# TROPIC01

## ODU_TR01 - User API

**Version: 1.4.0**

**Git tag:**

tropicsquare

# 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 **CFG_START_UP** CO. |
| 0.3 | 13.3.2023 | Ondrej Ille | Add **DATA_IN\*** fields to *Ping*. Add ranges to field sizes. Change *Get_Serial_Code* to *Serial_Code_Get*. |
| 0.4 | 28.3.2023 | Ondrej Ille | Fix *Get_Info_Req* chunk size. Fix *R_Mem_Data_\** command size to 444 Bytes. Add padding to *ECDSA_Sign*, *EDDSA_Sign*, *Attest_Key_\** and *MAC_And_Destroy*. Update *CMD_ID* values to be non-linear. Add *Attest_Key_Read* L3 Command Definition. Add **CFG_UAP_ATTEST_KEY_READ** CO. Change adressing of COs to be non-linear and to correspond to order of *CMD_ID* 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 *ECC_Key_Erase* and **CFG_UAP_ECC_KEY_ERASE**. |
| 0.8 | 16.5.2023 | Prasoon Dwivedi | Fix *Encrypted_Cmd_Abt* options. Fix **CFG_UAP_ECC_KEY_ERASE** CO fields. |
| 0.9 | 24.5.2023 | Henri L'Hote | Add missing **SLOT_EXPIRED** to *R_Mem_Data_Write*. Typo fixes. |
| 0.10 | 19.6.2023 | Henri L'Hote | Removed **UDATA_LEN** from *R_Mem_Data_Read*. |
| 0.11 | 26.6.2023 | Ondrej Ille | Change CO addresses so that functional COs and configuration COs are in contiguous address regions. Change **ADDRESS** 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 **CFG_ALARM_MODE** CO. Change polarity of bits in **CFG_START_UP**. Remove **CFG_STARTUP[MBIST]**. |
| 0.15 | 1.2.2024 | Ondrej Ille | Add **CFG_STARTUP[MBIST_DIS]**, **CFG_STARTUP[RNGTEST_DIS]**, **CFG_STARTUP[MAINTENANCE_ENA]**, **CFG_STARTUP[CPU_FW_VERIFY_DIS]** and **CFG_STARTUP[SPECT_FW_VERIFY_DIS]**. |
| 0.16 | 6.2.2024 | Candice Lam | Grammar check. Consistency fix. |
| 0.17 | 1.3.2024 | Ondrej Ille | Add **SLEEP_KIND**=DEEP_-SLEEP_MODE. Add **CFG_SLEEP_MODE[DEEP_SLEEP_MODE_EN]** CO. Encode **SLEEP_KIND** more meaningfully. |
| 0.18 | 7.3.2024 | Ondrej Ille | Rework **CFG_SENSORS** to the latest state of Alarms. Flip its polarity. |
| 0.19 | 14.3.2024 | Ondrej Ille | Add **CFG_DEBUG** CO. And ***Get_Log_Req***. |
| 0.20 | 26.3.2024 | Ondrej Ille | Clarify **PKEY_INDEX** starts from 0. Change COs that refer to Pairing Key Slots to be indexed from 0. |
| 0.21 | 3.5.2024 | Ondrej Ille | Extend ***Ping*** size to 4096 bytes. |
| 0.22 | 15.5.2024 | Adam Vrba Ondrej Ille | Modify Slot Numbering to be consistently from 0. Add ***Pairing_Key_Invalidate***. Add **CFG_UAP_PAIRING_KEY_INVALIDATE**. |
| 0.23 | 15.5.2024 | Ondrej Ille | Swap "CFG" and "FUNC" in **CFG_(R|I)_CONFIG_*** COs. For **CFG_R_CONFIG_ERASE** remove split completely. |
| 0.24 | 13.6.2024 | Adam Vrba | Add padding to all L3 Commands / Results. Rename ***Encrypted_Cmd_Abt*** to ***Encrypted_Session_Abt*** |
| 0.25 | 28.8.2024 | Ondrej Ille | Add **CFG_START_UP[RFU_1]** bit. |
| 1.0 | 4.10.2024 | Jarda Hrabalek | Change L2 API for secured FW update. Changed commands ***Mutable_FW_Update*_*** |
| 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 **CPU_FW_VERIFY_DIS** and **SPECT_FW_VERIFY_DIS** fields from **CFG_START_UP**. |
| 1.0.3 | 26.11.2024 | Jarda Hrabalek | Update API *Get_Info_Req* |
| 1.0.4 | 5.12.2024 | Ondrej Ille | Remove **CFG_UAP_SERIAL_CODE_GET**. |
| 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 **CFG_GPO** |
| 1.3.0 | 18.6.2025 | Adam Vrba | Change minimum size of EdDSA sign MSG field to 0. |
| 1.4.0 | 3.9.2025 | Adam Vrba | Raise User Data slot size from 444 Bytes to 475 Bytes. |

# Contents

# 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 |
|---|---|
| **Information** | |
| **Name** | *Get_Info_Req* |
| **Description** | Request to obtain information about TROPIC01. |
| **API function name** | get_info_req |
| **Request** | |
| **REQ_ID** | 0x01 |
| **REQ_LEN** | 0x02 |
| **REQ_DATA** | (length: 2 byte(s)) |
| **OBJECT_ID** | |
| **Description** | The Identifier of the requested object. |
| **Size** | 1 |
| **Possible values** | • **X509_CERTIFICATE** (0x00): The X.509 Certificate Store (object size 3840B).<br>• **CHIP_ID** (0x01): The chip ID - the chip silicon revision and unique device ID (object size 128B).<br>• **RISCV_FW_VERSION** (0x02): Current version of Immutable FW (object size 4B)<br>• **SPECT_FW_VERSION** (0x04): The SPECT bootloader is a part of RISC-V bootloader. Returns dummy value. (object size 4B)<br>• **FW_BANK** (0xb0): The FW header read from the FW bank defined by BLOCK_INDEX choices (object size 128B). |
| **BLOCK_INDEX** | |
| **Description** | X509_CERTIFICATE: index of the 128B certificate block 0 - 29<br>FW_BANK: enumeration of possible values defined for this field<br>CHIP_ID, *_FW_VERSION: do not care |
| **Size** | 1 |
| **Possible values** | • **CPU_BANK_1** (0x01): RISC-V FW bank 1<br>• **CPU_BANK_2** (0x02): RISC-V FW bank 2<br>• **SPECT_BANK_1** (0x11): SPECT FW bank 1<br>• **SPECT_BANK_2** (0x12): RISC-V FW bank 2 |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x01 - 0x80 |
| **RSP_DATA** | (length: 1 - 128 byte(s)) |
| **OBJECT** | |

| Description | The data content of the requested object block. |
|---|---|
| **Size** | 1 - 128 |
| **RSP_CRC** | (length: 2 bytes) |

Table 1: Get_Info_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Resend_Req*** |
| **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 |
|---|---|
| **Information** | |
| **Name** | ***Startup_Req*** |
| **Description** | Request for TROPIC01 to reset. |
| **API function name** | startup_req |
| **Request** | |
| **REQ_ID** | 0xb3 |
| **REQ_LEN** | 0x01 |
| **REQ_DATA** | (length: 1 byte(s)) |
| **STARTUP_ID** | |
| **Size** | 1 |
| **Possible values** | • **REBOOT** (0x01):  Restart, then initialize as if a power-cycle was applied.<br>• **MAINTENANCE_REBOOT** (0x03): Restart, then initialize. Stay in Start-up mode and do not load the mutable FW from R-Memory. |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x00 |
| **RSP_DATA** | (length: 0 byte(s)) |
| **RSP_CRC** | (length: 2 bytes) |

Table 3: Startup_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Mutable_FW_Update_Req*** |
| **Description** | 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. |
| **API function name** | mutable_fw_update_req |
| **Request** | |
| **REQ_ID** | 0xb0 |
| **REQ_LEN** | 0x68 |
| **REQ_DATA** | (length: 104 byte(s)) |
| **SIGNATURE** | |
| **Description** | Signature of SHA256 hash of all following data in this packet. |
| **Size** | 64 |
| **HASH** | |
| **Description** | SHA256 HASH of first FW chunk of data sent using Mutable_-FW_Update_Data. |
| **Size** | 32 |
| **TYPE** | |
| **Description** | FW type which is going to be updated. |
| **Size** | 2 |
| **Possible values** | ● **FW_TYPE_CPU** (0x01): FW for RISC-V main CPU.<br>● **FW_TYPE_SPECT** (0x02): FW for SPECT coprocessor. |
| **PADDING** | |
| **Description** | Zero value. |
| **Size** | 1 |
| **HEADER_VERSION** | |
| **Description** | Current value is 1. |
| **Size** | 1 |
| **VERSION** | |
| **Size** | 4 |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x00 |
| **RSP_DATA** | (length: 0 byte(s)) |

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

Table 4: Mutable_FW_Update_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Mutable_FW_Update_Data_Req*** |
| **Description** | 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. |
| **API function name** | mutable_fw_update_data_req |
| **Request** | |
| **REQ_ID** | 0xb1 |
| **REQ_LEN** | 0x26 - 0xfe |
| **REQ_DATA** | (length: 38 - 254 byte(s)) |
| **HASH** | |
| **Description** | SHA256 HASH of the next FW chunk of data sent using Mutable_FW_Update_Data. |
| **Size** | 32 |
| **OFFSET** | |
| **Description** | The offset of the specific bank to write the FW chunk data to. |
| **Size** | 2 |
| **DATA** | |
| **Description** | The binary data to write. Data size should be a multiple of 4. |
| **Size** | 4 - 220 |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x00 |
| **RSP_DATA** | (length: 0 byte(s)) |
| **RSP_CRC** | (length: 2 bytes) |

Table 5: Mutable_FW_Update_Data_Req syntax

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

Table 6: Get_Log_Req syntax

# 4   Application API

## 4.1   L2 Request / Response frames

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Get_Info_Req*** |
| **Description** | Request to obtain information about TROPIC01. |
| **API function name** | get_info_req |
| **Request** | |
| **REQ_ID** | 0x01 |
| **REQ_LEN** | 0x02 |
| **REQ_DATA** | (length: 2 byte(s)) |
| **OBJECT_ID** | |
| **Description** | The Identifier of the requested object. |
| **Size** | 1 |
| **Possible values** | • **X509_CERTIFICATE** (0x00): The X.509 Certificate Store (object size 3840B). <br> • **CHIP_ID** (0x01): The chip ID - the chip silicon revision and unique device ID (object size 128B). <br> • **RISCV_FW_VERSION** (0x02): Current version of RISC-V FW (object size 4B) <br> • **SPECT_FW_VERSION** (0x04): Current version of SPECT FW (object size 4B) |
| **BLOCK_INDEX** | |
| **Description** | X509_CERTIFICATE: index of the 128B certificate block 0 - 29 <br> CHIP_ID, *_FW_VERSION: do not care |
| **Size** | 1 |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x01 - 0x80 |
| **RSP_DATA** | (length: 1 - 128 byte(s)) |
| **OBJECT** | |
| **Description** | The data content of the requested object block. |
| **Size** | 1 - 128 |
| **RSP_CRC** | (length: 2 bytes) |

Table 7: Get_Info_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Handshake_Req*** |
| **Description** | Request to execute a Secure Channel Handshake and establish a new Secure Channel Session (TROPIC01 moves to Secure Channel Mode). |
| **API function name** | handshake_req |
| **Request** | |
| **REQ_ID** | 0x02 |
| **REQ_LEN** | 0x21 |
| **REQ_DATA** | (length: 33 byte(s)) |
| **E_HPUB** | |
| **Description** | The Host MCU's Ephemeral X25519 public key. A little endian encoding of the x-coordinate from the public Curve25519 point. |
| **Size** | 32 |
| **PKEY_INDEX** | |
| **Description** | 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). |
| **Size** | 1 |
| **Possible values** | • **PAIRING_KEY_SLOT_0** (0x00): Corresponds to $S_{H0Pub}$.<br>• **PAIRING_KEY_SLOT_1** (0x01): Corresponds to $S_{H1Pub}$.<br>• **PAIRING_KEY_SLOT_2** (0x02): Corresponds to $S_{H2Pub}$.<br>• **PAIRING_KEY_SLOT_3** (0x03): Corresponds to $S_{H3Pub}$. |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x30 |
| **RSP_DATA** | (length: 48 byte(s)) |
| **E_TPUB** | |
| **Description** | TROPIC01's X25519 Ephemeral key. |
| **Size** | 32 |
| **T_TAUTH** | |
| **Description** | The Secure Channel Handshake Authentication Tag. |
| **Size** | 16 |
| **RSP_CRC** | (length: 2 bytes) |

Table 8: Handshake_Req syntax

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

Table 9: Encrypted_Cmd_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Encrypted_Session_Abt_Req*** |
| **Description** | Request to abort current Secure Channel Session and execution of L3 command (TROPIC01 moves to Idle Mode). |
| **API function name** | encrypted_session_abt_req |
| **Request** | |
| **REQ_ID** | 0x08 |
| **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 10: Encrypted_Session_Abt_Req syntax

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

Table 12: Sleep_Req syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Startup_Req*** |
| **Description** | Request for TROPIC01 to reset. |
| **API function name** | startup_req |
| **Request** | |
| **REQ_ID** | 0xb3 |
| **REQ_LEN** | 0x01 |
| **REQ_DATA** | (length: 1 byte(s)) |
| **STARTUP_ID** | |
| **Size** | 1 |
| **Possible values** | • **REBOOT** (0x01): Restart, then initialize as if a power-cycle was applied. <br> • **MAINTENANCE_REBOOT** (0x03): Restart, then initialize. Stay in Start-up mode and do not load the mutable FW from R-Memory. |
| **REQ_CRC** | (length: 2 bytes) |
| **Response** | |
| **RSP_LEN** | 0x00 |
| **RSP_DATA** | (length: 0 byte(s)) |
| **RSP_CRC** | (length: 2 bytes) |

Table 13: Startup_Req syntax

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

Table 14: Get_Log_Req syntax

## 4.2   L3 Commands / Result packets

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

Table 15: Ping syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Pairing_Key_Write*** |
| **Description** | Command to write the X25519 public key to a Pairing Key slot. |
| **API function name** | pairing_key_write |
| **Command** | |
| **CMD_SIZE** | 0x24 |
| **CMD_ID** | 0x10 |
| **CMD_DATA** | (length: 35 byte(s)) |
| **SLOT** | |
| **Description** | The Pairing Key slot. Valid values are 0 - 3. |
| **Size** | 2 |
| **Possible values** | • **PAIRING_KEY_SLOT_0** (0x00): Corresponds to $S_{H0Pub}$.<br>• **PAIRING_KEY_SLOT_1** (0x01): Corresponds to $S_{H1Pub}$.<br>• **PAIRING_KEY_SLOT_2** (0x02): Corresponds to $S_{H2Pub}$.<br>• **PAIRING_KEY_SLOT_3** (0x03): Corresponds to $S_{H3Pub}$. |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 1 |
| **S_HIPUB** | |
| **Description** | The X25519 public key to be written in the Pairing Key slot specified in the SLOT field. |
| **Size** | 32 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 16: Pairing_Key_Write syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Pairing_Key_Read*** |
| **Description** | Command to read the X25519 public key from a Pairing Key slot. |
| **API function name** | pairing_key_read |
| **Command** | |
| **CMD_SIZE** | 0x03 |
| **CMD_ID** | 0x11 |
| **CMD_DATA** | (length: 2 byte(s)) |
| **SLOT** | |
| **Description** | The Pairing Key slot. Valid values are 0 - 3. |
| **Size** | 2 |
| **Possible values** | • **PAIRING_KEY_SLOT_0** (0x00): Corresponds to $S_{H0Pub}$. <br> • **PAIRING_KEY_SLOT_1** (0x01): Corresponds to $S_{H1Pub}$. <br> • **PAIRING_KEY_SLOT_2** (0x02): Corresponds to $S_{H2Pub}$. <br> • **PAIRING_KEY_SLOT_3** (0x03): Corresponds to $S_{H3Pub}$. |
| **Result** | |
| **RES_SIZE** | 0x24 |
| **RESULT** | (1 Byte) |
| **Possible values** | • **PAIRING_KEY_EMPTY** (0x15): The Pairing key slot is in "Blank" state. A Pairing Key has not been written to it yet. <br> • **PAIRING_KEY_INVALID** (0x16): The Pairing key slot is in "Invalidated" state. The Pairing key has been invalidated. |
| **RES_DATA** | (length: 35 byte(s)) |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 3 |
| **S_HIPUB** | |
| **Description** | The X25519 public key to be written in the Pairing Key slot specified in the SLOT field. |
| **Size** | 32 |

Table 17: Pairing_Key_Read syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***Pairing_Key_Invalidate*** |
| **Description** | Command to invalidate the X25519 public key in a Pairing Key slot. |
| **API function name** | pairing_key_invalidate |
| **Command** | |
| **CMD_SIZE** | 0x03 |
| **CMD_ID** | 0x12 |
| **CMD_DATA** | (length: 2 byte(s)) |
| **SLOT** | |
| **Description** | The Pairing Key slot. Valid values are 0 - 3. |
| **Size** | 2 |
| **Possible values** | • **PAIRING_KEY_SLOT_0** (0x00): Corresponds to $S_{H0Pub}$.<br>• **PAIRING_KEY_SLOT_1** (0x01): Corresponds to $S_{H1Pub}$.<br>• **PAIRING_KEY_SLOT_2** (0x02): Corresponds to $S_{H2Pub}$.<br>• **PAIRING_KEY_SLOT_3** (0x03): Corresponds to $S_{H3Pub}$. |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 18: Pairing_Key_Invalidate syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | *R_Config_Write* |
| **Description** | Command to write a single CO to R-Config. |
| **API function name** | r_config_write |
| **Command** | |
| **CMD_SIZE** | 0x08 |
| **CMD_ID** | 0x20 |
| **CMD_DATA** | (length: 7 byte(s)) |
| **ADDRESS** | |
| **Description** | The CO address offset for TROPIC01 to compute the actual CO address. |
| **Size** | 2 |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 1 |
| **VALUE** | |
| **Description** | The CO value to write in the computed address. |
| **Size** | 4 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 19: R_Config_Write syntax

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

Table 20: R_Config_Read syntax

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

Table 21: R_Config_Erase syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***I_Config_Write*** |
| **Description** | Command to write a single bit of CO (from I-Config) from 1 to 0. |
| **API function name** | i_config_write |
| **Command** | |
| **CMD_SIZE** | 0x04 |
| **CMD_ID** | 0x30 |
| **CMD_DATA** | (length: 3 byte(s)) |
| **ADDRESS** | |
| **Description** | The CO address offset for TROPIC01 to compute the actual CO address. |
| **Size** | 2 |
| **BIT_INDEX** | |
| **Description** | The bit to write from 1 to 0. Valid values are 0-31. |
| **Size** | 1 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 22: I_Config_Write syntax

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

Table 23: I_Config_Read syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***R_Mem_Data_Write*** |
| **Description** | Command to write general purpose data in a slot from the User Data partition in R-Memory. |
| **API function name** | r_mem_data_write |
| **Command** | |
| **CMD_SIZE** | 0x05 - 0x1df |
| **CMD_ID** | 0x40 |
| **CMD_DATA** | (length: 4 - 478 byte(s)) |
| **UDATA_SLOT** | |
| **Description** | The slot of the User Data partition. Valid values are 0 - 511. |
| **Size** | 2 |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 1 |
| **DATA** | |
| **Description** | The data stream to be written in the slot specified in the UDATA_SLOT L3 field. |
| **Size** | 1 - 475 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **Possible values** | ● **WRITE_FAIL** (0x10): The slot is already written in. |
| **RES_DATA** | (length: 0 byte(s)) |

Table 24: R_Mem_Data_Write syntax

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

Table 25: R_Mem_Data_Read syntax

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

Table 26: R_Mem_Data_Erase syntax

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

Table 27: Random_Value_Get syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***ECC_Key_Generate*** |
| **Description** | Command to generate an ECC Key and store the key in a slot from the ECC Keys partition in R-Memory. |
| **API function name** | ecc_key_generate |
| **Command** | |
| **CMD_SIZE** | 0x04 |
| **CMD_ID** | 0x60 |
| **CMD_DATA** | (length: 3 byte(s)) |
| **SLOT** | |
| **Description** | The slot to write the generated key. Valid values are 0 - 31. |
| **Size** | 2 |
| **CURVE** | |
| **Description** | The Elliptic Curve the key is generated from. |
| **Size** | 1 |
| **Possible values** | • **P256** (0x01): P256 Curve - 64-byte long public key.<br>• **ED25519** (0x02): Ed25519 Curve - 32-byte long public key. |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 28: ECC_Key_Generate syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | *ECC_Key_Store* |
| **Description** | Command to store an ECC Key in a slot from the ECC Keys partition in R-Memory. |
| **API function name** | ecc_key_store |
| **Command** | |
| **CMD_SIZE** | 0x30 |
| **CMD_ID** | 0x61 |
| **CMD_DATA** | (length: 47 byte(s)) |
| **SLOT** | |
| **Description** | The slot to write the **K** field. Valid values are 0 - 31. |
| **Size** | 2 |
| **CURVE** | |
| **Description** | The Elliptic Curve the key is generated from. |
| **Size** | 1 |
| **Possible values** | • **P256** (0x01): P256 Curve - 64-byte long public key.<br>• **ED25519** (0x02): Ed25519 Curve - 32-byte long public key. |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 12 |
| **K** | |
| **Description** | The ECC Key to store. The key must be a member of the field given by the curve specified in the **CURVE** field. |
| **Size** | 32 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **RES_DATA** | (length: 0 byte(s)) |

Table 29: ECC_Key_Store syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***ECC_Key_Read*** |
| **Description** | Command to read the public ECC Key from a slot of the ECC Keys partition in R-Memory. |
| **API function name** | ecc_key_read |
| **Command** | |
| **CMD_SIZE** | 0x03 |
| **CMD_ID** | 0x62 |
| **CMD_DATA** | (length: 2 byte(s)) |
| **SLOT** | |
| **Description** | The slot to read the public ECC Key from. Valid values are 0 - 31. |
| **Size** | 2 |
| **Result** | |
| **RES_SIZE** | 0x30 - 0x50 |
| **RESULT** | (1 Byte) |
| **Possible values** | • **INVALID_KEY** (0x12): The key in the requested slot does not exist. |
| **RES_DATA** | (length: 47 - 79 byte(s)) |
| **CURVE** | |
| **Description** | The type of Elliptic Curve public key returned. |
| **Size** | 1 |
| **Possible values** | • **P256** (0x01): P256 Curve - 64-byte long public key.<br>• **ED25519** (0x02): Ed25519 Curve - 32-byte long public key. |
| **ORIGIN** | |
| **Description** | The origin of the key. |
| **Size** | 1 |
| **Possible values** | • **ECC_Key_Generate** (0x01): The key is from key generation on the device.<br>• **ECC_Key_Store** (0x02): The key is from key storage in the device. |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 13 |
| **PUB_KEY** | |
| **Description** | The public key from the ECC Key slot as specified in the **SLOT** field. |
| **Size** | 32 - 64 |

Table 30: ECC_Key_Read syntax

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

Table 31: ECC_Key_Erase syntax

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

Table 32: ECDSA_Sign syntax

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

Table 33: EDDSA_Sign syntax

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

Table 34: MCounter_Init syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***MCounter_Update*** |
| **Description** | Command to update the Monotonic Counter (decrement by 1). |
| **API function name** | mcounter_update |
| **Command** | |
| **CMD_SIZE** | 0x03 |
| **CMD_ID** | 0x81 |
| **CMD_DATA** | (length: 2 byte(s)) |
| **MCOUNTER_INDEX** | |
| **Description** | The index of the Monotonic Counter to update. Valid values are 0 - 15. |
| **Size** | 2 |
| **Result** | |
| **RES_SIZE** | 0x01 |
| **RESULT** | (1 Byte) |
| **Possible values** | • **UPDATE_ERR** (0x13): Failure to update the specified Monotonic Counter. The Monotonic Counter is already at 0.<br>• **COUNTER_INVALID** (0x14): The Monotonic Counter detects an attack and is locked. The counter must be reinitialized. |
| **RES_DATA** | (length: 0 byte(s)) |

Table 35: MCounter_Update syntax

| Parameter | Description |
|---|---|
| **Information** | |
| **Name** | ***MCounter_Get*** |
| **Description** | Command to get the value of the Monotonic Counter. |
| **API function name** | mcounter_get |
| **Command** | |
| **CMD_SIZE** | 0x03 |
| **CMD_ID** | 0x82 |
| **CMD_DATA** | (length: 2 byte(s)) |
| **MCOUNTER_INDEX** | |
| **Description** | The index of the Monotonic Counter to get the value of. Valid index values are 0 - 15. |
| **Size** | 2 |
| **Result** | |
| **RES_SIZE** | 0x08 |
| **RESULT** | (1 Byte) |
| **Possible values** | • **COUNTER_INVALID** (0x14): The Monotonic Counter detects an attack and is locked. The counter must be reinitialized. |
| **RES_DATA** | (length: 7 byte(s)) |
| **PADDING** | |
| **Description** | The padding by dummy data. |
| **Size** | 3 |
| **MCOUNTER_VAL** | |
| **Description** | The value of the Monotonic Counter specified by the **MCOUNTER_INDEX** field. |
| **Size** | 4 |

Table 36: MCounter_Get syntax

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

| Register name: | | CFG_START_UP | | |
|---|---|---|---|---|
| Address offset: | | 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. |

| Register name: | | CFG_SENSORS | | |
|---|---|---|---|---|
| Address offset: | | 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.<br>NO_ACTION : 0x1 : No action<br>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.<br>NO_ACTION : 0x1 : No action<br>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 encdoings.<br>NO_ACTION : 0x1 : No action<br>ENTER_ALARM_MODE : 0x0 : Enter Alarm Mode. |
| VOLTAGE_MON_DIS | RW W1C | 0x1 | 4:4 | TROPIC01 behavior when its voltage monitor detects overvoltage or undervoltage on VCC.<br>NO_ACTION : 0x1 : No action<br>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.<br>NO_ACTION : 0x1 : No action<br>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.<br>NO_ACTION : 0x1 : No action<br>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. |

| Register name: | | CFG_DEBUG | | |
|---|---|---|---|---|
| Address offset: | | 0x10 | | |
| Field | Type | Reset value | Bits | Description |
| FW_LOG_EN | RW W1C | 0x1 | 0:0 | TROPIC01 FW Logging enable. |

| Register name: | | CFG_GPO |
|---|---|---|

| Address offset: | | 0x14 | | |
|---|---|---|---|---|
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| GPO_FUNC | RW W1C | 0x1 | 2:0 | GPO pin functinality<br>ALWAYS_LOW : 0x5 : Always in logic low state.<br>ALWAYS_HIGH : 0x6 : Always in logic high state.<br>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 functinality 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 *Sleep_Req* 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 *Pairing_Key_Write* L3 Command packet to Pairing Key slot 0. | |
| WRITE_PKEY_SLOT_1 | RW W1C | 0xFF | 15:8 | Access privileges of the *Pairing_Key_Write* L3 Command packet to Pairing Key slot 1. | |
| WRITE_PKEY_SLOT_2 | RW W1C | 0xFF | 23:16 | Access privileges of the *Pairing_Key_Write* L3 Command packet to Pairing Key slot 2. | |

| WRITE_PKEY_SLOT_3 | RW W1C | 0xFF | 31:24 | Access privileges of the **_Pairing_Key_Write_** L3 Command packet to Pairing Key slot 3. |
|---|---|---|---|---|

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

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

| INVALIDATE_PKEY_SLOT_3 | RW W1C | 0xFF | 31:24 | Access privileges of the **Pairing_Key_Invalidate** 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 **R_Config_Erase** 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 **R_Config_Read** 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 **R_Config_Read** 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 **I_Config_Write** 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 **I_Config_Write** 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 **I_Config_Read** 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 **I_Config_Read** 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 **Ping** L3 Command packet. |

| Register name: | CFG_UAP_R_MEM_DATA_WRITE | | | |
|---|---|---|---|---|
| Address offset: | 0x110 | | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| WRITE_UDATA_SLOT_0_127 | RW W1C | 0xFF | 7:0 | Access privileges of the **R_Mem_Data_Write** 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 **R_Mem_Data_Write** 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 **R_Mem_Data_Write** 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 **R_Mem_Data_Write** L3 Command packet to slots 384 - 511 of the User Data partition in R-Memory. |

| Register name: | CFG_UAP_R_MEM_DATA_READ | | | |
|---|---|---|---|---|
| Address offset: | 0x114 | | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| READ_UDATA_SLOT_0_127 | RW W1C | 0xFF | 7:0 | Access privileges of the **R_Mem_Data_Read** 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 **R_Mem_Data_Read** 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 **R_Mem_Data_Read** 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 **R_Mem_Data_Read** L3 Command packet to slots 385 - 512 of the User Data partition in R-Memory. |

| Register name: | | CFG_UAP_R_MEM_DATA_ERASE | | |
|---|---|---|---|---|
| **Address offset:** | | 0x118 | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| ERASE_UDATA_SLOT_0_127 | RW W1C | 0xFF | 7:0 | Access privileges of the **R_Mem_Data_Erase** 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 **R_Mem_Data_Erase** 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 **R_Mem_Data_Erase** 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 **R_Mem_Data_Erase** L3 Command packet to slots 385 - 512 of the User Data partition in R-Memory. |

| Register name: | CFG_UAP_RANDOM_VALUE_GET |
|---|---|
| **Address offset:** | 0x120 |

| Field | Type | Reset value | Bits | Description |
|---|---|---|---|---|
| RANDOM_VALUE_GET | RW W1C | 0xFF | 7:0 | Access privileges of the **Random_Value_Get** 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 **ECC_Key_Generate** L3 Command packet to ECC Key slots 0-7. |
| GEN_ECCKEY_SLOT_8_15 | RW W1C | 0xFF | 15:8 | Access privileges of the **ECC_Key_Generate** L3 Command packet to ECC Key slots 8-15. |
| GEN_ECCKEY_SLOT_16_23 | RW W1C | 0xFF | 23:16 | Access privileges of the **ECC_Key_Generate** L3 Command packet to ECC Key slots 16-23. |
| GEN_ECCKEY_SLOT_24_31 | RW W1C | 0xFF | 31:24 | Access privileges of the **ECC_Key_Generate** 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 **ECC_Key_Store** L3 Command packet to ECC Key slots 0-7. |
| STORE_ECCKEY_SLOT_8_15 | RW W1C | 0xFF | 15:8 | Access privileges of the **ECC_Key_Store** L3 Command packet to ECC Key slots 8-15. |

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

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

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

| | | | | |
|---|---|---|---|---|
| ERASE_ECCKEY_SLOT_16_23 | RW W1C | 0xFF | 23:16 | Access privileges of the **ECC_Key_Erase** L3 Command packet to ECC Key slots 16-23. |
| ERASE_ECCKEY_SLOT_24_31 | RW W1C | 0xFF | 31:24 | Access privileges of the **ECC_Key_Erase** L3 Command packet to ECC Key slots 24-31. |

| Register name: | | CFG_UAP_ECDSA_SIGN | | |
|---|---|---|---|---|
| **Address offset:** | | 0x140 | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| ECDSA_ECCKEY_SLOT_0_7 | RW W1C | 0xFF | 7:0 | Access privileges of the **ECDSA_Sign** 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 **ECDSA_Sign** 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 **ECDSA_Sign** 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 **ECDSA_Sign** L3 Command packet to keys from ECC Key slots 24-31. |

| Register name: | | CFG_UAP_EDDSA_SIGN | | |
|---|---|---|---|---|
| **Address offset:** | | 0x144 | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| EDDSA_ECCKEY_SLOT_0_7 | RW W1C | 0xFF | 7:0 | Access privileges of the **EDDSA_Sign** 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 **EDDSA_Sign** 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 **EDDSA_Sign** 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 **EDDSA_Sign** L3 Command packet to keys from ECC Key slots 24-31. |

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

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

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

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

| Register name: | | CFG_UAP_MAC_AND_DESTROY | | |
|---|---|---|---|---|
| **Address offset:** | | 0x160 | | |
| **Field** | **Type** | **Reset value** | **Bits** | **Description** |
| MACANDD_0_31 | RW W1C | 0xFF | 7:0 | Access privileges of the **MAC_And_Destroy** 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 **MAC_And_Destroy** 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 **MAC_And_Destroy** 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 **MAC_And_Destroy** 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.