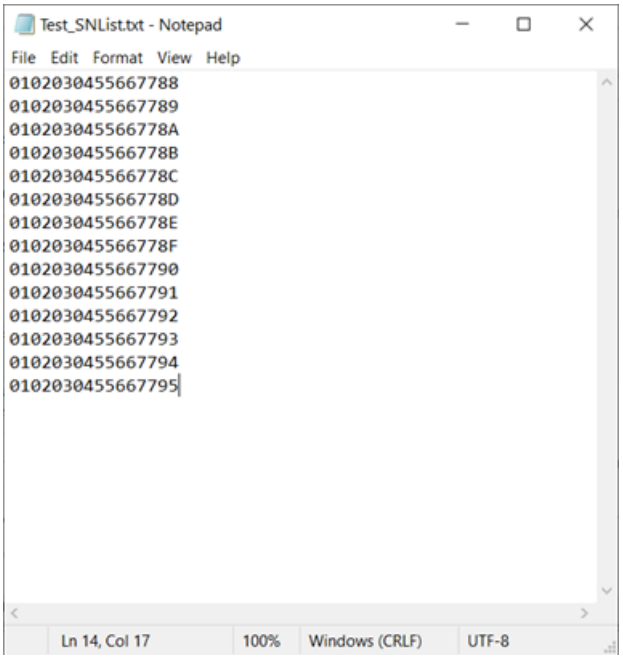
Example

An 8-byte serial number should be programmed at address `0x08000000`.

It should be programmed as follows in the memory:

```
0x08000000: 0x01 0x02 0x03 0x04 0x55 0x66 0x77 0x88
```

The associated serial number list in the file should look as follows:



The number of bytes to read per line is configured via the `Len` option in J-Flash. For more information, please refer to *Serial number settings* on page 35.

Which line Flasher will read at the next programming cycle is configured via the `Next SN` option in J-Flash. For more information, please refer to *Serial number settings* on page 35. In this case, `Next SN` needs to be set to 0, since programming should start with the serial number bytes defined in the first line of the file.

> **Note**
>
> If the number of bytes specified in a line of the serial number list file is less than the serial number length defined in the J-Flash project, the remaining bytes are filled with 0s by the Flasher Hub.

> **Note**
>
> If the number of bytes specified in a line of the serial number list file is greater than the serial number length defined in the J-Flash project, the remaining bytes will be ignored by the Flasher Hub.

4.1.4 Programming process

The Flasher Compact modules connected to the Flasher Hub will increment the serial number in SERIAL.TXT by the value defined in `Increment` after each successful programming cycle.

 © 2020-2022 SEGGER Microcontroller GmbH

For each programming cycle, the FLASHER.LOG file on the connected Flasher Compact modules is updated and contains the value from SERIAL.TXT that has been used for the programming cycle.

> **Note**
>
> The serial number in SERIAL.TXT will also be incremented if serial number programming is disabled, to make sure that for the Flasher Compact logfile there is a reference for which programming cycle passed and which did not. As long as serial number programming has not been enabled in the J-Flash project, the Flasher Compacts do not merge any serial number data into the image data to be programmed.

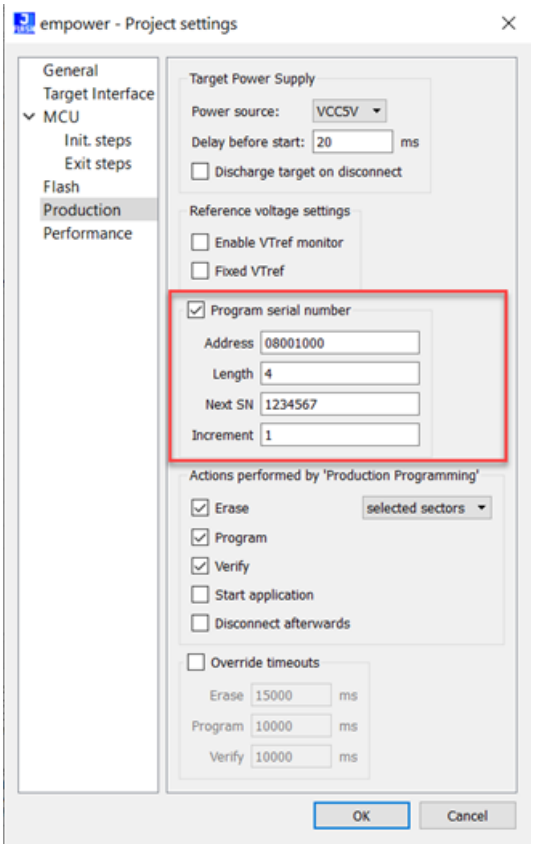
4.1.5 Example setup

Below, a small example is given on how to set up the Flasher Compacts for serial number programming. In the following example, 4-byte serial numbers starting at 1234567 (`0x12D687`) shall be programmed at address `0x08001000`.

Defining serial number address, length and start value

In the J-Flash project, the following needs to be defined:

- `Address` is `0x08001000`
- `Next SN` is 1234567
- `Increment` is 1
- `Len` is 4 (bytes)



 © 2020-2022 SEGGER Microcontroller GmbH

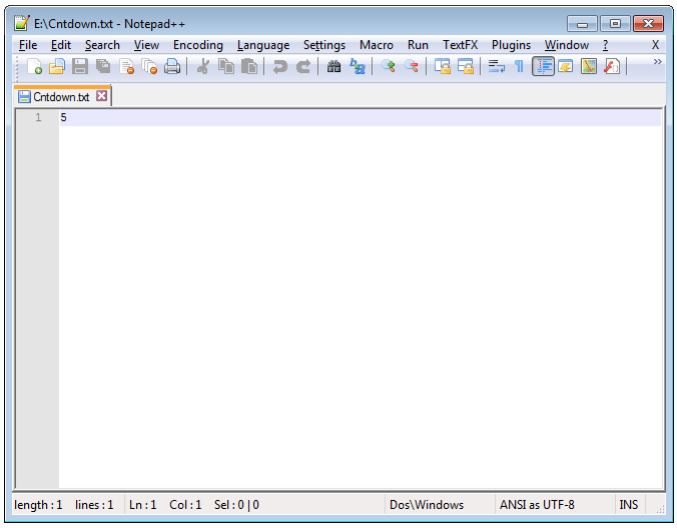
Downloading configuration, data, and serial number to the Flasher Compacts

After setting up the rest of the configuration (Target interface etc.) and selecting an appropriate data file, the configuration, data, and serial number file needs to be transferred to the associated Flasher Compact via the Flasher Hub, either using J-Flash, or using an FTP client, or using the file browser in the Flasher Hub’s web interface.

 © 2020-2022 SEGGER Microcontroller GmbH

4.2 Limiting the number of programming cycles

The Flasher Hub provides a mechanism to limit the number of programming cycles that can be performed in stand-alone mode with the configuration that is stored on the Flasher Compacts. To make use of this feature, a file called `Cntdown.txt` needs to be placed onto the Flasher Compacts module folder. This file simply contains a decimal number (32-bit unsigned integer) that describes how many programming cycles can be performed with the current setup.



This feature especially makes sense when used in combination with authorized flashing. For more information about authorized flashing, please refer to the Flasher User Guide.

> **Note**
>
> The number in `Cntdown.txt` is only updated on a successful programming cycle. Programming cycles that failed do not affect `Cntdown.txt`.

4.2.1 Modified fail/error LED indicator behavior

In case a `Cntdown.txt` is found at boot time, the fail/error LED of the Flasher Compact behaves differently. If the number of programming cycles left is 10 or below, the following will happen:

- The red error/fail LED will be lit for 1 second
- After this, it will blink/toggle x times @ 5 Hz, indicating the number of programming cycles left (blinking 5 times for 5 cycles left, etc.)

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 5

Patch data file

This chapter describes how a Flasher Hub based system can patch data files.

© 2020-2022 SEGGER Microcontroller GmbH

5.1 Patch file support

The Flasher Compact modules connected to the Flasher Hub support patch files, which allows to patch the content of the data to be programmed. Before starting the programming process in stand-alone mode, the Flasher Compact module will look for a file named `Patches.txt` being present. This file includes the patches. If this file is present, the number in `Serial.txt` describes the line number of the `Patches.txt` file that will be used for the current cycle (line counting starts at 0).

Each line in `Patches.txt` can hold up to 4 patches, where each patch can be up to 32 bytes in length.

Syntax

Each line begins with `<NumPatches>` followed by each patch `<Addr>,<NumBytes>:<Data>` in sequence and separated by commas. So the syntax for `<NumPatches>` = 4 would be as follows:

`<NumPatches>,<Addr>,<NumBytes>:<Data>,<Addr>,<NumBytes>:<Data>,<Addr>,<NumBytes>:<Data>,<Addr>,<NumBytes>:<Data>\r\n`

Find below a table which describes each parameter.

Parameter	Description
`<NumPatches>`	Describes the number of patches in this patch line. Max. value is 4.
`<Addr>`	Describes the address to be patched. Value is expected in hex.
`<NumBytes>`	Number of bytes for the current patch. Max. value is 20h (32 in decimal). Value is expected in hex.
`<Data>`	Describes the data to be patched. <Data> is always expected as 2 hexadecimal characters per byte.

> **Note**
>
> All values are expected in hexadecimal format (hex).
> The <Data> section is always preceded by ":", not ",".

Example

Please find below an example sequence which clarifies the usage of patch files.

`Patches.txt`, which is located on the Flasher Compact(s), contains the following line:

3,100025,3:AABBCC,100063,2:DDEE,100078,1:FF

`Serial.txt` contains a "0", which forces the Flasher Compact to use line 0 from `Patches.txt`.

After starting the programming cycle, the following data will be patched:

Addr `0x100025`: 3 byte `0xAA 0xBB 0xCC`
Addr `0x100063`: 2 byte `0xDD 0xEE`
Addr `0x100078`: 1 byte `0xFF`

Single patch via Telnet

Alternatively, you can start a programming cycle with patch data that is only valid for this one cycle (no need for a `Patches.txt` file):

Send the `#AUTO PATCH <module> <NumPatches>,<Addr>,<NumBytes>:<Data>`

command via the Flasher Hub Telnet interface. The parameters have the same function as described in the table above.

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 6

FTP server

This chapter describes the FTP server features.

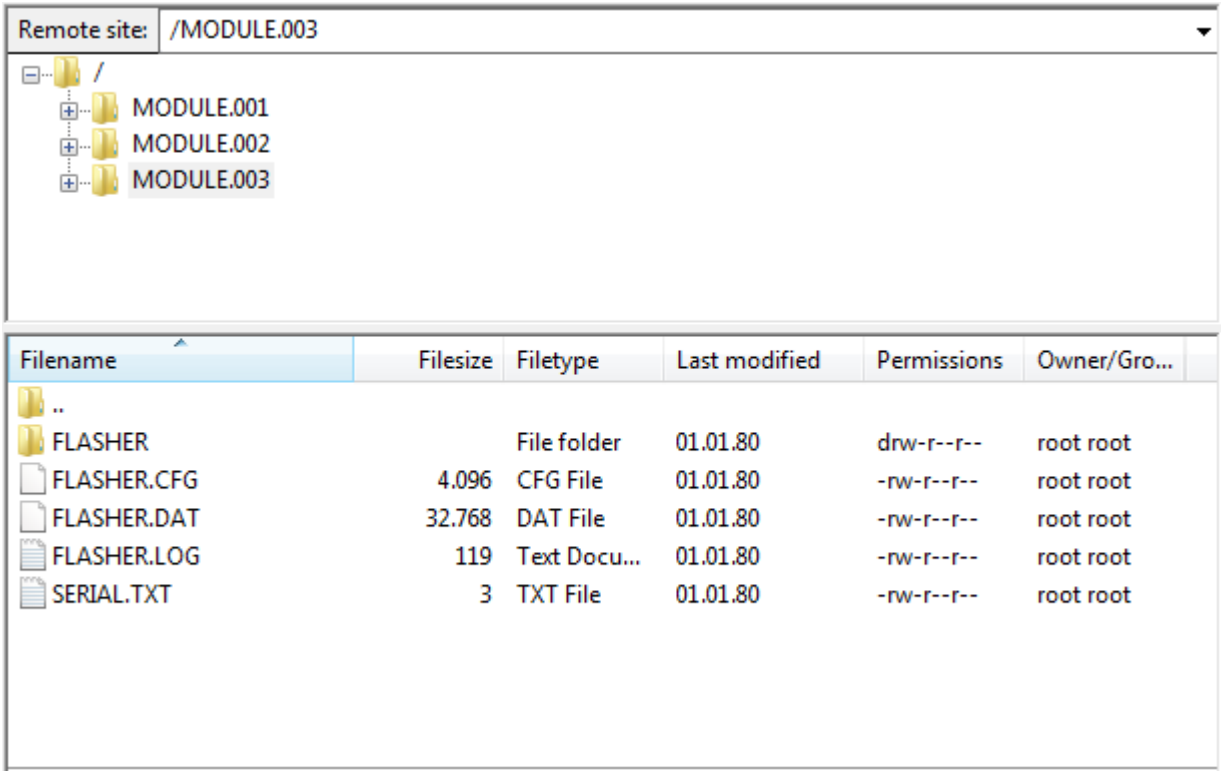
© 2020-2022 SEGGER Microcontroller GmbH

6.1 FTP server connection

The FTP server on the Flasher Hub provides easy access to the files on the connected Flasher Compact modules. The server supports a maximum of 2 simultaneous connections and works with all common FTP clients.

The FTP server root directory is a virtual directory and cannot be written to. It contains a subdirectory for each module, as well as an "All Modules" directory.

Note that the assignment of Flasher Compact serial numbers to the Module numbers (MODULE.xxx) can be changed via the Flasher Hub web interface.



The FTP server allows you to upload or download the target configuration and data files.

Files and folders uploaded to the "All Modules" directory automatically get copied to all connected Flasher Compacts. Note that the "All Modules" directory itself does not retain any files or folders and will always appear empty.

Please also note that you may have to perform a "Refresh" operation on the Module folders before the FTP client will display the 'true' content of these folders. This is because many FTP clients tend to cache the folder contents and don't necessarily refresh the display automatically, even if the folder contents has changed.

The Flasher Compacts create log files for executed operations. These files can be found in the modules folders and downloaded from there.

The IP setup is described here: *Setting up the IP interface* on page 18.

 © 2020-2022 SEGGER Microcontroller GmbH

6.1.1 Access data

Anonymous access to the FTP server is limited to read-only access to the file system. For write access, special login credentials have to be used:

Setting	Value
Host name	ftp://flasherhub-<serial_number>
Username	`admin`
Password	`1234`
Port	21 (Default FTP port)

> **Note**
>
> The access data for read/write access can not be modified and is intended to be used only as a convenience feature to avoid unintended modification of the Flasher's file system. It is not meant as a security feature.

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 7

Web server

This chapter describes the web server features.

© 2020-2022 SEGGER Microcontroller GmbH

7.1 Web server features

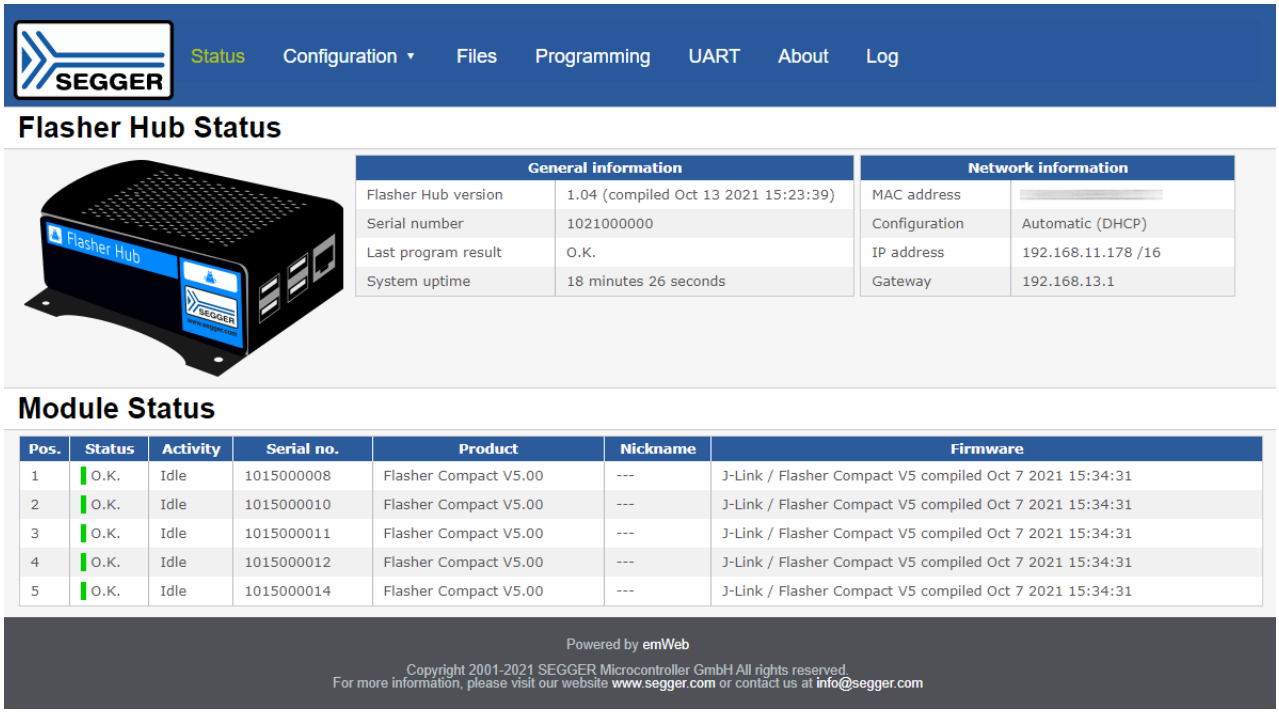
The Flasher Hub comes with a built-in web server, which provides a web interface for information and network configuration.

The web interface provides information about the Flasher Hub server version, serial number, and configuration. The Flasher Hub's web interface furthermore allows monitoring and configuration of the individual modules as well as the files stored on them.
The IP setup is described here: *Setting up the IP interface* on page 18.

7.1.1 Status page

The "Status" page is the landing page and shows some parameters and other information of the Flasher Hub. It also provides an overview of the connected modules.

> **Note**
>
> In order to update the firmware of modules in this tab, the current web session needs to be in Administrator mode. For more information please refer to *Web interface configuration* on page 50.

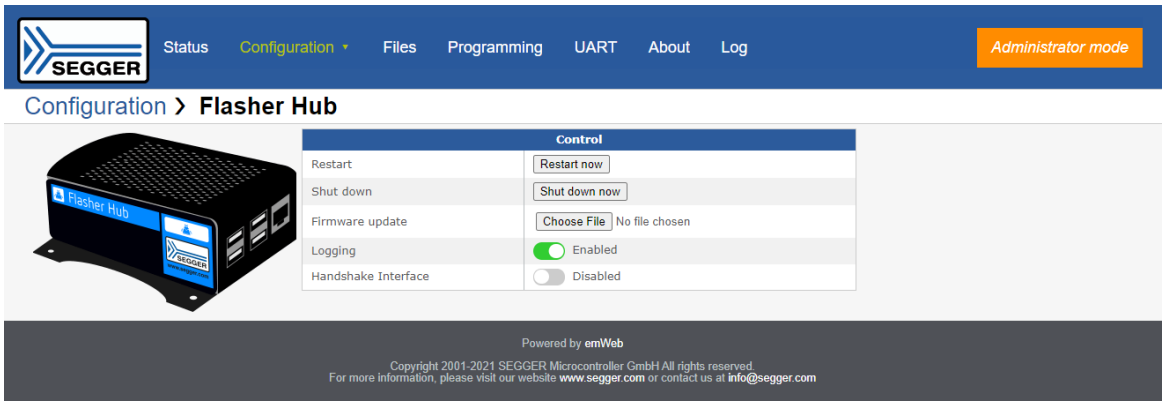


 © 2020-2022 SEGGER Microcontroller GmbH

7.1.2 Flasher Hub configuration page

The "Configuration > Flasher Hub" page allows restarting or shutting down the Flasher Hub as well as uploading firmware updates for it. It also comes with options for toggling Flasher Hub's logging and handshake interface (for future feature expansion use only - leave disabled for now).

> **Note**
>
> In order to restart, shutdown or toggle the settings in this tab, the current web session needs to be in Administrator mode. For more information please refer to *Web interface configuration* on page 50.



 © 2020-2022 SEGGER Microcontroller GmbH

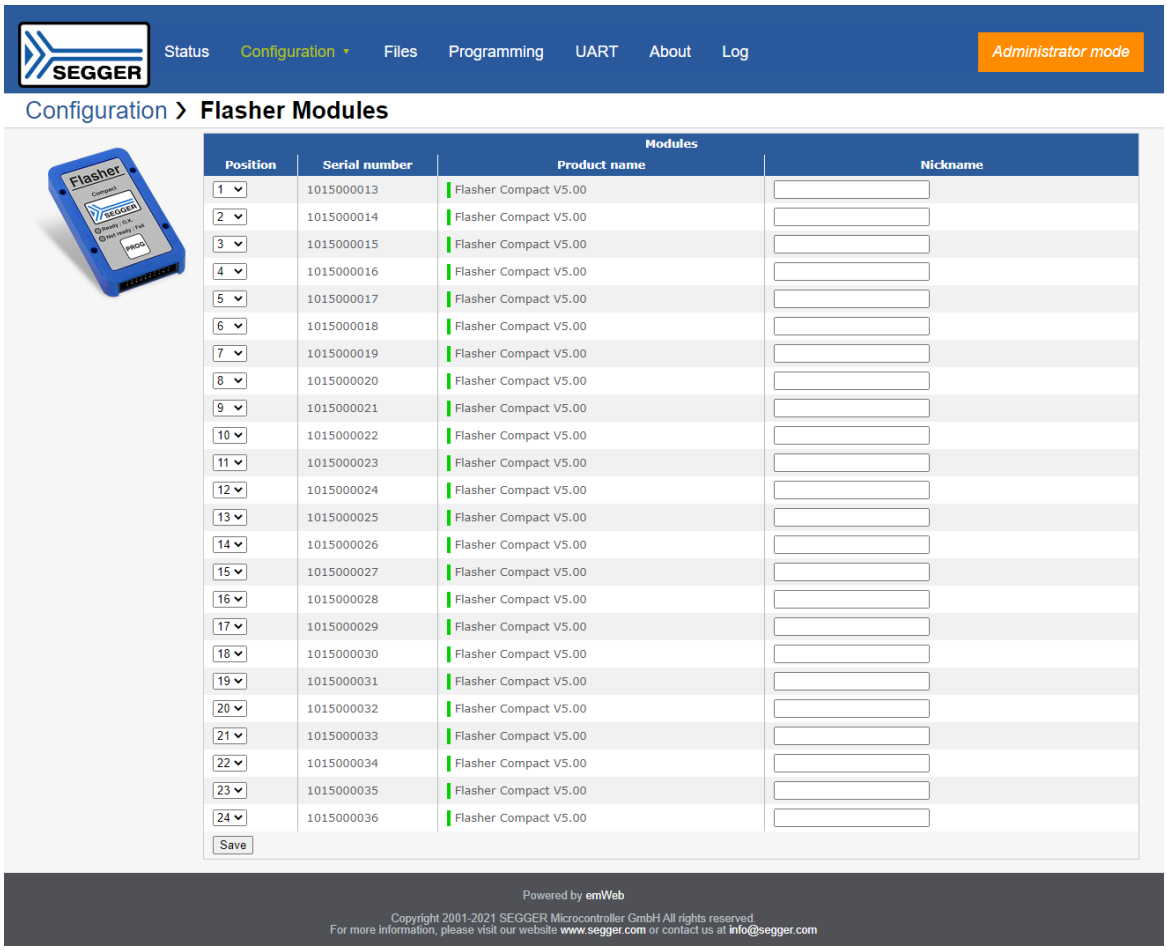
7.1.3 Flasher modules configuration page

The “Configuration > Flasher modules” page shows the serial numbers of the Flasher Compacts that are currently connected to the Flasher Hub, as well as each modules’ assigned nickname. This page also allows users to change the association between module number (Position) and Flasher Compact serial number.

To assign a specific Flasher Compact to a specific module number, choose the desired index from the drop-down menu. Please make sure each index is unique, otherwise the changes cannot be saved.

In addition to that, this page allows users to change the module nicknames.

> **Note**
>
> In order to reposition or rename modules in this tab, the current web session needs to be in Administrator mode. For more information please refer to *Web interface configuration* on page 50.



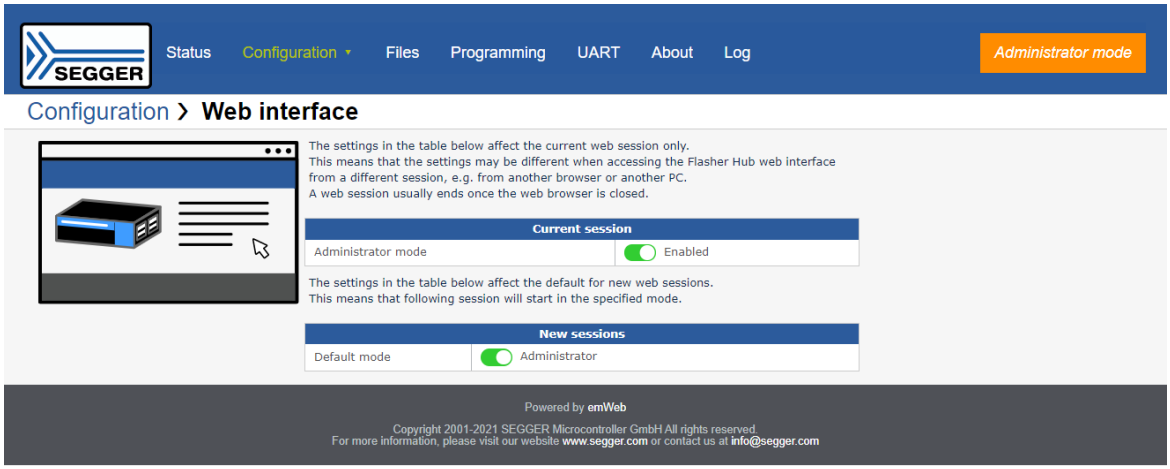
7.1.4 Flasher Hub web interface configuration page

The “Configuration > Web interface” page allows switching between “Administrator mode” and “Operator mode” for the current web session. A web session that is in “Operator mode” cannot apply changes to the Flasher Hub, e.g. module positions as well as nicknames cannot be changed, Flasher Hub and module firmware cannot be updated, etc.

In order to change the Flasher Hub setup, “Administrator mode” needs be enabled for the web session first.

Furthermore, the default mode for new web sessions can be changed in this tab.

For production environments where there are multiple users accessing the Flasher Hub, it is recommended to disable “Administrator mode” for new sessions by default so accidental setup changes are prevented.



7.1.5 Files page (File browser)

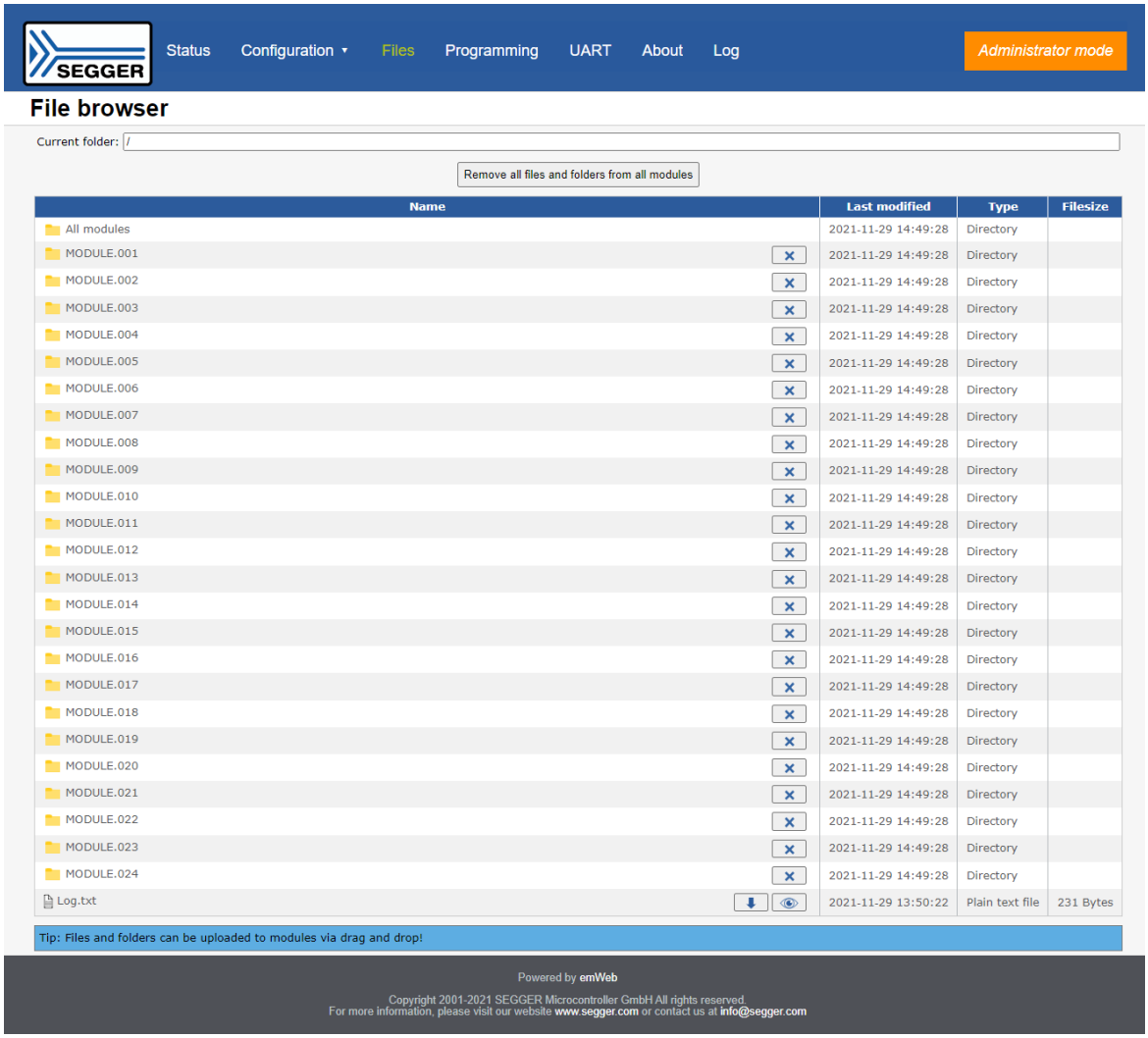
The file browser on the "Files" page is an alternative to using an FTP client to manage the files on the connected Flasher Compacts. Users can view the contents of each module, upload files (also via drag-and-drop), delete files and folders, create new folders, and more.

Just like when connecting via FTP, files uploaded to / new folders created in the "All modules" directory automatically get copied to all connected Flasher Compacts. Note that the "All modules" directory itself does not retain any files or folders and will always appear empty.

The FTP server root directory is a virtual directory and cannot be written to. Enter a specific module directory or the "All modules" directory to upload files.

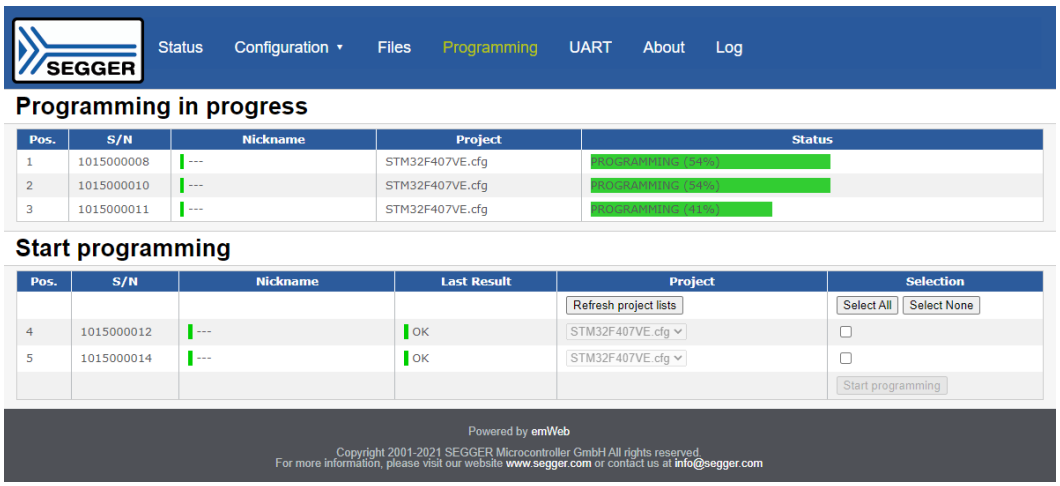
Note that the assignment of Flasher Compact serial numbers to the Module numbers (MODULE.xxx) can be changed via the Flasher configuration page.

> **Note**
>
> In order to upload or remove files and folders in this tab, the current web session needs to be in Administrator mode. For more information please refer to *Web interface configuration* on page 50.



7.1.6 Programming page

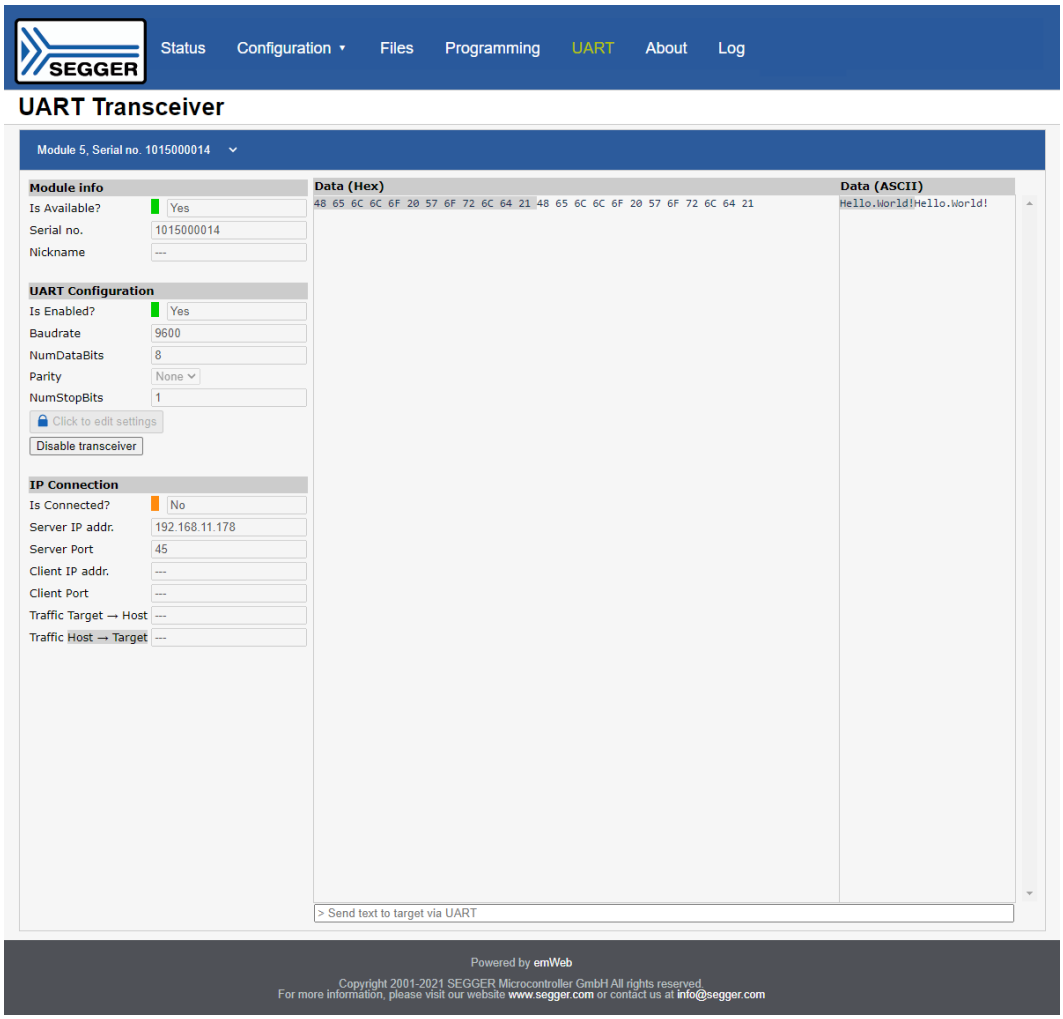
The “Programming” page allows users to start and monitor ongoing programming operations using the web interface. Users can select which modules should start programming as well as the project file to use.



 © 2020-2022 SEGGER Microcontroller GmbH

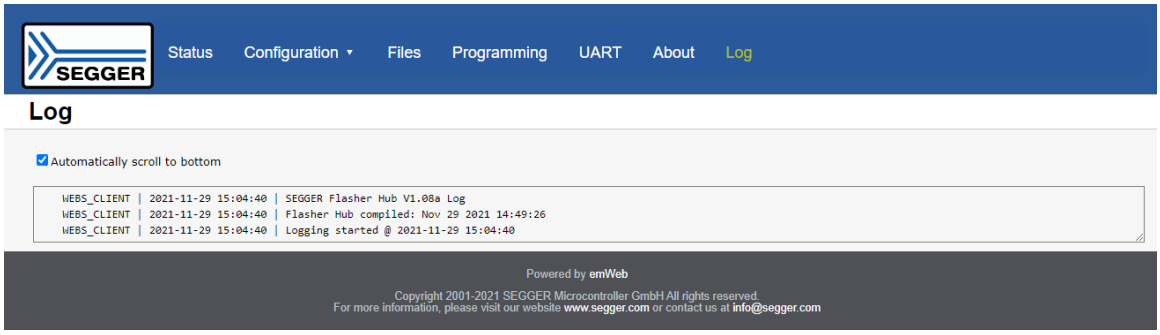
7.1.7 UART page

The “UART” page allows users to enable / disable and monitor the UART to TCP transceiver for each module.



7.1.8 Log page

The “Log” page shows the log kept by the Flasher Hub.



Chapter 8

Remote control

This chapter describes how to control the Flasher Hub based system via the integrated Telnet interface.

© 2020-2022 SEGGER Microcontroller GmbH

8.1 Overview

The integrated Telnet interface is the primary way to remote-control the Flasher Hub operation.

In the future, remote control via handshake lines may be added.

 © 2020-2022 SEGGER Microcontroller GmbH

8.2 ASCII command interface

8.2.1 Introduction

The Flasher Hub can be driven by any application or just a simple terminal using ASCII commands.

Every known command is acknowledged by the Flasher Hub and then executed. After command execution, the Flasher Hub sends an ASCII reply message.

> **Note**
>
> There are situations where the execution of a known command is rejected with #NACK:ERRxxx if the Flasher Hub is currently busy and the received command is not allowed to be sent while the Flasher Hub is busy.

8.2.2 General command and reply message format

- Any ASCII command has to start with the start delimiter #.
- Any ASCII command has to end with simple carriage return ('\r', ASCII code 13).
- Commands can be sent upper or lower case.

8.2.3 General usage

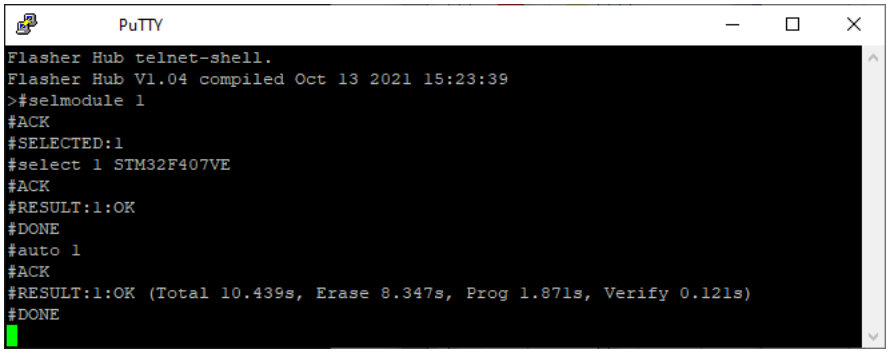
Reply messages must be considered in each case. In general, a new command must not be sent before a reply for the last one has been received. At least the "#ACK" message needs to be received by the controlling application before sending a new command for a flash module not yet executing a command.

For the Flasher Hub, all commands triggering a flash programming function (`#AUTO`, `#CANCEL`, `#ERASE`, `#PROGRAM`, `#VERIFY`) may be used for other modules, before the current operation has been finished. Please note that in this case the overall finish indicator "#DONE" will be sent when all commands have been executed.

When a flash programming function has finished, the debug logic of the MCU is disabled (powered down) and the target interface of the module is switched off (tristated).

8.2.4 Settings for ASCII interface via Telnet

A client application can connect to the Flasher Hub via Telnet on port 23. Find below a screenshot of a Flasher Hub being remote controlled via Telnet:



8.2.5 Commands and replies

The table below provides an overview about the commands which are supported by the current version of the Flasher Hub firmware. Click on the names for a detailed description:

 © 2020-2022 SEGGER Microcontroller GmbH

Commands to the Flasher Hub
#ASSIGN <Serial>, <Position>
#AUTO <Module1>[,<Module2>]
#AUTO NOPATCH <Module1>[,<Module2>]
#AUTO PATCH <Module1>[,<Module2>] [number of patches],[address],[number of patched bytes]:[data bytes]
#CANCEL <Module1>[,<Module2>]
#ERASE <Module1>[,<Module2>]
#FWVERSION
#FWVERSIONMOD <Module1>[,<Module2>]
#IPCONFIG
#POWERON <Module1>[,<Module2>], [PowerSource, Discharge Mode]
#POWEROFF <Module1>[,<Module2>]
#PROGRAM <Module1>[,<Module2>]
#PROJECT <Module1>[,<Module2>]
#PROTVER
#RESETIPCONFIG
#RESTART
#RESULT <Module1>[,<Module2>]
#SELECT <Module1>[,<Module2>] <Filename>
#SELMODULE <Module1>[,<Module2>]
#SERIAL
#SERIALMOD <Module1>[,<Module2>]
#SETVTREF <Module1>[,<Module2>] [voltage]
#START <Module1>[,<Module2>]
#STATUS
#TERMINAL <Module1>[,<Module2>] <Baudrate>,<NumDataBits>,<Parity>,<NumStopBits>
#VERIFY <Module1>[,<Module2>]

Replies from the Flasher Hub
#ACK
#NACK
#OK
#OK:<NumBytes>:<Data>
#OK:<Data>
#OK:<Module>:<Data>
#RESULT:<Module>:
#DONE
#ERRxxx

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1 Commands to the Flasher Hub

8.2.5.1.1 Command #ASSIGN

The serial number of a Flasher Compact is assigned to the module position of the Flasher Hub.
When the new module position is taken, a swap is performed. The Flasher Compact with the serial number must be connected to the Flasher Hub.

Command structure:

```
#ASSIGN <module>, <position>
```

Example sequence:

Command sent to the Flasher Hub	Reply
##ASSIGN 1015000001, 1	
	#ACK
	#OK

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.2 Command #AUTO

The #AUTO command behaves exactly like pushing the programming button on a connected Flasher Compact.

Usually, the following command sequence will be performed when receiving the #AUTO command:

- The Flasher Hub erases the target MCU (if not blank)
- The Flasher Hub programs the target MCU
- The Flasher Hub verifies the target MCU

Depending on the settings chosen in the **Production** tab in the J-Flash tool, this sequence can differ from the one shown above.

Command structure:

```
#AUTO [module1][, module2][, module3]
```

Alternatively the modules can be replaced with:

- `*`, will execute the `auto` command using all modules selected by the latest executed `selmodule` command.
- `all`, will execute the `auto` command using all modules which can be detected.

Results provided by the Flasher Hub:

Result	Meaning
#OK	Programming done successfully.
#ERRxxx	Error occurred during operation. xxx represents the error code, which may be followed by an additional error text.

During execution of the #AUTO command, the Flasher Hub automatically sends "status" messages via the terminal connection to reflect the state of execution.

Example sequence:

Command sent to the Flasher Hub	Reply
#AUTO 1	
	#ACK
	#RESULT:1:OK (Total 13.993s, Erase 0.483s, Prog 9.183s, Verify 2.514s)

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.3 Command #AUTO NOPATCH

The #AUTO NOPATCH command allows to ignore an existing patch file for the programming.

This command exists because the default behavior of the #auto command is such that an existing patch file (patch.txt in the module folder) is applied to a data if the #auto command is executed.

The Flasher Hub responds with

- #0K if no error occurred
- #ERRxxx if any error occurred during operation. xxx represents the error code, which may be followed by an additional error text.

For further information about the usage of the #AUTO PATCH command, please refer to *Patch file support* on page 42.

Example sequence:

Command sent to the Flasher Hub	Reply
#AUTO NOPATCH 1	
	#ACK
	#RESULT:1:OK (Total 13.993s, Erase 0.483s, Prog 9.183s, Verify 2.514s)

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.4 Command #AUTO PATCH

The #AUTO PATCH command allows patching of the content of the data to be programmed.

The Flasher Hub responds with

- #OK if no error occurred
- #ERRxxx if any error occurred during operation. xxx represents the error code, which may be followed by an additional error text.

For further information about the usage of the #AUTO PATCH command please refer to *Patch file support* on page 42.

Example sequence:

Command sent to the Flasher Hub	Reply
#AUTO PATCH 1 1,0,8:0011223344556677	
	#ACK
	#RESULT:1:OK (Total 13.993s, Erase 0.483s, Prog 9.183s, Verify 2.514s)

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.5 Command #CANCEL

This command can be sent to abort a running operation. It may take a while until the current operation is actually canceled.

Command structure:

```
#Cancel [module1][, module2][, module3]
```

Example:

Command sent to the Flasher Hub	Reply
#AUTO 1	
	#ACK
#CANCEL 1	
	#ACK
	#RESULT:1:ERR255:Error while flash-ing

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.6 Command #ERASE

This command can be sent to erase all selected target flash sectors.

Command structure:

`#ERASE [module1][,module2][,module3]`

Results provided by the Flasher Hub:

Result	Meaning
`#OK`	Erase done successfully.
`#ERRxxx: TEXT`	Error message with text

Example sequence:

Command sent to the Flasher Hub	Reply
`#ERASE 1`	
	`#ACK`
	`#RESULT:1:OK (Total 0.362s, Erase 0.252s)`

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.7 Command #FWVERSION

This command returns the firmware version of the Flasher Hub.

Command structure:

```
#FWVERSION
```

Example sequence:

Command sent to the Flasher Hub	Reply
#FWVERSION	
	#ACK
	#OK:1:1.00
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.8 Command #FWVERSIONMOD

This command returns the firmware version of one or more modules connected to the Flasher Hub.

Command structure:

```
#FWVERSIONMOD [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#FWVERSIONMOD 1,2,3	
	#ACK
	#OK:1:J-Link / Flasher Compact V5 compiled Mar 17 2021 11:50:31
	#OK:2:J-Link / Flasher Compact V5 compiled Mar 17 2021 11:50:31
	#OK:3:J-Link / Flasher Compact V5 compiled Mar 17 2021 11:50:31
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.9 Command #IPCONFIG

This command returns the current IP configuration.

Command structure:

`#IPCONFIG`

Example sequence:

Command sent to the Flasher Hub	Reply
`#IPCONFIG`	
	`#ACK`
	`#RESULT:IP address:192.168.1.111`
	`#RESULT:subnet mask:255.255.0.0`
	`#RESULT:Gateway:192.168.1.1`
	`#RESULT:IP mode:automatic(DHCP) assigned`
	`#DONE`

IP mode can be `automatic(DHCP) assigned` or `User assigned`.

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.10 Command #POWERON

This command can be used to turn ON the target power (5V on pin 19 of the attached Flasher Compacts) without any erase, program or verify action.

Command structure:

`#POWERON [module1][,module2][,module3] [Power Source],[Discharge]`

The `#POWERON` command expects the following parameters:

Parameter	Meaning
Power Source	0 = internal power
Discharge	0 = no discharge

Example sequence:

Command sent to the Flasher Hub	Reply
`#POWERON 1,2,3 0,0`	
	#ACK
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.11 Command #POWEROFF

This command can be used to turn OFF the target power (5V on pin 19 of the attached Flasher Compacts) without any erase, program or verify action.

Command structure:

```
#POWEROFF [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#POWEROFF 1,2,3	
	#ACK
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.12 Command #PROGRAM

This command can be used instead of #AUTO to program a target without erasing the target before programming and without performing a final verification.

Command structure:

```
#PROGRAM [module1][,module2][,module3]
```

The Flasher Hub will reply with the following sequence of messages:

Example sequence:

Command sent to the Flasher Hub	Reply
#PROGRAM 1	
	#ACK
	#RESULT:1:OK (Total 9.963s, Prog 9.183s)

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.13 Command #PROJECT

Returns the selected projects from the specified module numbers.

Command structure:

`#PROJECT <module1>[,module2][,module3]`

The Flasher Hub will reply with the following sequence of messages:

Example sequence:

Command sent to the Flasher Hub	Reply
`#PROJECT 1,2`	
	`#ACK`
	`#RESULT:1:OK:FLASHER.UNI`
	`#RESULT:2:OK:PROJECT.CFG`
	`#DONE`

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.14 Command #PROTVER

This command can be used to check the version of the ASCII command protocol in use.

Command structure:

#PR0TVER

Example sequence:

Command sent to the Flasher Hub	Reply
#PR0TVER	
	#ACK
	#OK:2.02b)
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.15 Command #RESETIPCONFIG

Reset the IP configuration to DHCP.

Command structure:

```
#RESETIPCONFIG
```

Example sequence:

Command sent to the Flasher Hub	Reply
#RESETIPCONFIG	
	#ACK
	#RESULT:The new IP configuration will take effect after restarting Flasher Hub.
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.16 Command #RESTART

Resets the IP configuration to DHCP.

Command structure:

#RESTART

Example sequence:

Command sent to the Flasher Hub	Reply
#RESTART	

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.17 Command #RESULT

This command can be sent any time, even during other command execution. The Flasher Hub responds with the result of the previously executed command.

Command structure:

```
#RESULT [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#RESULT 1,2,3	
	#ACK
	#RESULT:1:OK ((Total 2.216s, Erase 0.126s, Prog 1.231s, Verify 0.144s)
	#RESULT:2:OK ((Total 2.216s, Erase 0.126s, Prog 1.231s, Verify 0.144s)
	#RESULT:3:OK ((Total 2.216s, Erase 0.126s, Prog 1.231s, Verify 0.144s)
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.18 Command #SELECT

The #SELECT command is used to select a specific configuration and data file pair to be used by the connected Flasher Compact modules to program the target.

Command structure:

```
#select [module1][,module2][,module3] [Project Name]
```

The #SELECT command expects the following parameters:

Parameter	Meaning
Project Name	The [Project Name] specifies the name of file pair without extensions (.CFG and .DAT) on the Flasher Compact modules to be selected. The Flasher Compacts save the selected configuration and data file in the FLASHER.INI file. Therefore, this selection is remembered even after power-cycling the Flasher Compacts.

Example sequence:

Command sent to the Flasher Hub	Reply
#SELECT 1,2,3 emPower	
	#ACK
	#RESULT:1:OK
	#RESULT:2:OK
	#RESULT:3:OK
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.19 Command #SELMODULE

This command is used to select one or more modules in a Flasher Hub system. The module numbers are separated by a comma. If all modules shall be selected, the keyword "all" can be used (`#SELMODULE ALL`) instead of a list with all module numbers.

Command structure:

```
#SELMODULE [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
`#SELMODULE 1,2,3`	
	#ACK
	`#SELECTED:1,2,3`

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.20 Command #SERIAL

The #SERIAL command is used query the serial number of the Flasher Hub.

Command structure:

```
#serial
```

Example sequence:

Command sent to the Flasher Hub	Reply
#serial	
	#ACK
	#RESULT:1021000001
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.21 Command #SERIALMOD

The #SERIALMOD command is used query the serial numbers of the connected Flasher Compact modules.

Command structure:

```
#SERIALMOD [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#serialmod 1,2,3	
	#ACK
	#RESULT:1:1015000015
	#RESULT:2:1015000016
	#RESULT:3:1015000017
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.22 Command #SETVTREF

This command can be used to set a fixed voltage for I/O pins of the target interface.

Command structure:

```
#SETVTREF [module1][,module2][,module3] [voltage level]
```

The SETVTREF command expects the following parameters:

Parameter	Meaning
voltage level	The IO voltage level for the target interface in mV.

Example sequence:

Command sent to the Flasher Hub	Reply
#SETVTREF 1,2,3 3300	
	#ACK
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.23 Command #START

This command can be sent to start the application using the method configured in the J-Flash project.

Command structure:

```
#START [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#START 1,2,3	
	#ACK
	#RESULT:1:OK (Total 0.083s)
	#RESULT:2:OK (Total 0.082s)
	#RESULT:3:OK (Total 0.084s)
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.24 Command #STATUS

This command can be sent any time, even during other command execution. The Flasher Hub responds with its current state. All defined state messages are described under *Replies from Flasher Hub* on page 85.

Command structure:

#STATUS

Example sequence:

Command sent to the Flasher Hub	Reply
#STATUS	
	#ACK
	#STATUS:READY

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.25 Command #TERMINAL

This command enables/ disables the UART transceiver for the given module(s). For more information, please refer to *the UART to TCP transceiver* on page 23.

Command structure:

```
#TERMINAL [module1][,module2][,module3] [Baudrate],[NumDataBits],[Parity],[NumStopBits]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#TERMINAL 1,2,3 9600,8,N,1	
	#ACK
	#DONE

Command structure:

```
#TERMINAL [module1][,module2][,module3] off
```

Example sequence:

Command sent to the Flasher Hub	Reply
#TERMINAL 1,2,3 off	
	#ACK
	#DONE

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.1.26 Command #VERIFY

This command can be used to verify the target flash content against the data stored in the Flasher Compact module.

Command structure:

```
#VERIFY [module1][,module2][,module3]
```

Example sequence:

Command sent to the Flasher Hub	Reply
#VERIFY 1,2,3	
	#ACK
	#RESULT:1:OK (Total 0.206s, Verify 0.129s)
	#RESULT:2:OK (Total 0.210s, Verify 0.131s)
	#RESULT:3:OK (Total 0.207s, Verify 0.128s)

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.2 Replies from the Flasher Hub

The reply messages from the Flasher Hub follow the same data format as commands. Any reply message starts with ASCII start delimiter #, ends with simple carriage return (ASCII code 13), and is sent in uppercase. In contrast to commands, replies can be followed by a descriptive message, which provides more detailed information about the reply. This description is sent in mixed case. The #OK reply, for example, is such a reply. It is followed by a string containing information about the performance time needed for the operations:

```
#OK (Total 13.993s, Erase 0.483s, Prog 9.183s, Verify 2.514s)
```

The following reply messages from the Flasher Hub are defined:

8.2.5.2.1 #ACK

The Flasher Hub replies with the #ACK message on reception of any defined command before the command itself is executed.

8.2.5.2.2 #NACK

The Flasher Hub replies with #NACK, if an undefined command was received.

8.2.5.2.3 #OK:<module>:<data>

The Flasher Hub replies with #OK, if a command other than #STATUS or #RESULT was executed and ended without any error. #OK may be followed by a module number and/or data.

8.2.5.2.4 #STATUS:<status>

The Flasher Hub replies with its current state.

The following status messages are currently defined:

Message	Description
#STATUS:READY	Flasher Hub is ready to receive a new command.
#STATUS:BUSY	Flasher Hub is currently executing a previously received command.

 © 2020-2022 SEGGER Microcontroller GmbH

8.2.5.2.5 #RESULT:<module>:<data>

The Flasher Hub reports the result of an operation on a specific module. If the operation has been completed successfully, it will report the outcome with a single message of this type followed by the last result of the operation.

8.2.5.2.6 #DONE

This message is being sent once all operations are finished and all connected Flasher Compact modules are back in idle state.

A typical sequence for using the Flasher Hub is shown below:

```
Flasher Hub telnet-shell telnet-shell.
Flasher Hub V1.01a compiled May 17 2021 10:19:45
#SELMODULE 1,2
#ACK
#SELECTED:1,2
#AUTO *
#ACK
#RESULT:1:#ERR255:Error while flashing
#RESULT:2:#OK (Total 2.653s, Erase 0.327s, Prog 1.960s, Verify 0.234s)
#DONE
```

8.2.5.2.7 #ERRxxx <data>

If any command other than #STATUS or #RESULT was terminated with an error, the Flasher Hub cancels the command and replies with an error message instead of the #OK message.

Some error codes may be followed by a colon and an additional error text.

For example:

```
#ERR007:CANCELED.
```

The error code numbers are described in the following table:

Message	Description
#ERR007	Flasher Hub received a #CANCEL command and has canceled the current operation.
#ERR008	Flasher Hub is already busy with execution of previous command.
#ERR009	Failed to allocate memory.
#ERR010	Failed to open file.
#ERR011	Failed to read file.
#ERR012	Failed to write file.
#ERR013	Failed to delete file.
#ERR098	Failed to delete file.
#ERR098	Could not allocate memory for device specific algorithm.
#ERR099	Device specific algorithm is not yet supported by this firmware version. Please check for a firmware update.
#ERR101	Could not find device programming algorithm.
#ERR102	Could not open the data file.
#ERR255	Undefined error occurred. This reply is followed by an error string.

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 9

Support and FAQs

This chapter contains troubleshooting tips together with solutions for common problems which might occur when using the Flasher Hub. There are several steps you can take before contacting support. Performing these steps can solve many problems and often eliminates the need for assistance. This chapter also contains a collection of frequently asked questions (FAQs) with answers.

© 2020-2022 SEGGER Microcontroller GmbH

9.1 Contacting support

Before contacting support, make sure you tried to solve your problem by trying your Flasher Hub with a different PC and (if possible) with a different target system to see if it works there. If the device functions correctly, the USB setup on the original machine or your target hardware is the source of the problem, not the Flasher Hub.

If you need to contact support, send the following information to *ticket_flasher@segger.com*

- A detailed description of the problem
- Flasher Hub serial number
- Information about your target hardware (processor, board, etc.).
- `FLASHER.JFLASH`, `FLASHER.CFG`, `FLASHER.DAT` (if possible), `FLASHER.LOG`, `SERIAL.TXT` file from the connected Flasher Compact. To get these files, please download them via FTP or via the Flasher Hub web interface.

The Flasher Hub is sold directly by SEGGER.

 © 2020-2022 SEGGER Microcontroller GmbH

9.2 Frequently Asked Questions

Maximum target interface speed

Q: What is the maximum target interface speed supported by Flasher Compact?
A: The Flasher Compact’s maximum supported target interface speed is 50MHz.

Maximum download speed

Q: What is the maximum download speed supported by Flasher Compact?
A: The maximum download speed is currently about 3 Mbytes/second when downloading into RAM. The actual speed depends on various factors, such as target interface speed, clock speed, host CPU core etc.

Flasher Hub web interface access

Q: How do I connect to the Fasher Hub’s web interface?
A: Locate the serial number of your Flasher Hub on the product housing. Then point your browser to http://flasherhub-<serial_number>/index.htm

Flasher Hub IP address

Q: Where can I find the IP address assigned to my Flasher Hub?
A: Connect to the Flasher Hub web interface as described above. The Flasher Hub IP address is located on the “Overview” page.

 © 2020-2022 SEGGER Microcontroller GmbH

Chapter 10

Mechanics

If you need to mount the Flasher Hub into the production environment, a drawing of the mechanical layout can be downloaded here:
https://www.segger.com/downloads/flasher#Mechanics.

© 2020-2022 SEGGER Microcontroller GmbH

Chapter 11

Literature and references

This chapter lists documents, which we think may be useful to gain a deeper understanding of technical details.

© 2020-2022 SEGGER Microcontroller GmbH

Reference	Title	Comments
[J-Link]	J-Link / J-Trace User Guide	This document describes J-Link and J-Trace. It is publicly available from SEGGER (*https://www.segger.com*).
[J-Flash]	J-Flash User Guide	This document describes J-Flash. It is publicly available from SEGGER (*https://www.segger.com*).
[Flasher Hub wiki]	Flasher Hub wiki pages	*https://wiki.segger.com/Flasher_Hub*.

 © 2020-2022 SEGGER Microcontroller GmbH