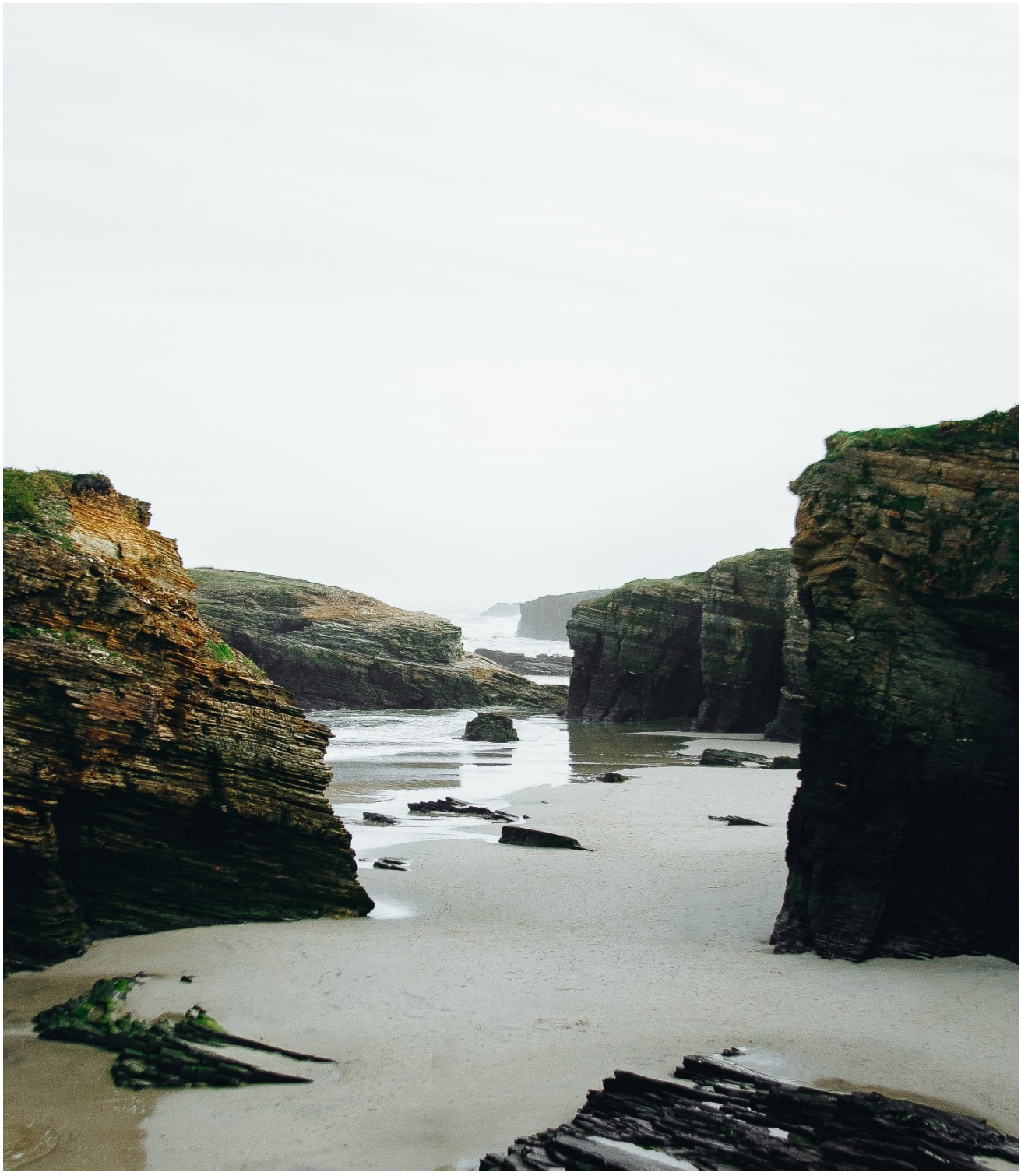
SMH Technologies®

19/12/2023

Driver v. 5.00

Moreno Ortolan

Interfacing FlashRunner 2.0 with INFINEON PSoC6



SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

INFINEON PSoC6 Introduction

32-bit PSoC™ Arm® Cortex® Microcontroller

The PSoC™ 6 family is built on an ultra-low-power architecture, and the MCUs feature low-power design techniques that are ideal for battery powered applications.

The dual-core Arm® Cortex®-M4 and Cortex-M0+ architecture lets designers optimize for power and performance simultaneously. Using its dual cores combined with configurable memory and peripheral protection units, the PSoC™ 6 MCU delivers the highest level of protection defined by the Platform Security Architecture (PSA) from Arm.

Designers can use the MCU's rich analog and digital peripherals to create custom analog front-ends (AFEs) or digital interfaces for innovative system components such as MEMS sensors, electronic-ink displays.

The PSoC™ 6 MCU features the latest generation of industry-leading CAPSENSE™ capacitive-sensing technology, enabling modern touch and gesture-based interfaces that are robust and reliable.

PSoC™ 6 MCU, paired with Infineon's AIROC™ Wi-Fi, AIROC™ Bluetooth®, or AIROC™ combos radio modules, is the perfect solution for secure, low-power, feature-rich IoT products.

32-bit PSoC™ Arm® Cortex® Microcontroller subcategories

- 32-bit PSoC™ 6 Arm® Cortex®-M4 / M0+
  - › Overview
  - › PSoC™ 61 - Entry-level
  - › PSoC™ 62 - Performance
  - › PSoC™ 63 - Bluetooth™ Low Energy
  - › PSoC™ 64 - Secured MCU

32-bit PSoC™ 61 - Entry Level

The PSoC™ 61 programmable line, built on an ultra-low-power 40-nm platform, features an Arm® Cortex®-M4 CPU, with low-power Flash technology, programmable digital and analog resources, and best-in-class CAPSENSE™ technology for touch and proximity applications.

Security is built in to the platform architecture with hardware cryptographic accelerators, memory and peripheral protection units. It's designed for applications in the Internet of Things, such as wearables, smart home, industrial IoT, portable medical devices, etc.

PSoC™ 61 programmable line is offered in a variety of packages including BGA, WLCSP, QFN, TQFP, and supports up to 2 MB of flash, 1 MB of on-chip SRAM, and up to 104 GPIOs.

32-bit PSoC™ 62 - Performance

The PSoC™ 62 performance line, built on an ultra-low-power 40-nm platform, is a combination of Arm® Cortex®-M4 and Arm® Cortex®-M0+ CPUs, with low-power Flash technology, programmable digital and analog resources, and best-in-class CAPSENSE™ technology for touch and proximity applications.

Security is built in to the platform architecture with hardware cryptographic accelerators, memory and peripheral protection units. It's designed for applications in the Internet of Things, such as wearables, smart home, industrial IoT, portable medical devices, etc.

PSoC™ 62 performance line is offered in a variety of packages including BGA, WLCSP, QFN, TQFP, and supports up to 2 MB of flash, 1 MB of on-chip SRAM, and up to 104 GPIOs.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

32-bit PSoC™ 63 - MCU with AIROC™ Bluetooth® LE

Infineon offers PSoC™ 63 MCU with AIROC™ Bluetooth® Low Energy (LE) to drive your low-power IoT devices. It has dual-core 150-MHz Arm® Cortex®-M4 and 100-MHz Arm® Cortex®-M0+ processors, offering industry's highest compute capability, optimized for AI/ML Edge applications.

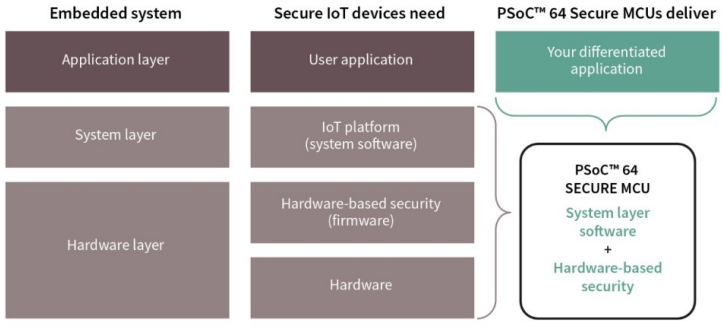
The PSoC™ 63 MCU with AIROC™ Bluetooth® LE integrates programmable analog front ends, industry-leading CAPSENSE™ touch sensing user interface, and Bluetooth® LE radio.
It includes a royalty-free Bluetooth® LE protocol stack compatible with Bluetooth® 5.4.

The PSoC™ 63 with AIROCTM Bluetooth® LE MCU supports rich configurable MCU peripherals with 84 programmable GPIOs and six overvoltage-tolerant capable pins, making it suitable for a wide range of IoT applications.

32-bit PSoC™ 64 - Secured MCU

The PSoC™ 64 line incorporates all of the key features of PSoC™ 6 with preconfigured security and connectivity software to support secure onboarding, secure boot, secure firmware updates and secure runtime services based on Trusted Firmware-M (TF-M)

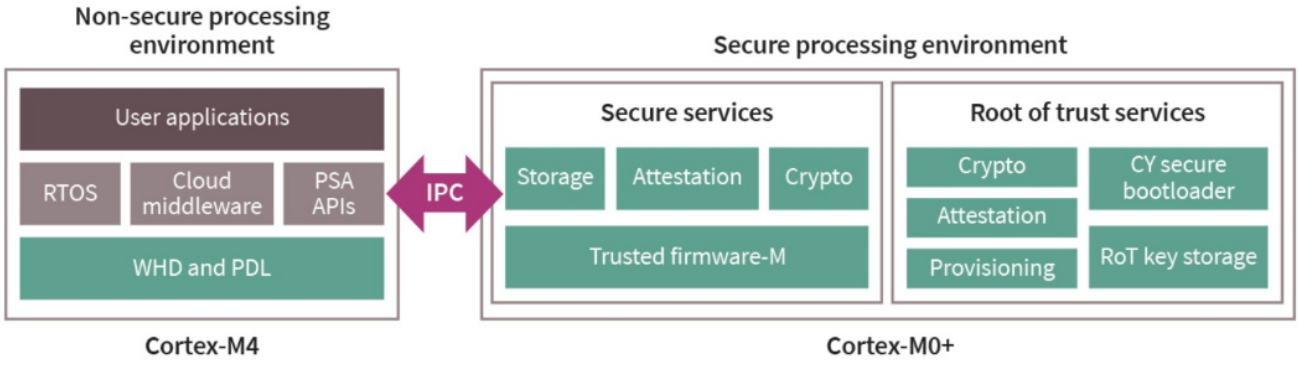
With a growing number of devices connecting to the internet, security must be established between hardware, cloud applications and servers, and finally users and services.
PSoC™ 64 Secure MCUs integrate the award-winning, ultra-low power PSoC™ 6 architecture with well-structured open-source IoT platform software to deliver a secure solution that Auxiliarys.



The dual-core architecture of PSoC™ 6 is ideal for establishing isolated processing environments.

The Cortex-M4 processor is used to establish a Non-Secure Processing Environment (NSPE) and the Cortex-M0+ is used to establish a Secure Processing Environment (SPE) through the use of protection units built into PSoC™ 6.

Trusted Firmware-M running in the SPE communicates to the NSPE through a hardware-based Inter-Processor Interface (IPC). The root-of trust is isolated form the SPE and provides an immutable identity for the device and enables secure key storage. Security services include secure boot, provisioning, and attestation.



SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

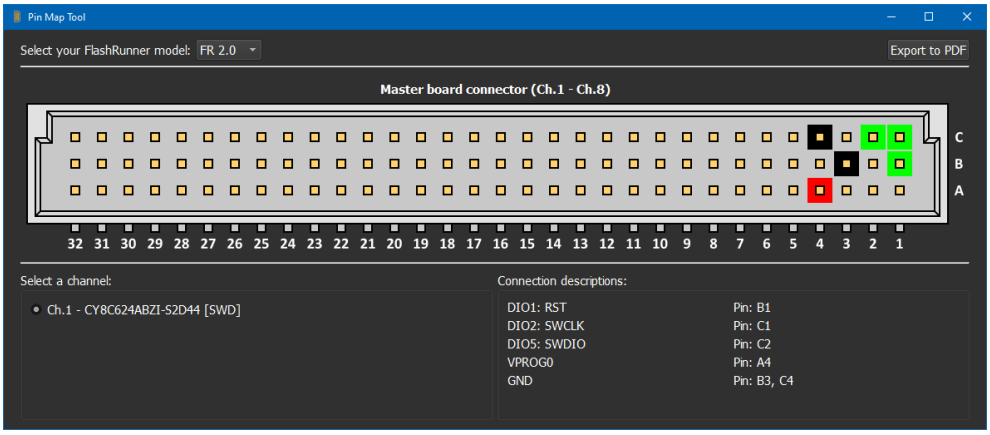
info@smh-tech.com

INFINEON PSoC6 Protocol and PIN map

PSoC6 devices support the SWD protocol.

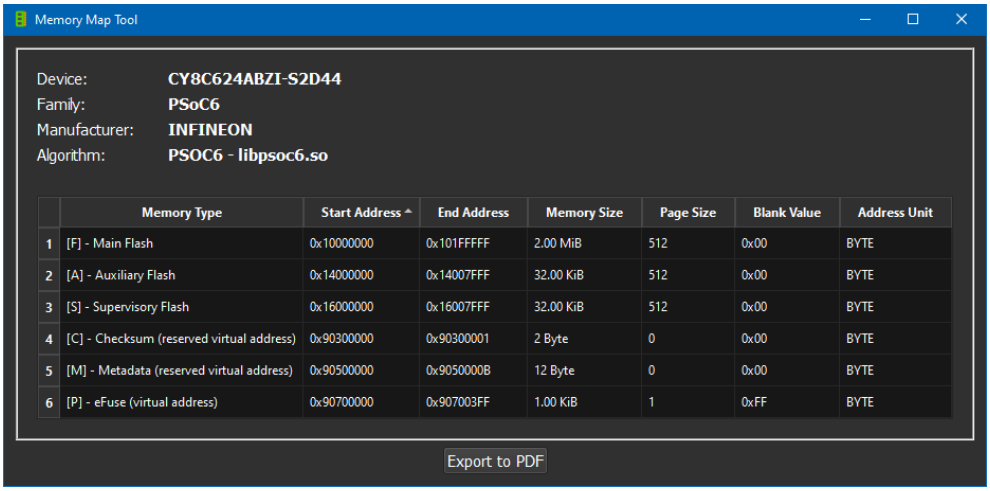
```
#TCSETPAR CMODE <SWD>
```

INFINEON PSoC6 PIN MAP



INFINEON PSoC6 Memory Map

Memory Type	Start Address	End Address	Memory Size	Page Size	Blank Value	Address Unit
[F] - Main Flash	0x10000000	0x101FFFFF	2.00 MiB	512	0x00	BYTE
[A] - Auxiliary Flash	0x14000000	0x14007FFF	32.00 KiB	512	0x00	BYTE
[S] - Supervisory Flash	0x16000000	0x16007FFF	32.00 KiB	512	0x00	BYTE
[C] - Checksum (reserved virtual address)	0x90300000	0x90300001	2 Byte	0	0x00	BYTE
[M] - Metadata (reserved virtual address)	0x90500000	0x9050000B	12 Byte	0	0x00	BYTE
[P] - eFUSE (virtual address)	0x90700000	0x907003FF	1.00 KiB	1	0xFF	BYTE



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

SMH Technologies®

INFINEON PSoC6 eFUSE

This is an OTP area (One-Time-Programmable) and this means that once a bit is blown, so it has the '1' state, it cannot return to the '0' state.

Single eFUSE can be changed in state bit-by-bit by putting the value in .FRB file and using the program command.

Only bits that are different from '0' will be written because the original state of a bit of eFUSE is '0' and then can become '1' by blowing.
If an eFUSE byte (different from 0x0 and 0xFF) is on the FRB file but the target device has already that byte written, the eFUSE **is NOT blown another time**.

If an eFUSE byte (different from 0x0 and 0xFF) is on .FRB file but it is different from eFUSE byte read from device, only '1' bits will be written.

After programming (blowing process) a check operation (verify) checks byte-by-byte the eFUSE area, comparing .FRB eFUSE values with current device values read from eFUSE area.

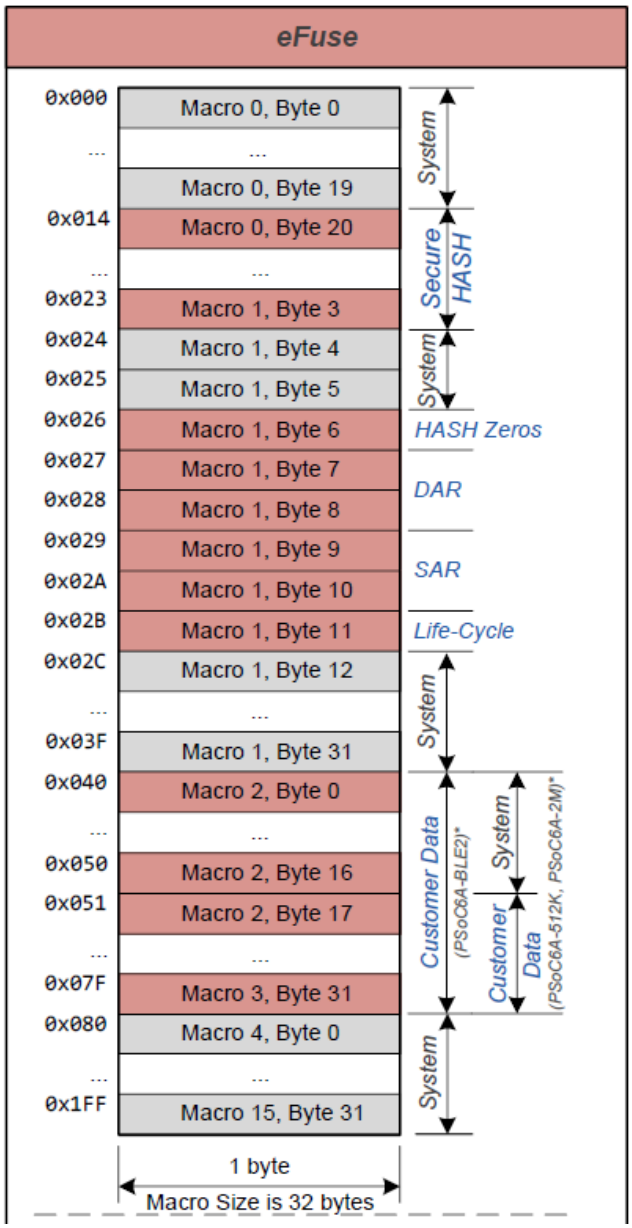
The driver writes bit-by-bit the eFUSE value but the minimum blowing size is the value of **1 byte** in .FRB file, this means that in the .FRB you have to consider the entire 8-bit value of eFUSE.

Since the eFUSE area is not accessed via the address space then FlashRunner refers to a virtual address which is declared in the programming specification of Infineon.

This address starts from 0x90700000 up to 0x907003FF and corresponds to the eFUSE **bit** area (**not byte area**).

According to specifications, each eFUSE byte is made up of 8 eFUSEs bits and each location of the virtual address space corresponds to the eFUSEs bit to be programmed.

Please see Infineon's programming specifications to better understand how eFUSE memory area Auxiliarys.



INFINEON PSoC6 SWD and JTAG Locking

In PSoC6 devices there is the possibility to lock permanently the DAP (Debug Access Port) of the SWD/JTAG.
This operation is irreversible and once the DAP is disabled, it is no longer possible to connect with the device.

To disable the DAP, it is necessary to set the **TOC2_FLAGS** which are 2 bytes located at the memory address **0x16007DF8** of the Supervisory Flash.

To lock the device, you can use the command `#TPCMD PATCH_SUPERVISORY_MEMORY`. Please refer to the command description.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

Here are the **TOC2_FLAGS**:

Bit	Name	Description
bit [1:0]	**CLOCK_CONFIG**	Indicates clock frequency configuration. The clock should stay the same after Flash boot execution 0 = 8 MHz, IMO, no FLL 1 = 25 MHz IMO + FLL 2 = 50 MHz IMO + FLL 3 = Use ROM boot clock configuration
bit [4:2]	**LISTEN_WINDOW**	Determines the Listen window to allow sufficient time to acquire debug port. 0 = 20 ms (Default) 1 = 10 ms 2 = 1 ms 3 = 0 ms (No Listen window) 4 = 100 ms
bit [6:5]	**SWJ_PINS_CTL**	Determines if SWJ pins are configured in SWJ mode by Flash boot. 0 = Do not enable SWJ pins in Flash boot. Listen window is skipped 1 = Do not enable SWJ pins in Flash boot. Listen window is skipped 2 = Enable SWJ pins in Flash boot (default) 3 = Do not enable SWJ pins in Flash boot. Listen window is skipped
bit [8:7]	**APP_AUTH_CTL**	Determines if the application image digital signature verification (authentication) is performed: 0 = Authentication is enabled (default) 1 = Authentication is disabled 2 = Authentication is enabled (recommended) 3 = Authentication is enabled
bit [10:9]	**FB_BOOTLOADER_CTL**	Determine if the internal bootloader in Flash boot is disabled: 0 = Internal bootloader is disabled 1 = Internal bootloader is launched if the other bootloader conditions are met (default) 2 = Internal bootloader is disabled 3 = Internal bootloader is disabled.

Set **LISTEN_WINDOW** = **3** to disable the listening window of time to acquire the device in SWD or JTAG protocol.

Set **SWJ_PINS_CTL** = **0** to disable SWD or JTAG pin function (dedicated for debugging and flashing) after the Flash Boot has been executed.

INFINEON PSoC6 Driver Parameters

The standard parameters are used to configure some specific options inside PSoC6 driver.

#TCSETPAR ENTRY_CLOCK

Syntax: `#TCSETPAR ENTRY_CLOCK <Frequency>`

`<Frequency>` Accepted parameters 4000000, 2000000, 1000000, 500000, 100000 Hz

Description: Set the JTAG/SWD frequency used in the Connect procedure before raising the PLL of the device, if the device PLL is available

Note: Default value 1.00 MHz

#TCSETPAR ACQUIRING_SEQUENCE

Syntax: `#TCSETPAR ACQUIRING_SEQUENCE`

Description: This parameter defines the entry mode
The acquiring chip procedures are defined by Infineon and you can find them in the Reference Manuals of PSoC6 devices

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

The *ACQUIRE_CHIP* is a procedure that uses the reset line to establish the communication between the FlashRunner and the target device

The other procedure, *ALTERNATE_ACQUIRE_CHIP* does not use the reset line to establish the communication.

Note: By default, the driver uses the *ACQUIRE_CHIP* method

#TCSETPAR BLANKCHECK_IN_PROGRAM_FLASH

Syntax: `#TCSETPAR BLANKCHECK_IN_PROGRAM_FLASH <Value>`

<Value> Accepted values are Yes or No

Description: The blankcheck command can be executed on the Main Flash memory
There are two possible ways to perform the blankcheck operation

The first choice is to perform the command #TPCMD BLANKCHECK F
The second choice is to perform the operation during the #TPCMD PROGRAM F command

Note: None

#TCSETPAR EFUSE_MARGIN_LEVEL

Syntax: `#TCSETPAR EFUSE_MARGIN_LEVEL <NOMINAL_RESISTANCE|LOW_RESISTANCE|HIGH_RESISTANCE>`

NOMINAL_RESISTANCE → default read condition
LOW_RESISTANCE → -50% from nominal resistance
HIGH_RESISTANCE → +50% from nominal resistance

Description: This command is used to in order to define the Margin Verify mode for the eFUSE
The margin verify is available only for the eFUSE memory area

This verification procedure differs from the classic one in re-reading the values stored in the memory by changing certain levels of current and supply voltage of the flash memory

This allows greater reliability as the data are read back in power range conditions other than the typical ones
The procedure provided by Infineon is carried out automatically by the HW of the device during the writing

Note: All other information regarding this command is **Under NDA**

#TCSETPAR SAMPLING_POINT

Syntax: `#TCSETPAR SAMPLING_POINT <Value>`

<Value> Accepted values are in the range 1-15

Description: Use this parameter to permanently set the sampling point of the FPGA
It is recommended to leave this parameter with the default value

Note: Default value 17

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

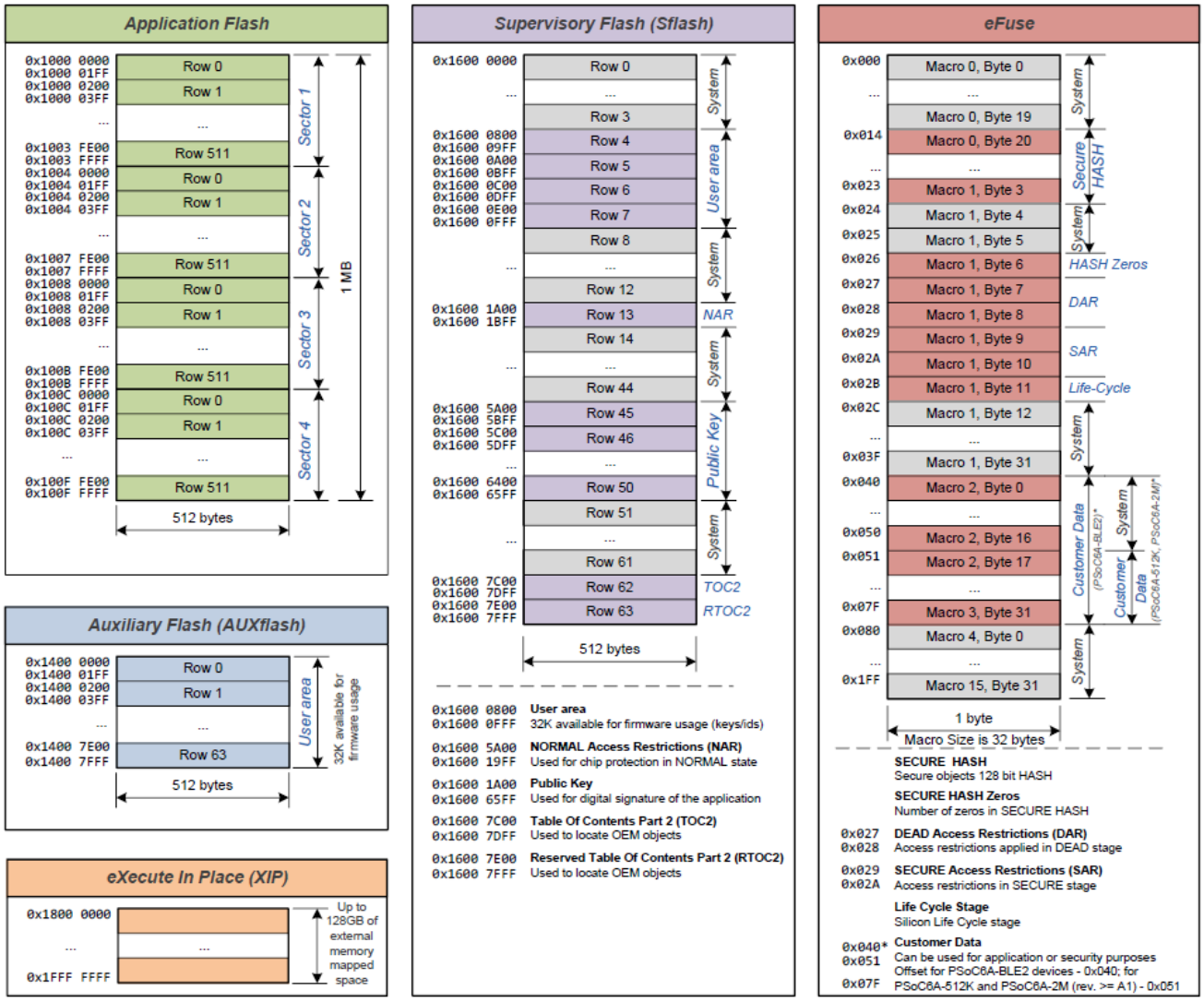
→smh-tech.com

info@smh-tech.com

INFINEON PSoC6 Driver Commands

Here you can find the complete list of all available commands for PSoC6 driver.

```
F → Main Flash
A → Auxiliary Flash
S → Supervisory Flash
C → Checksum (reserved virtual address) On this virtual memory area is stored the checksum of the firmware
M → Metadata (reserved virtual address) On this virtual memory area is stored the Silicon ID and other relevant data
P → eFUSE (virtual address)
```



#TPCMD CONNECT

#TPCMD CONNECT

This function performs the entry and is the first command to be executed when starting the communication with the device.
There are two types of connect, regarding this please read #TCSETPAR ACQUIRING_SEQUENCE.
Here you see can the log of a standard connect with the Acquire Chip procedure selected:

```
---#TPCMD CONNECT
Protocol selected SWD.
Toggling XRES pin to execute Acquire Chip procedure.
ID-Code read correctly at 1.00 MHz after 8 retries.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

```
JTAG-SWD Debug Port enabled.
Move PSoC6 internal state to Test Mode.
PSoC6 enter into Test Mode.
Scanning AP map to find all APs:
 * AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[1] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[2] IDR: 0x24770011, Type: AMBA AHB3 bus.
Scanning AP to find all cores:
 * AP[1] Found Cortex M0+ revision r0p1.
         CPUID: 0x410CC601.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xF0000000.
 * AP[2] Found Cortex M4 revision r0p1.
         CPUID: 0x410FC241.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xE00FF000.
Try to halt the Cortex M0+ core:
 * AP[1] Cortex M0+ Core halted [0.002 s].
Try to halt the Cortex M4 core:
 * AP[2] Cortex M4 Core halted [0.001 s].
Try to execute the Acquire Chip method procedure:
 * AP[1] Cortex M0+ core Vector Table base 0xFFFF0000.
 * Vector Table value means that the Flash is empty or TOC is corrupted.
 * Acquire Chip method procedure completed.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 5-7].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Read device informations:
 * FamilyId: 0x0102.
 * Revision: 0x12.
 * SiliconID: 0xE453.
 * Protect state: 0x2: Virgin - [NORMAL] - Secure - Dead.
 * Life cycle stage: 0x1 - Normal.
 * Flash boot version: 0xA301.
 * SROM firmware version: 0x0701.
Read device unique ID:
 * Unique ID0: 0x008C7902, Unique ID1: 0xAF1C1C01, Unique ID2: 0x7A040601.
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
 * Disabled Direct Execute system call functionality.
> Check PSoC6 Silicon ID passed from SMH database [0xE453].
 * Check PSoC6 Silicon ID from source file not available.
Time for Connect: 0.155 s.
>|
```

ID-Code read correctly at 1.00 MHz after 8 retries

This is normal behaviour because when the PSoC6 device is powered on the SWD/JTAG port is not immediately available.
The PSoC6 driver tries to read the ID code several times before the SWD/JTAG port is activated.

Check PSoC6 Silicon ID from source file not available

This warning means that the FRB file is not present or the Silicon ID is not available inside FRB file.
Therefore, it is not possible to compare the Silicon ID read from the device with the one present in the file to be programmed.

#TPCMD MASSERASE

#TPCMD MASSERASE <F|A>

This function performs a masserase for Main Flash or Auxiliary Flash memory.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

#TPCMD ERASE

`#TPCMD ERASE <F|A>`
This function performs a sector erase for all the Main Flash or Auxiliary Flash memory.

`#TPCMD ERASE <F|A> <start address> <size>`
This function performs a sector erase for selected part of Main Flash or Auxiliary Flash memory.
Enter the Start Address and Size in hexadecimal format.

#TPCMD BLANKCHECK

`#TPCMD BLANKCHECK <F|A>`
Blankcheck is only available for Main Flash and Auxiliary Flash memory.
Verify if all memory is erased.

`#TPCMD BLANKCHECK <F|E> <start address> <size>`
Blankcheck is only available for Main Flash and Auxiliary Flash memory.
Verify if selected part of memory is erased.
Enter the Start Address and Size in hexadecimal format.

#TPCMD PROGRAM

`#TPCMD PROGRAM <F|A|S|P>`
Program available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Programs all memory of the selected type based on the data in the FRB file.

The Supervisory Flash contains bytes whose value corresponds to the PSoC6 settings and protections.
Writing random values in this area produces unpredictable results, so if you need to modify only some specific bits, please use the `#TPCMD PATCH_SUPERVISORY_FLASH` command.

The eFUSE is an OTP area (One-Time-Programmable) and once a byte is written it cannot be changed anymore.
Note that both Supervisory Flash and eFUSE have zones that cannot be programmed.

`#TPCMD PROGRAM <F|A|S|P> <start address> <size>`
Program available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Programs selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

The Supervisory Flash contains bytes whose value corresponds to the PSoC6 settings and protections.
Writing random values in this area produces unpredictable results, so if you need to modify only some specific bits, please use the `#TPCMD PATCH_SUPERVISORY_FLASH` command.

The eFUSE is an OTP area (One-Time-Programmable) and once a byte is written it cannot be changed anymore.
Note that both Supervisory Flash and eFUSE have zones that cannot be programmed.

#TPCMD VERIFY

`#TPCMD VERIFY <F|A|S|P> <R>`
R: Readout Mode.
Verify Readout available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Verify all memory of the selected type based on the data in the FRB file.

`#TPCMD VERIFY <F|A|S|P> <R> <start address> <size>`
R: Readout Mode.
Verify Readout available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Verify selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

`#TPCMD VERIFY <F|A|S|P> <S>`

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

S: Checksum 32 Bit Mode.
Verify Checksum available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Verify all memory of the selected type based on the data in the FRB file.

`#TPCMD VERIFY <F|A|S|P> <S> <start address> <size>`
S: Checksum 32 Bit Mode.
Verify Checksum available for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
Verify selected part of memory based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD VERIFY_MARGIN_EFUSE

`#TPCMD MARGIN_VERIFY`
Margin Verify is available only for eFUSE memory.
Verify Margin of all eFUSEs.

`#TPCMD MARGIN_VERIFY <Virtual Start Address> <Virtual Size>`
Margin Verify is available only for eFUSE memory.
Verify Margin selected part of eFUSEs.

#TPCMD READ

`#TPCMD READ <F|A|S|P>`
`#TPCMD READ <F|A|S|P> <start address> <size>`
Read function for Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
The result of the read command will be visible into the Terminal.

Note that some eFUSE bytes cannot be read for protection reasons.

#TPCMD DUMP

`#TPCMD DUMP <F|A|S|P>`
`#TPCMD DUMP <F|A|S|P> <start address> <size>`
Dump command for the Main Flash, Auxiliary Flash, Supervisory Flash and eFUSE.
The result of the dump command will be stored in the FlashRunner 2.0 internal memory.

#TPCMD GET_UNIQUE_ID

Syntax: `#TPCMD GET_UNIQUE_ID`

Prerequisites: none

Description: This function reads the unique ID of the device
The result of this command will be printed on the Real Time Log and on the Terminal

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD GET_UNIQUE_ID
Read device unique ID:
 * Unique ID0: 0x008CCB31, Unique ID1: 0xAF323D12, Unique ID2: 0x7A081D01.
Time for Get Unique ID: 0.001 s.
```

#TPCMD GET_DEVICE_INFORMATIONS

Syntax: `#TPCMD GET_DEVICE_INFORMATIONS`

Prerequisites: none

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Description: This function gets the device information as the family, unique ID and more

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
Read device informations:
 * FamilyId: 0x0102.
 * Revision: 0x12.
 * SiliconID: 0xE457.
 * Protect state: 0x2: Virgin - [NORMAL] - Secure - Dead.
 * Life cycle stage: 0x1 - Normal.
 * Flash boot version: 0xA301.
 * SROM firmware version: 0x0701.
Read device unique ID:
 * Unique ID0: 0x008CCB31, Unique ID1: 0xAF323D12, Unique ID2: 0x7A081D01.
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
 * Disabled Direct Execute system call functionality.
Time for Get Device Informations: 0.003 s.
```

#TPCMD OVERVIEW_SUPERVISORY_FLASH

Syntax: `#TPCMD OVERVIEW_SUPERVISORY_FLASH`

Prerequisites: none

Description: This command reads the Supervisory Flash printing its content on the Workbench Real Time Log. Here below there is a little extract of the result of the command execution.

> *Supervisory Flash - Table of Contents Part 2 (TOC2), used to locate OEM objects:*
> *Address 0x16007C00: FC010000 20122101 00000000 00000010 00000000 00000000 etc ...*
>
> *The data has to be read in this way:*
> *most significant byte 0x00 then 0x00 then 0x01 and then, the less significant byte is 0xFC.*
> *If you want to reprogram this value the right data value is 0x000001FC*

#TPCMD OVERVIEW_SUPERVISORY_FLASH_NADR

Syntax: `#TPCMD OVERVIEW_SUPERVISORY_FLASH_NADR`

Prerequisites: none

Description: This command analyses the NADR (Normal/Dead Device Access Restrictions) located into the Supervisory Flash

Examples: Correct command execution: 😊

```
---#TPCMD OVERVIEW_SUPERVISORY_FLASH_NADR
Analyze NADR located into Supervisory Flash:
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
 * Disabled Direct Execute system call functionality.
Dead device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
  * Entire Supervisory Flash main region is accessible.
  * Entire Flash main region is accessible.
  * Entire SRAM main region is accessible.
  * Disabled Direct Execute system call functionality.
Time for Overview Supervisory Flash NADR: 0.002 s.
```

#TPCMD OVERVIEW_SUPERVISORY_FLASH_TOC2

Syntax: `#TPCMD OVERVIEW_SUPERVISORY_FLASH_TOC2`

Prerequisites: none

Description: This command analyses the TOC2 located into the Supervisory Flash

Examples: Correct command execution: 😊

```
---#TPCMD OVERVIEW_SUPERVISORY_FLASH_TOC2
Analyze TOC2 located into Supervisory Flash:
  * Object size in bytes for CRC calculation starting from offset 0x00: 0x000001FC.
  * Magic number: 0x01211220.
  * Address of Key Storage Flash Blocks. This also marks the top of flash available: 0x00000000.
  * Null-terminated table of pointers representing the SMIF configuration structure: 0x10000000.
  * Address of First User Application Object: 0x00000000.
  * Format of First User Application Object: 0x10000000 - Simplified.
  * Address of Second User Application Object: 0x00000000.
  * Format of Second User Application Object: 0x00000000 - Basic.
  * Number of additional objects to be verified for SECURE_HASH: 0x00000000.
  * Address of signature verification key: 0x00000000.
  * Controls default configuration: 0x00000042.
    * Clock frequency configuration: 50MHz, IMO + FLL (default).
    * Listen window to allow sufficient time to acquire debug port: 20 ms (default).
    * SWJ pins configuration: Enable SWJ pins in Flash boot (default).
    * Image digital signature verification: Authentication is enabled.
  * CRC16-CCITT: 0xE7FA0000.
Time for Overview Supervisory Flash TOC2: 0.002 s.
```

#TPCMD GENERATE_HASH

Syntax: `#TPCMD GENERATE_HASH <FACTORY_HASH|ALL_OBJECTS>`

FACTORY_HASH → Generates the factory hash
ALL_OBJECTS → Generates the hash of all objects according to TOC1 and TOC2 tables

Prerequisites: none

Description: This command returns the truncated SHA-256 of the Flash boot programmed in the Supervisory Flash
Please refer to specific Reference Manual of your PSoC6 device to have full description of how this command works

#TPCMD CHECK_FACTORY_HASH

Syntax: `#TPCMD CHECK_FACTORY_HASH`

Prerequisites: none

Description: This command generates the factory hash according to TOC1 table and compares with the FACTORY1_HASH fuses
Please refer to specific Reference Manual of your PSoC6 device to have full description of how this command works

#TPCMD COMPUTE_BASIC_HASH

Syntax: `#TPCMD COMPUTE_BASIC_HASH <BASIC|CRC8SAE> <Start address> <Size Bytes>`

Prerequisites: none

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Description: This command generates the hash of the flash region provided as input.
The command allows to select between BASIC (Basic Hash) or CRC8SAE.
The Size Bytes parameter must be grater or equal to 1
Please refer to specific Reference Manual of your PSoC6 device to have full description of how this command works

Examples: Correct command execution: 😊

Example with BASIC parameter:

```
---#TPCMD COMPUTE_BASIC_HASH BASIC 0x10000000 0x10
Compute basic Hash:
 * Data Hash: 0x38.
Time for Compute BASIC Hash: 0.001 s.
```

Example with CRC8SAE parameter:

```
---#TPCMD COMPUTE_BASIC_HASH CRC8SAE 0x10000000 0x10
Compute CRC8SAE Hash:
 * Data Hash: 0xE9.
Time for Compute CRC8SAE Hash: 0.001 s.
```

#TPCMD COMPUTE_CHECKSUM

Syntax: `#TPCMD COMPUTE_CHECKSUM <FLASH|AUXILIARY_FLASH|SUPERVISORY_FLASH> <PAGE|WHOLE_MEMORY> <Row ID>`

Prerequisites: none

Description: This command returns the sum of each byte read.
The Row ID parameter is needed only if the previous parameter is PAGE.
Please refer to specific Reference Manual of your PSoC6 device to have full description of how this command Works

#TPCMD PATCH_SUPERVISORY_MEMORY

Syntax: `#TPCMD PATCH_SUPERVISORY_MEMORY <Address> <Value> <Mask>`

Prerequisites: none

Description: This command allows the user to patch some specific data at a certain address (into the Supervisory Flash) without erasing the other addresses
Here below there are some examples:

1. `#TPCMD PATCH_SUPERVISORY_MEMORY 0x16007C00 0x12345678 0x00000000`

   This command does not program the value 0x12345678 because the mask is all 0x00000000 and for this reason the value will not be programmed at that address.

   More precisely we read all the data aligned to 512 bytes and we patch the data at 0x16007C00 with the value inserted by the user only if the corresponding bit is equal to 1 in the mask field.

   For example, if you set the mask equal to 0x0000000F you change only the bits where the mask is 1 and the other bits will not change.

2. `#TPCMD PATCH_SUPERVISORY_MEMORY 0x16007C00 0x000001FC 0xFFFFFFFF`

   This command allows to program the value 0x000001FC at the address 0x16007C00 without considering the memory content at this address because the mask is all 1.

   Here some examples with custom mask:

   This is the memory content using the command `#TPCMD OVERVIEW_SUPERVISORY_FLASH`:

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

Address 0x16000800: BBAA00FF FFFFFFFF FFFFFFFF FFFFFFFF

If for example you want to change only the first byte at the address 0x16000800 (the first byte is 0xBB) with a new value like 0xCC, you must use the following syntax:

3. `#TPCMD PATCH_SUPERVISORY_MEMORY 0x16000800 0x000000CC 0x000000FF`

In this case using the mask with only the first byte with all bits at 1 you change only the first byte of the memory to 0xCC and the rest will be untouched.

Infact, if you execute another time the command `#TPCMD OVERVIEW_SUPERVISORY_FLASH` you can see that:

Address 0x16000800: CCAA00FF FFFFFFFF FFFFFFFF FFFFFFFF

Now if for example you want to change another address like 0x16000808 using the command

`#TPCMD PATCH_SUPERVISORY_MEMEORY` 0x16000808 0xADDEADDE 0xFFFFFFFF

You can see the new content of the memory

Address 0x16000800: CCAA00FF FFFFFFFF DEADDEAD FFFFFFFF

As you can see the previous address has been left as it was before.

#TPCMD TRANSITION_TO_RMA

Syntax: `#TPCMD TRANSITION_TO_RMA` <UNIQUE ID0> <UNIQUE ID1> <UNIQUE ID2> <SRAM ADDRESS>

Prerequisites: none

Description: This command converts parts from secure or secure with debug state into the RMA life-cycle stage
Please refer to specific Reference Manual of your PSoC6 device to have full description of how this command works

#TPCMD TRANSITION_TO_SECURE

Syntax: `#TPCMD TRANSITION_TO_SECURE` <D|S> < SECURE_ACCESS_RESTRICT 32Bit> <DEAD_ACCESS_RESTRICT 32Bit>

Prerequisites: none

Description: This command validates the FACTORY_HASH and programs the SECURE_HASH, secure access restrictions and dead access restrictions into eFUSE.
The first parameter of the command is <D|S>
D → SECURE_WITH_DEBUG life-cycle stage, with this parameter debuggers can read Flash memory and perform operations on Auxiliary Flash and read device information.
S → SECURE life-cycle stage

The second parameter is <SECURE_ACCESS_RESTRICT 32Bit>
For this parameter, please refer to specific Reference Manual of your PSoC6 device

The third parameter is <DEAD_ACCESS_RESTRICT 32Bit>
For this parameter, please refer to specific Reference Manual of your PSoC6 device

The new secure state is applied when a new Power on Reset is provided to the device. If you try to perform a new execution on the FlashRunner project with the command `#TPCMD GET_DEVICE_INFORMATIONS` it is possible to check the new life-cycle stage

NOTE: *This command can be performed only one time on same device because command writes eFUSE that are One Time Programmable values.*

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

For other information please refer to specific Reference Manual of the PSoC6 device.
If you want to use the command `#TPCMD TRANSITION_TO_SECURE` *remember to execute it as last operation before the* `#TPCMD DISCONNECT` *command.*

#TPCMD READ_EFUSE_BYTE

Syntax: `#TPCMD READ_EFUSE_BYTE <Fuse byte [0-127]>`

Prerequisites: none

Description: This command allows to read a specific byte on the eFUSE memory area.

#TPCMD RUN

Syntax: `#TPCMD RUN <Time [s]>`

<Time [s]>	Time in seconds (i.e., 2 s). This time is an optional parameter.

Prerequisites: none

Description: Move the Reset line up and down quickly if no parameter <Time [s]> is inserted.
`#TPCMD RUN <Time [s]>` instead moves the Reset line down and high, waits for the entered time.
This command typically can be used to execute the firmware programmed in the device.

#TPCMD READ_MEM8

Syntax: `#TPCMD READ_MEM8 <Address> <Byte Count>`

<Address>	Address in HEX format (i.e., 0x52002020)
<Byte Count>	Byte count in decimal format (i.e., 8 -> eight bytes)

Prerequisites: none

Description: Read memory byte per byte from target PSoC6 device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM8 0x52002020 8
Read[0x52002020]: 0xF0
Read[0x52002021]: 0xAA
Read[0x52002022]: 0x16
Read[0x52002023]: 0x14
Read[0x52002024]: 0x00
Read[0x52002025]: 0x00
Read[0x52002026]: 0x00
Read[0x52002027]: 0x00
Time for Read Mem: 0.002 s
```

#TPCMD READ_MEM16

Syntax: `#TPCMD READ_MEM16 <Address> <16-bit Word Count>`

<Address>	Address in HEX format (i.e., 0x52002020)
<16-bit Word Count>	16-bit Word count in decimal format (i.e., 4 -> four 16-bit words)

Prerequisites: none

Description: Read memory 16-bit word per 16-bit word from target PSoC6 device

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM16 0x52002020 4
Read[0x52002020]: 0xAAF0
Read[0x52002022]: 0x1416
Read[0x52002024]: 0x0000
Read[0x52002026]: 0x0000
Time for Read Mem: 0.002 s
```

#TPCMD READ_MEM32

Syntax: #TPCMD READ_MEM32 <Address> <32-bit Word Count>

<Address>	Address in HEX format (i.e., 0x52002020)
<32-bit Word Count>	32-bit Word count in decimal format (i.e., 2 -> two 32-bit words)

Prerequisites: none

Description: Read memory 32-bit word per 32-bit word from target PSoC6 device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM32 0x52002020 2
Read[0x52002020]: 0x1416AAF0
Read[0x52002024]: 0x00000000
Time for Read Mem: 0.002 s
```

#TPCMD DISCONNECT

#TPCMD DISCONNECT

Disconnect function. Power off and exit.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

SMH Technologies®

INFINEON PSoC6 Driver Examples

Here you can see a complete example of INFINEON PSoC6 projects.

1 – INFINEON PSoC6 2.00 MB example Commands

```
#TCSETPAR ACQUIRING_SEQUENCE ACQUIRE_CHIP
#TCSETPAR ENTRY_CLOCK 1000000
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR VPROG0 2500
#TCSETPAR CMODE SWD
#TPSETSRC 2_00MB.frb
#TPSTART
#TPCMD CONNECT
#IFERR TPCMD BLANKCHECK F
#THEN TPCMD MASSERASE F
#THEN TPCMD BLANKCHECK F
#TPCMD PROGRAM F
#TPCMD VERIFY F R
#TPCMD VERIFY F S
#TPCMD DISCONNECT
#TPEND
```

1 – INFINEON PSoC6 2.00 MB example Real Time Log

```
---#TPSTART
Load SWD FPGA version 0x00001215.
Detected PSoC6 device: CY8C6xxA.
Selected PSoC6 Acquire Sequence.
>|
---#TPCMD CONNECT
Protocol selected SWD.
Toggling XRES pin to execute Acquire Chip procedure.
ID-Code read correctly at 1.00 MHz after 8 retries.
JTAG-SWD Debug Port enabled.
Move PSoC6 internal state to Test Mode.
PSoC6 enter into Test Mode.
Scanning AP map to find all APs:
 * AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[1] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[2] IDR: 0x24770011, Type: AMBA AHB3 bus.
Scanning AP to find all cores:
 * AP[1] Found Cortex M0+ revision r0p1.
         CPUID: 0x410CC601.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xF0000000.
 * AP[2] Found Cortex M4 revision r0p1.
         CPUID: 0x410FC241.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xE00FF000.
Try to halt the Cortex M0+ core:
 * AP[1] Cortex M0+ Core halted [0.002 s].
Try to halt the Cortex M4 core:
 * AP[2] Cortex M4 Core halted [0.001 s].
Try to execute the Acquire Chip method procedure:
 * AP[1] Cortex M0+ core Vector Table base 0xFFFF0000.
 * Vector Table value means that the Flash is empty or TOC is corrupted.
 * Acquire Chip method procedure completed.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 5-7].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Read device informations:
 * FamilyId: 0x0102.
 * Revision: 0x12.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
 * SiliconID: 0xE453.
 * Protect state: 0x2: Virgin - [NORMAL] - Secure - Dead.
 * Life cycle stage: 0x1 - Normal.
 * Flash boot version: 0xA301.
 * SROM firmware version: 0x0701.
Read device unique ID:
 * Unique ID0: 0x008C7902, Unique ID1: 0xAF1C1C01, Unique ID2: 0x7A040601.
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
 * Disabled Direct Execute system call functionality.
> Check PSoC6 Silicon ID passed from SMH database [0xE453].
 * Check PSoC6 Silicon ID from source file not available.
Time for Connect: 0.155 s.
>|
---#IFERR TPCMD BLANKCHECK F
Start Blankcheck operation.
Time for Blankcheck F: 0.108 s.
>|
---#TPCMD PROGRAM F
Start Program operation.
Time for Program F: 13.274 s.
>|
---#TPCMD VERIFY F R
Start Verify Readout operation.
Time for Verify Readout F: 0.822 s.
>|
---#TPCMD VERIFY F S
Start Verify Checksum 32bit operation.
Time for Verify Checksum 32bit F: 0.077 s.
>|
---#TPCMD DISCONNECT
>|
```

1 – INFINEON PSoC6 2.00 MB example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.155 s
Conditional Blankcheck Flash	0.118 s
Program Flash	13.274 s
Verify Readout Flash	0.822 s
Verify Checksum Flash	0.077 s
Cycle Time	**00:14.490 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

2 – INFINEON PSoC6 1.00 MB example Commands

```
#TCSETPAR ACQUIRING_SEQUENCE ACQUIRE_CHIP
#TCSETPAR ENTRY_CLOCK 1000000
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE SWD
#TPSETSRC 1MB.frb
#TPSTART
#TPCMD CONNECT
#TPCMD MASSERASE F
#TPCMD BLANKCHECK F
#TPCMD PROGRAM F
#TPCMD VERIFY F R
#TPCMD VERIFY F S
#TPCMD DISCONNECT
#TPEND
```

2 – INFINEON PSoC6 1.00 MB example Real Time Log

```
---#TPSTART
Load SWD FPGA version 0x00001215.
Detected PSoC6 device: CY8C6xx8.
Selected PSoC6 Acquire Sequence.
>|
---#TPCMD CONNECT
Protocol selected SWD.
Toggling XRES pin to execute Acquire Chip procedure.
ID-Code read correctly at 1.00 MHz after 9 retries.
JTAG-SWD Debug Port enabled.
Move PSoC6 internal state to Test Mode.
PSoC6 enter into Test Mode.
Scanning AP map to find all APs:
 * AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[1] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[2] IDR: 0x24770011, Type: AMBA AHB3 bus.
Scanning AP to find all cores:
 * AP[1] Found Cortex M0+ revision r0p1.
         CPUID: 0x410CC601.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xF0000000.
 * AP[2] Found Cortex M4 revision r0p1.
         CPUID: 0x410FC241.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xE00FF000.
Try to halt the Cortex M0+ core:
 * AP[1] Cortex M0+ Core halted [0.002 s].
Try to halt the Cortex M4 core:
 * AP[2] Cortex M4 Core halted [0.001 s].
Try to execute the Acquire Chip method procedure:
 * AP[1] Cortex M0+ core Vector Table base 0x10000000.
 * AP[1] Cortex M0+ core Reset Address 0x10000183.
 * Set specific software breakpoint 0xD0000181.
 * Trigger a software reset to restart CPU.
 * Reconnect and waiting for CPU to hit the breakpoint:
 * Breakpoint software used correctly. Program Counter value is 0x10000182.
 * Acquire Chip method procedure completed.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 4 [Range 3-6].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Read device informations:
 * FamilyId: 0x0102.
 * Revision: 0x12.
 * SiliconID: 0xE457.
 * Protect state: 0x2: Virgin - [NORMAL] - Secure - Dead.
 * Life cycle stage: 0x1 - Normal.
 * Flash boot version: 0xA301.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
 * SROM firmware version: 0x0701.
Read device unique ID:
 * Unique ID0: 0x008CCB31, Unique ID1: 0xAF323D12, Unique ID2: 0x7A081D01.
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
 * Disabled Direct Execute system call functionality.
> Check PSoC6 Silicon ID passed from SMH database [0xE457].
FRB CRC32 check passed.
FRB Headers collected.
 * Check PSoC6 Silicon ID from source file not available.
Time for Connect: 0.226 s.
>|
---#TPCMD MASSERASE F
Start Masserase operation.
Time for Masserase F: 0.035 s.
>|
---#TPCMD BLANKCHECK F
Start Blankcheck operation.
Time for Blankcheck F: 0.054 s.
>|
---#TPCMD PROGRAM F
Start Program operation.
Time for Program F: 7.047 s.
>|
---#TPCMD VERIFY F R
Start Verify Readout operation.
Time for Verify Readout F: 0.405 s.
>|
---#TPCMD VERIFY F S
Start Verify Checksum 32bit operation.
Time for Verify Checksum 32bit F: 0.040 s.
>|
---#TPCMD DISCONNECT
>|
```

2 – INFINEON PSoC6 1.00 MB example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.153 s
Masserase Flash	0.035 s
Blankcheck Flash	0.054 s
Program Flash	7.047 s
Verify Readout Flash	0.405 s
Verify Checksum Flash	0.040 s
Cycle Time	**00:07.861 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

3 – INFINEON PSoC6 512 KB example Commands

```
#TCSETPAR ACQUIRING_SEQUENCE ACQUIRE_CHIP
#TCSETPAR ENTRY_CLOCK 1000000
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE SWD
#TPSETSRC 512KB.frb
#TPSTART
#TPCMD CONNECT
#TPCMD MASSERASE F
#TPCMD BLANKCHECK F
#TPCMD PROGRAM F
#TPCMD VERIFY F R
#TPCMD VERIFY F S
#TPCMD DISCONNECT
#TPEND
```

3 – INFINEON PSoC6 512 KB example Real Time Log

```
---#TPSTART
Load SWD FPGA version 0x00001215.
Detected PSoC6 device: CY8C6xx5.
Selected PSoC6 Acquire Sequence.
>|
---#TPCMD CONNECT
Protocol selected SWD.
Toggling XRES pin to execute Acquire Chip procedure.
ID-Code read correctly at 1.00 MHz after 6 retries.
JTAG-SWD Debug Port enabled.
Move PSoC6 internal state to Test Mode.
PSoC6 enter into Test Mode.
Scanning AP map to find all APs:
 * AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[1] IDR: 0x84770001, Type: AMBA AHB3 bus.
 * AP[2] IDR: 0x24770011, Type: AMBA AHB3 bus.
Scanning AP to find all cores:
 * AP[2] Found Cortex M4 revision r0p1.
         CPUID: 0x410FC241.
         Implementer Code: 0x41 - [ARM].
         ROM table base address 0xE00FF000.
Try to halt the Cortex M4 core:
 * AP[2] Cortex M4 Core halted [0.001 s].
Try to execute the Acquire Chip method procedure:
 * AP[2] Cortex M4 core Vector Table base 0xFFFF0000.
 * Vector Table value means that the Flash is empty or TOC is corrupted.
 * Acquire Chip method procedure completed.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 4 [Range 3-6].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Read device informations:
 * FamilyId: 0x0105.
 * Revision: 0x12.
 * SiliconID: 0xE717.
 * Protect state: 0x2: Virgin - [NORMAL] - Secure - Dead.
 * Life cycle stage: 0x1 - Normal.
 * Flash boot version: 0xA301.
 * SROM firmware version: 0x0701.
Read device unique ID:
 * Unique ID0: 0x008CCB32, Unique ID1: 0xAF523E02, Unique ID2: 0x7A081801.
Normal device access restrictions:
 * Enabled Cortex M0-AP access.
 * Enabled Cortex M4 AP access.
 * Enabled system AP access.
 * Entire Supervisory Flash main region is accessible.
 * Entire Flash main region is accessible.
 * Entire SRAM main region is accessible.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
 * Disabled Direct Execute system call functionality.
> Check PSoC6 Silicon ID passed from SMH database [0xE717].
 * Check PSoC6 Silicon ID from source file not available.
Time for Connect: 0.151 s.
>|
---#TPCMD MASSERASE F
Start Masserase operation.
Time for Masserase F: 0.035 s.
>|
---#TPCMD BLANKCHECK F
Start Blankcheck operation.
Time for Blankcheck F: 0.027 s.
>|
---#TPCMD PROGRAM F
Start Program operation.
Time for Program F: 3.231 s.
>|
---#TPCMD VERIFY F R
Start Verify Readout operation.
Time for Verify Readout F: 0.203 s.
>|
---#TPCMD VERIFY F S
Start Verify Checksum 32bit operation.
Time for Verify Checksum 32bit F: 0.021 s.
>|
---#TPCMD DISCONNECT
>|
```

3 – INFINEON PSoC6 512 KB example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.151 s
Masserase Flash	0.035 s
Blankcheck Flash	0.027 s
Program Flash	3.231 s
Verify Readout Flash	0.203 s
Verify Checksum Flash	0.021 s
Cycle Time	**00:03.721 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

INFINEON PSoC6 Driver Changelog

Info about driver version 5.00 - 19/12/2023
First driver version for PSoC6 devices.

SMH Technologies®
SMH Technologies S.r.l.
UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING
HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale
Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021
→smh-tech.com
info@smh-tech.com