基本操作ガイド

Flasher ARM

株式会社 エンビテック

Copyright (c) 2013-2020 EmblTeK Co., Ltd. http://www.embitek.co.jp



日本代理店 SEGGER Microcontroller GmbH



本書は製品メーカーの SEGGER Microcontroller GmbH(以下 SEGGER 社といいます)の英語版ユーザー・マニュアルの情報をベースに翻訳した資料です。本資料の全部又は一部を、弊社の事前の書面による同意がない限り転載または複製することは出来ません。

本資料に記載の全ての情報の使用に起因する損害、第三者の知的財産権、権利又はその他の経費に対して SEGGER 社及び株式会社エンビテックは一切責任を負いません。 本資料の内容は、予告なく変更されることがあります。

商標

「EmblTeK」、EmblTeK ロゴは株式会社エンビテックの商標または登録商標です。その他、本資料に記載しているプロセッサ名、ツール名および製品名は、それぞれ各社の商標または登録商標です。



1.	インストール	. 3
2.	新規プロジェクト作成方法 「J-Flash」ツールを起動してプロジェクトを作成します プロジェクトの設定を行います	.5 .5 .9
3.	パソコン経由でフラッシュ書き込み	13
4.	スタンドアロンモードでのフラッシュ書き込み ASCII コマンド・モード	17 20
5.	データ Verify モードの設定	22
6.	MSD モードで接続	23
7.	複数イメージファイルのロード及び書き込み設定	24
8.	パッチ(Patch)ファイルの書き込み	25
9.	バッチ(Batch)モードプログラミング	27
10.	プログラミングサイクル数の制限を設定	29
11.	シリアル番号プログラミングの設定	30
12.	セキュア(オーソライズド)モードの設定 セキュアエリアの作成・追加 データのセキュア化 セキュアモード操作の注意事項	35 35 36 37
13.	CRC チェックサム表示について. J-Flash ツールでの CRC 表示 スタンドアロンモードでの CRC 表示 スタンドアロンモードのログファイルの確認方法	38 39 40 42

1. インストール

Flasher-ARM 本体をパソコンに USB 経由で接続する前に、SEGGER 社の WEB ページ(以下の URL)から最新版の J-Flash ツールプログラムをダウンロードしてパソコンにインストールしてください。

https://www.segger.com/downloads/flasher

Software for Flasher ARM, Flasher ATE, Flasher Portable PLUS, Flasher PRO, Flasher Secure Version Date File size 1 J-Link Software and Documentation pack for Windows V6.54c 💌 [2019-11-07] 45,190 KB 🛓 DOWNLOAD Installing the software will automatically install the J-Link USB drivers and offers to update applications which use the J-Link DLL. Multiple versions of the J-Link software can be installed on the same PC without problems; they will co-exist in different directories. J-Link Software and Documentation pack for macOS 24,712 KB V6.54c 🔻 [2019-11-07] 🛓 DOWNLOAD I J-Link Software and Documentation pack for Linux, DEB installer, 32-bit V6.54c 💌 [2019-11-07] 22,768 KB 🛓 DOWNLOAD I J-Link Software and Documentation pack for Linux, DEB installer, 64-bit V6.54c 🔻 [2019-11-07] 31,528 KB 🛓 DOWNLOAD

ダウンロードページのご使用条件をご確認いただいてから J-Link プログラムをダウンロードしてください。

License agreement						
Please read the following license agreement.						
The "software" means all J-Link related software components included in the J-Link software & documentation pack provided by SEGGER which can be downloaded at: http://www.segger.com/jlink-software.html						
"Licensor" shall mean SEGGER except under the following circumstances: If Licensee acquired the software as a bundled component of a third party product or service, then such third party shall either include this license terms or be Licensor and provide similar terms which will not exceed those set forth in this agreement.						
1. LICENSE AGREEMENT This license is a legal agreement between YOU (either an individual or a single entity) and SEGGER Microcontroller GmbH (called SEGGER). By downloading and/or using J-Link software, you agree to be bound by the terms of this agreement.						
2. LICENSE GRANT These terms apply to the entire J-Link software & documentation pack, especially to its software components, developed by SEGGER Microcontroller GmbH, which can be downloaded at: http://www.segger.com/jlink-software.html or as beta version at: http://www.segger.com/jlink-software-beta-version.html. The software consists of proprietary, unpublished products of SEGGER, protected under copyright and trade secret laws.						
I accept all the terms of the preceding license agreement.						
I confirm that I am not located in (or a national resident of) any embargoed country listed in the following document: Embargoed countries. In addition, I confirm that I am not listed on the following European Union Denied Party/Person Lists: Council Regulation (EC) No 881/2002, Council Regulation (EC) No 2580/2001.						
Download software Back						
Ø Emblitek						

プログラムインストール後、以下のように「*SEGGER¥JL inkARM_Vxyz*」フォルダ下に J-Link プログラム ツール、マニュアルドキュメント及び各種サンプルプロジェクトがロードされます。 デフォルトインストールフォルダ:「*C:¥Program Files (x86)¥SEGGER¥JL inkARM_Vxyz*」

<SEGGER¥JL i nkARM_Vxyz>

```
⊢ Doc
  - Manuals
  └ ReleaseNotes
⊢ ETC
       ⊢ Samples
  ⊢ DCC
  ⊢ GDB
  ⊢ JFIash
    └ ProjectFiles
      ⊢ AnalogDevices
      ├ Freescale ………… Freescale デバイスマイコンのサンプルプロジェクト
      - NXP
      ⊢ Fujitsu
      ⊢ .....
      └ Toshiba ………… Toshiba デバイスマイコンのサンプルプロジェクト
  ⊢ JFIashSPI
   └ ProjectFiles
  ⊢ JLink
  ⊢ RDI
  ∟ RTT
- USBDriver
⊢ JFlashARM. exe ······· J-Flash ツールプログラム
├ JLink.exe ·······ログラム
          ⊢ JMem. exe ····
       ••••
          ………………………… その他のプログラム
⊢ *. exe
```





2. 新規プロジェクト作成方法

「J-Flash」ツールを起動してプロジェクトを作成します

Flasher 本体をパソコンに接続してパソコンのスタートメニューから「J-Flash」プログラムを起動します。





新規プロジェクトを作成します。





anufacture <mark>r</mark> NXP	5	デバイスメーカ- を選択		
/lanufacturer	Device	Lore	Flash size	RAM size
IXP	MK63FN1M0xxx12 (allow security)	Cortex-M4	1024 KB	256 KB
IXP	MK63FN1M0xxx12	Cortex-M4	1024 KB	256 KB
IXP	MK64FN1M0xxx12 (allow security)	Cortex-M4	1024 KB	192 KB
IXP	MK64FN1M0xxx12	Cortex-M4	1024 KB	192 KB
IXP	MK64FX512xxx12 (allow securiture		512 KB + 128 KB	192 KB
IXP	MK64FX512xxx12		512 KB + 128 KB	192 KB
IXP	MK65FN2M0xxx18 (allow		2048 KB	256 KB
IXP	MK65FN2M0xxx18	を選択	2048 KB	256 KB
IXP	MK65FX1M0xxx18 (allow		1024 KB + 256 KB	256 KB
IXP	MK65FX1M0xxx18		1024 KB + 256 KB	256 KB
IXP	MK66FN2M0xxx ¹ mow security)	Co <mark>l</mark> tex-M4	2048 KB	256 KB
IXP	MK66FN2M ² Axx18	Coltex-M4	2048 KB	256 KB
IAF	MN66FATMUXXX18 (allow security)	Cortex-M4	1024 KB + 256 KB	256 KB
IXP	MK66FX1M0xxx18	Cortex-M4	1024 KB + 256 KB	256 KB
IXP	MK70FN1M0xxx12 (allow security)	Cortex-M4	1024 KB	128 KB
IXP	MK70FN1M0xxx12	Cortex-M4	1024 KB	128 KB
IXP	MK70FN1M0xxx15 (allow security)	Cortex-M4	1024 KB	128 KB
IXP	MK70FN1M0xxx15	Cortex-M4	1024 KB	128 KB
IXP	MK70FX512xxx12 (allow security)	Cortex-M4	512 KB + 512 KB	128 KB
IXP	MK70FX512xxx12	Cortex-M4	512 KB + 512 KB	128 KB
IXP	MK70FX512xxx15 (allow security)	Cortex-M4	512 KB + 512 KB	128 KB
IXP	MK70FX512xxx15	Cortex-M4	512 KB + 512 KB	128 KB
IXP	MK80FN256xxx15 (allow security)	Cortex-M4	256 KB + 128 MB	256 KB
IXP	MK80FN256xxx15	Cortex-M4	256 KB + 128 MB	256 KB
IXP	MK80FN256xxx15_ParallelMode (al	Cortex-M4	256 KB + 128 MB	256 KB
IXP	MK80EN256xxx15_ParallelMode	Cortex-M4	256 KB + 128 MB	256 KB

Create New Project	
Target Device	
NXP MK66FN2M0xxx18	
Little endian 💌	
SwD 4000	
<u>K</u>	8



🔜 SEGGER J-Fla	sh V6.54 - [new p	project *]				- • •	
<u>File Edit Viev</u>	v <u>T</u> arget <u>O</u> ptio	ons <u>W</u> indow	<u>H</u> elp				
File Edit Viev	v Target Optio	xx18	Help				
Base address Flash size	0x0 2048 KB		SEGG	ER)			
Application log started - J-Flash V6.54 (J-Flash compiled Oct 28 2019 10:34:53) - JLinkARMdII V6.54 (DLL compiled Oct 28 2019 10:33:23) Creating new project - New project created successfully							
List of MCU devic	ces read successfu	ully (7448 De	vices)			//.	

「File」→「Save project」又は「Save project as..」メニューコマンドでプロジェクトを保存します。

🔜 SEGGER J-Flash V6.54 - [new project *]					
File	Edit View Target C	Options	Window	Help	
	Open data file			Ctrl+0	
	Merge data file				
	Save data file			Ctrl+S	
	Save data file as				
	New project				
	Open project				
	Save project		9		
	Save project as				
	Close project				
	Save Flasher config file.				
	Save Flasher data file				



Save project as	×
保存する場所(D: 🌗 emPower 🗨 🗲 🗈 📸 🖛	
名前 更新日時	
検索条件に一致する項目はありません。	
デスクトップ	
に ライブラリ	
コンピューター プロジェクト名	\mathbf{i}
ネットワーク・	Þ
ファイル名(N): empower jflash 「【保存(S)	
ファイルの種類(工): J-Flash project file (*.jflash) 🗨 キャンセル	



プロジェクトの設定を行います

J-Flash ツールの「Options」→「Project Settings..」メニューコマンドでプロジェクト設定ダイログウを開きます。



ターゲットボードのハードウエア仕様を確認して、JTAG・SWD インターフェースモードを選択します。 クロック設定を指定します。





Project settings		? <mark>- ×)</mark>
General Target Interface MCU Flash	2 tion	
JTAG		
JTAG speed before init steps	JTAG speed after init steps	
C Auto selection	C Auto selection C Adaptive clocking	
● 4000 ▼ kHz		
JTAG scan chain information	⊻erify D etect	
C Simple configuration Position 0	IRPre 0	
0 is closest to	TDO. Sum of IRLens of devices closer to TDO. IRLen of ARM7/ARM9	
C Detailed configuration	devices is 4.	
TDO		
TDI		JTAG の場合は
Add Insert Delete	e Edit Up Down	スキャンチェーンの
		設定も必要
	OK +7	」 ンセル適用(<u>A</u>)

基本的に、内蔵フラッシュの書き込みの場合は不要ですが、必要でしたら外部バスの初期化及びクロック設定等の初期化処理を「Init Steps」項目コマンドで設定します。

Project settings	?
General Target Interface MCU Flash Production Peri	
Device	
Core Mask 0F000FFF	
Cortex-M4 ✓ Use target BAM (faster) Little endian ✓ Addr 1FFF0000 256 KB ✓	
Init steps	
# Action Value0 Value1 Comment	
U neset U Ums neset and hait target	
4	必要の場合はターゲット 依存分の初期化設定を行い ます
Add Insert Delete Edit Up Dgwn	
OK*	**>セル 適用(A)



Reset モード設定について:

# Action 0 Reset	Value0 0	Value1 0 ms	Comment
0 Reset	0	0 ms	The second se
			Heset and halt target
Add	Insert Delete	<u>E</u> dit	<u>U</u> p D <u>o</u> wn



Reset Type	Type 説明					
0	Normal (デフォルト):マイコンデバイス種類により DLL 側で自動判断、通常 JTAG/SWD ソフト ウエアコマンドでリセットを行います。					
1	Core:JTAG/SWD ソフトウエアコマンドで Cortex-M コアのみリセット					
2	ResetPin:15PinのRESET 信号からハードウエアリセットをかけてマイコンデバイスの JTAG/SWD ポートに接続します。					
3	Connect under Reset: 15PinのRESET 信号からのハードウエアリセット中にマイコンデバイスのJTAG/SWD ポートに接続します。					
4	Reset core & peripherals, halt after bootloader: リセット後に内蔵ブートローダーROM を 起動されるマイコンデバイス用、ブートローダー実行後に CPU を停止します。					
5	Reset core & peripherals, halt before bootloader: リセット後に内蔵ブートローダーROM を起動されるマイコンデバイス用、ブートローダー実行前に CPU を停止します。					
6	Reset for Freescale Kinetis devices					
7	Reset for Analog Devices CPUs (ADI Halt after kernel)					
8	Reset core and peripherals					
9	Reset for LPC1200 devices					
10	Reset for Samsung S3FN60D devices					



Project settings		? 🔀	
General Target Interface MCU Flash Production Perf	ormance		
Target power supply Reference volta	ge settings		
Power source NONE Enable Enable	01-000	ターケットに電源を	
Delay before start 20 ms . 1000	mV max.	共有する設定	
Discharge target on disconnect	3300 h		
Program serial number			
Address 00000000 Len 4			
Next SN 1 Increment			
Actions performed by "Production Programming"	verride timeouts		
	rase 15000 ms		
IV Program Pro	dram 10000 ms	AUTO コマンド	
Start application Via SEB access	zeniy] toosa	内容の設定)
			\sim
		8	
	ОК	キャンセル 適用(A)	

「File」 \rightarrow 「Save project」 メニューコマンドでプロジェクトを保存します。



3. パソコン経由でフラッシュ書き込み

Flasher 本体をパソコンに接続してパソコンのスタートメニューから「J-Flash」プログラムを起動します。

「File→Open data file」コマンドでアプリケーション・データファイル (*. mot、*. srec、*. hex、*. bin、*. raw) を開きます。

🐌 Eval-With-emPower	*	名前	更新日時	種類
\mu emPower		emPower K66 bin	2018/09/24 14:48	BIN ファイル
📜 Eirmuara Dounarado			2010/05/2111.10	0111 27 170

バイナリデータファイルの場合は、ダイアログボックスからベースアドレスを設定します。 例えば、内蔵フラッシュのベースアドレスが「0x1800000」の場合は、ダイアログボックスに「18000000」を入 力します。emPower 評価ボードのマイコンの内蔵フラッシュベースアドレスは「0x00000000」です。

Enter start address	x
Start address (hex) 0x	OK
	Cancel

SEGGER J-Flash V6.54 - [C:¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower;flash *]												
<u>File Edit View Target Options W</u>	<u>File Edit View Target Options Window Help</u>											
Name Value Host connection USB [Device 0]	Address: 0x0 x1 x2 x4 Address: 0 1 2 3 4 5 6 7 8 9 A B C D E F ASCII ASCII											
Init SWD speed 4000 kHz SWD speed 4000 kHz MCU NXP MK66FN2M0xxx18 Core Cortex-M4 Endian Little Check core ID Yes (0x4BA00477) Use target RAM 256 KB @ 0x1FFF0000 Flash memory Internal bank 0	00000 00											
Flash size 2048 KB	00080 7F 04 00 00 81 04 00 00 83 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 85 04 00 00 104 00 00 95 04 00 00 104 00 00 104 00 00 104 00 00 104 00 00 11 104 00 00 12 1 1 1 04 00 00 1											
Application log started - J-Flash V654 (J-Flash compiled Oct 28 2019 10:34:53) - JLinkARMdII V654 (DLL compiled Oct 28 2019 10:33:23) Creating new project - New project created successfully Saving project file [C*SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower jflash] - Project saved successfully Opening data file [C*SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥emPower K66bin] - Data file opened successfully (636088 bytes, 1 range, CRC of data = 0x30F68AA8, CRC of file = 0x30F68AA8)												
Ready												



ターゲットボードに電源を入れて、J-Flash プロジェクトの「Target→Connect」コマンドでボード側の CPU と の接続を確認します。

NOTE

JTAG/SWD 接続が正しく出来ない場合の原因:

- ターゲットインターフェースモードの設定(JTAG?SWD?FINE 等)
- インターフェースクロック設定が高すぎる(4000kHz以下の設定値でお試しください) •
- ターゲットボード依存分の初期化設定が不十分(J-Flash プロジェクトの「Init steps」項目に必要な初 • 期化を設定してください)

ターゲットメニューから「Target → Manual Programming」コマンドで、「Erase sectors、Program、Verify」 コマンドを順番に行います。

HAS	SEGGER J-Flash V6.54 - [C:¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower.jflash *]															
F	ile Edit View	Targ	get Options W	/ind	dow Help											
	J Project - em	n	Connect				pp	ort¥ General¥]-Flash¥	AO¥Eva	-Wit	h-em	Pow	er¥e	mPo	wer¥	
Γ	Name	N	Disconnect				-	v1 v2 v4								
	Host connection	l.	Test			×	1		7 0	0	٨	D	C	П	E	
	Target interface	9	Production Prog	rar	mming	F7	00	03 20 11 04 00	00 4D	04	00	00	89	7B	04	
	SWD speed	4	Manual Program	۱m	ing	Þ	<u> </u>	Secure Chip		04	00	00	00	00	00	
	MCU	NXP M	KEEN2M0vvv18	П	00020	100		Unsecure Chip		00	00	00	57	04	00	
	Core	Cortex-M4			00030			Chock Blank	E2	61	04	00	A1	7A	04	
	Endian Chaolana ID	Little	40 4 00 477)		00040	_ 5F	_	CHECK BIdHK	F2	04	00	00	65	04	00	
	Lheck core ID	256 KP	Ux4BAUU477J	0x48A00477) (B /@ 0v1EEE0000	Ш	00050	67		Erase Sectors	F3	04	00	00	6D	04	00
		200112	0.000		00060	6F		Erase Chip	F4	04	00	00	75	04	00	
	Flash memory Base address	Interna 0x0	l bank 0		00070	77		Program	E5	04	00	00	7D	04	00	
	Flash size	2048 K	В		00080	7F		Drogram & Varify	56	04	00	00	85	04	00	
					00090	87		Program & verily	FO	04	00	00	8D	04	00	
					000A0	8F		Verify	F8	04	00	00	95	04	00	
					000B0	97		Read back	•	04	00	00	9B	04	00	
					000C0	9D			50	04	00	00	81	79	04	
					000D0	21		Start Application	F9	04	00	00	AB	04	00	

LOG

Blank checking 0x00140000 - 0x0015FFFF
 Blank checking 0x00160000 - 0x0015FFFF
 Blank checking 0x00180000 - 0x0019FFFF
 Blank checking 0x001A0000 - 0x001BFFFF
 Blank checking 0x0010000 - 0x001BFFFF
 Blank checking 0x001E0000 - 0x001FFFFF
 Blank checking 0x001E0000 - 0x001FFFFF

- Start of restoring - End of restoring - Erase operation completed successfully - Completed after 1.642 sec



J-Flash V6.54	
Erase operation completed successfully - Completed after 1.642 sec	
ОК	
J-Flash	23
Programming target memory	
Programming range 0x00038800 - 0x00038FFF (2 KB) 36%	
Cancel	
J-Flash V6.54	J-Flash V6.54
Target programmed successfully - Completed after 8.388 sec	Target verified successfully - Completed after 0.848 sec
ОК	ОК

マニュアルコマンド:

コマンド	説明
Connect	JTAG 経由でターゲットのマイコンに接続します。プロジェクト設定のターゲ
	ット I/F モードにて接続及び初期化を行います。
Disconnect	ターゲットの JTAG 接続を解除します。
Check blank	ターゲットのフラッシュデバイスのブランクチェックを行います。
Erase sectors	プロジェクト設定で選択されている全てのセクターをイレーズ(OxFF デー
	タ)します。
Erase chip	フラッシュデバイスの全てのデータをイレーズします。
Program	データイメージをターゲット上のフラッシュメモリに書き込みます。
Program & Verify	データイメージをターゲット上のフラッシュメモリに書き込んでデータ内容
	を比較(ベリファイ)します。
Verify	データファイルとフラッシュデバイスのデータをベリファイします。
Read back >	フラッシュデバイスのメモリデータを読み出して、新しいデータファイルと
	して保存します。"Selected sectors"コマンドでデータを読み出す場合は、
	プロジェクト設定で選択されたセクターのデータを読み出します。
	"Entire chip" コマンドの場合は、全てのセクターのデータを読み出します。
	"Range"コマンドの場合は、ユーザー指定のフラッシュメモリ空間のデ
	ータを読み出してファイルとして保存します。



ターゲットメニューから「Target → Production Programming」コマンドを実行しますと、「Connect→Erase→ Program→Verify」コマンドが順番に実行されます。

🔜 SEGGER J-Fla	sh V6.	54 - [C:¥SEGGE	R_Support¥_G	eneral¥)-Fla	sh¥FA	Q¥Eva	al-Wit	h-en	nPow	/er¥e	emPo	ower	¥em	powe	er.jfla	sh *]
File Edit View	Tar	get Options	Window Help)														
Project - em	מו	Connect			ppo	rt¥ G	enera	I¥J-Fla	ash¥l	FAQ¥	Eval	-Wit	h-en	Pow	/er¥e	emPo	wer	¥en
Name	Ň	Disconnect			H	_	×1	×2	×4									
Host connection	ι	Test		+	1	2 1	3 1	5	 	7	8	q	Α	R	C	П	F	F
Target interface	9	Production Pro	ogramming	F7	00	03 2	20 11	04	00	00	4D	04	00	00	89	7B	04	00
SWD speed	4	Manual Progra	amming	Þ	04	00 0	0 53	04	00	00	55	04	00	00	00	00	00	00
MCU				00	00	00 0	00 00	00	00	00	00	00	00	00	57	04	00	00
Core	Cortex	-M4	00030	00	00	00 0	00 00	-00	00	00	15	61	04	00	Α1	7A	04	00
Endian	Little		00040	5F	04	00 0	0 61	-04	00	00	63	04	00	00	65	04	00	00





LOG - End of flash programming Flash programming performed for 1 range (638976 bytes) 0x0000 - 0x9BFFF (156 Sectors, 624 KB) Start of verifying flash End of verifying flash - Start of restoring - End of restoring - Executing exit sequence De-initialized successfully - Target erased, programmed and verified successfully - Completed after 10.230 sec



ł

4. スタンドアロンモードでのフラッシュ書き込み

Flasher 本体をパソコンに接続してパソコンのスタートメニューから「J-Flash」プログラムを起動します。

「File→Open data file」コマンドでアプリケーション・データファイル (*. mot、*. srec、*. hex、*. bin、*. raw) を開きます。

🐌 Eval-With-emPower	*	名前	更新日時	種類
퉬 emPower		emPower, K66 bin	2018/09/24 14:48	BIN ファイル
📜 Firmuara Dounarada			2010/05/2111.10	5111 27 170

バイナリデータファイルの場合は、ダイアログボックスからベースアドレスを設定します。 例えば、内蔵フラッシュのベースアドレスは「0x1800000」の場合は、ダイアログボックスに「18000000」を入 力します。emPower 評価ボードのマイコンのフラッシュアドレスは「0x00000000」です。

Enter start address	X
Start address (hex) 0x	OK
	Cancel

SEGGER J-Flash V6.54 - [C:¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower.jflash *]											
<u>File Edit View Target Options Window H</u> elp											
Name Value	C:¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥emPower Address: 0x0 x1 x2 x4										
Host connection USB (Device U) Target interface SWD Init SWD speed 4000 kHz SWD speed 4000 kHz MCU NXP MK66FN2M0xxx18 Core Cortex-M4 Endian Little Check core ID Yes (0x4BA00477) Use target RAM 256 KB @ 0x1FFF0000 Flash memory Internal bank 0 Base address 0x0 Flash size 2048 KB	Address 0 1 2 3 4 5 6 7 8 9 A B C D E F ASCII 00000 00 00 03 20 11 04 00 04 00 08 9 7 8 9 A B C D E F ASCII • 00010 51 04 00 00 55 04 00										
Application log started - J-Flash V6.54 (J-Flash compiled Oct 28 2019 10:34:53) - JLinkARM.dll V6.54 (DLL compiled Oct 28 2019 10:38:23) Creating new project - New project created successfully Saving project file [C¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower jflash] - Project saved successfully Opening data file [C¥SEGGER_Support¥ General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower K66bin] - Data file opened successfully (636088 bytes, 1 range, CRC of data = 0x30F68AA8, CRC of file = 0x30F68AA8)											
Ready											



J-Flash メニュー「File→Download config & data file to Flasher」コマンドで、プロジェクトのコンフィグ レーション設定ファイル (*. CFG) 及びデータファイル (*. DAT) を Flasher-ARM 本体にダウンロードします。 Flasher-ARM 本体のフラッシュ ROM に「FLASHER. CFG と FLASHER. DAT」ファイル及びデバイス関連の PCode (PEX ファイル) 等必要なデータがロードされます。

🔜 SEGGER J-Flash V6.54 - [C:¥SEGGER_Support¥_General¥J-Flash¥FAQ¥Eval-With-emPower¥emPower¥empower.jflash *]																		
File	Edit View Target Options Window	Help																
	Open data file	Ctrl+0	upp	ort¥	Gen	eral	¥]-Fla	ash¥	FAO¥	Eval	-Wit	h-en	1Pow	/er¥e	emPo	wer	¥em	Pow
	Merge data file																	
	Save data file	Ctrl+S	H	-		×⊥	<u>×</u>	<u>×4</u>						-				
	Save data file as		1	2	3	4	5	6	7	8	9	A	B	C		E	F	AS
	New project		00	03	20		04	00	00	4U	04	00	00	89	/B	04	00	···
	New project		04	00	00	53	04	00	00	55	04	00	00	00	00	00	00	Q.
	Open project		00	00	00	00	00	00	00	00	00	00	00	57	04	00	00	••
	Save project		00	00	00	00	00	00	00	15	61	04	00	Α1	7A	04	00	
	Save project as		04	00	00	61	04	00	00	63	04	00	00	65	04	00	00	
	Close project		04	00	00	69	04	00	00	6B	04	00	00	6D	04	00	00	g.
	Cours Flasher config file		04	00	00	71	04	00	00	73	04	00	00	75	04	00	00	ο.
	Save Flasher config file		04	00	00	79	04	00	00	7B	04	00	00	7D	04	00	00	₩.
	Save Elasher data file		04	00	00	81	04	00	00	83	04	00	00	85	04	00	00	Ι.
	Download config & data file to Flasher		04	00	00	89	04	00	00	8B	04	00	00	8D	04	00	00	
-	Download serial number file to Flasher		04	00	00	91	04	00	00	93	04	00	00	95	04	00	00	
	Export setup file		04	00	00	99	04	00	00	5D	04	00	00	9B	04	00	00	
			04	00	00	9F	04	00	00	Α1	04	00	00	81	79	04	00	
	Recent Files	•	7A	04	00	Α7	04	00	00	Α9	04	00	00	AB	04	00	00	!z

NOTE

- J-Flash プロジェクトに JTAG 及び MCU 設定を自動モードに設定された場合は、「File→Download config & data file to Flasher」コマンドがエラーで終了してしまいます。Flasher-ARM 本体をスタンドアロンモードに設定するために、J-Flash プロジェクト設定ダイログの JTAG 及び MCU 設定をマニュアルモードで設定してください。
- 「Automatically detect flash memory」オプションを有効に設定した場合は、Flasher-ARM 本体がターゲットボードのフラッシュの CFI テーブル情報を読み出しますので、ターゲットに電源を入れてから「File→Download config & data file to Flasher」コマンドを実行してください。

スタンドアロンモード設定の前に J-Flash ツールからのターゲット接続及び「Production Programming」 コマンドでフラッシュ書き込みを必ず確認してください。



J-Flash プログラムとの接続を終了して Flasher-ARM 本体をスタンドアロン・モード(パソコン接続不要)で起動します。ターゲットボードに電源を入れて(Flasher-ARM 本体からボードに電源を供給しない場合)、本体の PROG/START ボタンを押してターゲットフラッシュの書き込み及びデータ確認を行います。



スタンドアロンモードの操作ステータスは本体上の LED 表示で現れます。

State	LED 表示	ステータス内容
#0	緑色 LED:点滅なし(ON)	スタンドアロンモードでのスタートイベントを待っている状態です。
#1	緑色 LED: 遅い点滅	ターゲットボードのフラッシュの読み書き操作を実行中です。 Erase:点滅速度は 6. 25Hz ぐらい Program:点滅速度は 1. 67Hz ぐらい Verify:点滅速度は 5Hz ぐらい
#2	緑色 LED:点滅なし(ON) 赤色 LED:OFF 又は ON	赤色 LED OFF: フラッシュ書き込みが正常に終了した状態を示します。 赤色 LED ON: フラッシュ書き込みがエラーで終了した状態を示しま す。自動的に State#0 に入りますが、State#1 に入るまで赤色 LED は 消えません。



ASCIIコマンド・モード

Flasher ARM をスタンドアロンモードに設定し、Flasher-ARM 本体の RS232 コネクタをシリアルケーブルで接続 して、パソコンのシリアルポートコンソールからフラッシュ書き込み等の遠隔操作が可能です。 パソコン側に RS232 ターミナル(TeraTerm 等)ソフトウエアを開きます。

- Baudrate: 9600bps
- 8 data bits
- No Parity
- 1 Stop bit

📟 Tera Term - VT	
File Edit Setup Control Window Help	
<pre>\$BTL: Bootloader started successfully \$BTL: Initializing file system OK \$BTL: Checking NAND flashOK \$BTL: Trying to open firmware file ERR: File not found \$BTL: Check for valid firmware OK \$BTL: Starting firmware \$FW: Firmware started successfully \$FW: Initializing file system OK \$FW: Checking NAND flash OK \$FW: Checking NAND flash OK \$FW: Checking NAND flash OK \$FW: Initializing NET module OK \$FW: Initializing FTP server OK \$FW: Initializing telnet server OK \$FW: Initializing telnet server OK \$FW: Initializing telnet server OK \$FW: Initializing CMD module OK</pre>	

遠隔操作コマンドのメッセージフォーマット

- ASCII コマンド文字列は「#」で始まります。
- ASCII コマンド文字列は「'¥r'」で終了します。
- ASCIIコマンドは、大文字又は小文字が可能です。
- Flasher コマンド

#AUTO	#AUTO PATCH	#AUTO NOINFO
#BAUDRATE <baudrate></baudrate>	#CANCEL	#ERASE
#PROGRAM	#RESULT	#SELECT <filename></filename>
#START	#STATUS	#VERIFY



■ 「#AUTO」コマンド

本 ASCII コマンドで J-Flash プロジェクトの「Production」タブページでの設定の通りに「Production Programming」コマンド処理(ERASE→書き込み→ベリファイ)を実行します。

コマンドレスポンス回答:

#OK

正常終了 通常は、以下のようなメッセージが表示されます。

#ACK #STATUS:INITIALIZING #STATUS:CONNECTING #STATUS:UNLOCKING #STATUS:ERASING #STATUS:PROGRAMMING #STATUS:VERIFYING #OK (Total 13.993s, Erase 0.483s, Prog 9.183s, Verify 2.514s)

#ERRxxx コマンド操作がエラーで終了しました。エラーコードは「xxx」で表示されます。

#START

本 ASCII コマンドでターゲットフラッシュのアプリケーションを起動します。

 コマンドレスポンス回答: #0K
 正常終了
 通常は、以下のようなメッセージが表示されます。

> #ACK #STATUS:INITIALIZING #STATUS:CONNECTING #OK (Total 1.148s)

#ERRxxx コマンド操作がエラーで終了しました。エラーコードは「xxx」で表示されます。

#STATUS

本 ASCII コマンドで Flasher-ARM の現在のステータスを取得します。

コマンドレスポンス回答:
 現在のステータスが表示されます。



5. データ Verify モードの設定

ベリファイ処理はアプリケーションイメージデータ(スタンドアロンモードの場合は Flasher ARM 本体に保存されているデータ)とフラッシュに書き込みましたデータを比較します。 比較方法は以下の 2 種類をサポートします。

- チェックサム(CRC)で確認
- データバイト毎に比較

CRC チェックサム方法について以下 URL の情報をご参照ください。 https://wiki.segger.com/CRC_calculation

比較方法の選択は J-Flash プロジェクト→Performance タブ下の「Perform Verify」項目の設定値で指定可能です。

「via CRC」→ チェックサム(CRC)で確認 「via readback」→ 各データバイトを比較

Project settings	Project settings	
General Target Interface MCU Flash Production Performance	General Target Interface MCU Flash	
On "Erase selected sectors" (F3)	On "Erase selected sectors" (F3)	
On "Erase chip" (F4)	On "Erase chip" (F4)	
Perform blank check	Perform blank check	
On "Program" (F5)	On "Program" (F5)	
Skip blank data	Skip blank data	
On 'Verify''	On 'Verify''	
Perform verify Via CRC	Perform verify via readback v	

ベリファイ処理の時に使用するチェックサムは、イメージファイル全領域の CRC ではなく、フラッシュに書き込むデータ分だけの CRC 計算になります。

J-FlashのV6.xxのCRC 計算式とV5.xx及びV4.xxのCRC 計算式が異なっております。

22/42

6. MSD モードで接続

Flasher-ARM 本体を USB ディスク(Mass Storage Device) としてパソコンに接続するモードを、MSD モードといいます。Flasher-ARM 本体のスタートボタンを押した状態でパソコンに USB 接続をした場合は、 Flasher-ARM 本体は MSD モードで起動します。

MSD モードで接続した場合は、Flasher-ARM 本体の内蔵フラッシュのファイルデータのコピー、削除等の操作が パソコンから自由に出来ます。J-Flash プロジェクトで作成したコンフィグレーションファイル (FLASHER. CFG) とイメージデータファイル (FLASHER. DAT)を MSD モードで Flasher-ARM 本体にコピーすることが可能です。



Flasher-ARM 本体のファイルシステムでは 8.3 式(12345678.123)が使用されており、ファイル名の最大文字数が「8」となっております。Long ファイル名はサポートしておりませんので注意してください。



FLASHER

7. 複数イメージファイルのロード及び書き込み設定

複数プログラムイメージを Flasher 本体(合計サイズ 120MB 以内)に登録することで、簡単に、ターゲットのバ ージョンにマッチするプログラムのロード、又は異なる複数ターゲットのプログラムダウンロードに対応出来ま す。

毎回パソコン側の J-Flash ツールからプログラムをダウンロードする必要がなく、シリアルコンソールからダウ ンロードするプログラムを選択して実行します。一度選択したプログラムファイル名は本体に登録されるので、 プログラムを変更しない限り毎回コンソールから選択コマンドを行う必要がありません。

以下、1つの Flasher-ARM 本体での複数イメージファイルの取り扱い手順を説明します。

- ① J-Flash プログラムの「File→Save flasher config file...」及び「File→Save flasher datafile...」 メニューコマンドにて複数プロジェクトの CFG+DAT ファイル(例:ファームウエア 1: FLASHER1. CFG・ FLASHER1. DAT、ファームウエア 2: FLASHER2. CFG・FLASHER2. DAT…)をローカルフォルダに用意します。注 意点:Flasher-ARM 本体のファイルシステムでは 8.3 式(12345678.123)が使用されており、ファイル名の 最大文字数が「8」となっております。
- ② Flasher-ARM本体をMSDモードでパソコンに接続して、CFG及びDATファイル(FLASHER1. CFG・FLASHER1. DAT、 FLASHER2. CFG・FLASHER2. DAT…)をFlasher-ARM本体ディスクにコピーします。
- ③ ターゲットボードのフラッシュに書き込むイメージセット(ファームウエア 1、ファームウエア 2…)を以下の方法で選択します。
- A) MSD モードで Flasher-ARM 本体のディスク上の「FLASHER. INI」ファイルを編集してデフォルトプロジェクトファイル名を指定します。「FLASHER. INI」ファイルが存在しない場合は以下の内容でファイルを作成してください。
 - FLASHER. INI ファイル(ファームウエア1を選択する場合) [FILES] DataFile = "FLASHER1.DAT" ConfigFile = "FLASHER1.CFG"
 - FLASHER. INI ファイル (ファームウエア 2 を選択する場合) [FILES] DataFile = "FLASHER2.DAT" ConfigFile = "FLASHER2.CFG"

MSD モードを終了して、スタートボタンを押してターゲットフラッシュに「FLASHER. INI」 ファイルで指定されたファームウエアの書き込みが出来ます。

B) Flasher-ARM本体をRS232シリアルケーブルでパソコンに接続して、パソコン側のシリアルコンソールASCII ターミナル(9600BPS/8Bit、Local Echo ON)を開きます。Flasher-ARM 本体をスタンドアロンモードに起 動すると、パソコン側のシリアルコンソールから Flasher-ARM の操作が可能です。シリアルコンソールから 「#SELECT <filename>」コマンドで、本体にダウンロードしたプログラム(ファームウエア1、ファームウ エア2…)を選択します。

「#SELECT FLASHER1」→ ファームウエア1を選択 「#SELECT FLASHER2」→ ファームウエア2を選択 「#SELECT」コマンドで選択したプログラムがデフォルトプログラムに設定されるので、いつでも本体 のボタン又はコンソールから「#AUT0」コマンドでターゲットボードのフラッシュ書き込みが出来ます。



8. パッチ (Patch) ファイルの書き込み

スタンドアロンモードにおいて Flasher は、ユーザー指定パッチファイルでターゲットデータの内容に パッチを適用する機能をサポートしています。スタンドアロンモードでプログラミングプロセスを開始する前に、 Flasher は、本体ディスクに存在する「Patches.txt」ファイルを探します。「Patches.txt」ファイルにパッチ コマンドが含まれています。

「Patches.txt」パッチファイルが存在する場合、「Serial.txt」にある数値で指定された「Patches.txt」の行番号のコマンドサイクルが実行されます(行番号は0から始まります)。

「Patches. txt」の各行は、各パッチの長さは最大 32 バイトで、4 パッチまで保持することができます。

構文

<NumPatches>==4 の場合、

<NumPatches>,<Addr>,<NumBytes>,<Data>,<Addr>,<NumBytes>,<Data>,<Addr>, <NumBytes>,<Data>,<Addr>,<NumBytes>:<Data>¥r¥n

各パッチラインは<Addr>,<NumBytes>:<Data>が含まれています各パラメータを記述したテーブルの下に検索してください。

パラメータ	説明
<numpatches></numpatches>	このパッチラインのパッチの数を設定します。最大値は4パッチです。
<addr></addr>	パッチを適用するアドレスを記述します。 値は 16 進数で設定します。
<numbytes></numbytes>	現在のパッチのバイト数です。 最大値が 20H(10 進数で 32)です。
<data></data>	パッチが適用されるデータを設定します。 <data>は常に1バイト当たり2文字として設定します。</data>

ご注意点:

すべての値は 16 進形式(16 進数)で設定してください。 <Data>セクション項目の前に「,」ではなく必ず「:」を設定してください。

事例:)

以下のサンプルセキュエンスでパッチコマンドの使用方法を説明します。 Flasher 本体ディスクの「Patches. txt」ファイルに以下の行が記述されています。

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

「Serial.txt」ファイルには「0」が記述されています。

0

これにより「Patches. txt」の0行のパッチコマンドが実行され、以下のようにパッチデータの書き込みが行われます。



アドレス	バイト数	データ
0x100025	3	0xAA 0xBB 0xCC
0x100063	2	0xDD 0xEE
0x100078	1	0xFF

● RS232 経由でのパッチサポート

RS232 ターミナルから ASCII コマンドを発行して1行のパッチコマンドサイクルは実行できます(「Patches.txt」ファイルは不要です)。

#AUTO PATCH <NumPatches>, <Addr>, <NumBytes>:<Data>

「#AUTO」コマンドで書き込むメインイメージデータに加えて、指定パッチデータの書き込みも行います。



9. バッチ (Batch) モードプログラミング

バッチプログラミング機能を使用してスタンドアロンモードで複数の書き込みセッション(例:同じ JTAG チェ ーンでの複数ターゲットデバイスへの書き込み又は、同じターゲットに複数データファイルの書き込み)を連続 に実行することは可能です。

Flasher ARM ではバッチプログラミングで連続に使用する書き込みセッションの数に制限はありません。

以下、バッチプログラミング機能の設定手順を示します。

- J-Flash プログラムの「File→Save flasher config file...」及び「File→Save flasher datafile...」 メニューコマンドにて複数プロジェクトの CFG+DAT ファイル(例:ファームウエア 1: FLASHER1. CFG・ FLASHER1. DAT、ファームウエア 2: FLASHER2. CFG・FLASHER2. DAT…)をローカルフォルダに用意します。
- ② Flasher-ARM本体をMSDモードでパソコンに接続して、CFG及びDATファイル(FLASHER1. CFG・FLASHER1. DAT、 FLASHER2. CFG・FLASHER2. DAT…)をFlasher-ARM本体ディスクにコピーします。
- ③ MSD モードで Flasher-ARM 本体のディスク上の「FLASHER. INI」ファイルを編集してデフォルトプロジェクトファイル名を指定します。「FLASHER. INI」ファイルが存在しない場合は以下の内容でファイルを作成してください。
 - FLASHER. INI ファイル (書き込み順:ファームウエア 1→ファームウエア 2) [BATCH] DataFile = "FLASHER1.DAT" ConfigFile = "FLASHER1.CFG" DataFile1 = "FLASHER2.DAT" ConfigFile1 = "FLASHER2.CFG"

MSD モードを終了して、スタートボタンを押してターゲットフラッシュに「FLASHER. INI」ファイルで指定され た書き込みセッションが実行されます。

フラッシュ書き込み中にエラーが発生した場合はバッチプログラミングが中止されます。

ご注意点:

Flasher ARM では「バッチプログラミング」と「複数イメージファイルサポート」機能は同時に使用することは出来ません。そのため、FLASHER. INI ファイルに[BATCH]コマンドを指定する場合は同時に[FILE]コマンドの設定は出来ません。



使用例1)同じターゲットフラッシュに2つのイメージファイルを書き込む

- 設定済みの J-Flash プロジェクトを開きます。
- 「File→Save flasher config file...」メニューコマンドでコンフィグレーション設定ファイル (DEFAULT. CFG)を保存します。
- J-Flash プロジェクトにデータ1ファイル(ブートローダイメージ)を開きます。
- 「File→Save flasher datafile...」メニューコマンドでDAT1ファイル(BOOT.DAT)を保存します。
- J-Flash プロジェクトにデータ2ファイル(アプリケーションイメージ)を開きます。
- 「File→Save flasher datafile...」メニューコマンドでDAT2ファイル(APP.DAT)を保存します。
- Flasher-ARM 本体を MSD モードでパソコンにディスクとしてマウントして DEFAULT. CFG、BOOT. DAT、 APP. DAT ファイルを Flasher 本体にコピーします。
- Flasher 本体ディスク上の「FLASHER. INI」ファイルを開きます。「FLASHER. INI」ファイルが存在しない場合は新規で作成します。内容でファイルに[BATCH]コマンドを指定します。
 - [BATCH] DataFile = "BOOT.DAT" ConfigFile = "DEFAULT.CFG" DataFile1 = "APP.DAT" ConfigFile1 = "DEFAULT.CFG"

使用例2) STM32F1 と STM32F4 のデュアルコア環境でコア毎のアプリケーションを書き込む

- 設定済みの J-Flash プロジェクトを開きます。
- 「File→Save flasher config file...」と「File→Save flasher datafile...」メニューコマンドで STM32F1 用の CFG 及び DAT ファイル (STM32F1. cfg、F1DATA. dat) を保存します。
- 「File→Save flasher config file...」と「File→Save flasher datafile...」メニューコマンドで STM32F4 用の CFG 及び DAT ファイル (STM32F4. cfg、F4DATA0. dat、F4DATA1. dat)を保存します。 Flasher-ARM 本体を MSD モードでパソコンにディスクとしてマウントして STM32F1. cfg、 F1DATA. dat、STM32F4. cfg、F4DATA0. dat、F4DATA1. dat ファイルを Flasher 本体にコピーします。
- Flasher 本体ディスク上の「FLASHER. INI」ファイルを開きます。「FLASHER. INI」ファイルが存在しない場合は新規で作成します。内容でファイルに[BATCH]コマンドを指定します。

[BATCH]

DataFile = "F1DATA.dat" ConfigFile = "STM32F1.cfg"

- DataFile1 = "F4DATA0.dat"
- ConfigFile1 = "STM32F4.cfg"
- DataFile2 = "F4DATA1.dat" ConfigFile2 = "STM32F4.cfg"



10. プログラミングサイクル数の制限を設定

Flasher は、スタンドアロンモードで書き込みを行うプログラミングサイクル数の制限機能をサポートします。 この機能を使用するには、現在のセットアップで行うプログラミングサイクル数を示す進数(符号無し 32 ビッ ト整数)が記述された「Cntdown.txt」ファイルを Flasher 本体に配置してください。

事例:)

Cntdown.txt - メモ帳	
ファイル(E) 編集(E) 書式(Q) 表示(V) ヘルプ(H)	
388	~
	-

通常、この機能はオーソライズドフラッシュと組み合わせて使用されます。

■ ご注意点:

「Cntdown.txt」ファイルのプログラミングサイクル数は書き込みが成功した場合のみ更新されます。失敗 したプログラミングサイクル数は、Cntdown.txt に影響されません。 プログラミングサイクル数を制限する機能は V4 以下の Flasher ARM にサポートしておりません。

● 本体上の LED 表示について

プログラミングサイクル数を制限する機能が使用される場合(「Cntdown.txt」ファイルが本体に存在する場合) は、本体上の赤色 LED 表示の動作が少し異なります。 残りのプログラミングサイクル数が10以下の場合は、

- 赤い Fail/error LED が1秒間点灯します。
- ・ その後、5 Hz 周期で残りのプログラミングサイクル数回分 LED がトグル/点滅します。例えば、残りは5 サイクルの場合は5回点滅します。



11. シリアル番号プログラミングの設定

Flasher は、スタンドアロンモードで以下2通りの方法でターゲット装置にシリアル番号の書き込みをサポートします。

- J-Flash プロジェクトの指定番号から連続にインクリメントしてシリアル番号(サイズ:1~4バイト)を書き込みます。スタート番号、インクリメントサイズ、シリアル番号サイズ及びアドレスを J-Flash プロジェクトにて設定可能です。
- Flasher 本体にロードされているシリアル番号リストファイルに従ってカスタムシリアル番号を書き込み ます。シリアル番号リストファイル内のスタート番号を示す行番号、インクリメントする行数、シリアル番 号サイズ及びアドレスを J-Flash プロジェクトにて設定可能です。シリアル番号リストファイルの作成・指 定はユーザーにて行います。

ご注意点:

現在、シリアル番号プログラミング機能はスタンドアロンモードの場合のみ使用可能です。

● シリアル番号の設定

J-Flash プロジェクト設定ダイログの「Program serial number」項目オプションにチェックを入れて、以下の 画面のようにスタンドアロンモードでシリアル番号のプログラミング操作に必要なコンフィグレーション設定 を行います。

Project settings	? 💌
General Target Interface MCU Flash Production Performance	
Target power supply Reference voltage settings	
Power source NDNE	
Delay before start 20 ms min. 1000 mV max. 5500 m	V
Discharge target on disconnect Fixed VTref 3300 m	V
Program serial number	
Address 08001000 Len 4	
Next SN 1234567 Increment	
Actions performed by "Production Programming"	
Erase selected sectors Erase 15000 ms	
Program Program 10000 ms	
Venty Venty 10000 ms	
ОК	キャンセル 適用(<u>A</u>)



設定項目名	解説
Address	シリアル番号を保存する先のアドレス値を指定します。
Len	シリアル番号のデータサイズ(バイト数)を設定します。
	シリアル番号リストファイルを使用しない場合:
	1~4 バイトのシリアル番号は指定可能です。Len 項目に「8」が設定されている場合は、
	指定アドレスにシリアル番号に加えてそのシリアル番号の補数も書き込みます。
	シリアル番号リストファイルを使用する場合:
	リストファイルから Len 項目に指定されているバイト数のシリアル番号を読み出します。
	リストファイルに Len 項目値より短いサイズのシリアル番号が設定されている場合は、シ
	リアル番号の残りのデータを補数値ではなく「0」に設定します。
Next SN	シリアル番号リストファイルを使用しない場合:
	「Next SN」項目は、次にフラッシュへ書き込むシリアル番号の値を示します。
	フラッジュメモリに保存するシリアル番号のメモリバイトオーターはリトルエンディア ンとなります。
	シリアル番号リストファイルを使用する場合:
	「Next SN」項目は、次に書き込むシリアル番号が指定されている、リストファイル内の
	行番号を示します。リストファイルの初めの行からシリアル番号を書き込む場合は、「Next
	SN」 頃日に「U」 を改正してく/こさい。
Increment	「Next SN」項目値のインクリメント数を設定します。

● シリアル番号ファイル

J-Flash ツールの「File→Download serial number file to Flasher」メニューコマンド実行により、 「<JFlashProjectName>.jflash」プロジェクトファイルと同じフォルダに J-Flash プロジェクト設定の内容を ベースに「Next SN」項目値を指定した「 <JFlashProjectName>_Serial.txt 」ファイルが作成されます。この シリアル番号ファイルはFlasher 本体ディスクに「 SERIAL.TXT 」ファイルとして保存されます。 シリアル番号ファイルは ASCII エンコードデータファイルでユーザーにて編集可能です。

シリアル番号リストファイル

連続にインクリメントする 1~4 バイト数のシリアル番号の標準設定方法とは別に、シリアル番号リストファイルにて不連続式のシリアル番号のカスタム設定方法もサポートします。

「File→Download serial number file to Flasher」メニューコマンド実行の時に J-Flash ツールがプロジェクトファイル「 <JFlashProjectName>.jflash」と同じフォルダでシリアル番号リストファイル「 <JFlashProjectName>_SNList.txt 」を探します。このリストファイルが存在する場合は、Flasher 本体ディスクに「 SNList.TXT 」ファイルとして保存されます。

ユーザー側で作成するシリアル番号リストファイル「 <JFlashProjectName>_SNList.txt 」の構文は以下の通りです。

• 1行に1つのシリアル番号を記述します。

・ シリアル番号の各バイトは2桁の16進数で記述します。

事例:)

以下、「0x08000000」アドレスに8バイト数のシリアル番号をプログラムする(書き込む)場合の事例を示しま

ớ EmblTeK

す。	
シリアル番号のスタート値:	0102030455667788
フラッシュメモリ上の配置:	0x08000000: 0x01 0x02 0x03 0x04 0x55 0x66 0x77 0x88

シリアル番号リストファイル:

Test_SNList.txt - Notepad		
<u>F</u> ile <u>E</u> dit F <u>o</u> rmat <u>V</u> iew	<u>H</u> elp	
0102030455667788 010203045566778A 010203045566778B 010203045566778C 010203045566778D 010203045566778E 010203045566778F 0102030455667791 0102030455667792 0102030455667793 0102030455667793 0102030455667795		~

シリアル番号リストファイルの行ごとに読み込むバイト数は、J-Flash プロジェクト設定ダイログの「Program serial number」項目の「Len」オプションで設定します。

次のプログラミングサイクルで書き込むシリアル番号が記述されている行番号は、J-Flash プロジェクト設定ダイログの「Next SN」項目で設定します。リストファイルの初めの行からシリアル番号を書き込む場合は、「Next SN」項目に「0」を設定してください。

ご注意点:

リストファイルにLen項目値より短いサイズのシリアル番号が設定されている場合は、シリアル番号の残りのデータを「0」に設定されます。

また、リストファイルにLen項目値より長いサイズのシリアル番号が設定されている場合は、Lenサイズ以降の残りのバイトデータが無視されます。

シリアル番号プログラミングプロセス

書き込み成功する各プログラミングサイクル後に、Flasher は、「SERIAL.TXT」のシリアル番号を「Increment」 項目で定義された値だけインクリメントします。各プログラミングサイクル後に、「SERIAL.TXT」のシリアル番 号情報を記録して Flasher 本体上の「FLASHER.LOG」ログファイルが更新されます。

■ ご注意点:

シリアル番号プログラミング機能は無効に設定されている場合も、「SERIAL.TXT」のシリアルのシリアル番号が インクリメントされます。J-Flash プロジェクトにてシリアル番号プログラミング機能を有効に設定しない限り、 ターゲットフラッシュにシリアル番号の書き込みを行いません。

● Flasher にシリアル番号ファイルのダウンロード

Flasher 本体にシリアル番号ファイルをロードする際は、J-Flash ツールの「File→Download serial number file to Flasher」 メニューコマンドを実行してください。

■ ご注意点:

「File→Download config & data file to Flasher」メニューコマンドでは J-Flash プロジェクトのコンフィグ レーション設定及びデータファイルが Flasher 本体にロードされますが、シリアル番号ファイルは本体ディスク にロードされません。

● サンプル事例

以下の簡単なサンプルでシリアル番号プログラミングの設定方法を解説します。 1234567(0x120687)で始まる4バイト数のシリアル番号を0x08001000アドレスにプログラミングします。

<u>Step-1:パラメータの設定</u>

Project settings 🔋 🕱			
General Target Interface MCU Flash Production Performance			
- Target power supply	Reference voltage settings		
Power source NONE	Enable VTref monitor		
Delay before start 20 ms	min. 1000 mV max. 5500 mV		
Discharge target on disconnect	Fixed VTref 3300 mV		
Program serial number Address 08001000 Len 4 Next SN 1234567 Increment 1			
Actions performed by "Production Programmi	ng" Override timeouts		
✓ Program	<u>P</u> rogram 10000 ms		
l ✓ ⊻erify	⊻erify 10000 ms		
Start application Via SFR access	_		



パラメータ	設定値	解説
Address	08001000	アドレス→ 0x08001000
Next SN	1234567	1234567 (0x12D687)
Increment	1	1バイト
Len	4	4バイト

Step-2:コンフィグレーション設定、データ及びシリアル番号のダウンロード

J-Flash プロジェクトに必要なコンフィグレーション設定(ターゲット I/F、フラッシュ構成等)、データファイルの設定後に、J-Flash ツールの以下のメニューコマンドでコンフィグレーション設定値、データ及びシリアル番号を Flasher 本体にダウンロードします。

- \cdot File \rightarrow Download config & data file to Flasher
- \cdot File \rightarrow Download serial number file to Flasher

LOG Jownloading serial number file to emulator - Downloading "Serial.txt" (4 bytes) - Connecting via USB to J-Link device 0 - J-Link firmware: J-Link ARM / Flasher ARM V4 compiled Oct 18 2019 15:32:10 - Downloading (4 bytes, CRC: 0xDF795F9C) - File downloaded successfully	
<	

シリアル番号を Flasher ヘロードする時に、J-Flash ツールは「 <JFlashProjectName>_Serial.txt 」ファイル を作成します。

*	名前	更新日時	種類	サイズ
	empower.jflash	2019/11/10 18:01	JFLASH ファイル	2 KB
	emPower_K66.bin	2018/09/24 14:48	BIN ファイル	622 KB
	empower_Serial.txt	2019/11/11 11:23	text file	1 KB

12. セキュア(オーソライズド)モードの設定

ハードウェアバージョン4以降のFlasher ARM は、MSD モードでマウントした場合や RS232 ターミナルでの ASCII コマンドで本体ディスクに保存されているデータの盗み出しが出来ないオーソライズド(セキュア)フラッシュ 書き込み機能をサポートします。社外や海外の工場での生産書き込みの場合は、オーソライズドモードはより安 心です。

セキュアエリアの作成・追加

デフォルトでは、Flasher 本体はすべてのディスク領域はパブリックエリアとして設定された状態(MSD や RS232 ターミナルから本体のディスク全体がアクセス可能)で出荷されます。オーソライズドモードの設定には、セキ ュアエリア「secure area」の作成・設定が必要です。オーソライズドモードを設定しますと、Flasher 本体デ ィスクの半分の領域(現在のモデルは 64MB まで)はセキュアエリアとしてリザーブします。

セキュアエリアを削除しますとすべてのディスク領域はパブリックエリアと設定します。

J-Link ソフトウェアパッケージの「J-Link Commander」ツールから以下のコマンドでセキュアエリアの作成及 び削除は可能です。

- securearea create
- securearea remove



ご注意点:

セキュアエリアの作成及び削除を行う場合は、Flasher本体に保存されているコンフィグレーションプロジェクト設定及びデータファイルが削除されますのでデータを事前にバックアップしてください。



データのセキュア化

コンフィグレーション設定及びデータファイルをセキュアエリアにコピーする前に必ず事前にスタンドアロン モードでフラッシュ書き込み動作を確認してください。動作確認後に、以下の順でコンフィグレーション設定及 びデータファイルをセキュアエリアに移動します。

● Flasher 本体を MSD モードでパソコンに接続します。



- パソコンにマウントされた Flasher 本体ディスクに「**_SECURE**」フォルダを作成します。
- セキュア化するコンフィグレーション設定及びデータファイルを「_SECURE」フォルダに移動します。



● Flasher 本体を再起動します(MSD モードで再起動しますとデータはセキュアエリアにまだ移動されませんのでご注意ください)。再起動後にデータがセキュアエリアに移動されます。移動処理完了後に Flasher 本体の Power/Ready LED を点灯します。セキュアエリアに移動するデータサイズにより Flasher 本体の Power/Ready LED の点灯まで少し時間がかかります。データがセキュアエリアに移動されますと Flasher 本体のパブリックエリアからデータ及び「_SECURE」フォルダが削除されます。その後、Flasher はスタンドアロンモードで操作可能です。

セキュアモード操作の注意事項

- 通常のスタンドアロンモード操作での複数イメージファイルサポート、パッチファイルサポート等の機能は オーソライズドモードでも使用可能です。
- セキュアエリアに移動されたデータを本体から読み出すことは出来ません。オーソライズドモードで動作す る場合であっても、操作ログファイル「FLASHER.LOG」だけはパブリックエリアに配置され、更新されま す。
- パブリックエリアに「FLASHER.LOG」ログファイル以外に他のファイルやフォルダが作成された場合は、 オーソライズドモードでのスタンドアロンモード操作が失敗します。その際、エラー情報が「FLASHER.LOG」 ログファイルに記述されます。Flasher-ARMの場合は、TELNET/RS232 コンソール上にエラーログが表示され ます。
- パブリックエリアからセキュアエリアへのファイル移動は、何度も行うことができます。セキュアエリアに データ(ファイス数)が移動される度、最初にセキュアエリアのすべての内容は消去されます。



13. CRC チェックサム表示について

Flasher ツールは、チェックサム計算は「CRC32-CCITT」式を使用します。 CRC チェックサム方法について以下 URL の情報をご参照ください。 <u>https://wiki.segger.com/CRC_calculation</u> <u>https://wiki.segger.com/CRC</u>

CPC32_CCITT nolynomial	Normal form: 0x04C11DB7
	Reversed form: 0xEDB88320

J-Flash プロジェクトで選択されているすべてのセクタアドレス空間のデータに対して CRC チェックサムを計算 します。選択されているセクタアドレス空間範囲で、書き込み対象アプリケーションデータイメージの未定デー 夕及び未使用アドレス空間のデータは「0xFF」として計算します。

Project settings		? <mark>- x -</mark>
General Target Interface MCU Fla	ash Production Performance	
FlashBank Bank 0 (Internal program fla	ash) 💌 Rename Add Delete	
Use custom <u>R</u> AMCode		
Base Addr 00000000	Organization 32 🖃 Bits x 1 🚽 Chip(s)	
Info Internal flash bank 0 (Static	: sector layout)	
Manufacturer NXP		
Device MK66FN2M0xxx18		
Size 2048 KB Sectors	s 512	
C Start/End sector Start Addr Sector[0]: 0x0 ▼ End Addr Sector[511]: 0x1FFFFF ▼ Selected ranges: 512 Sectors, 1 Range: 0x0 - 0x1FFFFF	Individual sectors Sector(0): 0x0 - 0xFFF Sector(1): 0x1000 - 0x1FFF Sector(2): 0x2000 - 0x2FFF Sector(3): 0x3000 - 0x3FFF Sector(4): 0x4000 - 0x4FFF Sector(5): 0x5000 - 0x5FFF Sector(6): 0x6000 - 0x6FFF Sector(6): 0x600 - 0x6FFF	
Disable flash bank		
	OK キャン/セル	適用(A)
		ALCONT NUMBER

J-Flash ツールでの CRC 表示

J-Flash プロジェクトにアプリケーションイメージデータファイルをロードする時に、ログウインドウに以下の ようにイメージファイルの CRC 結果が表示されます。



J-Flash ツールにはチェックサム値の出力用メニューコマンドはありません。

J-Flash ツールのターゲットメニュー「Target → Manual Programming → Program & Verify」コマンドを実行 しますと、Verify 結果ダイアログボックス及びログウインドウに書き込んだデータの全フラッシュセクタ領域 の CRC 計算結果が表示されます。





スタンドアロンモードでの CRC 表示

「File→Download config & data file to Flasher」コマンドで、Flasher ARM をスタンドアロンモードに設定 する時に以下のように Flasher 本体にロードする CFG、DAT ファイルの CRC 値は LOG ウインドウに表示されます。

LOG	
Downloading files to emulator - Downloading <u>"Flasher.cfg" (4096 bytes)</u> - Connecting via USB to J-Link device 0 - J-Link firmware: J-Link PRO / Flasher PRO V4 compiled Oct 18 2019 15:35:11 - Downloading (4096 bytes, CRC: 0x6C3A560D) - File downloaded successfully	
HER LOG	
 Preparing DAT file range: 0x00097800 - 0x00097FFF (2048 bytes). Preparing DAT file range: 0x00098000 - 0x000987FF (2048 bytes). Preparing DAT file range: 0x00098000 - 0x00098FFF (2048 bytes). Preparing DAT file range: 0x00099000 - 0x000997FF (2048 bytes). Preparing DAT file range: 0x00099800 - 0x00099FFF (2048 bytes). Preparing DAT file range: 0x00098000 - 0x00098FFF (2048 bytes). Preparing DAT file range: 0x00098000 - 0x00099FFF (2048 bytes). Preparing DAT file range: 0x0009A000 - 0x0009A7FF (2048 bytes). Preparing DAT file range: 0x0009A800 - 0x0009AFFF (2048 bytes). Preparing DAT file range: 0x0009B000 - 0x0009B4B7 (1208 bytes). Preparing DAT file range: 0x0009B000 - 0x0009B4B7 (1208 bytes). Preparing DAT file range: 0x0009B000 - 0x0009B4B7 (1208 bytes). 	

Flasher 本体に保存する DAT ファイル(FLASHER. DAT 等)はユーザーアプリケーションデータ以外にフラッシュ 書き込み用アルゴリズムコードデータ及びタイムスタンプ情報も含まれています。そのため、J-Flash ツールの 「Program & Verify」ターゲットメニューコマンドで表示される CRC 値と DAT ファイルの CRC 値は異なります。

スタンドアロンモードで RS232 コンソールから「#FCRC」コマンドで Flasher 本体に保存されているデータファ イル(DAT、CFG ファイル)の CRC チェックサム値の確認は可能です。







また、以下の手順で書き込みコマンド毎にログファイルに DAT ファイルの CRC 値を記録する設定は可能です。

Flasher ARM 本体を MSD モードでパソコンに接続して、ディスクフォルダを開きます。 Flasher 本体ディスクの「FLASHER. INI」ファイルをテキストエディタで開き、 以下のように「ShowDatCRCAfterProgramming = "1"」コマンドを設定した CONFIG 項目を追加します。

[CONFIG] ShowDatCRCAfterProgramming = "1"

ディスクに「FLASHER. INI」ファイルが存在しない場合は、新規の「FLASHER. INI」ファイルを作成して、 以下のように FILES と CONFIG 項目を追加します。

[FILES] DataFile = "FLASHER.DAT" ConfigFile = "FLASHER.CFG" [CONFIG]

ShowDatCRCAfterProgramming = "1"

名前	更新日時	種類
퉬 FLASHER		ファイル フォル
FLASHER.CFG	2019/11/11 19:06	CFG ファイル
FLASHER.DAT	2019/11/11 19:06	DAT ファイル
FLASHER.LOG		LOG ファイル
		text file
FLASHER.INI	2019/11/11 19:20	構成設定



スタンドアロンモードでのフラッシュ書き込みコマンド実行後に正常に終了しました書き込みサイクルのログ に DAT ファイルの CRC 値も記録されます。



スタンドアロンモードのログファイルの確認方法

Flasher ARM 本体を MSD モードでパソコンに接続して、ディスクフォルダを開きます。 本体ディスクに保存されている「FLASHER.LOG」ファイルをテキストエディタで開き、ログ内容を確認します。

<u> </u>	名前	更新日時	種類
	🍌 FLASHER		ファイル フォル…
	FLASHER.CFG	2019/11/11 19:06	CFG ファイル
=	FLASHER.DAT	2019/11/11 19:06	DAT ファイル
	FLASHER.INI	2019/11/11 19:28	構成設定
	FLASHER.LOG		LOG ファイル
	SERIAL.TXT		text file

□ FLASHER.LOG - メモ帳	
ファイル(E) 編集(E) 書式(<u>Q</u>) 表示(⊻) ヘルプ(<u>H</u>)	
SN: 3 - O.K. (15061 ms) - Dat file crc 0xC68FC86B SN: 4 - O.K. (15047 ms) - Dat file crc 0xC68FC86B SN: 5 - O.K. (15033 ms) - Dat file crc 0xC68FC86B	*
	E