ModusToolbox™ software OpenOCD CLI user guide

About this document

There may be a newer version of this document available here.

Scope and purpose

This user guide provides technical information for the ModusToolbox™ version of the OpenOCD command line tool, including how to use it in stand-alone mode. OpenOCD is an Open Source programmer/debugger software that is installed as part of either the ModusToolbox™ Programming tools package.

Intended audience

This document is intended for anyone who wants to use the OpenOCD CLI as a stand-alone tool.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Monospace`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- 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.
- MPN – Marketing part number. Associated with each specific device and used to order a device or find information about a device from Infineon. For example, CY8C616FMI-BL603, CY8C616FMI-BL673.
- NVM – Non-Volatile Memory
- 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™ – A family of microcontroller integrated circuits by Infineon. These chips include a CPU core and mixed-signal arrays of configurable integrated analog and digital peripherals.
- MCU – Microcontroller unit.
- 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.
- WFA – 'Wait for action' special inner mechanism of bootloader to execute non-standard actions.
- mp4/kp3 – MiniProg4/KitProg3 external and on-board Infineon brand probes.

About this document

Reference documents

Refer to the following documents for more information as needed:

- OpenOCD v0.12.0 user guide: http://openocd.org/doc-release/pdf/openocd.pdf
- ModusToolbox™ software installation guide
- ModusToolbox™ Programming tools release notes
- ModusToolbox™ Programmer GUI user guide

Table of contents

About this document . . . 1

Table of contents . . . 3

1 Introduction . . . 5
1.1 Overview . . . 5
1.2 Supported OS . . . 5
1.3 Supported devices . . . 5
1.4 Supported hardware (probes) . . . 6
1.5 Installation . . . 6
1.6 Error codes . . . 6

2 Getting started . . . 7
2.1 Connect the device . . . 7
2.2 List the connected targets . . . 7
2.3 Program the target MCU . . . 8
2.4 Using configuration files only . . . 9
2.5 Using configuration files and command line . . . 9
2.6 Using multiple commands . . . 10
2.7 Remote debugging . . . 10

3 Supported target configurations . . . 11

4 Command-line options . . . 13
4.1 --file (-f) . . . 13
4.2 --search (-s) . . . 14
4.3 --command (-c) . . . 14
4.4 --debug (-d) . . . 15
4.5 --log_output (-l) . . . 15
4.6 --help (-h) . . . 15
4.7 --version (-v) . . . 16

5 OpenOCD commands overview . . . 17

6 OpenOCD commands description . . . 20
6.1 General OpenOCD commands . . . 20
6.2 KitProg3/MiniProg4 driver commands . . . 37
6.3 Flash driver commands . . . 41
6.4 cmsis_flash flash driver commands . . . 46
6.5 Other commands . . . 47
6.6 CYW4390x commands . . . 49
6.7 AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip commands . . . 49

7 Global variables . . . 51
7.1 PSOC™ Edge E84 global variables . . . 51

Table of contents

7.2 PSOC™ Control C3 global variables . . . 52
7.3 PSOC™ 6 MCU global variables . . . 53
7.4 PSOC™ 4 MCU global variables . . . 55
7.5 AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip global variables . . . 55
7.6 XMC7xxx/XMC5xxx and TRAVEO™ T2G global variables . . . 56

8 Usage examples . . . 57
8.1 Erase main flash rows 0…10 of PSOC™ Edge MCU . . . 57
8.2 Display memory contents of PSOC™ Edge MCU . . . 58
8.3 Program the PSOC™ Edge MCU with verification (Intel HEX file) . . . 58
8.4 Program the EFuse region of PSOC™ 6 MCU . . . 59
8.5 Modify individual bytes of PSOC™ Edge MCU in main flash and display results . . . 60
8.6 Read the memory of PSOC™ Edge MCU to binary file . . . 61
8.7 Start the GDB server and leave it running . . . 61
8.8 Power example . . . 62
8.9 Define and use QSPI flashloader . . . 63

Revision history . . . 65

Disclaimer . . . 68

1 Introduction

This chapter includes:

- Overview
- Supported OS
- Supported devices
- Supported hardware (probes)
- Installation
- Error codes

1.1 Overview

The ModusToolbox™ OpenOCD command-line interface (CLI) is based on the Open On-Chip Debugger (OpenOCD) product. 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.

This document covers the ModusToolbox™-specific CLI extensions of OpenOCD. See the official documentation at:

http://openocd.org/documentation/

The latest released version of ModusToolbox™ OpenOCD is available from the GitHub repository:

https://github.com/Infineon/openocd/releases

1.2 Supported OS

- Windows 11 (x64)
- Ubuntu 22.04 "Jammy Jellyfish", Ubuntu 24.04 "Noble Numbat", Ubuntu 26.04 "Resolute Racoon"
- macOS 14 "Sonoma", macOS 15 "Sequoia", and macOS 26 "Tahoe" (native Arm processors)

1.3 Supported devices

- AIROC™ CYW20809 Bluetooth® LE system on chip
- PSOC™ 6 and PSOC™ 64
- PSOC™ 4, PSOC™ 4 HVPA, PSOC™ 4 HVMS
- PMG1, CCGx
- AIROC™ CYW4390x [1)]
- XMC7xxx/ XMC5xxx
- TRAVEO™ T2G Body High
- TRAVEO™ T2G Body Entry
- TRAVEO™ T2G Cluster 4M/6M/6M DDR MCU
- TRAVEO™ Cluster Entry 4M MCU
- FX3G2 and FX2G3
- PAG2S
- PSOC™ Control C3
- PSOC™ Edge E84

[1] Currently, OpenOCD does not provide a "built-in" Flash driver for the CYW4390x chip. All Flash-related operations are fully implemented in the TCL scripts. The behavior of the TCL-based driver is slightly different from the built-in one. Refer to the CYW4390x commands section for details.

1.4 Supported hardware (probes)

- SEGGER J-Link stand-alone programmer (via libusbK driver for pre-2022 year probes)
- SEGGER J-Link on-board programmer
- Infineon KitProg3 on-board programmer
- Infineon MiniProg4 standalone programmer
- FTDI-based adapter on CYW954907AEVAL1F / CYW943907AEVAL1F kits

1.5 Installation

The ModusToolbox™ OpenOCD CLI software is installed as part of either ModusToolbox™ Programming tools package. Refer to the ModusToolbox™ Programming tools release notes for details.

You can also download the latest version from the GitHub repository:

https://github.com/Infineon/openocd/releases

1.5.1 Enable J-Link for work with OpenOCD

On Windows, to use pre-2022 year J-Link, update the driver as described in the article from SEGGER Knowledge Base of OpenOCD support. This ModusToolboxTM version of OpenOCD was tested with the libusbK v3.1.0.0 driver for the J-Link probe.

For on-board programmers and post 2022 standalone probes, configure J-Link to use native WinUSB driver as described in J-Link WinUSB driver selection.

1.6 Error codes

The OpenOCD tool returns '0' as the response code on successful completion; on a failure, it returns '1'.

2 Getting started

This chapter includes:

- Connect the device
- List the connected targets
- Program the target MCU
- Using configuration files only
- Using configuration files and command line
- Using multiple commands
- Remote debugging

2.1 Connect the device

Connect the host computer to a probe or kit device; e.g. KitProg3 kit with the PSOC™ Edge MCU target, used in the following examples. Make sure that the target MCU is attached to your probe.

2.2 List the connected targets

This example displays the target names available for the PSOC™ Edge MCU connected to the KitProg3 programmer. The programmer communicates with the PSOC™ Edge MCU over the SWD interface.

2.2.1 Windows:

1. Open a command-line window, enter the following command to change the directory to the ModusToolbox™ Programming tools or ModusToolbox™ tools installation folder:

```
cd %installation folder%\openocd\bin
```

2. Run the following command:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"targets; shutdown"
```

```
Open On-Chip Debugger 0.12.0+dev-5.12.0.4118 (2025-11-20-06:34)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Info : auto-selecting first available session transport "swd". To override use 'transport select <transport>'
***************************************
** SERIES:   PSE84
** DEVICE:   PSE84xGxS2
** BOARD:    Generic
***************************************
transport: swd
adapter name: kitprog3
adapter speed: 4000 kHz
adapter srst delay: 50
cortex_m reset_config sysresetreq
Info : Using CMSIS-flash algorithms 'PSE84_RRAM_NVM' for bank 'cat1d.cm33.main_ns' (footprint 5752 bytes)
Info : CMSIS-flash: ELF path: ../flm/infineon/pse8xxgp/PSE84_RRAM_NVM.FLM
Info : CMSIS-flash: Address range:     0x22000000-0x2207FFFF
Info : CMSIS-flash: Program page size: 0x00000400 bytes
Info : CMSIS-flash: Erase sector size: 0x00000400 bytes, unified
    TargetName         Type       Endian TapName            State
--  ------------------ ---------- ------ ------------------ ------------
 0  cat1d.sys          mem_ap     little cat1d.cpu          unknown
 1  cat1d.sys33        mem_ap     little cat1d.cpu          unknown
 2* cat1d.cm33         cortex_m   little cat1d.cpu          unknown
shutdown command invoked
```

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

2.2.2 Linux:

1. Open the terminal window, go to the directory where ModusToolbox™ Programming tools package is installed (for example, ~/openocd/bin).
2. Run the following command:

```
./openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"targets; shutdown"
```

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

2.2.3 macOS:

1. Open the terminal window, go to the directory where the ModusToolbox™ Programming tools package is installed (for example, ~/openocd/bin).
2. Run the following command:

```
./openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"targets; shutdown"
```

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

2.3 Program the target MCU

These examples show how to initialize the KitProg3 probe with different MCUs, program the flash with the appropriate *firmware_[device].hex* file, verify the programmed data, and finally shut down the OpenOCD programmer.

Example (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "set SMIF_BANKS { 0 {addr 0x60000000 size
0x01000000 psize 0x00000200 esize 0x00040000} }" -f target/infineon/pse84xgxs2.cfg -c "program
c:/hex/firmware_pse8.hex verify exit"
```

Example (KitProg3 & PSOC™ 6 MCU

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8cxxa.cfg -c "program
c:/hex/firmware_psoc6.hex verify exit"
```

Example (KitProg3 & PSOC™ 64 "Secure Boot" MCU

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cyxx64x7.cfg -c "program
cd:/hex/firmware_psoc64.hex verify exit"
```

***Note**: The cyxx64x7.cfg configuration file programming of the internal flash is performed via the SYS_AP access port. OpenOCD will not affect CM0_AP and CM4_AP by default, so both cores will not be visible to OpenOCD. Choose the access port using the TARGET_AP variable.*

Programming of the external memory is done by the flash loader, so the CM4 access port must be used for QSPI memory programming. After choosing the CM4 access port, the QSPI memory bank will be exposed automatically.

Example (KitProg3 & PSOC™ 4 MCU

```
openocd -s ../scripts -f interface/kitprog3.cfg -f  target/infineon/psoc4.cfg -c "program
cd:/hex/firmware_psoc4.hex verify exit"
```

2.4 Using configuration files only

The whole configuration can be stored in a single *sample.cfg* file. This approach is useful, when the operations to be performed will be repeated without any change.

For example, the following configuration file describes the PSOC™ Edge MCU connected using the KitProg3 debug probe. This file initializes the target device, programs the flash with the *firmware.hex* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

```
source [ find interface/kitprog3.cfg ]
transport select swd
set SMIF_BANKS { 0 {addr 0x60000000 size 0x01000000 psize 0x00000200 esize 0x00040000} }
source [ find target/infineon/pse84xgxs2.cfg]
program d:/firmware.hex verify exit
```

Execute the following command:

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

2.5 Using configuration files 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.

For more information on Tcl scripting, see Tcl Crash Course.

The example *sample.cfg* file contents:

```
source [ find interface/kitprog3.cfg ]
transport select swd
set SMIF_BANKS { 0 {addr 0x60000000 size 0x01000000 psize 0x00000200 esize 0x00040000} }
source [ find target/infineon/pse84xgxs2.cfg]
```

Execute the following command:

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

2.6 Using multiple commands

In general, OpenOCD can execute multiple commands defined using Tcl or defined in configuration files. The order of execution is subsequent and defined by the inclusion in the OpenOCD command.

So, in the following example, the commands defined in the *sample.cfg* file will be executed first, and the programming operation second:

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

2.7 Remote debugging

Remote debugging is described for Eclipse and VS Code in the KBA Remote debugging a ModusToolbox™ application.

3 Supported target configurations

Target configuration files are in the *target/infineon* directory of the OpenOCD tree. To connect the ModusToolbox™ OpenOCD CLI to a device, pass one of the following configuration files as the argument for the --file (-f) command-line option; for example, `-f target/infineon/psoc4.cfg`.

The chip name must be used with flash driver commands. To get the correct chip name for each config, use the following table.

***Note**: Old target config files will be removed in the future releases of OpenOCD, consider using new config files.*

Old target config	Target config	Chip name	Supported devices
N/A	infineon/pse84xgxs2.cfg	cat1d	PSE84xGxS2 devices of PSOC™ Edge E84 series:EPC2 security, Total NVM: 512 KB, Programmable NVM: 356KB
N/A	infineon/pse84xgxs4.cfg	cat1d	PSE84xGxS4 devices of PSOC™ Edge E84 series:EPC4 security, Total NVM: 512 KB, Programmable NVM: 188KB
N/A	infineon/psc3.cfg	psc3	PSC3x2, PSC3x3, PSC3x5 series of PSOC™ Control C3 devices
N/A	infineon/psc3x6.cfg	psc3	PSC3x6 series of PSOCTM Control C3 devices
N/A	infineon/psc3x7.cfg	psc3	PSC3x7 series of PSOCTM Control C3 devices
N/A	infineon/psc3x8.cfg	psc3	PSC3x8 series of PSOCTM Control C3 devices
psoc6.cfg	infineon/cy8c6xx.cfg	psoc6	CY8C6xx7, CY8C6xx6
psoc6_2m.cfg	infineon/cy8cxxa.cfg	psoc6	CY8C6xxA, CY8C6xx8
psoc6_512k.cfg	infineon/cy8cxx5.cfg	psoc6	CY8C6xx5
psoc6_256k.cfg	infineon/cy8cxx4.cfg	psoc6	CY8C6xx4
psoc6_secure.cfg	infineon/cyxx64x7.cfg	psoc6	CYB06447, CYB06447-BL
psoc6_2m_secure.cfg	infineon/cyxx64xa.cfg	psoc6	CYS0644A, CYB0644A
psoc6_512k_secure.cfg	infineon/cyxx64x5.cfg	psoc6	CYB06445
psoc4.cfg	infineon/psoc4.cfg	psoc4	All PSOC™ 4 MCU, CCGx targets except PSOC™ 4500H MCU
psoc4500.cfg	infineon/psoc4500h.cfg	psoc4	PSOC™ 4500H MCU
pag2s.cfg	infineon/pag2s.cfg	psoc4	PAG2S MCU
psoc4hv_a0.cfg	infineon/psoc4hv_a0.cfg	psoc4	PSOC™ 4-HV devices
cyw208xx.cfg	infineon/cyw20829.cfg	cyw20829	AIROC™ CYW208xx Wi-Fi & Bluetooth® combo chips
cyw55500.cfg	infineon/cyw55500.cfg	cyw55500	AIROC™ CYW55500 devices
N/A	bcm4390x.cfg [2)]	N/A	AIROC™ CYW4390x family of devices
cat1c.cfg	infineon/xmc7xxx.cfg	cat1c	XMC7100/XMC7200 series of devices

3 Supported target configurations

Old target config	Target config	Chip name	Supported devices
cat1c.cfg	infineon/cytxbb.cfg	cat1c	CYT3BB/CYT4BB series of TRAVEO™ T2G Body High MCU devices
cat1c.cfg	infineon/cyt4bf.cfg	cat1c	CYT4BF series of TRAVEO™ T2G Body High MCU devices
cat1c.cfg	infineon/cyt4dn.cfg	traveo2	CYT4DN series of TRAVEO™ T2G Cluster 6M MCU devices
N/A	infineon/cytedn.cfg	traveo2	CYT4EN series of TRAVEOTM T2G Cluster 6M MCU DDR devices
traveo2_1m_a0.cfg	infineon/cyt2b7.cfg	traveo2	CYT2B7 series of TRAVEO™ T2G Body Entry MCU devices
traveo2_2m.cfg	infineon/cyt2b9.cfg	traveo2	CYT2B9 series of TRAVEO™ T2G Body Entry MCU devices
traveo2_be_4m.cfg	infineon/cyt2bl.cfg	traveo2	CYT2BL series of TRAVEO™ T2G Body Entry MCU devices
traveo2_512k_a0.cfg	infineon/cyt2b6.cfg	traveo2	CYT2B6 series of TRAVEO™ T2G Body Entry MCU devices
N/A	infineon/cyt3dl.cfg	traveo2	CYT3DL series of TRAVEO™ T2G Cluster 4M MCU devices
N/A	infineon/cyt2dl.cfg	traveo2	CYT2CL series of TRAVEO™ Cluster Entry 4M MCU devices
N/A	infineon/cyt6bx.cfg	traveo2	CYT6BJ series of TRAVEO™ T2G 16M MCU devices
fx3g2.cfg	infineon/fx3gx.cfg	fx3	FX3G2 and FX2G3 series of devices
macaw.cfg	infineon/macaw.cfg	macaw	Macaw series of devices
N/A	infineon/xmc5100.cfg	xmc5xxx	XMC5100 devices
N/A	infineon/xmc5200.cfg	xmc5xxx	XMC5200 devices
N/A	infineon/xmc5300.cfg	xmc5xxx	XMC5300 devices

[2] The CYW9WCD1EVAL1 kit is equipped with an onboard FTDI-based JTAG adapter. OpenOCD provides board-level configuration file for this kit which will configure the JTAG adapter and the CYW4390x chip automatically. Use single *board/cyw9wcd1eval1.cfg* configuration file for the CYW9WCD1EVAL1 board (instead of separate files for the probe and chip).

4 Command-line options

OpenOCD command-line options can be combined in a single command-line.

The most important options and commands include:

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

4.1 --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/infineon/
pse84xgxs2.cfg
```

The output should appear similar to the following:

```
transport: jtag
adapter name: jlink
adapter speed: 2000 kHz
adapter srst delay: 50
Info : Using CMSIS-flash algorithms 'PSE84_RRAM_NVM' for bank 'cat1d.cm33.main_ns' (footprint 5752 bytes)
Info : CMSIS-flash: ELF path: ../flm/infineon/pse8xxgp/PSE84_RRAM_NVM.FLM
Info : CMSIS-flash: Address range:     0x22000000-0x2207FFFF
Info : CMSIS-flash: Program page size: 0x00000400 bytes
Info : CMSIS-flash: Erase sector size: 0x00000400 bytes, unified
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : J-Link Ultra V5 compiled Apr 15 2024 17:37:33
Info : Hardware version: 5.00
Info : VTarget = 1.764 V
Info : clock speed 2000 kHz
Info : JTAG tap: cat1d.cpu tap/device found: 0x2ed80107 (mfg: 0x083 (ILC Data Device), part: 0xed80, ver: 0x2)
** Target is not in Test Mode
************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Cortex-M33 r1p0 processor detected
Info : [cat1d.cm33] target has 8 breakpoints, 4 watchpoints
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
```

The "tap/service found" message should appear with no warnings, which means the JTAG communication is working.

4.2 --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 file first found with a matching file name is used.

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

Example (J-Link & PSOC™ Edge MCU target):

```
openocd -s ../scripts -f interface/jlink.cfg -f target/infineon/pse84xgxs2.cfg
```

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

4.3 --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 (J-Link & PSOC™ Edge MCU target):

```
openocd -s ../scripts -f interface/jlink.cfg -f target/infineon/pse84xgxs2.cfg -c "targets;
shutdown"
```

4.4 --debug (-d)

Specifies the debug level. The debug level is 2 by default.

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

This affects the kind of messages sent to the server log:

- Level 0: Error messages only
- Level 1: Level 0 messages + warnings
- Level 2: Level 1 messages + informational messages
- Level 3: Level 2 messages + debugging messages

Example:

```
openocd -d1
```

4.5 --log_output (-l)

Redirects the log output to the file *logfile.txt*.

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

Example (KitProg3 & PSOC™ Edge MCU target):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -l log.txt -c
"targets; shutdown"
```

4.6 --help (-h)

Displays the help message.

```
openocd -h
```

4.7 --version (-v)

Displays the OpenOCD version.

```
openocd -v
```

5 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 a configuration file.

The command can be invoked with the `-c` command option.

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

5 OpenOCD commands overview

Command	Description
flash padded_value	Sets the default value used for padding-any-image sections
flash rmw	Can be used to modify flash individual bytes
add_verify_range	Allows specifying the memory regions to be compared during the verify operation
show_verify_ranges	Displays all active verify ranges for all targets that were added using the add_verify_range command. This command does not take any arguments
clear_verify_ranges	Deletes all verify ranges for the specified target that were added using the add_verify_range command
verify_image	Verifies a file against the target memory starting at a given address
verify_image_checksum	Verifies a file against the target memory starting at a given address
load_image	Loads an image from a file to the target memory offset from its load address
dump_image	Dumps bytes of the target memory to a binary file
kitprog3 acquire_config	Controls device acquisition parameters, and optionally enables acquisition during the early initialization phase
kitprog3 acquire_psoc	Performs device acquisition
kitprog3 power_config	Controls KitProg3/MiniProg4 internal power supply parameters and optionally enables power
kitprog3 led_control	Controls KitProg3/MiniProg4 LEDs
kitprog3 get_power	Reports the target voltage in millivolts
kitprog3 bridge_control	Configures the KitProg3 I2C/SPI-USB and UART-USB bridging off/on
psoc6/cat1c/traveo2/xmc5xxx sflash_restrictions	Enables or disables writes to dedicated SFlash regions
psoc6/cat1c/traveo2/xmc5xxx allow_efuse_program	Allows or disallows writes to the EFuse region
psoc6/cat1c/traveo2/xmc5xxx reset_halt	Simulates a broken vector catch on MCU
psoc4/cat1c/traveo2/xmc5xxx ecc_error_reporting	Enables or disables the ECC error reporting
cat1c/traveo2/xmc5xxx wflash blank_map [first_sector [last_sector | 'last']]	Displays per-word validity map of the given sectors of Fork Flash
cat1c/traveo2/xmc5xxx wflash write_image <filename> [offset]	Programs individual 32-bit words from given file to the Work Flash
cat1c/traveo2/xmc5xxx wflash write_words <address> <word_1> [word_2] ... [word_N]	Modifies individual 32-bit words in Work Flash
psoc6 secure_acquire	Performs acquisition of PSOC™ 64 "Secure Boot" MCUs
psoc4 reset_halt	Performs the alternate acquire sequence

5 OpenOCD commands overview

Command	Description
psoc4 mass_erase	Performs a mass erase operation on the given flash bank
psoc4 chip_protect	Changes chip protection mode to PROTECTED
psoc4 kill_mode	Changes chip protection mode to KILL
source	Reads a file and executes it as a Tcl script
find	Finds and returns the full path to a file with the Tcl script
set	Creates a Tcl variable
add_script_search_dir	Adds a directory to the file/script search path
sleep	Waits for a given number of milliseconds before resuming

6 OpenOCD commands description

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

Most of the examples described in this section can be executed against different PSOC™ Edge/6/4 MCU targets. See Supported target configurations for the detailed list of available target devices and corresponding OpenOCD configuration files.

- General OpenOCD commands
- KitProg3/MiniProg4 driver commands
- Flash driver commands
- cmsis_flash flash driver commands
- Other commands
- CYW4390x commands
- AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip commands

6.1 General OpenOCD commands

6.1.1 version

Displays a string identifying the OpenOCD version.

Example:

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

6.1.2 help

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

```
help [string]
```

Example:

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

6.1.3 shutdown

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

```
shutdown [error]
```

Example:

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

6.1.4 log_output

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

```
log_output [filename]
```

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"log_output d:/log.txt; targets; shutdown"
```

6.1.5 debug_level

Displays the debug level. If *n* (from 0..3) is provided, set it to that level.

This affects the kind of messages sent to the server log:

- Level 0: Error messages only
- Level 1: Level 0 messages + warnings
- Level 2: Level 1 messages + informational messages
- Level 3: Level 2 messages + 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 (KitProg3 & PSOC™ Edge MCU):

This example sets the debug level to L3 at the start of the target initialization, then switches to L1 for programming operation.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"debug_level 3; init; reset init; debug_level; program c:/firmware/firmware.hex verify;
shutdown"
```

6.1.6 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`).

The `signals` option 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 (default)
- trst_only
- srst_only
- trst_and_srst

See the OpenOCD documentation for details.

Example (KitProg3 & PSOC™ Edge MCU):

This example configures the reset configuration to trst_and_srst for the target MCU.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c
"reset_config trst_and_srst; targets; shutdown"
```

6.1.7 adapter speed

Sets a non-zero speed in kHz for the debug adapter. Therefore, to specify 3 MHz, provide `3000`.

```
adapter speed <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 lower than that peak rate.

For example, most Arm® cores accept up to 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:

This example configures SWD transport and sets the clock for J-Link to 2 MHz.

```
openocd -s ../scripts -c "transport select swd” -f interface/jlink.cfg -f target/infineon/
pse84xgxs2.cfg -c "init; reset init; adapter speed 2000; program c:/firmware/firmware.hex
verify; shutdown"
```

6.1.8 adapter serial

Specifies the serial number of the adapter device to use, and can be used to specify which device to use if multiple devices are connected to the host PC. If not specified, serial numbers are not considered.

```
adapter serial <serial_number>
```

Example:

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "adapter serial 1A1509F301237400; shutdown"
```

6.1.9 transport list

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

Example:

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

6.1.10 transport select

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

```
transport select <transport_name>
```

When invoked with the `transport_name` option, OpenOCD 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"
```

6.1.11 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 relevant only to boards with more than one target.

```
targets [name]
```

Example (KitProg3 & PSOC™ Edge MCU):

Displays all available targets of the connected PSOC™ Edge MCU:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "targets;
shutdown"
```

Selects the SYS AP of the PSOC™ Edge MCU as the current target:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "targets
cat1d.sys; target current"
```

6.1.12 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 (J-Link & PSOC™ Edge MCU):

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

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Cortex-M33 r1p0 processor detected
Info : [cat1d.cm33] target has 8 breakpoints, 4 watchpoints
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
   TapName             Enabled  IdCode     Expected   IrLen IrCap IrMask
-- ------------------- -------- ---------- ---------- ----- ----- ------
 0 cat1d.cpu              Y     0x2ed80107 0x00000000     8 0x01  0x0f
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.13 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, it displays that many units; otherwise, only one item is displayed.

Example KitProg3 + PSOC™ Edge MCU):

Displays two 32-bit words of memory of the PSOC™ Edge MCU.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; mdw 0x34008100 2; shutdown"
```

```
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
0x34008100: abcd1234 abcd1234
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.14 mw(w)(h)(b)

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

```
mww [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.

Example (KitProg3 + PSOC™ Edge MCU target):

Write a 32-bit word to the memory of the PSOC™ Edge MCU.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; mww 0x34008110 0xAAAABBBB; mdw 0x34008110; shutdown"
```

```
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
0x34008110: aaaabbbb
shutdown command invoked
```

6.1.15 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 and programming flash. 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 (KitProg3 + PSOC™ 6 MCU target):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
shutdown"
```

6.1.16 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; shutdown"
```

6.1.17 program

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

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

The only required parameter is `filename`; others are optional.

- `preverify` – Performs the verification step before flash programming. Programming will be skipped if the flash contents match the data file.
- `verify` – Compares the contents of the data file filename with the contents of the flash after flash programming
- `reset` – "reset run" is called if this parameter is given (see reset [run] [halt] [init] for details)
- `exit` – OpenOCD is shut down if this parameter is given.
- `offset` – If relocation offset is specified, it is added to the base address for each section in the image.

Example (KitProg3 & PSOC™ Edge MCU):

This example specifies QSPI configuration and and connects to the KitProg3 probe with the PSOC™ Edge MCU target, programs flash with the *app_combined.hex* file, verifies programmed data, and finally shuts down the OpenOCD programmer.

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "set SMIF_BANKS { 0 {addr 0x60000000 size
0x01000000 psize 0x00000200 esize 0x00040000} }" -f target/infineon/pse84xgxs2.cfg -c "program
c:/firmware/app_combined.hex verify exit"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
** Programming Started **
auto erase enabled
Info : Flash write discontinued at 0x60347890, next section at 0x60580400
Info : Section start address 0x60340400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60340000
Info : Padding image section 0 at 0x60347890 with 1904 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60348000 .. 0x6037ffff
[100%] [################################] [ Erasing     ]
[100%] [################################] [ Programming ]
Info : Section start address 0x60580400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60580000
Info : Padding image section 1 at 0x6058288c with 1908 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60583000 .. 0x605bffff
[100%] [################################] [ Erasing     ]
[100%] [################################] [ Programming ]
Info : Padding image section 2 at 0x701077d0 with 2096 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x70108000 .. 0x7013ffff
[100%] [################################] [ Erasing     ]
[100%] [################################] [ Programming ]
wrote 77824 bytes from file c:/firmware/app_combined.hex in 5.665688s (13.414 KiB/s)
** Programming Finished **
** Verify Started **
verified 69868 bytes in 1.619264s (42.137 KiB/s)
** Verified OK **
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.18 erase_all

This command erases the flash memory. Some SFlash rows could be skipped due to restrictions, see psoc6/cat1c/traveo2/xmc5xxx sflash_restrictions command.

Example (KitProg3 & PSOC™ 6 MCU target):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg  -c "init;
reset init; erase_all; exit"
```

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
Erasing main flash bank...
[100%] [################################] [ Erasing     ]
erased sectors 0 through 355 on flash bank 0 in 6.059051s
Info : cat1d.dap: powering down debug domain...
```

6.1.19 flash banks

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

Example (KitProg3 & PSOC™ 6 MCU):

```
./openocd -s ../scripts -f interface/kitprog3.cfg -c "set SMIF_BANKS { 0 {addr 0x60000000 size
0x01000000 psize 0x00000200 esize 0x00040000} }" -f target/infineon/pse84xgxs2.cfg -c "init;
flash probe 0; flash probe 1; flash probe 2; flash probe 3; flash banks; shutdown"
```

```
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
flash 'cmsis_flash' found at 0x22011000
flash 'virtual' found at 0x32011000
flash 'cmsis_flash' found at 0x60000000
flash 'virtual' found at 0x70000000
#0 : cat1d.cm33.main_ns (cmsis_flash) at 0x22011000, size 0x00059000, buswidth 4, chipwidth 4
#1 : cat1d.cm33.main_s (virtual) at 0x32011000, size 0x00059000, buswidth 4, chipwidth 4
#2 : cat1d.cm33.smif0_ns (cmsis_flash) at 0x60000000, size 0x01000000, buswidth 4, chipwidth 4
#3 : cat1d.cm33.smif0_s (virtual) at 0x70000000, size 0x01000000, buswidth 4, chipwidth 4
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.20 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash info 0; shutdown"
```

Prints the information about flash bank 0:

```
        #351: 0x00057c00 (0x400 1kB) not protected
        #352: 0x00058000 (0x400 1kB) not protected
        #353: 0x00058400 (0x400 1kB) not protected
        #354: 0x00058800 (0x400 1kB) not protected
        #355: 0x00058c00 (0x400 1kB) not protected
CMSIS Flash Device: PSE84_RRAM_NVM
 Address range: 0x22000000 - 0x2207ffff
 Program page size: 1024 bytes
 Sectors:
   512 x 1024    bytes @ 0x22000000

shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.21 flash protect

Enables (*on*) or disables (*off*) protection of flash blocks in flash bank `num`, starting at protection block `first` and continuing up to and including `last`.

***Note**: This command is applicable for PSOC™ 4 MCU only.*

```
flash protect num first last (on|off)
```

Providing a `last` block of last specifies "to the end of the flash bank". The `num` parameter is a value shown by flash banks. The protection block is usually identical to a flash sector. Some devices may utilize a protection block distinct from the flash sector. See `flash info` for a list of protection blocks.

Example (KitProg3 & PSOC™ 4 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/psoc4.cfg -c "init; reset
init; flash protect 0 0 last on; shutdown"
```

Protect all sectors from being written in flash bank 0:

```
**********************************************
** Silicon: 0x257C, Family: 0xB5, Rev.: 0x11 (A0)
** Detected Device: CY8C4147AZI-S475
** Detected Family: PSoC 4100S Plus
** Detected Main Flash size, kb: 128
** Chip Protection: protection OPEN
**********************************************
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring PSoC device...
target halted due to debug-request, current mode: Thread
xPSR: 0xa1000000 pc: 0x10000040 msp: 0x20003fe8
** Device acquired successfully
Info : ignoring flash probed value, using configured bank size
set protection for sectors 0 through 511 on flash bank 0
shutdown command invoked
```

6.1.22 flash erase_sector

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

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

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.

***Note**: On PSOC™ 4, PSOC™ 4 HVPA, and PSOC™ 4 HVMS MCU devices, per-sector erase operation is not supported, only mass-erase is available. This command is ignored on the device unless a full erase of the flash is requested (flash erase_sector 0 0 last).*

Example (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash erase_sector 0 0 last; shutdown"
```

Erases all sectors in flash bank 0:

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [################################] [ Erasing     ]
erased sectors 0 through 355 on flash bank 0 in 4.319694s
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.23 flash erase_address

Erases the 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 the 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, the flash is unprotected before erase starts.

***Note**: On PSOC™ 4, PSOC™ 4 HVPA, and PSOC™ 4 HVMS MCU devices, per-sector erase operation is not supported, only mass-erase is available. This command is ignored on the device unless a full erase of the flash is requested (e.g., `flash erase_address 0 65536` for 64 KB parts).*

Example (KitProg3 & PSOC™ Edge MCU):

In this example, OpenOCD erases the 2-KB block starting at address 0x32011000:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash erase_address 0x32011000 2048; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [################################] [ Erasing     ]
erased address 0x32011000 (length 2048) in 0.119226s (16.775 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.24 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 (KitProg3 & PSOC™ Edge MCU):

Writes the binary file *firmware.bin* to flash bank 0 with starting at offset 0:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init; reset
init; flash write_bank 0 d:/firmware.bin 0x0; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [################################] [ Programming ]
wrote 8192 bytes from file c:/firmware/firmware.bin to flash bank 0 at offset 0x00000000 in 0.142290s (56.223 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.25 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. If a relocation `offset` is specified, 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), or `s19` (Motorola s19). The relevant flash sectors will be erased prior to programming if the erase parameter is given. If `unlock` is provided, the flash banks are unlocked before `erase` and `program`. The flash bank to use is inferred from the address of each image section.

Attention**: **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 reapply it after it has been removed by the `unlock` flag.

Example (KitProg3 & PSOC™ Edge MCU):

Writes the ELF image *firmware.hex* to the PSOC™ Edge MCU:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash write_image erase c:/firmware/firmware.hex; shutdown"
```

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
auto erase enabled
[100%] [################################] [ Erasing     ]
[100%] [################################] [ Programming ]
Warn : no flash bank found for address 0x2206a000
wrote 190464 bytes from file c:/firmware/firmware.hex in 3.639579s (51.105 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.26 flash fill(w)(h)(b)

Fills the 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 (KitProg3 & PSOC™ Edge MCU):

Fills the 32-KB block of memory starting at address 0x32011000 with the pattern 0x5A:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash fillw 0x32011000 0x5A5A5A5A 0x00002000; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [##################################] [ Programming ]
wrote 32768 bytes to 0x32011000 in 0.596181s (53.675 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.27 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 (KitProg3 & PSOC™ Edge MCU):

Reads the 32-KB block of bank #0 memory and writes it to the binary file:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash read_bank 0 read_bank_0.bin 0x0 0x2000; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
wrote 8192 bytes to file ./read_bank_0.bin from flash bank 0 at offset 0x00000000 in 0.123977s (64.528 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.28 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 (KitProg3 & PSOC™ Edge MCU):

Verifies the content of bank #0.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash verify_bank 0 c:/firmware/firmware.bin 0x0; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
read 8192 bytes from file c:/firmware/firmware.bin and flash bank 0 at offset 0x00000000 in 0.123997s (64.518 KiB/s)
contents match
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.29 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, it defaults to 0xff.

Example (KitProg3 & PSOC™ Edge MCU):

Sets a padded value to 0xFF for bank #0.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash padded_value 0 0xFF; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
Default padded value set to 0xff for flash bank 0
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.30 flash rmw

The command is intended to modify flash individual bytes.

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

The command can be used to program the 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**: The `flash rmw` command is a custom command implemented in ModusToolbox™ OpenOCD CLI to extend its functionality.*

Example (KitProg3 & PSOC™ Edge MCU):

Modifies 8 bytes of the PSOC™ 6 MCU flash at address 0x10001234.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; flash rmw 0x32011234 DEADBEEFBAADC0DE; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
modified 8 byte(s) in 1024 byte region at 0x32011000 in 0.309674s (3.229 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.31 add_verify_range

The command allows specifying memory regions to be compared during the `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 the 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 ModusToolbox™ OpenOCD CLI to extend its functionality.*

6.1.32 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -s scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
show_verify_ranges; add_verify_range cat1d.cm33 0x22011000 0x1000; show_verify_ranges; exit"
```

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
cat1d.cm33  0x22011000 0x00001000
Info : cat1d.dap: powering down debug domain...
```

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

6.1.33 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>
```

***Note**: The clear_verify_ranges command is a custom command implemented in ModusToolbox™ OpenOCD CLI to extend its functionality.*

6.1.34 verify_image

Verifies `filename` against the 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 (KitProg3 & PSOC™ Edge MCU):

Verifies a *firmware.hex* image against the target memory.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; load_image c:/firmware/firmware.hex; verify_image c:/firmware/firmware.hex;
shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
1040384 bytes written at address 0x24002000
downloaded 1040384 bytes in 5.129456s (198.072 KiB/s)
verified 1040384 bytes in 5.055619s (200.965 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.35 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 (KitProg3 & PSOC™ Edge MCU):

Verifies a *firmware.hex* image against the target memory of the PSOC™ Edge MCU using the CRC checksum only.

```
openocd -s ../scripts -f interface/jlink.cfg -c "transport select swd" -f target/infineon/
pse84xgxs2.cfg -c "init; reset init; verify_image_checksum d:/firmware.hex 0x0; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
1040384 bytes written at address 0x24002000
downloaded 1040384 bytes in 5.118777s (198.485 KiB/s)
verified 1040384 bytes in 5.115273s (198.621 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.36 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 the 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 the binary file firmware.bin to the RAM of the PSOC™ Edge MCU.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
reset init; load_image c:/firmware/firmware.bin 0x22110000; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
364544 bytes written at address 0x22110000
downloaded 364544 bytes in 1.699955s (209.417 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.1.37 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 8 KB of the PSOC™ Edge MCU memory to the file *dump_mem.bin*.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init; reset
init; dump_image d:/dump_mem.bin 0x10001234 0x2000; shutdown"
```

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
dumped 8192 bytes in 0.038664s (206.911 KiB/s)
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.2 KitProg3/MiniProg4 driver commands

The KitProg3/MiniProg4 probe implements the CMSIS-DAP protocol defined by Arm® with some extensions. Consequently, the KitProg3/MiniProg4 driver in OpenOCD is a wrapper around the native CMSIS-DAP driver that extends its functionality with the KitProg3-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 KitProg3 driver exposes several custom Tcl configuration commands. All commands in this section must be prefixed with the name of the driver – "kitprog3".

6.2.1 kitprog3 acquire_config

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

```
acquire_config <status> [target_type] [mode] [attempts] [timeout] [ap]
```

- `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 – PSOC 4
  - 1 – PSOC 5
  - 2 – PSOC 6
  - 3 – T2G, XMC7xxx and XMC5xxx
  - 4 – AIROC
- `mode` – Specifies the acquisition mode. This parameter is mandatory only if `status=on`. The possible values: 0 – Reset, 1 – Power Cycle. The mode affects only the first step (how to reset the part at the start of the acquisition flow).
  - Reset mode: To start programming, the host toggles the XRES line and then sends SWD/JTAG commands
  - Power Cycle mode: To start programming, the KitProg3-based probe powers on the MCU and then starts sending the SWD/JTAG commands. The XRES line is not used. Power Cycle mode support is optional and should be used only if the XRES pin is not available on the part's package.

    ***Note**: Before using Power Cycle acquisition, make sure that the target is not powered externally!*
- `attempts` – The number of attempts to acquire the target device. This parameter is ignored when `status=on`.
- `timeout` – (Optional) Timeout value in seconds. The maximum value for the timeout is 30 seconds.
- `ap` – Access port to use for the acquisition. The value of this parameter should be in range 0…255. This parameter is mandatory if the `timeout` parameter is specified.

Example (KitProg3 & PSOC™ 6 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "kitprog3
acquire_config on 2 0; init; reset init; shutdown"
```

6.2.2 kitprog3 acquire_psoc

Performs device acquisition. Can be called at any time. The acquisition procedure must be configured using `acquire_config` before calling this command.

Example (KitProg3 & PSOC™ 6 MCU)::

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "kitprog3
acquire_config on 2 0; init; kitprog3 acquire_psoc; reset init; shutdown"
```

6.2.3 kitprog3 power_config

Controls the KitProg3-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` –Mandatory; enables or disables power supply during the initialization phase. Possible values: on|off.
- `voltage` – The power supply voltage in millivolts. This parameter is optional. Either default (2.5 volts) or kit-specific voltage will be applied if this parameter is not specified.

Example (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "kitprog3
power_config on 3300; init; shutdown"
```

```
Open On-Chip Debugger 0.12.0+dev-5.1.0.3042 (2024-04-17-08:26)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Info : auto-selecting first available session transport "swd". To override use 'transport select <transport>'.
adapter speed: 2000 kHz
adapter srst delay: 25
adapter srst pulse_width: 25
** Auto-acquire enabled, use "set ENABLE_ACQUIRE 0" to disable
cortex_m reset_config sysresetreq
cortex_m reset_config sysresetreq
Info : Using CMSIS-DAPv2 interface with VID:PID=0x04b4:0xf151, serial=0711062303210400
Info : CMSIS-DAP: SWD supported
Info : CMSIS-DAP: JTAG supported
Info : CMSIS-DAP: Atomic commands supported
Info : CMSIS-DAP: FW Version = 2.0.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 1 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : KitProg3: FW version: 2.60.1443
Info : KitProg3: Pipelined transfers enabled
Info : KitProg3: Asynchronous USB transfers enabled
Info : kitprog3: powering up target device using KitProg3 (VTarg = 3300 mV)
Info : VTarget = 3.292 V
```

6.2.4 kitprog3 led_control

Controls the KitProg3 LEDs. Can be called only after the initialization phase.

```
kitprog3 led_control <type>
```

- `type` – Mandatory; 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 -f target/infineon/pse84xgxs2.cfg -c "init;
kitprog3 led_control 2"
```

6.2.5 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 -f target/infineon/pse84xgxs2.cfg -c "init;
kitprog3 get_power; shutdown"
```

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
VTarget = 1.795 V
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.2.6 kitprog3 bridge_control

Configures the KitProg3 I2C/SPI-USB and UART-USB bridging off/on. Can be called only after the initialization phase. This command is used by default to optimize transfer performance in specific operations write_image/ load_image/verify_image/etc. via exclusion of KP3 bridging request processing. It s not to be used during debug/attach/other operation(s) that might require KP3 I2C/SPI-USB and UART-USB to be available and fully-functional.

```
kitprog3 bridge_control <state>
```

- state - Mandatory; sets bridge state. The possible values are:
  - 0 = ON
  - 1 = OFF

Example:

```
./openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/pse84xgxs2.cfg -c "init;
kitprog3 bridge_control 1; mww 0x34000000 0xFFFFFFFF 0x10000; mdw 0x34000000 0x10; kitprog3
bridge_control 0; shutdown"
```

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
0x34000000: ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
0x34000020: ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
shutdown command invoked
Info : cat1d.dap: powering down debug domain...
```

6.3 Flash driver commands

This section contains a list of custom commands exposed by the target MCU's Flash driver.

6.3.1 psoc6/cat1c/traveo2/xmc5xxx sflash_restrictions

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

```
psoc6 sflash_restrictions <mode>
cat1c sflash_restrictions <mode>
traveo2 sflash_restrictions <mode>
xmc5xxx sflash_restrictions <mode>
```

The command can be called at any time.

- `mode` – Mandatory; 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.

    Be aware that the NAR sub-region cannot be overwritten or erased if the new data is less restrictive than the existing data.

    Warning: ***Unintentional writing to this region may corrupt your device!***

  - Erase of USER/TOC/KEY and program of the whole SFlash region is allowed.

Writes to SFlash regions other than USER/TOC/KEY/NAR is possible only on the VIRGIN silicon, so the mode=3 option is mostly intended for internal use. It is useful for flash boot developers and validation teams. Note that erase (programming with 0x00 for PSOC™ 6, or 0xFF for XMC7xxx and TRAVEO™ T2G MCUs) is performed only for the USER, TOC2, and KEY regions; it is skipped for other SFlash regions regardless of this command.

Example (KitProg3 & PSOC™ 6 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init; reset
init; psoc6 sflash_restrictions 2; shutdown"
```

```
** SFlash SiliconID:    0xE2062200
** Flash Boot version: 0x021D8001
** Chip Protection: NORMAL
** psoc6.cpu.cm4: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x1600400c msp: 00000000
Warn : SFlash programming allowed for regions: USER, TOC, KEY, NAR
shutdown command invoked
```

6.3.2 psoc6/cat1c/traveo2/xmc5xxx 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. Be aware that EFuses are one-time programmable. Once an EFuse is blown, there is no way to revert its state. EFuse programming must be allowed for lifecycle transitions to work.

```
psoc6 allow_efuse_program <on|off>
cat1c allow_efuse_program <on|off>
traveo2 allow_efuse_program <on|off>
xmc5xxx allow_efuse_program <on|off>
```

Example (KitProg3 & PSOC™ 6 MCU):

Writes 1 bit to the EFuse region at address 0x907003FF of the PSOC™ 6 MCU:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init;
reset init; psoc6 allow_efuse_program on; flash fillb 0x907003FF 1 1; flash read_bank 3 d:/
dump_efuse.bin 0x3FF 0x1; shutdown"
```

```
Warn : Programming of efuses now ALLOWED
Info : MainFlash size overridden: 1024 kB
Info : Start address 0x907003ff breaks the required alignment of flash bank psoc6_efuse_cm0
Info : Padding 1023 bytes from 0x90700000
Info : The Life Cycle stage is not present in the programming file
wrote 1 bytes to 0x907003ff in 0.062402s (0.016 KiB/s)
wrote 1 bytes to file d:/dump_efuse.bin from flash bank 3 at offset 0x000003ff in 0.015601s (0.063 KiB/s
)
shutdown command invoked
```

6.3.3 psoc6/cat1c/traveo2/xmc5xxx reset_halt

The command simulates a broken vector catch on PSOC™ 6, XMC7xxx, TRAVEO™ T2G and XMC5xxx MCUs.

```
psoc6 reset_halt <mode>
cat1c reset_halt <mode>
traveo2 reset_halt <mode>
xmc5xxx 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` – (Optional) The type of reset to be performed. Possible values are `sysresetreq` and `vectreset`. If not specified, SYSRESETREQ is used for the CM0 core and VECTRESET is used for other cores in the system.

Example (KitProg3 & PSOC™ 6 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init; reset
init; psoc6 reset_halt vectreset; shutdown"
```

```
** SFlash SiliconID:   0xE2062200
** Flash Boot version: 0x021D8001
** Chip Protection: NORMAL
** psoc6.cpu.cm4: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x1600400c msp: 00000000
Info : psoc6.cpu.cm0: bkpt @0x100014B9, issuing VECTRESET
shutdown command invoked
```

6.3.4 psoc4/cat1c/traveo2/xmc5xxx ecc_error_reporting

Enables or disables the ECC error reporting during OpenOCD operations.

```
psoc4 ecc_error_reporting <on|off>
cat1c ecc_error_reporting <on|off>
traveo2 ecc_error_reporting <on|off>
xmc5xxx ecc_error_reporting <on|off>
```

OpenOCD supports the detection and reporting of ECC errors during the flash read operation. In the current implementation, OpenOCD 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, OpenOCD 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.

***Note**: In this section, psoc4 means only PSOC™ 4 HV devices.*

Example (KitProg3 & XMC7000 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cytxbb.cfg -c "init; reset
init; cat1c ecc_error_reporting on; shutdown"
```

6.3.5 cat1c/traveo2/xmc5xxx wflash blank_map [first_sector [last_sector | 'last']]

Displays per-word validity map of the given sectors of work flash. It accepts two optional parameters with the following rules:

- If no parameters are given, command displays validity map for all sectors
- If one parameter is given, parameter means sector number and the command displays validity map for given sector
- If two parameters are given, command displays validity map for range of sectors (param1 ... param2). The word 'last' can be used as a parameter #2 (same as erase_sector).

Example (KitProg3 & XMC7000 MCU):

```
> cat1c wflash blank_map 0

WorkFlash word validity map:
0x14000000 (#000): -+++-+-+------------------------------------------------------
0x14000100 (#000): ------------------------------------------------------------
0x14000200 (#000): ------------------------------------------------------------
0x14000300 (#000): ------------------------------------------------------------
0x14000400 (#000): ------------------------------------------------------------
0x14000500 (#000): ------------------------------------------------------------
0x14000600 (#000): ------------------------------------------------------------
0x14000700 (#000): ------------------------------------------------------------
```

6.3.6 cat1c/traveo2/xmc5xxx wflash write_image <filename> [offset]

Programs individual 32-bit words from given file to the work flash. All data in the file that does not belong to WFlash region is skipped. All unaligned data is trimmed to make the starting address and length of the data aligned on 32-bit boundaries. Appropriate warnings are displayed in this case. Command works with elf, hex, srec and bin files. Optional offset can be specified (same as flash write_image).

Example (KitProg3 & XMC7000 MCU):

The file contains unaligned data and also data which does not belong to the WFlash region.

```
> cat1c wflash write_image foo.hex

Warn : Section [0x13fff950, 0x13fff991) will be skipped
Warn : Section [0x13fffdac, 0x13fffe19) will be skipped
Warn : Section [0x13fffff0, 0x1400005d) will be truncated to [0x14000000, 0x1400005c)
Warn : Section [0x14000290, 0x14000299) will be truncated to [0x14000290, 0x14000298)
Warn : Section [0x140003b0, 0x140003e5) will be truncated to [0x140003b0, 0x140003e4)
[100%] [################################] [ Programming ]
```

6.3.7 cat1c/traveo2/xmc5xxx wflash write_words <address> <word_1> [word_2] ... [word_N]

Command is similar to 'flash rmw' except:

- Starting address must be aligned on 32-bit boundary
- Command works with 32-bit words which must be separated with a space

Example (KitProg3 & XMC7000 MCU):

```
> cat1c wflash write_words 0x14000004 0xDEADBEEF 0xBAADF00D 0xBAADC0DE
> cat1c wflash write_words 0x14000014 0x01234567
> cat1c wflash write_words 0x1400001C 0x89ABCDEF
> cat1c wflash blank_map 0

[100%] [################################] [ Programming ]
[100%] [################################] [ Programming ]
[100%] [################################] [ Programming ]
WorkFlash word validity map:
0x14000000 (#000): -+++-+-+------------------------------------------------
0x14000100 (#000): ------------------------------------------------------------
0x14000200 (#000): ------------------------------------------------------------
0x14000300 (#000): ------------------------------------------------------------
0x14000400 (#000): ------------------------------------------------------------
0x14000500 (#000): ------------------------------------------------------------
0x14000600 (#000): ------------------------------------------------------------
0x14000700 (#000): ------------------------------------------------------------
```

6.3.8 psoc6 secure_acquire

Performs acquisition of PSOC™ 64 "Secure Boot" MCUs.

```
psoc6 secure_acquire <magic_num_addr> <mode> <handshake> <timeout>
```

Parameters:

- `magic_num_addr` – Address in RAM to poll for the magic number. This address is different across different PSOC™ 6 MCU devices:
  - CYB06447, CYB06447-BL - 0x08044804
  - CYS0644A, CYB0644A - 0x080FE004
  - CYB06445 - 0x0803E004
- `mode` – Mode of acquisition. Possible values: `run`, `halt`.
  - In run mode, the command will perform reset and will wait for the "secure" application to open the corresponding access port.
  - In halt mode, a "secure" handshake will be performed right after reset to prepare the device for flash programming.
- `handshake` – Specifies whether full or short acquisition procedure should be executed. The short acquisition procedure simply waits until "secure" FW opens the given access port. This is intended for multi-core configuration when full acquisition has already been done with the other CPU core.
  - Possible values: `handshake` – full acquisition, `no_handshake` – short acquisition
- `timeout` – Timeout in milliseconds

6.3.9 psoc4 reset_halt

Performs the alternate acquire sequence as described in the PSOC™ 4 MCU programming specification.

```
psoc4 reset_halt
```

The command detects the location of the application entry points, sets a hardware breakpoint at that location, and issues a `SYSRESETREQ` reset.

Example (KitProg3 & PSOC™ 4 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/psoc4.cfg -c "init; reset
init; psoc4 reset_halt; shutdown"
```

6.3.10 psoc4 mass_erase

Performs mass erase operation on the given flash bank. The list of all flash banks can be obtained using `flash banks` command. This command is a shortcut and performs the same operation as the `flash erase_sector <bank_id> 0 last` command. The peculiarity of this command is that erasing of the mflash bank also erases the flashp bank. If the chip is in PROTECTED state, this command moves the protection state of the device from PROTECTED to OPEN and erases the entire flash device.

```
psoc4 mass_erase <bank_id>
```

Example (KitProg3 & PSOC™ 4 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/psoc4.cfg -c "init; reset init; psoc4
mass_erase 0; shutdown"
```

6.3.11 psoc4 chip_protect

Changes the chip protection mode to PROTECTED. This mode disables all debug access to the user code or memory. Access to most registers is still available; debug access to registers to reprogram flash is not available. Protection mode can be changed back to OPEN by performing the mass erase operation described above.

```
psoc4 chip_protect
```

Example (KitProg3 & PSOC™ 4 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/psoc4.cfg -c "init; reset
init; psoc4 chip_protect; shutdown"
```

6.3.12 psoc4 kill_mode

Moves target from OPEN to KILL state. KILL state is a state where debug pins are switched OFF and any access to target is prohibited. Both debugging and flashing new image is impossible. This mode is useful if you have completely finalized image and there is a need to make it impossible to read out image via debug pins. From the user perspective, target executes flashed application but is not accessible via openOCD or any other tool.

***Note**: This command is irreversible. So in order to prevent misuse, additionally, the variable `ALLOW_KILL_MODE` must be set to YES via openOCD arguments.*

Example (KitProg3 & PSOC™ 4 MCU):

```
openocd -s ../scripts -f interface/kitprog3.cfg -c "set ALLOW_KILL_MODE YES" -f target/infineon/
psoc4.cfg -c "init; reset init; psoc4 kill_mode; shutdown"
```

6.4 cmsis_flash flash driver commands

The "cmsis_flash" is a generic driver which uses the standard CMSIS flash loaders to program the flash. On Infineon devices, this driver is typically used for external flash programming but it also can be used for other purposes.

6.4.1 cmsis_flash init

Some types of flash banks are not mapped to the CPU address space right after reset. For example, the external flash connected to an SMIF peripheral requires special configuration of the MCU's hardware blocks in order to be mapped to the CPU address space. Usually, the flash loader's `Init()` function is responsible for enabling such mapping.

```
cmsis_flash init [bank_num]
```

This command loads the flash loader to the RAM and executes the `Init()` function. Beware that this function is intrusive. It requires that the MCU is acquired in "good-state" and all CPUs are halted (e.g., `reset init` is performed) before it can be called.

This command takes one optional argument – the number of the flash bank to be initialized. This command will initialize all cmsis_flash banks if no argument is specified.

6.4.2 cmsis_flash prefer_sector_erase

Controls driver strategy used during mass-erase of the flash bank. There are two possible strategies:

- Use the `EraseChip` API (if available)
- Use per-sector erase using the `EraseSector` API

The `EraseChip` method is used by default. This method is usually faster but it does not display the progress of the erase operation. The EraseChip API is optional; the driver will fall back to per-sector erase if the `EraseChip` API is not implemented in the flash loader. The other downside of this method is that depending on the flash loader implementation, it may erase all external memory banks, not only the bank specified in the `erase_sector` command.

Per-sector erase is usually slower but it displays the progress information and always erases the single flash bank specified in the `erase_sector` command.

```
cmsis_flash prefer_sector_erase [bank_num] <0/1|false/true>
```

Command takes two arguments:

- `bank_num` – (Optional) Flash bank number to enable/disable the per-sector erase strategy. This option will be applied to all cmsis_flash banks if this argument is omitted.
- `parameter_value` – Mandatory boolean value specifying whether per-sector strategy should be enabled or disabled.

6.5 Other commands

6.5.1 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -c "source [find interface/kitprog3.cfg]; source [find target/infineon/
pse84xgxs2.cfg]; targets; shutdown"
```

6.5.2 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -c "source [find interface/kitprog3.cfg]; source [find target/infineon/
pse84xgxs2.cfg]; targets; shutdown"
```

6.5.3 set

Stores a value to a named variable, first creating the variable if it does not already exist. Declaration variable must be done before usage.

```
set VARNAME value
```

Example (KitProg3 & PSOC™ Edge MCU):

```
openocd -s ../scripts -c "set ENABLE_CM0 0; source [find interface/kitprog3.cfg]; source [find
target/infineon/cy8c6xxx.cfg]; targets; shutdown"
```

6.5.4 sleep

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

```
sleep msec
```

Example:

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

6.5.5 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 (KitProg3 & PSOC™ Edge MCU):

```
openocd -c "add_script_search_dir ../scripts; source [find interface/kitprog3.cfg]; source
[find target/infineon/pse84xgxs2.cfg]; targets; shutdown"
```

6.6 CYW4390x commands

The CYW4390x chip supports only limited subset of flash-related commands. Currently, the following flash programming commands are supported:

6.6.1 program

Programs a given binary programming into the flash of the target device.

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

The only required parameter is filename; the others are optional.

- `reset` – Calls `reset run` (see reset [run] [halt] [init] for details)
- `exit` – Shuts down OpenOCD

Limitations compared to standard `program` command:

- Only binary files are supported. Files in any format other than binary will be programmed as a binary data. For example, if a HEX file is programmed, the flash will contain textual representation of the file.
- The `preverify`, `verify`, and `offset` parameters are not supported. Verification is done automatically during programming.

Example:

The next example connects ModusToolbox™ OpenOCD CLI to the CYW9WCD1EVAL1 board with the CYW4390x MCU target, programs the flash with the *firmware.bin* file, resets the target, and finally shuts down the OpenOCD programmer.

```
openocd -s ../scripts -f board/cyw9wcd1eval1.cfg -c "program d:/firmware.bin reset exit"
```

***Note**: For erasing CYW4390x chip use the `erase_all` command.*

6.7 AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip commands

6.7.1 provision_no_secure

Performs a transition from NORMAL to NORMAL_NO_SECURE lifecycle. CYW20829 devices come from a factory in the NORMAL lifecycle stage. In this stage, the boot code will not launch the programmed application after reset. The lifecycle must be changed to either SECURE or NORMAL_NO_SECURE stage to use the device. Normally, this task is performed using cysecuretools; this command is just a shortcut which simplifies the process. Transition to the SECURE lifecycle is not supported by this command and must be performed using cysecuretools.

***Note**: This command must be executed before the init command.*

```
provision_no_secure <service_app> <app_params> [service_app_addr] [params_addr]
```

Required parameters:

- `service_app` – File name of the binary service application image, with path
- `app_params` – File name of the service application's parameters image, with path

Optional parameters:

`service_app_addr` – Address in the RAM where the service application will be loaded; 0x20004000 by default.

`params_addr` - Address in the RAM where the service application's parameters will be loaded; 0x2000D000 by default.

Example:

The next example connects ModusToolbox™ OpenOCD CLI to the CYW9WCD1EVAL1 board with the CYW4390x MCU target, programs the flash with the *firmware.bin* file, resets the target, and finally shuts down the OpenOCD programmer.

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cyw20829.cfg -c
"provision_no_secure D:/service_app.bin D:/app_params.bin; exit"
```

7 Global variables

The global variables listed in this section control the behavior of a target configuration file (e.g., psoc6.cfg). They are set in the command-line before any configuration file such as kitprog3.cfg or psoc6.cfg. See the command set for details.

- PSOC™ 6 MCU global variables
- PSOC™ 4 MCU global variables
- AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip global variables
- XMC7xxx/XMC5xxx and TRAVEO™ T2G global variables
- PSOC™ Control C3 global variables
- PSOC™ Edge E84 global variables

7.1 PSOC™ Edge E84 global variables

7.1.1 ENABLE_ACQUIRE

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

Possible values:

- 1 – Test mode acquisition enabled (default, recommended)
- 0 – Test mode acquisition disabled (used for attaching to the running target, not recommended for programming and debugging)

***Note**: For PSOC™ Edge E84, Test mode acquisition is implemented for MiniProg4, KitProg3, J-Link, and cmsis-dap probes via SWD and JTAG protocols. For the slow host PC, however, or when using USB virtualization, and for the devices with small Listen window duration, it is recommended to use MiniProg4/KitProg3 and SWD.*

7.1.2 ENABLE_CM33, ENABLE_CM55

Specifies the CPU cores' visibility to OpenOCD. Useful for single or multi-core applications debugging.

Possible values:

- 1 – Core is enabled.
- 0 – Core is disabled.

Only CM33 core is enabled by default, allowing any application programming and allowing debugging of CM33-based applications. CM55 core is disabled after reset, and enabling it in OpenOCD is recommended only for CM55 or Multi-Core applications debugging. When enabling CM55, the CM33 core must also be enabled.

7.1.3 DEBUG_CERTIFICATE

Specifies the location of the debug certificate (token) binary file to re-enable the CPU Access Port(s) for programming and debugging operations if the port(s) was closed per applied policies.

***Note**: Debug certificate is of no use if the policy for Access Port is 'permanent disable'.*

7.1.4 DEBUG_CERTIFICATE_ADDR

(Optional) Specifies the SRAM location where the debug certificate (token) will be loaded. The default address is 0x34000000.

7.1.5 SMIF_BANKS

Defines QSPI memory banks. This variable is a two-dimensional associative Tcl array of the following format:

```
set SMIF_BANKS {
  1 {addr <XIPaddr1> size <BankSz1>}
  2 {addr <XIPaddr2> size <BankSz2>}
  ...
  N {addr <XIPaddrN> size <BankSzN>}
}
```

Where:

- `XIPaddrN` – XIP mapping address
- `BankSzN` – Total size of this flash bank, in bytes

7.1.6 RRAM_MAIN_OFFSET

Defines the offset for the users' space in the RRAM_MAIN_NVM region, to exclude the system data and proprietary boot assets from programming operations. The default value is 0x11000.

Users may free up to 28 KB at the start of the programmable RRAM region by replacing the extended boot image and setting the appropriate offset in the OpenOCD command line.

Example:

```
openocd -s scripts -f interface/kitprog3.cfg -c "set RRAM_MAIN_OFFSET 0xA000" -f target/
infineon/pse84xgxs2.cfg -c "init; reset init; flash banks; shutdown"
```

```
#0 : cat1d.cm33.main_ns (cmsis_flash) at 0x2200a000, size 0x00060000, buswidth 4, chipwidth 4
#1 : cat1d.cm33.main_s (virtual) at 0x3200a000, size 0x00060000, buswidth 4, chipwidth 4
```

```
openocd -s scripts -f interface/kitprog3.cfg -c "set RRAM_MAIN_OFFSET 0xA000" -f target/
infineon/pse84xgxs4.cfg -c "init; reset init; flash banks; shutdown"
```

Refer to "PSOC™ Edge MCU Programming Specifications" for the NVM memory definitions. Refer to "Getting started with PSOC™ Edge security" application note for the details of replacing the extended boot image.

```
#0 : cat1d.cm33.main_ns (cmsis_flash) at 0x2200a000, size 0x00036000, buswidth 4, chipwidth 4
#1 : cat1d.cm33.main_s (virtual) at 0x3200a000, size 0x00036000, buswidth 4, chipwidth 4
```

7.2 PSOC™ Control C3 global variables

7.2.1 ENABLE_ACQUIRE

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

***Note**: Test mode acquisition is an Infineon method for acquiring or 'unbricking' a device in case of critical user application misbehavior; for example, repurposing debug pins or disablement critical clocks. Test mode is a sequence of reset and requesting bootloader for not launching user application.*

***Note**: For PSOC™ Control C3, Test mode acquisition works for mp4/kp3/jlink and swd/jtag, but in the most harsh cases described it is recommended to use mp4/kp3 + swd.*

***Note**: The main goal of test mode acquisition is to prevent launching the user application as a part of target acquisition. Also, the option must be set to enable WFA acquisition (procedure of unlocking AP by means of debug certificate).*

Possible values:

- 1 – Test mode acquisition enabled (default).
- 0 – Test mode acquisition disabled.

7.2.2 DEBUG_CERTIFICATE_RQST

Specifies the type of the debug request token. To be used for secure debugging of the memory regions protected by protected firmware or OEM application policies.

Possible values:

- OEM – OEM application (default).
- PROT_FW – for protected firmware.

7.2.3 DEBUG_CERTIFICATE

Specifies location of the debug certificate binary file. This file is needed to configure a secure debug session in cases when CM33 access port is closed due to policies applies. OpenOCD checks status of the CM33 access port during initialization phase and after each reset. If it's closed, it attempts to reopen the CM33 AP by sending a debug certificate to target and issuing WFA request. This variable should contain the full path to the debug certificate binary file.

***Note**: Debug certificate is of no use if policy for CM33 AP is 'permanent disable'.*

7.2.4 ENABLE_CM33

Allows specifying the CPU cores to be visible to OpenOCD. Useful for cases when only SYS-AP needed. OpenOCD never affects disabled cores.

***Note**: Flash operations are done only with enabled CM33 core.*

Possible values:

- 1 – CM33 core is enabled (default).
- 0 – Core is disabled.

7.3 PSOC™ 6 MCU global variables

7.3.1 ENABLE_ACQUIRE

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

Possible values:

- 1 – Reset acquisition enabled (default with KitProg3/MiniProg4)

- 2 – Power Cycle acquisition enabled. The voltage level can be controlled by using ENABLE_POWER_SUPPLY.
- 0 – Acquisition disabled (default for other debug adapters)

7.3.2 ENABLE_POWER_SUPPLY

Controls the internal power supply of KitProg3/MiniProg4 adapters. If this command is specified, the KitProg3 driver enables the power supply, thus powering on the target during initialization.

Possible values:

- 0 – Power supply disabled
- Any other value defines target voltage in millivolts.
- default – Sets the last used voltage before KitProg3/MiniProg4 was powered off.

7.3.3 ENABLE_CM0, ENABLE_CM4

Allows specifying the CPU cores to be visible to OpenOCD. OpenOCD never affects disabled cores.

Possible values:

- 1 – Corresponding core is enabled.
- 0 – Core is disabled.

7.3.4 TARGET_AP

Applicable for "secure" (PSOC™ 64) MCUs only. Enables the choice of DAP access port that will be used for programming.

Possible values:

- sys_ap – SYS_AP (AP #0, default)
- cm0_ap – CM0_AP (AP #1)
- cm4_ap – CM4_AP (AP #2). Choosing this access port will enable external SMIF memory banks.

7.3.5 FLASH_RESTRICTION_SIZE

Applicable for "secure" (PSOC™ 64) MCUs only. Use this variable to limit the size of accessible flash so OpenOCD will not affect flash locations where the "secure" CyBootloader is located. The default value of this variable varies across different PSOC™ 64 MCUs.

7.3.6 ENABLE_WFLASH, ENABLE_SFLASH, ENABLE_EFUSE

Applicable for "secure" (PSOC 64) MCUs only. Enables the corresponding flash bank when set to a non-zero value. The WorkFlash is enabled by default on PSOC™ 64 CYS0644A, CYB0644A, CYB06447 and CYB06447-BL MCU devices. SFlash and eFuse banks are disabled by default on all PSOC™ 64 MCU targets.

7.3.7 SMIF_BANKS

Defines QSPI memory banks. This variable is a two-dimensional associative Tcl array of the following format:

```
set SMIF_BANKS {
  1 {addr <XIPaddr1> size <BankSz1> psize <ProgramSz1> esize <EraseSz1>}
  2 {addr <XIPaddr2> size <BankSz2> psize <ProgramSz2> esize <EraseSz2>}
  ...
  N {addr <XIPaddrN> size <BankSzN> psize <ProgramSzN> esize <EraseSzN>}
}
```

Where:

- `XIPaddrN` – XIP mapping address
- `BankSzN` – Total size of this flash bank, in bytes
- `ProgramSzN` – Minimal programming granularity (program block size), in bytes
- `EraseSzN` – Minimal erase granularity (erase block size), in bytes

7.4 PSOC™ 4 MCU global variables

7.4.1 PSOC4_USE_ACQUIRE

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

Possible values:

- 1 – Reset acquisition enabled (default with KitProg3/MiniProg4)
- 2 – Power Cycle acquisition enabled. The voltage level can be controlled by using ENABLE_POWER_SUPPLY.
- 0 – Acquisition disabled (default for other debug adapters)

7.5 AIROC™ CYW20829 Wi-Fi & Bluetooth® combo chip global variables

7.5.1 DEBUG_CERTIFICATE

Allows to specify the location of the debug certificate binary file. This variable is used to configure a secure debug session. OpenOCD checks the status of the CM33 access port during the initialization phase and after each reset. It will attempt to reopen the CM33 AP by sending a debug certificate to a target and issuing WFA request #2.

This variable should contain the full path to the debug certificate binary file.

7.5.2 DEBUG_CERTIFICATE_ADDR

(Optional) Allows to specify the location in the RAM where the debug certificate will be loaded. The default address 0x2000FC00 will be used if this variable is not set by the user.

7.5.3 SMIF_BANKS

See SMIF_BANKS under PSOC 6™ MCU.

7.6 XMC7xxx/XMC5xxx and TRAVEO™ T2G global variables

7.6.1 ENABLE_ACQUIRE

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

Possible values:

- 1 – Reset acquisition enabled (default with KitProg3/MiniProg4)
- 0 – Acquisition disabled (default for other debug adapters)

7.6.2 ENABLE_POWER_SUPPLY

Controls the internal power supply of KitProg3/MiniProg4 adapters. If this command is specified, the KitProg3 driver enables the power supply, thus powering on the target during initialization.

Possible values:

- 0 – Power supply disabled
- Any other value defines target voltage in millivolts.
- default – Sets the last used voltage before KitProg3/MiniProg4 was powered off.

7.6.3 ENABLE_CM71

Allows specifying the CPU cores to be visible to OpenOCD. Useful for single core XMC7xxx devices. OpenOCD never affects disabled cores.

Possible values:

- 1 – CM71 core is enabled.
- 0 – Core is disabled.

7.6.4 SMIF_BANKS

Defines QSPI memory banks. This variable is a two-dimensional associative Tcl array of the following format:

```
set SMIF_BANKS {
  1 {addr <XIPaddr1> size <BankSz1> psize <ProgramSz1> esize <EraseSz1>}
  2 {addr <XIPaddr2> size <BankSz2> psize <ProgramSz2> esize <EraseSz2>}
  ...
  N {addr <XIPaddrN> size <BankSzN> psize <ProgramSzN> esize <EraseSzN>}
}
```

Where:

- `XIPaddrN` – XIP mapping address
- `BankSzN` – Total size of this flash bank, in bytes
- `ProgramSzN` – Minimal programming granularity (program block size), in bytes
- `EraseSzN` – Minimal erase granularity (erase block size), in bytes

8 Usage examples

Most of the examples in this chapter assume that you have a PSOC™ Edge MCU target connected to the PC via the KitProg3/MiniProg4 or J-Link debug probe. The current working directory is the default install directory (for example, *c:\Infineon\Tools\ModusToolboxProgtools-1.6\openocd\bin* on Windows).

For convenience, the pse8_kp3_board.cfg config file has been created in the same directory as the OpenOCD executable. The file contains the default configuration suitable for the PSOC™ Edge MCU kits:

```
source [ find interface/kitprog3.cfg ]
set SMIF_BANKS { 0 {addr 0x60000000 size 0x01000000 psize 0x00000200 esize 0x00040000} }
source [ find target/infineon/pse84xgxs2.cfg]
init
reset init
```

See Supported target configurations for a detailed list of available target devices and corresponding OpenOCD configuration files.

8.1 Erase main flash rows 0...10 of PSOC™ Edge MCU

```
openocd -s ../scripts -f psoc6_kp3_board.cfg -c "flash erase_sector 0 0 10; exit"
```

A possible output of OpenOCD:

```
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: FW version: 2.80.1529
Info : KitProg3: Pipelined transfers enabled
Info : KitProg3: Asynchronous USB transfers enabled
Info : VTarget = 1.795 V
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
Info : clock speed 4000 kHz
Info : SWD DPIDR 0x4c013477
** Target acquired in Test Mode
Info : [cat1d.cm33] Cortex-M33 r1p0 processor detected
Info : [cat1d.cm33] target has 8 breakpoints, 4 watchpoints
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
[100%] [##############################] [ Erasing     ]
erased sectors 0 through 10 on flash bank 0 in 0.266261s
Info : cat1d.dap: powering down debug domain...
```

8.2 Display memory contents of PSOC™ Edge MCU

In this example, OpenOCD reads memory contents of the RAM memory as well as a flash memory.

```
openocd -s ../scripts -f pse8_kp3_board.cfg -c "init; reset init; mdw 0x34080000 32; flash mdw
0x70000000 32; exit"
```

A possible output of OpenOCD:

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
0x34080000: 06d606d8 80dcf5ea a7411223 c454d7d4 4f5b1146 8410ddf1 99832288 f3e4643f
0x34080020: 8a536940 3405d489 7e9b31c1 4a54f9a6 38f6000a a457f85e a5802702 650073d3
0x34080040: ba3fe065 8f49f9d9 4e7e6974 500a45f7 df74c215 47042f7c c99c9112 50a2ad34
0x34080060: c14f90ea 5133a8fa fbd30261 e20b9d87 1f8c6f68 c8d7df4d f3df72a9 ca5c66f1
0x70000000: fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe
0x70000020: fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe
0x70000040: fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe
0x70000060: fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe fefefefe
Info : cat1d.dap: powering down debug domain...
```

8.3 Program the PSOC™ Edge MCU 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 pse8_kp3_board.cfg -c "program c:/firmware/app_combined.hex verify
reset; exit"
```

A possible output of OpenOCD:

```
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
** Programming Started **
auto erase enabled
Info : Flash write discontinued at 0x60347890, next section at 0x60580400
Info : Section start address 0x60340400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60340000
Info : Padding image section 0 at 0x60347890 with 1904 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60348000 .. 0x6037ffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
Info : Section start address 0x60580400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60580000
Info : Padding image section 1 at 0x6058288c with 1908 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60583000 .. 0x605bffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
Info : Padding image section 2 at 0x701077d0 with 2096 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x70108000 .. 0x7013ffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
wrote 77824 bytes from file c:/firmware/app_combined.hex in 5.825991s (13.045 KiB/s)
** Programming Finished **
** Verify Started **
verified 69868 bytes in 1.594959s (42.779 KiB/s)
** Verified OK **
** Resetting Target **
Info : cat1d.dap: powering down debug domain...
```

8.4 Program the EFuse region of PSOC™ 6 MCU

This example writes a single bit of data to the EFuse region of the PSOC™ 6 MCU at address 0x907003FE:

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xxx.cfg -c "init;
reset init; psoc6 allow_efuse_program on; flash fillb 0x907003FE 1 1; flash read_bank 3 d:/
dump_efuse.bin 0x3FE 0x1; exit"
```

A possible output of OpenOCD:

```
Open On-Chip Debugger 0.10.0+dev-2.1.0.72 (2019-01-12-12:22)
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: 200
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 : VTarget = 3.297 V
Info : kitprog3: acquiring PSoC device...
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x6ba02477
Info : psoc6.cpu.cm0: hardware has 4 breakpoints, 2 watchpoints
Info : psoc6.cpu.cm4: hardware has 6 breakpoints, 4 watchpoints
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 conf
igure hardware srst support.
Info : kitprog3: acquiring PSoC device...
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00001f2c msp: 0x080477a8
** Device acquired successfully
** SFlash SiliconID:   0xE2062200
** Flash Boot version: 0x021D8001
** Chip Protection: NORMAL
** psoc6.cpu.cm4: Ran after reset and before halt...
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x1600400c msp: 00000000
Warn : Programming of efuses now ALLOWED
Info : MainFlash size overridden: 1024 kB
Info : Start address 0x907003fe breaks the required alignment of flash bank psoc6_efuse_cm0
Info : Padding 1022 bytes from 0x90700000
Info : Padding at 0x907003ff with 1 bytes (bank write end alignment)
Info : The Life Cycle stage is not present in the programming file
wrote 1 bytes to 0x907003fe in 0.041024s (0.024 KiB/s)
wrote 1 bytes to file d:/dump_efuse.bin from flash bank 3 at offset 0x000003fe in 0.016009s (0.061 KiB/s
```

8.5 Modify individual bytes of PSOC™ Edge MCU in main flash and display results

```
openocd -s ../scripts -f pse8_kp3_board.cfg -c "mdw 0x22011800 8; flash rmw 0x22011800
11223344; flash mdw 0x22011800 8; exit"
```

A possible output of OpenOCD:

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
0x22011800: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
modified 4 byte(s) in 1024 byte region at 0x22011800 in 0.319986s (3.125 KiB/s)
0x22011800: 44332211 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Info : cat1d.dap: powering down debug domain...
```

8.6 Read the memory of PSOC™ Edge MCU to binary file

The example reads 1024 KB of the QSPI PSOC™ Edge MCU memory to a file named *dump_mem.bin*.

```
openocd -s ../scripts -f pse8_kp3_board.cfg -c "dump_image d:/dump_mem.bin 0x70000000 0x100000;
exit"
```

A possible output of OpenOCD:

```
**************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
**************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
dumped 1048576 bytes in 4.927227s (207.825 KiB/s)
Info : cat1d.dap: powering down debug domain...
```

8.7 Start the GDB server and leave it running

```
openocd -s ../scripts -f pse8_kp3_board.cfg
```

A possible output of OpenOCD:

```
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: FW version: 2.80.1529
Info : KitProg3: Pipelined transfers enabled
Info : KitProg3: Asynchronous USB transfers enabled
Info : VTarget = 1.795 V
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
Info : clock speed 4000 kHz
Info : SWD DPIDR 0x4c013477
** Target acquired in Test Mode
Info : [cat1d.cm33] Cortex-M33 r1p0 processor detected
Info : [cat1d.cm33] target has 8 breakpoints, 4 watchpoints
************************************
** Silicon: 0xED94, Family: 0x115, Rev.: 0x21 (B0)
** Detected Device: PSE846GPS2DBZC4A
** SROM Boot version: 2.0.0.6022
** RRAM Boot version: 2.0.0.7127
** SE RT Services Base version: 1.0.0.2361
** SE RT Services version: 0.0.0.0
** Extended Boot version: 1.1.0.1700
** Boot Status : CYBOOT_SUCCESS
** Life Cycle  : DEVELOPMENT
************************************
Info : [cat1d.sys] Examination succeed
Info : [cat1d.sys33] Examination succeed
Info : [cat1d.cm33] Examination succeed
Info : gdb port disabled
Info : gdb port disabled
Info : starting gdb server for cat1d.cm33 on 3333
Info : Listening on port 3333 for gdb connections
Info : kitprog3: acquiring device in Test Mode using custom target sequence...
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
Info : [cat1d.cm33] Current domain secure state: Secure
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
```

8.8 Power example

This example cycles between setting different voltage levels (1800, 2500, 0, and lastly 2500).

```
openocd -s ../scripts -f interface/kitprog3.cfg -f target/infineon/cy8c6xx.cfg -c "kitprog3
power_config on 1800; init; kitprog3 get_power; kitprog3 power_config on 2500;kitprog3
get_power;kitprog3 power_config off;kitprog3 get_power;kitprog3 power_config on;kitprog3
get_power; shutdown"
```

A possible output of OpenOCD:

```
**************************************
** Silicon: 0xE206, Family: 0x100, Rev.: 0x22 (B1)
** Detected Device: CY8C6247BZI-D54
** Detected Main Flash size, kb: 1024
** Flash Boot version: 1.20.1.29
** Chip Protection: NORMAL
**************************************
Info : [psoc6.cpu.cm0] Examination succeed
Info : [psoc6.cpu.cm4] Cortex-M4 r0p1 processor detected
Info : [psoc6.cpu.cm4] target has 6 breakpoints, 4 watchpoints
Info : [psoc6.cpu.cm4] Examination succeed
Info : starting gdb server for psoc6.cpu.cm0 on 3333
Info : Listening on port 3333 for gdb connections
Info : starting gdb server for psoc6.cpu.cm4 on 3334
Info : Listening on port 3334 for gdb connections
VTarget = 1.798 V
Info : kitprog3: powering up target device using KitProg3 (VTarg = 2500 mV)
VTarget = 2.496 V
Info : kitprog3: powering down target device using KitProg3
Polling target psoc6.cpu.cm0 failed, trying to reexamine
Error: Error connecting DP: cannot read IDR
Error: [psoc6.cpu.cm0] Examination failed
Examination failed, GDB will be halted. Polling again in 100ms
VTarget = 0.007 V
Info : Power dropout detected, running power_dropout proc.
Power dropout, target voltage: 0.008 V
Info : kitprog3: powering up target device using KitProg3 (default voltage)
Power restore, target voltage: 2.496 V
Info : SWD DPIDR 0x6ba02477
Info : [psoc6.cpu.cm0] external reset detected
Polling target psoc6.cpu.cm0 failed, trying to reexamine
Info : [psoc6.cpu.cm0] Cortex-M0+ r0p1 processor detected
Info : [psoc6.cpu.cm0] target has 4 breakpoints, 2 watchpoints
Info : [psoc6.cpu.cm0] Examination succeed
Info : [psoc6.cpu.cm4] external reset detected
VTarget = 2.496 V
shutdown command invoked
Info : psoc6.dap: powering down debug domain...
```

8.9 Define and use QSPI flashloader

While some devices support several external memories or patched flashloaders, which contain data about how external memory is configured, the mechanism to changeswitch between default flashloaders and custom/ patched ones is created -the enablement requires them the QSPI_FLASHLOADER and SMIF_BANKS should be redefined. For example, to program

See QSPI Configurator user guide for more information about patching flashloaders.

```
openocd -s ../scripts -c "set QSPI_FLASHLOADER C:/firmware/PSE84_OCTAL_DDR_SMIF.FLM" -f
pse8_kp3_board.cfg -c "init; reset init; program c:/firmware/app_combined.hex verify; exit"
```

A possible output of OpenOCD:

8 Usage examples

```
** Target acquired in Test Mode
Info : [cat1d.cm33] external reset detected
[cat1d.cm33] halted due to debug-request, current mode: Thread
xPSR: 0xf9000000 pc: 0x120015d0 msp: 0x34005c88
** Programming Started **
auto erase enabled
Info : Flash write discontinued at 0x60347890, next section at 0x60580400
Info : Section start address 0x60340400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60340000
Info : Padding image section 0 at 0x60347890 with 1904 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60348000 .. 0x6037ffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
Info : Section start address 0x60580400 breaks the required alignment of flash bank cat1d.cm33.smif0_ns
Info : Padding 1024 bytes from 0x60580000
Info : Padding image section 1 at 0x6058288c with 1908 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x60583000 .. 0x605bffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
Info : Padding image section 2 at 0x701077d0 with 2096 bytes (bank write end alignment)
Warn : Adding extra erase range, 0x70108000 .. 0x7013ffff
[100%] [##################################] [ Erasing     ]
[100%] [##################################] [ Programming ]
wrote 77824 bytes from file c:/firmware/app_combined.hex in 5.964752s (12.742 KiB/s)
** Programming Finished **
** Verify Started **
verified 69868 bytes in 1.553341s (43.925 KiB/s)
** Verified OK **
Info : cat1d.dap: powering down debug domain...
```

Revision history

Revision	Date	Description
**	2019-01-17	New document.
*A	2019-01-24	Updated installation procedures for Windows and Linux sections.
*B	2019-04-16	Updated for version 2.2. Added CY8C6xx5 configuration. Added descriptions for show_verify_ranges, clear_verify_ranges, and psoc6 secure_app commands.
*C	2019-06-26	Removed all Traveo II (automotive) related information
*D	2019-12-13	Changed document name from CYPRESS™ Programmer 2.2 OpenOCD CLI User Guide to Cypress OpenOCD CLI User Guide. Clean-up in whole document. Deleted section 2 – Cypress Programmer Installation. Updated Supported Target Configurations table. Added description for "TARGET_AP" and "FLASH_RESTRICTION_SIZE" variables. Added "psoc6 secure_acquire" and deleted "psoc6 secure_app" commands. Added "cmsis_dap_serial" command description.
*E	2020-03-16	Added PSoC 4-related descriptions.
*F	2020-03-19	Added "flash protect" command description. Added description of ENABLE_WFLASH, ENABLE_SFLASH, ENABLE_EFUSE variables.
*G	2020-06-11	Added mention of the latest released version of Cypress OpenOCD is located on GitHub: https://github.com/cypresssemiconductorco/openocd/releases Updated section "Supported MCU Devices" Updated section "Supported Target Configurations" Updated description of "kitprog3 acquire_config" and "psoc6 secure_acquire" commands
*H	2020-07-29	Added configuration file for CY8C6xx4 device
*I	2020-02-08	Added description for "psoc4 chip_protect" command Added configuration file for PSoC 4500H Updated "flash erase_sector" and "flash erase_address" sections – document limitations of PSoC4 chips Updated "PSOC4_USE_ACQUIRE", "ENABLE_ACQUIRE", "kitprog3 acquire_config" sections – add support for power-cycle acquisition mode
*J	2021-09-10	Updated document title to ModusToolbox™ software OpenOCD CLI user guide Added description for the cmsis_flash driver and related commands Added description for global variables used with CYW20829 target Added CYW20829-specific commands Added CYW4390x-specific commands and limitations Added example of remote PSoC6 debugging configuration
*K	2022-08-30	The "kitprog3 power_control" command is now deprecated Support for XMC7xxx and TRAVEO™ T2G Body High devices Removed PSOC4_USE_MEM_AP variable as it is no longer required Added section 7.4 XMC7xxx and TRAVEO™ T2G global variables Updated section 6.3 with new CAT1C-related commands
*L	2023-05-05	Updated list of supported operating systems

Revision history

Revision	Date	Description
*M	2024-04-30	Updated list of supported operating systems Updated "About this document" and "Overview", "Installation", "List the connected targets" sections with relevant info on how to install ModusToolbox™ software OpenOCD Updated section "Supported devices" Updated section "Supported target configurations" Updated section "OpenOCD commands overview" Added description for "adapter serial" command Removed description of the deprecated "cmsis_dap_serial " command Updated "psoc4/cat1c ecc_error_reporting" command title
*N	2024-08-07	Updated section "Supported target configurations" Updated section "Supported devices"
*O	2024-10-04	Updated list of supported operating systems Updated section "Supported devices" Updated section "Supported target configurations" Updated section "OpenOCD commands overview" Updated section "Flash driver commands” Added sections on PSOC™ Control C3 Updated section "CYW4390x commands" Updated section "Remote debugging" Updated PSoC™ product category trademark to PSOC™
*P	2025-02-26	Updated list of supported operating systems Updated section "Remote debugging" Updated section "Flash driver commands" Updated section "PSOC™ Control C3 global variables"
*Q	2025-06-17	Added support of PSOC™ Edge E84 devices Added information about changing drivers for J-Link Added chip name information in supported target configurations Minor corrections
*R	2025-09-24	Updated section "Supported devices" Updated section "Supported target configurations" Updated examples and screen captures in the whole document with PSOC™ Edge info, where applicable Minor formatting fixes Added examples on power config and patched QSPI flashloaders usage Removed descriptions of deprecated commands
*S	2025-12-03	Updated list of supported operating systems Updated section "Supported target configurations" Updated section "Supported devices" Updated examples in "Getting Started" section Updated section "PSOC™ Control C3 global variables"
*T	2026-03-13	Updated list of supported operating systems Updated section "Supported target configurations" Updated section "Supported hardware (probes)"

Revision history

Revision	Date	Description
*U	2026-08-15	Updated "Supported target configurations" section Updated "Supported devices" section Updated information about J-Link support Updated "OpenOCD commands overview" section Updated “KitProg3/MiniProg4 driver commands” section
*V	2026-09-23	Updated the supported OS.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-09-23

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-ssl1750969375076

Important notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability. Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

If this document is marked as 'Preliminary', 'Target', 'Draft' or 'Proposal', Infineon reserves the right to change all information given in this document at any time without notice. Subject to the development and release of a Product for series supply by Infineon, the technical specifications of the Product are set forth in the relevant datasheet provided by Infineon without such marking.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury.

However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer. Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG). If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited.

For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit www.infineon.com.