

AC784xx_DFP EEP

4.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	2
2.1	File List	2
3	Class Documentation	3
3.1	Eep_Hal_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	Callback	3
3.1.2.2	EepBlockSize	4
3.1.2.3	EepFlashBaseAddr	4
3.1.2.4	EepGroupNum	4
3.1.2.5	EepRamBaseAddr	4
3.1.2.6	EepSize	4
3.2	Eep_Hal_VersionType Struct Reference	5
3.2.1	Detailed Description	5
3.2.2	Member Data Documentation	5
3.2.2.1	Major	5
3.2.2.2	Minor	5
3.2.2.3	Patch	5

4 File Documentation	6
4.1 AC784xx_API_Reference_Manual_EEP.pdf File Reference	6
4.2 Eep_Hal.c File Reference	6
4.2.1 Detailed Description	7
4.2.2 Macro Definition Documentation	7
4.2.2.1 ALIGN_DOWN	7
4.2.2.2 ALIGN_UP	8
4.2.2.3 EEP_BLOCK_MAX_NUM	8
4.2.2.4 EEP_FLASH_ERASED_VALUE	8
4.2.2.5 EEP_FLASH_GROUP_BASE	8
4.2.2.6 EEP_FLASH_GROUP_END	9
4.2.2.7 EEP_FLASH_MAX_UNIT_SIZE	9
4.2.2.8 EEP_GROUP_FLAG_VALUE	9
4.2.2.9 EEP_GROUP_LOGIC_ADDR_END	9
4.2.2.10 EEP_GROUP_LOGIC_ADDR_START	10
4.2.2.11 EEP_GROUP_MAX_NUM	10
4.2.2.12 EEP_LOGIC_ADDR_FROM_FLASH	10
4.2.2.13 EEP_LOGIC_ADDR_TO_RAM	10
4.2.2.14 EEP_MAX_SIZE	10
4.2.2.15 EEP_MIN_SIZE	11
4.2.2.16 EEP_RAM_ADDR_TO_LOGIC	11
4.2.2.17 EEP_RAM_BLOCK_BASE	11
4.2.2.18 EEP_RAM_BLOCK_END	11
4.2.2.19 EEP_XOR_PATTERN	12
4.2.3 Function Documentation	12
4.2.3.1 Eep_Hal_AbortFreeRes()	12
4.2.3.2 Eep_Hal_Erase()	12
4.2.3.3 Eep_Hal_GetStatus()	13
4.2.3.4 EEP_Hal_GetVersion()	13
4.2.3.5 Eep_Hal_Init()	14
4.2.3.6 Eep_Hal_Read()	14

4.2.3.7	Eep_Hal_Refresh()	15
4.2.3.8	Eep_Hal_Write()	15
4.3	Eep_Hal.h File Reference	16
4.3.1	Detailed Description	17
4.3.2	Typedef Documentation	17
4.3.2.1	Eep_CallbackType	17
4.3.3	Enumeration Type Documentation	17
4.3.3.1	Eep_Hal_StatusType	17
4.3.4	Function Documentation	18
4.3.4.1	Eep_Hal_AbortFreeRes()	18
4.3.4.2	Eep_Hal_Erase()	18
4.3.4.3	Eep_Hal_GetStatus()	19
4.3.4.4	Eep_Hal_GetVersion()	20
4.3.4.5	Eep_Hal_Init()	20
4.3.4.6	Eep_Hal_Read()	21
4.3.4.7	Eep_Hal_Refresh()	22
4.3.4.8	Eep_Hal_Write()	22
Index		24

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Eep_Hal_ConfigType		
EEP configuration structure	3
Eep_Hal_VersionType		
EEP version structure	5

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_EEP.pdf	6
Eep_Hal.c	
This file provides EEP integration functions	6
Eep_Hal.h	
This file provides EEP api	16

Chapter 3

Class Documentation

3.1 Eep_Hal_ConfigType Struct Reference

EEP configuration structure.

```
#include <Eep_Hal.h>
```

Public Attributes

- uint16 [EepSize](#)
- uint16 [EepBlockSize](#)
- uint16 [EepGroupNum](#)
- uint32 [EepRamBaseAddr](#)
- uint32 [EepFlashBaseAddr](#)
- [Eep_CallbackType](#) [Callback](#)

3.1.1 Detailed Description

EEP configuration structure.

Definition at line 85 of file Eep_Hal.h.

3.1.2 Member Data Documentation

3.1.2.1 Callback

[Eep_CallbackType](#) [Eep_Hal_ConfigType::Callback](#)

Callback when current block have not enough buffer to storage.

Definition at line 92 of file Eep_Hal.h.

3.1.2.2 EepBlockSize

```
uint16 Eep_Hal_ConfigType::EepBlockSize
```

The block size of EEP (byte), range: 1~1023, recommend: 512.

Definition at line 88 of file Eep_Hal.h.

3.1.2.3 EepFlashBaseAddr

```
uint32 Eep_Hal_ConfigType::EepFlashBaseAddr
```

The base address of FLASH used for EEP emulation.

Definition at line 91 of file Eep_Hal.h.

3.1.2.4 EepGroupNum

```
uint16 Eep_Hal_ConfigType::EepGroupNum
```

The backup number of EEP, range: 2~4, recommend: 4.

Definition at line 89 of file Eep_Hal.h.

3.1.2.5 EepRamBaseAddr

```
uint32 Eep_Hal_ConfigType::EepRamBaseAddr
```

The base address of RAM used for EEP emulation.

Definition at line 90 of file Eep_Hal.h.

3.1.2.6 EepSize

```
uint16 Eep_Hal_ConfigType::EepSize
```

The total size of EEP (byte), range: 512, 1024, 2048, 4096.

Definition at line 87 of file Eep_Hal.h.

The documentation for this struct was generated from the following file:

- [Eep_Hal.h](#)

3.2 Eep_Hal_VersionType Struct Reference

EEP version structure.

```
#include <Eep_Hal.h>
```

Public Attributes

- uint32 [Major](#)
- uint32 [Minor](#)
- uint32 [Patch](#)

3.2.1 Detailed Description

EEP version structure.

Definition at line 98 of file Eep_Hal.h.

3.2.2 Member Data Documentation

3.2.2.1 Major

```
uint32 Eep_Hal_VersionType::Major
```

Definition at line 100 of file Eep_Hal.h.

3.2.2.2 Minor

```
uint32 Eep_Hal_VersionType::Minor
```

Definition at line 101 of file Eep_Hal.h.

3.2.2.3 Patch

```
uint32 Eep_Hal_VersionType::Patch
```

Definition at line 102 of file Eep_Hal.h.

The documentation for this struct was generated from the following file:

- [Eep_Hal.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_EEP.pdf File Reference

4.2 Eep_Hal.c File Reference

This file provides EEP integration functions.

```
#include "Device_Register.h"
#include "Eep_Hal.h"
#include "Fls_Hal.h"
#include "AC784xx_Fls_Reg.h"
#include "System_AC784xx.h"
```

Macros

- #define EEP_FLASH_ERASED_VALUE (0xFFFFFFFFU)
group size define rule:
- #define EEP_GROUP_FLAG_VALUE (0x00000000U)
EEP flash operate group flag.
- #define EEP_MIN_SIZE (512U)
EEP min size in byte.
- #define EEP_MAX_SIZE (4096U)
EEP max size in byte.
- #define EEP_BLOCK_MAX_NUM (8U)
EEP block max number.
- #define EEP_GROUP_MAX_NUM (4U)
EEP group max number.
- #define EEP_FLASH_MAX_UNIT_SIZE (16U)
EEP flash max program unit size.
- #define EEP_RAM_BLOCK_BASE(blk)
The EEP driver each RAM block base address.
- #define EEP_RAM_BLOCK_END(blk)
The EEP driver each RAM block end address.
- #define EEP_GROUP_LOGIC_ADDR_START(blk) (((uint32)(blk) * (uint32)(EepConfigPtr->EepBlockSize))
The EEP driver each group logic base address.
- #define EEP_GROUP_LOGIC_ADDR_END(blk) (((uint32)(blk) + 1U) * (uint32)(EepConfigPtr->EepBlockSize))

- *The EEPROM driver each group logic end address.*
• #define `EEP_RAM_ADDR_TO_LOGIC(addr)` $((addr) - EepConfigPtr->EepRamBaseAddr)$
- *The EEPROM ram address to logic address.*
• #define `EEP_LOGIC_ADDR_TO_RAM(addr)` $((addr) + EepConfigPtr->EepRamBaseAddr)$
- *The EEPROM ram address to logic address.*
• #define `EEP_LOGIC_ADDR_FROM_FLASH(addr)` $((addr) \& 0xffU)$
- *The EEPROM ram address to logic address.*
• #define `EEP_FLASH_GROUP_BASE(blk, grp)`
- *The EEPROM driver each flash group base address.*
• #define `EEP_FLASH_GROUP_END(blk, grp)`
- *The EEPROM driver each flash group end address.*
• #define `ALIGN_UP(val, align)` $((val) + (align) - 1U) \& (\sim((align) - 1U))$
- #define `ALIGN_DOWN(val, align)` $((val) \& (\sim(align) - 1U))$
- #define `EEP_XOR_PATTERN` `0x5aa5U`

Functions

- Hal_StatusType `Eep_Hal_Write` (uint32 Addr, const uint8 *Data, uint16 Size)
Main EEPROM write interface.
- Hal_StatusType `Eep_Hal_Read` (uint32 Addr, uint8 *Data, uint16 Size)
Main EEPROM read interface.
- Hal_StatusType `Eep_Hal_Erase` (void)
Erase entire EEPROM storage space.
- Hal_StatusType `Eep_Hal_Init` (const `Eep_Hal_ConfigType` *ConfigPtr)
Initialize EEPROM driver.
- Hal_StatusType `Eep_Hal_Refresh` (void)
Refresh EEPROM data from flash.
- `Eep_Hal_StatusType` `Eep_Hal_GetStatus` (void)
Retrieve EEPROM driver status.
- void `Eep_Hal_AbortFreeRes` (void)
Release the acquired resource when Eep_Hal is aborted abnormally.
- void `EEP_Hal_GetVersion` (`Eep_Hal_VersionType` *Version)
Get driver version information.

4.2.1 Detailed Description

This file provides EEPROM integration functions.

4.2.2 Macro Definition Documentation

4.2.2.1 ALIGN_DOWN

```
#define ALIGN_DOWN(  
    val,  
    align ) ((val) & (~(align) - 1U))
```

Definition at line 110 of file Eep_Hal.c.

4.2.2.2 ALIGN_UP

```
#define ALIGN_UP(  
    val,  
    align ) (((val) + (align) - 1U) & (~((align) - 1U)))
```

Definition at line 109 of file Eep_Hal.c.

4.2.2.3 EEP_BLOCK_MAX_NUM

```
#define EEP_BLOCK_MAX_NUM (8U)
```

EEP block max number.

Definition at line 70 of file Eep_Hal.c.

4.2.2.4 EEP_FLASH_ERASED_VALUE

```
#define EEP_FLASH_ERASED_VALUE (0xFFFFFFFFU)
```

group size define rule:

1. Total EEP size (Byte) = EEP SRAM size
2. The min EEP_GROUP_SIZE = EEP_SIZE (word) * 2 * EepFlashMinProgByteNum
3. EEP_FLASH_SIZE = EEP_GROUP_SIZE * Total groupEEP flash erased value

Definition at line 58 of file Eep_Hal.c.

4.2.2.5 EEP_FLASH_GROUP_BASE

```
#define EEP_FLASH_GROUP_BASE(  
    blk,  
    grp )
```

Value:

```
(EepConfigPtr->EepFlashBaseAddr + \  
    (((uint32) (blk) * (uint32) (EepConfigPtr->  
    EepGroupNum)) + (grp)) * EepFlashGroupSize)
```

The EEP driver each flash group base address.

Definition at line 102 of file Eep_Hal.c.

4.2.2.6 EEP_FLASH_GROUP_END

```
#define EEP_FLASH_GROUP_END(  
    blk,  
    grp )
```

Value:

```
(EepConfigPtr->EepFlashBaseAddr + \  
    (((uint32) (blk) * (uint32) (EepConfigPtr->  
    EepGroupNum)) + (grp) + 1U) * EepFlashGroupSize))
```

The EEP driver each flash group end address.

Definition at line 106 of file Eep_Hal.c.

4.2.2.7 EEP_FLASH_MAX_UNIT_SIZE

```
#define EEP_FLASH_MAX_UNIT_SIZE (16U)
```

EEP flash max program unit size.

Definition at line 76 of file Eep_Hal.c.

4.2.2.8 EEP_GROUP_FLAG_VALUE

```
#define EEP_GROUP_FLAG_VALUE (0x00000000U)
```

EEP flash operate group flag.

Definition at line 61 of file Eep_Hal.c.

4.2.2.9 EEP_GROUP_LOGIC_ADDR_END

```
#define EEP_GROUP_LOGIC_ADDR_END(  
    blk ) (((uint32) (blk) + 1U) * (uint32) (EepConfigPtr->EepBlockSize))
```

The EEP driver each group logic end address.

Definition at line 90 of file Eep_Hal.c.

4.2.2.10 EEP_GROUP_LOGIC_ADDR_START

```
#define EEP_GROUP_LOGIC_ADDR_START(  
    blk ) ((uint32)(blk) * (uint32)(EepConfigPtr->EepBlockSize))
```

The EEP driver each group logic base address.

Definition at line 87 of file Eep_Hal.c.

4.2.2.11 EEP_GROUP_MAX_NUM

```
#define EEP_GROUP_MAX_NUM (4U)
```

EEP group max number.

Definition at line 73 of file Eep_Hal.c.

4.2.2.12 EEP_LOGIC_ADDR_FROM_FLASH

```
#define EEP_LOGIC_ADDR_FROM_FLASH(  
    addr ) ((addr)& 0xffU)
```

The EEP ram address to logic address.

Definition at line 99 of file Eep_Hal.c.

4.2.2.13 EEP_LOGIC_ADDR_TO_RAM

```
#define EEP_LOGIC_ADDR_TO_RAM(  
    addr ) ((addr) + EepConfigPtr->EepRamBaseAddr)
```

The EEP ram address to logic address.

Definition at line 96 of file Eep_Hal.c.

4.2.2.14 EEP_MAX_SIZE

```
#define EEP_MAX_SIZE (4096U)
```

EEP max size in byte.

Definition at line 67 of file Eep_Hal.c.

4.2.2.15 EEP_MIN_SIZE

```
#define EEP_MIN_SIZE (512U)
```

EEP min size in byte.

Definition at line 64 of file Eep_Hal.c.

4.2.2.16 EEP_RAM_ADDR_TO_LOGIC

```
#define EEP_RAM_ADDR_TO_LOGIC(  
    addr ) ((addr) - EepConfigPtr->EepRamBaseAddr)
```

The EEP ram address to logic address.

Definition at line 93 of file Eep_Hal.c.

4.2.2.17 EEP_RAM_BLOCK_BASE

```
#define EEP_RAM_BLOCK_BASE(  
    blk )
```

Value:

```
(EepConfigPtr->EepRamBaseAddr + \  
    EepBlockSize)) ((uint32) (blk) * (uint32) (EepConfigPtr->
```

The EEP driver each RAM block base address.

Definition at line 79 of file Eep_Hal.c.

4.2.2.18 EEP_RAM_BLOCK_END

```
#define EEP_RAM_BLOCK_END(  
    blk )
```

Value:

```
(EepConfigPtr->EepRamBaseAddr + \  
    EepBlockSize)) (((uint32) (blk) + 1U) * (uint32) (EepConfigPtr->
```

The EEP driver each RAM block end address.

Definition at line 83 of file Eep_Hal.c.

4.2.2.19 EEP_XOR_PATTERN

```
#define EEP_XOR_PATTERN 0x5aa5U
```

Definition at line 112 of file Eep_Hal.c.

4.2.3 Function Documentation

4.2.3.1 Eep_Hal_AbortFreeRes()

```
void Eep_Hal_AbortFreeRes (  
    void )
```

Release the acquired resource when Eep_Hal is aborted abnormally.

Note

Function ID: DES_EEP_API_207

Definition at line 1128 of file Eep_Hal.c.

4.2.3.2 Eep_Hal_Erase()

```
Hal_StatusType Eep_Hal_Erase (  
    void )
```

Erase entire EEP storage space.

Erase Flash page with verification.

Returns

Operation status (STATUS_SUCCESS/STATUS_ERROR)

Note

Function ID: DES_EEP_API_203

Warning

Requires EEP_SDK_NON_EXTENDED_API undefined

Remarks

Performs:

1. Flash sector erase
2. RAM mirror refresh
3. Status management

Definition at line 979 of file Eep_Hal.c.

4.2.3.3 Eep_Hal_GetStatus()

```
Eep_Hal_StatusType Eep_Hal_GetStatus (
    void )
```

Retrieve EEP driver status.

Returns

Current driver state

Note

Function ID: DES_EEP_API_204

Return values

<i>EEP_HAL_UNINIT</i>	Driver not initialized
<i>EEP_HAL_INITIALIZED</i>	Initialized but inactive
<i>EEP_HAL_BUSY</i>	Write operation in progress
<i>EEP_HAL_IDLE</i>	Ready for operations

Definition at line 1118 of file Eep_Hal.c.

4.2.3.4 EEP_Hal_GetVersion()

```
void EEP_Hal_GetVersion (
    Eep_Hal_VersionType * Version )
```

Get driver version information.

Parameters

out	<i>Version</i>	Pointer to version structure
-----	----------------	------------------------------

Note

Function ID: DES_EEP_API_206

Remarks

Populates:

- Major version
- Minor version
- Patch version

Definition at line 1144 of file Eep_Hal.c.

4.2.3.5 Eep_Hal_Init()

```
Hal_StatusType Eep_Hal_Init (
    const Eep_Hal_ConfigType * ConfigPtr )
```

Initialize EEP driver.

Parameters

in	<i>ConfigPtr</i>	Pointer to configuration structure
----	------------------	------------------------------------

Returns

Initialization status

Note

Function ID: DES_EEP_API_200

Warning

Configuration must persist after initialization

Remarks

Performs:

1. Parameter validation
2. Memory mapping setup
3. Initial data synchronization

Definition at line 1033 of file Eep_Hal.c.

4.2.3.6 Eep_Hal_Read()

```
Hal_StatusType Eep_Hal_Read (
    uint32 Addr,
    uint8 * Data,
    uint16 Size )
```

Main EEP read interface.

Parameters

in	<i>Addr</i>	Virtual EEP address
out	<i>Data</i>	Data buffer pointer
in	<i>Size</i>	Data length in bytes

Returns

Operation status

Note

Function ID: DES_EEP_API_201

Remarks

Direct memory read from RAM mirror

Definition at line 920 of file Eep_Hal.c.

4.2.3.7 Eep_Hal_Refresh()

```
Hal_StatusType Eep_Hal_Refresh (  
    void )
```

Refresh EEP data from flash.

Returns

Operation status

Note

Function ID: DES_EEP_API_205

Remarks

Internal mechanism for:

1. Data consistency recovery
2. Post-erase reinitialization
3. Multi-block synchronization

Definition at line 1102 of file Eep_Hal.c.

4.2.3.8 Eep_Hal_Write()

```
Hal_StatusType Eep_Hal_Write (  
    uint32 Addr,  
    const uint8 * Data,  
    uint16 Size )
```

Main EEP write interface.

Parameters

in	<i>Addr</i>	Virtual EEP address (offset within configured space)
in	<i>Data</i>	Pointer to source data buffer
in	<i>Size</i>	Data length in bytes

Returns

Operation status (STATUS_SUCCESS/STATUS_ERROR)

Note

Function ID: DES_EEP_API_202

Warning

Address range must be within configured EEP size

Remarks

Implements:

1. Address validation
2. Block-wise write operations
3. Status tracking

Definition at line 850 of file Eep_Hal.c.

4.3 Eep_Hal.h File Reference

This file provides EEP api.

```
#include "Device_Register.h"
```

Classes

- struct [Eep_Hal_ConfigType](#)
EEP configuration structure.
- struct [Eep_Hal_VersionType](#)
EEP version structure.

Typedefs

- typedef void(* [Eep_CallbackType](#)) (void)
Callback type for when current block have not enough buffer to storage.

Enumerations

- enum [Eep_Hal_StatusType](#) {
[EEP_HAL_UNINIT](#) = 0x00U, [EEP_HAL_INITIALIZED](#) = 0x01U, [EEP_HAL_IDLE](#) = 0x02U, [EEP_HAL_TIMEOUT](#) = 0x03U,
[EEP_HAL_BUSY](#) = 0x04U }
EEP operate status.

Functions

- Hal_StatusType [Eep_Hal_Init](#) (const [Eep_Hal_ConfigType](#) *ConfigPtr)
Initialize EEP driver.
- Hal_StatusType [Eep_Hal_Write](#) (uint32 Addr, const uint8 *Data, uint16 Size)
Main EEP write interface.
- Hal_StatusType [Eep_Hal_Read](#) (uint32 Addr, uint8 *Data, uint16 Size)
Main EEP read interface.
- Hal_StatusType [Eep_Hal_Erase](#) (void)
Erase Flash page with verification.
- Hal_StatusType [Eep_Hal_Refresh](#) (void)
Refresh EEP data from flash.
- [Eep_Hal_StatusType](#) [Eep_Hal_GetStatus](#) (void)
Retrieve EEP driver status.
- void [Eep_Hal_AbortFreeRes](#) (void)
Release the acquired resource when Eep_Hal is aborted abnormally.
- void [EEP_Hal_GetVersion](#) ([Eep_Hal_VersionType](#) *Version)
Get driver version information.

4.3.1 Detailed Description

This file provides EEP api.

4.3.2 Typedef Documentation

4.3.2.1 Eep_CallbackType

```
typedef void(* Eep_CallbackType) (void)
```

Callback type for when current block have not enough buffer to storage.

Definition at line 67 of file Eep_Hal.h.

4.3.3 Enumeration Type Documentation

4.3.3.1 Eep_Hal_StatusType

```
enum Eep\_Hal\_StatusType
```

EEP operate status.

Enumerator

EEP_HAL_UNINIT	EEP operate not initial.
EEP_HAL_INITIALIZED	EEP operate initialized.
EEP_HAL_IDLE	EEP operate idle when FLASH command has been finished.
EEP_HAL_TIMEOUT	EEP operate timeout when FLASH command excute timeout.
EEP_HAL_BUSY	EEP operate busy when FLASH command not finished.

Definition at line 72 of file Eep_Hal.h.

4.3.4 Function Documentation**4.3.4.1 Eep_Hal_AbortFreeRes()**

```
void Eep_Hal_AbortFreeRes (  
    void )
```

Release the acquired resource when Eep_Hal is aborted abnormally.

Note

Function ID: DES_EEP_API_207

Definition at line 1128 of file Eep_Hal.c.

4.3.4.2 Eep_Hal_Erase()

```
Hal_StatusType Eep_Hal_Erase (  
    void )
```

Erase Flash page with verification.

Parameters

in	<i>pageAddress</i>	Start address of Flash page to erase
----	--------------------	--------------------------------------

Returns

Operation status (STATUS_SUCCESS/STATUS_ERROR)

Note

Function ID: DES_EEP_API_207

Warning

Must be executed within flash operation critical section

Remarks

Operation sequence:

1. Verify page erase status
2. Perform erase if verification fails
3. Re-verify after erase operation

Erase Flash page with verification.

Returns

Operation status (STATUS_SUCCESS/STATUS_ERROR)

Note

Function ID: DES_EEP_API_203

Warning

Requires EEP_SDK_NON_EXTENDED_API undefined

Remarks

Performs:

1. Flash sector erase
2. RAM mirror refresh
3. Status management

Definition at line 979 of file Eep_Hal.c.

4.3.4.3 Eep_Hal_GetStatus()

```
Eep_Hal_StatusType Eep_Hal_GetStatus (  
    void )
```

Retrieve EEP driver status.

Returns

Current driver state

Note

Function ID: DES_EEP_API_204

Return values

<i>EEP_HAL_UNINIT</i>	Driver not initialized
<i>EEP_HAL_INITIALIZED</i>	Initialized but inactive
<i>EEP_HAL_BUSY</i>	Write operation in progress
<i>EEP_HAL_IDLE</i>	Ready for operations

Definition at line 1118 of file Eep_Hal.c.

4.3.4.4 EEP_Hal_GetVersion()

```
void EEP_Hal_GetVersion (
    Eep_Hal_VersionType * Version )
```

Get driver version information.

Parameters

out	<i>Version</i>	Pointer to version structure
-----	----------------	------------------------------

Note

Function ID: DES_EEP_API_206

Remarks

Populates:

- Major version
- Minor version
- Patch version

Definition at line 1144 of file Eep_Hal.c.

4.3.4.5 Eep_Hal_Init()

```
Hal_StatusType Eep_Hal_Init (
    const Eep_Hal_ConfigType * ConfigPtr )
```

Initialize EEP driver.

Parameters

in	<i>ConfigPtr</i>	Pointer to configuration structure
----	------------------	------------------------------------

Returns

Initialization status

Note

Function ID: DES_EEP_API_200

Warning

Configuration must persist after initialization

Remarks

Performs:

1. Parameter validation
2. Memory mapping setup
3. Initial data synchronization

Definition at line 1033 of file Eep_Hal.c.

4.3.4.6 Eep_Hal_Read()

```
Hal_StatusType Eep_Hal_Read (
    uint32 Addr,
    uint8 * Data,
    uint16 Size )
```

Main EEP read interface.

Parameters

in	<i>Addr</i>	Virtual EEP address
out	<i>Data</i>	Data buffer pointer
in	<i>Size</i>	Data length in bytes

Returns

Operation status

Note

Function ID: DES_EEP_API_201

Remarks

Direct memory read from RAM mirror

Definition at line 920 of file Eep_Hal.c.

4.3.4.7 Eep_Hal_Refresh()

```
Hal_StatusType Eep_Hal_Refresh (
    void )
```

Refresh EEP data from flash.

Returns

Operation status

Note

Function ID: DES_EEP_API_205

Remarks

Internal mechanism for:

1. Data consistency recovery
2. Post-erase reinitialization
3. Multi-block synchronization

Definition at line 1102 of file Eep_Hal.c.

4.3.4.8 Eep_Hal_Write()

```
Hal_StatusType Eep_Hal_Write (
    uint32 Addr,
    const uint8 * Data,
    uint16 Size )
```

Main EEP write interface.

Parameters

in	<i>Addr</i>	Virtual EEP address (offset within configured space)
in	<i>Data</i>	Pointer to source data buffer
in	<i>Size</i>	Data length in bytes

Returns

Operation status (STATUS_SUCCESS/STATUS_ERROR)

Note

Function ID: DES_EEP_API_202

Warning

Address range must be within configured EEP size

Remarks

Implements:

1. Address validation
2. Block-wise write operations
3. Status tracking

Definition at line 850 of file Eep_Hal.c.

Index

AC784xx_API_Reference_Manual_EEP.pdf, [6](#)
ALIGN_DOWN
 [Eep_Hal.c, 7](#)
ALIGN_UP
 [Eep_Hal.c, 7](#)

Callback
 [Eep_Hal_ConfigType, 3](#)

EEP_BLOCK_MAX_NUM
 [Eep_Hal.c, 8](#)
EEP_FLASH_ERASED_VALUE
 [Eep_Hal.c, 8](#)
EEP_FLASH_GROUP_BASE
 [Eep_Hal.c, 8](#)
EEP_FLASH_GROUP_END
 [Eep_Hal.c, 8](#)
EEP_FLASH_MAX_UNIT_SIZE
 [Eep_Hal.c, 9](#)
EEP_GROUP_FLAG_VALUE
 [Eep_Hal.c, 9](#)
EEP_GROUP_LOGIC_ADDR_END
 [Eep_Hal.c, 9](#)
EEP_GROUP_LOGIC_ADDR_START
 [Eep_Hal.c, 9](#)
EEP_GROUP_MAX_NUM
 [Eep_Hal.c, 10](#)
EEP_Hal_GetVersion
 [Eep_Hal.c, 13](#)
 [Eep_Hal.h, 20](#)
EEP_LOGIC_ADDR_FROM_FLASH
 [Eep_Hal.c, 10](#)
EEP_LOGIC_ADDR_TO_RAM
 [Eep_Hal.c, 10](#)
EEP_MAX_SIZE
 [Eep_Hal.c, 10](#)
EEP_MIN_SIZE
 [Eep_Hal.c, 10](#)
EEP_RAM_ADDR_TO_LOGIC
 [Eep_Hal.c, 11](#)
EEP_RAM_BLOCK_BASE
 [Eep_Hal.c, 11](#)
EEP_RAM_BLOCK_END
 [Eep_Hal.c, 11](#)
EEP_XOR_PATTERN
 [Eep_Hal.c, 11](#)
Eep_CallbackType
 [Eep_Hal.h, 17](#)
Eep_Hal.c, [6](#)
 ALIGN_DOWN, [7](#)
 ALIGN_UP, [7](#)
 EEP_BLOCK_MAX_NUM, [8](#)
 EEP_FLASH_ERASED_VALUE, [8](#)
 EEP_FLASH_GROUP_BASE, [8](#)
 EEP_FLASH_GROUP_END, [8](#)
 EEP_FLASH_MAX_UNIT_SIZE, [9](#)
 EEP_GROUP_FLAG_VALUE, [9](#)
 EEP_GROUP_LOGIC_ADDR_END, [9](#)
 EEP_GROUP_LOGIC_ADDR_START, [9](#)
 EEP_GROUP_MAX_NUM, [10](#)
 EEP_Hal_GetVersion, [13](#)
 EEP_LOGIC_ADDR_FROM_FLASH, [10](#)
 EEP_LOGIC_ADDR_TO_RAM, [10](#)
 EEP_MAX_SIZE, [10](#)
 EEP_MIN_SIZE, [10](#)
 EEP_RAM_ADDR_TO_LOGIC, [11](#)
 EEP_RAM_BLOCK_BASE, [11](#)
 EEP_RAM_BLOCK_END, [11](#)
 EEP_XOR_PATTERN, [11](#)
 Eep_Hal_AbortFreeRes, [12](#)
 Eep_Hal_Erase, [12](#)
 Eep_Hal_GetStatus, [12](#)
 Eep_Hal_Init, [13](#)
 Eep_Hal_Read, [14](#)
 Eep_Hal_Refresh, [15](#)
 Eep_Hal_Write, [15](#)
Eep_Hal.h, [16](#)
 EEP_Hal_GetVersion, [20](#)
 Eep_CallbackType, [17](#)
 Eep_Hal_AbortFreeRes, [18](#)
 Eep_Hal_Erase, [18](#)
 Eep_Hal_GetStatus, [19](#)
 Eep_Hal_Init, [20](#)
 Eep_Hal_Read, [21](#)
 Eep_Hal_Refresh, [21](#)
 Eep_Hal_StatusType, [17](#)
 Eep_Hal_Write, [22](#)
Eep_Hal_AbortFreeRes
 [Eep_Hal.c, 12](#)
 [Eep_Hal.h, 18](#)
Eep_Hal_ConfigType, [3](#)
 Callback, [3](#)
 EepBlockSize, [3](#)
 EepFlashBaseAddr, [4](#)
 EepGroupNum, [4](#)
 EepRamBaseAddr, [4](#)
 EepSize, [4](#)
Eep_Hal_Erase
 [Eep_Hal.c, 12](#)
 [Eep_Hal.h, 18](#)
Eep_Hal_GetStatus
 [Eep_Hal.c, 12](#)
 [Eep_Hal.h, 19](#)
Eep_Hal_Init

- Eep_Hal.c, [13](#)
 - Eep_Hal.h, [20](#)
- Eep_Hal_Read
 - Eep_Hal.c, [14](#)
 - Eep_Hal.h, [21](#)
- Eep_Hal_Refresh
 - Eep_Hal.c, [15](#)
 - Eep_Hal.h, [21](#)
- Eep_Hal_StatusType
 - Eep_Hal.h, [17](#)
- Eep_Hal_VersionType, [5](#)
 - Major, [5](#)
 - Minor, [5](#)
 - Patch, [5](#)
- Eep_Hal_Write
 - Eep_Hal.c, [15](#)
 - Eep_Hal.h, [22](#)
- EepBlockSize
 - Eep_Hal_ConfigType, [3](#)
- EepFlashBaseAddr
 - Eep_Hal_ConfigType, [4](#)
- EepGroupNum
 - Eep_Hal_ConfigType, [4](#)
- EepRamBaseAddr
 - Eep_Hal_ConfigType, [4](#)
- EepSize
 - Eep_Hal_ConfigType, [4](#)
- Major
 - Eep_Hal_VersionType, [5](#)
- Minor
 - Eep_Hal_VersionType, [5](#)
- Patch
 - Eep_Hal_VersionType, [5](#)