

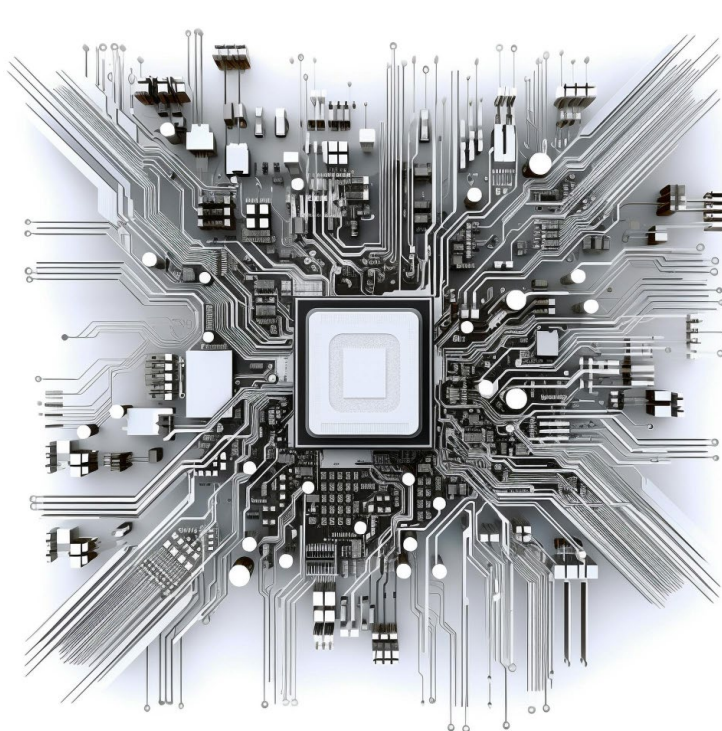


未対応開発環境サポート 独自デバイス対応・フラッシュローダ

J-Link / Flasherユーザカスタム対応

本書記載内容は最新ハードウェアバージョンの内容になりますので、既にお持ちのバージョンが古い場合
記載内容と異なる場合もございます。

J-Link / Flasherはこのようなニーズに独自対応できます。



- SEGGER社未対応のIDE,ユーザ独自開発ツールへのJ-Linkカスタム対応
- SEGGER社で対応していないCPU内蔵フラッシュにユーザ独自対応
- 外部(Q)SPIフラッシュ接続が未対応の場合の対応
- CPUと外部フラッシュの接続ピン配置をユーザ独自の接続に対応
- フラッシュの書込みアルゴリズム、書込みメモリエリアをカスタマイズ

J-Link / Flasherのユーザ独自対応フレームワークライセンス



ライセンス製品	J-Link SDK	J-Link DSK	Flasher DSK
カスタマイズ対象	J-Link PCドライバ	ターゲットフラッシュローダ	ターゲットフラッシュローダ
対応製品（推奨製品）	J-Link	J-Link・Flasher	Flasher PRO
概要	ユーザ開発環境・未対応IDEにJ-Link.dllネイティブ対応	未対応デバイス、対応済みデバイスのフラッシュローダをサンプルとして、カスタマイズ開発対応	幅広いコア対応で未対応デバイスへのカスタマイズ対応
未対応IDE・開発ツール対応	○	×	×
未対応デバイス対応 (Arm/Cortex/RX/RISC-V)	×	○	○
未対応デバイス対応 (Arm/Cortex/RX/RISC-Vコア以外)	×	×	○

未対応IDE、ユーザ独自開発ツールにJ-Linkをネイティブ対応



J-Link SDKはWindows / Linux / macOS 32bit / 64bit でほぼすべてのプログラミング言語で開発可能なフレームワークで、使いやすい C 言語 API 関数を提供します。C (Visual C++ 6 および Visual Studio 2010)、VB6 (Microsoft Visual Basic)、および VB.NET (Visual Studio 2010)、LabViewのスタートアップサンプルプロジェクトが付属

J-Linkとの接続は、J-Link標準ドライバ「JLINK.DLL」を用いて対応するため、J-Linkのソフトウェアバージョンアップに合わせて新デバイス対応などを行うことができます。

```
C:\Windows\py.exe
Enter the host interface.
0> USB (Default)
1> IP
HostIf> 0
Enter J-Link serial number
SN> 679000000
Please specify device / core. <Default>: unspecified
Type '?' for selection dialog
Device> MK66FN2M0xxx18
Please specify target interface:
0> JTAG
1> SMD (Default)
TIF> 1
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed> 4000
Connected to target successfully.
To close J-Link press <Enter>.
```

サンプルPythonスクリプト

```
Select C:\WINDOWS\system32\cmd.exe
SEGGER J-Link Commander V6.45e (Compiled Apr 9 2019 13:51:32)
DLL version V6.45e, compiled Apr 9 2019 13:51:07

Connecting to J-Link via USB...O.K.
Firmware: J-Link Pro V4 compiled Mar 21 2019 15:44:44
Hardware version: V4.00
S/N: 174402383
License(s): RDI, FlashBP, FlashDL, JFlash, GDB
IP-Addr: DHCP (no addr. received yet)
VRef=3.311V
Device "MK66FN2M0XXX18" selected.

Connecting to target via JTAG
InitTarget()
TotalIRLen = 4, IRPrint = 0x01
JTAG chain detection found 1 devices:
#0 Id: 0x4BA00477, IRLen: 04, CoreSight JTAG-DP
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
AP[1]: JTAG-AP (IDR: 0x001C0000)
Iterating through AP map to find AHB-AP to use
AP[0]: Core Found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241, Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E000, PID: 0000B00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E000, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E000, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E000, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B1059000, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: B1059000, PID: 000BB925 ETM
ROMTbl[0][6]: E0042000, CID: B1059000, PID: 003BB907 ETB
ROMTbl[0][7]: E0043000, CID: B1059000, PID: 001BB908 CSTF
Cortex-M4 Identified.
J-Link
```

サンプルCコードプロジェクト

J-Link Startup Sequence dialog box. It contains fields for J-Link Connection (USB Serial number: 174402383), Target Device (MK66FN2M0xxx18), Target Connection (Target interface: SWD, Interface speed [kHz]: 4000), and J-Link Settings File (Settings.jlink). A 'Connect' button is at the bottom right. A log window at the bottom shows the connection process.

サンプルVB / VB .NETアプリケーション

提供物一覧



File	Content
JLinkARMDLL.h	Default J-Link DLL header file. Contains function declarations, defines and typedefs used in the DLL. Must be included to use the DLL functions.
JLink.h	Fallback J-Link DLL header file. Like JLinkARMDLL.h, but with stdcall function declarations instead of C declaration. Must be used for applications which cannot use C declaration functions.
JLink.lib	J-Link import library to load the J-Link DLL functions. Searches for J-Link DLL in installed J-Link software if the DLL is not in executable directory.
JLinkARM.lib	J-Link import library to load the J-Link DLL functions. Searches for J-Link DLL in executable directory only. Should not be used when updating the DLL shall be possible.
JLinkARM.dll	The J-Link DLL itself to be copied into the application's executable directory.
UM08002_JLinkDLL.pdf	J-Link SDK documentation.
Release.html	J-Link SDK release notes.

サンプルプロジェクト

Project / Script	Target platform	Description
Samples_Startup_Python.py	Windows/ Linux/ macOS, Python 3.5.2	Python script showing how to call J-Link DLL functions and demonstrating the startup sequence.
Windows¥C¥ Start_JLink_VS2010.sln	Windows, Visual Studio 2010, C code	Visual Studio 2010 solution including example projects for the J-Link DLL startup sequence and flash download via J-Link.
Windows¥C¥ Start_JLink_VC6.dsw	Windows, Visual C++ 6.0, C code	Visual C++ 6 workspace, including example projects for the J-Link DLL startup sequence, flash download via J-Link, reading the target's JTAG ID, hardware testing, instruction tracing and the J-Link RTT Logger and J-Link Commander tool.
Windows¥VB.NET¥ Start_JLink_VBNET	Windows, VB.NET	Visual Basic .NET Visual Studio 2010 project showing how to call J-Link DLL functions and demonstrating the startup sequence.
Windows¥VB¥ Start_JLink_VB60	Windows, Microsoft Visual Basic 6.0	Visual Basic 6.0 project for the J-Link DLL startup sequence and the usage or the J-Link SDK in a VB application.
Windows¥LabView2011¥ Startup.lvproj	Windows, LabView 2011	LabView 2011 project for the J-Link DLL startup sequence and the usage or the J-Link SDK in a LabView application.
Windows¥CSharp¥ JLink_Start_CSharp.sln	Windows, Visual Studio 2010 C#	Visual Studio 2010 project for the J-Link DLL startup sequence and the usage or the J-Link SDK in a C# application.

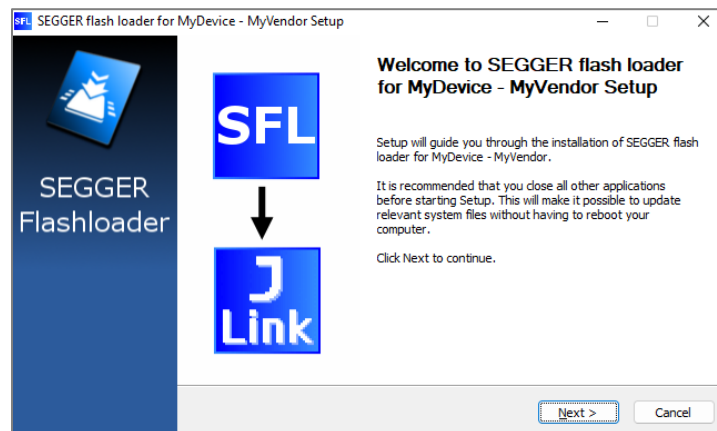
未対応デバイス対応



J-Link DSKは未対応デバイスをユーザ独自対応するためのフレームワークになります。
J-Link DSKを用いて、未対応のデバイス（CPU/(Q)SPIフラッシュなど）へのフラッシュ書込み
各種J-Linkソフトウェアを利用した開発が可能となります。

J-Link DSKは、J-Link / Flasherが対応しているArm/Cortex/RX/RISC-Vコアを搭載したデバイス対応が可能です。

SEGGERで対応しているデバイスフラッシュローダをカスタマイズする場合は、
別途費用で対象のデバイス対応ソースコードを提供することが可能です。
SEGGER対応のデバイスフラッシュローダソースコードを参考に、J-Link DSKフレーム
ワークでカスタマイズ対応が可能



J-Link DSKを用いて開発したフラッシュローダは、
インストーラサンプルを用いて、配布用インストーラを作成可能

提供物一覧



Name	Description
Flash loader templates	SEGGER Flash Loader templates. (At least one for each CPU architecture supported by the DSK)
Reference device entries	Folders and files that demonstrate the structure that is required for the J-Link software to recognize and list a device added via the DSK.
Installer	Source files and a build script for creating a custom Windows installer. The installer is intended can then be provided to end customers for automatically setting up everything necessary to integrate a device that was added via the DSK to an installation of the J-Link software.

参考情報

Flash Loader:

https://wiki.segger.com/SEGGER_Flash_Loader

J-Link Script:

https://wiki.segger.com/J-Link_script_files

多くのデバイスコアに対応したカスタム対応フレームワーク

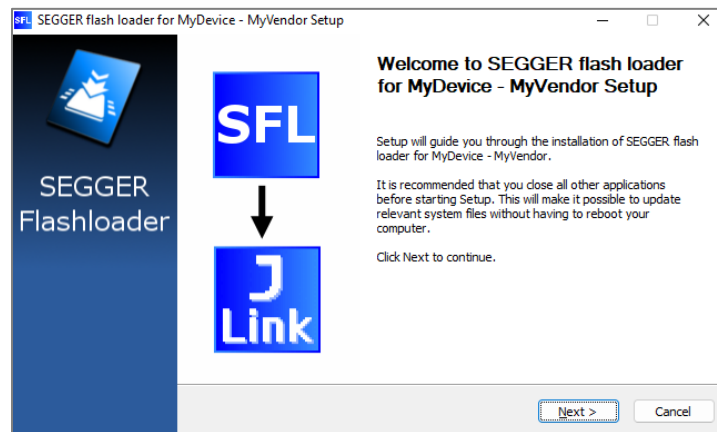


Flasher DSKは未対応デバイスをユーザ独自対応するためのフレームワークになります。
Flasher DSKを用いて、未対応のデバイス（CPU/(Q)SPIフラッシュなど）へのフラッシュ書込み
各種J-Linkソフトウェアを利用した開発が可能となります。

Flasher DSKは、J-Link / Flasherが対応しているArm/Cortex/RX/RISC-V以外のコアを搭載したデバイ
スを対象としたフレームワークになります。

※Flasher DSKには、J-Link DSKも同梱されていますので、Arm/Cortex/RX/RISC-Vコアを搭載した
デバイス対応、カスタマイズ対応も可能です。

※Flasher DSKで対応したデバイスフラッシュローダは、Flasher PROでの利用を前提としています
ので、J-LinkやFlasher ARMを利用する場合は、J-Link DSKを用いて対応ください。



Flasher DSKを用いて開発したフラッシュローダは、
インストーラサンプルを用いて、配布用インストーラを作成可能

提供物一覧



Name	Description
Emdedded Studio Development Pack	An Embedded Studio package extending Embedded Studio with a Flasher DSK project template for developing PCodes and DDFs as well as the required tools to validate and build custom flash loaders.
PCode Template	A PCode template.
Device Definition File (DDF) Template	A Device Definition File template.
Tools	Tools for building and validating PCodes and DDFs.

Flasher DSKスタートアップドキュメント

https://wiki.segger.com/Flasher/Flasher_DSK_-_Getting_Started

ライセンス方式とサポート対応

Company-Wide License



ライセンス製品	J-Link SDK	J-Link DSK	Flasher DSK
ライセンスモデル	Company-Wide	Company-Wide	Company-Wide
開発利用	ライセンス契約社内利用	ライセンス契約社内利用	ライセンス契約社内利用
開発利用制限	開発者数制限・開発プロジェクト制限なし	開発者数制限・開発プロジェクト制限なし	開発者数制限・開発プロジェクト制限なし
開発完了したツール対応オブジェクト	社内で利用可能 社外再配布はSEGGER社の確認必要	—	—
開発完了したフラッシュローダ	—	社内外への再配布可能 再配布の数量制限なし	社内外への再配布可能 再配布の数量制限なし
保守・サポート (SEGGER英語サポート)	ライセンス導入後6ヶ月 最大2時間までのサポート対応	ライセンス導入後6ヶ月 最大2時間までのサポート対応	ライセンス導入後6ヶ月 最大2時間までのサポート対応



製品については、お気軽に以下窓口へお問い合わせください。

TEL : 03-6240-2655
FAX : 03-6240-2656
e-mail : sales@embitek.co.jp
website : <https://www.embitek.co.jp>



EmbiTeK Online Shop

<https://www.embitek.shop/>



YouTube

<http://www.youtube.com/@embitek>

Arm Cortex/RXソフトウェア開発から量産をサポート

製品開発フローの課題に合わせて対応



デバッガ
開発ツール

RTOS



MPU 対応



機能安全認証
IEC61508 SIL3
IEC62304 class C

 SSL	 暗号ライブラリ	 セキュリティ認証	 GUI
 Modbus	 SSH	 ブートローダ	 圧縮・解凍
 IoT Toolkit HTTP client JSON Parser	 MQTT  Dropbox Client	 USB Host HID MassStorage Printer LAN Audio CCID Video MTP CDC FTDI MIDI HUB CP21xx UART	
 TCP/IP IPv4 / IPv6 ACD DNS client DNS-SD NetBIOS NS FTP server SNMP Agent PTP OC client Web Socket client Web Socket server DHCP server ARP mDNS server Loopback CoAP FTP client SNTP client TCP Web server PPP/PPPoE		 USB Device HID MTP CDC-NCM RNDIS Printer Audio Bulk MSD (virtualMSD) CDC-ACM CDC-ECM IP-over-USB MIDI Video DFU	
		 ファイルシステム NAND SPI/QSPI フラッシュ NOR SD SDHC SDXC MMC eMMC CF USB メモリ	

Arm Cortex / RX CPU

量産書込

