

**Please note that Cypress is an Infineon Technologies Company.**

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



# Cypress Auto Flash Utility

## User Guide

Document Number: 002-27684 Rev. \*B

Cypress Semiconductor Corp.  
An Infineon Technologies Company  
198 Champion Court  
San Jose, CA 95134-1709  
[www.infineon.com](http://www.infineon.com)  
[www.cypress.com](http://www.cypress.com)

## Copyrights

© Cypress Semiconductor Corporation, 2019-2020. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.

# Contents



<b>User Guide .....</b>	<b>1</b>
<b>Contents .....</b>	<b>3</b>
<b>1 Introduction .....</b>	<b>6</b>
Overview .....	6
Acronyms and Abbreviations .....	6
Supported OS .....	7
Supported MCU Devices .....	7
Supported Hardware (Probes).....	7
Document Convention .....	7
Revision History .....	8
<b>2 AutoFlashUtil Installation .....</b>	<b>9</b>
Windows Package Contents.....	9
System Prerequisites.....	9
Installation Procedure on Windows 10 (x86 / x64) .....	9
J-Link configuration .....	11
<b>3 Getting Started .....</b>	<b>12</b>
Connect Device .....	12
List Connected Targets .....	12
Program Traveo II Target Device with J-Link Probe .....	12
Program Traveo II Target Device with MiniProg4 Probe .....	13
Program Device Using Configuration File Only .....	14
Program Device Using Configuration File and Command Line .....	15
<b>4 Supported Target Configurations .....</b>	<b>16</b>
<b>5 Command-Line Options .....</b>	<b>17</b>
--file (-f) .....	17
--search (-s) .....	18
--command (-c) .....	19
--debug (-d) .....	19
--log_output (-l) .....	19

--help (-h) .....	19
--version (-v) .....	19
<b>6 OpenOCD Commands Overview .....</b>	<b>20</b>
<b>7 OpenOCD Commands Description .....</b>	<b>22</b>
General OpenOCD Commands .....	22
version .....	22
help .....	22
shutdown .....	22
log_output .....	22
debug_level .....	23
reset_config .....	23
adapter_khz .....	23
transport list .....	24
transport select .....	24
targets .....	24
scan_chain .....	25
md(w)(h)(b) .....	25
mw(w)(h)(b) .....	26
init .....	26
reset [run] [halt] [init] .....	27
program .....	27
flash banks .....	28
flash list .....	29
flash info .....	29
flash erase_sector .....	30
flash erase_address .....	31
flash write_bank .....	31
flash write_image .....	32
flash fill(w)(h)(b) .....	33
flash read_bank .....	34
flash verify_bank .....	34
flash padded_value .....	35
flash rmw .....	35
add_verify_range .....	36
show_verify_ranges .....	36
clear_verify_ranges .....	37
verify_image .....	37
verify_image_checksum .....	38
load_image .....	39
dump_image .....	39
MiniProg4 Driver Commands .....	40
kitprog3 acquire_config .....	40
kitprog3 acquire_psoc .....	40
kitprog3 power_config .....	41
kitprog3 power_control .....	41
kitprog3 led_control .....	42
kitprog3 get_power .....	42
Flash Driver Commands .....	43
traveo2 sflash_restrictions .....	43
traveo2 add_safe_sflash_region .....	43
traveo2 allow_efuse_program .....	44
traveo2 reset_halt .....	45

traveo2 ecc_error_reporting.....	45
enable_hyperram .....	46
enable_hyperram_x .....	46
Other Commands .....	46
source.....	46
find.....	47
set.....	47
sleep.....	47
add_script_search_dir.....	47
<b>8 Global Variables .....</b>	<b>48</b>
ENABLE_ACQUIRE .....	48
ENABLE_POWER_SUPPLY .....	48
ENABLE_HYPERFLASH, ENABLE_DUALQUADSPI .....	48
ENABLE_HYPERFLASH_x, ENABLE_DUALQUADSPI_x .....	48
ENABLE_SEMPERFLASH_0.....	49
HYPERFLASH_LOADER, HYPERRAM_LOADER, DUALQUADSPI_LOADER .....	49
SEMPERFLASH_LOADER_0, HYPERFLASH_LOADER_0, HYPERRAM_LOADER_0, DUALQUADSPI_LOADER_0 .....	49
MAIN_LARGE_SECTOR_NUM, MAIN_SMALL_SECTOR_NUM, WORK_LARGE_SECTOR_NUM, WORK_SMALL_SECTOR_NUM .....	49
<b>9 Error Codes .....</b>	<b>50</b>
<b>10 Usage Examples .....</b>	<b>51</b>
Erase Main Flash Rows 0...10 of Traveo II 1M device .....	52
Display Memory Contents (32 words at address 0x08000000) of Traveo II 1M device .....	53
Program Traveo II 1M Device with Verification (Intel HEX file) .....	54
Program EFuse region of Traveo II Device .....	55
Modify Individual Bytes of Traveo II 1M in Main Flash and Display Results .....	56
Start GDB Server and Leave It Running .....	57
<b>11 Documentation and Links.....</b>	<b>58</b>

# 1 Introduction



## Overview

Cypress Auto Flash Utility (AutoFlashUtil) is a flexible, cross-platform, integrated application to allow programming Cypress devices. It can perform Program, Erase, Verify, and Read operations on the flash of the target device. It can target an entire device, a specific region, a sector, and even a byte of a device.

The AutoFlashUtil is based on the Open On-Chip Debugger (OpenOCD) open source. OpenOCD is a powerful tool whose interface interacts with the target device via the JTAG/SWD debug ports. OpenOCD allows programming internal and external flash memories of a wide range of target devices, CFI-compatible flashes, and some CPLD/FPGA devices.

OpenOCD was originally developed by Dominic Rath at the University of Applied Sciences in Augsburg. The OpenOCD source code is now available through the GNU General Public License (GPL).

This document covers the Cypress-specific CLI extensions of OpenOCD. For more details about OpenOCD, refer to the official documentation available at <http://openocd.org/documentation/>.

## Acronyms and Abbreviations

- AutoFlashUtil – Cypress Auto Flash Utility
- OpenOCD - Open On-Chip Debugger. An open-source tool that allows programming internal and external flash memories of a wide range of target devices.
- CLI – Command-line interface.
- Tcl - Tool Command Language. A high-level, general-purpose, interpreted, dynamic programming language.
- SWD – Serial Wire Debug interface.
- JTAG – Joint Test Action Group. Specifies the use of a dedicated debug port implementing a serial communication interface for low-overhead access without requiring direct external access to the system address and data buses.
- TAP – JTAG Test Access Port.
- PSoC –Programmable System-on-Chip. A family of microcontroller integrated circuits by Cypress. These chips include a CPU core and mixed-signal arrays of configurable integrated analog and digital peripherals.
- MCU – Microcontroller Unit.
- UDD – Universal Device Database.
- FLD – Flash Loader Database.
- AP – Access Port register of ARM Cortex CPU. Used for programming and debugging, along with the corresponding SWD address bit selections.

- DP – Debug Port register of ARM Cortex CPU. Used for programming and debugging, along with the corresponding SWD address bit selections.
- Region – A logical area within the target device the programmer operates on.
- KP3 – KitProg3 device.
- MP4 – MiniProg4 device.

## Supported OS

- Windows 10 (x86 / x64)

## Supported MCU Devices

- Traveo II with up to 1M of flash size: CYT2B6, CYT2B7
- Traveo II with up to 2M of flash size: CYT2B9
- Traveo II with up to 6M of flash size: CYT4DN
- Traveo II with up to 8M of flash size: CYT4BF
- Traveo II with up to 4M of flash size: CYT3BB, CYT4BB

## Supported Hardware (Probes)

- SEGGER J-Link
- Cypress MiniProg4

Note: in order to update KitProg3 FW on MiniProg4 probe, use FW-loader tool available on Cypress GitHub [repository](#). The Firmware Loader does not install any drivers, but you can use it to upgrade (or downgrade) the KitProg firmware on a MiniProg4 probe.

## Document Convention

This guide uses the following conventions:

Convention	Usage
Courier New	Displays file locations and source code: C:\ ...cd\icc\, user entered text
<i>Italics</i>	Displays file names and reference documentation: <i>sourcefile.hex</i>
[bracketed, bold]	Displays keyboard commands in procedures: [Enter] or [Ctrl] [C]
<b>File &gt; New Project</b>	Represents menu paths: <b>File &gt; New Project &gt; Clone</b>
<b>Bold</b>	Displays commands, menu paths and selections, and icon names in procedures: Click the <b>Debugger</b> icon, and then click <b>Next</b> .
Text in gray boxes	Displays cautions or functionality unique to the device software.



## Revision History

Document Title: Cypress Auto Flash Utility User Guide Document Number: 002-27684		
Revision	Date	Description of Change
**	07/01/19	New document.
*A	10/04/19	Updated section 4 "Supported Target Configurations" Added traveo2 add_safe_sflash_region command information Added global variable ENABLE_SEMPERFLASH_0 Added "Output like the following should be displayed" before screenshots and highlighted in green important message for all screenshots Deleted MPN mention from "Acronyms and Abbreviations" Deleted "Installation" from "Introduction" section Changed the name of config file "traveo2_1m_0A.cfg" to "traveo2_1m_a0.cfg" in whole document Updated table of content User Guide clean-up in whole document
*B	05/27/20	Updated section 4 "Supported Target Configurations" Updated section "Supported MCU Devices" Added global variable HYPERFLASH_LOADER, HYPERRAM_LOADER, DUALQUADSPI_LOADER Added global variable SEMPERFLASH_LOADER_0, HYPERFLASH_LOADER_0, HYPERRAM_LOADER_0, DUALQUADSPI_LOADER_0 Clean-up over whole document

## 2 AutoFlashUtil Installation



### Windows Package Contents

The AutoFlashUtil package for Windows contains:

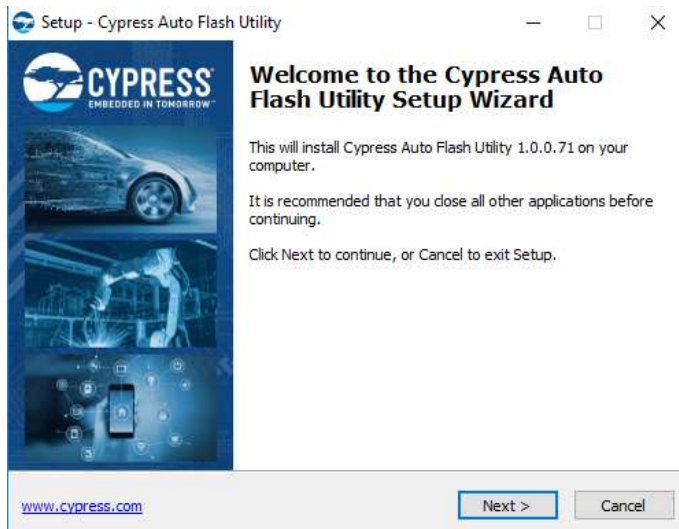
- *openocd.exe* – The CLI application to program Traveo II devices based on the Open On-Chip Debugger
- The drivers for the Cypress MiniProg4 hardware programmer

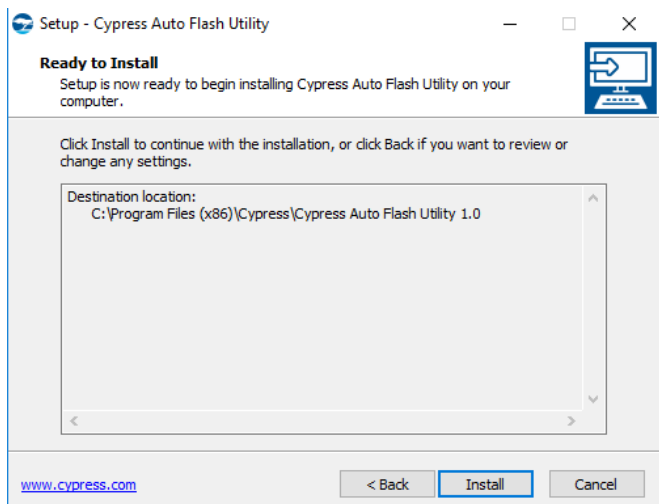
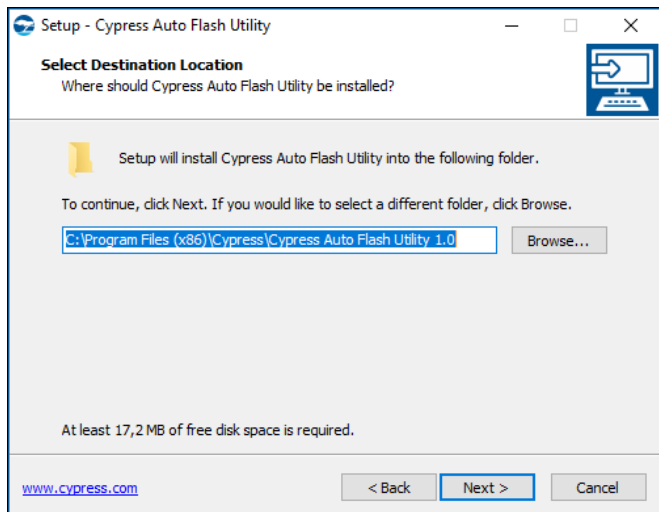
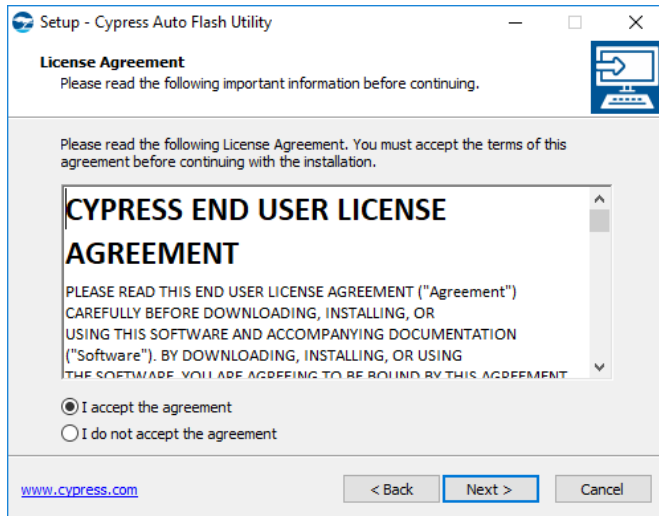
### System Prerequisites

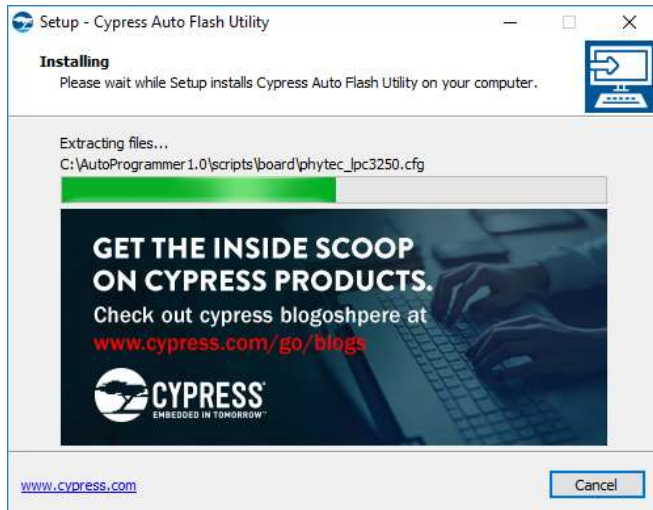
- Windows 10 x86/x64

### Installation Procedure on Windows 10 (x86 / x64)

1. Run the Windows installer for AutoFlashUtil.
2. Follow the instructions of the installation wizard:



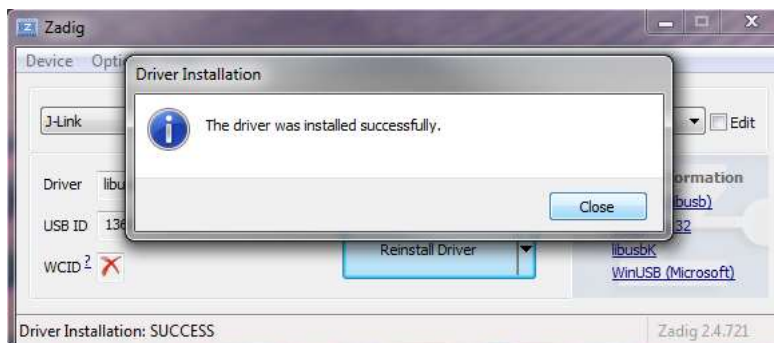
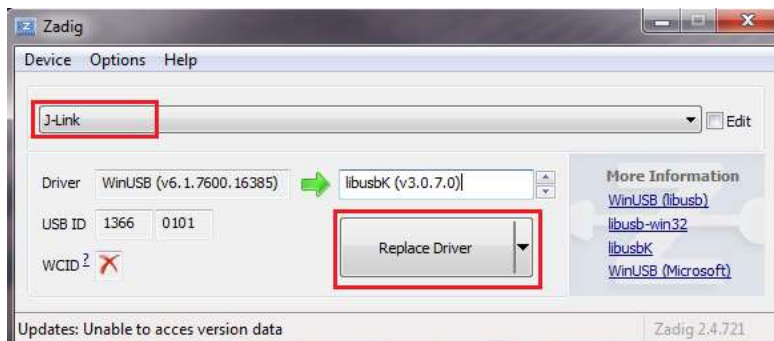




## J-Link configuration

In order to use J-Link with AutoFlashUtil, you need to replace the J-Link driver with the libusbK driver:

3. Download Zadig tool from <https://zadig.akeo.ie/>
4. Run zadig-2.4 executable.
5. Select “Options > List all devices”.
6. Select J-Link in the drop-down menu.
7. Select “libusbK (v3.0.7.0)” driver.
8. Click Replace Driver.



## 3 Getting Started



### Connect Device

Connect the host computer to a Probe device (e.g. J-Link) with the attached Traveo II target device, used in the examples below.

### List Connected Targets

The example below displays target names available in the Traveo II device connected to the J-Link hardware programmer. The programmer communicates with the Traveo II device over the SWD hardware interface.

Open the command-line window. Invoke “cd C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.1\bin” to change the directory to the AutoFlashUtil installation folder.

Execute the command:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "targets; shutdown"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "targets; shutdown"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html

swd
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100

  TargetName      Type      Endian  TapName      State
-----
0* traveo2.cpu.cm0 cortex_m    little  traveo2.cpu   unknown
1 traveo2.cpu.cm4  cortex_m    little  traveo2.cpu   unknown
shutdown command invoked
```

The command output displays the list of target names (JTAG TAPs) attached to the programming device.

### Program Traveo II Target Device with J-Link Probe

The next example initializes the J-Link probe with the Traveo II 1M target device, programs flash with *firmware.hex* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

Execute the command:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "program d:/firmware.hex verify exit"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_0A.cfg -c "program d:/firmware.hex verify exit"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html

swd
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.877 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
** Programming Started **
auto erase enabled
Info : Padding image section 0 at 0x100017a4 with 92 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x10001800 .. 0x10007fff
[100%] [#####] [ Erasing ]
[100%] [#####] [ Programming ]
wrote 6144 bytes from file d:/firmware.hex in 0.326196s (18.394 KiB/s)
** Programming Finished **
** Verify Started **
verified 6052 bytes in 0.142085s (41.596 KiB/s)
** Verified OK **
shutdown command invoked
```

**Note** See [Supported Target Configurations](#) for the list of available target configurations.

## Program Traveo II Target Device with MiniProg4 Probe

The next example initializes the MiniProg4 probe with the Traveo II 1M target device, programs flash with *firmware.elf* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

Execute the command:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/traveo2_1m_a0.cfg -c
"program d:/firmware.elf 0x0 verify exit"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/kitprog3.cfg -f target/traveo2_1m_a0.cfg -c "program d:/firmware.elf 0x0 verify exit"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1500 kHz
adapter speed: 1000 kHz
** Auto-acquire enabled, use "set ENABLE_ACQUIRE 0" to disable
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : CMSIS-DAP: SWD Supported
Info : CMSIS-DAP: FW Version = 2.0.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/ICK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : VTarget = 5.032 V
Info : kitprog3: acquiring PSoC device...
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
Info : traveo2.cpu.cm0: external reset detected
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : traveo2.cpu.cm4: external reset detected
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
Info : kitprog3: acquiring PSoC device...
target halted due to debug-request, current mode: Thread
xPSR: 0xa1000000 pc: 0x000002b8 msp: 0x0801f7f0
** Device acquired successfully
** traveo2.cpu.cm4: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x000001d0 msp: 0x0801f800
** Programming Started **
auto erase enabled
Info : Padding image section 1 at 0x10091adc with 292 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x10091c00 .. 0x10097fff
[100%] [#####] [ Erasing ]
[100%] [#####] [ Programming ]
wrote 72704 bytes from file d:/firmware.elf in 1.206723s (58.837 KiB/s)
** Programming Finished **
** Verify Started **
verified 72412 bytes in 1.179707s (59.943 KiB/s)
** Verified OK **
shutdown command invoked
```

## Program Device Using Configuration File Only

The whole configuration is stored in a single *sample.cfg* configuration file. For example, the following configuration file describes the Traveo II device connected using the J-Link debug probe. This file initializes the target device, programs flash with the *firmware.hex* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

```
source [find interface/jlink.cfg]
transport select swd
source [find target/traveo2_1m_a0.cfg]
program d:/firmware.hex verify exit
```

Execute the command:

```
openocd -s ../scripts -f path/to/sample.cfg
```

## Program Device Using Configuration File and Command Line

A significant part of the configuration file specifies the debug adapter, transport type, target chip, SWD frequency, reset type, etc. This part of the file reflects the hardware configuration and thus stays unchanged between sessions. In some cases, a combined method of passing the Tcl commands is more convenient:

The example *sample.cfg* file contents:

```
source [find interface/jlink.cfg]
transport select swd
source [find target/traveo2_1m_a0.cfg]
```

Execute the command:

```
openocd -s ../scripts -f path/to/sample.cfg -c "program d:/firmware.hex verify
exit"
```



## 4 Supported Target Configurations



To connect AutoFlashUtil to a target device, pass one of the following configuration files as the argument for the `--file` command-line option; for example, `-f target/traveo2_1m_a0.cfg`. The following configuration files are located in the `target/` directory of the OpenOCD tree.

Target configurations for silicon and pre-silicon support:

#	Target Config	Supported families	Revision
1	traveo2_00.cfg	CYT2B6*, CYT2B7 target configuration.	A
2	traveo2_1m_a0.cfg	CYT2B6*, CYT2B7- target configuration.	B, C, D
3	traveo2_2m.cfg	CYT2B9 target configuration.	A, B, C
4	traveo2_6m.cfg	CYT4DN target configuration.	A
5	traveo2_8m.cfg	CYT4BF,— target configuration.	A
6	traveo2_8m_b0.cfg	CYT4BF,— target configuration.	B, C, D
7	traveo2_6m_psvp.cfg	CYT4DN target configuration (pre-silicon support)	A
8	traveo2_8m_psvp.cfg	CYT4BF,— target configuration (pre-silicon support)	A
9	traveo2_8m_psvp_b.cfg	CYT4BF,— target configuration (pre-silicon support)	B
10	traveo2_4m.cfg	CYT3BB*, CYT4BB target configuration.	A
11	traveo2_c2d_4m.cfg	CYT3DL target configuration (pre-silicon support)	A

Note: for all series with “\*” above need to modify manually the Target Config files by following steps from MAIN\_LARGE\_SECTOR\_NUM, MAIN\_SMALL\_SECTOR\_NUM, WORK\_LARGE\_SECTOR\_NUM, WORK\_SMALL\_SECTOR\_NUM

## 5 Command-Line Options



OpenOCD is a command-line tool but it has only several command-line options. Several options can be combined in a single command-line.

The most important options and commands:

Option	Description
<a href="#">--file (-f)</a>	Specifies the configuration file to use.
<a href="#">--search (-s)</a>	Specifies the directory to search for configuration files.
<a href="#">--command (-c)</a>	Executes an OpenOCD command. See Section <a href="#">OpenOCD Commands Overview</a> for details.
<a href="#">--debug (-d)</a>	Specifies the debug level.
<a href="#">--log_output (-l)</a>	Redirects the log output to the file.
<a href="#">--help (-h)</a>	Displays the help message.
<a href="#">--version (-v)</a>	Displays the OpenOCD version.

### --file (-f)

Specifies the configuration file to use. Multiple configuration files can be specified from a command line. They are interpreted in the order they are specified in the command line.

```
openocd -f <filename.cfg>
openocd -f interface/ADAPTER.cfg -f target/TARGET.cfg
```

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag" -f
target/traveo2_lm_a0.cfg
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag" -f target/traveo2_1m_0a.cfg
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html

jtag
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.875 V
Info : clock speed 1000 kHz
Info : JTAG tap: traveo2.cpu tap/device found: 0x6ba00477 (mfg: 0x23b <ARM Ltd.>, part: 0xba00, ver: 0x6)
Info : JTAG tap: traveo2.bs tap/device found: 0x2e300069 (mfg: 0x034 <Cypress>, part: 0xe300, ver: 0x2)
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
```

The "tap/service found" message should display with no warnings. That means the JTAG communication is working.

## --search (-s)

Specifies the directory to search for configuration files. Multiple -s options can be specified. Configuration files and scripts are searched for in the following paths:

- the current directory
- any search directory specified on the command line using the -s option
- any search directory specified using the **add\_script\_search\_dir** command
- \$HOME/.openocd (not on Windows)
- a directory in the OPENOCD\_SCRIPTS environment variable (if set)
- the site wide-script library \$pkgdatadir/site
- the OpenOCD-supplied script library \$pkgdatadir/scripts.

The first found file with a matching file name is used.

```
openocd -s <directory>
```

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_1m_a0.cfg
```

In this example, the -s option specifies the relative path to the directory where the interface and target configurations are located.

## --command (-c)

Executes the Tcl command(s). Multiple commands can be executed by either specifying the multiple -c options or passing several commands to the single -c options. In the latter case, separate the commands with a semicolon.

```
openocd -c <command>
openocd -c <"command1; command2; ...">
```

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_lm_a0.cfg -c
"targets; shutdown"
```

## --debug (-d)

Specifies the debug level. This affects the kind of messages sent to the server log. Level 0 is error messages only; Level 1 adds warnings; Level 2 adds informational messages; and Level 3 adds debugging messages. The debug level is 2 by default.

```
openocd -d<n>
```

Example:

```
openocd -d1
```

## --log\_output (-l)

Redirects the log output to the file <logfile.txt>.

```
openocd -l <logfile.txt>
```

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_lm_a0.cfg -l
d:/log.txt -c "targets; shutdown"
```

## --help (-h)

Displays the help message.

```
openocd -h
```

## --version (-v)

Displays the OpenOCD version.

```
openocd -v
```

## 6 OpenOCD Commands Overview



The available OpenOCD Tcl commands are listed in the following table. You can combine several commands in a single command-line or pass them via the configuration file.

The command can be invoked with “-c *command*” command line option.

Command	Description
<a href="#">version</a>	Displays a string identifying the version of the OpenOCD.
<a href="#">help</a>	With no parameters, prints the help text for all commands.
<a href="#">shutdown</a>	Closes the OpenOCD server, disconnecting all clients.
<a href="#">log_output</a>	Redirects logging to the filename; the initial log output channel is <i>stderr</i> .
<a href="#">debug_level</a>	Displays the debug level.
<a href="#">reset_config</a>	Displays or modifies the reset configuration of your combination of the board and target.
<a href="#">adapter_khz</a>	Sets the non-zero speed in KHZ for the debug adapter.
<a href="#">transport list</a>	Displays the names of the transports supported by this version of OpenOCD.
<a href="#">transport select</a>	Selects which of the supported transports to use in this OpenOCD session.
<a href="#">targets</a>	Displays a table of all known targets or sets the current target to a given target with a given name.
<a href="#">scan_chain</a>	Displays the TAPs in the scan chain configuration, and their status.
<a href="#">md(w)(h)(b)</a>	Displays the contents of the address as 32-bit words (mdw), 16-bit half-words (mdh), or 8-bit bytes (mdb).
<a href="#">mw(w)(h)(b)</a>	Writes the specified word (32 bits), half-word (16 bits), or byte (8-bit) value, at the specified address.
<a href="#">init</a>	Terminates the configuration stage and enters the run stage.
<a href="#">reset [run] [halt] [init]</a>	Performs as hard a reset as possible, using SRST if possible.
<a href="#">program</a>	Programs a given programming file in the HEX, SREC, BIN or ELF formats into flash.
<a href="#">flash banks</a>	Prints a one-line summary of each flash bank of the target device.
<a href="#">flash list</a>	Retrieves a list of associative arrays for each device that was declared using a flash bank numbered from zero.
<a href="#">flash info</a>	Prints info about the flash bank, a list of protection blocks and their status.
<a href="#">flash erase_sector</a>	Erases sectors in a given bank.
<a href="#">flash erase_address</a>	Erases sectors starting at a given address.
<a href="#">flash write_bank</a>	Writes the binary file to a given flash bank.
<a href="#">flash write_image</a>	Writes the image file to the current target's flash bank(s).
<a href="#">flash fill(w)(h)(b)</a>	Fills flash memory with the specified word (32 bits), half-word (16 bits), or byte.

Command	Description
<a href="#">flash read_bank</a>	Reads bytes from the flash bank and writes the contents to the binary file.
<a href="#">flash verify_bank</a>	Compares the contents of the binary file with the contents of the flash.
<a href="#">flash padded_value</a>	Sets the default value used for padding-any-image sections.
<a href="#">flash rmw</a>	Can be used to modify flash individual bytes.
<a href="#">add_verify_range</a>	Allows specifying memory regions to be compared during <i>verify</i> operation.
<a href="#">show_verify_ranges</a>	Displays all active verify ranges for all targets that were added using the <i>add_verify_range</i> command. This command does not take any arguments.
<a href="#">clear_verify_ranges</a>	Deletes all verify ranges for the specified target that were added using the <i>add_verify_range</i> command.
<a href="#">verify_image</a>	Verifies a file against the target memory starting at a given address.
<a href="#">verify_image_checksum</a>	Verifies a file against the target memory starting at a given address.
<a href="#">load_image</a>	Loads an image from a file to the target memory offset from its load address.
<a href="#">dump_image</a>	Dumps bytes of the target memory to the binary file.
<a href="#">kitprog3 acquire_config</a>	Controls device acquisition parameters and optionally enables acquisition during the early initialization phase.
<a href="#">kitprog3 acquire_psoc</a>	Performs device acquisition.
<a href="#">kitprog3 power_config</a>	Controls the MP4 internal power supply parameters and optionally enables power.
<a href="#">kitprog3 power_control</a>	Turns on or off the MP4 internal power supply.
<a href="#">kitprog3 led_control</a>	Controls the MP4 LEDs.
<a href="#">kitprog3 get_power</a>	Reports the target voltage in millivolts.
<a href="#">traveo2 sflash_restrictions</a>	Enables or disables writes to SFlash regions other than USER, NAR, TOC2, and KEY.
<a href="#">traveo2 allow_efuse_program</a>	Allows or disallows writes to the EFuse region.
<a href="#">traveo2 reset_halt</a>	Simulates a broken Vector Catch on Traveo II devices.
<a href="#">traveo2 ecc_error_reporting</a>	Enables or Disables the ECC error reporting during flash <i>read</i> operations.
<a href="#">source</a>	Reads a file and executes it as a Tcl script.
<a href="#">find</a>	Finds and returns the full path to a file with the Tcl script.
<a href="#">set</a>	Creates a Tcl variable.
<a href="#">add_script_search_dir</a>	Adds a directory to the file/script search path.
<a href="#">sleep</a>	Waits for a given number of milliseconds before resuming.

# 7 OpenOCD Commands Description



This section includes all relevant OpenOCD commands along with their descriptions and usage examples.

All examples described in this section can be executed against different target devices (e.g. Traveo II 1M and Traveo II 8M). See [Supported Target Configurations](#) for the detailed list of available target devices and corresponding OpenOCD configuration files.

## General OpenOCD Commands

### version

Displays a string identifying the version of the OpenOCD.

Example:

```
openocd -c "version; shutdown"
```

### help

With no parameters, prints help text for all commands. Otherwise, prints each help-text-containing string. Not each command provides help text.

```
help [string]
```

Example:

```
openocd -c "help; shutdown"
```

### shutdown

Closes the OpenOCD server, disconnecting all clients (GDB, telnet, other). If option `error` is used, OpenOCD will return non-zero exit code to the parent process.

```
shutdown [error]
```

Example:

```
openocd -c "shutdown error"
```

### log\_output

Redirects logging to the filename; the initial log output channel is stderr.

```
log_output [filename]
```

**Example:**

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_1m_a0.cfg -c
"log_output d:/log.txt; targets; shutdown"
```

**debug\_level**

Displays the debug level. If *n* (from 0..3) is provided, then set it to that level. This affects the kind of messages sent to the server log. Level 0 is error messages only; Level 1 adds warnings; Level 2 adds informational messages; and Level 3 adds debugging messages. The default is Level 2, but that can be overridden on the command line along with the location of that log file (which is normally the server's standard output).

```
debug_level [n]
```

**Example:**

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_1m_a0.cfg -c
"debug_level 1; targets; shutdown"
```

**reset\_config**

Displays or modifies the reset configuration of your combination of the board and target.

```
reset_config <mode_flag> ...
```

The *mode\_flag* options can be specified in any order, but only one of each type – *signals*, *combination*, *gates*, *trst\_type*, *srst\_type* and *connect\_type* – may be specified at a time. If you don't provide a new value for a given type, its previous value (perhaps the default) remains unchanged. For example, do not say anything about TRST just to declare that if the JTAG adapter should want to drive SRST, it must explicitly be driven high (*srst\_push\_pull*).

*signals* specifies which of the reset signals is/are connected. For example, If the board doesn't connect SRST provided by the JTAG interface properly, OpenOCD cannot use it. The possible values are:

- none (the default)
- trst\_only
- srst\_only
- trst\_and\_srst

For more details, refer to the OpenOCD documentation.

**Example:**

```
openocd -s ../scripts -f interface/jlink.cfg -f target/traveo2_1m_a0.cfg -c
"reset_config trst_and_srst; targets; shutdown"
```

**adapter\_khz**

Sets a non-zero speed in KHZ for the debug adapter. Hence: 3000 is 3 MHz.

```
adapter_khz <max_speed_kHz>
```



JTAG interfaces usually support a limited number of speeds. The speed actually used will not be faster than the speed specified. Chip datasheets generally include a top JTAG clock rate. The actual rate is often a function of a CPU core clock, and is normally smaller than that peak rate. For example, most ARM cores accept at most one sixth of the CPU clock. Speed 0 (khz) selects the RTCK method. If your system uses RTCK, you will not need to change the JTAG clocking after a setup.

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag; adapter_khz 2000; shutdown"
```

## transport list

Displays the names of the transports supported by this version of OpenOCD.

Example:

```
openocd -c "transport list; shutdown"
```

## transport select

Selects which of the supported transports to use in this OpenOCD session.

```
transport select <transport_name>
```

When invoked with *transport\_name*, attempts to select the named transport. The transport must be supported by the debug adapter hardware and by the version of OpenOCD you are using (including the adapter's driver). If no transport has been selected and no *transport\_name* is provided, `transport select` auto-selects the first transport supported by the debug adapter. `transport select` always returns the name of the session's selected transport, if any.

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag"
```

## targets

With no parameter, this command displays a table of all known targets in a user-friendly form. With a parameter, this command sets the current target to a given target with a given *name*; this is only relevant to boards with more than one target.

```
targets [name]
```

Examples:

Displays all available targets of the connected Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "targets; shutdown"
```

Selects the CM4 core of the Traveo II device as the current target:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "targets traveo2.cpu.cm4; target current"
```

## scan\_chain

Displays the TAPs in the scan chain configuration, and their status. (Do not confuse this with the list displayed by the `targets` command. That only displays TAPs for CPUs configured as debugging targets.)

Example:

Displays JTAG TAPs of the Traveo II 1M device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag; adapter_khz 1000; init; scan_chain; shutdown"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/jlink.cfg -c "transport select jtag; adapter_khz 1000; init; scan_chain; shutdown"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.888 V
Info : clock speed 1000 kHz
Warn : There are no enabled taps. AUTO PROBING MIGHT NOT WORK!!
Info : JTAG tap: auto0.tap tap/device found: 0x6ba00477 (mfg: 0x23b <ARM Ltd.>, part: 0xba00, ver: 0x6)
Info : JTAG tap: auto1.tap tap/device found: 0x2e300069 (mfg: 0x034 <Cypress>, part: 0xe300, ver: 0x2)
Warn : AUTO auto0.tap - use "jtag newtap auto0 tap -irlen 4 -expected-id 0x6ba00477"
Warn : AUTO auto1.tap - use "jtag newtap auto1 tap -irlen 4 -expected-id 0x2e300069"
Warn : gdb services need one or more targets defined
```

TapName	Enabled	IdCode	Expected	IrLen	IrCap	IrMask
0 auto0.tap	Y	0x6ba00477	0x00000000	4	0x01	0x03
1 auto1.tap	Y	0x2e300069	0x00000000	4	0x01	0x03

```
shutdown command invoked
```

## md(w)(h)(b)

Displays the contents of address *addr*, as 32-bit words (*mdw*), 16-bit half-words (*mdh*), or 8-bit bytes (*mdb*).

```
mdw [phys] <addr> [count]
mdh [phys] <addr> [count]
mdb [phys] <addr> [count]
```

When the current target has a present and active MMU, *addr* is interpreted as a virtual address. Otherwise, or if the optional *phys* flag is specified, *addr* is interpreted as a physical address. If *count* is specified, displays that many units.

Examples:

Displays two 32-bit words of memory of the Traveo II 1M device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/traveo2_1m_a0.cfg -c "init; reset init; mdw 0x10000000 2; shutdown"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
0x10000000: 08024000 100014b9
shutdown command invoked
```

## mw(w)(h)(b)

Writes the specified *word* (32 bits), *halfword* (16 bits), or *byte* (8-bit) value, at the specified address *addr*.

```
mw  [phys] <addr> <word>
mwh [phys] <addr> <halfword>
mwb [phys] <addr> <byte>
```

When the current target has a present and active MMU, *addr* is interpreted as a virtual address. Otherwise, or if the optional *phys* flag is specified, *addr* is interpreted as a physical address.

Examples:

Writes a 32-bit word to the memory of the Traveo II 1M device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; mw 0x8000000 0xABCD1234; mdw
0x8000000; shutdown"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
0x08000000: abcd1234
shutdown command invoked
```

## init

This command terminates the configuration stage and enters the run stage. This helps to have the startup scripts manage tasks such as resetting the target, programming flash, etc. To reset the CPU upon a

startup, add "init" and "reset" at the end of the config script or at the end of the OpenOCD command line using the -c command line switch.

If this command does not appear in any startup/configuration file, OpenOCD executes the command for you after processing all configuration files and/or command-line options.

**Note** This command normally occurs at or near the end of your config file to force OpenOCD to initialize and make the targets ready. For example: If your config file needs to read/write memory on your target, initialization must occur before the memory read/write commands.

Example:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; shutdown"
```

## reset [run] [halt] [init]

Performs as hard a reset as possible, using SRST if possible. All defined targets will be reset, and target events will fire during the reset sequence.

The optional parameter specifies what should happen after a reset. If there is no parameter, a reset run is executed. The other options will not work on all systems. See [reset\\_config](#).

- `run` - Let the target run
- `halt` - Immediately halt the target
- `init` - Immediately halt the target, and execute the reset-init script.

Example:

Reset and Run the Traveo II 1M target:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset run; shutdown"
```

## program

Programs a given programming file in the HEX, SREC, ELF or BIN formats into the flash of the target device.

```
program <filename> [verify] [reset] [exit] [offset]
```

The only required parameter is *filename*, the others are optional.

- `verify` - Compares the contents of the binary file *filename* with the contents of the flash.
- `reset` - "reset run" is called if this parameter is given (see [reset](#) for details).
- `exit` - OpenOCD is shut down if this parameter is given.
- `offset` - A relocation offset may be specified, then it is added to the base address for each section in the image

The next example connects AutoFlashUtil to the J-Link probe with the Traveo II 1M target device, programs flash with the *firmware.hex* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "program d:/firmware.hex verify exit"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffe msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
** Programming Started **
auto erase enabled
Info : Padding image section 0 at 0x100017a4 with 92 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x10001800 .. 0x10007fff
[100%] [#####] [ Erasing ]
[100%] [#####] [ Programming ]
wrote 6144 bytes from file d:/firmware.hex in 0.334201s (17.953 KiB/s)
** Programming Finished **
** Verify Started **
verified 6052 bytes in 0.137082s (43.114 KiB/s)
** Verified OK **
shutdown command invoked
```

## flash banks

Prints a one-line summary of each flash bank of the target device.

Example (J-Link + Traveo II 1M device):

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; flash probe 0; flash probe 1; flash probe 2;
flash probe 3; flash banks; shutdown"
```

Output like the following should be displayed:

```

Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
swd
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.879 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
flash 'traveo21' found at 0x10000000
flash 'traveo21' found at 0x14000000
flash 'traveo21' found at 0x17000000
flash 'traveo21_efuse' found at 0x90700000
#0 : traveo2_main_cm0 (traveo21) at 0x10000000, size 0x00010000, buswidth 4, chipwidth 4
#1 : traveo2_work_cm0 (traveo21) at 0x14000000, size 0x00018000, buswidth 4, chipwidth 4
#2 : traveo2_super_cm0 (traveo21) at 0x17000000, size 0x00008000, buswidth 4, chipwidth 4
#3 : traveo2_efuse_cm0 (traveo21_efuse) at 0x90700000, size 0x00000400, buswidth 1, chipwidth 1
#4 : traveo2_main_cm4 (virtual) at 0x10000000, size 0x00000000, buswidth 0, chipwidth 0
#5 : traveo2_work_cm4 (virtual) at 0x14000000, size 0x00000000, buswidth 0, chipwidth 0
#6 : traveo2_super_cm4 (virtual) at 0x17000000, size 0x00000000, buswidth 0, chipwidth 0
#7 : traveo2_efuse_cm4 (virtual) at 0x90700000, size 0x00000400, buswidth 1, chipwidth 1
shutdown command invoked

```

## flash list

Retrieves a list of associative arrays for each device that was declared using a flash bank numbered from zero.

Example:

```

openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; flash list"

```

Output like the following should be displayed:

```

Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffe msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
<name traveo21 base 268435456 size 0 bus_width 0 chip_width 0> <name traveo21 base 335544320 size 0 bus_width 0 chip_width 0> <name traveo21 base 385875968 size 0 bus_width 0 chip_width 0> <name traveo21_efuse base 2423259136 size 1024 bus_width 1 chip_width 1> <name virtual base 268435456 size 0 bus_width 0 chip_width 0> <name virtual base 335544320 size 0 bus_width 0 chip_width 0> <name virtual base 385875968 size 0 bus_width 0 chip_width 0> <name virtual base 2423259136 size 1024 bus_width 1 chip_width 1>
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections

```

## flash info

```
flash info <num> [sectors]
```

Prints info about the flash bank *num*, a list of protection blocks and their status. Uses sectors to show a list of sectors instead. The *num* parameter is a value shown by flash banks. This command will first query the hardware, it does not print cached and possibly stale information.

Example:

Prints information about flash bank 0 of the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash info 0; shutdown"
```

Output like the following should be displayed:

```
# 39: 0x00102000 <0x2000 8kB> not protected
# 40: 0x00104000 <0x2000 8kB> not protected
# 41: 0x00106000 <0x2000 8kB> not protected
# 42: 0x00108000 <0x2000 8kB> not protected
# 43: 0x0010a000 <0x2000 8kB> not protected
# 44: 0x0010c000 <0x2000 8kB> not protected
# 45: 0x0010e000 <0x2000 8kB> not protected
Silicon ID: 0xE3EA2101
Protection: VIRGIN
```

## flash erase\_sector

Erase sectors in the bank *num*, starting at Sector *first* up to and including Sector *last*.

```
flash erase_sector <num> <first> <last>
```

The sector numbering starts at 0. Providing the *last* sector of *last* specifies "to the end of the flash bank". The *num* parameter is a value shown by flash banks.

Example:

Erases all sectors in flash bank 0 of the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash erase_sector 0 0 last;
shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [*****] [ Erasing ]
erased sectors 0 through 45 on flash bank 0 in 2.587552s
shutdown command invoked
```



## flash erase\_address

Erases sectors starting at *address* for the *length* bytes.

```
flash erase_address [pad] [unlock] <address> <length>
```

Unless *pad* is specified, *address* must begin a flash sector, and *address* + *length* - 1 must end a sector. Specifying *pad* erases extra data at the beginning and/or end of the specified region, as needed to erase only full sectors. The flash bank to use is inferred from the *address*, and the specified *length* must stay within that bank. As a special case, when *length* is zero and *address* is the start of the bank, the whole flash is erased. If *unlock* is specified, then the flash is unprotected before *erase* starts.

Example:

Erases the 32-KB block starting at address 0x10000000 of the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; flash erase_address 0x10000000
0x8000; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CY12B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x000002b8 msp: 0x0801f7f0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [#####] [ Erasing ]
erased address 0x10000000 (length 32768) in 0.106064s (301.705 KiB/s)
shutdown command invoked
```

## flash write\_bank

Writes the binary *filename* to flash bank *num*, starting at *offset* bytes from the beginning of the bank.

```
flash write_bank <num> <filename> <offset>
```

The *num* parameter is a value shown by flash banks.

Example:

Writes the binary file *firmware.bin* to flash bank 0 of the Traveo II 1M device starting at offset 0:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; flash write_bank 0 d:/firmware.bin
0x0; shutdown"
```



Output like the following should be displayed:

```
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x000002b8 msp: 0x0801f7f0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [#####] [ Programming ]
wrote 32768 bytes from file d:/firmware.bin to flash bank 0 at offset 0x00000000 in 0.807484s <39.629 KiB/s>
shutdown command invoked
```

## flash write\_image

Writes the image *filename* to the current target's flash bank(s).

```
flash write_image [erase] [unlock] <filename> [offset] [type]
```

Only loadable sections from the image are written. A relocation *offset* may be specified, then it is added to the base address for each section in the image. The file *[type]* can be specified explicitly as *bin* (binary), *ihex* (Intel hex), *elf* (ELF file), *s19* (Motorola s19). The relevant flash sectors will be erased prior to programming if the *erase* parameter is given. If *unlock* is provided, then the flash banks are unlocked before *erase* and *program*. The flash bank to use is inferred from the address of each image section.

**Warning** Be careful using the *erase* flag when the flash is holding data you want to preserve. Portions of the flash outside those described in the image's sections might be erased with no notice.

- When a section of the image being written does not fill out all the sectors it uses, the unwritten parts of those sectors are necessarily also erased, because sectors cannot be partially erased.
- Data stored in sector "holes" between image sections are also affected. For example, "flash write\_image erase ..." of an image with one byte at the beginning of a flash bank and one byte at the end erases the entire bank – not just the two sectors being written.

Also, when flash protection is important, you must re-apply it after it has been removed by the *unlock* flag.

Example:

Writes the ELF image *firmware.elf* to the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash write_image erase
d:/firmware.elf; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
auto erase enabled
Info : Padding image section 1 at 0x10091adc with 292 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x10091c00 .. 0x10097fff
[100%] [#####] [ Erasing ]
[100%] [#####] [ Programming ]
wrote 72704 bytes from file d:/firmware.elf in 1.912146s (37.131 KiB/s)
shutdown command invoked
```

## flash fill(w)(h)(b)

Fills flash memory with the specified word (32 bits), half-word (16 bits), or byte (8-bit) pattern, starting at *address* and continuing for *length* units (word/half-word/byte).

```
flash fillw <address> <word> <length>
flash fillh <address> <halfword> <length>
flash fillb <address> <byte> <length>
```

No *erase* is done before writing; when needed, that must be done before issuing this command. Writes are done in blocks of up to 1024 bytes, and each *write* is verified by reading back the data and comparing it to what was written. The flash bank to use is inferred from the address of each block, and the specified length must stay within that bank.

Example:

Fills the 32-KB block of the Traveo II 1M device memory starting at address 0x10000000 with the pattern 0x5A:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; flash erase_sector 0 0 last; flash
fillw 0x10000000 0x5A5A5A5A 0x2000; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [#####] [ Erasing ]
erased sectors 0 through 45 on flash bank 0 in 2.692615s
[100%] [#####] [ Programming ]
wrote 32768 bytes to 0x10000000 in 1.387832s (23.058 KiB/s)
shutdown command invoked
```

## flash read\_bank

Reads the *length* bytes from the flash bank *num* starting at *offset* and writes the contents to the binary *filename*. The *num* parameter is a value shown by flash banks.

```
flash read_bank <num> <filename> <offset> <length>
```

Example:

Reads the 32-KB block of bank #0 from the Traveo II 1M device memory and writes it to the binary file:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash read_bank 0 d:/read_bank_0.bin
0x0 0x8000; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x5a5a5a38
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
wrote 32768 bytes to file d:/read_bank_0.bin from flash bank 0 at offset 0x00000000 in 0.649390s (49.277 KiB/s)
shutdown command invoked
```

## flash verify\_bank

Compares the contents of the binary file *filename* with the contents of the flash *num* starting at *offset*. Fails if the contents do not match. The *num* parameter is a value shown by flash banks.

```
flash verify_bank <num> <filename> <offset>
```

Example:

Verifies the content of bank #0 of the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash verify_bank 0 d:/firmware.bin
0x0; shutdown"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
read 32768 bytes from file d:/firmware.bin and flash bank 0 at offset 0x00000000 in 0.644387s (49.660 KiB/s)
contents match
shutdown command invoked
```

## flash padded\_value

Sets the default value used for padding-any-image sections.

```
flash padded_value <num> <value>
```

This should normally match the flash bank erased value. If not specified by this command or the flash driver, then it defaults to 0xff.

Example:

Sets a padded value to 0xFF for bank #0 of the Traveo II device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash padded_value 0 0xFF; shutdown"
```

Output like the following should be displayed:

```
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Default padded value set to 0xff for flash bank 0
shutdown command invoked
```

## flash rmw

The command is intended to modify flash individual bytes.

```
flash rmw <address> <data>
```

The command can be used to program data to an arbitrary flash address preserving all data that belongs to the same flash sector.

- **address** – The start address for the programming.
- **data** – The hexadecimal string with data to be programmed. The format of the string is shown in the following example:

**Note** `flash rmw` is a custom command implemented in OpenOCD to extend its functionality.

Example:

Modifies 8 bytes of the Traveo II 1M device flash at address 0x10001234.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; flash rmw 0x10001234
DEADBEEFBAADC0DE; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x81000003 pc: 0x00000048 msp: 0xab503ca0
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [#####] [Erasing ]
[100%] [#####] [Programming ]
modified 8 byte(s) in 32768 byte region at 0x10000000 in 1.640984s <19.500 KiB/s>
shutdown command invoked
```

## add\_verify\_range

The command allows specifying memory regions to be compared during *verify* operation.

```
add_verify_range <target> <address> <size>
```

By default, when no regions are defined, all the regions present in the firmware image file are compared with corresponding target memory. This breaks the verification process for some non-memory-mapped regions such as EFuses. When the target has at least one *verify* region specified, only data that belongs to that *verify* region is verified.

- **target** – The target device to assign *verify* regions.
- **address** – The start address of the region.
- **size** – The size of the region, in bytes.

**Note** The `add_verify_range` command is a custom command implemented in OpenOCD to extend its functionality.

## show\_verify\_ranges

This command displays all active verify ranges for all targets that were added using the `add_verify_range` command. This command does not take any arguments.

**Example output:**

```
openocd.exe -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; show_verify_ranges; exit"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
traveo2.cpu.cm0 0x08000000 0x00800000
traveo2.cpu.cm0 0x10000000 0x00800000
traveo2.cpu.cm0 0x14000000 0x00800000
traveo2.cpu.cm0 0x17000000 0x00800000
traveo2.cpu.cm4 0x08000000 0x00800000
traveo2.cpu.cm4 0x10000000 0x00800000
traveo2.cpu.cm4 0x14000000 0x00800000
traveo2.cpu.cm4 0x17000000 0x00800000
```

**Note** The `show_verify_ranges` command is a custom command implemented in OpenOCD to extend its functionality.

**clear\_verify\_ranges**

This command deletes all verify ranges for the specified target that were added using the `add_verify_range` command.

```
clear_verify_ranges <target>
```

**Example output:**

```
openocd.exe -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; clear_verify_ranges traveo2.cpu.cm4;
show_verify_ranges; exit"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
traveo2.cpu.cm0 0x08000000 0x00800000
traveo2.cpu.cm0 0x10000000 0x00800000
traveo2.cpu.cm0 0x14000000 0x00800000
traveo2.cpu.cm0 0x17000000 0x00800000
```

**Note** The `clear_verify_ranges` command is a custom command implemented in OpenOCD to extend its functionality.

**verify\_image**

Verifies *filename* against target memory starting at *address*. The file format may optionally be specified (bin, ihex, or elf) This will first attempt a comparison using a CRC checksum, if that fails, it will try a binary compare.

```
verify_image <filename> <address> [bin|ihex|elf]
```

Example:

Verifies a *firmware.elf* image against the target memory of the Traveo II 1M device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; verify_image d:/firmware.elf 0x0;
shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
verified 72412 bytes in 1.223734s (57.786 KiB/s)
shutdown command invoked
```

## verify\_image\_checksum

Verifies *filename* against the target memory starting at *address*. The file format may optionally be specified (bin, ihex, or elf). This perform a comparison using a CRC checksum only.

```
verify_image_checksum <filename> <address> [bin|ihex|elf]
```

Example:

Verifies a *firmware.elf* image against the target memory of the Traveo II 1M device using the CRC check only.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; verify_image_checksum
d:/firmware.elf 0x0; shutdown"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x81000003 pc: 0x00000048 msp: 0xab503ca0
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
verified 72412 bytes in 1.221732s (57.881 KiB/s)
shutdown command invoked
```

## load\_image

Loads an image from file *filename* to the target memory offset by *address* from its load address. The file format may optionally be specified (bin, ihex, elf, or s19). Also, the following arguments may be specified: *min\_addr* - ignore data below *min\_addr* (this is w.r.t. to the target's load address + *address*) *max\_length* - maximum number of bytes to load.

```
load_image filename address [[bin|ihex|elf|s19] min_addr max_length]
```

Example:

Loads binary file *firmware.bin* to the RAM of the Traveo II 1M device.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; load_image d:/firmware.bin
0x8000000; shutdown"
```

Output like the following should be displayed:

```
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
32768 bytes written at address 0x08000000
downloaded 32768 bytes in 0.637382s (50.205 KiB/s)
shutdown command invoked
```

## dump\_image

Dumps *size* bytes of the target memory starting at *address* to the binary file named *filename*.

```
dump_image <filename> <address> size
```

Example:

Dumps 8KB of the Traveo II 1M device memory to file *dump\_mem.bin*.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; dump_image d:/dump_mem.bin
0x10001234 0x2000; shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x81000003 pc: 0x00000048 msp: 0xab503ca0
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
dumped 8192 bytes in 0.159095s (50.284 KiB/s)
shutdown command invoked
```



## MiniProg4 Driver Commands

The MiniProg4 (MP4) probe implements the CMSIS-DAP protocol defined by Arm with some extensions. Consequently, the MP4 driver in OpenOCD is a wrapper around the native CMSIS-DAP driver that extends its functionality with the MP4/KP3-specific extensions.

A full list of the CMSIS-DAP-specific configuration commands can be found in the OpenOCD official documentation.

Besides the standard CMSIS-DAP options, the MP4/KP3 driver exposes several custom Tcl configuration commands. All commands in this section must be prefixed with the name of the driver – “kitprog3”.

### kitprog3 acquire\_config

The command controls device acquisition parameters and optionally enables acquisition during the early initialization phase. Can be called at any time.

```
kitprog3 acquire_config <status> [target_type] [mode] [attempts]
```

- **status** – A mandatory parameter, enables or disables the acquisition procedure during the initialization phase. The possible values: on, off.
- **target\_type** – Specifies the target device type. This parameter is mandatory only if status=on. The possible values:
  - 0 – PSoC4
  - 1 – PSoC5
  - 2 – PSoC6
  - 3 – Traveo-II
- **mode** – Specifies Acquisition mode. This parameter is mandatory only if status=on. The possible values: 0 – Reset, 1 – Power Cycle.
- **attempts** – The number of attempts to acquire the target device. This parameter is mandatory only if status=on.

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/traveo2_lm_a0.cfg -c  
"kitprog3 acquire_config on 3 0; init; reset init; shutdown"
```

### kitprog3 acquire\_psoc

Performs device acquisition and is called only after the initialization phase. The acquisition procedure must be configured using acquire\_config prior to calling this command.

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/traveo2_lm_a0.cfg -c  
"kitprog3 acquire_config on 3 0; init; kitprog3 acquire_psoc; reset init; shutdown"
```

## kitprog3 power\_config

Controls the MP4 internal power supply parameters and optionally enables power during the early initialization phase. Can be called at any time.

```
kitprog3 power_config <status> [voltage]
```

- **status** – A mandatory parameter, enables or disables power supply during the initialization phase. The possible values: on|off.
- **voltage** – The power supply voltage in millivolts. This parameter is mandatory only if status=on.

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "kitprog3 power_config on 3300;
init; shutdown"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/kitprog3.cfg -c "kitprog3 power_config on 3300; init; shutdown"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1500 kHz
Info : CMSIS-DAP: SWD Supported
Info : CMSIS-DAP: FW Version = 2.0.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : kitprog3: powering up target device using KitProg3 (VTarg = 3300 mV)
Info : VTarg = 3.299 V
Info : clock speed 1500 kHz
Warn : gdb services need one or more targets defined
shutdown command invoked
```

## kitprog3 power\_control

The command turns on or off the MP4 internal power supply. Can be called only after the initialization phase.

```
kitprog3 power_control <status>
```

The voltage must be configured using `power_config` prior to calling this command.

- **status** – A mandatory parameter, enables or disables power supply: on|off.

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "kitprog3 power_config on 3300;
init; kitprog3 power_control off; shutdown"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/kitprog3.cfg -c "kitprog3 power_config on 3300; init; kitprog3 power_control off; shutdown"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1500 kHz
Info : CMSIS-DAP: SWD Supported
Info : CMSIS-DAP: FW Version = 2.0.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : kitprog3: powering up target device using KitProg3 (UTarg = 3300 mV)
Info : UTarg = 3.300 V
Info : clock speed 1500 kHz
Warn : gdb services need one or more targets defined
Info : kitprog3: powering down target device using KitProg3
shutdown command invoked
```

## kitprog3 led\_control

Controls the MP4 LEDs. Can be called only after the initialization phase.

```
kitprog3 led_control <type>
```

- **type** – A mandatory parameter, specifies the type of the LED indication. The possible values:
  - 0 – READY
  - 1 – PROGRAMMING
  - 2 – SUCCESS
  - 3 – ERROR

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "init; kitprog3 led_control 2"
```

## kitprog3 get\_power

Reports the target voltage in millivolts. Can be called only after the initialization phase.

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "init; kitprog3 get_power; shutdown"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f interface/kitprog3.cfg -c "init; kitprog3 get_power; shutdown"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1500 kHz
Info : CMSIS-DAP: SWD Supported
Info : CMSIS-DAP: FW Version = 2.0.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : UTarg = 3.300 V
Info : clock speed 1500 kHz
Warn : gdb services need one or more targets defined
UTarg = 3.300 V
shutdown command invoked
```

## Flash Driver Commands

This section contains flash driver commands for Traveo II devices.

### traveo2 sflash\_restrictions

The command enables or disables writes to Sflash regions other than USER, NAR, TOC2, and KEY.

```
traveo2 sflash_restrictions <mode>
```

The command can be called at any time. Writes to these regions are possible only on the VIRGIN silicon, so the command is mostly intended for internal use. It is useful for flash boot developers and validation teams. Note that *erase* (performed by programming with 0xFF for Traveo II) is performed only for the USER, NAR, TOC2, and KEY regions; it is skipped for other Sflash regions regardless of this command.

- **mode** – A mandatory parameter, specifies the behavior of Sflash programming. The possible values:
  - 0 – Erase/Program of Sflash is prohibited
  - 1 – Erase and Program of USER/TOC/KEY is allowed
  - 2 – Erase of USER/TOC/KEY and Program of USER/TOC/KEY/NAR is allowed
  - 3 – Erase of USER/TOC/KEY and Program of the whole Sflash region is allowed.

Example (Traveo II 1M device):

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; traveo2 sflash_restrictions 2;
shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffe msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Warn : SFlash programming allowed for regions: USER, TOC, KEY, NAR
shutdown command invoked
```

### traveo2 add\_safe\_sflash\_region

This command is intended to define memory regions in SFlash which are safe to erase and/or program. It also maps different operation types (Program or Erase) to different *sflash\_restrictions* value.

The command has the following syntax:

```
traveo2 add_safe_sflash_region <address> <size> <allowed_operation_bitmap>
```

First two parameters define address and size of the SFlash region. Third parameter is a single byte which is interpreted as binary mask which controls types of allowed flash operations on different *sflash\_restrictions* level. The following table shows bit mapping of this parameter:

Bit number	7	6	5	4	3	2	1	0
Operation type	Program				Erase			
<i>sflash_restrictions</i> level	3	2	1	0	3	2	1	0

Given flash operation is allowed at given *sflash\_restrictions* level if corresponding bit is set.

For example, for hexadecimal value 0xC0:

Bit number	7	6	5	4	3	2	1	0
Operation type	Program				Erase			
<i>sflash_restrictions</i> level	3	2	1	0	3	2	1	0
<b>Bit value (0xC0)</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

This value can be decoded as follows: Programming of the region is allowed only at *sflash\_restrictions* level 2 and 3. Erase of the region is disallowed at any *sflash\_restrictions* level. Disallowed operations will be skipped with a warning.

## traveo2 allow\_efuse\_program

Allows or disallows writes to the EFuse region. Can be called any time. Writes to the EFuse region are skipped by default. EFuse programming must be allowed for Life Cycle transitions to work.

```
traveo2 allow_efuse_program <on|off>
```

Example:

Writes 1 bit to the EFuse region at address 0x907003FC of the Traveo II 1M device:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; traveo2 allow_efuse_program on;
flash fillb 0x907003FC 1 1; flash read_bank 3 d:/dump_efuse.bin 0x3FC 0x1;
shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Warn : Programming of efuses now ALLOWED
Info : Start address 0x907003fc breaks the required alignment of flash bank traveo2_efuse_cm0
Info : Padding 1020 bytes from 0x90700000
Info : Padding at 0x907003fd with 3 bytes (bank write end alignment)
Info : The Life Cycle stage is not present in the programming file
wrote 1 bytes to 0x907003fc in 0.077046s (0.013 KiB/s)
wrote 1 bytes to file d:/dump_efuse.bin from flash bank 3 at offset 0x000003fc in 0.033020s (0.030 KiB/s)
>
shutdown command invoked
```

## traveo2 reset\_halt

The command simulates a broken Vector Catch on Traveo II devices.

```
traveo2 reset_halt <mode>
```

The command retrieves the address of the Vector Table from the VECTOR\_TABLE\_BASE registers, detects the location of the application entry points, sets a hardware breakpoint at that location and performs a reset of the target. The type of the reset can be specified by the optional <mode> parameter.

Parameters:

- **mode** – The type of a reset to be performed. The possible values are `sysresetreq` and `vectreset`. This parameter is optional. If it is not specified, `SYSRESETREQ` is used for the CM0 core and `VECTRESET` is used for other cores in the system.

Example (Traveo II 1M device):

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_1m_a0.cfg -c "init; reset init; traveo2 reset_halt vectreset;
shutdown"
```

## traveo2 ecc\_error\_reporting

Enables or Disables the ECC error reporting during AutoFlashUtil operations.

```
traveo2 ecc_error_reporting <on|off>
```

OpenOCD supports the detection and reporting of ECC errors during the flash *read* operation. In the current implementation, AutoFlashUtil reads word-by-word a requested amount of data and checks for the ECC status after each Read. This ensures all ECC errors for all memory locations are properly detected. If an ECC error occurs, AutoFlashUtil retrieves the address of the faulty location from the hardware. All ECC errors along with their locations are reported to the user by means of warning messages. This process will be performed until all requested data has been read.

Example (Traveo II 1M device):

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; traveo2 ecc_error_reporting on;
shutdown"
```

Output like the following should be displayed:

```
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02472
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Info : ECC error reporting is now Enabled
shutdown command invoked
```

## enable\_hyperram

This command is applicable for TVII-B-H-8M devices only. This command is intended to initialize external HyperRAM bank and map it to CPU address space. Command can be issued any time after 'init; reset init'

## enable\_hyperram\_x

This command is applicable for TVII-C-2D-6M devices only. This command is intended to initialize external HyperRAM bank connected to MII-Fx peripheral and map it to CPU address space. Command can be issued any time after 'init; reset init'.

The 'x' can have of the following values:

- 0 – External memory connected to SMIF0 peripheral
- 1 – External memory connected to SMIF1 peripheral

## Other Commands

### source

Reads a file and executes it as a script. It is usually used with the result of the *find* command.

```
source [find FILENAME]
```

Example (J-Link + Traveo II target):

```
openocd -s ../scripts -c "source [find interface/jlink.cfg]; source [find
target/traveo2_lm_a0.cfg]; targets; shutdown"
```

## find

Finds and returns a full path to a file with a given name. It is usually used as an argument of the `source` command. This command uses an internal search path. (Do not try to use a filename which includes the `"#"` character. That character begins Tcl comments.)

```
source [find FILENAME]
```

Example:

```
openocd -s ../scripts -c "source [find interface/jlink.cfg]; source [find
target/traveo2_lm_a0.cfg]; targets; shutdown"
```

## set

Stores a value to a named variable, first creating the variable if it does not already exist.

```
set VARNAME value
```

Example:

```
openocd -s ../scripts -c "set ENABLE_CM0 0; source [find interface/jlink.cfg];
transport select swd; source [find target/traveo2_lm_a0.cfg]; targets; shutdown"
```

## sleep

Waits for at least `msec` milliseconds before resuming. Useful in a combination with script files.

```
sleep msec
```

Example:

```
openocd -c "sleep 1000; shutdown"
```

## add\_script\_search\_dir

Adds a directory to a file/script search path. Equivalent to the `--search` command-line option.

```
add_script_search_dir [directory]
```

Example:

```
openocd -c "add_script_search_dir ../scripts; source [find interface/jlink.cfg];
transport select swd; source [find target/traveo2_lm_a0.cfg]; targets; shutdown"
```



## 8 Global Variables



The global variables listed below control the behavior of a target configuration file (traveo2.cfg, etc.). They are set in the command line prior to any configuration file, such as kitprog3.cfg or traveo2\_1m\_a0.cfg. See the command [set](#) for details.

### ENABLE\_ACQUIRE

Enables or disables acquisition of the target device in Test mode.

The possible values:

- 1 – Acquisition enabled (default with MiniProg4).
- 0 – Acquisition disabled (default for other debug adapters).

### ENABLE\_POWER\_SUPPLY

Controls internal power supply of MiniProg4 adapter. If this command is specified, the MiniProg4 driver, enables power supply thus powering on the target during initialization.

The possible values:

- 0 – Power supply disabled.
- Any other value defines target voltage in millivolts.

### ENABLE\_HYPERFLASH, ENABLE\_DUALQUADSPI

These variables are applicable for TVII-B-H-4M and TVII-B-H-8M devices only. Allows to turn on corresponding External Flash Banks connected to SMIF0 peripheral.

The possible values:

- 1 – Corresponding flash bank is enabled.
- 0 – Flash bank is disabled (default).

### ENABLE\_HYPERFLASH\_x, ENABLE\_DUALQUADSPI\_x

These variables are applicable for TVII-C-2D-6M devices only. Allows to turn on corresponding External Flash Banks connected to SMIFx peripheral.

The 'x' can have of the following values:

- 0 – External memory connected to SMIF0 peripheral
- 1 – External memory connected to SMIF1 peripheral

The possible values:

- 1 – Corresponding flash bank is enabled.
- 0 – Flash bank is disabled (default).

## **ENABLE\_SEMPERFLASH\_0**

This variable is applicable for TVII-C-2D-6M devices only. Allows to turn on corresponding External Semper Flash Bank connected to SMIF0 peripheral.

The possible values:

- 1 – Corresponding flash bank is enabled.
- 0 – Flash bank is disabled (default).

## **HYPERFLASH\_LOADER, HYPERRAM\_LOADER, DUALQUADSPI\_LOADER**

These variables are applicable for TVII-B-H-4M and TVII-B-H-8M devices only. They allow to override Flash Loader binary to be used with corresponding SMIF Flash micro. Flash Loader binaries are located in \flm\cypress\traveo2\ subfolder. This variable should contain full filename of the loader with file extension.

Example usage:

```
set SEMPERFLASH_LOADER_0 TV2_6M_SI_A0_SemperFlash.elf
```

## **SEMPERFLASH\_LOADER\_0, HYPERFLASH\_LOADER\_0, HYPERRAM\_LOADER\_0, DUALQUADSPI\_LOADER\_0**

These variables are applicable for TVII-C-2D-6M device only. They allow to override Flash Loader binary to be used with corresponding SMIF Flash micro. Flash Loader binaries are located in \flm\cypress\traveo2\ subfolder. This variable should contain full filename of the loader with file extension.

Example usage:

```
set DUALQUADSPI_LOADER TV2_8M_SI_B0_DualQuadSPI.elf
```

## **MAIN\_LARGE\_SECTOR\_NUM, MAIN\_SMALL\_SECTOR\_NUM, WORK\_LARGE\_SECTOR\_NUM, WORK\_SMALL\_SECTOR\_NUM**

These variables are defined in corresponding device configuration files (e.g. traveo2\_2m.cfg) and cannot be overridden from the command line. They are intended to simplify implementation of configuration files for new Traveo-II devices. Variables define the number of large and small sectors for Main and Work Flash. New devices with different flash sizes can be easily added by using one of the existing config files as a template and modifying values of these variables.

As a rule, if new device is a dual-core part, such as TraveoII-1M, traveo2\_1m\_a0.cfg should be used as a template. For triple-code devices please use traveo2\_8m.cfg file.

## 9 Error Codes

The OpenOCD tool returns 0 as an error code on successful completion, on a failure it returns 1.

## 10 Usage Examples



All the examples in this chapter assume you have a Traveo II target device connected to the PC via the MiniProg4 or J-Link debug probe. The current working directory is the default install directory (for example, C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin on Windows x64).

For convenience, the *traveo2\_mp4\_board.cfg* config file has been created in the same directory as the OpenOCD executable. The file contains the default configuration allowing programming the Traveo II 1M device connected to MiniProg4:

```
source [ find interface/kitprog3.cfg ]
source [ find target/traveo2_1m_a0.cfg ]
]
init
reset init
```

Another config file - *jlink\_traveo2.cfg*, can be used for working with Traveo II devices. The file contains the following configuration allowing programming the Traveo II 1M device connected to J-Link Probe:

```
source [ find interface/jlink.cfg ]
transport select swd
source [ find target/traveo2_1m_a0.cfg ]
init
reset init
```

See [Supported Target Configurations](#) for the detailed list of available target devices and corresponding OpenOCD configuration files.

## Erase Main Flash Rows 0...10 of Traveo II 1M device

```
openocd -s ../scripts -f jlink_traveo2.cfg -c "flash erase_sector 0 0 10; exit"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f jlink_traveo2
.cfg -c "flash erase_sector 0 0 10; exit"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.885 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CWT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0xab503ca0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
[100%] [#####] [Erasing ]
erased sectors 0 through 10 on flash bank 0 in 0.866519s
```

## Display Memory Contents (32 words at address 0x08000000) of Traveo II 1M device

```
openocd -s ../scripts -f jlink_traveo2.cfg -c "mdw 0x08000000 32; exit"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f jlink_traveo2
.cfg -c "mdw 0x08000000 32; exit"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.879 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x000002b8 msp: 0x0801f7f0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
0x08000000: e7feb662 b3b43e16 37fc4650 6b0686c5 aa7672ec f6368a00 b592acbb 7afb2b18
0x08000020: 418232c0 2159f0f6 23b6a4cc 6d41b78c b0b5011a 16ffccf4 e11a695f 7f84ec0d
0x08000040: 5d620e1f 6ccc170 0e1b3d85 de387b68 6fba5856 c698cda5 76480f40 a5cc6b86
0x08000060: 9e229960 dccc9410 b66d11f4 e28049be 5fe97b97 857c7914 e41676d3 cc724489
```

## Program Traveo II 1M Device with Verification (Intel HEX file)

OpenOCD supports programming of the elf, Intel HEX, Motorola SREC, and binary file formats. For binary files, the relocation offset must be specified as an argument to the *program* command.

```
openocd -s ../scripts -f jlink_traveo2.cfg -c "program d:/BlinkyLED.hex verify
reset; exit"
```

Output like the following should be displayed:

```
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.877 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (<A>)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x000002b8 msp: 0x0801f7f0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x81000000 pc: 0x17002100 msp: 0x0801f7f0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
** Programming Started **
auto erase enabled
Info : Padding image section 0 at 0x100017a4 with 92 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x10001800 .. 0x10007fff
[100%] [#####] [ Erasing ]
[100%] [#####] [ Programming ]
wrote 6144 bytes from file d:/BlinkyLED.hex in 0.308185s (19.469 KiB/s)
** Programming Finished **
** Verify Started **
verified 6052 bytes in 0.138082s (42.802 KiB/s)
** Verified OK **
** Resetting Target **
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
```

## Program EFuse region of Traveo II Device

This example writes a single bit of data to the EFuse region of the Traveo II 1M device at address 0x907003FB:

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f
target/traveo2_lm_a0.cfg -c "init; reset init; traveo2 allow_efuse_program on;
flash fillb 0x907003FB 1 1; flash read_bank 3 d:/dump_efuse.bin 0x3FB 0x1; exit"
```

Output like the following should be displayed:

```
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
swd
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.887 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Warn : Programming of efuses now ALLOWED
Info : Start address 0x907003fb breaks the required alignment of flash bank traveo2_efuse_cm0
Info : Padding 1019 bytes from 0x90700000
Info : Padding at 0x907003fc with 4 bytes (bank write end alignment)
Info : The Life Cycle stage is not present in the programming file
wrote 1 bytes to 0x907003fb in 0.086052s (0.011 KiB/s)
wrote 1 bytes to file d:/dump_efuse.bin from flash bank 3 at offset 0x000003fb in 0.036021s (0.027 KiB/s)
>
```



## Modify Individual Bytes of Traveo II 1M in Main Flash and Display Results

```
openocd -s ../scripts -f jlink_traveo2.cfg -c "mdw 0x10000000 8; flash rmw
0x10000002 11223344; mdw 0x10000000 8; exit"
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f jlink_traveo2
.cfg -c "mdw 0x10000000 8; flash rmw 0x10000002 11223344; mdw 0x10000000 8; exit"
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.887 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CYT2B78CAE
** Flash Boot version 1.00
** Chip Protection: UIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or conf
igure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffe msp: 0x08023fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
0x10000000: 08024000 100014b9 0000000d 1000151d 00000000 00000000 00000000 00000000
[100%] [#####] [Erasing ]
[100%] [#####] [Programming ]
modified 4 byte(s) in 32768 byte region at 0x10000000 in 1.592955s (20.088 KiB/s)
0x10000000: 22114000 10004433 0000000d 1000151d 00000000 00000000 00000000 00000000
```

## Start GDB Server and Leave It Running

```
openocd -s ../scripts -f jlink_traveo2.cfg
```

Output like the following should be displayed:

```
C:\Program Files (x86)\Cypress\Cypress Auto Flash Utility 1.0\bin>openocd -s ../scripts -f jlink_traveo2
.cfg
Open On-Chip Debugger 0.10.0+dev-1.0.0.101 (2019-06-21-07:01)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
adapter speed: 1000 kHz
** Test Mode acquire not supported by selected adapter
cortex_m reset_config sysresetreq
cortex_m reset_config vectreset
adapter_nsrst_delay: 100
Info : J-Link V9 compiled Oct 25 2018 11:46:07
Info : Hardware version: 9.30
Info : VTarget = 4.875 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : traveo2.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
*****
** Use overridden Main Flash size, kb: 1088
** Use overridden Work Flash size, kb: 96
** Silicon: 0xE3EA, Family: 0x101, Rev.: 0x21 (*A)
** Detected Device: CWT2B78CAE
** Flash Boot version 1.00
** Chip Protection: VIRGIN
*****
Info : traveo2.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Info : Listening on port 3334 for gdb connections
Info : SWD DPIDR 0x6ba02477
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or config
ure hardware srst support.
** traveo2.cpu.cm0: Ran after reset and before halt...
target halted due to debug-request, current mode: Handler HardFault
xPSR: 0x61000003 pc: 0xffffffff msp: 0x22113fe0
Info : Vector Table address invalid (0x00000000), reset_halt skipped
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000010c msp: 0x0801f800
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
```

## 11 Documentation and Links



OpenOCD v0.10.0 User Guide:

<http://openocd.org/doc-release/pdf/openocd.pdf>