

# GLOBAL\_SCU\_XMC1

[Home](#)

## Apps

Here is a list of all modules:

- [License Terms and Copyright Information](#)
- [Abbreviations and Definitions](#)
- [Overview](#)
- [Architecture Description](#)
- [APP Configuration Parameters](#)
- [Enumerations](#)
- [Data structures](#)
- [Methods](#)
- [Usage](#)
- [Release History](#)
- [GLOBAL\\_SCU\\_XMC1](#)



# GLOBAL\_SCU\_XMC1

[Home](#)

## License Terms and Copyright Information

### License Terms and Copyright Information

Copyright (c) 2015, Infineon Technologies AG All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

To improve the quality of the software, users are encouraged to share modifications, enhancements or bug fixes with Infineon Technologies AG ([dave@infineon.com](mailto:dave@infineon.com)).

---

# GLOBAL\_SCU\_XMC1

Home

## Abbreviations and Definitions

### Abbreviations and Definitions

<b>Abbreviations:</b>	
DAVE™	Digital Application Virtual Engineer
APP	DAVE Application
API	Application Programming Interface
GUI	Graphical User Interface
MCU	Microcontroller Unit
SW	Software
HW	Hardware
LLD	Low Level Driver
SCU	System Control Unit
IO	Input Output

<b>Definitions:</b>	
Singleton	Only single instance of the APP is permitted
Sharable	Resource sharing with other APPs is permitted
initProvider	Provides the initialization routine
Physical connectivity	Hardware inter/intra peripheral (constant) signal connection
Conditional connectivity	Constrained hardware inter/intra peripheral signal connection
Aggregation	Indicates consumption of low level (dependent) APPs

---

--

# GLOBAL\_SCU\_XMC1

Home

## Overview

### Overview

The **GLOBAL\_SCU\_XMC1** APP is a system APP. The **GLOBAL\_SCU\_XMC1** APP provides functionality to register an event callback routine to be executed on occurrence of a certain event. Other APPs or user code in the project can make use of this functionality.

The **GLOBAL\_SCU\_XMC1** APP handles the following events:

#### **SCU\_IRQ0:**

- Flash ECC double bit error
- Flash operation completion
- PSRAM Parity error
- USIC0 Parity error
- USIC1 Parity error (only XMC14xx series)
- MultiCAN SRAM Parity Error (only XMC14xx series)
- Loss of clock
- Loss of external OSC\_HP clock (only XMC14xx series)

#### **SCU\_IRQ1:**

- Standby clock failure
- DCO1 Out of SYNC (only XMC14xx series)
- VDDP pre-warning
- VDROP
- VCLIP
- Temperature measurement Completion
- Temperature too high
- Temperature too low
- WDT pre-warning

- RTC periodic
- RTC alarm
- RTCCTR register update
- RTCATIM0 register update
- RTCATIM1 register update
- RTCTIM0 register update
- RTCTIM1 register update

### **SCU\_IRQ2:**

- Out of range comparator [0-7] (Not available in XMC11xx series)
- Analog comparator [0-3] (Not available in XMC11xx series. ACMP3 only available in XMC14xx series)

Other APPs or user code can register an event callback routine using GLOBAL\_SCU\_XMC1\_RegisterCallback function.

The **GLOBAL\_SCU\_XMC1** APP uses the XMC\_SCU\_IRQHandler, found in the XMC Peripheral Library, to handle the event and calling the registered event callback routine.

### **Supported Devices**

1. XMC1400 Series
2. XMC1300 Series
3. XMC1200 Series
4. XMC1100 Series

### **References**

1. XMC1400 Reference Manual
2. XMC1300 Reference Manual
3. XMC1200 Reference Manual
4. XMC1100 Reference Manual

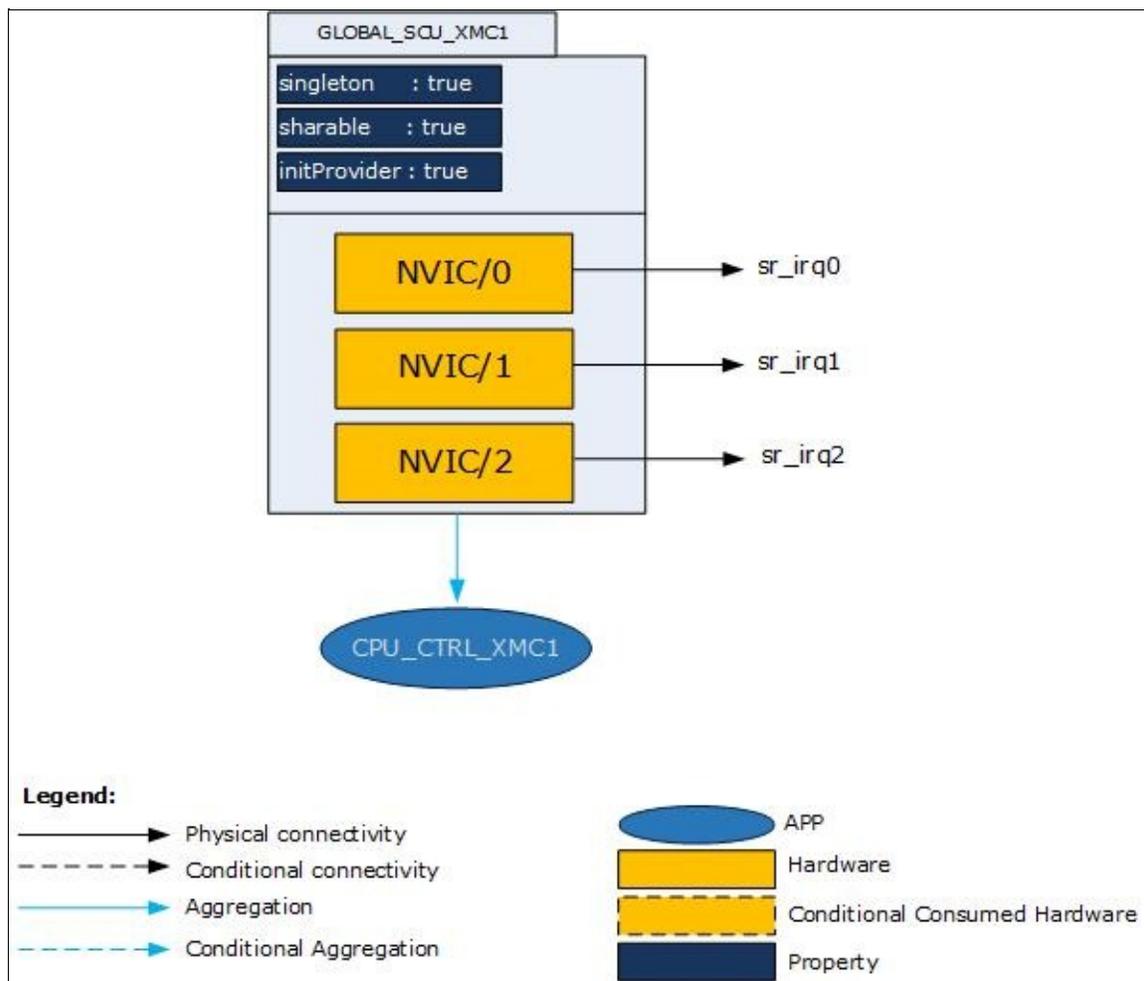


# GLOBAL\_SCU\_XMC1

Home

## Architecture Description

### Architecture Description



**Figure 1** : Architecture of **GLOBAL\_SCU\_XMC1** APP

The above diagram represents the internal software architecture of the **GLOBAL\_SCU\_XMC1** APP. A **GLOBAL\_SCU\_XMC1** APP instance exists in a DAVE™ project with fixed attributes as shown above. This APP consumes three NVIC nodes of the MCU. In addition, it requires

the consumption of the CPU\_CTRL\_XMC1 APP for its configuration and functioning.

An instantiated APP (after code generation) generates a specific data structure with the GUI configuration. The name of this data structure can be modified by changing the APP instance label (e.g. change label from default GLOBAL\_SCU\_XMC1\_0 to MY\_GLOBAL\_SCU).

### Signals:

The following table describes the list of IO signals for **GLOBAL\_SCU\_XMC1** APP.

**Table 1:** APP Input Output signals

Signal Name	Input/Output	Availability	Description
sr_irq0	Input	Always	This signal is reserved to NVIC Node 0.
sr_irq1	Input	Always	This signal is reserved to NVIC Node 1.
sr_irq2	Input	Always	This signal is reserved to NVIC Node 2.



# GLOBAL\_SCU\_XMC1

Home

## APP Configuration Parameters

### App Configuration Parameters

Interrupt Settings

Service Request 0 Settings

Enable interrupt at initialization

Interrupt Priority

Preemption priority

Service Request 1 Settings

Enable interrupt at initialization

Interrupt Priority

Preemption priority

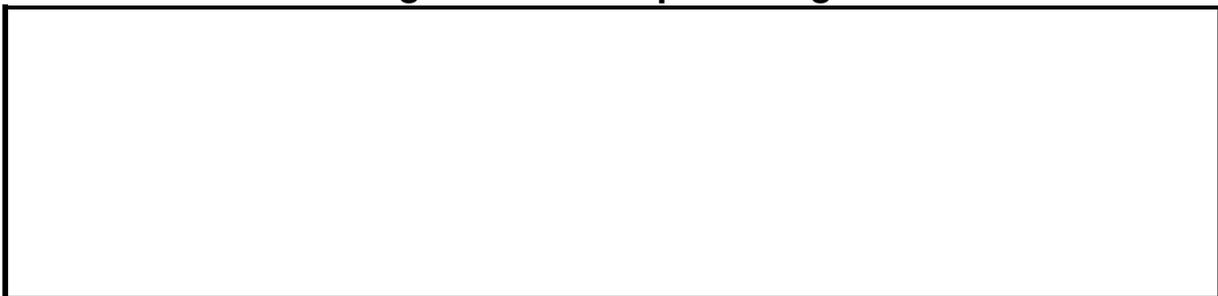
Service Request 2 Settings

Enable interrupt at initialization

Interrupt Priority

Preemption priority

Figure 1: Interrupt Settings





# GLOBAL\_SCU\_XMC1

[Home](#)

## Enumerations

```
enum GLOBAL_SCU_XMC1
GLOBAL_SCU_XMC1
= 0U,
GLOBAL_SCU_XMC1
= 1U }
typedef enum GLOBAL_SCU_XMC1_STATUS GLOBAL_SCU_XMC1
```

## Typedef Documentation

**typedef enum GLOBAL\_SCU\_XMC1\_STATUS GLOBAL\_SCU\_XMC1**

APP status

---

## Enumeration Type Documentation

### enum GLOBAL\_SCU\_XMC1\_STATUS

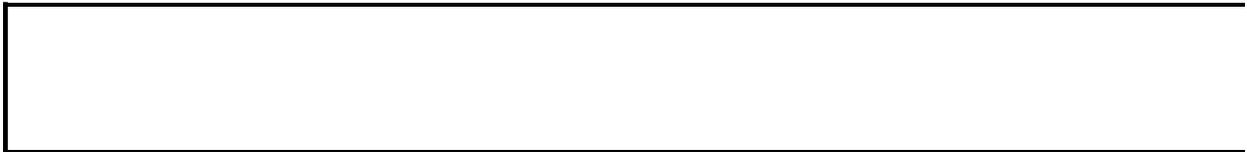
APP status

#### Enumerator:

*GLOBAL\_SCU\_XMC1\_STATUS\_SUCCESS* Initialization  
status success

*GLOBAL\_SCU\_XMC1\_STATUS\_FAILURE* Initialization  
status failure

Definition at line **140** of file **GLOBAL\_SCU\_XMC1.h**.



# GLOBAL\_SCU\_XMC1

[Home](#)

## Data structures

---

<code>typedef</code>	<code>GLOBAL_SCU_XMC:</code>
<code>XMC_SCU_INTERRUPT_EVENT_HANDLER_t</code>	
<code>typedef XMC_SCU_INTERRUPT_EVENT_t</code>	<code>GLOBAL_SCU_XMC:</code>
<code>typedef struct</code>	<code>GLOBAL_SCU_XMC:</code>
<code>GLOBAL_SCU_XMC1_CONFIG</code>	
<code>typedef struct GLOBAL_SCU_XMC1</code>	<code>GLOBAL_SCU_XMC:</code>

---

## Typedef Documentation

**typedef struct GLOBAL\_SCU\_XMC1\_CONFIG GLOBAL\_SCU\_XMC1\_CONFIG\_t**

Configuration parameters

**typedef XMC\_SCU\_INTERRUPT\_EVENT\_HANDLER\_t GLOBAL\_SCU\_INTERRUPT\_EVENT\_HANDLER\_t**

Event handler type

Definition at line **160** of file **GLOBAL\_SCU\_XMC1.h**.

**typedef XMC\_SCU\_INTERRUPT\_EVENT\_t GLOBAL\_SCU\_XMC1\_INTERRUPT\_EVENT\_t**

Event type

Definition at line **165** of file **GLOBAL\_SCU\_XMC1.h**.

**typedef struct GLOBAL\_SCU\_XMC1 GLOBAL\_SCU\_XMC1\_t**

Runtime handler



# GLOBAL\_SCU\_XMC1

Home

## Methods

<code>DAVE_APP_VERSION_t</code>	<b>GLOBAL_SCU_XMC1_GetAppVer</b> (void) Get <b>GLOBAL_SCU_XMC1</b> APP ve
<code>GLOBAL_SCU_XMC1_STATUS_t</code>	<b>GLOBAL_SCU_XMC1_Init</b> ( <b>GLOBAL_SCU_XMC1_t</b> *const h) Initialize the <b>GLOBAL_SCU_XMC1</b>
<code>GLOBAL_SCU_XMC1_STATUS_t</code>	<b>GLOBAL_SCU_XMC1_RegisterC</b> (const <b>GLOBAL_SCU_XMC1_EVE</b> event, const <b>GLOBAL_SCU_XMC1_EVENT_H</b> callback) Register an event callback routine function registers an event callback for a specific event. Note: User nee enable the event generation separa <b>GLOBAL_SCU_XMC1_EnableEve</b> <b>GLOBAL_SCU_XMC1_DisableEve</b>
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_EnableIRQ</b> (irq_num) Enable the handling of events.
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_DisableIRQ</b> (irq_num) Disable the handling of events.
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_EnableEve</b> <b>GLOBAL_SCU_XMC1_EVENT_t</b> e Enable a specific event.
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_DisableEve</b> <b>GLOBAL_SCU_XMC1_EVENT_t</b> e

Disable a specific event.

## Methods

---

## Function Documentation

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_DisableEvent ( const**

Disable a specific event.

**Parameters:**

**event** Events to be disabled. See  
GLOBAL\_SCU\_XMC1\_EVENT\_t.

Definition at line **317** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_DisableIRQ ( uint8\_t**

Disable the handling of events.

**Parameters:**

**void**

Definition at line **297** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_EnableEvent ( const**

Enable a specific event.

**Parameters:**

**event** Events to be enabled. See  
GLOBAL\_SCU\_XMC1\_EVENT\_t.

Definition at line **307** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_EnableIRQ ( uint8\_t i**

Enable the handling of events.

**Parameters:**

**void**

Definition at line **287** of file **GLOBAL\_SCU\_XMC1.h**.

**DAVE\_APP\_VERSION\_t GLOBAL\_SCU\_XMC1\_GetAppVersion ( voi**

Get **GLOBAL\_SCU\_XMC1** APP version.

**Returns:**

DAVE\_APP\_VERSION\_t APP version details (major, minor and patch number)

**Description:**

The function can be used to check application software compatibility with a specific version of the APP.

Example Usage:

```
#include <DAVE.h>

int main(void)
{
    DAVE_APP_VERSION_t version;

    // Initialize GLOBAL_SCU_XMC1 APP.
    DAVE_Init();

    version = GLOBAL_SCU_XMC1_GetAppVersion();
    if (version.major != 4U)
    {
        // Probably, not the right version.
    }
}
```

```
// More code here
while(1)
{
}
return (0);
}
```

Definition at line **91** of file **GLOBAL\_SCU\_XMC1.c**.

## **GLOBAL\_SCU\_XMC1\_STATUS\_t GLOBAL\_SCU\_XMC1\_Init ( GLOB**

Initialize the **GLOBAL\_SCU\_XMC1** APP.

This function initializes the required CPU\_XMC1 APP if not yet done, sets the priority of the SCU\_0/SCU\_1/SCU\_2 IRQ, and enables the IRQ if the user has selected so. This function is called from DAVE\_Init() or top level APP requiring the **GLOBAL\_SCU\_XMC1** APP.

### **Parameters:**

**handle** Runtime handler

### **Returns:**

**GLOBAL\_SCU\_XMC1\_STATUS\_t** Indicates initialization status

Definition at line **105** of file **GLOBAL\_SCU\_XMC1.c**.

References **GLOBAL\_SCU\_XMC1::config**, **GLOBAL\_SCU\_XMC1\_CONFIG::enable\_at\_init**, **GLOBAL\_SCU\_XMC1\_STATUS\_FAILURE**, **GLOBAL\_SCU\_XMC1\_STATUS\_SUCCESS**, **GLOBAL\_SCU\_XMC1::initialized**, and **GLOBAL\_SCU\_XMC1\_CONFIG::priority**.

## GLOBAL\_SCU\_XMC1\_STATUS\_t GLOBAL\_SCU\_XMC1\_RegisterCa

Register an event callback routine This function registers an event callback routine for a specific event. Note: User needs to enable the event generation separately. See GLOBAL\_SCU\_XMC1\_EnableEvent and GLOBAL\_SCU\_XMC1\_DisableEvent.

### Parameters:

**event** Event for which the callback is register  
**callback** Routine to handle occurrence of event

### Returns:

GLOBAL\_SCU\_XMC1\_STATUS\_t Indicates registration status

```
#include <DAVE.h>

void CbListener(void)
{
    // Callback function implementation
};

int main(void)
{
    DAVE_Init();

    GLOBAL_SCU_XMC1_RegisterCallback(GLOBAL_SCU_XM
C1_EVENT_RTC_ALARM, CbListener);
    GLOBAL_SCU_XMC1_EnableEvent(GLOBAL_SCU_XMC1_EV
ENT_RTC_ALARM);
    while(1);
}
```

Definition at line **214** of file **GLOBAL\_SCU\_XMC1.c**.

---

---

# GLOBAL\_SCU\_XMC1

Home

## Usage

### Usage

**GLOBAL\_SCU\_XMC1** is a global DAVE™ APP. It is consumed by the RTC and other (top level) APPs. For information on how **GLOBAL\_SCU\_XMC1** is being used, refer for example to the RTC, WATCHDOG related APPs help documentation.



# GLOBAL\_SCU\_XMC1

Home

## Release History

### Release History

--

--

# GLOBAL\_SCU\_XMC1

[Home](#)

[Data Structures](#) | [Defines](#)

## GLOBAL\_SCU\_XMC1

## Data Structures

---

struct **GLOBAL\_SCU\_XMC1\_CONFIG**

---

struct **GLOBAL\_SCU\_XMC1**



	Register an event callback needs to enable the event GLOBAL_SCU_XMC1_Di
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_E</b> Enable the handling of eve
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_D</b> Disable the handling of ev
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_E</b> Enable a specific event.
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1_D</b> Disable a specific event.

## Define Documentation

**#define GLOBAL\_SCU\_XMC1\_EVENT\_ACMP3 XMC\_SCU\_INTERF**

Analog comparator-3 output event.

Definition at line **127** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_DCO1\_OUT\_SYNC XMC\_S**

DCO1 Out of SYNC Event.

Definition at line **126** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_COMPLETED XMC**

Flash operation completion event.

Definition at line **110** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_ERROR XMC\_SCU**

Flash ECC double bit error event.

Definition at line **109** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_LOCI XMC\_SCU\_INTERRU**

Loss of clock event.

Definition at line **103** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_LOSS\_EXT\_CLOCK XMC\_SCU\_INTEF**

Loss of external OSC\_HP clock event.

Definition at line **125** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_PESRAM XMC\_SCU\_INTEF**

PSRAM Parity error event.

Definition at line **104** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_PEUSIC0 XMC\_SCU\_INTEF**

USIC0 Parity error event.

Definition at line **105** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_ALARM XMC\_SCU\_INTEF**

RTC alarm event.

Definition at line **84** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_PERIODIC XMC\_SCU\_INTEF**

RTC periodic event.

Definition at line **83** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_RTCATIM0\_UPDATED XMC\_SCU\_INTEF**

RTCATIM0 register update event.

Definition at line [114](#) of file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#define GLOBAL_SCU_XMC1_EVENT_RTCATIM1_UPDATED XMC
```

RTCATIM1 register update event.

Definition at line [115](#) of file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#define GLOBAL_SCU_XMC1_EVENT_RTCCTR_UPDATED XMC_!
```

RTCCTR register update event.

Definition at line [113](#) of file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#define GLOBAL_SCU_XMC1_EVENT_RTCTIM0_UPDATED XMC_
```

RTCTIM0 register update event.

Definition at line [116](#) of file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#define GLOBAL_SCU_XMC1_EVENT_RTCTIM1_UPDATED XMC_
```

RTCTIM1 register update event.

Definition at line [117](#) of file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#define GLOBAL_SCU_XMC1_EVENT_STDBYCLKFAIL XMC_SCU
```

Standby clock failure event.

Definition at line **112** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_DONE XMC\_SCU\_INT**

Temperature measurement Completion event.

Definition at line **118** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_HIGH XMC\_SCU\_INTE**

Temperature too high event.

Definition at line **119** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_LOW XMC\_SCU\_INTE**

Temperature too low event.

Definition at line **120** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_VCLIP XMC\_SCU\_INTERRI**

VCLIP event.

Definition at line **111** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_VDDPI XMC\_SCU\_INTERRI**

VDDP pre-warning event.

Definition at line **85** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_VDROP XMC\_SCU\_INTERF**

VDROP event.

Definition at line **92** of file **GLOBAL\_SCU\_XMC1.h**.

**#define GLOBAL\_SCU\_XMC1\_EVENT\_WDT\_WARNING XMC\_SCU**

WDT pre-warning event.

Definition at line **82** of file **GLOBAL\_SCU\_XMC1.h**.

---

## Typedef Documentation

**typedef struct GLOBAL\_SCU\_XMC1\_CONFIG GLOBAL\_SCU\_XMC1\_CONFIG\_t**

Configuration parameters

**typedef XMC\_SCU\_INTERRUPT\_EVENT\_t GLOBAL\_SCU\_XMC1\_EVENT\_t**

Event type

Definition at line **165** of file **GLOBAL\_SCU\_XMC1.h**.

**typedef struct GLOBAL\_SCU\_XMC1 GLOBAL\_SCU\_XMC1\_t**

Runtime handler

---

## Function Documentation

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_DisableEvent ( const**

Disable a specific event.

**Parameters:**

**event** Events to be disabled. See  
GLOBAL\_SCU\_XMC1\_EVENT\_t.

Definition at line **317** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_DisableIRQ ( uint8\_t**

Disable the handling of events.

**Parameters:**

**void**

Definition at line **297** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_EnableEvent ( const**

Enable a specific event.

**Parameters:**

**event** Events to be enabled. See  
GLOBAL\_SCU\_XMC1\_EVENT\_t.

Definition at line **307** of file **GLOBAL\_SCU\_XMC1.h**.

**\_\_STATIC\_INLINE void GLOBAL\_SCU\_XMC1\_EnableIRQ ( uint8\_t i**

Enable the handling of events.

**Parameters:**

**void**

Definition at line **287** of file **GLOBAL\_SCU\_XMC1.h**.

## **GLOBAL\_SCU\_XMC1\_STATUS\_t GLOBAL\_SCU\_XMC1\_Init ( GLOB**

Initialize the **GLOBAL\_SCU\_XMC1** APP.

This function initializes the required CPU\_XMC1 APP if not yet done, sets the priority of the SCU\_0/SCU\_1/SCU\_2 IRQ, and enables the IRQ if the user has selected so. This function is called from DAVE\_Init() or top level APP requiring the **GLOBAL\_SCU\_XMC1** APP.

**Parameters:**

**handle** Runtime handler

**Returns:**

**GLOBAL\_SCU\_XMC1\_STATUS\_t** Indicates initialization status

Definition at line **105** of file **GLOBAL\_SCU\_XMC1.c**.

References **config**,

**GLOBAL\_SCU\_XMC1\_CONFIG::enable\_at\_init**,

**GLOBAL\_SCU\_XMC1\_STATUS\_FAILURE**,

**GLOBAL\_SCU\_XMC1\_STATUS\_SUCCESS**, **initialized**, and

**GLOBAL\_SCU\_XMC1\_CONFIG::priority**.

## **GLOBAL\_SCU\_XMC1\_STATUS\_t GLOBAL\_SCU\_XMC1\_RegisterCa**

Register an event callback routine This function registers an event callback routine for a specific event. Note: User needs to enable the event generation separately. See GLOBAL\_SCU\_XMC1\_EnableEvent and GLOBAL\_SCU\_XMC1\_DisableEvent.

**Parameters:**

**event** Event for which the callback is register  
**callback** Routine to handle occurrence of event

**Returns:**

GLOBAL\_SCU\_XMC1\_STATUS\_t Indicates registration status

```
#include <DAVE.h>

void CbListener(void)
{
    // Callback function implementation
};

int main(void)
{
    DAVE_Init();

    GLOBAL_SCU_XMC1_RegisterCallback(GLOBAL_SCU_XM
C1_EVENT_RTC_ALARM, CbListener);
    GLOBAL_SCU_XMC1_EnableEvent(GLOBAL_SCU_XMC1_EV
ENT_RTC_ALARM);
    while(1);
}
```

Definition at line **214** of file **GLOBAL\_SCU\_XMC1.c**.



# GLOBAL\_SCU\_XMC1

[Home](#)

[Data Structures](#)

[Data Structure Index](#)

[Data Fields](#)

## Data Structures

Here are the data structures with brief descriptions:

<a href="#">GLOBAL_SCU_XMC1</a>	
---------------------------------	--

<a href="#">GLOBAL_SCU_XMC1_CONFIG</a>	
--	--

--

# GLOBAL\_SCU\_XMC1

[Home](#)

[Data Structures](#)

[Data Structure Index](#)

[Data Fields](#)

[Data Fields](#)

## GLOBAL\_SCU\_XMC1

### Struct Reference

[GLOBAL\\_SCU\\_XMC1](#)

---

## Detailed Description

Runtime handler

Definition at line **181** of file **GLOBAL\_SCU\_XMC1.h**.

```
#include <GLOBAL_SCU_XMC1.h>
```

## Data Fields

---

const	<b>GLOBAL_SCU_XMC1_CONFIG_t</b>	*	<b>config</b>
			<b>bool initialized</b>

---

## Field Documentation

**const GLOBAL\_SCU\_XMC1\_CONFIG\_t\* GLOBAL\_SCU\_XMC1::cor**

Instance configuration options

Definition at line **183** of file **GLOBAL\_SCU\_XMC1.h**.

Referenced by **GLOBAL\_SCU\_XMC1\_Init()**.

**bool GLOBAL\_SCU\_XMC1::initialized**

Initialization status

Definition at line **184** of file **GLOBAL\_SCU\_XMC1.h**.

Referenced by **GLOBAL\_SCU\_XMC1\_Init()**.

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

- **GLOBAL\_SCU\_XMC1.h**



# GLOBAL\_SCU\_XMC1

[Home](#)

[Data Structures](#)

[Data Structure Index](#)

[Data Fields](#)

[Data Fields](#)

## GLOBAL\_SCU\_XMC1\_CONFIG Struct Reference

[GLOBAL\\_SCU\\_XMC1](#)

---

## Detailed Description

Configuration parameters

Definition at line **171** of file **GLOBAL\_SCU\_XMC1.h**.

```
#include <GLOBAL_SCU_XMC1.h>
```

## Data Fields

---

uint8\_t **priority** [GLOBAL\_SCU\_XMC1\_IRQNUM]

bool **enable\_at\_init** [GLOBAL\_SCU\_XMC1\_IRQNUM]

---

## Field Documentation

**bool GLOBAL\_SCU\_XMC1\_CONFIG::enable\_at\_init**[GLOBAL\_SCU\_XMC1\_CONFIG]

Interrupt enable for Node

Definition at line **174** of file **GLOBAL\_SCU\_XMC1.h**.

Referenced by **GLOBAL\_SCU\_XMC1\_Init()**.

**uint8\_t GLOBAL\_SCU\_XMC1\_CONFIG::priority**[GLOBAL\_SCU\_XMC1\_CONFIG]

Node Interrupt Priority

Definition at line **173** of file **GLOBAL\_SCU\_XMC1.h**.

Referenced by **GLOBAL\_SCU\_XMC1\_Init()**.

---

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

- **GLOBAL\_SCU\_XMC1.h**



# GLOBAL\_SCU\_XMC1

Home

Data Structures

Data Structure Index

Data Fields

## Data Structure Index

G

**G**

GLOBAL\_SCU\_XMC1\_CONFIG

GLOBAL\_SCU\_XMC1

G

--

# GLOBAL\_SCU\_XMC1

<a href="#">Home</a>			
<a href="#">Data Structures</a>	<a href="#">Data Structure Index</a>	<a href="#">Data Fields</a>	
<a href="#">All</a>	<a href="#">Variables</a>		

Here is a list of all documented struct and union fields with links to the struct/union documentation for each field:

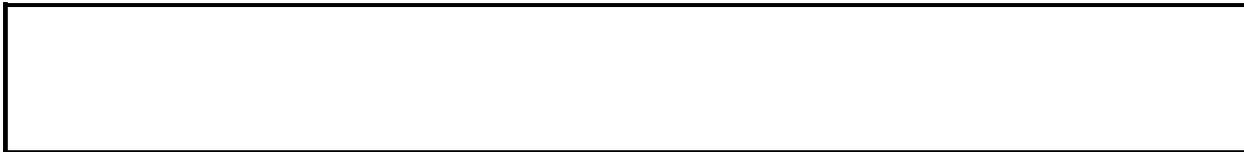
- config : [GLOBAL\\_SCU\\_XMC1](#)
- enable\_at\_init : [GLOBAL\\_SCU\\_XMC1\\_CONFIG](#)
- initialized : [GLOBAL\\_SCU\\_XMC1](#)
- priority : [GLOBAL\\_SCU\\_XMC1\\_CONFIG](#)



# GLOBAL\_SCU\_XMC1

Home			
Data Structures	Data Structure Index	Data Fields	
All	Variables		

- config : **GLOBAL\_SCU\_XMC1**
- enable\_at\_init : **GLOBAL\_SCU\_XMC1\_CONFIG**
- initialized : **GLOBAL\_SCU\_XMC1**
- priority : **GLOBAL\_SCU\_XMC1\_CONFIG**



# GLOBAL\_SCU\_XMC1

[Home](#)

[File List](#)

[Globals](#)

## File List

Here is a list of all documented files with brief descriptions:

[GLOBAL\\_SCU\\_XMC1.c \[code\]](#)

[GLOBAL\\_SCU\\_XMC1.h \[code\]](#)



# GLOBAL\_SCU\_XMC1

[Home](#)

[File List](#)

[Globals](#)

[Functions](#)

## GLOBAL\_SCU\_XMC1.c

### File Reference

---

## Detailed Description

**Date:**

2015-07-30

NOTE: This file is generated by DAVE. Any manual modification done to this file will be lost when the code is regenerated.

Definition in file [GLOBAL\\_SCU\\_XMC1.c](#).

```
#include "global_scu_xmc1.h"
```

## Functions

<code>DAVE_APP_VERSION_t</code>	<b><code>GLOBAL_SCU_XMC1_GetAppVe</code></b> (void) Get <b><code>GLOBAL_SCU_XMC1</code></b> APP ve
<b><code>GLOBAL_SCU_XMC1_STATUS_t</code></b>	<b><code>GLOBAL_SCU_XMC1_Init</code></b> ( <b><code>GLOBAL_SCU_XMC1_t</code></b> *const h) Initialize the <b><code>GLOBAL_SCU_XMC1</code></b>
<b><code>GLOBAL_SCU_XMC1_STATUS_t</code></b>	<b><code>GLOBAL_SCU_XMC1_RegisterC</code></b> (const <b><code>GLOBAL_SCU_XMC1_EVE</code></b> event, const <b><code>GLOBAL_SCU_XMC1_EVENT_H</code></b> handler) Register an event callback routine function registers an event callback for a specific event. Note: User nee enable the event generation separa <b><code>GLOBAL_SCU_XMC1_EnableEve</code></b> <b><code>GLOBAL_SCU_XMC1_DisableEve</code></b>

[Go to the source code of this file.](#)



# GLOBAL\_SCU\_XMC1

[Home](#)

[File List](#)

[Globals](#)

[Data Structures](#) | [Defines](#)

## GLOBAL\_SCU\_XMC1.h File Reference

---

## Detailed Description

**Date:**

2015-07-30

NOTE: This file is generated by DAVE. Any manual modification done to this file will be lost when the code is regenerated.

Definition in file [GLOBAL\\_SCU\\_XMC1.h](#).

```
#include <xmc_scu.h> #include <DAVE_Common.h>
#include "CPU_CTRL_XMC1\cpu_ctrl_xmc1.h"
#include "global_scu_xmc1_conf.h"
#include "global_scu_xmc1_extern.h"
```

## Data Structures

---

struct **GLOBAL\_SCU\_XMC1\_CONFIG**

---

struct **GLOBAL\_SCU\_XMC1**

## Defines

```
#define GLOBAL_SCU_XMC1_EVENT_WDT_WARNING XMC_SCU
#define GLOBAL_SCU_XMC1_EVENT_RTC_PERIODIC XMC_SCU
#define GLOBAL_SCU_XMC1_EVENT_RTC_ALARM XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_VDDPI XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_VDROP XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_LOCI XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_PESRAM XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_PEUSIC0 XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_FLASH_ERROR XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_FLASH_COMPLETED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_VCLIP XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_STDBYCLKFAIL XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_RTCCTR_UPDATED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_RTCATIM0_UPDATED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_RTCATIM1_UPDATED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_RTCTIM0_UPDATED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_RTCTIM1_UPDATED XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_TSE_DONE XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_TSE_HIGH XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_TSE_LOW XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_LOSS_EXT_CLOCK XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_DCO1_OUT_SYNC XMC_SCU_INTERRUPT
#define GLOBAL_SCU_XMC1_EVENT_ACOMP3 XMC_SCU_INTERRUPT
```

## Typedefs

<code>typedef</code>	<code>GLOBAL_SCU_XMC:</code>
<code>XMC_SCU_INTERRUPT_EVENT_HANDLER_t</code>	
<code>typedef XMC_SCU_INTERRUPT_EVENT_t</code>	<code>GLOBAL_SCU_XMC:</code>
<code>typedef struct</code>	<code>GLOBAL_SCU_XMC:</code>
<code>GLOBAL_SCU_XMC1_CONFIG</code>	
<code>typedef struct GLOBAL_SCU_XMC1</code>	<code>GLOBAL_SCU_XMC:</code>

## Functions

<code>DAVE_APP_VERSION_t</code>	<b>GLOBAL_SCU_XMC1</b> (void) Get <b>GLOBAL_SCU_X</b>
<b>GLOBAL_SCU_XMC1_STATUS_t</b>	<b>GLOBAL_SCU_XMC1</b> ( <b>GLOBAL_SCU_XMC</b> Initialize the <b>GLOBAL_</b>
<b>GLOBAL_SCU_XMC1_STATUS_t</b>	<b>GLOBAL_SCU_XMC1</b> (const <b>GLOBAL_SCU_</b> event, const <b>GLOBAL_SCU_XMC1</b> callback) Register an event callb function registers an ev for a specific event. No enable the event gener <b>GLOBAL_SCU_XMC1.</b> <b>GLOBAL_SCU_XMC1.</b>
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1</b> irq_num) Enable the handling of
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1</b> irq_num) Disable the handling of
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1</b> <b>GLOBAL_SCU_XMC1</b> Enable a specific even
<code>__STATIC_INLINE void</code>	<b>GLOBAL_SCU_XMC1</b> <b>GLOBAL_SCU_XMC1</b> Disable a specific ever
	<b>GLOBAL_SCU_XMC1</b> <b>GLOBAL_SCU_XMC1</b>

```
enum = 0U,  
      GLOBAL_SCU_XMC1  
      1U }
```

```
typedef enum GLOBAL_SCU_XMC1_STATUS GLOBAL_SCU_XMC1
```

[Go to the source code of this file.](#)

---



# GLOBAL\_SCU\_XMC1

Home						
File List	Globals					
All	Functions	Typedefs	Enumerations	Enumerator	Defines	
g						

Here is a list of all documented functions, variables, defines, enums, and typedefs with links to the documentation:

- g -

- GLOBAL\_SCU\_XMC1\_CONFIG\_t : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_DisableEvent() : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_DisableIRQ() : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EnableEvent() : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EnableIRQ() : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_ACMP3 : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_DCO1\_OUT\_SYNC : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_COMPLETED : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_ERROR : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_HANDLER\_t : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_LOCI : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_LOSS\_EXT\_CLOCK : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_PESRAM : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_PEUSIC0 : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_ALARM :

- GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_PERIODIC :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCATIM0\_UPDATED :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCATIM1\_UPDATED :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCCTR\_UPDATED :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCTIM0\_UPDATED :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCTIM1\_UPDATED :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_STDBYCLKFAIL :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_t : **GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_DONE :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_HIGH :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_LOW :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_VCLIP : **GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_VDDPI : **GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_VDROP :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_EVENT\_WDT\_WARNING :  
**GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_GetAppVersion() :  
**GLOBAL\_SCU\_XMC1.h , GLOBAL\_SCU\_XMC1.c**
- GLOBAL\_SCU\_XMC1\_Init() : **GLOBAL\_SCU\_XMC1.h ,**  
**GLOBAL\_SCU\_XMC1.c**
- GLOBAL\_SCU\_XMC1\_RegisterCallback() :  
**GLOBAL\_SCU\_XMC1.h , GLOBAL\_SCU\_XMC1.c**
- GLOBAL\_SCU\_XMC1\_STATUS : **GLOBAL\_SCU\_XMC1.h**
- GLOBAL\_SCU\_XMC1\_STATUS\_FAILURE :  
**GLOBAL\_SCU\_XMC1.h**

- GLOBAL\_SCU\_XMC1\_STATUS\_SUCCESS :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_STATUS\_t : **GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_t : **GLOBAL\_SCU\_XMC1.h**
- 
-

# GLOBAL\_SCU\_XMC1

<a href="#">Home</a>						
<a href="#">File List</a>	<a href="#">Globals</a>					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Typedefs</a>	<a href="#">Enumerations</a>	<a href="#">Enumerator</a>	<a href="#">Defines</a>	

- [GLOBAL\\_SCU\\_XMC1\\_DisableEvent\(\)](#) : [GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_DisableIRQ\(\)](#) : [GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_EnableEvent\(\)](#) : [GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_EnableIRQ\(\)](#) : [GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_GetAppVersion\(\)](#) :  
[GLOBAL\\_SCU\\_XMC1.c](#) , [GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_Init\(\)](#) : [GLOBAL\\_SCU\\_XMC1.c](#) ,  
[GLOBAL\\_SCU\\_XMC1.h](#)
- [GLOBAL\\_SCU\\_XMC1\\_RegisterCallback\(\)](#) :  
[GLOBAL\\_SCU\\_XMC1.c](#) , [GLOBAL\\_SCU\\_XMC1.h](#)



# GLOBAL\_SCU\_XMC1

Home						
File List	Globals					
All	Functions	Typedefs	Enumerations	Enumerator	Defines	

- GLOBAL\_SCU\_XMC1\_CONFIG\_t: [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_HANDLER\_t:  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_t: [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_STATUS\_t: [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_t: [GLOBAL\\_SCU\\_XMC1.h](#)



# GLOBAL\_SCU\_XMC1

Home						
File List	Globals					
All	Functions	Typedefs	Enumerations	Enumerator	Defines	

- GLOBAL\_SCU\_XMC1\_STATUS : [GLOBAL\\_SCU\\_XMC1.h](#)



# GLOBAL\_SCU\_XMC1

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	Defines

- GLOBAL\_SCU\_XMC1\_STATUS\_FAILURE :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_STATUS\_SUCCESS :  
[GLOBAL\\_SCU\\_XMC1.h](#)



# GLOBAL\_SCU\_XMC1

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	Defines

- GLOBAL\_SCU\_XMC1\_EVENT\_ACMP3 :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_DCO1\_OUT\_SYNC :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_COMPLETED :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_FLASH\_ERROR :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_LOCI : [GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_LOSS\_EXT\_CLOCK :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_PESRAM :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_PEUSIC0 :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_ALARM :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTC\_PERIODIC :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCATIM0\_UPDATED :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCATIM1\_UPDATED :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCCTR\_UPDATED :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCTIM0\_UPDATED :  
[GLOBAL\\_SCU\\_XMC1.h](#)
- GLOBAL\_SCU\_XMC1\_EVENT\_RTCTIM1\_UPDATED :

- **GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_STDBYCLKFAIL :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_DONE :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_HIGH :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_TSE\_LOW :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_VCLIP : **GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_VDDPI : **GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_VDROP :  
**GLOBAL\_SCU\_XMC1.h**
  - GLOBAL\_SCU\_XMC1\_EVENT\_WDT\_WARNING :  
**GLOBAL\_SCU\_XMC1.h**
- 
-

# GLOBAL\_SCU\_XMC1

Home		
File List	Globals	

## GLOBAL\_SCU\_XMC1.h

[Go to the documentation of this file.](#)

```
00001
00060 #ifndef GLOBAL_SCU_XMC1_H
00061 #define GLOBAL_SCU_XMC1_H
00062
00063 /*****
*****
*****
00064  * HEADER FILES
00065  *****/
00066
00067 #include <xmc_scu.h>
00068 #include <DAVE_Common.h>
00069 #include "CPU_CTRL_XMC1\cpu_ctrl_xmc1.h"
00070
00071 #include "global_scu_xmc1_conf.h"
00072
00077 /*****
*****
*****
00078  * MACROS
00079  *****/
00080 #define GLOBAL_SCU_XMC1_IRQNUM (3U)
00081
```

```
00082 #define GLOBAL_SCU_XMC1_EVENT_WDT_WARNING
      XMC_SCU_INTERRUPT_EVENT_WDT_WARN
00083 #define GLOBAL_SCU_XMC1_EVENT_RTC_PERIODIC
      XMC_SCU_INTERRUPT_EVENT_RTC_PERIODIC
00084 #define GLOBAL_SCU_XMC1_EVENT_RTC_ALARM
      XMC_SCU_INTERRUPT_EVENT_RTC_ALARM
00085 #define GLOBAL_SCU_XMC1_EVENT_VDDPI
      XMC_SCU_INTERRUPT_EVENT_VDDPI
00086 #if defined(COMPARATOR)
00087 #define GLOBAL_SCU_XMC1_EVENT_ACOMP0
      XMC_SCU_INTERRUPT_EVENT_ACOMP0
00088 #define GLOBAL_SCU_XMC1_EVENT_ACOMP1
      XMC_SCU_INTERRUPT_EVENT_ACOMP1
00089 #define GLOBAL_SCU_XMC1_EVENT_ACOMP2
      XMC_SCU_INTERRUPT_EVENT_ACOMP2
00091 #endif
00092 #define GLOBAL_SCU_XMC1_EVENT_VDROP
      XMC_SCU_INTERRUPT_EVENT_VDROP
00093 #if defined(COMPARATOR)
00094 #define GLOBAL_SCU_XMC1_EVENT_ORC0
      XMC_SCU_INTERRUPT_EVENT_ORC0
00095 #define GLOBAL_SCU_XMC1_EVENT_ORC1
      XMC_SCU_INTERRUPT_EVENT_ORC1
00096 #define GLOBAL_SCU_XMC1_EVENT_ORC2
      XMC_SCU_INTERRUPT_EVENT_ORC2
00097 #define GLOBAL_SCU_XMC1_EVENT_ORC3
      XMC_SCU_INTERRUPT_EVENT_ORC3
00098 #define GLOBAL_SCU_XMC1_EVENT_ORC4
      XMC_SCU_INTERRUPT_EVENT_ORC4
00099 #define GLOBAL_SCU_XMC1_EVENT_ORC5
      XMC_SCU_INTERRUPT_EVENT_ORC5
00100 #define GLOBAL_SCU_XMC1_EVENT_ORC6
      XMC_SCU_INTERRUPT_EVENT_ORC6
00101 #define GLOBAL_SCU_XMC1_EVENT_ORC7
      XMC_SCU_INTERRUPT_EVENT_ORC7
00102 #endif
00103 #define GLOBAL_SCU_XMC1_EVENT_LOCI
```

```
XMC_SCU_INTERRUPT_EVENT_LOCI
00104 #define GLOBAL_SCU_XMC1_EVENT_PESRAM
      XMC_SCU_INTERRUPT_EVENT_PESRAM
00105 #define GLOBAL_SCU_XMC1_EVENT_PEUSIC0
      XMC_SCU_INTERRUPT_EVENT_PEUSIC0
00106 #if defined(USIC1)
00107 #define GLOBAL_SCU_XMC1_EVENT_PEUSIC1
      XMC_SCU_INTERRUPT_EVENT_PEUSIC1
00108 #endif
00109 #define GLOBAL_SCU_XMC1_EVENT_FLASH_ERROR
      XMC_SCU_INTERRUPT_EVENT_FLASH_ERROR
00110 #define GLOBAL_SCU_XMC1_EVENT_FLASH_COMPLETE
D   XMC_SCU_INTERRUPT_EVENT_FLASH_COMPLETED
00111 #define GLOBAL_SCU_XMC1_EVENT_VCLIP
      XMC_SCU_INTERRUPT_EVENT_VCLIP
00112 #define GLOBAL_SCU_XMC1_EVENT_STDBYCLKFAIL
      XMC_SCU_INTERRUPT_EVENT_STDBYCLKFAIL
00113 #define GLOBAL_SCU_XMC1_EVENT_RTCCTR_UPDATED
      XMC_SCU_INTERRUPT_EVENT_RTCCTR_UPDATED
00114 #define GLOBAL_SCU_XMC1_EVENT_RTCATIM0_UPDAT
ED   XMC_SCU_INTERRUPT_EVENT_RTCATIM0_UPDATED
00115 #define GLOBAL_SCU_XMC1_EVENT_RTCATIM1_UPDAT
ED   XMC_SCU_INTERRUPT_EVENT_RTCATIM1_UPDATED
00116 #define GLOBAL_SCU_XMC1_EVENT_RTCTIM0_UPDATE
D   XMC_SCU_INTERRUPT_EVENT_RTCTIM0_UPDATED
00117 #define GLOBAL_SCU_XMC1_EVENT_RTCTIM1_UPDATE
D   XMC_SCU_INTERRUPT_EVENT_RTCTIM1_UPDATED
00118 #define GLOBAL_SCU_XMC1_EVENT_TSE_DONE
      XMC_SCU_INTERRUPT_EVENT_TSE_DONE
00119 #define GLOBAL_SCU_XMC1_EVENT_TSE_HIGH
      XMC_SCU_INTERRUPT_EVENT_TSE_HIGH
00120 #define GLOBAL_SCU_XMC1_EVENT_TSE_LOW
      XMC_SCU_INTERRUPT_EVENT_TSE_LOW
00121 #if defined(CAN)
00122 #define GLOBAL_SCU_INTERRUPT_EVENT_PEMCAN
      XMC_SCU_INTERRUPT_EVENT_PEMCAN
00123 #endif
```

```

00124 #if (UC_SERIES == XMC14)
00125 #define GLOBAL_SCU_XMC1_EVENT_LOSS_EXT_CLOCK
        XMC_SCU_INTERRUPT_EVENT_LOSS_EXT_CLOCK
00126 #define GLOBAL_SCU_XMC1_EVENT_DC01_OUT_SYNC
        XMC_SCU_INTERRUPT_EVENT_DC01_OUT_SYNC
00127 #define GLOBAL_SCU_XMC1_EVENT_ACOMP3
        XMC_SCU_INTERRUPT_EVENT_ACOMP3
00128 #endif
00129 /*****
*****
*****
00130  * ENUMS
00131  *****/
00132
00140 typedef enum GLOBAL_SCU_XMC1_STATUS
00141 {
00142     GLOBAL_SCU_XMC1_STATUS_SUCCESS = 0U,
00143     GLOBAL_SCU_XMC1_STATUS_FAILURE = 1U
00144 } GLOBAL_SCU_XMC1_STATUS_t;
00145
00153 /*****
*****
*****
00154  * DATA STRUCTURES
00155  *****/
00156
00160 typedef XMC_SCU_INTERRUPT_EVENT_HANDLER_t GL
OBAL_SCU_XMC1_EVENT_HANDLER_t;
00161
00165 typedef XMC_SCU_INTERRUPT_EVENT_t GLOBAL_SCU
_XMC1_EVENT_t;
00166
00171 typedef struct GLOBAL_SCU_XMC1_CONFIG

```

```

00172 {
00173     uint8_t priority[GLOBAL_SCU_XMC1_IRQNUM];

00174     bool enable_at_init[GLOBAL_SCU_XMC1_IRQNUM
];
00175 } GLOBAL_SCU_XMC1_CONFIG_t;
00176
00177
00181 typedef struct GLOBAL_SCU_XMC1
00182 {
00183     const GLOBAL_SCU_XMC1_CONFIG_t * config;
00184     bool initialized;
00185 } GLOBAL_SCU_XMC1_t;
00193 /******
*****
*****
00194  * API PROTOTYPES
00195  *****
*****
*****
00196
00231 DAVE_APP_VERSION_t GLOBAL_SCU_XMC1_GetAppVer
sion(void);
00232
00247 GLOBAL_SCU_XMC1_STATUS_t GLOBAL_SCU_XMC1_Init
(GLOBAL_SCU_XMC1_t *const handle);
00248
00279 GLOBAL_SCU_XMC1_STATUS_t GLOBAL_SCU_XMC1_Reg
isterCallback(const GLOBAL_SCU_XMC1_EVENT_t event,
00280
                const GLOBAL_SCU_XMC1_EVENT_HANDLER_t
callback);
00281
00287 __STATIC_INLINE void GLOBAL_SCU_XMC1_EnableI
RQ(uint8_t irq_num)
00288 {
00289     NVIC_EnableIRQ((IRQn_Type)irq_num);

```

```
00290 }
00291
00297 __STATIC_INLINE void GLOBAL_SCU_XMC1_Disable
IRQ(uint8_t irq_num)
00298 {
00299     NVIC_DisableIRQ((IRQn_Type)irq_num);
00300 }
00301
00307 __STATIC_INLINE void GLOBAL_SCU_XMC1_EnableE
vent(const GLOBAL_SCU_XMC1_EVENT_t event)
00308 {
00309     XMC_SCU_INTERRUPT_EnableEvent(event);
00310 }
00311
00317 __STATIC_INLINE void GLOBAL_SCU_XMC1_Disable
Event(const GLOBAL_SCU_XMC1_EVENT_t event)
00318 {
00319     XMC_SCU_INTERRUPT_DisableEvent(event);
00320 }
00321
00328 #ifdef __cplusplus
00329 }
00330 #endif
00331
00332 #include "global_scu_xmc1_extern.h"
00333
00334 #endif
```



# GLOBAL\_SCU\_XMC1

Home	
File List	Globals

## GLOBAL\_SCU\_XMC1.c

[Go to the documentation of this file.](#)

```
00001
00055 /*****
*****
*****
00056  * HEADER FILES
00057  *****/
00058 #include "global_scu_xmc1.h"
00059 /*****
*****
00060  * MACROS
00061  *****/
00062 #define GLOBAL_SCU_XMC1_CHECK_EVENT(event) (
(event == GLOBAL_SCU_XMC1_EVENT_WDT_WARNING) || \
00063
        (event == GLOBAL_SCU_XMC1_EVENT_RTC_PERIOD
IC) || \
00064
        (event == GLOBAL_SCU_XMC1_EVENT_RTC_ALARM)
|| \
00065
                                (event
== GLOBAL_SCU_XMC1_EVENT_LOSS_EXT_CLOCK) || \
```

```

00066
                                                    (event
== GLOBAL_SCU_XMC1_EVENT_DC01_OUT_SYNC))
00067
00068 /******
*****
*****
00069 * LOCAL DATA
00070 *****
*****
***** /
00071
00072 /******
*****
*****
00073 * LOCAL ROUTINES
00074 *****
*****
***** /
00075 #if (UC_SERIES != XMC14)
00076 void SCU_0_IRQHandler(void);
00077 void SCU_1_IRQHandler(void);
00078 void SCU_2_IRQHandler(void);
00079 #else
00080 void IRQ0_Handler(void);
00081 void IRQ1_Handler(void);
00082 void IRQ2_Handler(void);
00083 #endif
00084 /******
*****
*****
00085 * API IMPLEMENTATION
00086 *****
*****
***** /
00087
00088 /*

```

```

00089  * API to retrieve the version of the GLOBAL
_SCU_XMC1 APP
00090  */
00091  DAVE_APP_VERSION_t GLOBAL_SCU_XMC1_GetAppVer
sion(void)
00092  {
00093      DAVE_APP_VERSION_t version;
00094
00095      version.major = GLOBAL_SCU_XMC1_MAJOR_VERS
ION;
00096      version.minor = GLOBAL_SCU_XMC1_MINOR_VERS
ION;
00097      version.patch = GLOBAL_SCU_XMC1_PATCH_VERS
ION;
00098
00099      return (version);
00100  }
00101
00102  /* Function to configure SCU Interrupts bas
ed on user configuration.
00103  *
00104  */
00105  GLOBAL_SCU_XMC1_STATUS_t GLOBAL_SCU_XMC1_Init
(GLOBAL_SCU_XMC1_t*const handle)
00106  {
00107      GLOBAL_SCU_XMC1_STATUS_t initstatus;
00108
00109      XMC_ASSERT("GLOBAL_SCU_XMC1_Init:HandlePtr
NULL", (handle != NULL));
00110
00111      initstatus = (GLOBAL_SCU_XMC1_STATUS_t)CPU
_CTRL_XMC1_Init(CPU_CTRL_HANDLE);
00112
00113      if (initstatus == GLOBAL_SCU_XMC1_STATUS_S
UCCESS)
00114      {
00115          if (handle->initialized == false)

```

```

00116     {
00117         NVIC_SetPriority((IRQn_Type)0U, han
dle->config->priority[0]);
00118         NVIC_SetPriority((IRQn_Type)1U, han
dle->config->priority[1]);
00119         NVIC_SetPriority((IRQn_Type)2U, han
dle->config->priority[2]);
00120
00121         /* enable the IRQ0 */
00122         if (handle->config->enable_at_init
[0] == true)
00123             {
00124 #if (UC_SERIES == XMC14)
00125                 XMC_SCU_SetInterruptControl(
0, XMC_SCU_IRQCTRL_SCU_SR0_IRQ0);
00126 #endif
00127                 NVIC_EnableIRQ((IRQn_Type)0U
);
00128             }
00129         /* enable the IRQ1 */
00130         if (handle->config->enable_at_init
[1] == true)
00131             {
00132 #if (UC_SERIES == XMC14)
00133                 XMC_SCU_SetInterruptControl(
1, XMC_SCU_IRQCTRL_SCU_SR1_IRQ1);
00134 #endif
00135                 NVIC_EnableIRQ((IRQn_Type)1U
);
00136             }
00137         /* enable the IRQ2 */
00138         if (handle->config->enable_at_init
[2] == true)
00139             {
00140 #if (UC_SERIES == XMC14)
00141                 XMC_SCU_SetInterruptControl(
2, XMC_SCU_IRQCTRL_SCU_SR2_IRQ2);

```

```

00142 #endif
00143         NVIC_EnableIRQ((IRQn_Type)2U
);
00144     }
00145     handle->initialized = true;
00146 }
00147 else
00148 {
00149     initstatus = GLOBAL_SCU_XMC1_STATUS_SU
CESS;
00150 }
00151 }
00152 else
00153 {
00154     initstatus = GLOBAL_SCU_XMC1_STATUS_FAIL
URE;
00155 }
00156
00157 return (initstatus);
00158 }
00159
00160 #if (UC_SERIES != XMC14)
00161 /*
00162  * @brief SCU_0 Interrupt Handler
00163  */
00164 void SCU_0_IRQHandler(void)
00165 {
00166     XMC_SCU_IRQHandler(0);
00167 }
00168
00169 /* SCU_1 Interrupt Handler.
00170  *
00171  */
00172 void SCU_1_IRQHandler(void)
00173 {
00174     XMC_SCU_IRQHandler(1);
00175 }

```

```
00176
00177 /*  SCU2 Interrupt Handler.
00178  *
00179  */
00180 void SCU_2_IRQHandler(void)
00181 {
00182     XMC_SCU_IRQHandler(2);
00183 }
00184 #else
00185 /*
00186  * @brief  IRQ0 Interrupt Handler
00187  */
00188 void IRQ0_Handler(void)
00189 {
00190     XMC_SCU_IRQHandler(0);
00191 }
00192
00193 /*  IRQ1 Interrupt Handler.
00194  *
00195  */
00196 void IRQ1_Handler(void)
00197 {
00198     XMC_SCU_IRQHandler(1);
00199 }
00200
00201 /*  IRQ2 Interrupt Handler.
00202  *
00203  */
00204 void IRQ2_Handler(void)
00205 {
00206     XMC_SCU_IRQHandler(2);
00207 }
00208
00209 #endif
00210
00211 /*
00212  * @brief  Function to register callback eve
```

```
nt
00213  */
00214 GLOBAL_SCU_XMC1_STATUS_t GLOBAL_SCU_XMC1_Reg
isterCallback(const GLOBAL_SCU_XMC1_EVENT_t event,
00215
                const GLOBAL_SCU_XMC1_EVENT_HANDLER_t
handler)
00216 {
00217     GLOBAL_SCU_XMC1_STATUS_t status;
00218
00219     XMC_ASSERT("GLOBAL_SCU_XMC1_RegisterCallba
ck: Invalid event", (GLOBAL_SCU_XMC1_CHECK_EVENT(e
vent)));
00220     XMC_ASSERT("GLOBAL_SCU_XMC1_RegisterCallba
ck: NULL Handle", (handler != NULL));
00221
00222     status = (GLOBAL_SCU_XMC1_STATUS_t)XMC_SCU
_INTERRUPT_SetEventHandler(event, handler);
00223
00224     return (status);
00225 }
```

