Quick Start Guide

This guide applies to UniFlash v4 and later. For all UniFlash releases, please see here (https://processors.wiki.ti.com/index.php /Category:CCS_UniFlash_Release_Notes_Archive).

You can chose to use the standalone desktop version, or the cloud version. The desktop version is available from https://www.ti.com/tool/uniflash (https://www.ti.com/tool/uniflash). The cloud version is available from the cloud tools portal (https://dev.ti.com (https://dev.ti.com)).

The first time you access a device from the Cloud using either UniFlash or the Cloud IDE, you will be prompted to install TI Cloud Agent. For more details you can refer to TI Cloud Agent (TI_Cloud_Agent).

Device-family specific guides are available for:

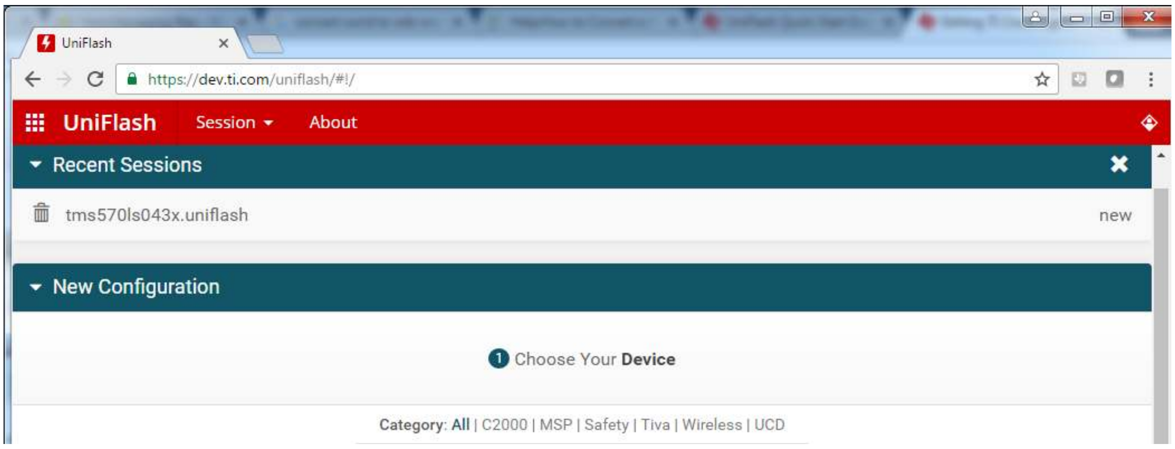
- IWR14xx/IWR16xx (https://processors.wiki.ti.com/images /f/f5/Mmwave_uniflash_user_guide_v1.0.pdf)
- CC32xx Image Creator (https://www.ti.com/lit/swru469)
- Sitara/K2G (https://software-dl.ti.com/processor-sdk-rtos/esd/docs/latest /rtos/index_board.html#uniflash)
- MSP and SimpleLink Bootloader Programming (https://www.ti.com /lit/SLAU799)

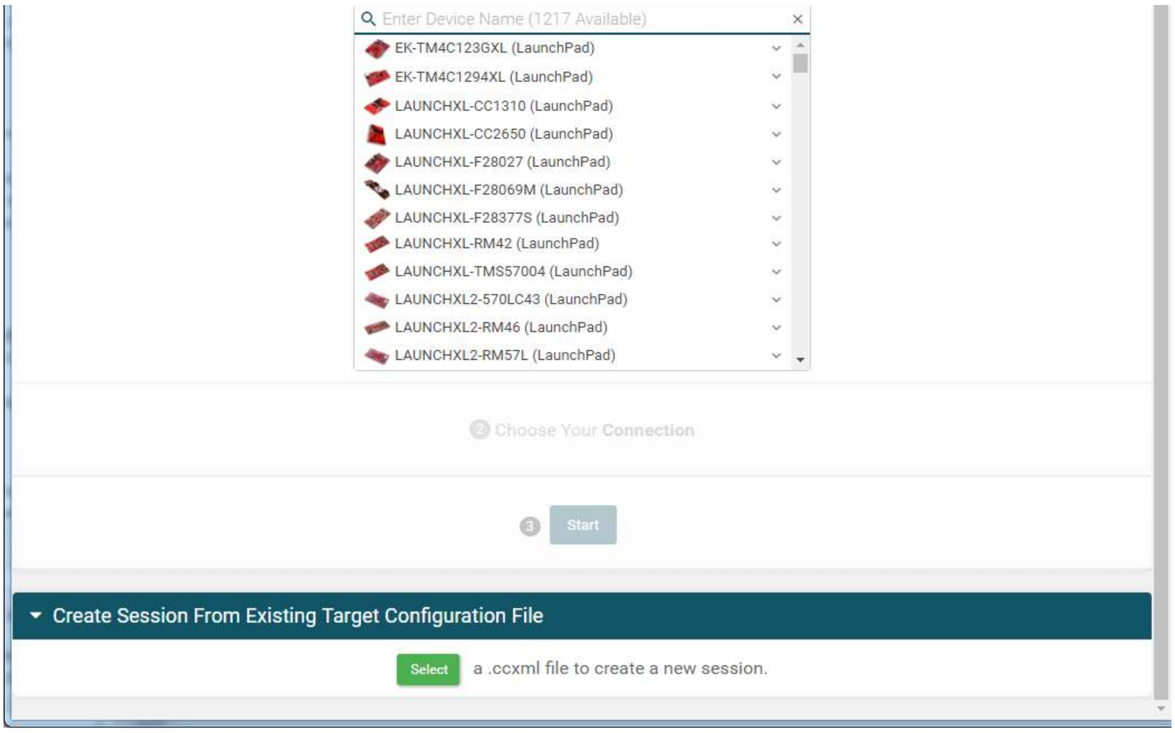
Session

To perform flash operations on your device, you will need to launch a flash session configured for that device. You can start a new session by:

- Selecting a Launchpad or device/connection combination. When you select a Launchpad, the connection is automatically choosen for you.
- Loading a previously saved session. Loading a session will restore your previous selection (Device, connection, and settings).
- Selecting a CCXML file created by CCS. This is useful if you have configuration settings in the CCXML file beyond a simple device and connection selection.

Start a new session from this page:



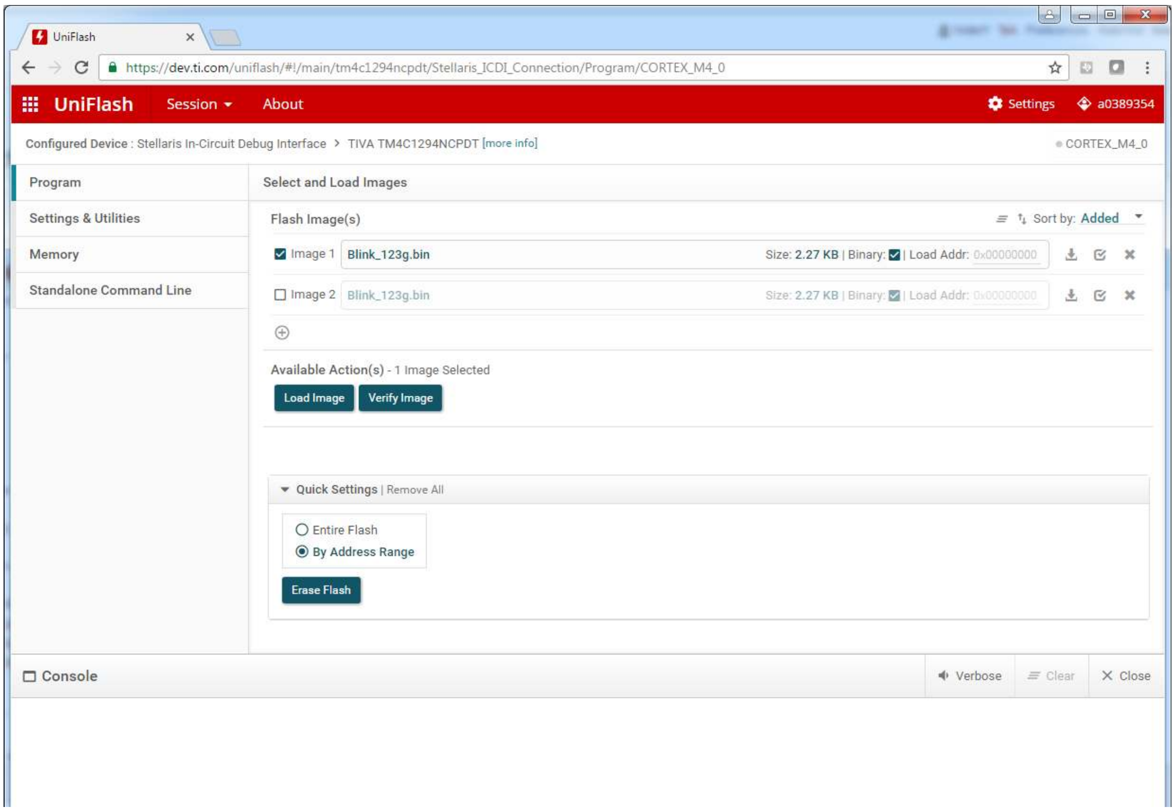

Enter Device Name (1217 Available)
EK-TM4C123GXL (LaunchPad)
EK-TM4C1294XL (LaunchPad)
LAUNCHXL-CC1310 (LaunchPad)
LAUNCHXL-CC2650 (LaunchPad)
LAUNCHXL-F28027 (LaunchPad)
LAUNCHXL-F28069M (LaunchPad)
LAUNCHXL-F28377S (LaunchPad)
LAUNCHXL-RM42 (LaunchPad)
LAUNCHXL-TMS57004 (LaunchPad)
LAUNCHXL2-570LC43 (LaunchPad)
LAUNCHXL2-RM46 (LaunchPad)
LAUNCHXL2-RM57L (LaunchPad)
2 Choose Your Connection
3 Start
Create Session From Existing Target Configuration File
Select a .ccxml file to create a new session.

Program the device

Choose a program to flash by clicking the Browse button, and you can either load the program or use it to verify against the content on the device.

Click on the "Binary" checkbox if you want to load the file as binary, a start address is needed for this.

Click on the "+" icon to add and load multiple flash images. Only the checked images will be flashed. The user need to be sure that the images does not overwrite each other in the target memory.

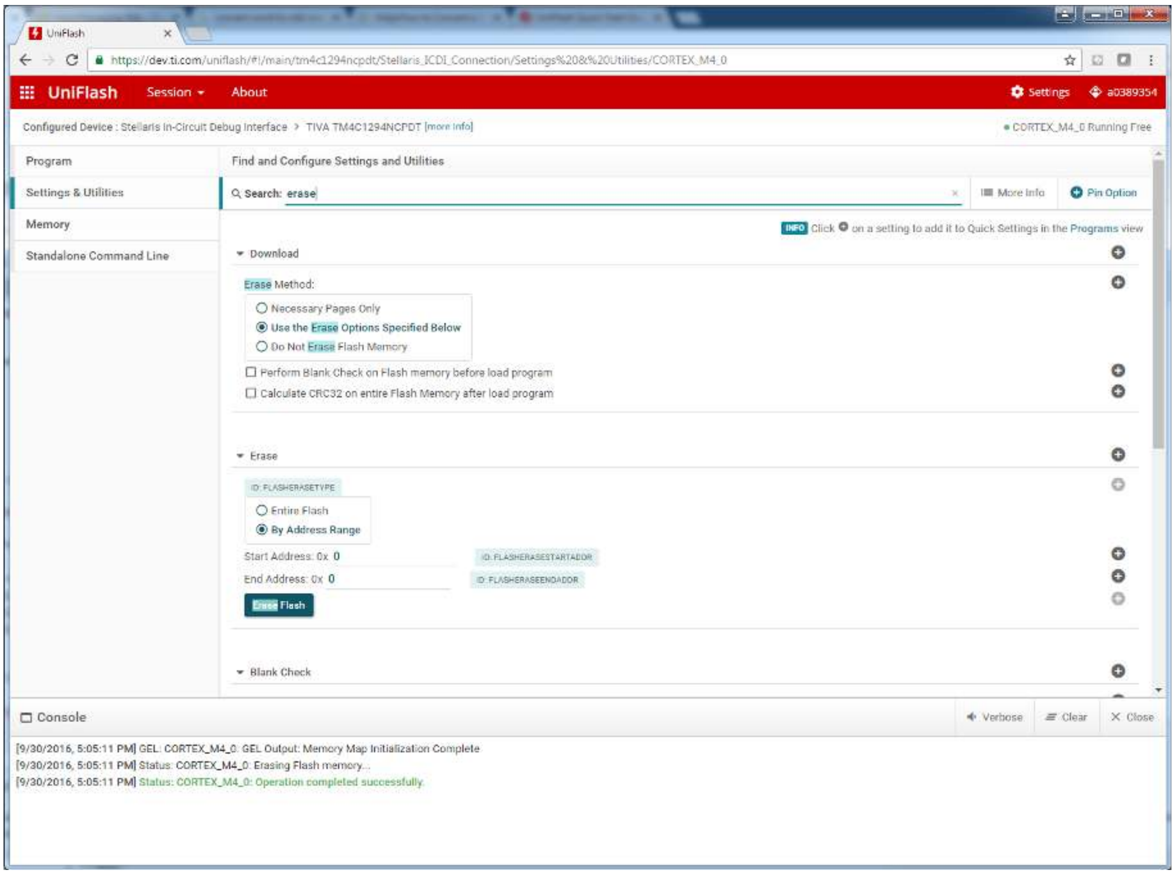

UniFlash
https://dev.ti.com/uniflash/#!/main/tm4c1294ncpdt/Stellaris_ICDI_Connection/Program/CORTEX_M4_0
UniFlash
Session
About
Settings
a0389354
Configured Device : Stellaris In-Circuit Debug Interface > TIVA TM4C1294NCPDT [more info]
CORTEX_M4_0
Program
Settings & Utilities
Memory
Standalone Command Line
Select and Load Images
Flash Image(s)
Sort by: Added
Image 1 Blink_123g.bin Size: 2.27 KB | Binary: | Load Addr: 0x00000000
Image 2 Blink_123g.bin Size: 2.27 KB | Binary: | Load Addr: 0x00000000
Available Action(s) - 1 Image Selected
Load Image
Verify Image
Quick Settings | Remove All
Entire Flash
By Address Range
Erase Flash
Console
Verbose
Clear
Close

Erase the device and other settings and utilities

In the *Settings and Utilities* view, you can find various settings to configure your operation. To quickly navigate this page, use the search filter to find the relevant items.

A commonly used *Erase Flash* utility is also found here under the *Erase* Section. You can *pin* this *Erase* button to the quick settings in the *Program* view discussed earlier in section 2.

To do that, click on the *Pin Options* button right at the end of search filter line. A column of "+" buttons will appear to allow you to customize on what to be added to the quick settings. Once you are done, you can use this in the *Program* view.



Configured Device : Stellaris In-Circuit Debug Interface > TIVA TM4C1294NCPDT [more info]

CORTEX_M4_0 Running Free

Find and Configure Settings and Utilities

Search: erase

INFO Click on a setting to add it to Quick Settings in the Programs view

Download

Erase Method:
- Necessary Pages Only
- [✓ Use the Erase Options Specified Below]
- Do Not Erase Flash Memory

☐ Perform Blank Check on Flash memory before load program
☐ Calculate CRC32 on entire Flash Memory after load program

Erase

ID: FLASHERASETYPE
- Entire Flash
- [✓ By Address Range]

Start Address: 0x 0 ID: FLASHERASESTARTADDR
End Address: 0x 0 ID: FLASHERASEENDADDR

Blank Check

```
[9/30/2016, 5:05:11 PM] GEL: CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Erasing Flash memory...
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Operation completed successfully.
```

Supported file formats

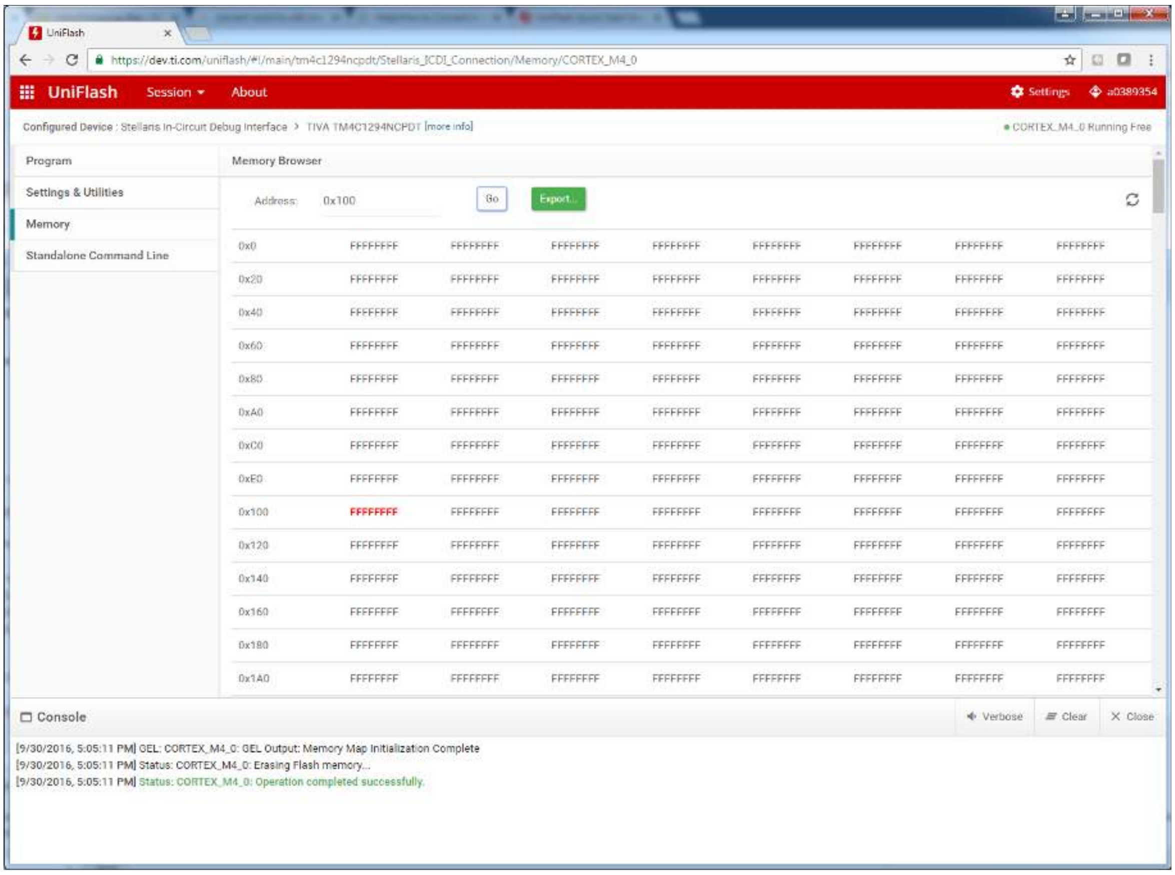
When loading a file to FLASH or RAM, the following formats are supported: TI COFF, TI ELF, Intel Hex, Motorola S-Record, Tektronix Hex, TI-TXT, and binary.

When saving memory to a file, TI COFF and binary formats are supported.

Memory browser and memory export

The *Memory View* allows you to quickly browse through the target memory. The view is read only.

Click on the "Export" button to export a range of memory into either a binary (.BIN) file or a COFF (.out) file. COFF format allows multiple memory sections to be exported.



Configured Device : Stellaris In-Circuit Debug Interface > TIVA TM4C1294NCPDT [more info]

CORTEX_M4_0 Running Free

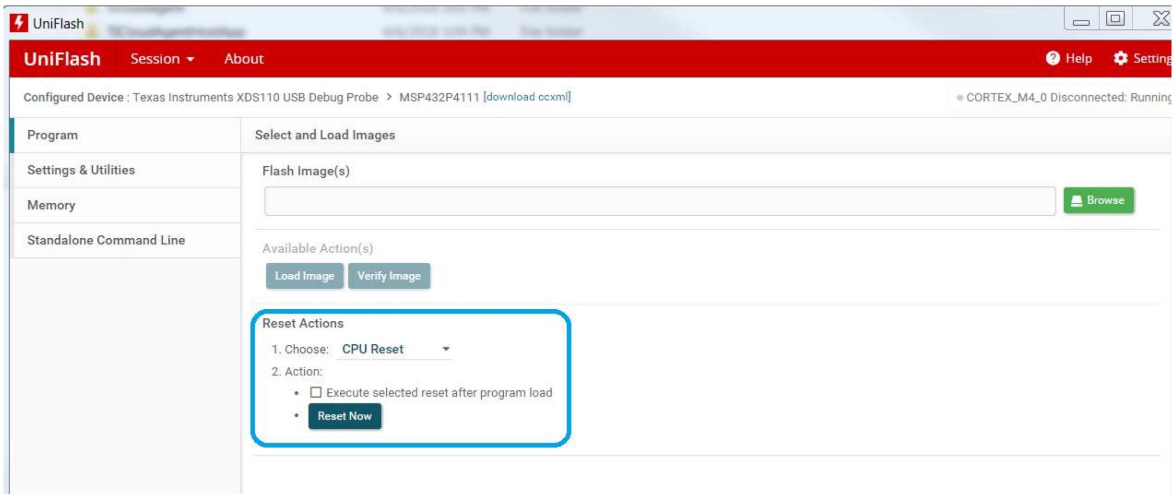
Address: 0x100

0x0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x20	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x40	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x60	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x80	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0xA0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0xC0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0xE0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x100	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x120	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x140	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x160	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x180	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
0x1A0	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF

```
[9/30/2016, 5:05:11 PM] GEL: CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Erasing Flash memory...
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Operation completed successfully.
```

Some devices may define short-cuts to specific memory ranges. If short-cut ranges are defined, they appear after the address field as below. For example, LPRF devices such as CC13xx and CC26xx may define short-cuts for CCFG and FCFG1.

Reset Actions



Configured Device : Texas Instruments XDS110 USB Debug Probe > MSP432P4111 [download ccxml]

CORTEX_M4_0 Disconnected: Running

Reset Actions

1. Choose: CPU Reset
2. Action:
   - ☐ Execute selected reset after program load
   - Reset Now

From the GUI, it is possible to specify reset actions to take after a program is loaded or to be immediately applied. The resets available will vary by connected device and are determined when the query link is clicked. After the list of resets is loaded, the selected reset can be applied immediately with the "Reset Now" button or after a program has been loaded.

CLI Device Detector

In offline install of UniFlash, there is a command line utility in the root installation folder called "detect-devices.bat" or "detect-devices.sh" to help you detect the list of plugged in devices and their COM port. Example output:

```
C:\ti\uniflash_4.2.2.1689>detect-devices.bat
0) Unknown TIMSP430-USB
1) CC2650 LaunchPad rev 1.0 TIXDS110_Connection L1000610 COM36
```

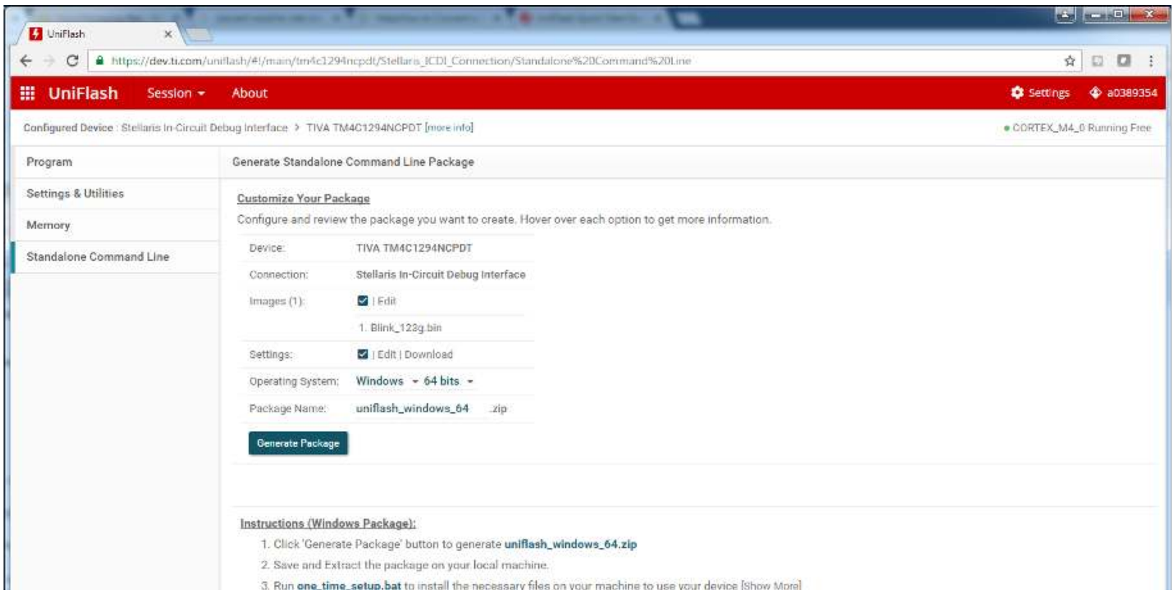
Standalone Command line tool

Often in production line environment all you really want is a CLI tool that does one job: Flash the device with verification. This new UniFlash v4 feature is tailor made for this use case.

After you have chosen the image you would like to flash and configured the flash settings accordingly, go to the "Standalone Command line" view to choose the deployment platform for the CLI tool. Hit "Generate Package" and a zip file will be created; containing everything you need for the deployment platform to flash your target.

You might need to run the driver installation script if the deployment machine does not have the drivers installed. Otherwise, just run "dslite" from the zip package to flash the target with the file you were working on in the GUI tool with the same flash settings.

While the cloud version of UniFlash can generate CLI package for all deployment platforms, the offline version of the tool can only generate package of the current platform.



Configured Device : Stellaris In-Circuit Debug Interface > TIVA TM4C1294NCPDT [more info]

● CORTEX_M4_0 Running Free

Generate Standalone Command Line Package

Customize Your Package

Configure and review the package you want to create. Hover over each option to get more information.

Device:	TIVA TM4C1294NCPDT
Connection:	Stellaris In-Circuit Debug Interface
Images (1):	✓ | Edit
	1. Blink_123g.bin
Settings:	✓ | Edit | Download
Operating System:	Windows 64 bits
Package Name:	uniflash_windows_64 .zip

Instructions (Windows Package):

1. Click 'Generate Package' button to generate uniflash_windows_64.zip
2. Save and Extract the package on your local machine.
3. Run one_time_setup.bat to install the necessary files on your machine to use your device [Show More]

4. Run **dslite.bat** to configure and connect to your device. [Show More]

```
[9/30/2016, 5:05:11 PM] GEL: CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Erasing Flash memory...
[9/30/2016, 5:05:11 PM] Status: CORTEX_M4_0: Operation completed successfully.
```

Updating USB Entry in Generated MSP430 Package

To switch between USB1, USB2,or USB3 connection in a generated MSP430 package, you can open up the connection XML file in your generated package (\ccs_base \common\targetdb\connections\TIMSP430-USB*.xml).

Look for the line

, and change the value field manually (101 for USB1, 102 for USB2, 103 for USB3).

Command Line Interface

UniFlash Desktop comes with a Command Line Interface (CLI) in the package. To access CLI, go to the install directory and look for the DSLite startup script (dslite.bat for Windows and dslite.sh for Linux/OSX).

DSLite comes with a few different modes, with the default mode being 'flash', use for flash programming on your device. Other modes might be added in the future to expand the functionality. To see the list of available modes, use the --listMode option.

Available Modes for UniFlash CLI:

```
* flash [default] - on-chip flash programming
* memory          - export memory to a file
* load            - simple loader [use default options]
* mspflasher      - support MSPFlasher command line parameters [deprecated]
```

Flash mode

The Flash mode (default) can be used for Flash Programming.

Usage:

```
dslite flash --config=ccxml-file [options] [flash-file1 ...]
```

Options:

```
 -c [ --config ] arg                 Target Configuration (ccxml) file, must 
be
                                     supplied other than running --help. CCXM
L
                                     file can be generated and exported using
                                     the Uniflash GUI or Code Composer Studi
o.
 -N [ --list-cores ]                 List the available cores and index of th
e
                                     current configuration (index to be used
                                     with -n afterwards).
 -n [ --core ] arg (=0)              Zero based index of core to operate on.
 -S [ --list-settings ] arg          Takes a regex and list the final values
                                     applied to the matching settings. (ie; u
se
                                     .* to list all available settings)
 -p [ --list-ops ]                   List the available flash operations.
 -I [ --list-device-cmds ]           List the available device commands, if
                                     any.
 -l [ --load-settings ] arg          Apply settings file generated by UniFlas
h
                                     GUI.
 -s [ --setting ] arg                Override a specific setting by using
                                     id=value.  Can be specified multiple
                                     times.
 -b [ --before ] arg                 Operation(s) to perform before loading.
                                     Can be specified multiple times, and are
                                     performed in the order they appear on th
e
                                     command line. Get a list of operations b
y
                                     using --list-ops.
 -f [ --flash ]                      Load the file(s) specified at the end of
                                     the command line to flash.  This is
                                     assumed to be set unless verify is set.
 -v [ --verify ]                     Verify the file(s) specified at the end 
of
                                     the command line.
 -a [ --after ] arg                  Optional operation to perform after
                                     loading.  Can be specified multiple time
s,
                                     and are performed in the order they appe
ar
                                     on the command line. Get a list of
                                     operations by using --list-ops.
 -R [ --list-resets ]                List of resets that can be done after
                                     program load and verification.
 -r [ --reset ] arg                  Set reset operation to be done after
                                     program load and verification.
 -O [ --post-flash-device-cmd ] arg Execute the given device command. Get a
                                     list of device commands by using
                                     --list-device-cmds.
 -t [ --timeout ] arg                Timeout in seconds, infinite if
                                     unspecified.
 -g [ --log ] arg                    Enable detailed logging to the specified
                                     file.
 -e [ --verbose ]                 Outputs progress messages to the consol
e.
 -h [ --help ]                    Print this message.
```

Tutorial and Examples

The easiest way to create a CLI package that by default flashes a particular file, given a particular set of setting, is to use the "Generate CLI Package" feature (see Standalone Command line tool (UniFlash_v4_Quick_Guide#Standalone_Command_line_tool)).

Once you unzip the package, you can find the target configuration file in the user_files/configs folder. The image you have selected in the program tab of the GUI is in the user_files/images folder, and all the device/flash settings you have applied in the GUI are saved in user_files/settings/generated.ufsettings.

By default, running dslite without any argument will automatically use the generated.ufsettings file to configure the device and apply the device/flash settings, and flash the target with the image file you have selected in the GUI.

What dslite does behind the scene is calling dslite.exe with the following command, for example:

```
 dslite.exe flash -c user_files/configs/cc2650f128.ccxml -l user_files/settings/generated.ufsettings -e -f -v user_files/images/cc2650_modem_onepass.out
```

Where

```
-   -c is specifying the config file,
-   -l is specifying the user settings file,
-   -e is for verbose mode,
-   -f is specifying that we want to flash,
-   -v is specifying that we want to verify after flashing.
```

At the end of the options, we specify the image files to be flashed.

When running dslite.bat with arguments, it basically calls dslite.exe and forward your argument to the executable instead of using the generated ones shown above. Since all actions need -c to be specified (except when bringing up help with -h), if you are only working on one device, you can skip that part by writing you own batch/shell script to launch dslite.bat with a fixed .ccxml file.

Sometimes certain devices have device-specific commands. To view them, use:

```
dslite.bat -c <your ccxml file> --list-device-cmds
```

This will show a list of available commands, which may look different from device to device. For example, with CC13xx and CC26xx, it may look like the following:

```
------ Device Commands -------------------------
  PinReset: CC13xx and CC26xx specific reset.
------------------------------------------------
```

Dslite may return an error if no device-specific commands exist:

```
Failed: 'GEL_CLI_ListCommands()' not found
```

To call a device-specific command, use:

` dslite.bat -c --post-flash-device-cmd

In the C13xx and CC26XX case,

```
dslite.bat -c <your ccxml file> --post-flash-device-cmd PinReset
```

If you want to reset the device after program load, take a look at what kind of resets are available by using the --list-resets or -R option. For example, on C28377D it using the following command:

```
dslite-C28xx_CPU1.bat -c user_files\configs\tms320f28377d.ccxml -R
```

Would give the following output:

```
--- Available Reset Operations ---
        0. CPU Reset
```

To issue the reset, do

```
dslite-C28xx_CPU1.bat -c user_files\configs\tms320f28377d.ccxml -r 0
```

To list available cores, do

```
dslite.bat -c F2838xD.ccxml -N
```

Example Output:

```
--- Available Cores ---
0: Texas Instruments XDS2xx USB Debug Probe_0/C28xx_CPU1
1: Texas Instruments XDS2xx USB Debug Probe_0/CPU1_CLA1
2: Texas Instruments XDS2xx USB Debug Probe_0/C28xx_CPU2
3: Texas Instruments XDS2xx USB Debug Probe_0/CPU2_CLA1
4: Texas Instruments XDS2xx USB Debug Probe_0/Cortex_M4_0
```

Memory mode

The Memory mode allows you to export device memory to a file

Usage:

```
dslite --mode memory [options] --config=ccxml-file
--range=address,length --output=output-file
```

Options:

```
 -c [ --config ] arg    Configuration (ccxml) file.
 -f [ --coff ]          Output in coff format.  Default format is binary.
 -r [ --range ] arg     Memory location and length to read in the form
                        'address,length' or 'address@page,length'.  This can
                        be specified multiple times if using the coff forma
t.
 -s [ --size ] arg (=8) Bit size of each memory value to read (for endiannes
s
                        considerations, binary format only).
 -o [ --output ] arg    File to write the extracted memory to.
 -t [ --timeout ] arg   Timeout in seconds, infinite if unspecified.
 -g [ --log ] arg       Enable detailed logging to the specified file.
 -e [ --verbose ]       Outputs progress messages to the console.
 -r [ --core ] arg (=0) Zero based index of core to operate on.
 -h [ --help ]          Print this message.
```

Load mode

The Load mode can be used for a simple load operation (with default options)

Usage:

```
dslite --mode load --config=file.ccxml [options] coff-or-binary-file1
```

Options:

```
 -c [ --config ] arg    Configuration (ccxml) file.
 -f [ --file ] arg      Elf, coff or binary file(s) to flash or verify.  To
                        specify a binary file, append ',address' after the
                        file.  This option exists only for compatibility - t
he
                        file to load can instead just be passed as the last
                        token on the command line.
 -t [ --timeout ] arg   Timeout in seconds, infinite if unspecified.
 -g [ --log ] arg       Enable detailed logging to the specified file.
 -r [ --core ] arg (=0) Zero based index of core to operate on.
 -h [ --help ]          Print this message.
```

CC13xx/CC26xx Mass Erase from CLI

The CC13xx and CC26XX have some unique requirements for mass erase. To support these requirements, a separate CLI mode is used for mass erase. To execute a mass erase on these familes, use the following command:

```
dslite --mode cc13xx-cc26xx-mass-erase -d XDS110
```

Supported debug probes include XDS110, XDS100v3, and XDS200.

Error code

UniFlash CLI sets the error level variable when exiting from the script, where a value of 0 means that the previous operation passed; any other value means that the previous value failed. To query for pass or success of the operation, run the following:

Windows:

```
> echo %errorlevel%
```

Linux/OSX:

```
> echo $?
```

MSPFlasher mode

The MSPFlasher mode supports all of the same options and features of the existing mspflasher standalone flashing tool. This mode is deprecated as the Flash mode can be used to program MSP430 devices as well.

Scripting

The command line interface can be called from the scripting language of your choice.

For complex scripting tasks, Debug Server Scripting (DSS) is supported in UniFlash v4. To use DSS, go to your `

\deskdb\content\TICloudAgent\{os}\ccs_base\scripting\bin\` folder in your install directory, and start dss using the dss.bat/dss.sh script.

Note: A Java Runtime Environment (JRE) is needed to run DSS (32-bit JRE on Windows, 64-bit JRE on Linux and Mac). This was included as part of UniFlash v3, but is not included in the v4 installer. More information is available at this link: UniFlashv4_DSS (UniFlashv4_DSS)

Defect Reporting and Logging

If you think you have encountered a bug, please report it in the e2eforum here: https://e2e.ti.com/support/development_tools/code_composer_studio/ (https://e2e.ti.com/support/development_tools/code_composer_studio/)

Please send in the logs with your post as well as they will make things much clearer.

To turn on the logs:

1. go to the welcome/new session page(where you setup the device and connection type) and click on refresh to reload the uniflash app

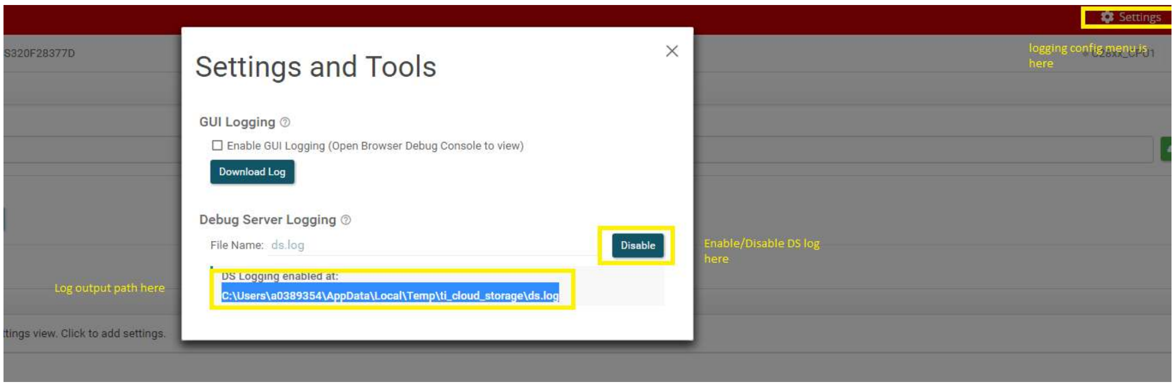
2. launch your session

2. go to "Settings" on the top right nav bar and click on "Enable" under the "Debug Server Logging". This may take a while and display a path to the DS log. Now close the popup.

3. operate the GUI to reproduce the bug

4. go back to "Settings", click on "Download Log" under "GUI Logging" to obtain the GUI debug log, then go obtain the log file shown as the path under "Debug Server

Logging", send both files by attach to your post.


Settings
logging config menu is here
Settings and Tools
GUI Logging
Enable GUI Logging (Open Browser Debug Console to view)
Download Log
Debug Server Logging
File Name: ds.log
Disable
Enable/Disable DS log here
DS Logging enabled at:
C:\Users\a0389354\AppData\Local\Temp\ti_cloud_storage\ds.log
Log output path here

Notes

1. For multi-core devices, UniFlash only supports programming one core at a time (both GUI and CLI).