

GLOBAL_CCU4

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_CCU4

Home

License Terms and Copyright Information

License Terms and Copyright Information

Copyright (c) 2016, 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).

GLOBAL_CCU4

Home

Abbreviations and Definitions

Abbreviations and Definitions

Abbreviations:	
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
I/O	Input/Output
CCU	Capture Compare Unit
CC	Capture Compare Slice
SCU	System Control Unit

Definitions:	
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)

DAVE™ APPs



GLOBAL_CCU4

Home

Overview

Overview

The primary purpose of the **GLOBAL_CCU4** APP is to share the common resources in the CCU4 module among top level APPs. It provides functions to accomplish the following:

1. Enable CCU4 module
2. Start CC4y (y = [0..3]) timers synchronously
3. Shadow transfer of selected values (in the GUI) in multi-channel mode

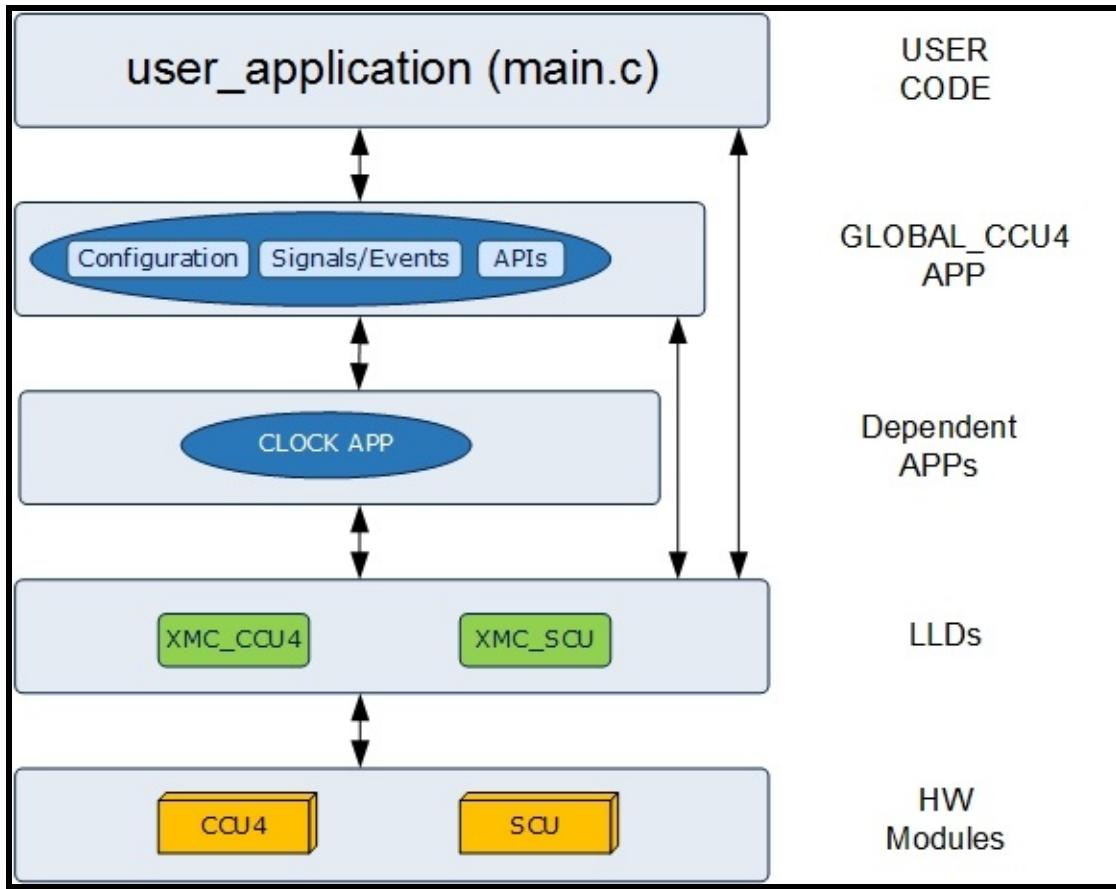


Figure 1 : Hardware and Software connectivity of **GLOBAL_CCU4 APP**

The figure above shows the layered architecture of the **GLOBAL_CCU4 APP** for DAVE™. The LLD layer provides an abstraction for the CCU4 hardware module. The **GLOBAL_CCU4 APP** uses CCU4 and SCU LLDs for its functionality.

Supported Devices

The APP supports the following devices:

1. XMC4800/XMC4700 Series
2. XMC4500 Series
3. XMC4400 Series
4. XMC4300 Series
5. XMC4200 / XMC4100 Series
6. XMC1400 Series
7. XMC1300 Series

8. XMC1200 Series
9. XMC1100 Series

References

1. XMC4800 / XMC4700 Reference Manual
 2. XMC4500 Reference Manual
 3. XMC4400 Reference Manual
 4. XMC4300 Reference Manual
 5. XMC4200 / XMC4100 Reference Manual
 6. XMC1400 Reference Manual
 7. XMC1300 Reference Manual
 8. XMC1200 Reference Manual
 9. XMC1100 Reference Manual
-



GLOBAL_CCU4

Home

Architecture Description

Architecture Description

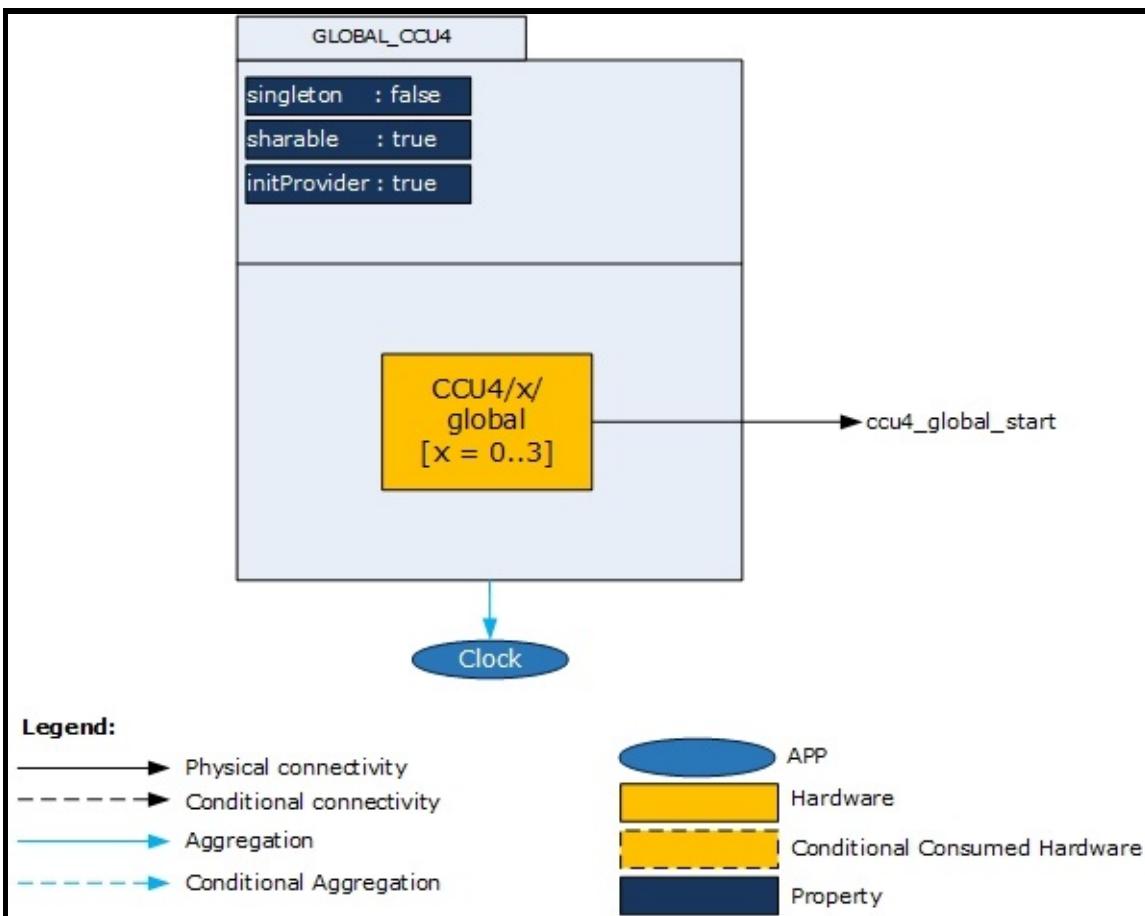


Figure 1 : Architecture of **GLOBAL_CCU4** APP

The diagram above represents the internal software architecture of the **GLOBAL_CCU4** APP. A **GLOBAL_CCU4** APP instance exists in a DAVE™ project with fixed attributes as shown. Each instance of this APP configures one CCU4 module in the MCU. In addition, the APP requires the consumption of the CLOCK APP for its configuration and functioning. The **GLOBAL_CCU4** APP also provides an output signal

for inter-peripheral connections.

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_CCU4_0 to GLOBAL_CCU4_SVM).

Signals:

The following table presents the signals provided by the APP for inter-peripheral connections:

Table 1: APP I/O signals

Signal Name	Input/Output	Availability	Description
ccu4_global_start	Output	Always	Global Synchronous Start: This signal can be used to connect with top level APPs like PWM to start the timers synchronously.

APPs Consumed:

The following table presents the APPs consumed to support the functionality:

Table 2: APPs Consumed

APP Name	Consumption	Description
CLOCK_XMC4	Conditionally consumed. Initialise the clock For XMC4000 devices.	settings.

CLOCK_XMC1

Conditionally consumed. Initialise the clock
For XMC1000 devices. settings.

GLOBAL_CCU4

Home

APP Configuration Parameters

App Configuration Parameters

General Settings	
Clock frequency [MHz]:	120
Multi channel mode shadow transfer:	Period and Compare

Figure 1: General Settings

GLOBAL_CCU4

Home

Enumerations

enum **GLOBAL_CCU4_STATUS** {
 GLOBAL_CCU4_STATUS_S = 0U,
 GLOBAL_CCU4_STATUS_F
}

Return status of the **GLOBAL_CCU4** APP. More...

typedef enum **GLOBAL_CCU4_STATUS** **GLOBAL_CCU4_STATUS_t**
Return status of the **GLOBAL_CCU4** APP.

Enumeration Type Documentation

enum GLOBAL_CCU4_STATUS

Return status of the **GLOBAL_CCU4** APP.

Enumerator:

GLOBAL_CCU4_STATUS_SUCCESS Status success

GLOBAL_CCU4_STATUS_FAILURE Status failure

Definition at line **104** of file **GLOBAL_CCU4.h**.

GLOBAL_CCU4

Home

Data Structures

Data structures

Data Structures

```
struct GLOBAL_CCU4  
typedef struct GLOBAL_CCU4 GLOBAL_CCU4_t
```

Typedef Documentation

typedef struct GLOBAL_CCU4 GLOBAL_CCU4_t

This saves the context of the **GLOBAL_CCU4 APP**.

GLOBAL_CCU4

Home

Methods

DAVE_APP_VERSION_t **GLOBAL_CCU4_GetAppVersion** (void)
Get GLOBAL_CCU4 APP version.

GLOBAL_CCU4_STATUS_t **GLOBAL_CCU4_Init**
(**GLOBAL_CCU4_t** *handle)
Initializes a **GLOBAL_CCU4** with
generated configuration.

_STATIC_INLINE void **GLOBAL_CCU4_SyncStartTriggerHigh**
(**uint32_t** ccucon_msk)
Start all the timers which are configured
to start externally on positive edge.

_STATIC_INLINE void **GLOBAL_CCU4_SyncStartTriggerLow**
(**uint32_t** ccucon_msk)
Start all the timers which are configured
to start externally on negative edge.

Methods

Function Documentation

DAVE_APP_VERSION_t GLOBAL_CCU4_GetAppVersion (void)

Get **GLOBAL_CCU4** APP version.

Returns:

DAVE_APP_VERSION_t APP version information (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_STATUS_t status;
    DAVE_APP_VERSION_t app_version;

    status = DAVE_Init();           // GLOBAL_CCU4_Init()
                                    // is called from DAVE_Init()

    app_version = GLOBAL_CCU4_GetAppVersion();

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

    while(1U)
    {
    }
    return 1;
}
```

```
}
```

Definition at line **70** of file **GLOBAL_CCU4.c**.

GLOBAL_CCU4_STATUS_t GLOBAL_CCU4_Init (GLOBAL_CCU4_1

Initializes a **GLOBAL_CCU4** with generated configuration.

Parameters:

handle pointer to the **GLOBAL_CCU4** APP handle structure.

Returns:

GLOBAL_CCU4_STATUS_t

GLOBAL_CCU4_STATUS_SUCCESS : if initialization is successful

GLOBAL_CCU4_STATUS_FAILURE : if initialization is failed

Description:

- Enable the module.
- Start the prescaler.

Example Usage:

```
#include <DAVE.h>
int main(void)
{
    DAVE_STATUS_t init_status;
    init_status = DAVE_Init();      // GLOBAL_CCU4_Init(&GLOBAL_CCU4_0) will be called from DAVE_Init()

    while(1)
{
```

```
    }
    return 1;
}
```

Definition at line **82** of file [GLOBAL_CCU4.c](#).

References [GLOBAL_CCU4_STATUS_SUCCESS](#),
[GLOBAL_CCU4::is_initialized](#), [GLOBAL_CCU4::mcs_action](#), and
[GLOBAL_CCU4::module_ptr](#).

__STATIC_INLINE void GLOBAL_CCU4_SyncStartTriggerHigh (uint32_t ccucon_msk)

Start all the timers which are configured to start externally on positive edge.

Parameters:

ccucon_msk mask for which kernels sync start has to be applied.

Note:

This mask has been generated in the APP handle and as a macro in global_ccu4_conf.h file. 1. The variable from the APP handle is useful while starting the specific kernel/s 2.

GLOBAL_CCU4_CCUCON_Msk Macro from global_ccu4_conf.h file can be used to start all the selected kernels at a time.

Return values:

none

Description:

The top level APPs have to be enabled, to start the timer externally with positive trigger edge.

Example Usage:

```
#include <DAVE.h>
int main(void)
{
    DAVE_STATUS_t status;

    status = DAVE_Init();           // GLOBAL_CCU4_Init()
                                    // is called from DAVE_Init()

    // Below can be used to start the specific kernels, by generating two instance of APP
    // GLOBAL_CCU4_SyncStartTriggerHigh((uint32_t)(GLOBAL_CCU4_0.syncstart_trigger_msk | GLOBAL_CCU4_1.syncstart_trigger_msk));
    // Below can be used to start all the kernels simultaneously
    GLOBAL_CCU4_SyncStartTriggerHigh(GLOBAL_CCU4_CUCON_Msk);

    while(1)
    {

    }

    return 1;
}
```

Definition at line [245](#) of file **GLOBAL_CCU4.h**.

__STATIC_INLINE void GLOBAL_CCU4_SyncStartTriggerLow (uint

Start all the timers which are configured to start externally on negative edge.

Parameters:

ccucon_msk mask for which kernels sync start has to be applied.

Note:

This mask has been generated in the APP handle and a macro in global_ccu4_conf.h file. 1. The variable from the APP handle is useful while starting the specific kernel/s 2.

GLOBAL_CCU4_CCUCON_Msk Macro from global_ccu4_conf.h file can be used to start all the selected kernels at a time.

Return values:

none

Description:

The top level APPs have to be enabled, to start the timer externally with negative trigger edge.

Example Usage:

```
#include <DAVE.h>
int main(void)
{
    DAVE_STATUS_t status;

    status = DAVE_Init();           // GLOBAL_CCU4_Init() is called from DAVE_Init()

    // Below can be used to start the specific kernels, by generating two instance of APP
    // GLOBAL_CCU4_SyncStartTriggerLow((uint32_t)(GLOBAL_CCU4_0.syncstart_trigger_msk | GLOBAL_CCU4_1.syncstart_trigger_msk));
    // Below can be used to start all the kernels simultaneously
    GLOBAL_CCU4_SyncStartTriggerLow(GLOBAL_CCU4_CC
```

```
    UCON_Msk);  
  
    while(1)  
    {  
    }  
  
    return 1;  
}
```

Definition at line **284** of file **GLOBAL_CCU4.h**.

GLOBAL_CCU4

Home

Usage

Usage

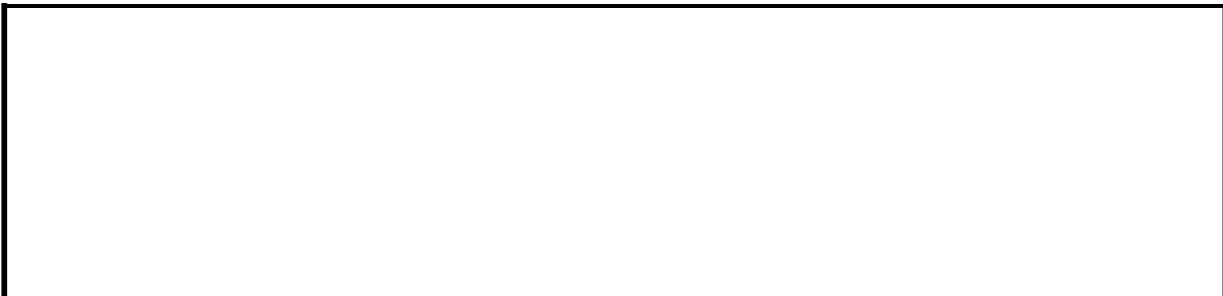
GLOBAL_CCU4 is a global DAVE™ APP. It is consumed by the PWM and other (top level) APPs. For information on how **GLOBAL_CCU4** is being used, refer to the PWM related APPs help documentation. e.g.: PWM, PWM_CCU4

GLOBAL_CCU4

Home

Release History

Release History



GLOBAL_CCU4

[Home](#)

[Data Structures](#)

[Data Structure Index](#)

[Data Fields](#)

Data Structures

Here are the data structures with brief descriptions:

[GLOBAL_CCU4](#)

GLOBAL_CCU4

[Home](#)

[Data Structures](#)

[Data Structure Index](#)

[Data Fields](#)

[Data Fields](#)

GLOBAL_CCU4 Struct Reference

[Data structures](#)

Detailed Description

This saves the context of the **GLOBAL_CCU4** APP.

Definition at line **124** of file **GLOBAL_CCU4.h**.

```
#include <GLOBAL_CCU4.h>
```

Data Fields

const uint32_t	module_frequency
const XMC_SCU_CCU_TRIGGER_t	syncstart_trigger_msk
XMC_CCU4_MODULE_t *const	module_ptr
XMC_CCU4_SLICE_MCMS_ACTION_t const	mcs_action
bool	is_initialized

Field Documentation

bool GLOBAL_CCU4::is_initialized

Indicates initialized state of particular instance of the APP

Definition at line [130](#) of file **GLOBAL_CCU4.h**.

Referenced by **GLOBAL_CCU4_Init()**.

XMC_CCU4_SLICE_MCMS_ACTION_t const GLOBAL_CCU4::mcs_

Shadow transfer of selected values in multi-channel mode

Definition at line [129](#) of file **GLOBAL_CCU4.h**.

Referenced by **GLOBAL_CCU4_Init()**.

const uint32_t GLOBAL_CCU4::module_frequency

fccu frequency

Definition at line [126](#) of file **GLOBAL_CCU4.h**.

XMC_CCU4_MODULE_t* const GLOBAL_CCU4::module_ptr

reference to module handle

Definition at line [128](#) of file **GLOBAL_CCU4.h**.

Referenced by **GLOBAL_CCU4_Init()**.

const XMC_SCU_CCU_TRIGGER_t GLOBAL_CCU4::syncstart_trig

Mask to start the timers synchronously

Definition at line [127](#) of file **GLOBAL_CCU4.h**.

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

- [GLOBAL_CCU4.h](#)
-



GLOBAL_CCU4

Home

Data Structures

Data Structure Index

Data Fields

Data Structure Index

G

G

GLOBAL_CCU4

G



GLOBAL_CCU4

Home		
Data Structures	Data Structure Index	Data Fields
All	Variables	

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

- is_initialized : [GLOBAL_CCU4](#)
 - mcs_action : [GLOBAL_CCU4](#)
 - module_frequency : [GLOBAL_CCU4](#)
 - module_ptr : [GLOBAL_CCU4](#)
 - syncstart_trigger_msk : [GLOBAL_CCU4](#)
-
-

GLOBAL_CCU4

Home		
Data Structures	Data Structure Index	Data Fields
All	Variables	

- is_initialized : GLOBAL_CCU4
 - mcs_action : GLOBAL_CCU4
 - module_frequency : GLOBAL_CCU4
 - module_ptr : GLOBAL_CCU4
 - syncstart_trigger_msk : GLOBAL_CCU4
-

GLOBAL_CCU4

Home

File List

Globals

File List

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

[GLOBAL_CCU4.c](#) [code]  

[GLOBAL_CCU4.h](#) [code]  

GLOBAL_CCU4

Home	
File List	Globals
	Functions

GLOBAL_CCU4.c File Reference

Detailed Description

Date:

2016-02-10

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_CCU4.c**.

```
#include "global_ccu4.h"
```

Functions

DAVE_APP_VERSION_t	GLOBAL_CCU4_GetAppVersion (void) Get GLOBAL_CCU4 APP version.
GLOBAL_CCU4_STATUS_t	GLOBAL_CCU4_Init (GLOBAL_CCU4_t *handle) Initializes a GLOBAL_CCU4 with generated configuration.

Function Documentation

GLOBAL_CCU4_STATUS_t GLOBAL_CCU4_Init(GLOBAL_CCU4_t handle)

Initializes a **GLOBAL_CCU4** with generated configuration.

Parameters:

handle pointer to the **GLOBAL_CCU4** APP handle structure.

Returns:

GLOBAL_CCU4_STATUS_t

GLOBAL_CCU4_STATUS_SUCCESS : if initialization is successful

GLOBAL_CCU4_STATUS_FAILURE : if initialization is failed

Description:

- Enable the module.
- Start the prescaler.

Example Usage:

```
#include <DAVE.h>
int main(void)
{
    DAVE_STATUS_t init_status;
    init_status = DAVE_Init();      // GLOBAL_CCU4_Init(&GLOBAL_CCU4_0) will be called from DAVE_Init()

    while(1)
    {
    }
    return 1;
}
```

Definition at line **82** of file **GLOBAL_CCU4.c**.

References **GLOBAL_CCU4_STATUS_SUCCESS**,
GLOBAL_CCU4::is_initialized, **GLOBAL_CCU4::mcs_action**, and
GLOBAL_CCU4::module_ptr.

Go to the source code of this file.

GLOBAL_CCU4

[Home](#)

[File List](#)

[Globals](#)

[Data Structures](#)

GLOBAL_CCU4.h File Reference

Detailed Description

Date:

2016-02-10

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_CCU4.h**.

```
#include <xmc_ccu4.h> #include <xmc_scu.h>
#include <DAVE_Common.h>
#include "global_ccu4_conf.h"
#include "../CLOCK_XMC4/clock_xmc4.h"
#include "global_ccu4_extern.h"
```

Data Structures

```
struct GLOBAL_CCU4
```

TypeDefs

```
typedef struct GLOBAL_CCU4 GLOBAL_CCU4_t
```

Functions

DAVE_APP_VERSION_t	GLOBAL_CCU4_GetAppVe Get GLOBAL_CCU4 APP ve
GLOBAL_CCU4_STATUS_t	GLOBAL_CCU4_Init (GLOBAL_CCU4_t *handle) Initializes a GLOBAL_CCU4 generated configuration.
__STATIC_INLINE void	GLOBAL_CCU4_SyncStart (uint32_t ccucon_msk) Start all the timers which are to start externally on positive
__STATIC_INLINE void	GLOBAL_CCU4_SyncStart (uint32_t ccucon_msk) Start all the timers which are to start externally on negative
enum	GLOBAL_CCU4_STATUS { GLOBAL_CCU4_STATUS_S 0U, GLOBAL_CCU4_STATUS_F Return status of the GLOBAL APP. More...
typedef enum GLOBAL_CCU4_STATUS	GLOBAL_CCU4_STATUS_t Return status of the GLOBAL APP.

Go to the source code of this file.

GLOBAL_CCU4

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	

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

- GLOBAL_CCU4_GetAppVersion() : [GLOBAL_CCU4.c](#) , [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_Init() : [GLOBAL_CCU4.h](#) , [GLOBAL_CCU4.c](#)
 - GLOBAL_CCU4_STATUS : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_STATUS_FAILURE : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_STATUS_SUCCESS : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_STATUS_t : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_SyncStartTriggerHigh() : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_SyncStartTriggerLow() : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_t : [GLOBAL_CCU4.h](#)
-
-
-

GLOBAL_CCU4

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	

- GLOBAL_CCU4_GetAppVersion() : **GLOBAL_CCU4.c** , **GLOBAL_CCU4.h**
 - GLOBAL_CCU4_Init() : **GLOBAL_CCU4.h** , **GLOBAL_CCU4.c**
 - GLOBAL_CCU4_SyncStartTriggerHigh() : **GLOBAL_CCU4.h**
 - GLOBAL_CCU4_SyncStartTriggerLow() : **GLOBAL_CCU4.h**
-

GLOBAL_CCU4

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	

- GLOBAL_CCU4_STATUS_t : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_t : [GLOBAL_CCU4.h](#)
-

GLOBAL_CCU4

Home					
File List	Globals				
All	Functions	Typedefs	Enumerations	Enumerator	

- GLOBAL_CCU4_STATUS : [GLOBAL_CCU4.h](#)
-

GLOBAL_CCU4

Home					
File List		Globals			
All	Functions	Typedefs	Enumerations	Enumerator	

- GLOBAL_CCU4_STATUS_FAILURE : [GLOBAL_CCU4.h](#)
 - GLOBAL_CCU4_STATUS_SUCCESS : [GLOBAL_CCU4.h](#)
-

GLOBAL_CCU4

Home

File List

Globals

GLOBAL_CCU4.h

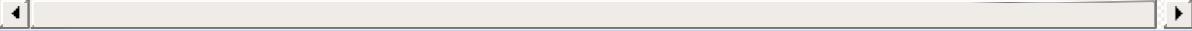
Go to the documentation of this file.

```
00001
00065 #ifndef GLOBAL_CCU4_H
00066 #define GLOBAL_CCU4_H
00067
00068 /*****
* ****
* ****
* ****
00069 * HEADER FILES
00070 ****
* ****
* ****
* ****
* ****
00071 #include <xmc_ccu4.h>
00072 #include <xmc_scu.h>
00073 #include <DAVE_Common.h>
00074 #include "global_ccu4_conf.h"
00075
00076 #if (UC_FAMILY == XMC4)
00077 #include "../CLOCK_XMC4/clock_xmc4.h"
00078
00079 #ifndef CLOCK_XMC4_CCUCLK_ENABLED
00080 #error "Error: GLOBAL_CCU4 APP missing clock
settings. Please review CLOCK_XMC4 settings"
00081 #endif
00082
00083 #endif
00084
00085 /*****
```

```
*****
*****
00086 * MACROS
00087 ****
*****
***** */
00088 #if (!(XMC_LIB_MAJOR_VERSION == 2U) && \
00089         (XMC_LIB_MINOR_VERSION >= 0U) && \
00090         (XMC_LIB_PATCH_VERSION >= 0U)))
00091 #error "GLOBAL_CCU4 requires XMC Peripheral Library v2.0.0 or higher"
00092 #endif
00093
00094 /*****
*****
***** */
00095 * ENUMS
00096 ****
*****
***** */
00104 typedef enum GLOBAL_CCU4_STATUS
00105 {
00106     GLOBAL_CCU4_STATUS_SUCCESS = 0U,
00107     GLOBAL_CCU4_STATUS_FAILURE
00108 } GLOBAL_CCU4_STATUS_t;
00112 /*****
*****
***** */
00113 * DATA STRUCTURES
00114 ****
*****
***** */
00115
00124 typedef struct GLOBAL_CCU4
00125 {
00126     const uint32_t module_frequency;
00127     const XMC_SCU_CCU_TRIGGER_t syncstart_trig
```

```
ger_msk;
00128     XMC_CCU4_MODULE_t* const module_ptr;
00129     XMC_CCU4_SLICE_MCMS_ACTION_t const mcs_action;
00130     bool is_initialized;
00131 } GLOBAL_CCU4_t;
00132
00136 /*****
00136 ****
00136 ****
00137 * API Prototypes
00138 ****
00138 ****
00138 ****
00139 #ifdef __cplusplus
00140 extern "C" {
00141 #endif
00142
00179 DAVE_APP_VERSION_t GLOBAL_CCU4_GetAppVersion(
void);
00180
00209 GLOBAL_CCU4_STATUS_t GLOBAL_CCU4_Init(GLOBAL
_CCU4_t* handle);
00210
00245 __STATIC_INLINE void GLOBAL_CCU4_SyncStartTr
iggerHigh(uint32_t ccucon_msk)
00246 {
00247     XMC_SCU_SetCcuTriggerHigh(ccucon_msk);
00248 }
00249
00284 __STATIC_INLINE void GLOBAL_CCU4_SyncStartTr
iggerLow(uint32_t ccucon_msk)
00285 {
00286     XMC_SCU_SetCcuTriggerLow(ccucon_msk);
00287 }
00288
00294 #include "global_ccu4_extern.h"
```

```
00295
00296 #ifdef __cplusplus
00297 }
00298 #endif
00299
00300 #endif /*CCUGLOBAL_H*/
```



GLOBAL_CCU4

Home	
File List	Globals

GLOBAL_CCU4.c

Go to the documentation of this file.

```
00001
00048 /*****
***** * HEADER FILES *
00049 * HEADER FILES
00050 ****
***** * MACROS *
00055 * MACROS
00056 ****
***** * LOCAL DATA *
00059 * LOCAL DATA
00060 ****
***** /*****
00061
```

```
00062 /*****
* ****
* ****
00063 * LOCAL ROUTINES
00064 ****
* ****
**** */
00065
00066 /*****
* ****
* ****
00067 * API IMPLEMENTATION
00068 ****
* ****
**** */
00069 /* Returns the version of the GLOBAL_CCU4 AP
P. */
00070 DAVE_APP_VERSION_t GLOBAL_CCU4_GetAppVersion(
void)
00071 {
00072     DAVE_APP_VERSION_t version;
00073
00074     version.major = GLOBAL_CCU4_MAJOR_VERSION;
00075     version.minor = GLOBAL_CCU4_MINOR_VERSION;
00076     version.patch = GLOBAL_CCU4_PATCH_VERSION;
00077
00078     return version;
00079 }
00080
00081 /* Initializes the slice with the generated
configuration */
00082 GLOBAL_CCU4_STATUS_t GLOBAL_CCU4_Init(GLOBAL
_CCU4_t* handle)
00083 {
00084     XMC_ASSERT("GLOBAL_CCU4_Init:NULL handler"
, (NULL != handle));
00085
```

```
00086     if (false == handle->is_initialized)
00087     {
00088         /* Enable CCU4 module */
00089         XMC_CCU4_Init(handle->module_ptr,handle->
mcs_action);
00090         /* Start the prescaler */
00091         XMC_CCU4_StartPrescaler(handle->module_p
tr);
00092         /* Restricts multiple initializations */
00093         handle->is_initialized = true;
00094     }
00095
00096     return (GLOBAL_CCU4_STATUS_SUCCESS);
00097 }
```

