

# FAQ・テクニカルガイド

## 「J-Link Commander」 ツール

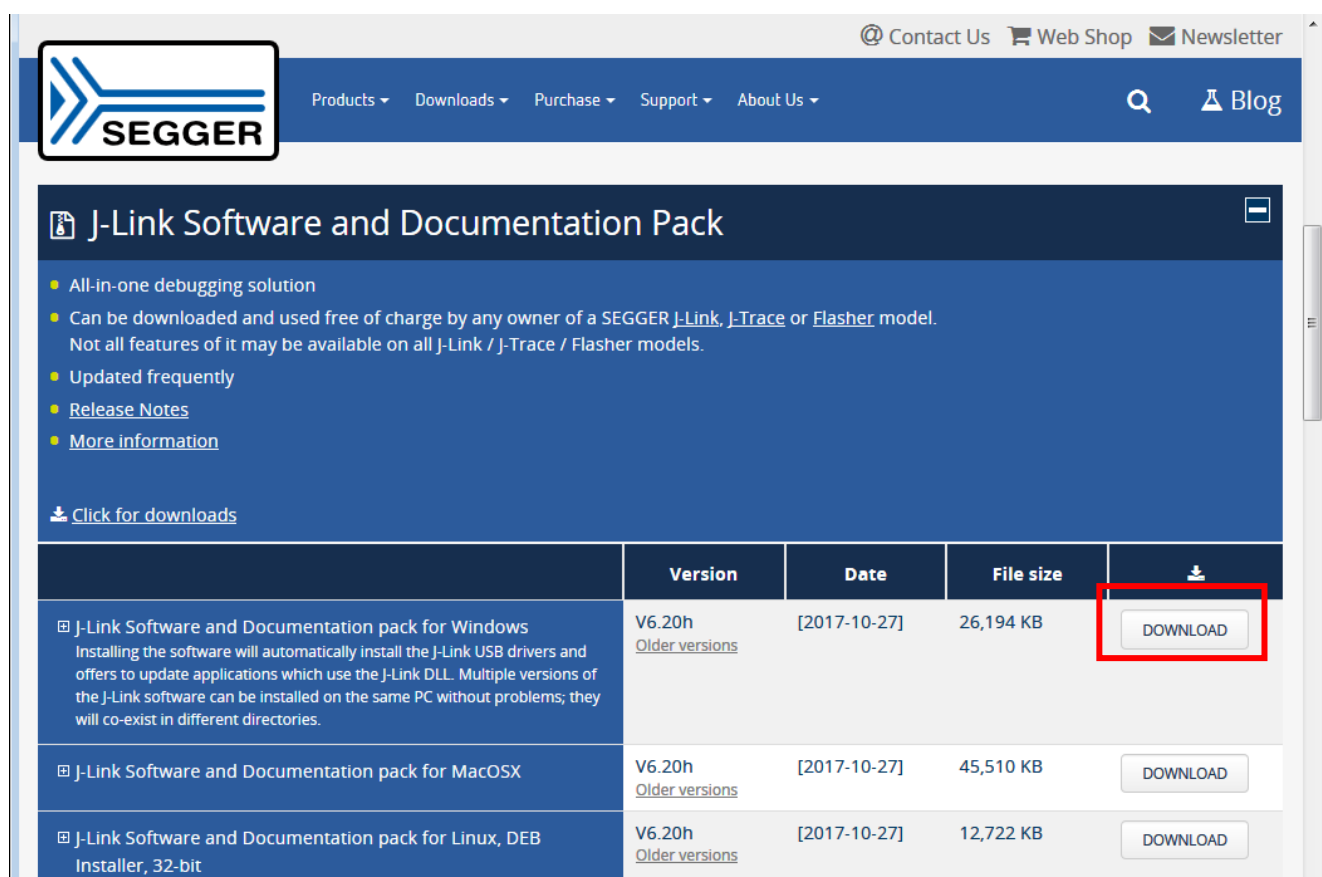
J-Link コマンドー (JLink.exe) は、USB ドライバのインストールやターゲット CPU との接続を検証だけでなく、ターゲット・システムの簡単な分析にも使用することができるツールです。また、ターゲットマイコンデバイスのメモリダンプ、CPU 停止、ステップ実行、実行などの簡単な動作確認が可能です。

### 1. 使用方法

#### J-Link ソフトウェアのインストール

J-Link/Flasher 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-Link ソフトウェア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/jlink>



SEGGER

Products ▾ Downloads ▾ Purchase ▾ Support ▾ About Us ▾

Q Blog

### J-Link Software and Documentation Pack

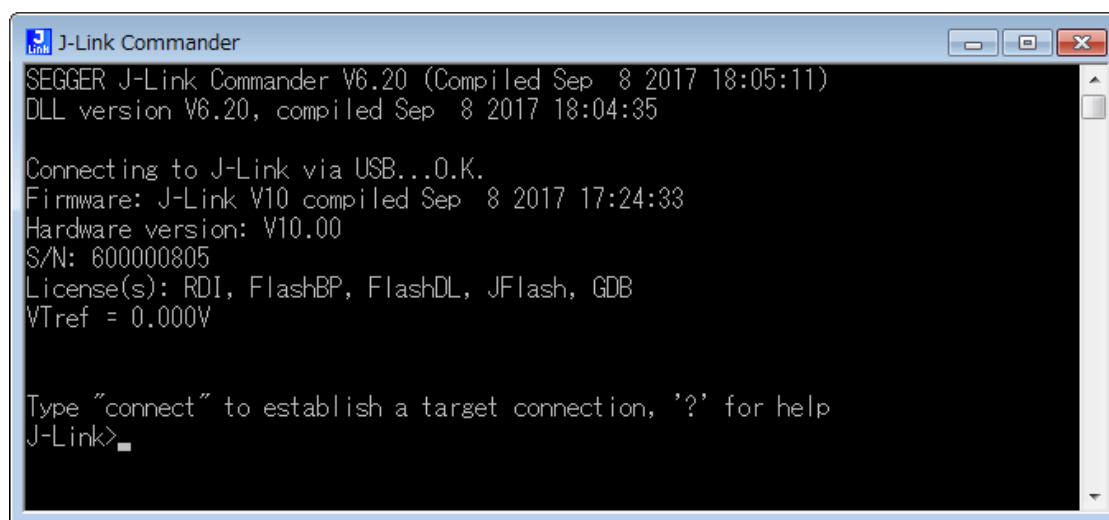
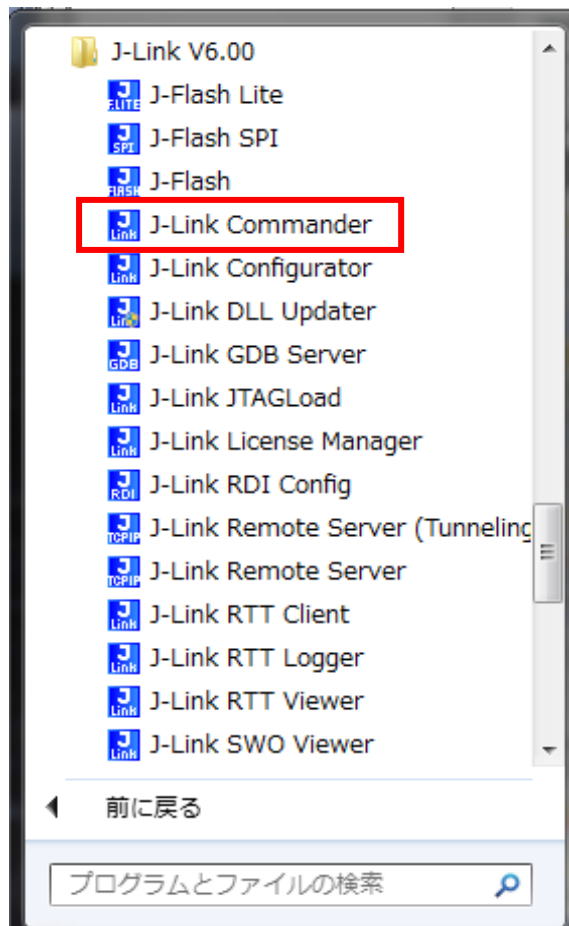
- All-in-one debugging solution
- Can be downloaded and used free of charge by any owner of a SEGGER J-Link, J-Trace or Flasher model.  
Not all features of it may be available on all J-Link / J-Trace / Flasher models.
- Updated frequently
- [Release Notes](#)
- [More information](#)

Click for downloads

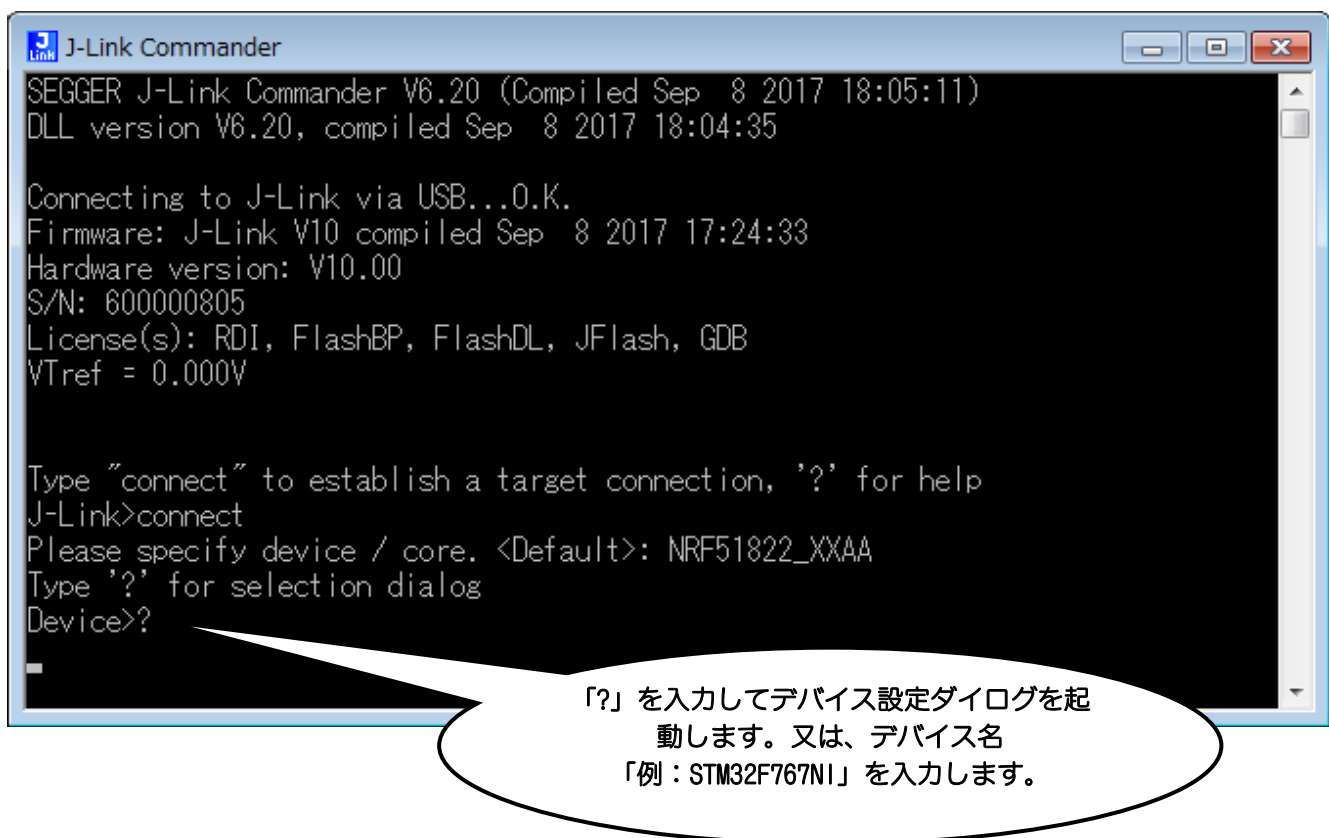
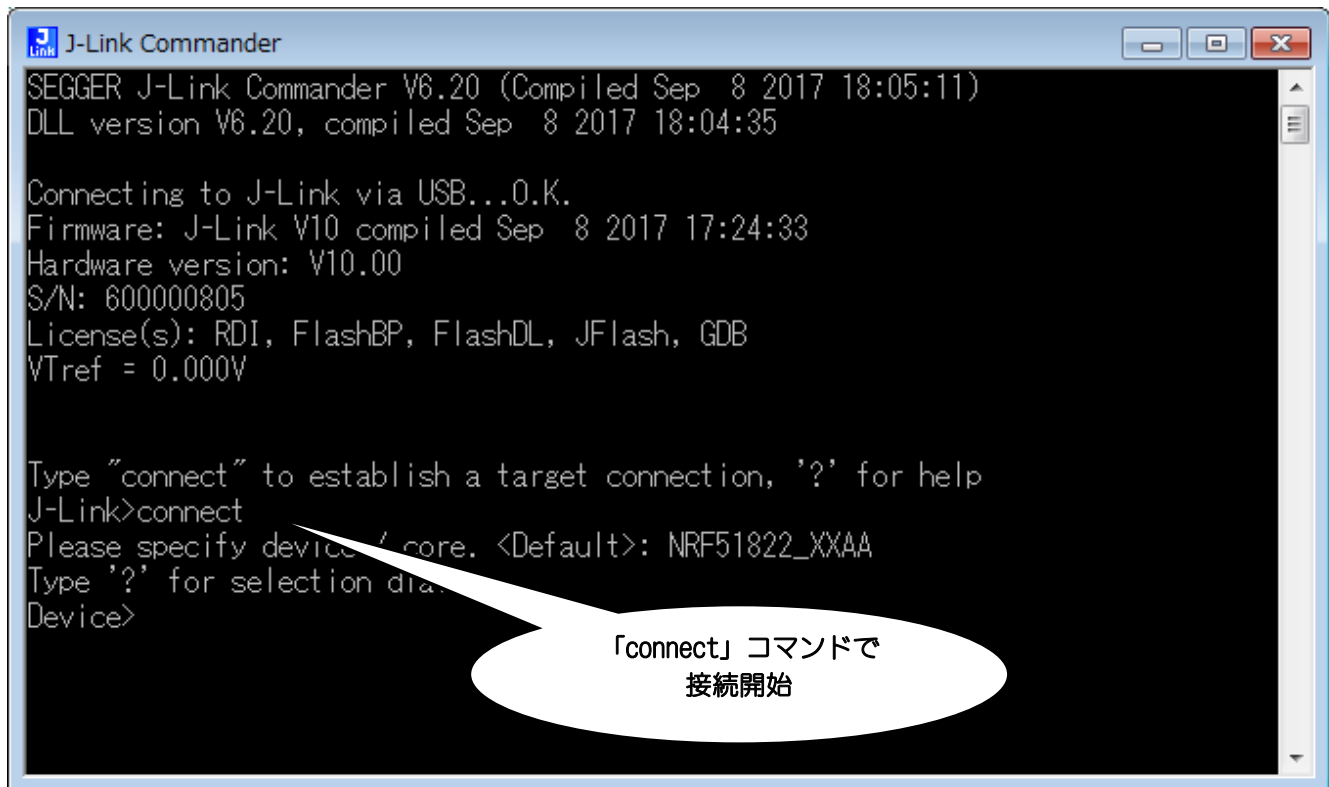
	Version	Date	File size	
<input checked="" type="checkbox"/> J-Link Software and Documentation pack for Windows Installing the software will automatically install the J-Link USB drivers and offers to update applications which use the J-Link DLL. Multiple versions of the J-Link software can be installed on the same PC without problems; they will co-exist in different directories.	V6.20h <a href="#">Older versions</a>	[2017-10-27]	26,194 KB	<a href="#">DOWNLOAD</a>
<input checked="" type="checkbox"/> J-Link Software and Documentation pack for MacOSX	V6.20h <a href="#">Older versions</a>	[2017-10-27]	45,510 KB	<a href="#">DOWNLOAD</a>
<input checked="" type="checkbox"/> J-Link Software and Documentation pack for Linux, DEB Installer, 32-bit	V6.20h <a href="#">Older versions</a>	[2017-10-27]	12,722 KB	<a href="#">DOWNLOAD</a>

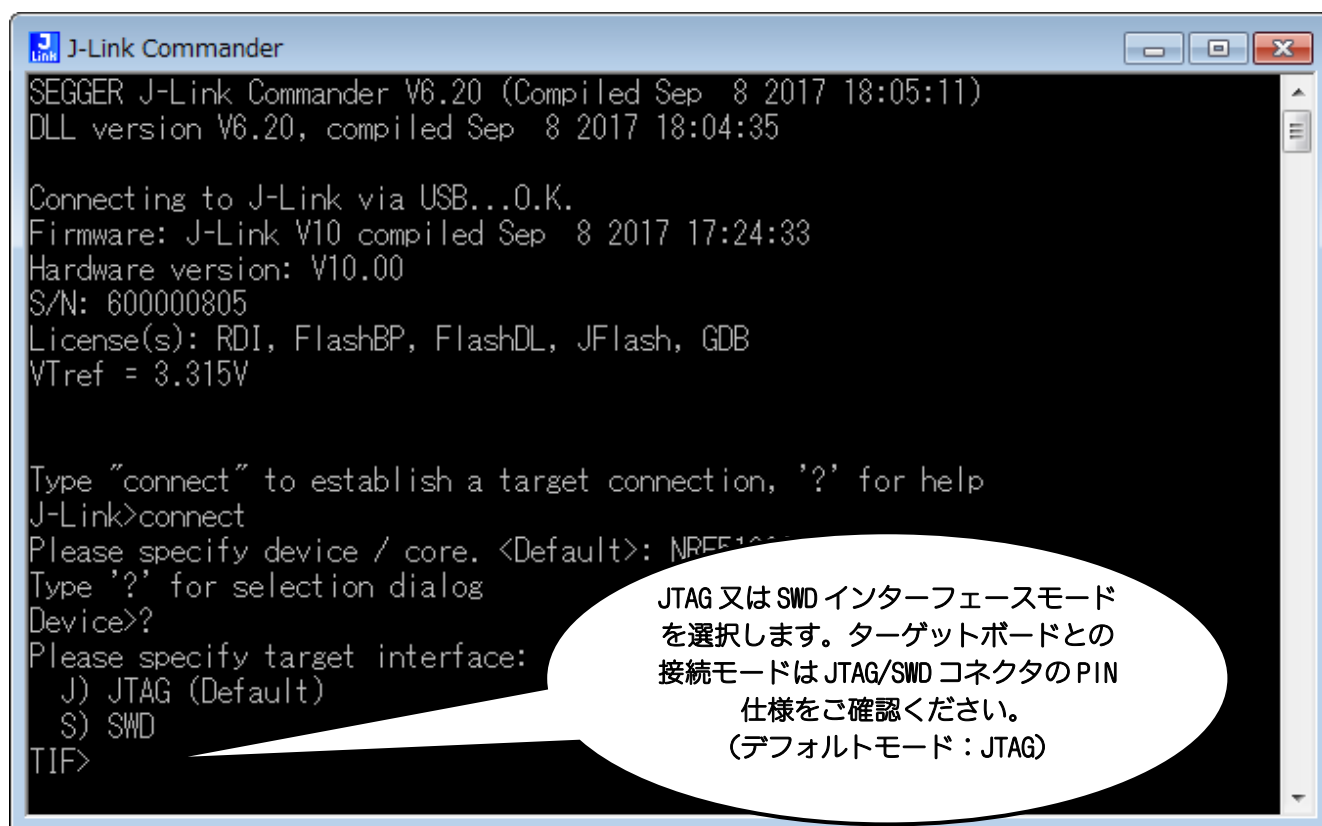
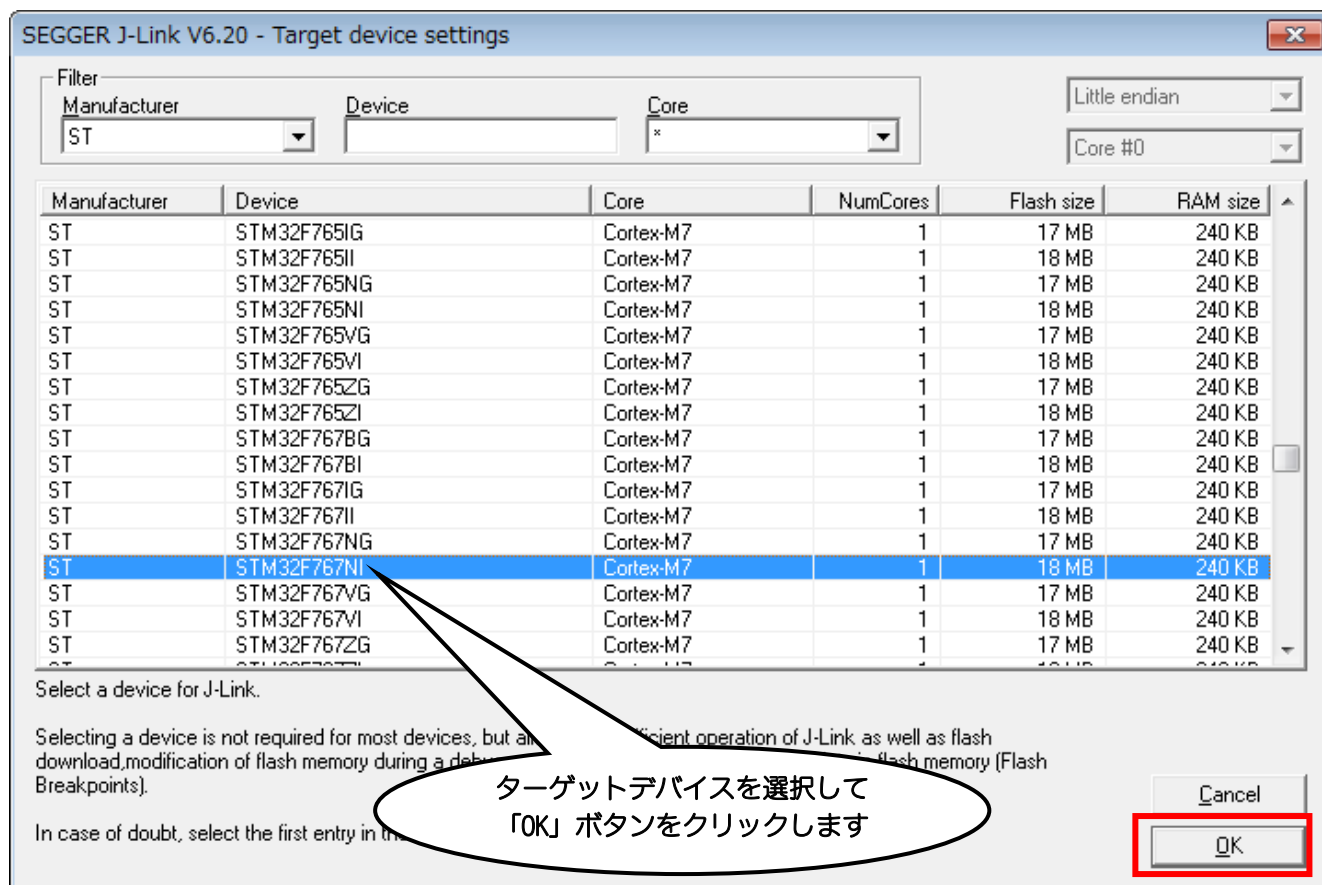
## 「J-Link Commander」ツールの起動

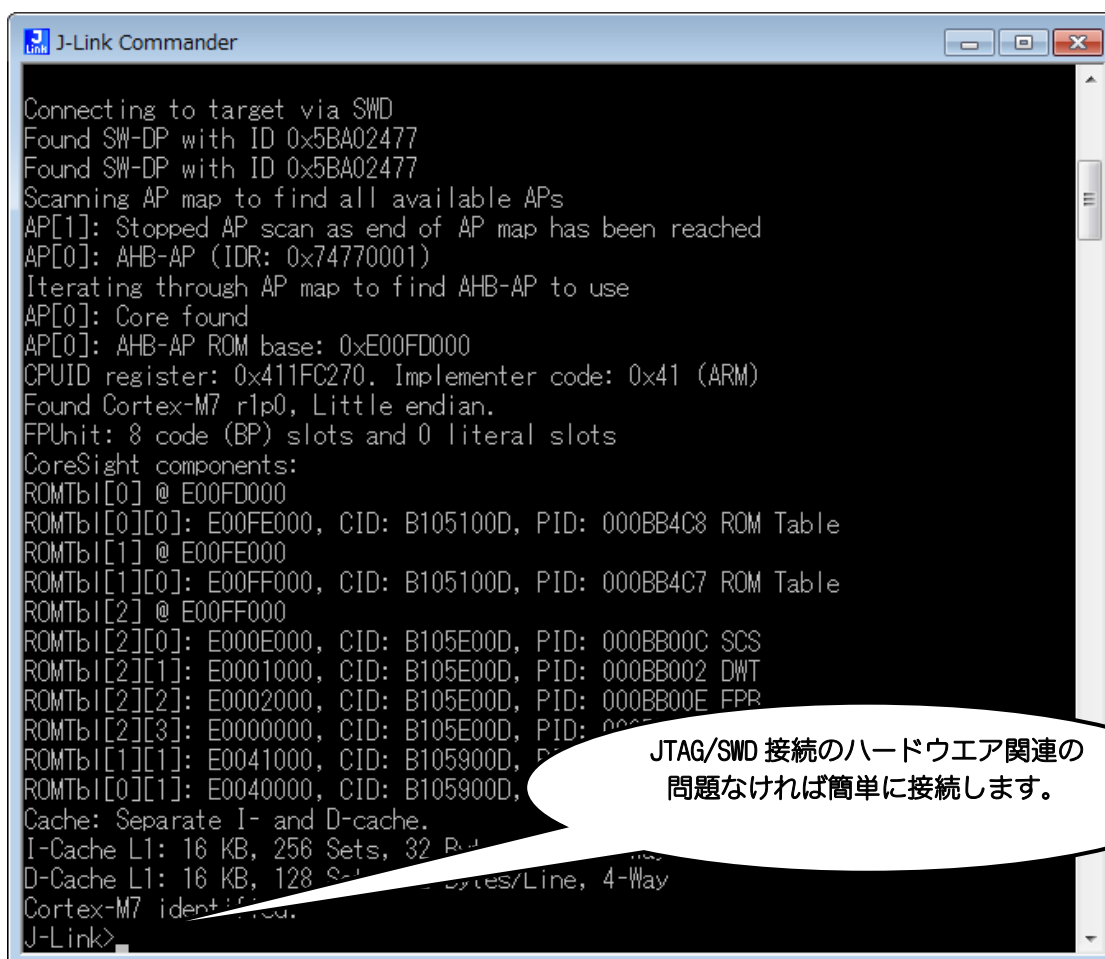
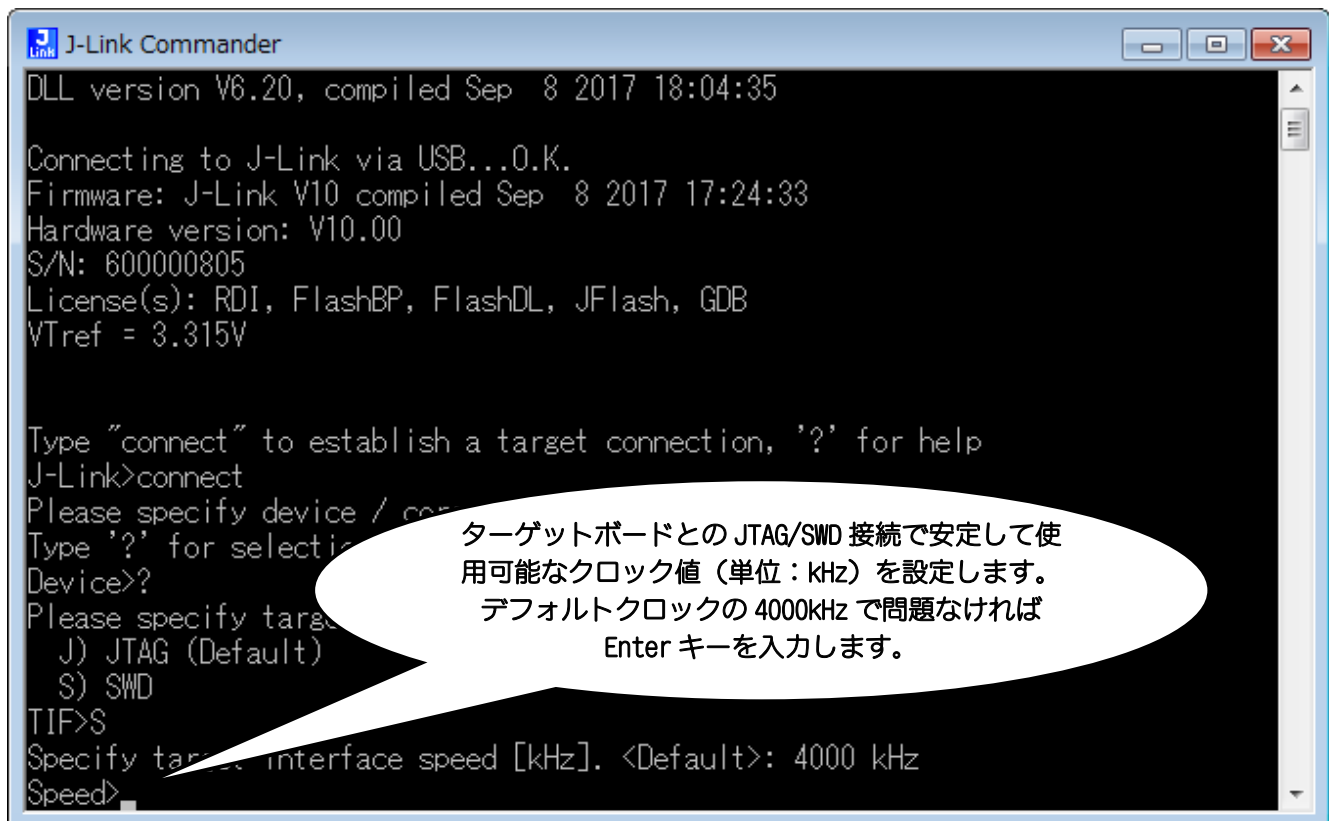
J-Link/Flasher 本体をパソコンに接続してパソコンのスタートメニューから「J-Link Commander」プログラムを起動します。

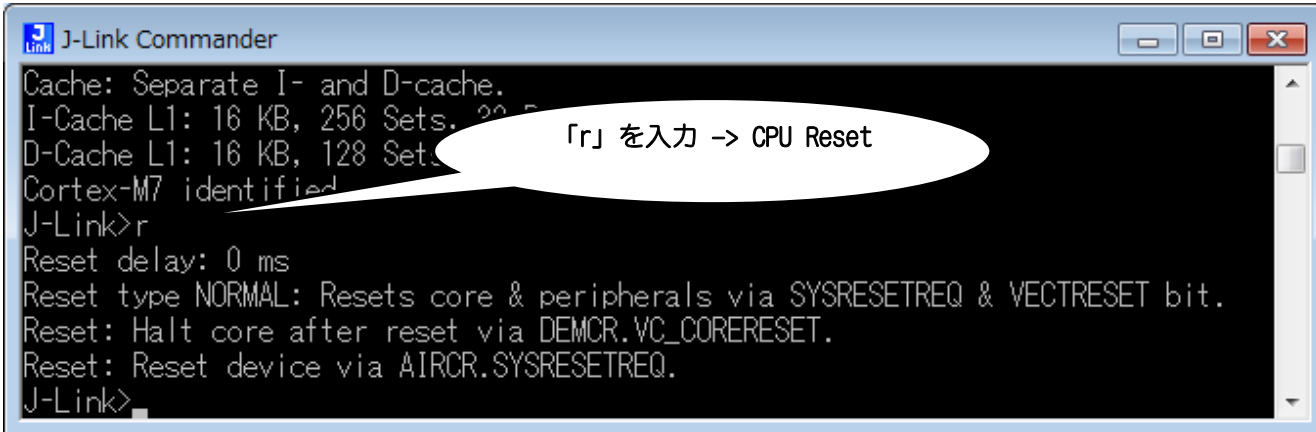


## ターゲットマイコンデバイスに接続





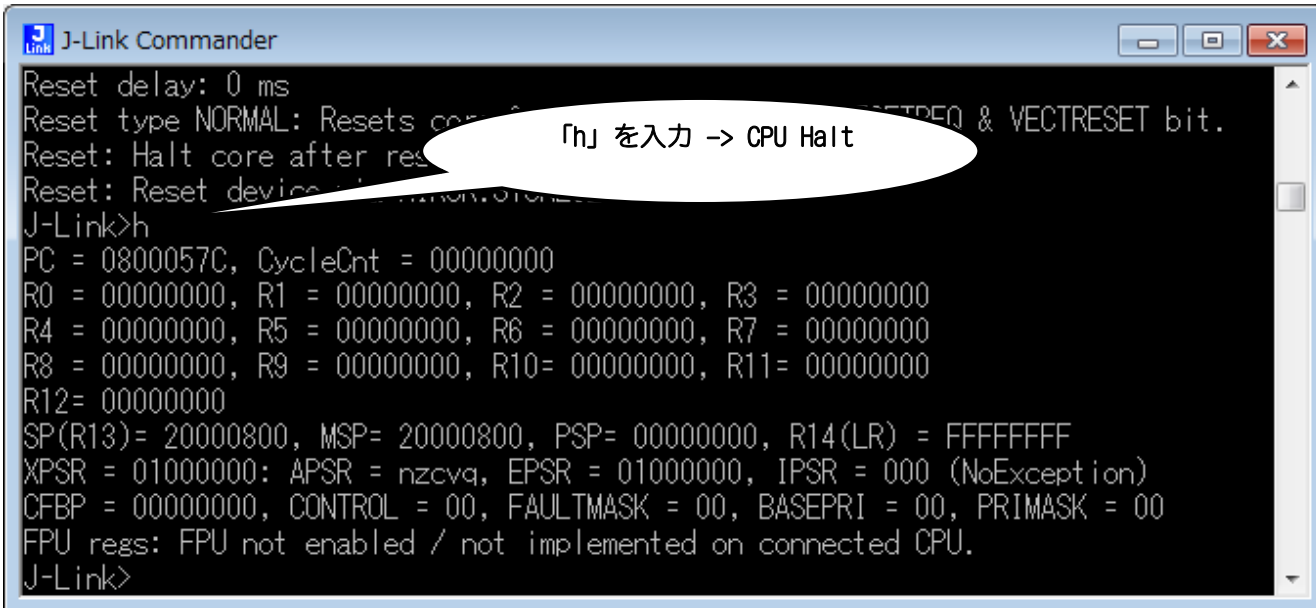




The screenshot shows the J-Link Commander window with the following text:

```
Cache: Separate I- and D-cache.  
I-Cache L1: 16 KB, 256 Sets, 20.5  
D-Cache L1: 16 KB, 128 Sets  
Cortex-M7 identified  
J-Link>r  
Reset delay: 0 ms  
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.  
Reset: Halt core after reset via DEMCR.VC_CORERESET.  
Reset: Reset device via AIRCR.SYSRESETREQ.  
J-Link>
```

A callout bubble points to the 'r' command, containing the text: 「r」を入力 -> CPU Reset

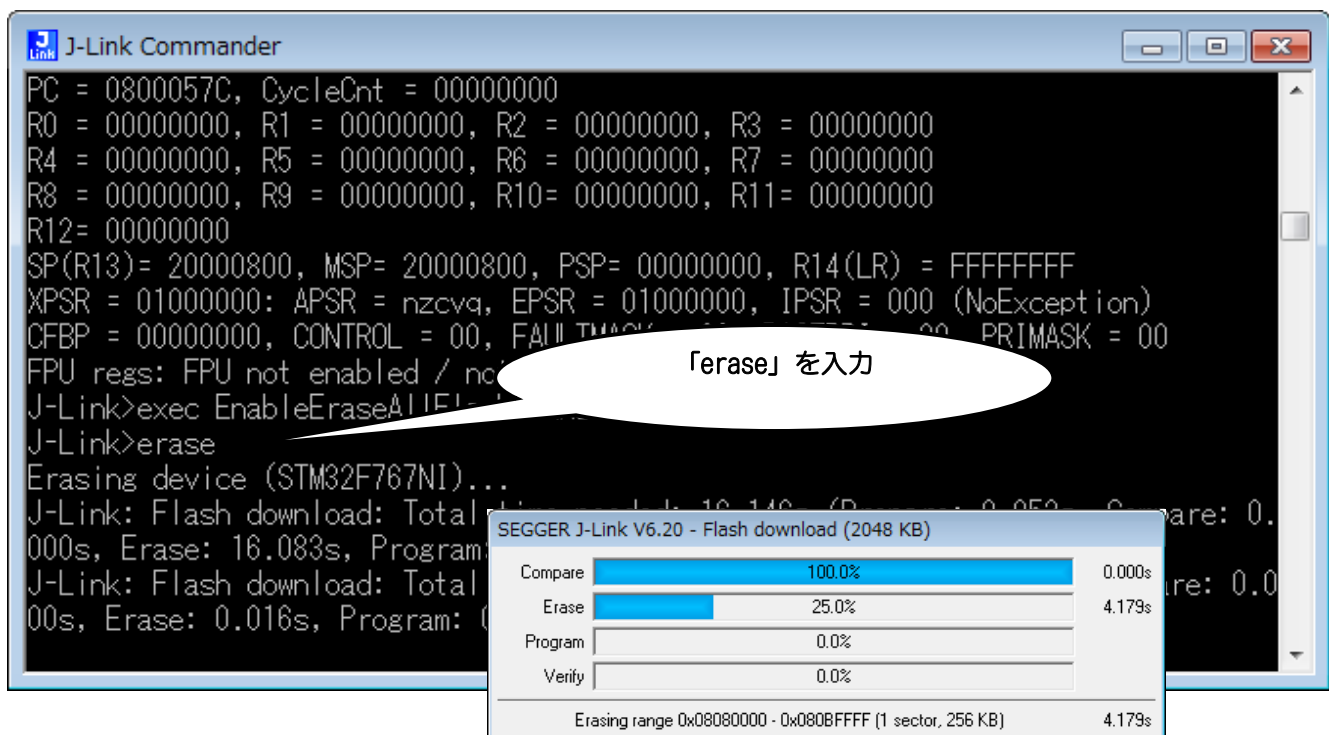
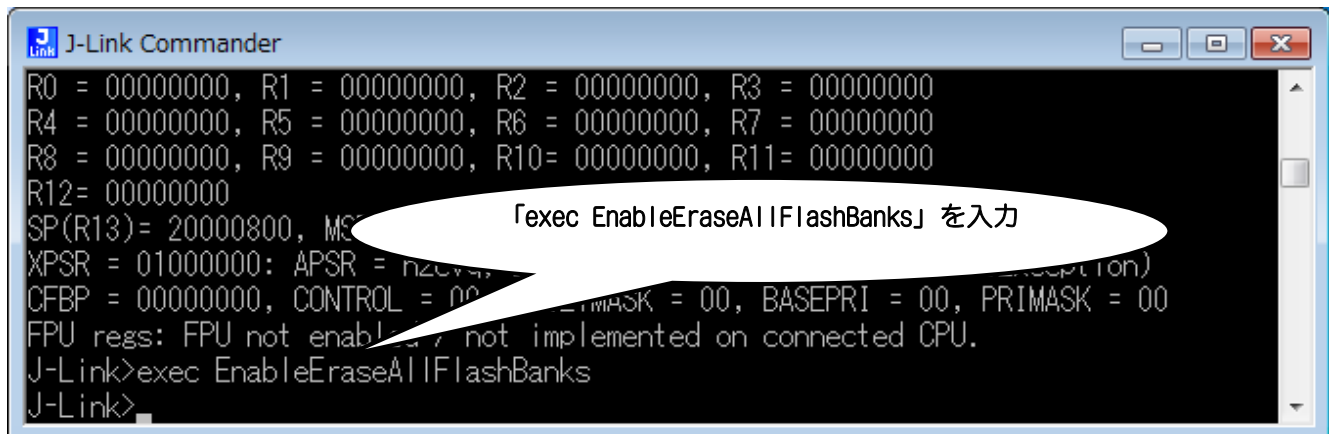


The screenshot shows the J-Link Commander window with the following text:

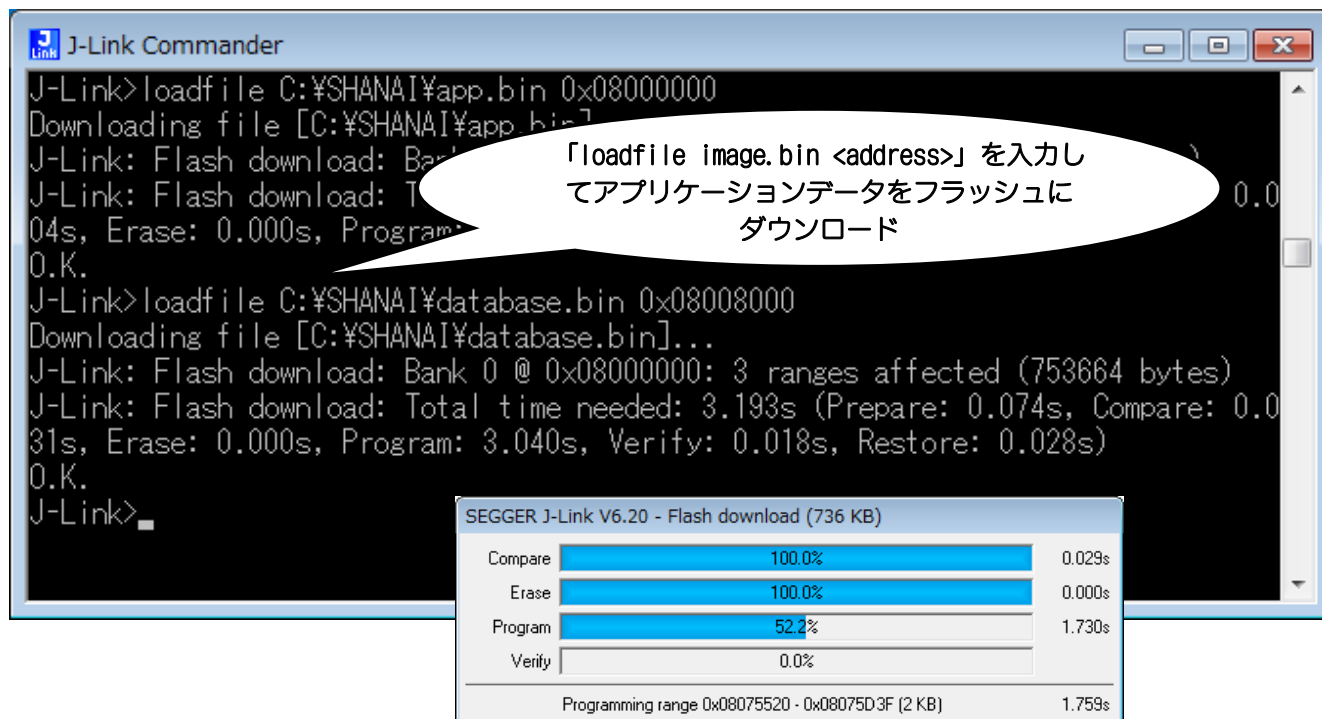
```
Reset delay: 0 ms  
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.  
Reset: Halt core after reset via DEMCR.VC_CORERESET.  
Reset: Reset device via AIRCR.SYSRESETREQ.  
J-Link>h  
PC = 0800057C, CycleCnt = 00000000  
R0 = 00000000, R1 = 00000000, R2 = 00000000, R3 = 00000000  
R4 = 00000000, R5 = 00000000, R6 = 00000000, R7 = 00000000  
R8 = 00000000, R9 = 00000000, R10= 00000000, R11= 00000000  
R12= 00000000  
SP(R13)= 20000800, MSP= 20000800, PSP= 00000000, R14(LR) = FFFFFFFF  
XPSR = 01000000: APSR = nzcvgq, EPSR = 01000000, IPSR = 000 (NoException)  
CFBP = 00000000, CONTROL = 00, FAULTMASK = 00, BASEPRI = 00, PRIMASK = 00  
FPU regs: FPU not enabled / not implemented on connected CPU.  
J-Link>
```

A callout bubble points to the 'h' command, containing the text: 「h」を入力 -> CPU Halt

## フラッシュを ERASE



## アプリケーションイメージのフラッシュ書き込み

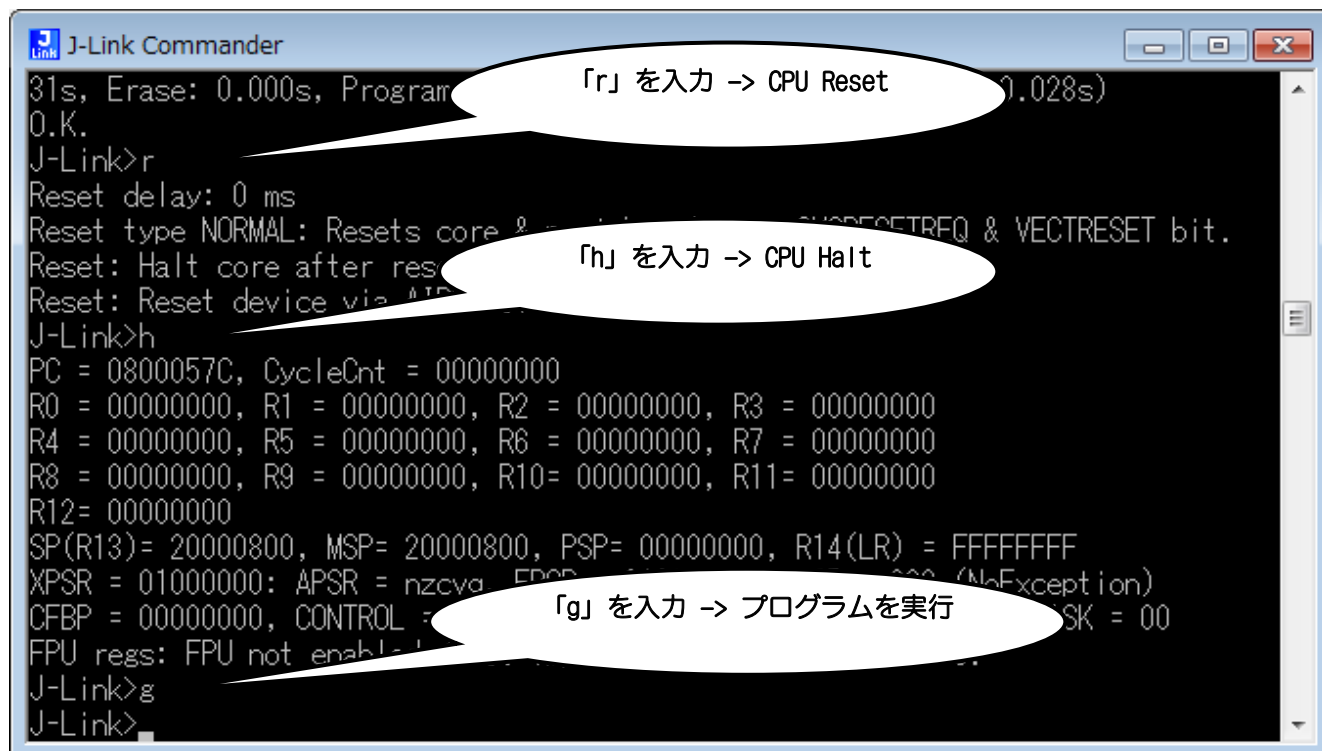


コマンド事例：

**loadfile image\_rom.srec**

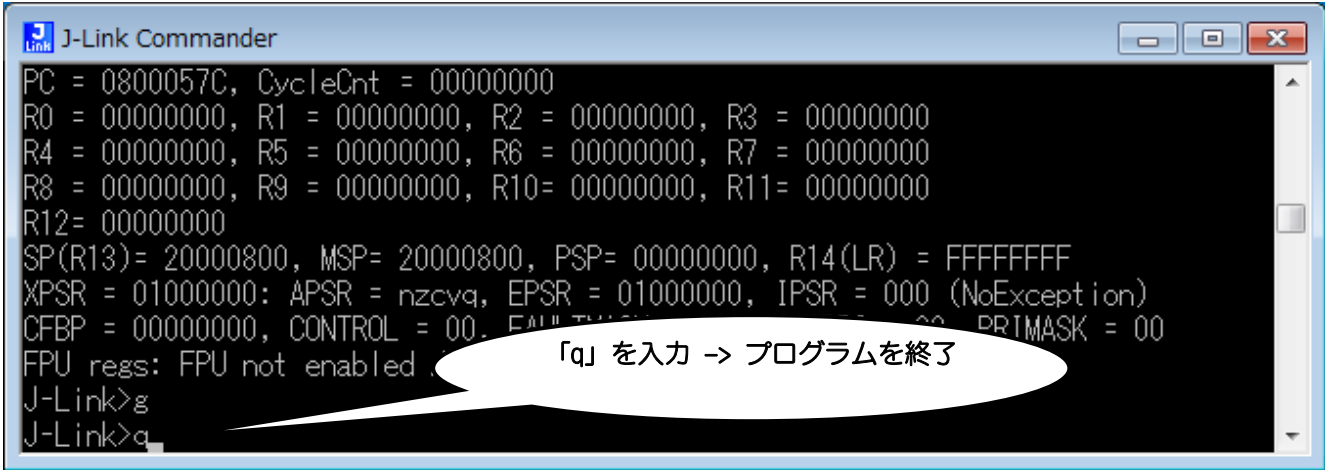
**loadfile image\_rom.hex**

**loadfile image\_rom.bin 0x18000000**





## 「J-Link Commander」 ツールを終了



The screenshot shows the J-Link Commander application window. The title bar reads 'J-Link Commander'. The main text area displays the following information:

```
PC = 0800057C, CycleCnt = 00000000
R0 = 00000000, R1 = 00000000, R2 = 00000000, R3 = 00000000
R4 = 00000000, R5 = 00000000, R6 = 00000000, R7 = 00000000
R8 = 00000000, R9 = 00000000, R10 = 00000000, R11 = 00000000
R12 = 00000000
SP(R13) = 20000800, MSP = 20000800, PSP = 00000000, R14(LR) = FFFFFFFF
XPSR = 01000000: APSR = nzcvcq, EPSR = 01000000, IPSR = 000 (NoException)
CFBP = 00000000, CONTROL = 00. FAULTMASK = 00, PRIMASK = 00
FPU regs: FPU not enabled
J-Link>g
J-Link>q
```

A white callout bubble with a black border points to the 'J-Link>q' command line. Inside the bubble, the text reads: 「q」を入力 -> プログラムを終了

※ 「J-Link Commander」 ツールの各コマンドの詳細はユーザーマニュアルをご参照ください。すべてのコマンドを 1 つの BAT ファイルとして実装可能ですので、1 クリックでの BAT コマンド実行も可能です。

## 2. コマンド一覧

以下の表は、J-Link コマンダーの利用可能なコマンドを示しています。すべてのコマンドは、それぞれのカテゴリ内でアルファベット順に記載されています。コマンドの詳細については各項目を参照してください。

コマンド	説明
Basic	
clrBP	ブレイクポイントをクリアにします。
clrWP	ウォッチポイントをクリアにします。
device	デバイスを選択します。
erase	選択したデバイスの内蔵フラッシュを消去します。
exit (qc, q)	J-Link コマンダーを終了します。
exitonerror (eoe)	J-Link コマンダーをエラー後に終了します。
f	ファームウェアの情報を出力します。
go (g)	CPU コアを開始します。
halt (h)	CPU コアを停止します。
hwinfo	ハードウェア情報を表示します。
is	チェーン選択レジスタの長さを表示します。
loadfile	データファイルをターゲットメモリにロードします。
log	ファイルに対してログを有効にします。
mem	メモリをロードします。
mem8	8bit のアイテムをロードします。
mem16	16bit のアイテムをロードします。
mem32	32bit のアイテムをロードします。
mem64	64bit のアイテムをロードします。
mr	RTCK ピンの応答時間を計測します。
ms	スキャンチェーンの長さを計測します。
power	ターゲットの電源を切り替えます。
r	ターゲットをリセットし、停止します。
regs	現在のレジスタ値すべてを表示します。
rnh	ターゲットを停止せずに、リセットします。
rreg	特定のレジストリ値を表示します。
rx	遅れてターゲットをリセットします。
savebin	バイナリファイルにターゲットメモリを保存します。
setBP	ブレイクポイントを設定します。
setPC	指定された値に PC を設定します。
setWP	ウォッチポイントを設定します。
sleep	指定された時間（ミリ秒単位）待機します。
speed	ターゲット・インターフェース速度を設定します。
st	現在のハードウェアの状態を表示します。
step (s)	ターゲットチップを1ステップ進めます。
unlock	デバイスのロックを解除します。
verifybin	データファイルとメモリを比較します。
w1	8bit のアイテムを書き込みます。
w2	16bit のアイテムを書き込みます。

w4	32bit のアイテムを書き込みます。
wreg	レジスタを書き込みます。
Flasher I/O	
fdelete (fdel)	エミュレータ上のファイルを削除します。
flist	エミュレータ上のディレクトリを並べます。
fread (frd)	エミュレータからファイルを読み取ります。
fshow	エミュレータからファイルを読み取り、表示します。
fsize (fsz)	エミュレータ上のファイルサイズを表示します。
fwrite (fwr)	エミュレータにファイルを書き込みます。
Connection	
ip	TCP/ IP を経由して J-Link Pro に接続します。
usb	USB を経由して J-Link Pro に接続します。

## clrBP

このコマンドは、J-Link で設定されたブレイクポイントを削除します。

形式： clrBP <BP\_Handle>

パラメータ	説明
BP_Handle	削除するブレイクポイントのハンドル。

例： clrBP 1

## clrWP

このコマンドは、J-Link で設定されたウォッチポイントを削除します。

形式： clrWP <WP\_Handle>

パラメータ	説明
WP_Handle	削除するウォッチポイントのハンドル。

例： clrWP 0x2

## device

このコマンドは、J-Link に接続する特定のデバイスを選択し、再接続を行います。ほとんどの場合、デバイスを個別に選択する必要はありません。デバイスを選択すると、J-Link flash のプログラミング機能とフラッシュメモリにおける無制限のブレイクポイントを使用できます。

デバイスの必要とする特定の処理を DLL が実行するために、一部のデバイスでは個別にデバイスの選択が必要です。一部のコマンドを使用する前にデバイスを設定してください。

形式： device <DeviceName>

パラメータ	説明
DeviceName	有効なデバイス名：デバイスが選択されています。 ？：デバイス選択ダイアログを表示します。

例： device stm32f407ig

## erase

このコマンドは、現在のデバイスのすべてのフラッシュセクターを消去します。事前に、デバイスを指定してください。

形式： erase

## exit (qc, q)

このコマンドは、ターゲット接続と J-Link への接続を終了して、J-Link コマンダーを終了します。

形式： q

## exitonerror (eoe)

このコマンドは、J-Link コマンダー終了時のエラー有無に関わらず、トグルスイッチでオンにします。

形式： exitOnError <1|0>%

パラメータ	説明
<1 0>	1: J-Link コマンダーをエラー終了します。 0: J-Link コマンダーをエラー終了しません。

例： eoe 1

## f

このコマンドは、ファームウェアとハードウェアのバージョン情報を出力します。機能セットに影響を及ぼさないために、マイナーなハードウェアの修正が非表示になる可能性があります。

形式： f

## fdelete (fdel)

このコマンドは、ファイル I/O をサポートするエミュレータで特定のファイルを削除します。

形式： fdelete <FileName>]

パラメータ	説明
FileName	Flasher から削除するファイル。

例： fdelete Flasher.dat

## flist

このコマンドは、ファイル I/O をサポートするエミュレータで Flasher のディレクトリツリーを表示します。

形式： flist

## fread (frd)

このコマンドは、ファイル I/O をサポートするエミュレータで、特定のファイルを読み取ります。オフセットは送信先と送信元のファイルの両方に適用されます。

形式： fread <EmuFile> <HostFile> [<Offset> [<NumBytes>]]

パラメータ	説明
EmuFile	フォームを読み取るファイル名。
HostFile	ホスト上の宛先ファイル。
Offset	データの読み取りが開始するファイル内のオフセットを指定します。
NumBytes	読み取る最大バイト数。

例： fread Flasher.dat C:\Project\Flasher.dat

## fshow

このコマンドは、ファイル I/O をサポートするエミュレータで、特定のファイルを読み取り、出力します。現在 Flasher のモデルのみがファイル I/O をサポートしています。

形式： fshow <FileName> [-a] [<Offset> [<NumBytes>]]

パラメータ	説明
EmuFile	Flasher から読み取るソースファイル名。
A	設定すると、入力内容は 16 進数で表示される代わりにテキストとして解析されます。
Offset	データの読み取りが開始するファイル内のオフセットを指定します。
NumBytes	読み取る最大バイト数。

例： fshow Flasher.dat

## fsize (fsz)

このコマンドは、ファイル I/O をサポートするエミュレータで、特定のファイルサイズを取得します。現在 Flasher のモデルのみがファイル I/O をサポートしています。

形式： fsize <FileName>]

パラメータ	説明
EmuFile	Flasher から読み取るソースファイル名。

例： fsize Flasher.dat

## fwrite (fwr)

このコマンドは、ファイル I/O をサポートするエミュレータで、特定のファイルを編集します。現在 Flasher のモデルのみがファイル I/O をサポートしています。

NumBytes には、一度に 512 バイトの制限があります。例えば、e. g. 1024 バイトを編集する場合、再送信時に適切なオフセットを使用して、コマンドを 2 回送信する必要があります。オフセットは、送信先と送信元のファイルの両方に適用されます。

形式： fwrite <EmuFile> <HostFile> [<Offset> [<NumBytes>]]

パラメータ	説明
EmuFile	フォームを読み取るファイル名。
HostFile	ホスト上の宛先ファイル。
Offset	データの読み取りが開始するファイル内のオフセットを指定します。
NumBytes	読み取る最大バイト数。

例： fwrite Flasher.dat C:\¥Project¥Flasher.dat

## go (g)

このコマンドは、CPU を起動します。ブレイクポイントの設定を回避するために、シミュレート/エミュレートする命令の最大数を定義してください。プログラムがフラッシュに配置され、フラッシュブレイクポイントが使用されている場合は特に有用です。命令をシミュレートすると、フラッシュの再プログラムが回避され、（単一の）ステッピングを高速化します。

形式： go [<NumSteps> [<Flags>]]

パラメータ	説明
NumSteps	命令の最大数はシミュレートすることができます。ブレイクポイント命令にヒットするたび、エミュレート・シミュレートできない命令がヒットしたか、NumSteps が達したときに、命令シミュレーションは停止します。
Flags	0：BP が NumSteps の範囲内にある場合、CPU を起動しないでください。 1：BP をオーバーステップしてください。

例： go //Simply starts the CPU  
go 20, 1

## halt (h)

このコマンドは、CPU コアを停止します。成功した場合、現在の CPU のレジスタを表示します。

形式： halt

## hwinfo

このコマンドは、ターゲットの消費電力に関する情報を取得します（ターゲットが J-Link を介して電力供給されている場合）。また、過電流が起こった場合は情報を提供します。

形式： hwinfo

## ip

このコマンドは、J-Link への既存の接続を終了し、TCP / IP を経由して新しい接続を開始します。IP アドレスを指定しなかった場合は、エミュレータの選択ダイアログが表示されます。

形式： ip [<Addr>]

パラメータ	説明
Addr	有効な値： IP Address：指定された IP アドレスと J-Link を接続します。 Host Name：ホスト名を解決し、それに接続します。 *：エミュレータ選択ダイアログを呼び出します。

例： ip 192.168.6.3

## is

このコマンドは、スキャンチェーン選択レジスタの長さについての情報を返します。

形式： is

## loadfile

このコマンドは、指定された宛先アドレスに、指定されたデータファイルをプログラムします。現在サポートされているデータファイルは次のとおりです。

- \*.mot
- \*.srec
- \*.s19
- \*.s
- \*.hex
- \*.bin

形式： loadfile <Filename> [<Addr>]

パラメータ	説明
Filename	ソースファイル名。
Addr	宛先アドレス（*.bin のファイルに必要）。

例： loadfile C:\¥Work¥test.bin 0x20000000

## log

このコマンドは、DLL のログ情報出力を可能にするログファイルにパスを設定します。  
ログファイルが既に存在する場合、現在のログファイルの内容が上書きされます。

形式： log <Filename>

パラメータ	説明
Filename	ログファイル名。

例： log C:\Work\log.txt

## mem

このコマンドは、ターゲット・システムからメモリを読み込みます。必要に応じて、ターゲット CPU はメモリを読み出すために停止します。

形式： mem [<Zone>:]<Addr>, <NumBytes> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
NumBytes	読み取るバイト数（最大バイト数は 0x10000）。

例： mem 0, 100

## mem8

このコマンドは、バイト単位でターゲット・システムからメモリを読み込みます。必要であれば、ターゲット CPU がメモリを読み取るために停止します。

形式： mem8 [<Zone>:]<Addr>, <NumBytes> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
NumBytes	読み取るバイト数（最大バイト数は 0x10000）。

例： mem8 0, 100



## mem16

このコマンドは、16 ビット単位でターゲット・システムからメモリを読み込みます。必要であれば、ターゲット CPU がメモリを読み取るために停止します。

形式： mem16 [<Zone>:]<Addr>, <NumBytes> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	アドレスの開始。
NumBytes	読み取るバイト数（最大バイト数は 0x8000）。

例： mem16 0, 100

## mem32

このコマンドは、32 ビット単位でターゲット・システムからメモリを読み込みます。必要であれば、ターゲット CPU がメモリを読み取るために停止します。

形式： mem32 [<Zone>:]<Addr>, <NumBytes> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
NumBytes	読み取るバイト数（最大バイト数は 0x4000）。

例： mem32 0, 100

## mem64

このコマンドは、64 ビット単位でターゲット・システムからメモリを読み込みます。必要であれば、ターゲット CPU がメモリを読み取るために停止します。

形式： mem64 [<Zone>:]<Addr>, <NumBytes> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
NumBytes	読み取るバイト数（最大バイト数は 0x4000）。

例： mem64 0, 100

## mr

このコマンドは、RTCK ピンの反応時間を測定します。

形式： mr [<RepCount>]

パラメータ	説明
RepCount	テストが繰り返される回数（デフォルト：1）。

例： mr 3

## ms

このコマンドは、指定されたスキャンチェーンのビット数を測定します。

形式： ms <ScanChain>

パラメータ	説明
ScanChain	測定されたスキャンチェーン。

例： ms 1

## power

このコマンドは、TAG コネクタの 19 ピンを超える電源の状態を設定します。J-Link の KS(Kickstart)バージョンは、デフォルトで有効になった 19 ピンを超える 5V の電源を持っています。この機能は、JTAG コネクタを介して電力を供給することができる一部のターゲットに有効です。

形式： power <State> [perm]

パラメータ	説明
State	有効な値：On、Off
Perm	デフォルトとして指定された状態の値を設定します。

例： f

## r

このコマンドは、ターゲットをリセットし、停止します。

形式： r

## regs

このコマンドは、現在のレジスタ値をすべて表示します。

形式： regs

## rnh

このコマンドは、デバイスを停止することなくリセットを実行します。

形式： rnh

## rreg

このコマンドは、指定された CPU レジスタの値を出力します。

形式： rreg <RegIndex>

パラメータ	説明
<a href="#">RegIndex</a>	読み込むレジスタ。

例： rreg 15

## rx

このコマンドは、ターゲットをリセットし、停止します。リセット後にミリ秒単位の遅延時間を定義することができます。この機能は、既にアプリケーションかブートローダーが含まれている一部のターゲット・デバイスに有効です。例えば、メモリ管理ユニット（MMU）または外部バスインターフェースのようなハードウェアを初期化するため、コアの停止前に時間を必要とします。

形式： rx <DelayAfterReset>

パラメータ	説明
<a href="#">DelayAfterReset</a>	ミリ秒単位の遅延。

例： rx 10

## savebin

このコマンドは、バイナリファイルにターゲットメモリを保存します。

形式： savebin <Filename>, <Addr>, <NumBytes> (hex)

パラメータ	説明
Filename	リンク先のファイル。
Addr	送信元アドレス。
NumBytes	読み取るバイト数。

例： savebin C:\¥Work¥test.bin 0x0000000 0x100

## setBP

このコマンドは、指定したアドレスに特定の種類のブレイクポイントを設定します。どのブレイクポイントのモードが有効かは、使用している CPU によって決まります。

形式： setBP <Addr> [[A/T]/[W/H]] [S/H]

パラメータ	説明
Addr	ブレイクポイントが設定されたアドレス。
A/T	(以下2点は、ARM7/ 9/11 と Cortex-R4 デバイスのみに対するものです) A: ARM モード T: THUMB モード
W/H	(以下2点は、MIPS デバイスのみに対するものです) W: MIPS32 モード (Word) H: MIPS16 モード (Half-word)
S/H	S: フォースソフトウェア BP H: フォースハードウェア BP

例： setBP 0x8000036

## setPC

このコマンドは、指定された値に PC を設定します。

形式： setpc <Addr>

パラメータ	説明
Addr	PC が設定するアドレス。

例： setpc 0x59C

## setWP

このコマンドは、指定されたパラメータに一致する新しいウォッチポイントを挿入します。ウォッチポイントユニットのデータ・アクセス・ビットと同様に、ウォッチポイントのイネーブルビットは、このコマンドが自動的に設定します。また、ビット DBTEXT、CHAIN と RANGE ビット（他のものと 1 ウォッチポイントを接続するために使用されます）は自動的にマスクされます。これらのビットを使用するためには、直接 ICE レジスタを書き込むことによって、ウォッチポイントを設定する必要があります。

形式： setWP <Addr> [<AccessType>] [<Size>] [<Data> [<DataMask> [<AddrMask>]]]

パラメータ	説明
Addr	ウォッチポイントが設定されたアドレス。
Accesstype	データイベントが設定された制御データを指定します。 R：読み取りアクセス W：書き込みアクセス
Size	有効な値：S8   S16   S32 選択したアドレスの n ビットのアクセス幅を監視するように指定します。
Data	ウォッチポイントが設定されたデータを指定します。
DataMask	比較に使用するデータマスクを指定します。1 に設定されたビットはマスクされるため、データ比較の際に考慮されません。特定のコアのために、コアデバッグロジックによってサポートされるビットマスクの組み合わせがすべてではないことに注意してください。一部のコアでは、完全なバイトのみをマスク（例えば PIC32）または同様のことをします。
AddrMask	比較のために使用されるアドレスマスクを指定します。1 に設定されたビットはマスクされるため、データ比較の際に考慮されません。特定のコアのために、コアデバッグロジックによってサポートされるビットマスクの組み合わせがすべてではないことに注意してください。一部のコアでは、完全なバイトのみをマスク（例えば PIC32）または同様のことをします。

例： setWP 0x20000000 W S8 0xFF

## sleep

このコマンドは、（ミリ秒単位で）指定された時間待機します。

形式： sleep <Delay>

パラメータ	説明
Delay	ミリ秒単位でスリープ状態になる時間。

例： sleep 200

## speed

このコマンドは、CPU コアとの通信速度を設定します。

形式： speed <Freq>|auto|adaptive

パラメータ	説明
<a href="#">Freq</a>	kHz の内のインターフェース周波数を指定します。
<a href="#">Auto</a>	JTAG 速度の自動検出を選択します。
<a href="#">Adaptive</a>	JTAG 速度と適応クロッキングを選択します。

例： speed 4000  
speed auto

## st

このコマンドは、現在のハードウェアステータスを表示します。TCK、TDI、TDO、TMS、TRES、TRUST の状態とターゲットがサポートしているインターフェース速度を出力します。また、ターゲット電圧も表示されます。

形式： st

## step (s)

このコマンドを呼び出す前にターゲットを停止してください。

このコマンドは、ターゲット上で単一ステップを実行します。ブレイクポイントが設定されていても、命令をオーバーストップします。ステップを進める命令の[逆アセンブル](#)を出力します。

形式： step

## unlock

このコマンドは、ユーザーソフトウェアの誤動作によってロックされたデバイスの解除を行います。

形式： unlock <DeviceName>

パラメータ	説明
<a href="#">DeviceName</a>	ロックを解除するデバイス・ファミリーの名前。 (以下はサポートされるデバイスです) LM3Sxxx Kinetis EFM32Gxxx

例： unlock Kinetis

## usb

このコマンドは、J-Link への既存の接続を終了し、USB 経由で新しい接続を開始します。  
ポート番号によって特定 J-Link を選択することが可能です。

形式： usb [<Port>]

パラメータ	説明
Port	有効な値：0..3

例： usb

## verifybin

このコマンドは、指定されたバイナリが、指定アドレスのターゲット・メモリに既にあるかどうかを確認します。

形式： verifybin <Filename>, <Addr>

パラメータ	説明
Filename	サンプル bin。
Addr	検証するメモリの開始アドレス。

例： verifybin C:\¥Work¥test.bin 0x00000000

## w1

このコマンドは、ターゲット・システムに 1 バイト単位で書き込みます。

形式： w1 [<Zone>:]<Addr>, <Data> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
Data	書き込むデータの 8 ビット。

例： w1 0x10, 0xFF

## w2

このコマンドは、ターゲット・システムに 16 ビットの単位で書き込みます。

形式： w2 [<Zone>:]<Addr>, <Data> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
Data	書き込むデータの 16 ビット。

例： w2 0x0, 0xFFFF

## w4

このコマンドは、ターゲット・システムに 32 ビットの単位で書き込みます。

形式： w4 [<Zone>:]<Addr>, <Data> (hex)

パラメータ	説明
Zone	アクセスするメモリゾーンの名前。
Addr	開始アドレス。
Data	書き込むデータの 32 ビット。

例： w4 0x0, 0xAABBCCFF

## wreg

このコマンドは、レジスタに書き込みます。値は CPU 起動時にレジスタに書き込まれます。

形式： wreg <RegName>, <Data>

パラメータ	説明
RegName	書き込むレジスタ。
Data	指定されたレジスタに書き込むデータ。

例： wreg R14, 0xFF



## 株式会社エンビテック

代理店販売（デバッガ、RTOS、GUI、ミドルウェア）

<https://www.embitek.co.jp>

日本代理店

SEGGER Microcontroller GmbH



---

〒130-0021 東京都墨田区緑 4-8-8 中井ビル 4F

Phone: 03-6240-2655 Fax: 03-6240-2656

---

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、権利また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

「EmbITeK」、EmbITeK 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。