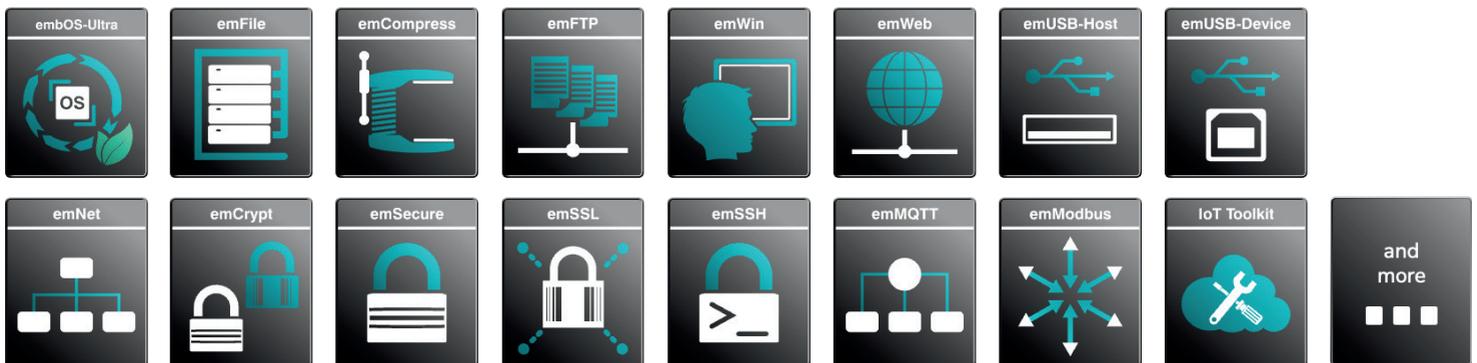


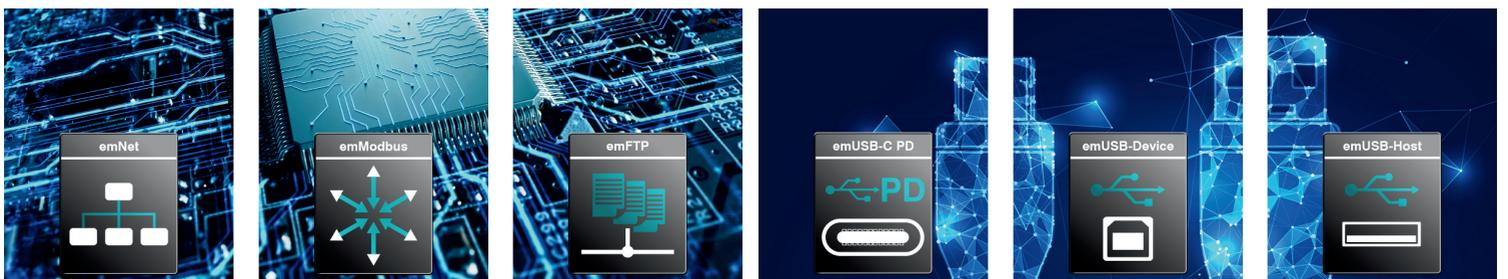


Create with Segger embOS, emPowerOS

Create the firmware structure using SEGGER's robust, reliable, and highly efficient software libraries. SEGGER provides a complete operating system for deeply embedded systems, offers a ground-breaking RTOS, and enables connectivity via IP, USB, and other interfaces. SEGGER's embedded software libraries also include a file system and compression and security algorithms and protocols.



Reliable communication protocols, USB Support and more



The IP stack for embedded devices

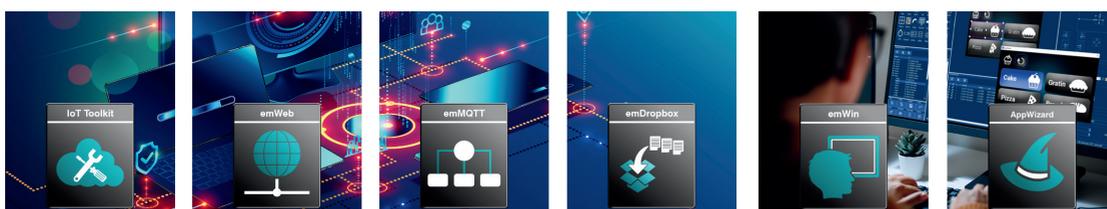
Adding industry-standard protocols

Efficient file transfer for IoT devices

USB-C power-delivery management

Boosting connectivity

IoT device enhancement



The IoT library collection

The web server

Full support for IoT protocols

Dropbox access

The embedded graphics library

The GUI design tool

➤ Reach out to us at segger@gsasmspl.com

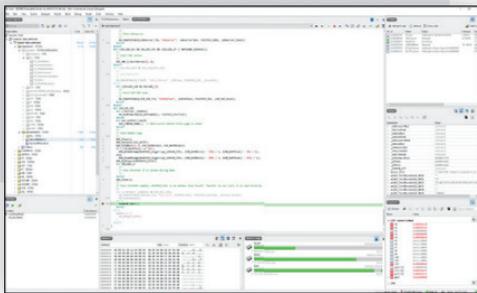
SEGGER Embedded Studio

Embedded Studio is an all-in-one solution for managing, building, testing, and deploying embedded applications.

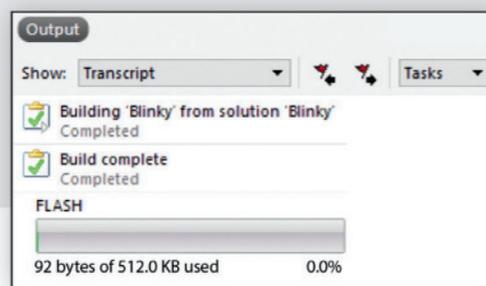
- Develop for any ARM or RISC-V-based microcontroller.
- A cross-platform IDE is available for Windows, macOS, and Linux.
- Powerful project management and build system.
- Outstanding code size and execution speed, using SEGGER's own toolchains or GCC.
- Advanced debugging and system analysis.
- Seamless integration with SEGGER J-Link debug probes.
- Pre-configured project templates for popular microcontrollers.



Code fast



Build small



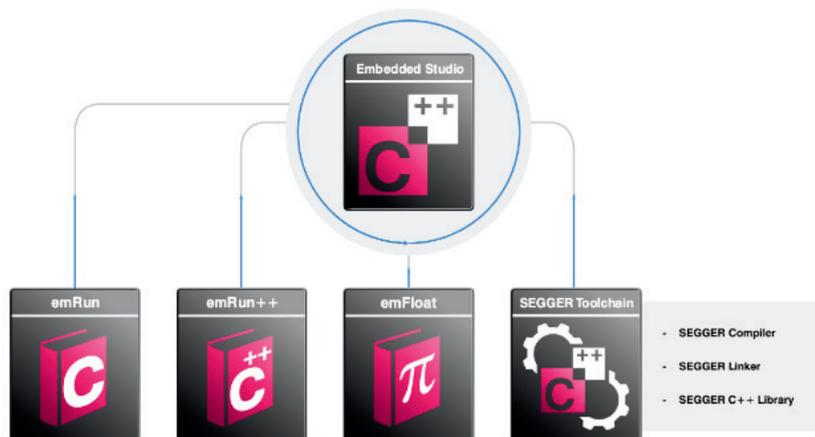
Use across platforms

arm

Built-in Toolchains

Embedded Studio comes with two toolchains: SEGGER and GCC, both of which are ready-to-run. External toolchains like LLVM, IAR, or the ARM/Keil compiler can also be used. All toolchains under one proverbial roof means maximum flexibility.

A complete, intuitive, and fast IDE



SEGGER Compiler



SEGGER Linker



SEGGER C++ library

➤ Reach out to us at segger@gsasmspl.com

SEGGER Debug & Trace Probes

Debug the application using SEGGER's market-leading debug and trace probes. The debugging process is a critical part of the build process. SEGGER's probes help optimize the application in preparation for the final stages of development. Crucial here are accurate conclusions from reliable probes and tools during intermittent test runs.



Debug using J-Link



J-Link PRO PoE

J-Link PRO

J-Link ULTRA+

J-Link PLUS

J-Link BASE

J-Link WiFi

	J-Link PRO PoE	J-Link PRO	J-Link ULTRA+	J-Link PLUS	J-Link BASE	J-Link WiFi
Ethernet	PoE	✓	✗	✗	✗	Wireless host
J-Flash	✓	✓	✓	✓	✗	✓
Ozone	✓	✓	✓	✓	✗	✓
Monitor Mode	✓	✓	✓	✓	✗	✓
Flash Breakpoints	∞	∞	∞	∞	✓	∞
Real-Time Transfer (RTT)	✓	✓	✓	✓	✓	✓
Mbyte/s	4.0	4.0	4.0	1.0	1.0	1.0

Ozone Debugger



J-Trace Pro overview

The J-Link debugger and performance analyzer

- Stand-alone graphical debugger
- Debugging of any toolchain and IDE1 output
- C/C++ source-level debugging and assembly instruction debugging
- Debugging of information windows for all purposes: disassembly, memory, globals and locals, (live) watches, and CPU and peripheral registers
- Source editor for fixing bugs immediately
- High-speed programming of an application into a target
- Direct use of built-in J-Link features (Unlimited Flash Breakpoints, Flash Download, Real Time Terminal, Instruction Trace)
- Scriptable project files for automatic setup
- New project wizard to ease basic configuration for new projects
- Free tool,* meaning no licensing costs and no hidden fees



	J-Trace PRO	J-Trace PRO Cortex	J-Trace PRO Cortex-M	J-Trace PRO RISC-V
Arm Cortex-A/R	✓	✓	✗	✗
Arm Cortex-M	✓	✓	✓	✗
RISC-V	✓	✗	✗	✓

- | | |
|---|--|
| <ul style="list-style-type: none"> ✓ Streaming trace ✓ Endless trace buffer ✓ Live code coverage ✓ Live code profiling ✓ Gbit-Ethernet ✓ SuperSpeed USB | <ul style="list-style-type: none"> ✓ J-Flash ∞ Flash breakpoints ✓ Ozone ✓ Monitor Mode ✓ Real-time transfer (RTT) ✓ 4.0 Mbyte/s |
|---|--|

Segger Flasher Series

Program the application into the final product using SEGGER's ultra-reliable line of Flasher programmers. The programming process is important in many project phases. During verification, programming is used to transfer an application to the target hardware, as well as for testing afterwards. Once an application is finalized, it is deployed to the intended hardware; together, they become the final product. In a service situation, a final product may need a reflash to ensure continued faultless operation or to revive a dead product.

	 Flasher Pro XL	 Flasher Pro	 Flasher Portable PLUS	 Flasher ARM	 Flasher Secure	 Flasher Compact
Extra-large memory	✓	✗	✗	✗	✗	✗
Ultra speed interface	✓	✓	✓	✓	✓	✓
J-Flash	✓	✓	✓	✓	✓	✓
Targets supported	∞	∞	∞	Arm	∞	∞
Authenticated flashing	✓	✓	✓	✓	✓	✓
USB powered	✓	✓	✓	✓	✓	✓
Remote control connector	✓	✓	✗	✓	✓	✗
Secure programming	✗	✗	✗	✗	✓	✗
Gang programming via hub	✗	✗	✗	✗	✗	✓
Display-supported UI	✗	✗	✓	✗	✗	✗

Flasher - fast and reliable parallel programming



- ✓ Parallel programming
- ✓ Up to 4 direct channels
- ✓ Up to 24 channels via Hubs
- ✓ High-speed programming
- ✓ Web-interface
- ✓ USB-C infrastructure



- ✓ Parallel programming
- ✓ Up to 12 direct channels
- ✓ Up to 24 channels via Hubs
- ✓ High-speed programming
- ✓ Web-interface
- ✓ USB-C infrastructure

