U-Flash

Software utility to prepare SEGGER Flasher units for stand-alone programming - User Guide

Document: UM08037
Software Version: 8.10
Date: February 5, 2025

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

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 U-Flash, formerly known as Universal Flash Loader Configurator.

Print date: February 5, 2025

Manual version	Revision	Date	By	Description
8.12e	0	250205	JC	Removed obsolete 3rd party references
8.10	0	241206	AW	Complete overhaul of all chapters
7.88	0	230718	MM	Added description for Universal Flash Loader’s command-line options.

Table of contents

Chapter 1

Introduction

This chapter provides a short overview about U-Flash and its features.

1.1 What is U-Flash?

U-Flash (formerly known as the Universal Flash Loader Configurator) is a software utility to prepare SEGGER Flasher units for stand-alone 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 to a connected SEGGER Flasher unit. U-Flash allows to save 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 is also used to create secure target binary packages for the SEGGER Flasher Secure programming solution.

1.1.1 Features

U-Flash supports the following features:

- Erase a device,
- Program a device,
- Verify a device,
- Read a device,
- Secure a device,
- Configure a device,
- Configure the target interface.

Please note that not all features are available for all devices.

1.1.2 Supported Cores / Devices

To see a list of devices supported by SEGGER Flasher programmers, please visit *https://www.segger.com/supported-devices/flasher/*.

If a device is listed as supported on the a.m. web page, but you can’t find it in U-Flash, please check if the device is supported in the SEGGER J-Flash programming utility instead.

Chapter 2

U-Flash

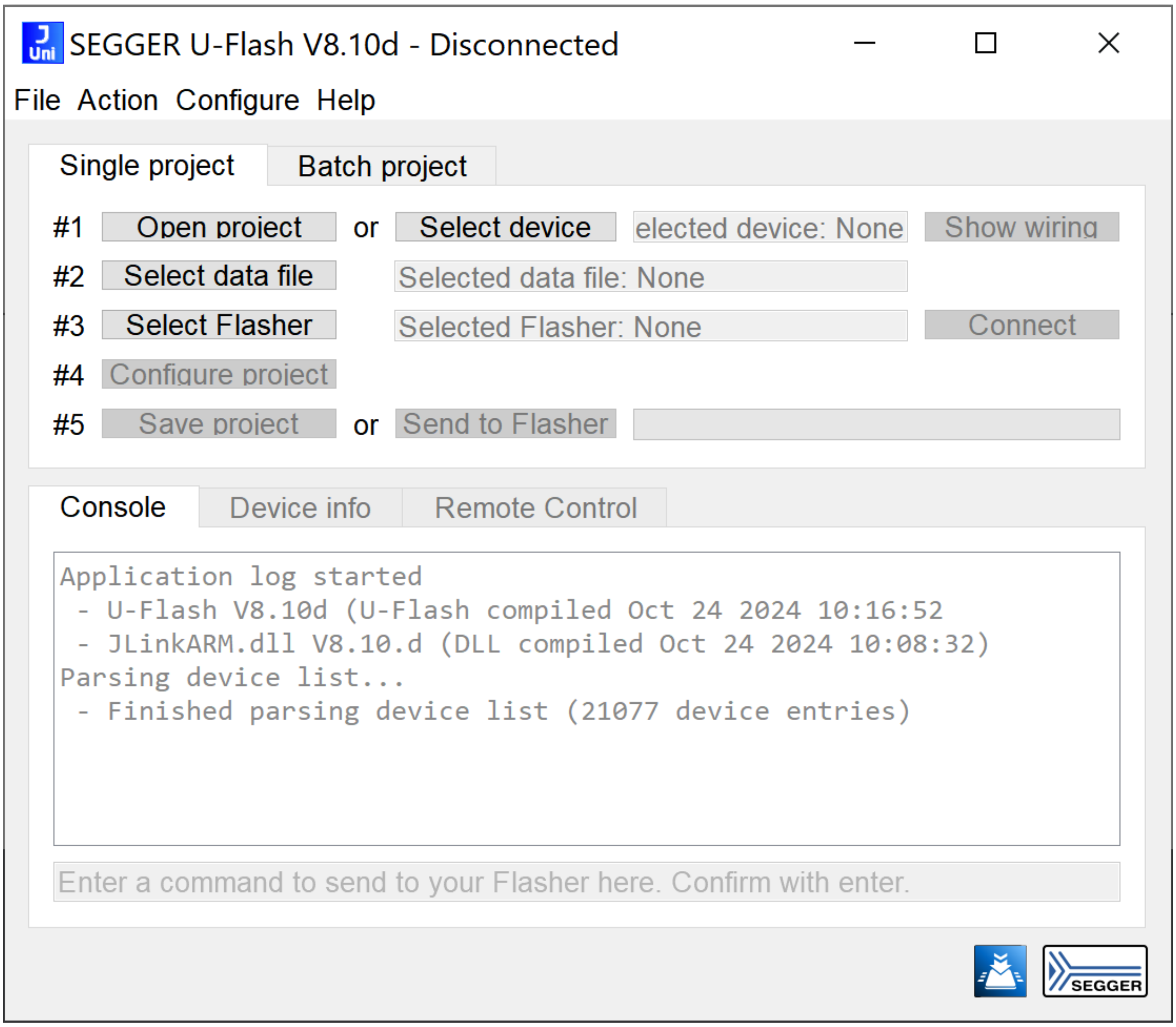
2.1 Overview

U-Flash is the software tool for setting up the project files needed for programming devices supported by the SEGGER Flasher family of production programmers, using the SEGGER Universal Flasher Loader technology.

2.2 Creating a new U-Flash project

2.2.1 Startup

When you start U-Flash, you will see the following startup screen for starting a new (single) U-Flash project:

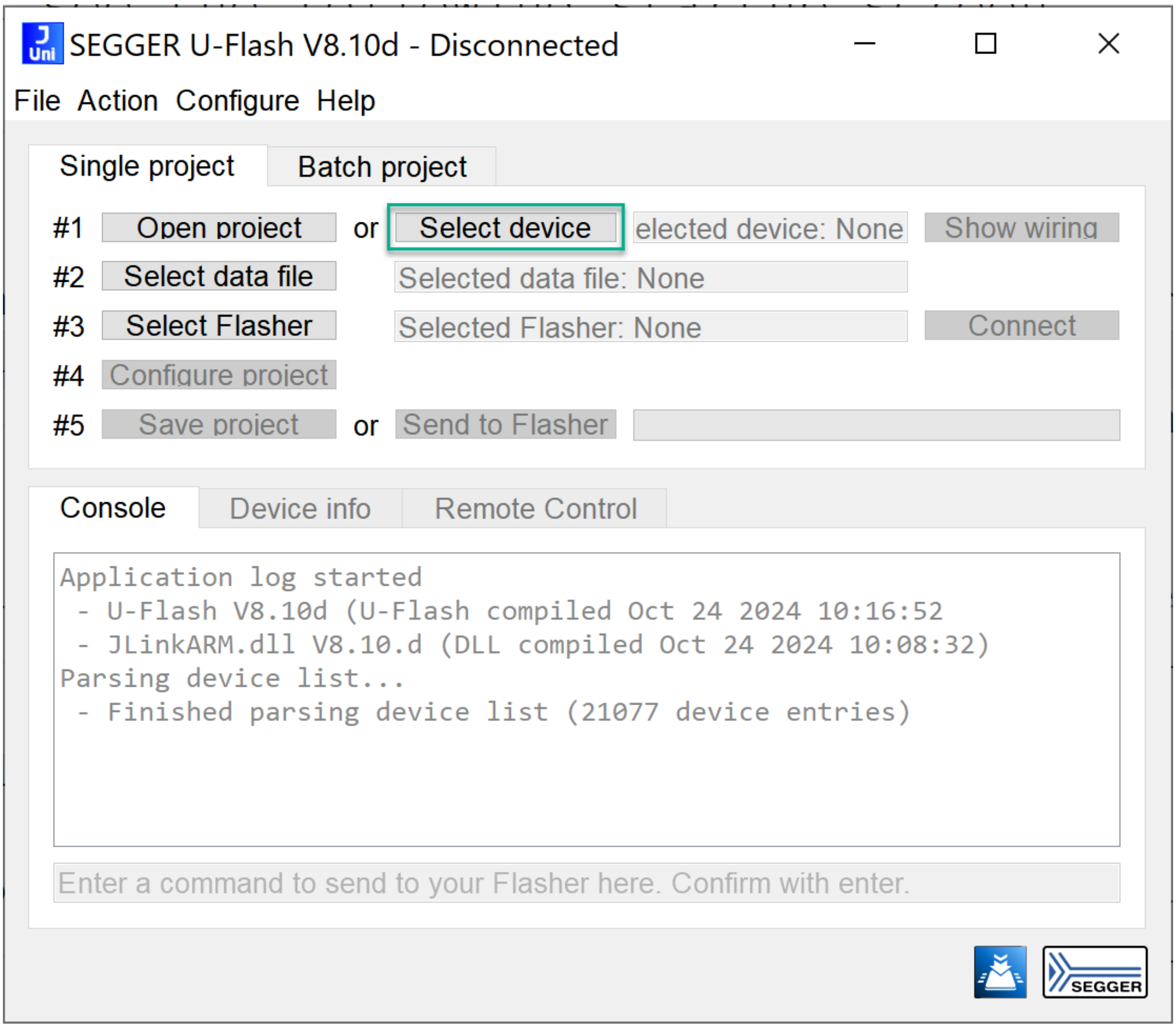


```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
```

Note: The default U-Flash project type is a **single** project. For information on how to create a **batch** project in U-Flash, please see the respective section further down in this document.

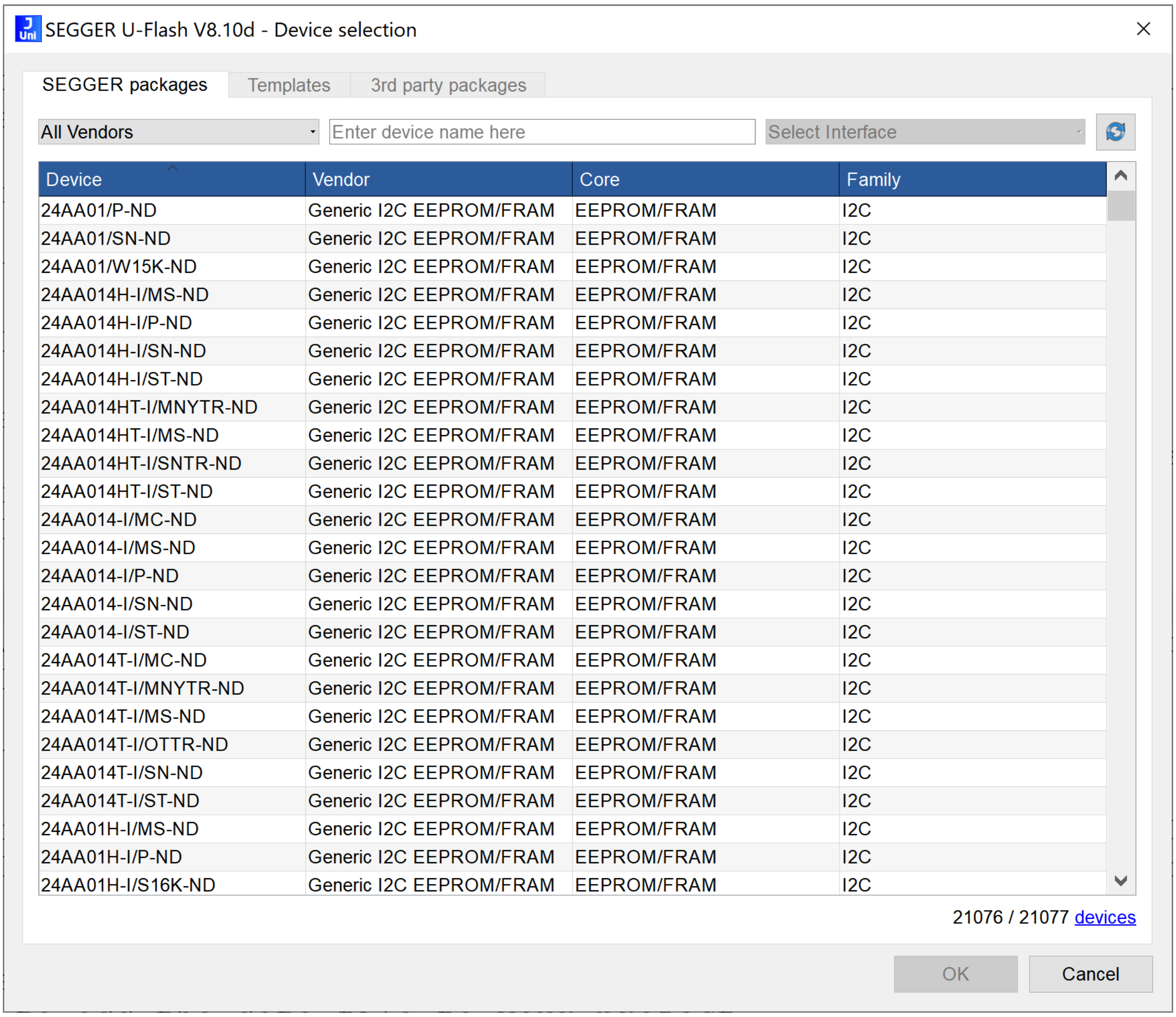
2.2.2 Selecting a device

To select the target device to be programmed, choose the menu item **Configure** > **Select Device**, or click the **Select device** button.



```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
```

This opens the device selection dialog, which lists all currently supported devices sorted by device name.



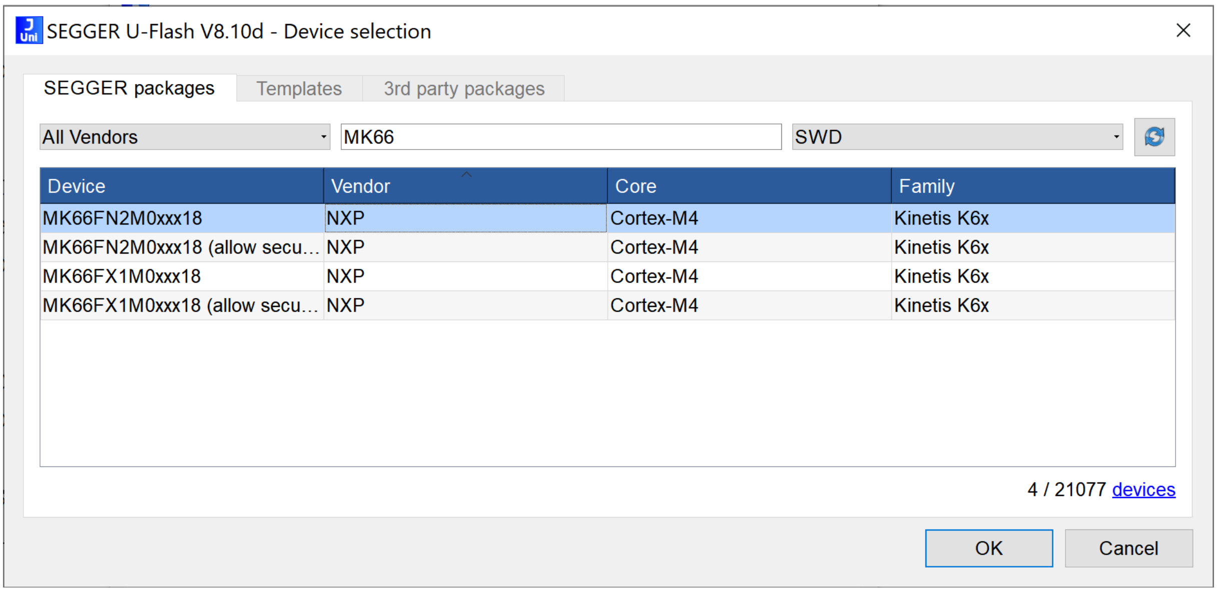
All Vendors

Device	Vendor	Core	Family
24AA01/P-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA01/SN-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA01/W15K-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014H-I/MS-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014H-I/P-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014H-I/SN-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014H-I/ST-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014HT-I/MNYTR-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014HT-I/MS-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014HT-I/SNTR-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014HT-I/ST-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014-I/MC-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014-I/MS-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014-I/P-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014-I/SN-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014-I/ST-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/MC-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/MNYTR-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/MS-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/OTTR-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/SN-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA014T-I/ST-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA01H-I/MS-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA01H-I/P-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C
24AA01H-I/S16K-ND	Generic I2C EEPROM/FRAM	EEPROM/FRAM	I2C

21076 / 21077 devices

By clicking on the blue column headers, you can also sort the device list by vendor, core, or device family.

You can narrow down the list by selecting a specific vendor from the pull-down menu or by entering the device name into the corresponding field.



All Vendors

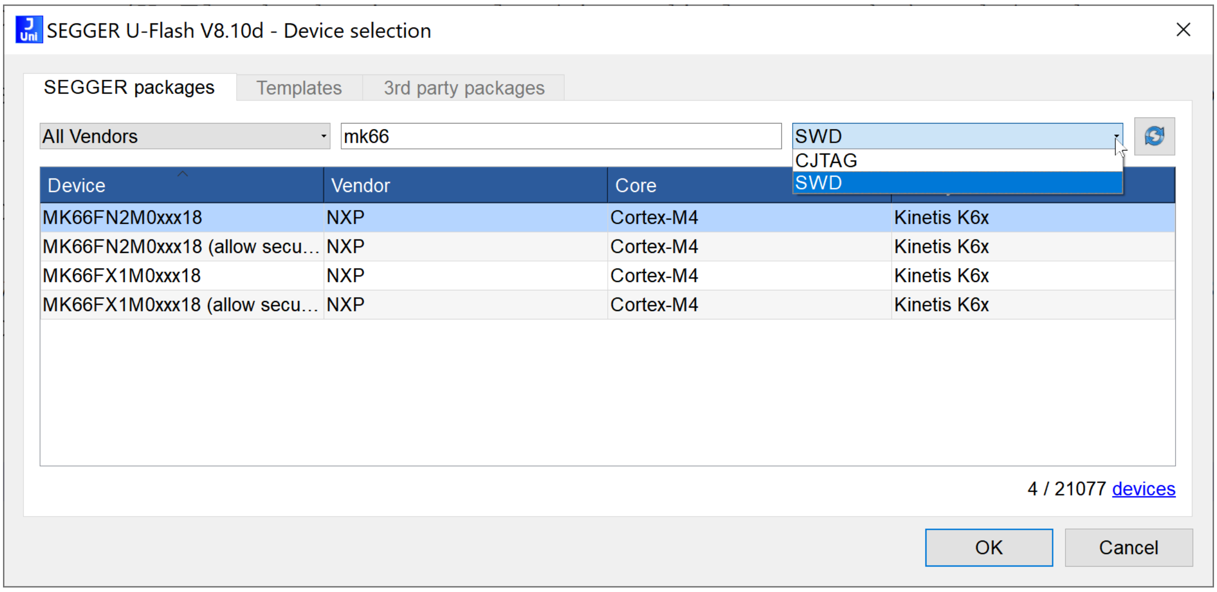
MK66

SWD

Device	Vendor	Core	Family
MK66FN2M0xxx18	NXP	Cortex-M4	Kinetis K6x
MK66FN2M0xxx18 (allow secu…	NXP	Cortex-M4	Kinetis K6x
MK66FX1M0xxx18	NXP	Cortex-M4	Kinetis K6x
MK66FX1M0xxx18 (allow secu…	NXP	Cortex-M4	Kinetis K6x

4 / 21077 devices

After you selected your device, you can choose one of the available target interfaces for the device from the interface pull-down menu.



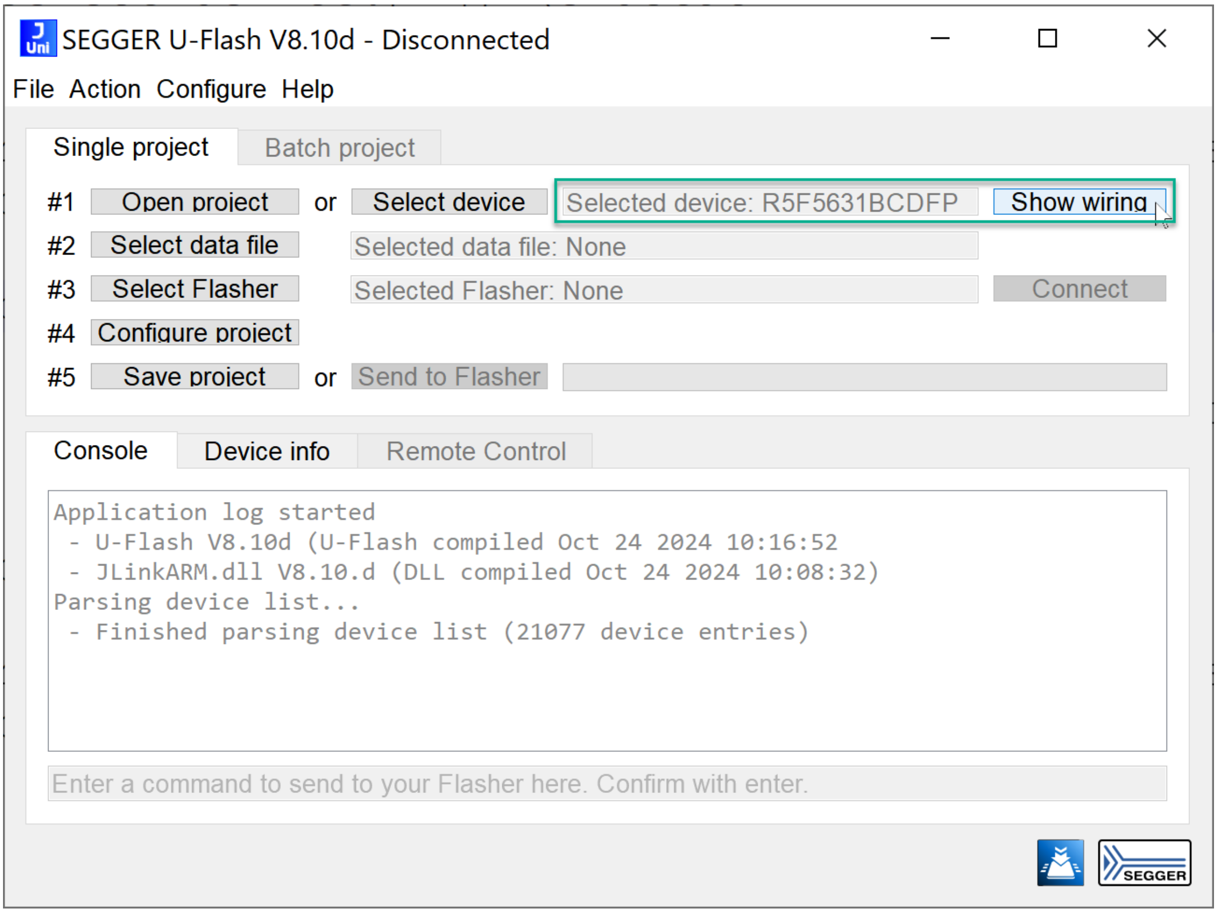
All Vendors | mk66 | SWD

Device	Vendor	Core	
MK66FN2M0xxx18	NXP	Cortex-M4	Kinetis K6x
MK66FN2M0xxx18 (allow secu…	NXP	Cortex-M4	Kinetis K6x
MK66FX1M0xxx18	NXP	Cortex-M4	Kinetis K6x
MK66FX1M0xxx18 (allow secu…	NXP	Cortex-M4	Kinetis K6x

4 / 21077 devices

Then click **OK**.

If you selected a target interface other than one of the common debug interfaces, clicking on the **Show wiring** button brings up the wiring diagram required to program the device. (Otherwise, this button is grayed out.)



Selected device: R5F5631BCDFP

Selected data file: None

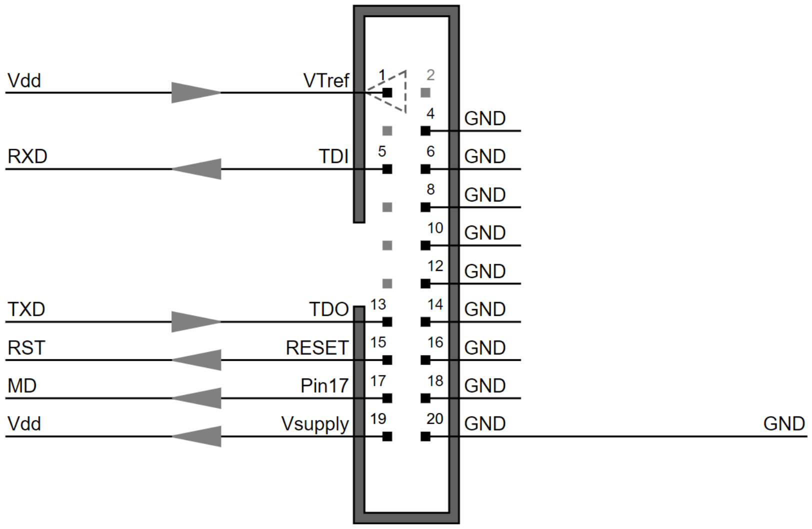
Selected Flasher: None

```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
```

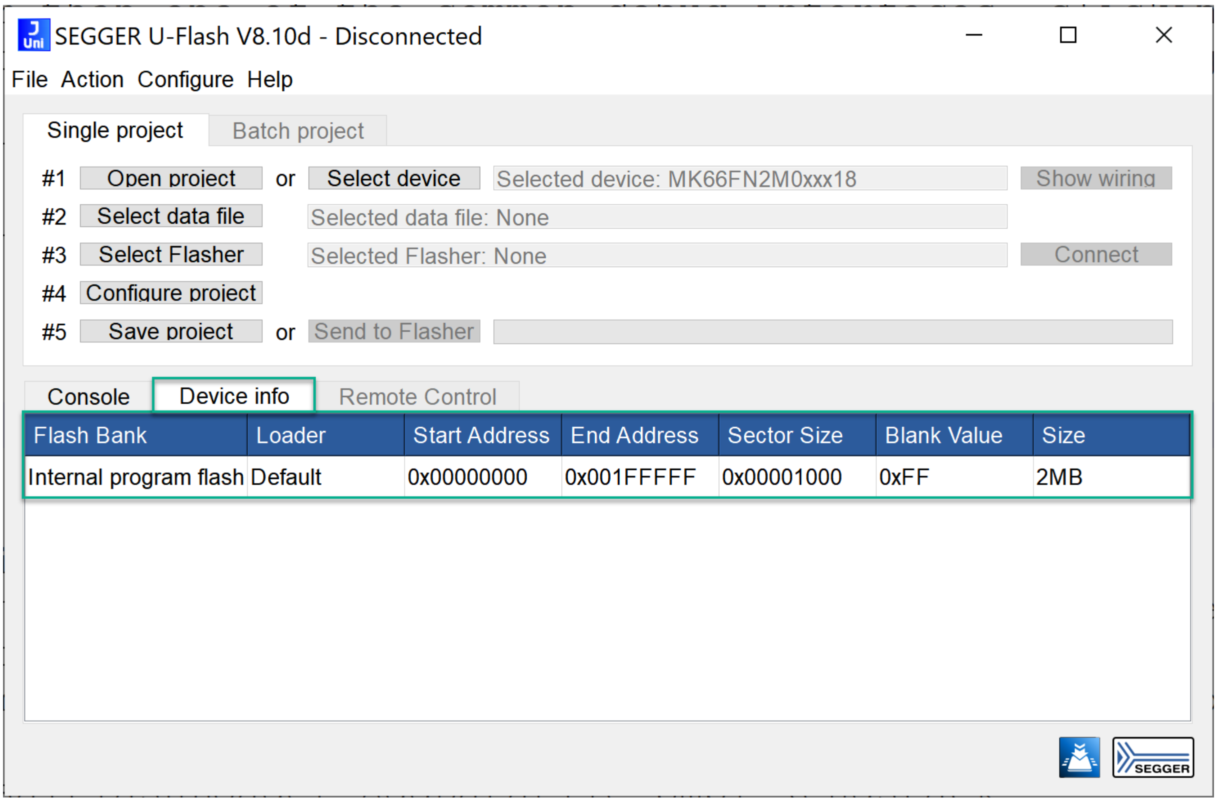
Enter a command to send to your Flasher here. Confirm with enter.

Information

Wiring (Wiring-RX63N-BL RX-Fine Adapter)



Clicking on the **Device info** tab provides you with flash-programming related information about the selected target device.



Selected device: MK66FN2M0xxx18

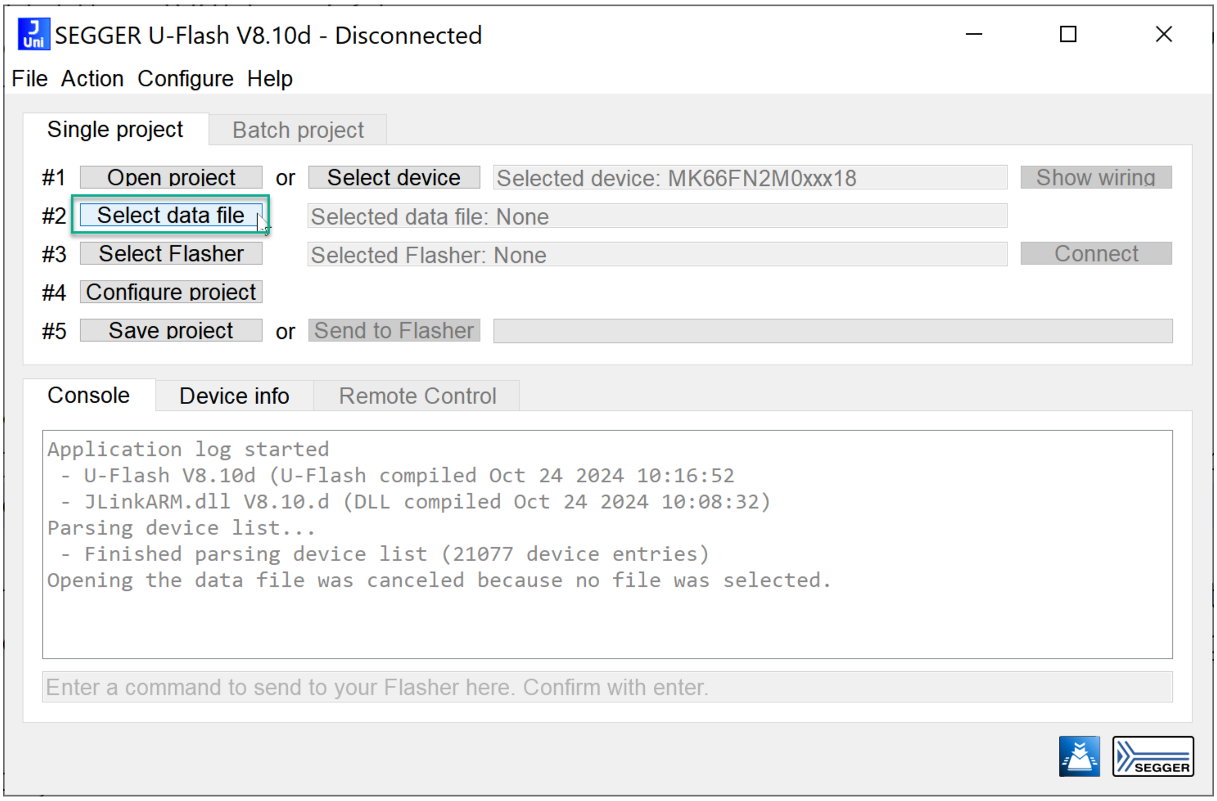
Selected data file: None

Selected Flasher: None

Flash Bank	Loader	Start Address	End Address	Sector Size	Blank Value	Size
Internal program flash	Default	0x00000000	0x001FFFFF	0x00001000	0xFF	2MB

2.2.3 Adding a data file

The next step is to add a data file (*.dat, *.elf, *.bin, *.hex, *.mot, *.s, *.s19, or *.srec) to your project. You can add the data file by selecting **File** > **Open Data File** from the menu and browsing to the data file. You can also click the **Select data file** button.



Selected device: MK66FN2M0xxx18

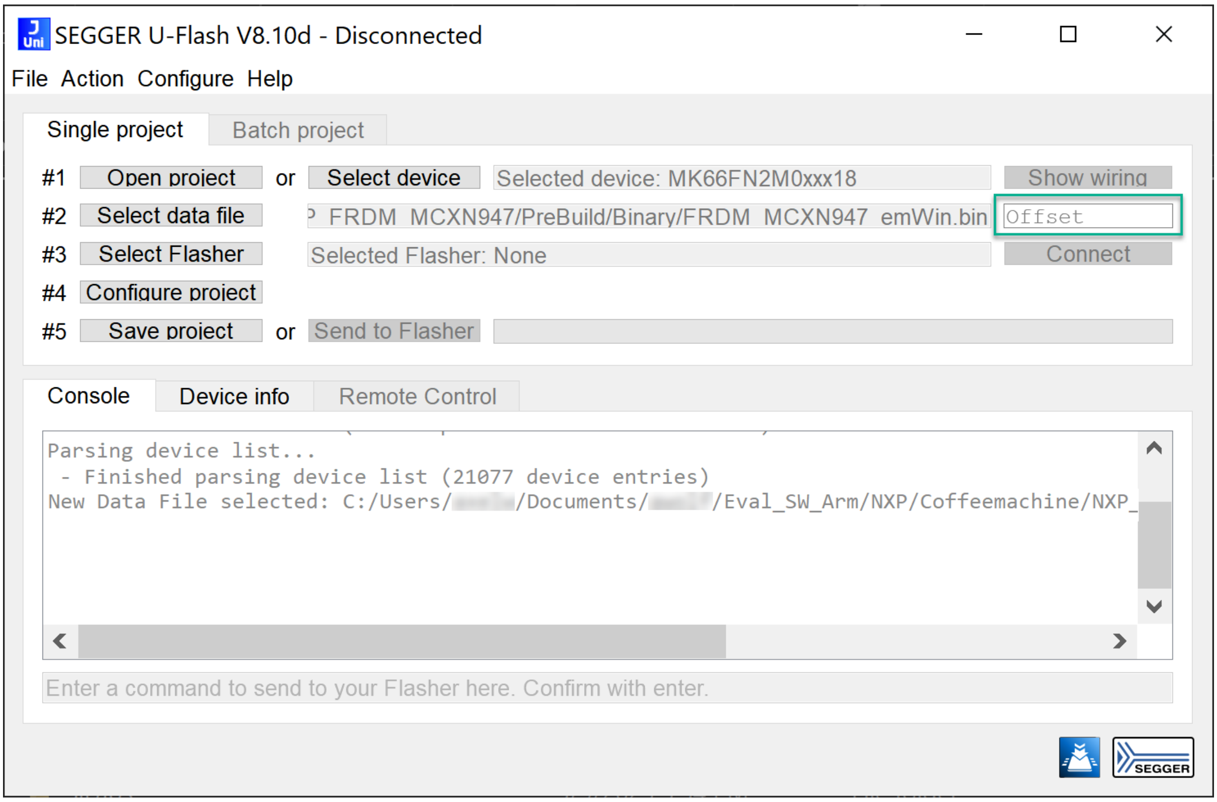
Selected data file: None

Selected Flasher: None

```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
Opening the data file was canceled because no file was selected.
```

Enter a command to send to your Flasher here. Confirm with enter.

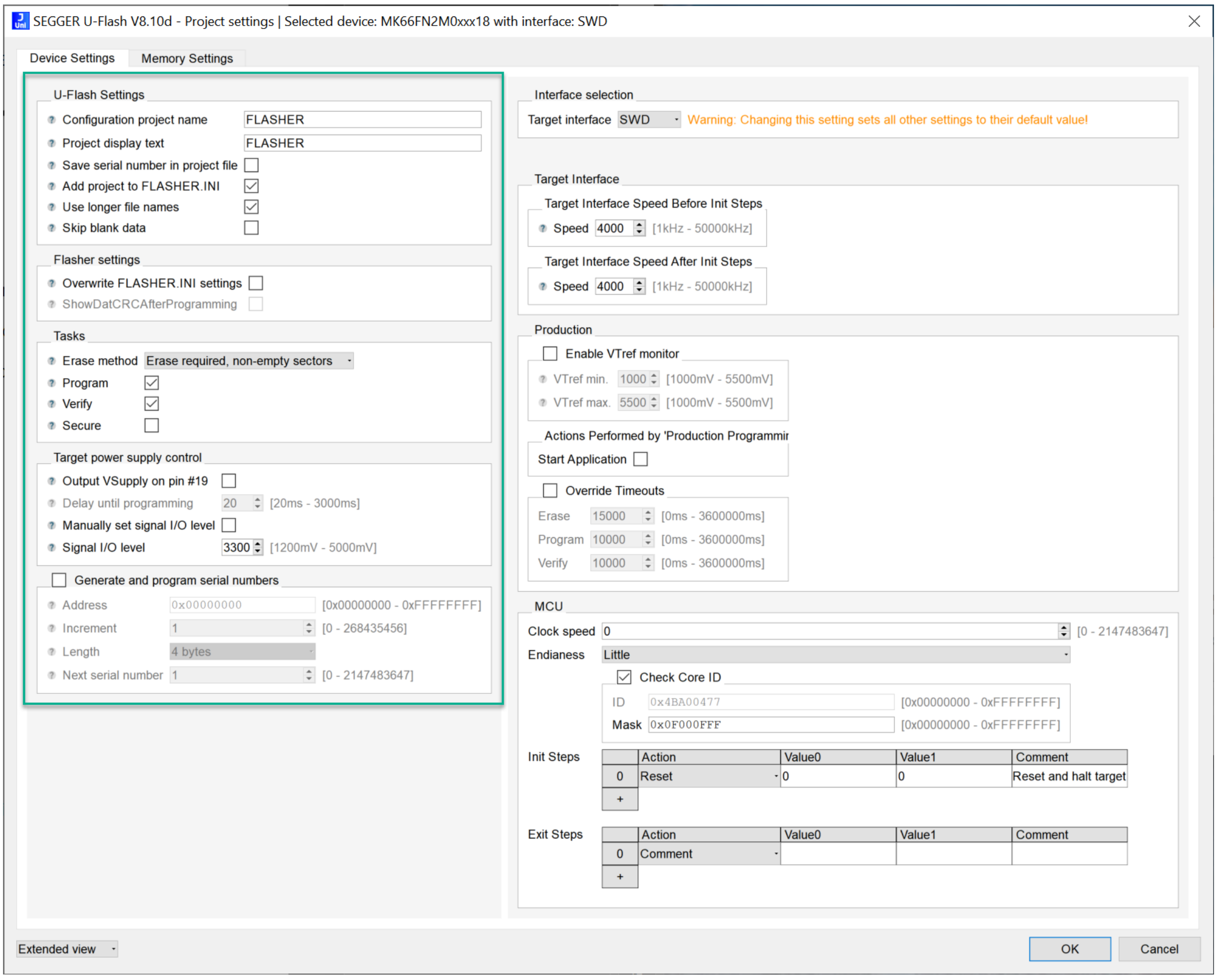
If you selected a binary (*.bin) file, the start address offset can be specified in the corresponding field to the right.


SEGGER U-Flash V8.10d - Disconnected
File Action Configure Help
Single project
Batch project
#1 Open project or Select device
Selected device: MK66FN2M0xxx18
Show wiring
#2 Select data file
P FRDM MCXN947/PreBuild/Binary/FRDM MCXN947 emWin.bin
Offset
#3 Select Flasher
Selected Flasher: None
Connect
#4 Configure project
#5 Save project or Send to Flasher
Console
Device info
Remote Control
Enter a command to send to your Flasher here. Confirm with enter.
SEGGER

```
Parsing device list...
 - Finished parsing device list (21077 device entries)
New Data File selected: C:/Users/     /Documents/     /Eval_SW_Arm/NXP/Coffeemachine/NXP_
```

2.2.4 Configuring common project options

Project settings can be configred in the **Configure** > **Project Settings** dialog, or by clicking the **Configure project** button. The project settings may vary depending on the selected device. You can choose between the **Basic view** (default) and the **Extended view** via the pull-down menu on the bottom left hand corner. The image below shows an example of the project settings dialog in extended view, highlighting the common project options on the left.



SEGGER U-Flash V8.10d - Project settings | Selected device: MK66FN2M0xxx18 with interface: SWD

Device Settings | Memory Settings

U-Flash Settings
- Configuration project name: FLASHER
- Project display text: FLASHER
- Save serial number in project file ☐
- Add project to FLASHER.INI ☑
- Use longer file names ☑
- Skip blank data ☐

Flasher settings
- Overwrite FLASHER.INI settings ☐
- ShowDatCRCAfterProgramming ☐

Tasks
- Erase method: Erase required, non-empty sectors
- Program ☑
- Verify ☑
- Secure ☐

Target power supply control
- Output VSupply on pin #19 ☐
- Delay until programming: 20 [20ms - 3000ms]
- Manually set signal I/O level ☐
- Signal I/O level: 3300 [1200mV - 5000mV]

☐ Generate and program serial numbers
- Address: 0x00000000 [0x00000000 - 0xFFFFFFFF]
- Increment: 1 [0 - 268435456]
- Length: 4 bytes
- Next serial number: 1 [0 - 2147483647]

Interface selection
- Target interface: SWD Warning: Changing this setting sets all other settings to their default value!

Target Interface
- Target Interface Speed Before Init Steps: Speed 4000 [1kHz - 50000kHz]
- Target Interface Speed After Init Steps: Speed 4000 [1kHz - 50000kHz]

Production
- ☐ Enable VTref monitor
  - VTref min. 1000 [1000mV - 5500mV]
  - VTref max. 5500 [1000mV - 5500mV]
- Actions Performed by 'Production Programmir
  - Start Application ☐
- ☐ Override Timeouts
  - Erase 15000 [0ms - 3600000ms]
  - Program 10000 [0ms - 3600000ms]
  - Verify 10000 [0ms - 3600000ms]

MCU
- Clock speed: 0 [0 - 2147483647]
- Endianess: Little
- ☑ Check Core ID
  - ID 0x4BA00477 [0x00000000 - 0xFFFFFFFF]
  - Mask 0x0F000FFF [0x00000000 - 0xFFFFFFFF]

Init Steps

	Action	Value0	Value1	Comment
0	Reset	0	0	Reset and halt target
+				

Exit Steps

	Action	Value0	Value1	Comment
0	Comment			
+				

The available common project settings are:

- U-Flash settings,
- Flasher settings,
- Tasks,
- Target power supply control, and
- Generate and program serial numbers.

Each option will be discussed in more detail below.

2.2.4.1 U-Flash settings

The U-Flash settings allow you to control the following properties:

- **Configuration project name**: This field specifies the name used in the Flasher's file system for the configuration file. The default name is either "FLASHER.UNI" or "FLASHER.CFG", depending on which device has been selected. The ".UNI" / ".CFG" extension will be added automatically. Note: Only the 8.3 file name format is supported, unless the "Use longer file names" option is checked (see below).

- **Project display text**: This field specifies an alternate project name as it will be displayed on the Flasher Portable Plus model. This setting has no effect for all other Flasher models (without display).
- **Save serial number in project file**: This option can be enabled to include the serial number of the selected Flasher unit in the U-Flash project file when it is saved. In this case, the same Flasher unit automatically gets selected again when opening the U-Flash project.
- **Add project to FLASHER.INI**: Creates or updates an entry for this project in the FLASHER.INI initialization file to automatically select this project for invocation with the `#auto` command or to include it in the selection list for the Flasher Portable Plus model.
- **Use longer file names**: This option enables the support of file names with a maximum length of 31 characters (including the file extension). If unchecked, file names are automatically shorthened to fit the 8.3 scheme.
- **Skip blank data**: Enable this option to prevent the creation of data areas for empty sectors in the .DAT file. Caution: Do not use this option with Infineon Aurix (TC2xx) or Aurix-2 (TC3xx) devices.

2.2.4.2 Flasher Settings

- **Overwrite FLASHER.INI settings**: Creates or updates the [CONFIG] section of the FLASHER.INI initialization file with the selected values below.
- **ShowDatCRCAfterProgramming**: Activates displaying the data file CRC after programming.

2.2.4.3 Tasks

The **Tasks** settings define which tasks will be executed during a programming cycle, started e.g. by sending the `#auto` command to the Flasher, or by pressing the PROG button.

- **Erase method**: This setting adds the "Erase" task when invoking the `auto` command and selects the erase method (e.g. bulk erase, erase non-empty sectors, erase required sectors, etc.)
- **Program**: This setting adds the "Program" task when invoking the `auto` command.
- **Verify**: This setting adds the "Verify" task when invoking the `auto` command.
- **Secure**: This setting adds the "Secure" task when invoking the `auto` command in order to perform security measures after programming.

For most devices, the "Erase", "Program", and "Verify" tasks are selected by default. However, the default selection may vary depending on the selected device.

2.2.4.4 Target power supply control

- **Output VSupply on pin #19**: If activated, the Flasher will turn on the target power supply on pin #19 of the Flasher's 20-pin connector during the programming cycle. When powered via USB, the output voltage is ~5V. Note: On Flasher Portable Plus V5 models, the output voltage varies between ~3.5V and ~4.2V, depending on the charge level of the internal battery.
- **Delay until programming**: If the option above is selected, this setting allows to add a 20ms to 3000ms delay between turning on the target power supply and starting a programming cycle.
- **Manually set signal I/O level**: Activate this option to set the nominal HIGH level of the I/O signal pins independently of the voltage at the VTref pin (pin #1) and therefore disable automatic voltage detection. This may be necessary if VTref is not available or a value other than indicated by VTref is required.
- **Signal I/O level**: If the setting above is activated, the entered value is used to set the HIGH level of the signal I/O pins.

2.2.4.5 Generate and program serial numbers

If this setting is activated, various selections can be made to specify the serial numbers that shall be programmed into the target memory during programming cycles.

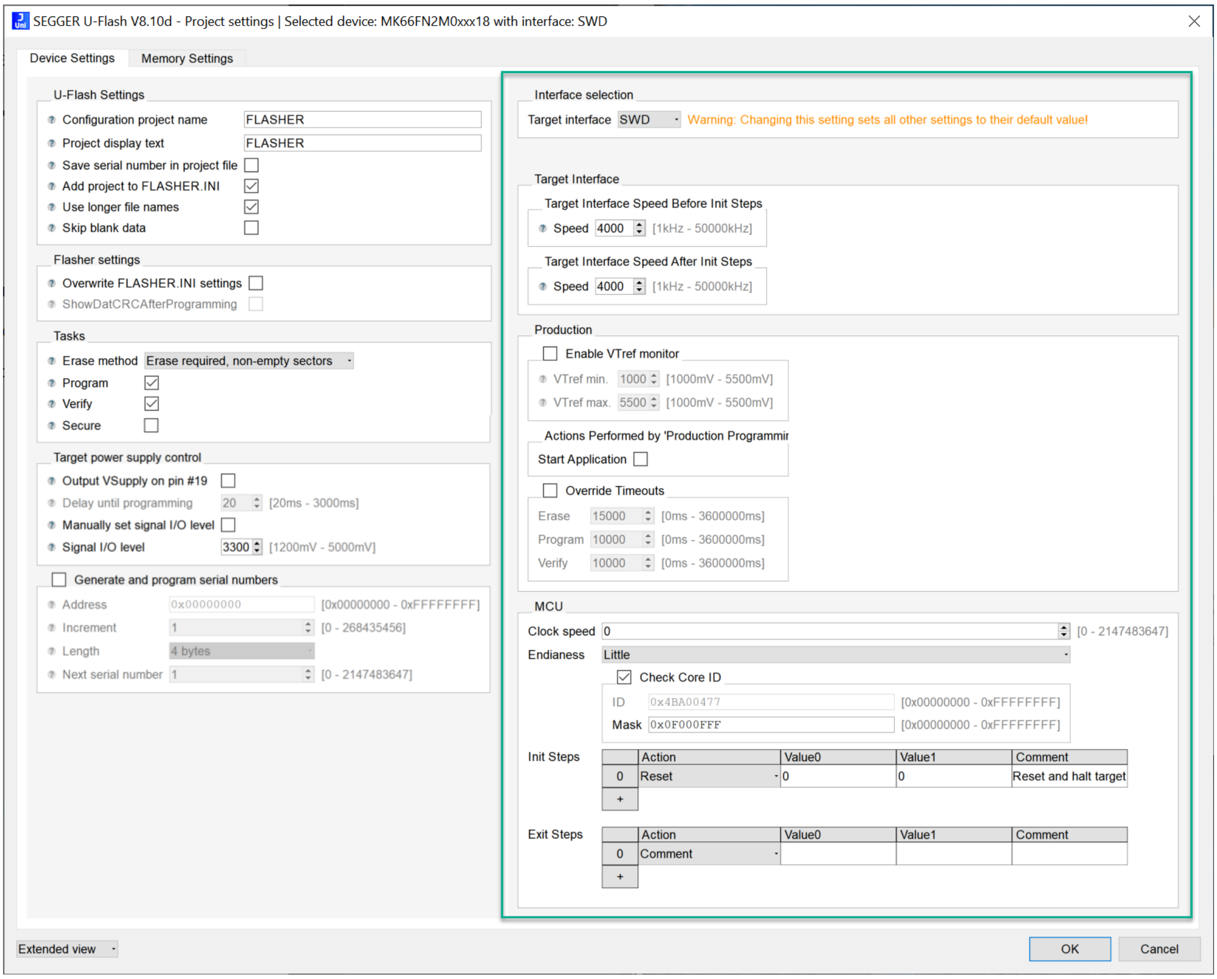
- **Address**: Memory address to which the generated serial number will be written during programming

- **Increment**: Increment from one generated serial number to the next
- **Length**: Defines the length (in bytes) allocated to the generated serial numbers
- **Next serial number**: Value of the next serial number to be generated

Note: On most Microchip PIC devices, the programming of serial numbers using this setting is only possible to a certain extent due to the limited 12-bit / 14-bit opcode size. We recommend to use the Patch file support feature to program serial numbers into these devices (please see Flasher User Guide (UM08022) for details.

2.2.5 Configuring device-specific options

Most devices and projects require some device-specific configurations. These configurations can be made on the right side of the project settings dialog.



SEGGER U-Flash V8.10d - Project settings | Selected device: MK66FN2M0xxx18 with interface: SWD

U-Flash Settings

- Configuration project name: FLASHER
- Project display text: FLASHER
- Save serial number in project file □
- Add project to FLASHER.INI ☑
- Use longer file names ☑
- Skip blank data □

Flasher settings

- Overwrite FLASHER.INI settings □
- ShowDatCRCAfterProgramming □

Tasks

- Erase method: Erase required, non-empty sectors
- Program ☑
- Verify ☑
- Secure □

Target power supply control

- Output VSupply on pin #19 □
- Delay until programming: 20 [20ms - 3000ms]
- Manually set signal I/O level □
- Signal I/O level: 3300 [1200mV - 5000mV]

□ Generate and program serial numbers

- Address: 0x00000000 [0x00000000 - 0xFFFFFFFF]
- Increment: 1 [0 - 268435456]
- Length: 4 bytes
- Next serial number: 1 [0 - 2147483647]

Interface selection

Target interface: SWD Warning: Changing this setting sets all other settings to their default value!

Target Interface

Target Interface Speed Before Init Steps

- Speed: 4000 [1kHz - 50000kHz]

Target Interface Speed After Init Steps

- Speed: 4000 [1kHz - 50000kHz]

Production

□ Enable VTref monitor

- VTref min. 1000 [1000mV - 5500mV]
- VTref max. 5500 [1000mV - 5500mV]

Actions Performed by 'Production Programmir

Start Application □

□ Override Timeouts

- Erase 15000 [0ms - 3600000ms]
- Program 10000 [0ms - 3600000ms]
- Verify 10000 [0ms - 3600000ms]

MCU

Clock speed: 0 [0 - 2147483647]

Endianess: Little

☑ Check Core ID

- ID: 0x4BA00477 [0x00000000 - 0xFFFFFFFF]
- Mask: 0x0F000FFF [0x00000000 - 0xFFFFFFFF]

Init Steps

	Action	Value0	Value1	Comment
0	Reset	0	0	Reset and halt target
+				

Exit Steps

	Action	Value0	Value1	Comment
0	Comment			
+				

Because there are so many device-specific options, it is not practical to attempt to cover them all in this manual. Plus, new devices with device-specific options are added frequently. When presented with these options, please use the tool tips in the device settings dialog to get details on these options.

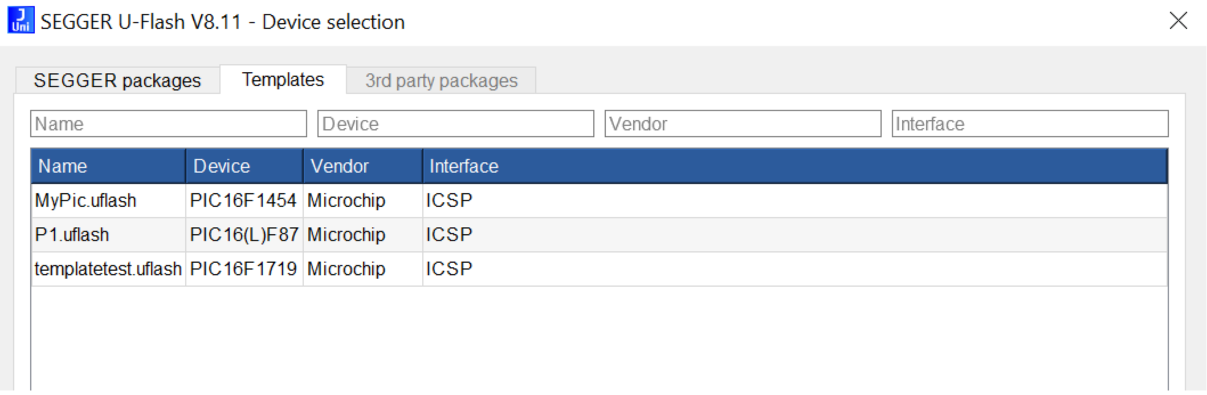
Note: One important setting in the device-specific options worth mentioning here is the **VerboseOutput** setting that is supported for more and more devices in U-Flash. The VerboseOutput setting controls the amount of information that is being output by the Flasher unit during the programming process. The default verbosity level is **none**, but for troubleshooting purposes, the verbosity level can be increased to other options, e.g. **info**, **debug**, etc. The available verbosity levels are device-specific.

2.2.6 Saving the U-Flash project

A new U-Flash project can be saved by selecting **File** > **Save project as** from the menu. The resulting project file has the extension **.uflash**. If you have already saved the project once, you can save the most recent changes with the menu item **File** > **Save project**.

2.2.7 Saving a U-Flash project as a template

A U-Flash project can be saved as a template by selecting **File** > **Save project as template** from the menu. After you save a U-Flash project as a template, you will then be able to select this template project in the device selection dialog from the **Templates** tab (instead of selecting a device only).

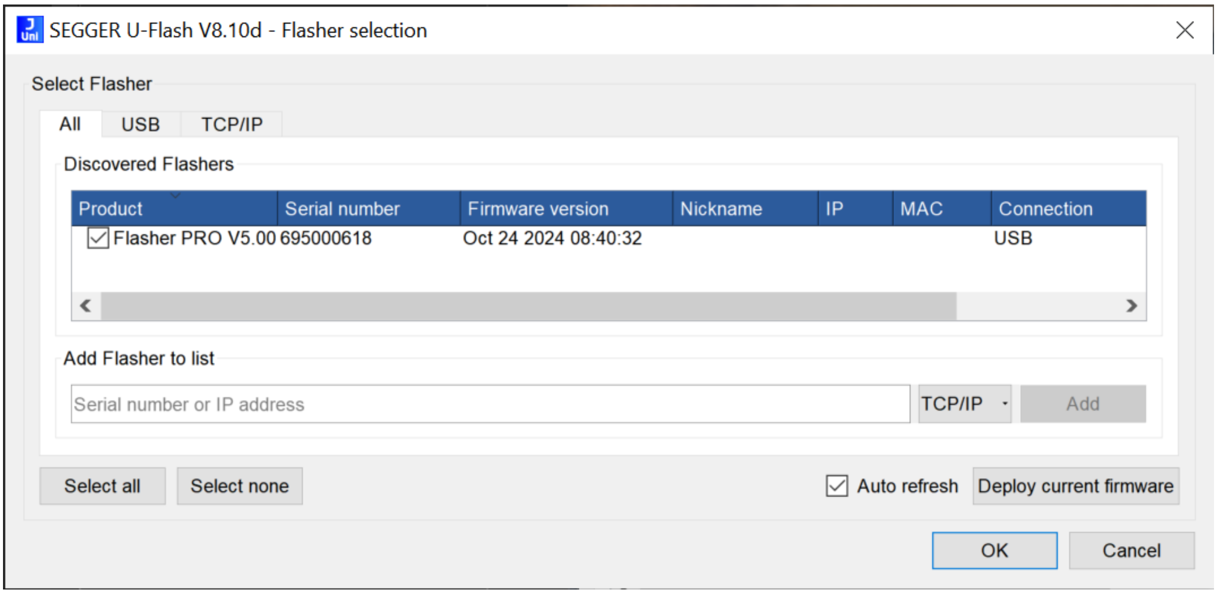


Name	Device	Vendor	Interface
MyPic.uflash	PIC16F1454	Microchip	ICSP
P1.uflash	PIC16(L)F87	Microchip	ICSP
templatetest.uflash	PIC16F1719	Microchip	ICSP

All settings and options in U-Flash will then be populated with the values from the loaded template. This allows to load a certain U-Flash project configuration without having to touch the original U-Flash project.

2.2.8 Selecting a Flasher unit

To select the Flasher unit to be used for programming, choose the menu item **Configure** > **Select Flasher**, or click the **Select Flasher** button. This opens the Flasher selection dialog.



Product	Serial number	Firmware version	Nickname	IP	MAC	Connection
☑ Flasher PRO V5.00	695000618	Oct 24 2024 08:40:32				USB

You can choose to have all discovered Flashers displayed, or just the ones that are connected via USB, or just the ones connected via TCP/IP by selecting the corresponding tab.

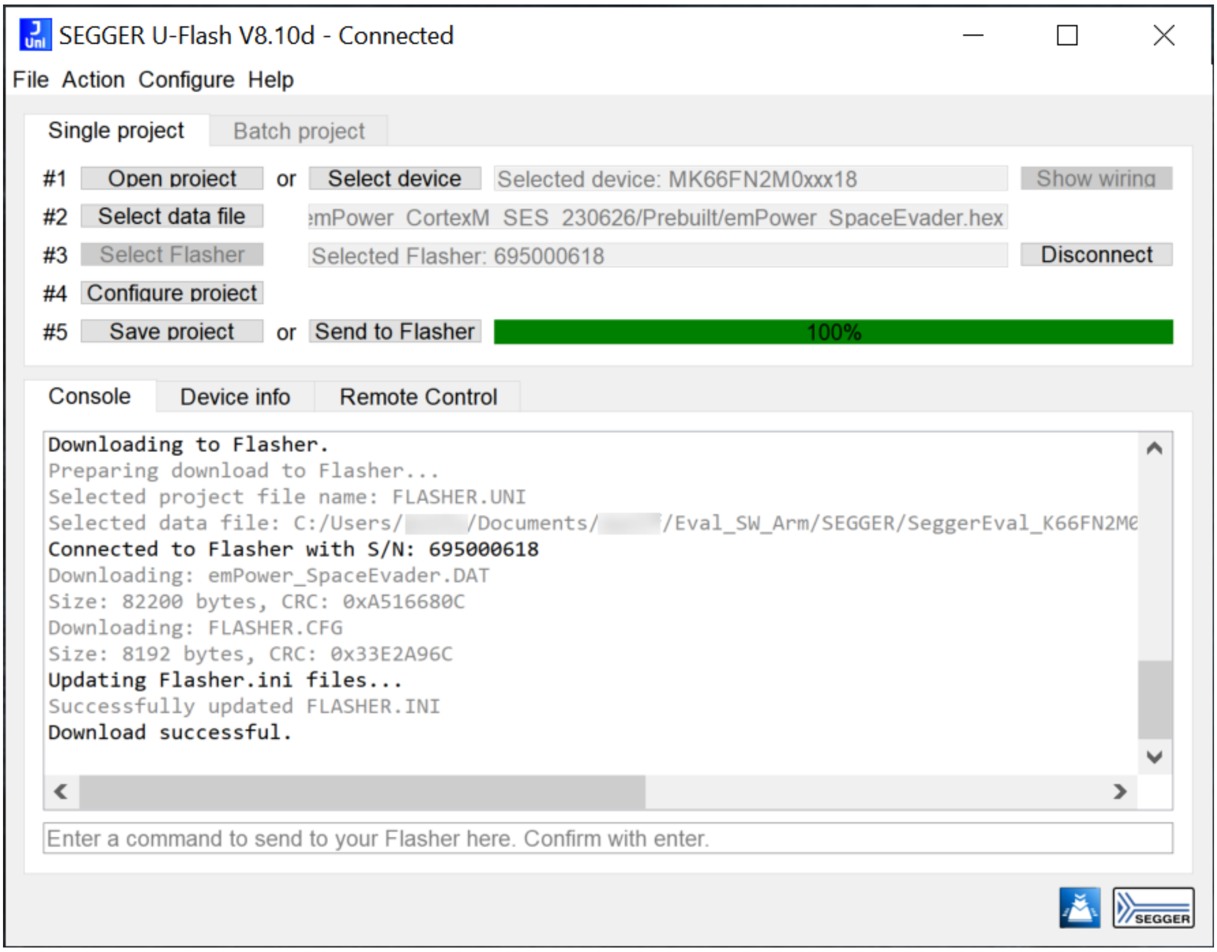
You can also manually add a Flasher unit to the list by providing its serial number or TCP/IP address.

By clicking on **Deploy current firmware**, the firmware of the selected Flasher unit(s) can be updated to the latest version available in the version of U-Flash used.

Note that one instance of U-Flash can only be connected to one specific Flasher unit at any given time. However, it is possible to select multiple Flashers to perform a firmware update. The selected Flashers will be updated sequentially when clicking the **Deploy current firmware** button.

2.2.9 Downloading files to the Flasher

After a Flasher unit has been selected, the project files can be downloaded to the Flasher by choosing the menu item **Action** > **Download configuration**, or by clicking the **Send to Flasher** button.



Selected device: MK66FN2M0xxx18

:mPower_CortexM_SES_230626/Prebuilt/emPower_SpaceEvader.hex

Selected Flasher: 695000618

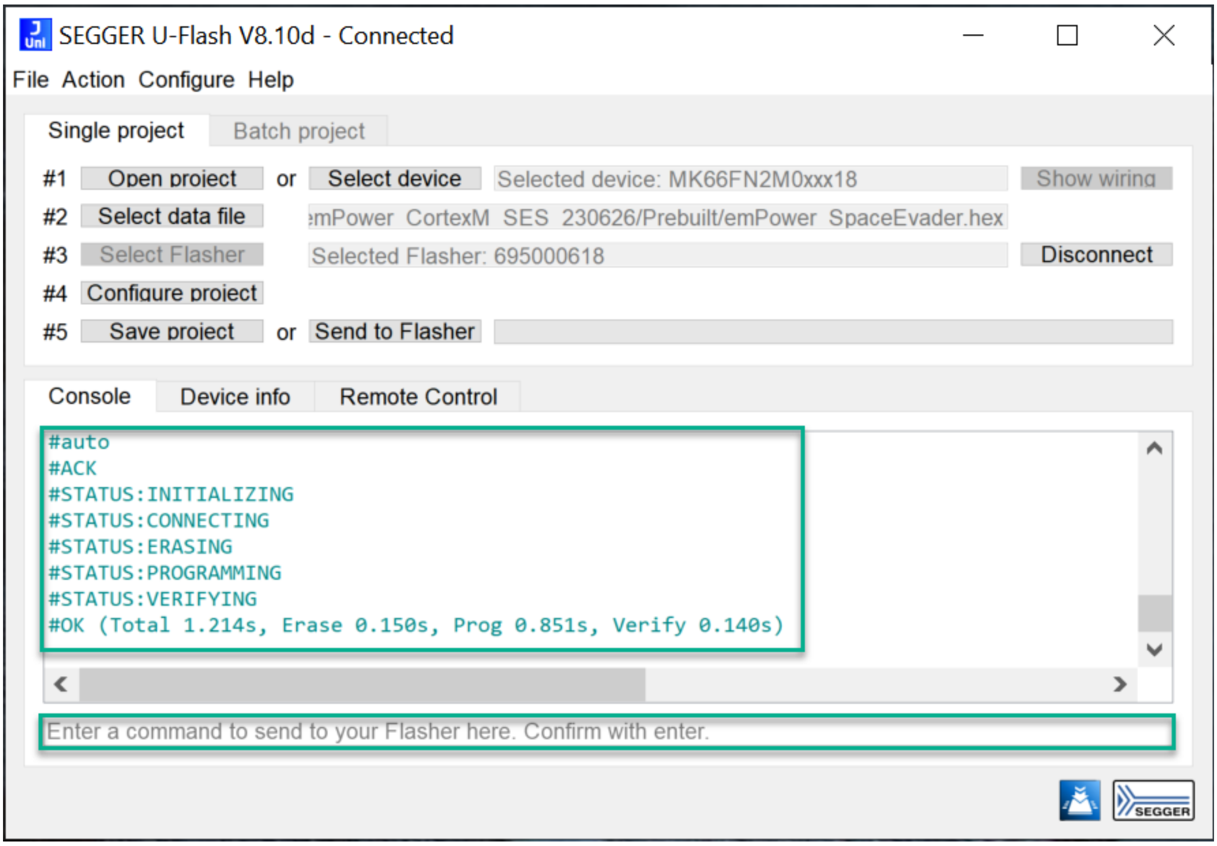
```
Downloading to Flasher.
Preparing download to Flasher...
Selected project file name: FLASHER.UNI
Selected data file: C:/Users/[redacted]/Documents/[redacted]/Eval_SW_Arm/SEGGER/SeggerEval_K66FN2M0
Connected to Flasher with S/N: 695000618
Downloading: emPower_SpaceEvader.DAT
Size: 82200 bytes, CRC: 0xA516680C
Downloading: FLASHER.CFG
Size: 8192 bytes, CRC: 0x33E2A96C
Updating Flasher.ini files...
Successfully updated FLASHER.INI
Download successful.
```

After the project files have been downloaded to the Flasher, you can start a programming cycle by sending the `#auto` command to the Flasher, or by simply pushing the PROG button on the Flasher unit.

2.2.10 Programming the target device via U-Flash

You also have the option to start a programming cycle from within U-Flash.

You can do so by typing the `#auto` command into the command line field below the console and confirming with the **Enter** key.



Selected device: MK66FN2M0xxx18

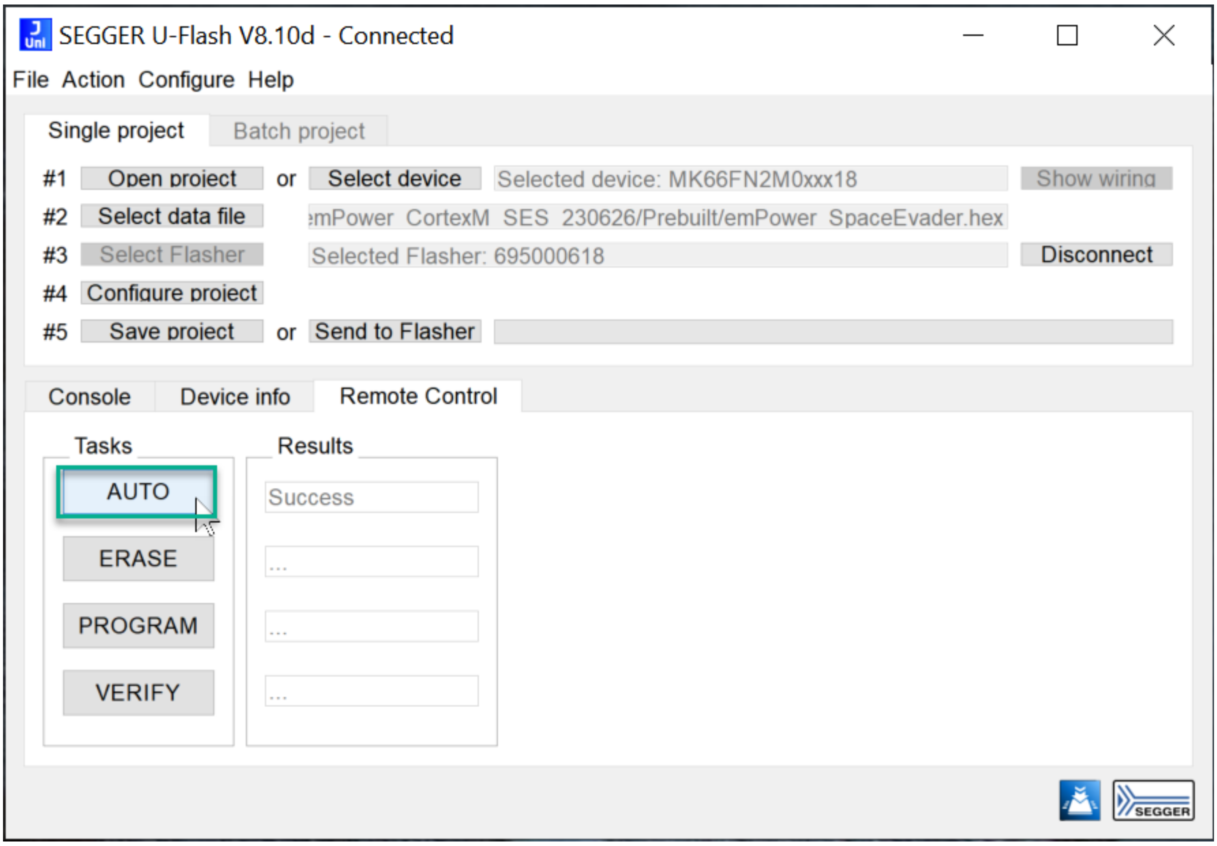
:mPower_CortexM_SES_230626/Prebuilt/emPower_SpaceEvader.hex

Selected Flasher: 695000618

```
#auto
#ACK
#STATUS:INITIALIZING
#STATUS:CONNECTING
#STATUS:ERASING
#STATUS:PROGRAMMING
#STATUS:VERIFYING
#OK (Total 1.214s, Erase 0.150s, Prog 0.851s, Verify 0.140s)
```

Enter a command to send to your Flasher here. Confirm with enter.

Another way to start a programming cycle from within U-Flash is to select the **Remote Control** tab and click on **AUTO**.



Selected device: MK66FN2M0xxx18

:mPower_CortexM_SES_230626/Prebuilt/emPower_SpaceEvader.hex

Selected Flasher: 695000618

Success

...

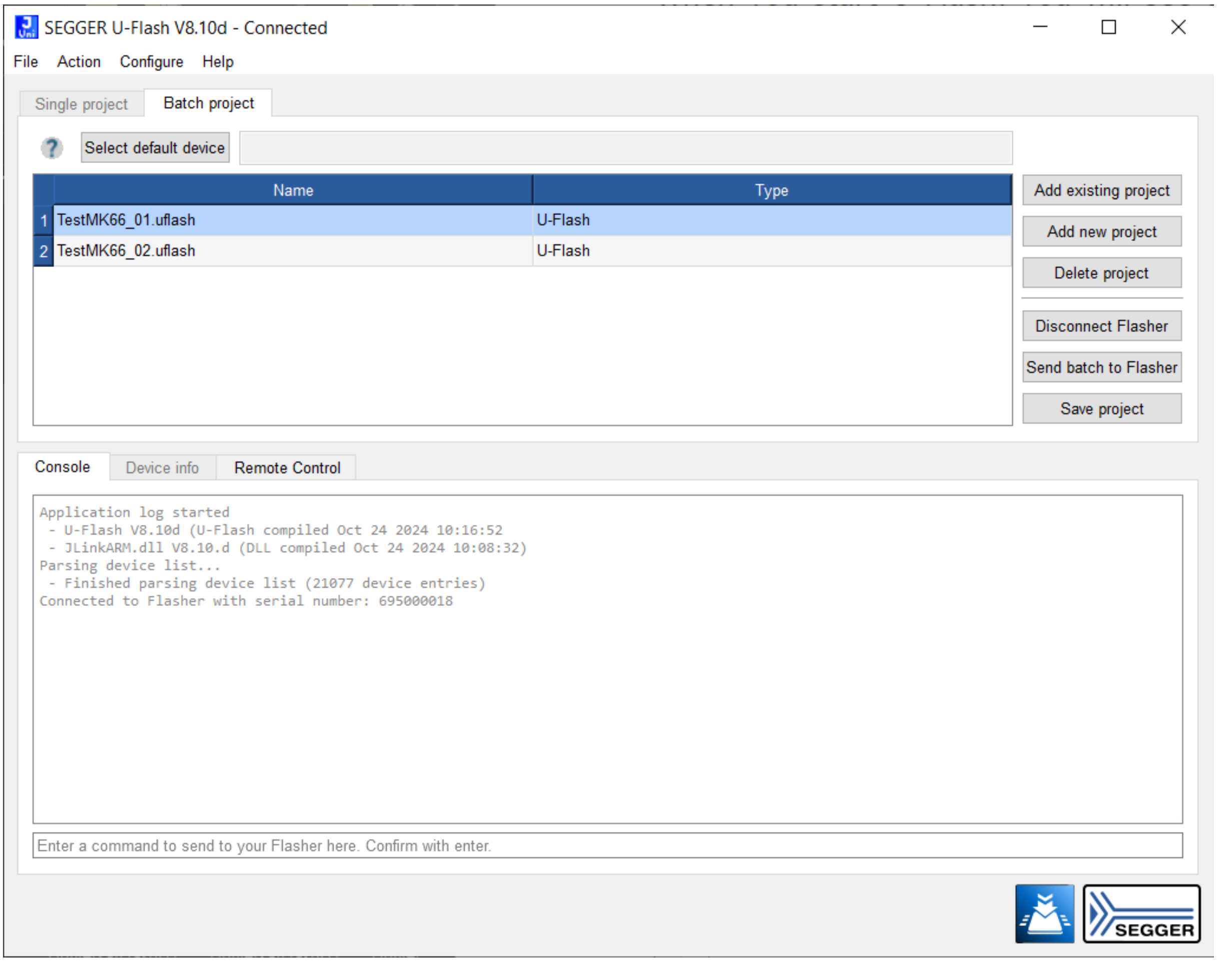
...

...

2.3 Creating a U-Flash batch project

U-Flash also offers the possibility to create a batch project. After starting U-Flash or after selecting **File** > **New project**, you can click the **Batch project** tab to open the batch composer window.

This allows you combine several separate U-Flash projects and download them as a batch project to the Flasher unit to be executed sequentially. You can add existing (single) U-Flash projects to the batch project, as well as create new projects to be added.

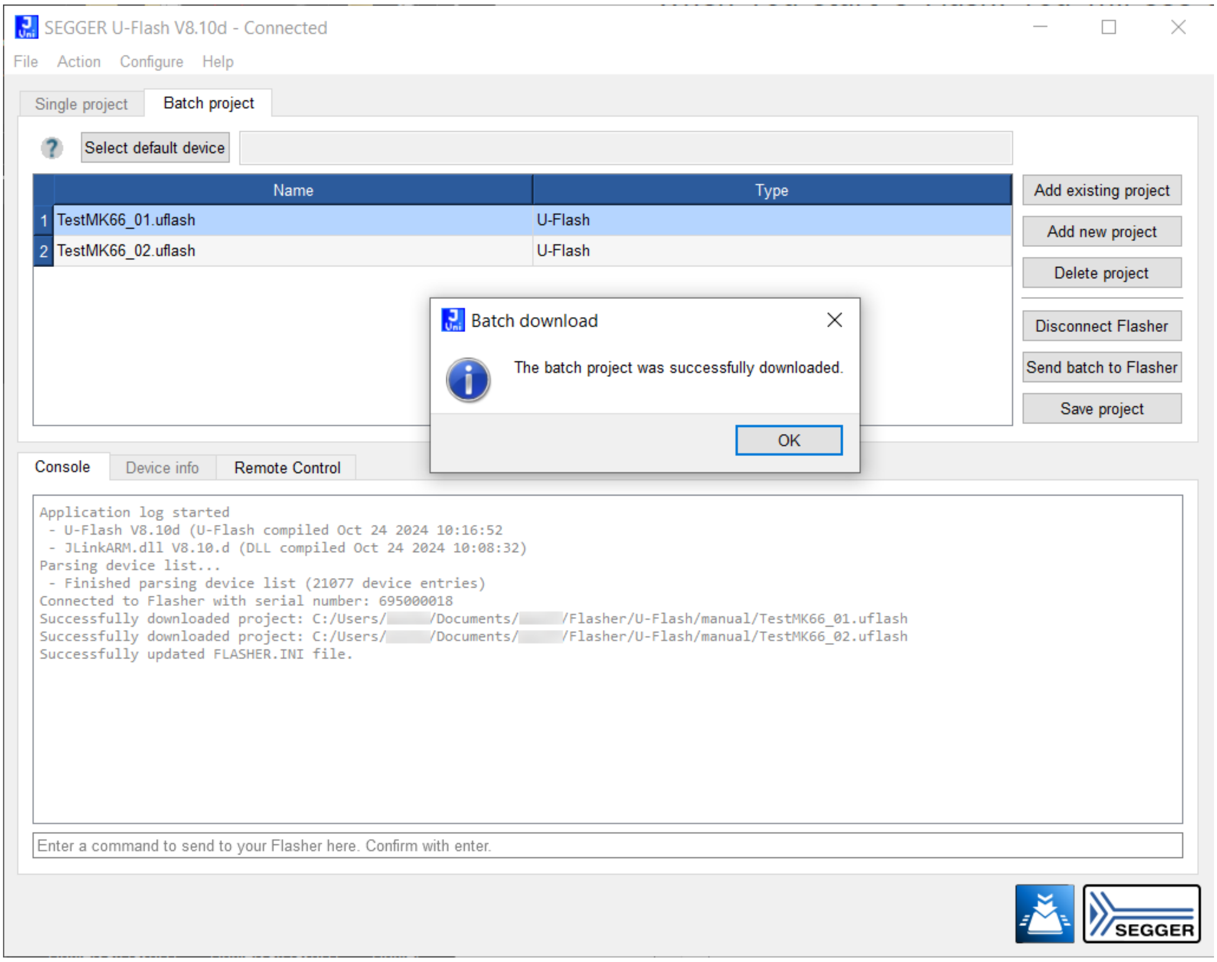


	Name	Type
1	TestMK66_01.uflash	U-Flash
2	TestMK66_02.uflash	U-Flash

```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
Connected to Flasher with serial number: 695000018
```

When creating a new U-Flash project from within the batch composer window, a new U-Flash instance is launched in batch composer mode. You can then configure the project, select a data file, etc. as you would for a single U-Flash project. Make sure that the **Configuration project name** in the U-Flash settings is unique. After you’re done, save the U-Flash project and close / exit the associated U-Flash instance. The newly created single U-Flash project will automatically be added to the batch project in the main U-Flash instance.

Clicking on **Send batch to Flasher** downloads the batch project to the connected Flasher.



	Name	Type
1	TestMK66_01.uflash	U-Flash
2	TestMK66_02.uflash	U-Flash

The batch project was successfully downloaded.

```
Application log started
 - U-Flash V8.10d (U-Flash compiled Oct 24 2024 10:16:52
 - JLinkARM.dll V8.10.d (DLL compiled Oct 24 2024 10:08:32)
Parsing device list...
 - Finished parsing device list (21077 device entries)
Connected to Flasher with serial number: 695000018
Successfully downloaded project: C:/Users/[redacted]/Documents/[redacted]/Flasher/U-Flash/manual/TestMK66_01.uflash
Successfully downloaded project: C:/Users/[redacted]/Documents/[redacted]/Flasher/U-Flash/manual/TestMK66_02.uflash
Successfully updated FLASHER.INI file.
```

Enter a command to send to your Flasher here. Confirm with enter.

Clicking on **Save project** saves the batch project to disk in ".uflash" format for later use.

2.4 Opening an existing U-Flash project

If you would like to open a previously saved U-Flash project (*.uflash), you can do so by selecting **File** > **Open project** from the menu.

Both single and batch U-Flash projects can be opened this way.

2.5 Saving the project-related configuration and data files

The generated configuration and data file for a U-Flash project can be saved to a hard drive, e.g. to store them in a version control system. Selecting **File** > **Save Flasher UNI file** saves the project configuration file. Selecting **File** > **Save Flasher DAT file** saves the project data file.

Overall, you will get:

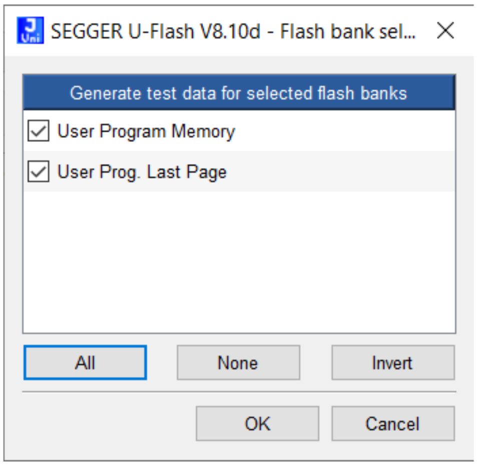
- <my_conf>.CFG or <my_conf>.UNI, depending on the device (contains the project setup),
- <my_data>.DAT (contains the project data to be programmed into the device’s memory),
- a *.pex and/or *.bin file with a device-specific flash loader (optional, depending on the device).

2.6 Generating test data files

U-Flash allows the generation of test data files.

2.6.1 Generating a test data file for writing

The menu item **File** > **Generate test data file** offers the option to generate a data file for writing to the selected chip for testing purposes.



SEGGER U-Flash V8.10d - Flash bank sel...
Generate test data for selected flash banks
☑ User Program Memory
☑ User Prog. Last Page
All
None
Invert
OK
Cancel

This option can be used to try out a programming cycle on a certain device if no “real” data file is available (yet). The user has the option to select the Flash banks to generate test data for. The generated *.MOT file contains random data and can be saved to disk by the user. U-Flash console output:

```
Test data file saved to: C:/…/my_test_data_file.MOT
```

The newly generated file then also becomes the active data file in the U-Flash project so it can be downloaded to the selected Flasher. U-Flash console output:

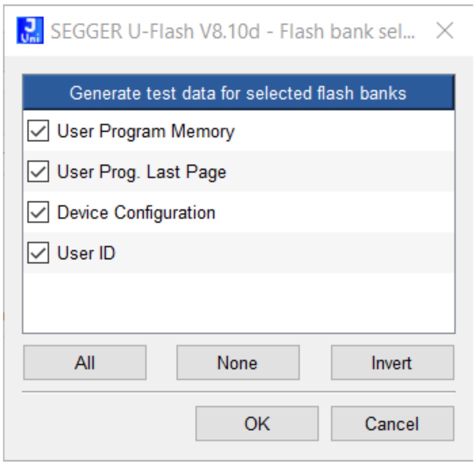
```
New Data File selected: C:/…/my_test_data_file.MOT
```

Please note that certain flash banks on the selected device are not presented to the user as options for test data generation. Examples for this are flash banks that hold device configuration data, or areas that contain OTP memory. If random data were to be written to these areas, the device might be rendered locked or otherwise unusable after a test programming cycle.

Please also note that this feature is not implemented for every device.

2.6.2 Generating a test data file for read

The menu item **File** > **Generate data file for read** offers the option to generate a data file that simulates the reading of the selected chip for testing purposes.



SEGGER U-Flash V8.10d - Flash bank sel...

Generate test data for selected flash banks

☑ User Program Memory

☑ User Prog. Last Page

☑ Device Configuration

☑ User ID

This option can be used to simulate a read cycle on the selected device. The user has the option to select the Flash banks to generate test data for. The generated *.MOT file contains the default values that would be found on a blank device and can be saved to disk by the user. U-Flash console output:

```
Test data file saved to: C:/…/my_read_test_data_file.MOT
```

The newly generated file then also becomes the active data file in the U-Flash project so it can be downloaded to the connected Flasher. U-Flash console output:

```
New Data File selected: C:/…/my_read_test_data_file.MOT
```

Please note that for read test data generation, all flash banks that are available on the selected device are presented to the user as options. But because all memory areas contains the default values that would be found on a blank device, no memory areas are changed in case of accidentally writing this test data file to the device, e.g. by sending an unintended `auto` command or by accidentally pressing the PROG button on the Flasher.

Please note that this feature is not implemented for every device.

2.7 Reading back data from a device

2.7.1 Reading from a device to the console (Action > Read)

The menu item **Action** > **Read** allows reading back the memory contents of a device and displaying it in the U-Flash console in **Motorola S3** format.

Please note that this feature is not implemented on all devices.

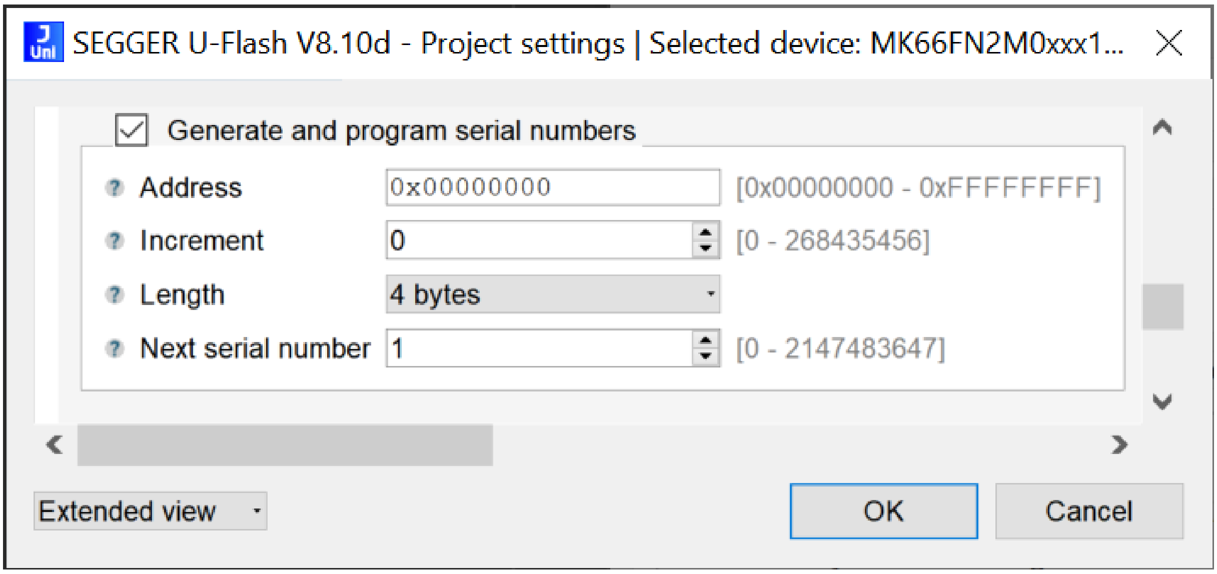
2.7.2 Reading from a device to a file (Action > Read to file)

The menu item **Action** > **Read to file** allows reading back the memory contents of a device and storing it in a .MOT file.

Please note that this feature is not implemented on all devices.

2.8 Serial Number programming

U-Flash can be configured to have the Flasher unit program simple individual serial numbers into the devices. This option can be configured in the corresponding fields in the project settings dialog.



SEGGER U-Flash V8.10d - Project settings | Selected device: MK66FN2M0xxx1...

☑ Generate and program serial numbers

Address	0x00000000	[0x00000000 - 0xFFFFFFFF]
Increment	0	[0 - 268435456]
Length	4 bytes	
Next serial number	1	[0 - 2147483647]

The corresponding options are:

- Generate and program serial numbers,
- Address,
- Increment,
- Length, and
- Next serial number.

Generate and program serial numbers

This checkbox activates or deactivates the serial number programming option.

Address

The address field defines the memory address in the devices where the serial number will be stored.

Increment

The increment defines the value to be added to the current serial number to generate the next serial number after a successful programming cycle.

Length

The length defines how many bytes (1, 2, or 4) the generated serial numbers shall have.

Next serial number

This values defines the serial number to be programmed into the next device. If it is the first device, you could also call this the starting serial number.

Note: On most Microchip PIC devices, the programming of serial numbers using this setting is only possible to a certain extent due to the limited 12-bit / 14-bit opcode size. We recommend to use the Patch file support feature to program serial numbers into these devices (please see Flasher User Guide (UM08022) for details).

2.9 Command line options

U-Flash can be controlled via a command line interface (CLI), e.g. from within scripts, in order to automate the download of projects onto a Flasher that is connected via USB to the host computer.

This chapter lists and describes all available command line options. Some options accept additional parameters which are enclosed in angle brackets, e.g. **<FILENAME>**. If these parameters are optional, they are also enclosed in square brackets, e.g. **[<SADDR>]**. Neither the angle brackets nor the square brackets must be typed on the command line. They are used only to denote (optional) parameters.

It is recommended to always use

```
-openprj<PRJFILENAME> -opendat<FILENAME>[,<SADDR>]
```

to make sure the correct project and data files are opened.

All command line options return 0 if the processing was successful. A return value other than 0 means that an error occurred.

Option	Description
Project:	
-openprj <FILEPATH>	Opens an existing project
-device <DEVICENAME>,<INTERFACE>	Creates a new project and selects the device with the selected target interface.
-saveprj	Saves the currently open project
-saveprjas <FILEPATH>	Saves the currently open project to the selected file path
Data file:	
-opendat <FILEPATH>[,<SADDR>]	Opens an existing data file and sets the offset for .bin files. Supported file types: *.dat, *.elf, *.bin, *.hex, *.mot, *.s, *.s19, *.srec
-savedat <DIRECTORY>	Saves the currently open data file to the given directory as a SEGGER .dat file.
Configuration file:	
-savecfg <FILEPATH>	Saves the configuration of the currently open project as SEGGER .uni or .cfg file depending on the selected device to the selected file path
Flasher:	
-sn <SERIALNUMBER>:<MODULENUMBER1,…>	Connects to the Flasher with the selected serial number
-download	Downloads the currently open project to the selected Flasher. The Flasher has to be selected with -sn first.
Programming *:	
-erase	Erases the target device with the erase method selected in the project settings
-program	Programs the target device with the selected data file
-verify	Verifies the target device’s memory with the selected data file
-read	Reads back the sectors defined in the selected data file from the target device

Option	Description
-auto	Executes the tasks selected in the project settings
Help:	
-h -help --help	Displays the available commands
Miscellaneous:	
-exit	Prevents showing the GUI and terminates the application automatically

*: A Flasher has to be selected with the -sn option and the correct project must be selected/downloaded before using these options. Only one of these commands can be used at a time.

2.10 Creating a support package

U-Flash provides the option to create a support package that can be attached to a SEGGER tech support ticket. The support package includes everything required by SEGGER’s engineering staff to do an analysis of a support case involving U-Flash. In order to generate a support package, go to **Help** > **Create support package**. Please note that, for the protection of your IP, all custom data in the data file is being replaced with `0xFF`. In most cases, this will still allow analysis and resolution of a submitted support ticket.

Chapter 3

Generating secure target binary packages for Flasher Secure

3.1 Introduction

Flasher Secure is a high-volume production programming system from SEGGER, capable of protecting a company’s IP regardless of the production site. It provides full control over the programming process at CMs and similar environments. For more information about Flasher Secure, please visit *https://www.segger.com/products/production/flasher/models/flasher-secure/*.

As part of the Flasher Secure system, secure target binary packages (which include the device-specific Target Encryption Link Package (TELP)) need to be created using U-Flash. This chapter describes this process.

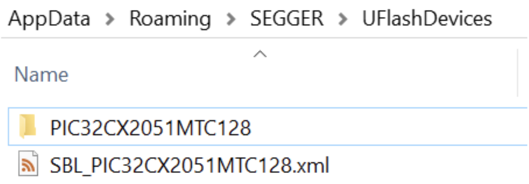
3.2 Adding TELPs to U-Flash

As a TELP is a commercial product requiring its own license, U-Flash (which is a free utility) does not include any TELPs by default. Therefore, after a device-specific TELP has been purchased and received, it needs to be added to U-Flash. Before doing so, please download the latest Flasher Software and Documentation Pack from the SEGGER website and install it (https://www.segger.com/downloads/flasher).

TELPs are delivered in a zip archive containing an xml file and a directory containing all other data. To install the TELP, the content of the zip archive needs to be copied to the **UFlashDevices** directory:

OS	Location
Windows	C:\Users\<USER>\AppData\Roaming\SEGGER\UFlashDevices
Linux	$HOME/.config/SEGGER/UFlashDevices
macOS	$HOME/Library/Application Support/SEGGER/UFlashDevices

On Windows, the installed TELP for the PIC32CX2051MTC128 device would look like this:


AppData > Roaming > SEGGER > UFlashDevices
Name
PIC32CX2051MTC128
SBL_PIC32CX2051MTC128.xml

To verify correct installation, open U-Flash, click “select device”, and search for the device name you just installed the TELP for. After you select the device, “SBL” (SEGGER Secure Boot Loader) will be shown in the interface selector.

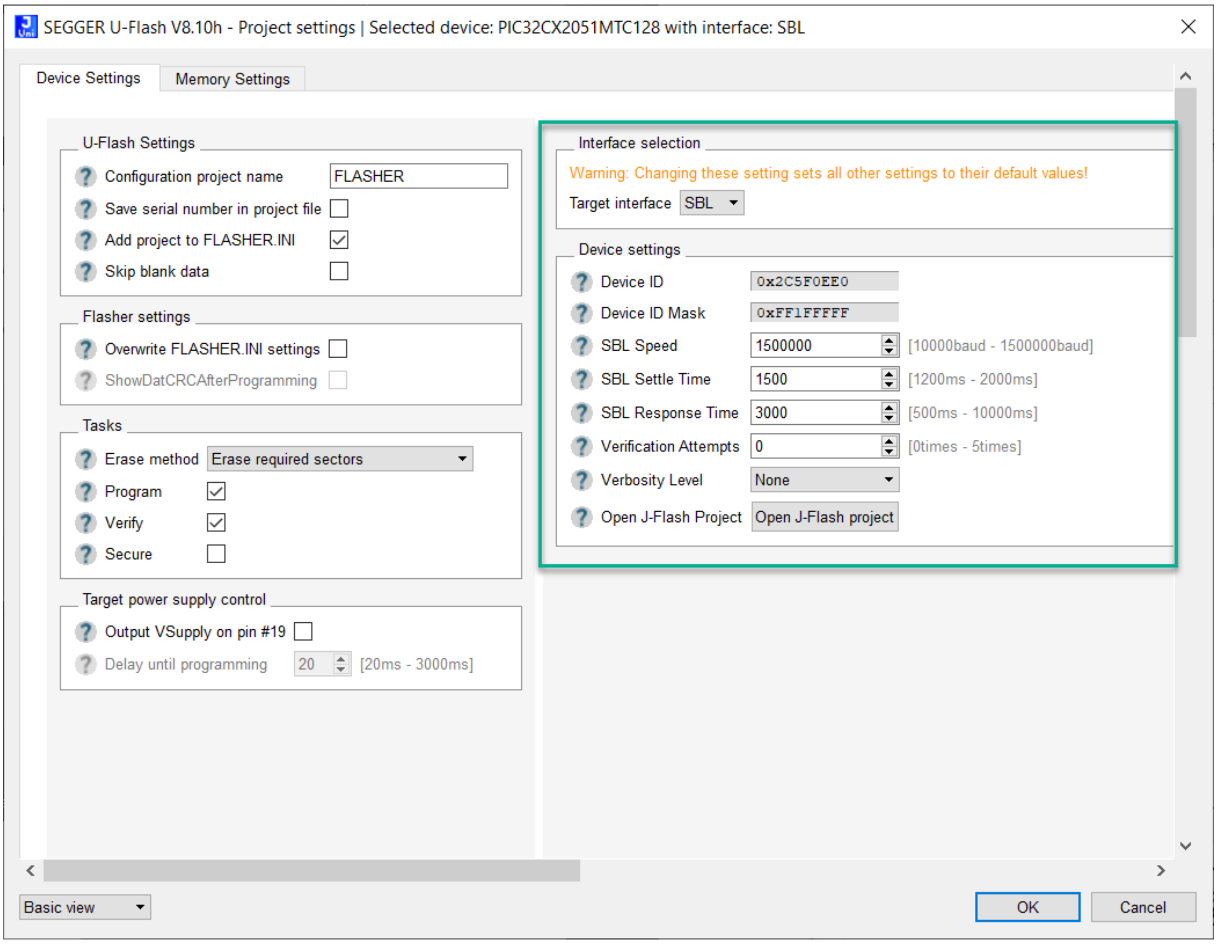
The process is the same when adding more TELPs for other devices. Just copy the additional xml files and folders to the UFlashDevices directory.

Note: This process applies to TELPs delivered after December 2024. Older TELPs had to be installed into the Flasher SW Pack installation directory.

3.3 Creating a secure target binary packages for Flasher Secure

Once you have installed the TELP, you'll be able to create a U-Flash project based on the device that uses "SBL" as the interface. Open U-Flash, click "select device", and search for a device for which a TELP has been installed. After you select the device, "SBL" (SEGGER Secure Boot Loader) will be shown in the interface selector.

In the Device Settings dialog, you will see some additional options to configure the SBL and its communication with the Flasher Secure programmer.



SEGGER U-Flash V8.10h - Project settings | Selected device: PIC32CX2051MTC128 with interface: SBL

U-Flash Settings
- Configuration project name: FLASHER
- Save serial number in project file □
- Add project to FLASHER.INI ☑
- Skip blank data □

Flasher settings
- Overwrite FLASHER.INI settings □
- ShowDatCRCAfterProgramming □

Tasks
- Erase method: Erase required sectors
- Program ☑
- Verify ☑
- Secure □

Target power supply control
- Output VSupply on pin #19 □
- Delay until programming: 20 [20ms - 3000ms]

Interface selection

Warning: Changing these setting sets all other settings to their default values!

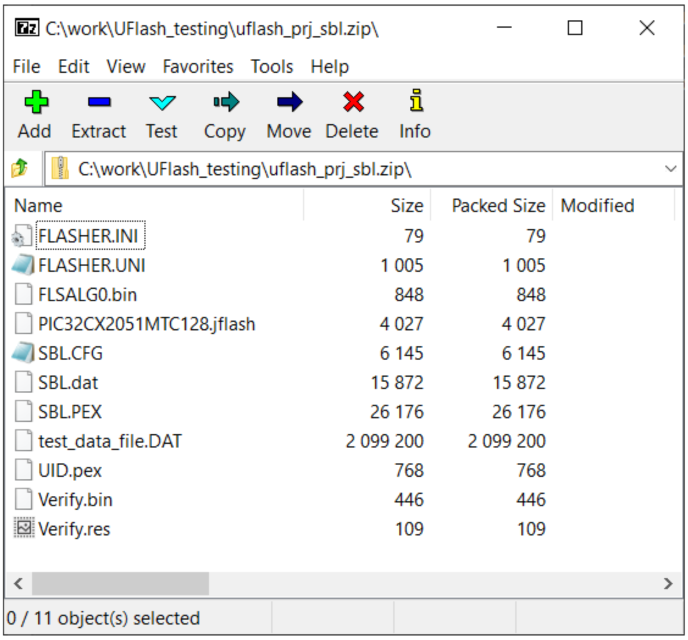
- Target interface: SBL

Device settings
- Device ID: 0x2C5F0EE0
- Device ID Mask: 0xFF1FFFFF
- SBL Speed: 1500000 [10000baud - 1500000baud]
- SBL Settle Time: 1500 [1200ms - 2000ms]
- SBL Response Time: 3000 [500ms - 10000ms]
- Verification Attempts: 0 [0times - 5times]
- Verbosity Level: None
- Open J-Flash Project: Open J-Flash project

Please click on the question marks in the settings dialog for additional information about each setting.

You can now add your data file as usual. Then save the U-Flash project. The secure target binary package (a non-compressed .zip file) will be created in the process (in the same directory as the U-Flash project). This is the secure target binary package that you can then upload to one of your projects in the Flasher Secure Server. For more information on this, please refer to the Flasher Secure User Guide (available on request).

Content of an example package:



C:\work\UFlash_testing\uflash_prj_sbl.zip\

Name	Size	Packed Size	Modified
FLASHER.INI	79	79	
FLASHER.UNI	1 005	1 005	
FLSALG0.bin	848	848	
PIC32CX2051MTC128.jflash	4 027	4 027	
SBL.CFG	6 145	6 145	
SBL.dat	15 872	15 872	
SBL.PEX	26 176	26 176	
test_data_file.DAT	2 099 200	2 099 200	
UID.pex	768	768	
Verify.bin	446	446	
Verify.res	109	109	

0 / 11 object(s) selected

Chapter 4

Literature and references

This chapter lists documents that may be useful to gain a deeper understanding of technical details.

Title	Comments
Flasher User Guide	This online document describes the SEGGER Flasher programmers. It is available at *https://wiki.segger.com/UM08022_Flasher*.
J-Flash User Guide	This document describes the SEGGER J-Flash utility. It is available at *https://www.segger.com/downloads/flasher/UM08003*.