

---

# TROPIC01

ODU\_TR01 - User API

Version: 1.4.0

Git tag:

Tropic Square  
November 11, 2025





## Version history

Version Tag	Date	Author	Description
0.1	9.1.2023	Ondrej Ille	Initial API version.
0.2	26.1.2023	Ondrej Ille	Add <b>CFG_START_UP</b> CO.
0.3	13.3.2023	Ondrej Ille	Add <b>DATA_IN*</b> fields to <b>Ping</b> . Add ranges to field sizes. Change <b>Get_Serial_Code</b> to <b>Serial_Code_Get</b> .
0.4	28.3.2023	Ondrej Ille	Fix <b>Get_Info_Req</b> chunk size. Fix <b>R_Mem_Data_*</b> command size to 444 Bytes. Add padding to <b>ECDSA_Sign</b> , <b>EDDSA_Sign</b> , <b>Attest_Key_*</b> and <b>MAC_And_Destroy</b> . Update <b>CMD_ID</b> values to be non-linear. Add <b>Attest_Key_Read</b> L3 Command Definition. Add <b>CFG_UAP_ATTEST_KEY_READ</b> CO. Change addressing of COs to be non-linear and to correspond to order of <b>CMD_ID</b> fields.
0.5	18.4.2023	Ondrej Ille	Use enumerated values with bullets for possible values of protocol fields.
0.6	19.4.2023	Ondrej Ille	Rename Attestation Keys to ECC Keys. Rename related L3 commands and COs.
0.7	28.4.2023	Ondrej Ille	Add <b>ECC_Key_Erase</b> and <b>CFG_UAP_ECC_KEY_ERASE</b> .
0.8	16.5.2023	Prasoon Dwivedi	Fix <b>Encrypted_Cmd_Abt</b> options. Fix <b>CFG_UAP_ECC_KEY_ERASE</b> CO fields.
0.9	24.5.2023	Henri L'Hote	Add missing <b>SLOT_EXPIRED</b> to <b>R_Mem_Data_Write</b> . Typo fixes.
0.10	19.6.2023	Henri L'Hote	Removed <b>UDATA_LEN</b> from <b>R_Mem_Data_Read</b> .
0.11	26.6.2023	Ondrej Ille	Change CO addresses so that functional COs and configuration COs are in contiguous address regions. Change <b>ADDRESS</b> of L3 Commands that modify config to two bytes.
0.12	27.7.2023	Candice Lam	Grammar check. Consistency fix.
0.13	15.9.2023	Jarda Hrabalek	Add start-up specific commands.



Version Tag	Date	Author	Description
0.14	18.9.2023	Ondrej Ille	Remove <b>CFG_ALARM_MODE</b> CO. Change polarity of bits in <b>CFG_START_UP</b> . Remove <b>CFG_STARTUP[MBIST]</b> .
0.15	1.2.2024	Ondrej Ille	Add <b>CFG_STARTUP[MBIST_DIS]</b> , <b>CFG_STARTUP[RNGTEST_DIS]</b> , <b>CFG_STARTUP[MAINTENANCE_ENA]</b> , <b>CFG_STARTUP[CPU_FW_VERIFY_DIS]</b> and <b>CFG_STARTUP[SPECT_FW_VERIFY_DIS]</b> .
0.16	6.2.2024	Candice Lam	Grammar check. Consistency fix.
0.17	1.3.2024	Ondrej Ille	Add <b>SLEEP_KIND=DEEP_</b> SLEEP_MODE. Add <b>CFG_SLEEP_MODE[DEEP_SLEEP_MODE_EN]</b> CO. Encode <b>SLEEP_KIND</b> more meaningfully.
0.18	7.3.2024	Ondrej Ille	Rework <b>CFG_SENSORS</b> to the latest state of Alarms. Flip its polarity.
0.19	14.3.2024	Ondrej Ille	Add <b>CFG_DEBUG</b> CO. And <b>Get_Log_Req</b> .
0.20	26.3.2024	Ondrej Ille	Clarify <b>PKEY_INDEX</b> starts from 0. Change COs that refer to Pairing Key Slots to be indexed from 0.
0.21	3.5.2024	Ondrej Ille	Extend <b>Ping</b> size to 4096 bytes.
0.22	15.5.2024	Adam Vrba Ondrej Ille	Modify Slot Numbering to be consistently from 0. Add <b>Pairing_Key_Invalidate</b> . Add <b>CFG_UAP_PAIRING_KEY_INVALIDATE</b> .
0.23	15.5.2024	Ondrej Ille	Swap "CFG" and "FUNC" in <b>CFG_(R I)_CONFIG_*</b> COs. For <b>CFG_R_CONFIG_ERASE</b> remove split completely.
0.24	13.6.2024	Adam Vrba	Add padding to all L3 Commands / Results. Rename <b>Encrypted_Cmd_Abt</b> to <b>Encrypted_Session_Abt</b>
0.25	28.8.2024	Ondrej Ille	Add <b>CFG_START_UP[RFU_1]</b> bit.
1.0	4.10.2024	Jarda Hrabalek	Change L2 API for secured FW update. Changed commands <b>Mutable_FW_Update*</b>
1.0.1	12.11.2024	Jarda Hrabalek	Update L2 API FW header structure.



Version Tag	Date	Author	Description
1.0.2	18.11.2024	Adam Vrba	Remove <b>CPU_FW_VERIFY_DIS</b> and <b>SPECT_FW_VERIFY_DIS</b> fields from <b>CFG_START_UP</b> .
1.0.3	26.11.2024	Jarda Hrabalek	Update API <b><i>Get_Info_Req</i></b>
1.0.4	5.12.2024	Ondrej Ille	Remove <b>CFG_UAP_SERIAL_CODE_GET</b> .
1.1.0	11.12.2024	Adam Vrba	Split the API to bootloader and application parts.
1.1.2	21.2.2025	Olha Harielina	Remove DEEP_SLEEP_MODE from L2 API.
1.2.0	11.4.2025	Adam Vrba	Add GPO pin function modes <b>CFG_GPO</b>
1.3.0	18.6.2025	Adam Vrba	Change minimum size of EdDSA sign MSG field to 0.
1.4.0	10.11.2025	Adam Vrba	Raise User Data slot size from 444 Bytes to 475 Bytes. Add <b>HARDWARE_FAIL</b> result code to <b><i>Pairing_Key_Write</i></b> , <b><i>Pairing_Key_Invalidate</i></b> , <b><i>(I/R)_Config_Write</i></b> and <b><i>R_Mem_Data_Write</i></b> L3 API commands. Rename <b>PAIRING_KEY_EMPTY</b> to <b>SLOT_EMPTY</b> and <b>PAIRING_KEY_INVALID</b> to <b>SLOT_INVALID</b> for <b><i>Pairing_Key_Read</i></b> . Changed <b>SLOT_NOT_EMPTY</b> to <b>WRITE_FAIL</b> for <b><i>R_Mem_Data_Write</i></b> .



---

## Contents

<b>1</b>	<b>Glossary</b>	<b>5</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
<b>3</b>	<b>Bootloader API</b>	<b>7</b>
<b>4</b>	<b>Application API</b>	<b>15</b>
4.1	L2 Request / Response frames . . . . .	15
4.2	L3 Commands / Result packets . . . . .	23
<b>5</b>	<b>User Configuration Objects</b>	<b>47</b>
5.1	Bootloader . . . . .	47
5.2	Application . . . . .	53
<b>6</b>	<b>Open Issues</b>	<b>66</b>



# 1 Glossary

- **API** : Application Processing Interface
- **CO** : Configuration Object
- **CRC** : Cyclic Redundancy Check
- **EdDSA** : Edwards Curve Digital Signature Algorithm
- **ECDSA** : Elliptic Curve Digital Signature Algorithm
- **FW** : Firmware
- **I-Config** : Irreversible Config
- **MCU** : Microcontroller
- **R-Config** : Reversible Config
- **ROM** : Read Only Memory



## 2 Introduction

This document describes TROPIC01's API:

- L2 Layer communication unit definitions - Request and Response frames
- L3 Layer communication unit definitions - Command and Result packets
- Configuration Objects (CO) - The memory layout of the Reversible Config (R-Config) and Irreversible Config (I-Config)

### Note

Each CO has a single address.

### Note

Tropic Square might write bits in I-Config COs during manufacturing. As a result, TROPIC01 might provide limited configuration options.

### Note

To read the L2 Response frame, Host MCU issues L2 Request frame with **REQ\_ID == *Get\_Response* = 0xAA**. For detailed information about the L2 communication layer, refer to Datasheet.



## 3 Bootloader API

Parameter	Description
<b>Information</b>	
<b>Name</b>	<i>Get_Info_Req</i>
<b>Description</b>	Request to obtain information about TROPIC01.
<b>API function name</b>	get_info_req
<b>Request</b>	
<b>REQ_ID</b>	0x01
<b>REQ_LEN</b>	0x02
<b>REQ_DATA</b>	(length: 2 byte(s))
<b>OBJECT_ID</b>	
<b>Description</b>	The Identifier of the requested object.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>X509_CERTIFICATE</b> (0x00): The X.509 Certificate Store (object size 3840B).</li><li>• <b>CHIP_ID</b> (0x01): The chip ID - the chip silicon revision and unique device ID (object size 128B).</li><li>• <b>RISCV_FW_VERSION</b> (0x02): Current version of Immutable FW (object size 4B)</li><li>• <b>SPECT_FW_VERSION</b> (0x04): The SPECT bootloader is a part of RISC-V bootloader. Returns dummy value. (object size 4B)</li><li>• <b>FW_BANK</b> (0xb0): The FW header read from the FW bank defined by BLOCK_INDEX choices (object size 128B).</li></ul>
<b>BLOCK_INDEX</b>	
<b>Description</b>	X509_CERTIFICATE: index of the 128B certificate block 0 - 29 FW_BANK: enumeration of possible values defined for this field CHIP_ID, *_FW_VERSION: do not care
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>CPU_BANK_1</b> (0x01): RISC-V FW bank 1</li><li>• <b>CPU_BANK_2</b> (0x02): RISC-V FW bank 2</li><li>• <b>SPECT_BANK_1</b> (0x11): SPECT FW bank 1</li><li>• <b>SPECT_BANK_2</b> (0x12): RISC-V FW bank 2</li></ul>
<b>REQ_CRC</b>	(length: 2 bytes)
<b>Response</b>	
<b>RSP_LEN</b>	0x01 - 0x80
<b>RSP_DATA</b>	(length: 1 - 128 byte(s))
<b>OBJECT</b>	





<b>Description</b>	The data content of the requested object block.
<b>Size</b>	1 - 128
<b>RSP_CRC</b>	(length: 2 bytes)

Table 1: Get\_Info\_Req syntax



Parameter	Description
Information	
Name	<b><i>Resend_Req</i></b>
Description	Request for TROPIC01 to resend the last L2 Response.
API function name	resend_req
Request	
REQ_ID	0x10
REQ_LEN	0x00
REQ_DATA	(length: 0 byte(s))
REQ_CRC	(length: 2 bytes)
Response	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 2: Resend\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Startup_Req</i></b>
<b>Description</b>	Request for TROPIC01 to reset.
<b>API function name</b>	startup_req
<b>Request</b>	
REQ_ID	0xb3
REQ_LEN	0x01
REQ_DATA	(length: 1 byte(s))
<b>STARTUP_ID</b>	
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>REBOOT</b> (0x01): Restart, then initialize as if a power-cycle was applied.</li><li>• <b>MAINTENANCE_REBOOT</b> (0x03): Restart, then initialize. Stay in Start-up mode and do not load the mutable FW from R-Memory.</li></ul>
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 3: Startup\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Mutable_FW_Update_Req</i></b>
<b>Description</b>	Request to start updating mutable FW. Supported only in Start-up mode (i.e. after Startup_Req with MAINTENANCE_REBOOT). Possible update only same or newer version.  NOTE: Chip automatically select memory space for FW storage and erase it.
<b>API function name</b>	mutable_fw_update_req
<b>Request</b>	
<b>REQ_ID</b>	0xb0
<b>REQ_LEN</b>	0x68
<b>REQ_DATA</b>	(length: 104 byte(s))
<b>SIGNATURE</b>	
<b>Description</b>	Signature of SHA256 hash of all following data in this packet.
<b>Size</b>	64
<b>HASH</b>	
<b>Description</b>	SHA256 HASH of first FW chunk of data sent using Mutable_FW_Update_Data.
<b>Size</b>	32
<b>TYPE</b>	
<b>Description</b>	FW type which is going to be updated.
<b>Size</b>	2
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>FW_TYPE_CPU</b> (0x01): FW for RISC-V main CPU.</li><li>• <b>FW_TYPE_SPECT</b> (0x02): FW for SPECT coprocessor.</li></ul>
<b>PADDING</b>	
<b>Description</b>	Zero value.
<b>Size</b>	1
<b>HEADER_VERSION</b>	
<b>Description</b>	Current value is 1.
<b>Size</b>	1
<b>VERSION</b>	
<b>Size</b>	4
<b>REQ_CRC</b>	(length: 2 bytes)
<b>Response</b>	
<b>RSP_LEN</b>	0x00
<b>RSP_DATA</b>	(length: 0 byte(s))



RSP_CRC	(length: 2 bytes)
---------	-------------------

Table 4: Mutable\_FW\_Update\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Mutable_FW_Update_Data_Req</i></b>
<b>Description</b>	Request to write a chunk of the new mutable FW to a R-Memory bank. Supported only in Start-up mode after Mutable_FW_Update_Req successfully processed.
<b>API function name</b>	mutable_fw_update_data_req
<b>Request</b>	
<b>REQ_ID</b>	0xb1
<b>REQ_LEN</b>	0x26 - 0xfa
<b>REQ_DATA</b>	(length: 38 - 250 byte(s))
<b>HASH</b>	
<b>Description</b>	SHA256 HASH of the next FW chunk of data sent using Mutable_FW_Update_Data.
<b>Size</b>	32
<b>OFFSET</b>	
<b>Description</b>	The offset of the specific bank to write the FW chunk data to.
<b>Size</b>	2
<b>DATA</b>	
<b>Description</b>	The binary data to write. Data size should be a multiple of 4.
<b>Size</b>	4 - 216
<b>REQ_CRC</b>	(length: 2 bytes)
<b>Response</b>	
<b>RSP_LEN</b>	0x00
<b>RSP_DATA</b>	(length: 0 byte(s))
<b>RSP_CRC</b>	(length: 2 bytes)

Table 5: Mutable\_FW\_Update\_Data\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Get_Log_Req</i></b>
<b>Description</b>	Get debug log message (for internal development purpose only). Note: Logging is irreversibly disabled during provisioning.
<b>API function name</b>	get_log_req
<b>Request</b>	
REQ_ID	0xa2
REQ_LEN	0x00
REQ_DATA	(length: 0 byte(s))
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00 - 0xfc
RSP_DATA	(length: 0 - 252 byte(s))
<b>LOG_MSG</b>	
<b>Description</b>	Debug log message
<b>Size</b>	0 - 252
RSP_CRC	(length: 2 bytes)

Table 6: Get\_Log\_Req syntax



## 4 Application API

### 4.1 L2 Request / Response frames

Parameter	Description
<b>Information</b>	
<b>Name</b>	<i><b>Get_Info_Req</b></i>
<b>Description</b>	Request to obtain information about TROPIC01.
<b>API function name</b>	get_info_req
<b>Request</b>	
<b>REQ_ID</b>	0x01
<b>REQ_LEN</b>	0x02
<b>REQ_DATA</b>	(length: 2 byte(s))
<b>OBJECT_ID</b>	
<b>Description</b>	The Identifier of the requested object.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>X509_CERTIFICATE</b> (0x00): The X.509 Certificate Store (object size 3840B).</li><li>• <b>CHIP_ID</b> (0x01): The chip ID - the chip silicon revision and unique device ID (object size 128B).</li><li>• <b>RISCV_FW_VERSION</b> (0x02): Current version of RISC-V FW (object size 4B)</li><li>• <b>SPECT_FW_VERSION</b> (0x04): Current version of SPECT FW (object size 4B)</li></ul>
<b>BLOCK_INDEX</b>	
<b>Description</b>	X509_CERTIFICATE: index of the 128B certificate block 0 - 29 CHIP_ID, *_FW_VERSION: do not care
<b>Size</b>	1
<b>REQ_CRC</b>	(length: 2 bytes)
<b>Response</b>	
<b>RSP_LEN</b>	0x01 - 0x80
<b>RSP_DATA</b>	(length: 1 - 128 byte(s))
<b>OBJECT</b>	
<b>Description</b>	The data content of the requested object block.
<b>Size</b>	1 - 128
<b>RSP_CRC</b>	(length: 2 bytes)

Table 7: Get\_Info\_Req syntax





Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Handshake_Req</i></b>
<b>Description</b>	Request to execute a Secure Channel Handshake and establish a new Secure Channel Session (TROPIC01 moves to Secure Channel Mode).
<b>API function name</b>	handshake_req
<b>Request</b>	
REQ_ID	0x02
REQ_LEN	0x21
REQ_DATA	(length: 33 byte(s))
<b>E_HPUB</b>	
<b>Description</b>	The Host MCU's Ephemeral X25519 public key. A little endian encoding of the x-coordinate from the public Curve25519 point.
<b>Size</b>	32
<b>PKEY_INDEX</b>	
<b>Description</b>	The index of the Pairing Key slot to establish a Secure Channel Session with (TROPIC01 fetches $S_{HiPub}$ from the Pairing Key slot specified in this field).
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>PAIRING_KEY_SLOT_0</b> (0x00): Corresponds to <math>S_{H0Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_1</b> (0x01): Corresponds to <math>S_{H1Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_2</b> (0x02): Corresponds to <math>S_{H2Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_3</b> (0x03): Corresponds to <math>S_{H3Pub}</math>.</li></ul>
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x30
RSP_DATA	(length: 48 byte(s))
<b>E_TPUB</b>	
<b>Description</b>	TROPIC01's X25519 Ephemeral key.
<b>Size</b>	32
<b>T_TAUTH</b>	
<b>Description</b>	The Secure Channel Handshake Authentication Tag.
<b>Size</b>	16
RSP_CRC	(length: 2 bytes)

Table 8: Handshake\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Encrypted_Cmd_Req</i></b>
<b>Description</b>	Request to execute an L3 Command.
<b>API function name</b>	encrypted_cmd_req
<b>Request</b>	
REQ_ID	0x04
REQ_LEN	0x01 - 0xfc
REQ_DATA	(length: 1 - 252 byte(s))
<b>L3_CHUNK</b>	
<b>Description</b>	The encrypted L3 command or a chunk of it.
<b>Size</b>	1 - 252
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x01 - 0xfc
RSP_DATA	(length: 1 - 252 byte(s))
<b>L3_CHUNK</b>	
<b>Description</b>	The encrypted L3 result or a chunk of it.
<b>Size</b>	1 - 252
RSP_CRC	(length: 2 bytes)

Table 9: Encrypted\_Cmd\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Encrypted_Session_Abt_Req</i></b>
<b>Description</b>	Request to abort current Secure Channel Session and execution of L3 command (TROPIC01 moves to Idle Mode).
<b>API function name</b>	encrypted_session_abt_req
<b>Request</b>	
REQ_ID	0x08
REQ_LEN	0x00
REQ_DATA	(length: 0 byte(s))
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 10: Encrypted\_Session\_Abt\_Req syntax



Parameter	Description
Information	
Name	<b><i>Resend_Req</i></b>
Description	Request for TROPIC01 to resend the last L2 Response.
API function name	resend_req
Request	
REQ_ID	0x10
REQ_LEN	0x00
REQ_DATA	(length: 0 byte(s))
REQ_CRC	(length: 2 bytes)
Response	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 11: Resend\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Sleep_Req</i></b>
<b>Description</b>	Request for TROPIC01 to go to Sleep Mode.
<b>API function name</b>	sleep_req
<b>Request</b>	
REQ_ID	0x20
REQ_LEN	0x01
REQ_DATA	(length: 1 byte(s))
<b>SLEEP_KIND</b>	
<b>Description</b>	The type of Sleep mode TROPIC01 moves to.
<b>Size</b>	1
<b>Possible values</b>	• <b>SLEEP_MODE</b> (0x05): Sleep Mode
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 12: Sleep\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Startup_Req</i></b>
<b>Description</b>	Request for TROPIC01 to reset.
<b>API function name</b>	startup_req
<b>Request</b>	
REQ_ID	0xb3
REQ_LEN	0x01
REQ_DATA	(length: 1 byte(s))
<b>STARTUP_ID</b>	
Size	1
Possible values	<ul style="list-style-type: none"><li>• <b>REBOOT</b> (0x01): Restart, then initialize as if a power-cycle was applied.</li><li>• <b>MAINTENANCE_REBOOT</b> (0x03): Restart, then initialize. Stay in Start-up mode and do not load the mutable FW from R-Memory.</li></ul>
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00
RSP_DATA	(length: 0 byte(s))
RSP_CRC	(length: 2 bytes)

Table 13: Startup\_Req syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Get_Log_Req</i></b>
<b>Description</b>	Get debug log message (for internal development purpose only). Note: Logging is irreversibly disabled in CFG_DEBUG[FW_-LOG_EN] during provisioning.
<b>API function name</b>	get_log_req
<b>Request</b>	
REQ_ID	0xa2
REQ_LEN	0x00
REQ_DATA	(length: 0 byte(s))
REQ_CRC	(length: 2 bytes)
<b>Response</b>	
RSP_LEN	0x00 - 0xfc
RSP_DATA	(length: 0 - 252 byte(s))
<b>LOG_MSG</b>	
<b>Description</b>	Debug log message
<b>Size</b>	0 - 252
RSP_CRC	(length: 2 bytes)

Table 14: Get\_Log\_Req syntax



## 4.2 L3 Commands / Result packets

Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Ping</i></b>
<b>Description</b>	A dummy command to check the Secure Channel Session communication.
<b>API function name</b>	ping
<b>Command</b>	
<b>CMD_SIZE</b>	0x01 - 0x1001
<b>CMD_ID</b>	0x01
<b>CMD_DATA</b>	(length: 0 - 4096 byte(s))
<b>DATA_IN</b>	
<b>Description</b>	The input data
<b>Size</b>	0 - 4096
<b>Result</b>	
<b>RES_SIZE</b>	0x01 - 0x1001
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 - 4096 byte(s))
<b>DATA_OUT</b>	
<b>Description</b>	The output data (loopback of the <b>DATA_IN</b> field).
<b>Size</b>	0 - 4096

Table 15: Ping syntax





Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Pairing_Key_Write</i></b>
<b>Description</b>	Command to write the X25519 public key to a Pairing Key slot.
<b>API function name</b>	pairing_key_write
<b>Command</b>	
<b>CMD_SIZE</b>	0x24
<b>CMD_ID</b>	0x10
<b>CMD_DATA</b>	(length: 35 byte(s))
<b>SLOT</b>	
<b>Description</b>	The Pairing Key slot. Valid values are 0 - 3.
<b>Size</b>	2
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>PAIRING_KEY_SLOT_0</b> (0x00): Corresponds to <math>S_{H0Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_1</b> (0x01): Corresponds to <math>S_{H1Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_2</b> (0x02): Corresponds to <math>S_{H2Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_3</b> (0x03): Corresponds to <math>S_{H3Pub}</math>.</li></ul>
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	1
<b>S_HIPUB</b>	
<b>Description</b>	The X25519 public key to be written in the Pairing Key slot specified in the SLOT field.
<b>Size</b>	32
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>HARDWARE_FAIL</b> (0x17): In case an HW error occurred during writing to OTP and the slot was invalidated and permanently lost.</li></ul>
<b>RES_DATA</b>	(length: 0 byte(s))

Table 16: Pairing\_Key\_Write syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Pairing_Key_Read</i></b>
<b>Description</b>	Command to read the X25519 public key from a Pairing Key slot.
<b>API function name</b>	pairing_key_read
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x11
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>SLOT</b>	
<b>Description</b>	The Pairing Key slot. Valid values are 0 - 3.
<b>Size</b>	2
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>PAIRING_KEY_SLOT_0</b> (0x00): Corresponds to <math>S_{H0Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_1</b> (0x01): Corresponds to <math>S_{H1Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_2</b> (0x02): Corresponds to <math>S_{H2Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_3</b> (0x03): Corresponds to <math>S_{H3Pub}</math>.</li></ul>
<b>Result</b>	
<b>RES_SIZE</b>	0x24
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>SLOT_EMPTY</b> (0x15): The requested slot is empty and contains no valid data.</li><li>• <b>SLOT_INVALID</b> (0x16): The Pairing key slot is in "Invalidated" state. The Pairing key has been invalidated.</li></ul>
<b>RES_DATA</b>	(length: 35 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>S_HIPUB</b>	
<b>Description</b>	The X25519 public key to be written in the Pairing Key slot specified in the SLOT field.
<b>Size</b>	32

Table 17: Pairing\_Key\_Read syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Pairing_Key_Invalidate</i></b>
<b>Description</b>	Command to invalidate the X25519 public key in a Pairing Key slot.
<b>API function name</b>	pairing_key_invalidate
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x12
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>SLOT</b>	
<b>Description</b>	The Pairing Key slot. Valid values are 0 - 3.
<b>Size</b>	2
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>PAIRING_KEY_SLOT_0</b> (0x00): Corresponds to <math>S_{H0Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_1</b> (0x01): Corresponds to <math>S_{H1Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_2</b> (0x02): Corresponds to <math>S_{H2Pub}</math>.</li><li>• <b>PAIRING_KEY_SLOT_3</b> (0x03): Corresponds to <math>S_{H3Pub}</math>.</li></ul>
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>HARDWARE_FAIL</b> (0x17): In case of some HW error.</li></ul>
<b>RES_DATA</b>	(length: 0 byte(s))

Table 18: Pairing\_Key\_Invalidate syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Config_Write</i></b>
<b>Description</b>	Command to write a single CO to R-Config.
<b>API function name</b>	r_config_write
<b>Command</b>	
<b>CMD_SIZE</b>	0x08
<b>CMD_ID</b>	0x20
<b>CMD_DATA</b>	(length: 7 byte(s))
<b>ADDRESS</b>	
<b>Description</b>	The CO address offset for TROPIC01 to compute the actual CO address.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	1
<b>VALUE</b>	
<b>Description</b>	The CO value to write in the computed address.
<b>Size</b>	4
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	• <b>HARDWARE_FAIL</b> (0x17): In case of some HW error.
<b>RES_DATA</b>	(length: 0 byte(s))

Table 19: R\_Config\_Write syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Config_Read</i></b>
<b>Description</b>	Command to read a single CO from R-Config.
<b>API function name</b>	r_config_read
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x21
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>ADDRESS</b>	
<b>Description</b>	The CO address offset for TROPIC01 to compute the actual CO address.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x08
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 7 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>VALUE</b>	
<b>Description</b>	The CO value TROPIC01 read from the computed address.
<b>Size</b>	4

Table 20: R\_Config\_Read syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Config_Erase</i></b>
<b>Description</b>	Command to erase the whole R-Config (convert the bits of all CO to 1).
<b>API function name</b>	r_config_erase
<b>Command</b>	
<b>CMD_SIZE</b>	0x01
<b>CMD_ID</b>	0x22
<b>CMD_DATA</b>	(length: 0 byte(s))
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 21: R\_Config\_Erase syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>I_Config_Write</i></b>
<b>Description</b>	Command to write a single bit of CO (from I-Config) from 1 to 0.
<b>API function name</b>	i_config_write
<b>Command</b>	
<b>CMD_SIZE</b>	0x04
<b>CMD_ID</b>	0x30
<b>CMD_DATA</b>	(length: 3 byte(s))
<b>ADDRESS</b>	
<b>Description</b>	The CO address offset for TROPIC01 to compute the actual CO address.
<b>Size</b>	2
<b>BIT_INDEX</b>	
<b>Description</b>	The bit to write from 1 to 0. Valid values are 0-31.
<b>Size</b>	1
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	• <b>HARDWARE_FAIL</b> (0x17): In case an HW error occurred during writing to OTP. This is fatal error and the chip will be switched permanently to ALARM.
<b>RES_DATA</b>	(length: 0 byte(s))

Table 22: I\_Config\_Write syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<i><b>I_Config_Read</b></i>
<b>Description</b>	Command to read a single CO from I-Config.
<b>API function name</b>	i_config_read
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x31
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>ADDRESS</b>	
<b>Description</b>	The CO address offset for TROPIC01 to compute the actual CO address.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x08
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 7 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>VALUE</b>	
<b>Description</b>	The CO value TROPIC01 read from the computed address.
<b>Size</b>	4

Table 23: I\_Config\_Read syntax





Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Mem_Data_Write</i></b>
<b>Description</b>	Command to write general purpose data in a slot from the User Data partition in R-Memory.
<b>API function name</b>	r_mem_data_write
<b>Command</b>	
<b>CMD_SIZE</b>	0x05 - 0x1df
<b>CMD_ID</b>	0x40
<b>CMD_DATA</b>	(length: 4 - 478 byte(s))
<b>UDATA_SLOT</b>	
<b>Description</b>	The slot of the User Data partition. Valid values are 0 - 511.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	1
<b>DATA</b>	
<b>Description</b>	The data stream to be written in the slot specified in the UDATA_SLOT L3 field.
<b>Size</b>	1 - 475
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>SLOT_NOT_EMPTY</b> (0x10): The word is already written.</li><li>• <b>HARDWARE_FAIL</b> (0x17): In case of some HW error.</li></ul>
<b>RES_DATA</b>	(length: 0 byte(s))

Table 24: R\_Mem\_Data\_Write syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Mem_Data_Read</i></b>
<b>Description</b>	Command to read the general purpose data from a slot of the User Data partition in R-Memory.
<b>API function name</b>	r_mem_data_read
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x41
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>UDATA_SLOT</b>	
<b>Description</b>	The slot of the User Data partition. Valid values are 0 - 511.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x04 - 0x1df
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 3 - 478 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>DATA</b>	
<b>Description</b>	The data stream read from the slot specified in the UDATA_SLOT L3 field.
<b>Size</b>	0 - 475

Table 25: R\_Mem\_Data\_Read syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>R_Mem_Data_Erase</i></b>
<b>Description</b>	Command to erase a slot from the User Data partition in R-Memory.
<b>API function name</b>	r_mem_data_erase
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x42
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>UDATA_SLOT</b>	
<b>Description</b>	The slot of the User Data partition. Valid values are 0 - 511.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 26: R\_Mem\_Data\_Erase syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>Random_Value_Get</i></b>
<b>Description</b>	Command to get random numbers generated by TRNG2.
<b>API function name</b>	random_value_get
<b>Command</b>	
<b>CMD_SIZE</b>	0x02
<b>CMD_ID</b>	0x50
<b>CMD_DATA</b>	(length: 1 byte(s))
<b>N_BYTES</b>	
<b>Description</b>	The number of random bytes to get.
<b>Size</b>	1
<b>Result</b>	
<b>RES_SIZE</b>	0x04 - 0x103
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 3 - 258 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>RANDOM_DATA</b>	
<b>Description</b>	The random data from TRNG2 in the number of bytes specified in the <b>N_BYTES</b> field.
<b>Size</b>	0 - 255

Table 27: Random\_Value\_Get syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>ECC_Key_Generate</i></b>
<b>Description</b>	Command to generate an ECC Key and store the key in a slot from the ECC Keys partition in R-Memory.
<b>API function name</b>	ecc_key_generate
<b>Command</b>	
<b>CMD_SIZE</b>	0x04
<b>CMD_ID</b>	0x60
<b>CMD_DATA</b>	(length: 3 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot to write the generated key. Valid values are 0 - 31.
<b>Size</b>	2
<b>CURVE</b>	
<b>Description</b>	The Elliptic Curve the key is generated from.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>P256</b> (0x01): P256 Curve - 64-byte long public key.</li><li>• <b>ED25519</b> (0x02): Ed25519 Curve - 32-byte long public key.</li></ul>
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 28: ECC\_Key\_Generate syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>ECC_Key_Store</i></b>
<b>Description</b>	Command to store an ECC Key in a slot from the ECC Keys partition in R-Memory.
<b>API function name</b>	ecc_key_store
<b>Command</b>	
<b>CMD_SIZE</b>	0x30
<b>CMD_ID</b>	0x61
<b>CMD_DATA</b>	(length: 47 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot to write the <b>K</b> field. Valid values are 0 - 31.
<b>Size</b>	2
<b>CURVE</b>	
<b>Description</b>	The Elliptic Curve the key is generated from.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>P256</b> (0x01): P256 Curve - 64-byte long public key.</li><li>• <b>ED25519</b> (0x02): Ed25519 Curve - 32-byte long public key.</li></ul>
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	12
<b>K</b>	
<b>Description</b>	The ECC Key to store. The key must be a member of the field given by the curve specified in the <b>CURVE</b> field.
<b>Size</b>	32
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 29: ECC\_Key\_Store syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>ECC_Key_Read</i></b>
<b>Description</b>	Command to read the public ECC Key from a slot of the ECC Keys partition in R-Memory.
<b>API function name</b>	ecc_key_read
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x62
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot to read the public ECC Key from. Valid values are 0 - 31.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x30 - 0x50
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>INVALID_KEY</b> (0x12): The key in selected slot is invalid or does not exist.</li></ul>
<b>RES_DATA</b>	(length: 47 - 79 byte(s))
<b>CURVE</b>	
<b>Description</b>	The type of Elliptic Curve public key returned.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>P256</b> (0x01): P256 Curve - 64-byte long public key.</li><li>• <b>ED25519</b> (0x02): Ed25519 Curve - 32-byte long public key.</li></ul>
<b>ORIGIN</b>	
<b>Description</b>	The origin of the key.
<b>Size</b>	1
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>ECC_Key_Generate</b> (0x01): The key is from key generation on the device.</li><li>• <b>ECC_Key_Store</b> (0x02): The key is from key storage in the device.</li></ul>
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	13
<b>PUB_KEY</b>	
<b>Description</b>	The public key from the ECC Key slot as specified in the <b>SLOT</b> field.
<b>Size</b>	32 - 64



---

Table 30: ECC\_Key\_Read syntax





Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>ECC_Key_Erase</i></b>
<b>Description</b>	Command to erase an ECC Key from a slot in the ECC Keys partition in R-Memory.
<b>API function name</b>	ecc_key_erase
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x63
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot to erase. Valid values are 0 - 31.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 31: ECC\_Key\_Erase syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>ECDSA_Sign</i></b>
<b>Description</b>	Command to sign a message hash with an ECDSA algorithm.
<b>API function name</b>	ecdsa_sign
<b>Command</b>	
<b>CMD_SIZE</b>	0x30
<b>CMD_ID</b>	0x70
<b>CMD_DATA</b>	(length: 47 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot (from the ECC Keys partition in R-Memory) to read the key for ECDSA signing.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	13
<b>MSG_HASH</b>	
<b>Description</b>	The hash of the message to sign (max size of 32 bytes).
<b>Size</b>	32
<b>Result</b>	
<b>RES_SIZE</b>	0x50
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	• <b>INVALID_KEY</b> (0x12): The key in selected slot is invalid or does not exist.
<b>RES_DATA</b>	(length: 79 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	15
<b>R</b>	
<b>Description</b>	ECDSA signature - The R part
<b>Size</b>	32
<b>S</b>	
<b>Description</b>	ECDSA signature - The S part
<b>Size</b>	32

Table 32: ECDSA\_Sign syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>EDDSA_Sign</i></b>
<b>Description</b>	Command to sign a message with an EdDSA algorithm.
<b>API function name</b>	eddsa_sign
<b>Command</b>	
<b>CMD_SIZE</b>	0x10 - 0x1010
<b>CMD_ID</b>	0x71
<b>CMD_DATA</b>	(length: 15 - 4111 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot (from the ECC Keys partition in R-Memory) to read the key for EdDSA signing.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	13
<b>MSG</b>	
<b>Description</b>	The message to sign (max size of 4096 bytes).
<b>Size</b>	0 - 4096
<b>Result</b>	
<b>RES_SIZE</b>	0x50
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	• <b>INVALID_KEY</b> (0x12): The key in selected slot is invalid or does not exist.
<b>RES_DATA</b>	(length: 79 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	15
<b>R</b>	
<b>Description</b>	EdDSA signature - The R part
<b>Size</b>	32
<b>S</b>	
<b>Description</b>	EdDSA signature - The S part
<b>Size</b>	32

Table 33: EDDSA\_Sign syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>MCounter_Init</i></b>
<b>Description</b>	Command to initialize the Monotonic Counter.
<b>API function name</b>	mcounter_init
<b>Command</b>	
<b>CMD_SIZE</b>	0x08
<b>CMD_ID</b>	0x80
<b>CMD_DATA</b>	(length: 7 byte(s))
<b>MCOUNTER_INDEX</b>	
<b>Description</b>	The index of the Monotonic Counter to initialize. Valid values are 0 - 15.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	1
<b>MCOUNTER_VAL</b>	
<b>Description</b>	The initialization value of the Monotonic Counter.
<b>Size</b>	4
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 0 byte(s))

Table 34: MCounter\_Init syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>MCounter_Update</i></b>
<b>Description</b>	Command to update the Monotonic Counter (decrement by 1).
<b>API function name</b>	mcounter_update
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x81
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>MCOUNTER_INDEX</b>	
<b>Description</b>	The index of the Monotonic Counter to update. Valid values are 0 - 15.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x01
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	<ul style="list-style-type: none"><li>• <b>UPDATE_ERR</b> (0x13): Update operation failed (i.e. mcounter done at 0).</li><li>• <b>COUNTER_INVALID</b> (0x14): The counter disabled or locked.</li></ul>
<b>RES_DATA</b>	(length: 0 byte(s))

Table 35: MCounter\_Update syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>MCounter_Get</i></b>
<b>Description</b>	Command to get the value of the Monotonic Counter.
<b>API function name</b>	mcounter_get
<b>Command</b>	
<b>CMD_SIZE</b>	0x03
<b>CMD_ID</b>	0x82
<b>CMD_DATA</b>	(length: 2 byte(s))
<b>MCOUNTER_INDEX</b>	
<b>Description</b>	The index of the Monotonic Counter to get the value of. Valid index values are 0 - 15.
<b>Size</b>	2
<b>Result</b>	
<b>RES_SIZE</b>	0x08
<b>RESULT</b>	(1 Byte)
<b>Possible values</b>	• <b>COUNTER_INVALID</b> (0x14): The counter disabled or locked.
<b>RES_DATA</b>	(length: 7 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>MCOUNTER_VAL</b>	
<b>Description</b>	The value of the Monotonic Counter specified by the <b>MCOUNTER_INDEX</b> field.
<b>Size</b>	4

Table 36: MCounter\_Get syntax



Parameter	Description
<b>Information</b>	
<b>Name</b>	<b><i>MAC_And_Destroy</i></b>
<b>Description</b>	Command to execute the MAC-and-Destroy sequence.
<b>API function name</b>	mac_and_destroy
<b>Command</b>	
<b>CMD_SIZE</b>	0x24
<b>CMD_ID</b>	0x90
<b>CMD_DATA</b>	(length: 35 byte(s))
<b>SLOT</b>	
<b>Description</b>	The slot (from the MAC-and-Destroy data partition in R-Memory) to execute the MAC_And_Destroy sequence. Valid values are 0 - 127.
<b>Size</b>	2
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	1
<b>DATA_IN</b>	
<b>Description</b>	The data input for the MAC-and-Destroy sequence.
<b>Size</b>	32
<b>Result</b>	
<b>RES_SIZE</b>	0x24
<b>RESULT</b>	(1 Byte)
<b>RES_DATA</b>	(length: 35 byte(s))
<b>PADDING</b>	
<b>Description</b>	The padding by dummy data.
<b>Size</b>	3
<b>DATA_OUT</b>	
<b>Description</b>	The data output from the MAC-and-Destroy sequence.
<b>Size</b>	32

Table 37: MAC\_And\_Destroy syntax



## 5 User Configuration Objects

Bootloader and Application shares the memory range of I/R-Config in defined non-volatile memory.

### 5.1 Bootloader

Address Offset	Register Name	Reset Value
0x0	CFG_START_UP	0x0000000F
0x8	CFG_SENSORS	0x0003FFFF
0x10	CFG_DEBUG	0x00000001
0x14	CFG_GPO	0x00000001





<b>Register name:</b>		CFG_START_UP		
<b>Address offset:</b>		0x0		
Field	Type	Reset value	Bits	Description
RFU_1	RW W1C	0x1	0:0	Reserved for future use 1
MBIST_DIS	RW W1C	0x1	1:1	Configuration of the mutable FW test during start-up. If the test fails, TROPIC01 enters Alarm Mode. TEST_ON : 0x0 : Self test executed. TEST_OFF : 0x1 : Self test skipped.
RNGTEST_DIS	RW W1C	0x1	2:2	PTRNG test configuration in Start-up mode. TEST_ON : 0x0 : PTRNG Test is executed. If failed, TROPIC01 enters Alarm Mode. TEST_OFF : 0x1 : PTRNG Test is skipped.
MAINTENANCE_ENA	RW W1C	0x1	3:3	Configuration of Maintenance restart. MAINTENANCE_FORBIDDEN : 0x0 : Maintenance restart is forbidden. MAINTENANCE_ALLOWED : 0x1 : Maintenance restart is allowed.

<b>Register name:</b>		CFG_SENSORS		
<b>Address offset:</b>		0x8		
Field	Type	Reset value	Bits	Description
PTRNG0_TEST_DIS	RW W1C	0x1	0:0	TROPIC01 behavior when TRNG0 detects low entropy or error on internal redundancy encodings. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.



PTRNG1_TEST_DIS	RW W1C	0x1	1:1	TROPIC01 behavior when TRNG1 detects low entropy or error on internal redundancy encodings. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
OSCILLATOR_MON_DIS	RW W1C	0x1	2:2	TROPIC01 behavior when its internal oscillator detects too low frequency. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
SHIELD_DIS	RW W1C	0x1	3:3	TROPIC01 behavior when its top metal layer active shield detects tampering or an error on internal redundancy encodings. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
VOLTAGE_MON_DIS	RW W1C	0x1	4:4	TROPIC01 behavior when its voltage monitor detects over-voltage or undervoltage on VCC. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
GLITCH_DET_DIS	RW W1C	0x1	5:5	TROPIC01 behavior when its glitch detector detects a glitch on VCC. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
TEMP_SENS_DIS	RW W1C	0x1	6:6	TROPIC01 behavior when its temperature sensor detects overtemperature or undertemperature. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.



LASER_DET_DIS	RW W1C	0x1	7:7	TROPIC01 behavior when its laser detector detects an laser attack. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
EM_PULSE_DET_DIS	RW W1C	0x1	8:8	TROPIC01 behavior when its Electromagnetic Pulse detects an laser attack. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
CPU_ALERT_DIS	RW W1C	0x1	9:9	TROPIC01 behavior when its RISCv CPU detects an attack on its memories, register file or instruction pipeline. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
PIN_VERIF_BIT_FLIP_DIS	RW W1C	0x1	10:10	TROPIC01 behavior when its Pin Verification engine detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
SCB_BIT_FLIP_DIS	RW W1C	0x1	11:11	TROPIC01 behavior when its Secure Channel Block detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
CPB_BIT_FLIP_DIS	RW W1C	0x1	12:12	TROPIC01 behavior when its Command Processing Block detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
ECC_BIT_FLIP_DIS	RW W1C	0x1	13:13	TROPIC01 behavior when its ECC engine detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.



R_MEM_BIT_FLIP_DIS	RW W1C	0x1	14:14	TROPIC01 behavior when its R Memory controller detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
EKDB_BIT_FLIP_DIS	RW W1C	0x1	15:15	TROPIC01 behavior when its Entropy and Key distribution engine detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
I_MEM_BIT_FLIP_DIS	RW W1C	0x1	16:16	TROPIC01 behavior when its I Memory controller detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.
PLATFORM_BIT_FLIP_DIS	RW W1C	0x1	17:17	TROPIC01 behavior when its platform management logic (silicon life-cycle and SoC control) detects bit flip on its redundancy encoding mechanisms. NO_ACTION : 0x1 : No action ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode.

<b>Register name:</b>		CFG_DEBUG		
<b>Address offset:</b>		0x10		
Field	Type	Reset value	Bits	Description
FW_LOG_EN	RW W1C	0x1	0:0	TROPIC01 FW Logging enable.

<b>Register name:</b>		CFG_GPO		
-----------------------	--	---------	--	--



Address offset:		0x14		
Field	Type	Reset value	Bits	Description
GPO_FUNC	RW W1C	0x1	2:0	GPO pin functionality ALWAYS_LOW : 0x5 : Always in logic low state. ALWAYS_HIGH : 0x6 : Always in logic high state. INTERRUPT : 0x7 : L2 result active high interrupt.



## 5.2 Application

Address Offset	Register Name	Reset Value
0x14	CFG_GPO	0x00000001
0x18	CFG_SLEEP_MODE	0x00000001
0x20	CFG_UAP_PAIRING_KEY_WRITE	0xFFFFFFFF
0x24	CFG_UAP_PAIRING_KEY_READ	0xFFFFFFFF
0x28	CFG_UAP_PAIRING_KEY_INVALIDATE	0xFFFFFFFF
0x30	CFG_UAP_R_CONFIG_WRITE_ERASE	0x000000FF
0x34	CFG_UAP_R_CONFIG_READ	0x0000FFFF
0x40	CFG_UAP_I_CONFIG_WRITE	0x0000FFFF
0x44	CFG_UAP_I_CONFIG_READ	0x0000FFFF
0x100	CFG_UAP_PING	0x000000FF
0x110	CFG_UAP_R_MEM_DATA_WRITE	0xFFFFFFFF
0x114	CFG_UAP_R_MEM_DATA_READ	0xFFFFFFFF
0x118	CFG_UAP_R_MEM_DATA_ERASE	0xFFFFFFFF
0x120	CFG_UAP_RANDOM_VALUE_GET	0x000000FF
0x130	CFG_UAP_ECC_KEY_GENERATE	0xFFFFFFFF
0x134	CFG_UAP_ECC_KEY_STORE	0xFFFFFFFF
0x138	CFG_UAP_ECC_KEY_READ	0xFFFFFFFF
0x13c	CFG_UAP_ECC_KEY_ERASE	0xFFFFFFFF
0x140	CFG_UAP_ECDSA_SIGN	0xFFFFFFFF
0x144	CFG_UAP_EDDSA_SIGN	0xFFFFFFFF
0x150	CFG_UAP_MCOUNTER_INIT	0xFFFFFFFF
0x154	CFG_UAP_MCOUNTER_GET	0xFFFFFFFF
0x158	CFG_UAP_MCOUNTER_UPDATE	0xFFFFFFFF
0x160	CFG_UAP_MAC_AND_DESTROY	0xFFFFFFFF



Register name:		CFG_GPO		
Address offset:		0x14		
Field	Type	Reset value	Bits	Description
GPO_FUNC	RW W1C	0x1	2:0	GPO pin functionality ALWAYS_LOW : 0x5 : Always in logic low state. ALWAYS_HIGH : 0x6 : Always in logic high state. INTERRUPT : 0x7 : L2 result active high interrupt.

Register name:		CFG_SLEEP_MODE		
Address offset:		0x18		
Field	Type	Reset value	Bits	Description
SLEEP_MODE_EN	RW W1C	0x1	0:0	When 1, TROPIC01 enters Sleep mode upon receiving a <b><i>Sleep_Req</i></b> L2 Request Frame with SLEEP_KIND=SLEEP_MODE

Register name:		CFG_UAP_PAIRING_KEY_WRITE		
Address offset:		0x20		
Field	Type	Reset value	Bits	Description
WRITE_PKEY_SLOT_0	RW W1C	0xFF	7:0	Access privileges of the <b><i>Pairing_Key_Write</i></b> L3 Command packet to Pairing Key slot 0.
WRITE_PKEY_SLOT_1	RW W1C	0xFF	15:8	Access privileges of the <b><i>Pairing_Key_Write</i></b> L3 Command packet to Pairing Key slot 1.
WRITE_PKEY_SLOT_2	RW W1C	0xFF	23:16	Access privileges of the <b><i>Pairing_Key_Write</i></b> L3 Command packet to Pairing Key slot 2.



WRITE_PKEY_SLOT_3	RW W1C	0xFF	31:24	Access privileges of the <b>Pairing_Key_Write</b> L3 Command packet to Pairing Key slot 3.
-------------------	-----------	------	-------	--

<b>Register name:</b>		CFG_UAP_PAIRING_KEY_READ		
<b>Address offset:</b>		0x24		
Field	Type	Reset value	Bits	Description
READ_PKEY_SLOT_0	RW W1C	0xFF	7:0	Access privileges of the <b>Pairing_Key_Read</b> L3 Command packet to Pairing Key slot 0.
READ_PKEY_SLOT_1	RW W1C	0xFF	15:8	Access privileges of the <b>Pairing_Key_Read</b> L3 Command packet to Pairing Key slot 1.
READ_PKEY_SLOT_2	RW W1C	0xFF	23:16	Access privileges of the <b>Pairing_Key_Read</b> L3 Command packet to Pairing Key slot 2.
READ_PKEY_SLOT_3	RW W1C	0xFF	31:24	Access privileges of the <b>Pairing_Key_Read</b> L3 Command packet to Pairing Key slot 3.

<b>Register name:</b>		CFG_UAP_PAIRING_KEY_INVALIDATE		
<b>Address offset:</b>		0x28		
Field	Type	Reset value	Bits	Description
INVALIDATE_PKEY_SLOT_0	RW W1C	0xFF	7:0	Access privileges of the <b>Pairing_Key_Invalidate</b> L3 Command packet to Pairing Key slot 0.
INVALIDATE_PKEY_SLOT_1	RW W1C	0xFF	15:8	Access privileges of the <b>Pairing_Key_Invalidate</b> L3 Command packet to Pairing Key slot 1.
INVALIDATE_PKEY_SLOT_2	RW W1C	0xFF	23:16	Access privileges of the <b>Pairing_Key_Invalidate</b> L3 Command packet to Pairing Key slot 2.





INVALIDATE_PKEY_SLOT_3	RW W1C	0xFF	31:24	Access privileges of the <b>Pairing_Key_Invalidate</b> L3 Command packet to Pairing Key slot 3.
------------------------	-----------	------	-------	---

Register name:		CFG_UAP_R_CONFIG_WRITE_ERASE		
Address offset:		0x30		
Field	Type	Reset value	Bits	Description
R_CONFIG_WRITE_ERASE	RW W1C	0xFF	7:0	Access privileges of the R_Config_Write and <b>R_Config_Erase</b> L3 Command packets to all COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.

Register name:		CFG_UAP_R_CONFIG_READ		
Address offset:		0x34		
Field	Type	Reset value	Bits	Description
R_CONFIG_READ_CFG	RW W1C	0xFF	7:0	Access privileges of the <b>R_Config_Read</b> L3 Command packet to the Configuration COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.
R_CONFIG_READ_FUNC	RW W1C	0xFF	15:8	Access privileges of the <b>R_Config_Read</b> L3 Command packet to the Functionality COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.

Register name:		CFG_UAP_I_CONFIG_WRITE		
Address offset:		0x40		



Field	Type	Reset value	Bits	Description
I_CONFIG_WRITE_CFG	RW W1C	0xFF	7:0	Access privileges of the <b>I_Config_Write</b> L3 Command packet to the Configuration COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.
I_CONFIG_WRITE_FUNC	RW W1C	0xFF	15:8	Access privileges of the <b>I_Config_Write</b> L3 Command packet to the Functionality COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.

Register name:		CFG_UAP_I_CONFIG_READ		
Address offset:		0x44		
Field	Type	Reset value	Bits	Description
I_CONFIG_READ_CFG	RW W1C	0xFF	7:0	Access privileges of the <b>I_Config_Read</b> L3 Command packet to the Configuration COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.
I_CONFIG_READ_FUNC	RW W1C	0xFF	15:8	Access privileges of the <b>I_Config_Read</b> L3 Command packet to the Functionality COs. Refer to the 'User Access Privileges' section in the TROPIC01 Datasheet.

Register name:		CFG_UAP_PING		
Address offset:		0x100		
Field	Type	Reset value	Bits	Description
PING	RW W1C	0xFF	7:0	Access privileges of the <b>Ping</b> L3 Command packet.



<b>Register name:</b>		CFG_UAP_R_MEM_DATA_WRITE		
<b>Address offset:</b>		0x110		
Field	Type	Reset value	Bits	Description
WRITE_UDATA_SLOT_0_127	RW W1C	0xFF	7:0	Access privileges of the <b><i>R_Mem_Data_Write</i></b> L3 Command packet to slots 0 - 127 of the User Data partition in R-Memory.
WRITE_UDATA_SLOT_128_255	RW W1C	0xFF	15:8	Access privileges of the <b><i>R_Mem_Data_Write</i></b> L3 Command packet to slots 128 - 255 of the User Data partition in R-Memory.
WRITE_UDATA_SLOT_256_383	RW W1C	0xFF	23:16	Access privileges of the <b><i>R_Mem_Data_Write</i></b> L3 Command packet to slots 256 - 383 of the User Data partition in R-Memory.
WRITE_UDATA_SLOT_384_511	RW W1C	0xFF	31:24	Access privileges of the <b><i>R_Mem_Data_Write</i></b> L3 Command packet to slots 384 - 511 of the User Data partition in R-Memory.

<b>Register name:</b>		CFG_UAP_R_MEM_DATA_READ		
<b>Address offset:</b>		0x114		
Field	Type	Reset value	Bits	Description
READ_UDATA_SLOT_0_127	RW W1C	0xFF	7:0	Access privileges of the <b><i>R_Mem_Data_Read</i></b> L3 Command packet to slots 0 - 127 of the User Data partition in R-Memory.
READ_UDATA_SLOT_128_255	RW W1C	0xFF	15:8	Access privileges of the <b><i>R_Mem_Data_Read</i></b> L3 Command packet to slots 128 - 255 of the User Data partition in R-Memory.



READ_UDATA_SLOT_256_383	RW W1C	0xFF	23:16	Access privileges of the <b><i>R_Mem_Data_Read</i></b> L3 Command packet to slots 256 - 383 of the User Data partition in R-Memory.
READ_UDATA_SLOT_384_511	RW W1C	0xFF	31:24	Access privileges of the <b><i>R_Mem_Data_Read</i></b> L3 Command packet to slots 385 - 512 of the User Data partition in R-Memory.

<b>Register name:</b>		CFG_UAP_R_MEM_DATA_ERASE		
<b>Address offset:</b>		0x118		
Field	Type	Reset value	Bits	Description
ERASE_UDATA_SLOT_0_127	RW W1C	0xFF	7:0	Access privileges of the <b><i>R_Mem_Data_Erase</i></b> L3 Command packet to slots 0 - 127 of the User Data partition in R-Memory.
ERASE_UDATA_SLOT_128_255	RW W1C	0xFF	15:8	Access privileges of the <b><i>R_Mem_Data_Erase</i></b> L3 Command packet to slots 128 - 255 of the User Data partition in R-Memory.
ERASE_UDATA_SLOT_256_383	RW W1C	0xFF	23:16	Access privileges of the <b><i>R_Mem_Data_Erase</i></b> L3 Command packet to slots 256 - 383 of the User Data partition in R-Memory.
ERASE_UDATA_SLOT_384_511	RW W1C	0xFF	31:24	Access privileges of the <b><i>R_Mem_Data_Erase</i></b> L3 Command packet to slots 385 - 512 of the User Data partition in R-Memory.

<b>Register name:</b>		CFG_UAP_RANDOM_VALUE_GET		
<b>Address offset:</b>		0x120		



Field	Type	Reset value	Bits	Description
RANDOM_VALUE_GET	RW W1C	0xFF	7:0	Access privileges of the <b>Random_Value_Get</b> L3 Command packet.

Register name:		CFG_UAP_ECC_KEY_GENERATE		
Address offset:		0x130		
Field	Type	Reset value	Bits	Description
GEN_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b>ECC_Key_Generate</b> L3 Command packet to ECC Key slots 0-7.
GEN_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b>ECC_Key_Generate</b> L3 Command packet to ECC Key slots 8-15.
GEN_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b>ECC_Key_Generate</b> L3 Command packet to ECC Key slots 16-23.
GEN_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b>ECC_Key_Generate</b> L3 Command packet to ECC Key slots 24-31.

Register name:		CFG_UAP_ECC_KEY_STORE		
Address offset:		0x134		
Field	Type	Reset value	Bits	Description
STORE_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b>ECC_Key_Store</b> L3 Command packet to ECC Key slots 0-7.
STORE_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b>ECC_Key_Store</b> L3 Command packet to ECC Key slots 8-15.



STORE_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b>ECC_Key_Store</b> L3 Command packet to ECC Key slots 16-23.
STORE_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b>ECC_Key_Store</b> L3 Command packet to ECC Key slots 24-31.

<b>Register name:</b>		CFG_UAP_ECC_KEY_READ		
<b>Address offset:</b>		0x138		
Field	Type	Reset value	Bits	Description
READ_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b>ECC_Key_Read</b> L3 Command packet to ECC Key slots 0-7.
READ_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b>ECC_Key_Read</b> L3 Command packet to ECC Key slots 8-15.
READ_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b>ECC_Key_Read</b> L3 Command packet to ECC Key slots 16-23.
READ_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b>ECC_Key_Read</b> L3 Command packet to ECC Key slots 24-31.

<b>Register name:</b>		CFG_UAP_ECC_KEY_ERASE		
<b>Address offset:</b>		0x13c		
Field	Type	Reset value	Bits	Description
ERASE_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b>ECC_Key_Erase</b> L3 Command packet to ECC Key slots 0-7.
ERASE_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b>ECC_Key_Erase</b> L3 Command packet to ECC Key slots 8-15.



ERASE_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b><i>ECC_Key_Erase</i></b> L3 Command packet to ECC Key slots 16-23.
ERASE_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b><i>ECC_Key_Erase</i></b> L3 Command packet to ECC Key slots 24-31.

<b>Register name:</b>		CFG_UAP_ECDSA_SIGN		
<b>Address offset:</b>		0x140		
Field	Type	Reset value	Bits	Description
ECDSA_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b><i>ECDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 0-7.
ECDSA_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b><i>ECDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 8-15.
ECDSA_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b><i>ECDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 16-23.
ECDSA_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b><i>ECDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 24-31.

<b>Register name:</b>		CFG_UAP_EDDSA_SIGN		
<b>Address offset:</b>		0x144		
Field	Type	Reset value	Bits	Description
EDDSA_ECCKEY_SLOT_0_7	RW W1C	0xFF	7:0	Access privileges of the <b><i>EDDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 0-7.
EDDSA_ECCKEY_SLOT_8_15	RW W1C	0xFF	15:8	Access privileges of the <b><i>EDDSA_Sign</i></b> L3 Command packet to keys from ECC Key slots 8-15.



EDDSA_ECCKEY_SLOT_16_23	RW W1C	0xFF	23:16	Access privileges of the <b>EDDSA_Sign</b> L3 Command packet to keys from ECC Key slots 16-23.
EDDSA_ECCKEY_SLOT_24_31	RW W1C	0xFF	31:24	Access privileges of the <b>EDDSA_Sign</b> L3 Command packet to keys from ECC Key slots 24-31.

<b>Register name:</b>		CFG_UAP_MCOUNTER_INIT		
<b>Address offset:</b>		0x150		
Field	Type	Reset value	Bits	Description
MCOUNTER_INIT_0_3	RW W1C	0xFF	7:0	Access privileges of the <b>MCOUNTER_Init</b> L3 Command packet to Monotonic counters 0-3.
MCOUNTER_INIT_4_7	RW W1C	0xFF	15:8	Access privileges of the <b>MCOUNTER_Init</b> L3 Command packet to Monotonic counters 4-7.
MCOUNTER_INIT_8_11	RW W1C	0xFF	23:16	Access privileges of the <b>MCOUNTER_Init</b> L3 Command packet to Monotonic counters 8-11.
MCOUNTER_INIT_12_15	RW W1C	0xFF	31:24	Access privileges of the <b>MCOUNTER_Init</b> L3 Command packet to Monotonic counters 12-15.

<b>Register name:</b>		CFG_UAP_MCOUNTER_GET		
<b>Address offset:</b>		0x154		
Field	Type	Reset value	Bits	Description
MCOUNTER_GET_0_3	RW W1C	0xFF	7:0	Access privileges of the <b>MCOUNTER_Get</b> L3 Command packet to Monotonic counters 0-3.
MCOUNTER_GET_4_7	RW W1C	0xFF	15:8	Access privileges of the <b>MCOUNTER_Get</b> L3 Command packet to Monotonic counters 4-7.





MCOUNTER_GET_8_11	RW W1C	0xFF	23:16	Access privileges of the <b>MCounter_Get</b> L3 Command packet to Monotonic counters 8-11.
MCOUNTER_GET_12_15	RW W1C	0xFF	31:24	Access privileges of the <b>MCounter_Get</b> L3 Command packet to Monotonic counters 12-15.

<b>Register name:</b>		CFG_UAP_MCOUNTER_UPDATE		
<b>Address offset:</b>		0x158		
Field	Type	Reset value	Bits	Description
MCOUNTER_UPDATE_0_3	RW W1C	0xFF	7:0	Access privileges of the <b>MCounter_Update</b> L3 Command packet to Monotonic counters 0-3.
MCOUNTER_UPDATE_4_7	RW W1C	0xFF	15:8	Access privileges of the <b>MCounter_Update</b> L3 Command packet to Monotonic counters 4-7.
MCOUNTER_UPDATE_8_11	RW W1C	0xFF	23:16	Access privileges of the <b>MCounter_Update</b> L3 Command packet to Monotonic counters 8-11.
MCOUNTER_UPDATE_12_15	RW W1C	0xFF	31:24	Access privileges of the <b>MCounter_Update</b> L3 Command packet to Monotonic counters 12-15.

<b>Register name:</b>		CFG_UAP_MAC_AND_DESTROY		
<b>Address offset:</b>		0x160		
Field	Type	Reset value	Bits	Description
MACANDD_0_31	RW W1C	0xFF	7:0	Access privileges of the <b>MAC_And_Destroy</b> L3 Command packet (when executing a MAC-and-Destroy sequence) to slots 0-31 of the MAC-and-Destroy Partition of R-Memory.



MACANDD_32_63	RW W1C	0xFF	15:8	Access privileges of the <b>MAC_And_Destroy</b> L3 Command packet (when executing a MAC-and-Destroy sequence) to slots 32-63 of the MAC-and-Destroy Partition of R-Memory.
MACANDD_64_95	RW W1C	0xFF	23:16	Access privileges of the <b>MAC_And_Destroy</b> L3 Command packet (when executing a MAC-and-Destroy sequence) to slots 64-95 of the MAC-and-Destroy Partition of R-Memory.
MACANDD_96_127	RW W1C	0xFF	31:24	Access privileges of the <b>MAC_And_Destroy</b> L3 Command packet (when executing a MAC-and-Destroy sequence) to slots 96-127 of the MAC-and-Destroy Partition of R-Memory.



## 6 Open Issues

Document does not contain any open issues.