Port_AUTOSAR

AUTHOR: Islam Ehab Ezzat
Version 1.1.2
Thu Dec 31 2020

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Data Structure Documentation

Dio_ChannelGroupType Struct Reference

```
#include <Dio.h>
```

Data Fields

- **uint8 mask**
- **uint8 offset**
- **Dio_PortType PortIndex**

Field Documentation

uint8 Dio_ChannelGroupType::mask

uint8 Dio_ChannelGroupType::offset

Dio_PortType Dio_ChannelGroupType::PortIndex

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

- **Dio.h**

Dio_ConfigChannel Struct Reference

```
#include <Dio.h>
```

Data Fields

- **Dio_PortType Port_Num**
- **Dio_ChannelType Ch_Num**

Field Documentation

Dio_ChannelType Dio_ConfigChannel::Ch_Num

Dio_PortType Dio_ConfigChannel::Port_Num

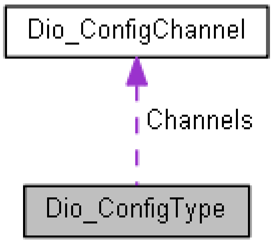
The documentation for this struct was generated from the following file:

- **Dio.h**

Dio_ConfigType Struct Reference

```
#include <Dio.h>
```

Collaboration diagram for Dio_ConfigType:



Data Fields

- **Dio_ConfigChannel Channels** [**DIO_CONFIGURED_CHANNLES**]

Field Documentation

Dio_ConfigChannel Dio_ConfigType::Channels[DIO_CONFIGURED_CHANNLES]

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

- **Dio.h**

Port_ConfigChannel Struct Reference

Used to set the configurations of a Specific pin in Port Module.

```
#include <Port.h>
```

Data Fields

- **Port_PinType pinNum**
- **Port_PortNumEnum portNum**
- **Port_PinDirectionType direction**
- **Port_PinModeType mode**
- **Port_InternalResistor resistor**
- **uint8 pinDirection_Changeable**
- **uint8 pinMode_Changeable**
- **uint8 initialValue**
- **uint8 slewRate**

Detailed Description

Used to set the configurations of a Specific pin in Port Module.

Field Documentation

Port_ConfigChannel::direction

member 'portNum' used to decide which port are used with the required pin You can find its vale **Port_PortNumEnum**

member 'direction' used to decide if the pin is input or output You can find its value **Port_PinDirectionType**

Port_ConfigChannel::initialValue

member 'pinMode_Changable' used to decide if the pin mode can be changeable during run time or not You can find its value PIN_MODE_CHANGEABILITY

member 'initialValue' used to set an initial value for output pins You can find its value INITIAL_VALUE

Port_ConfigChannel::mode

member 'direction' used to decide if the pin is input or output You can find its value **Port_PinDirectionType**

member 'mode' used to decide which mode is used to decide which mode is used (GPIO or any alternative function) You can find its value **Port_PinModeEnum** Note: if you want to use Analog function put **ANALOG_MODE_SELECTED**

uint8 Port_ConfigChannel::pinDirection_Changeable

member 'resistor' used to decide if the internal resistor is enabled or not and its mode if enabled You can find its value **Port_InternalResistor**

uint8 Port_ConfigChannel::pinMode_Changeable

member 'pinDirection_Changable' used to decide if the pin direction can be Changeable in run time or not You can find its value PIN_DIRECTION_CHANGEABILITY

Port_ConfigChannel::pinNum

member 'pinNum' used to hold Pin & port number You can find its value **Port_PinNumEnum**

Port_ConfigChannel::portNum

member 'pinNum' used to hold Pin & port number You can find its value **Port_PinNumEnum**

member 'portNum' used to decide which port are used with the required pin You can find its vale **Port_PortNumEnum**

Port_ConfigChannel::resistor

member 'mode' used to decide which mode is used (GPIO or any alternative function) You can find its value **Port_PinModeEnum** Note: if you want to use Analog function put **ANALOG_MODE_SELECTED**

member 'resistor' used to decide if internal resistor is used or not and its mode if used (Pull-up or Pull-Down) You can find its value **Port_InternalResistor**

Port_ConfigChannel::slewRate

member 'initialValue' used to set an initial value for output pins You can find its value INITIAL_VALUE

member 'slewRate' used to decide if Slew rate is enabled/disabled for the pin You can find its value SLEW_RATE

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

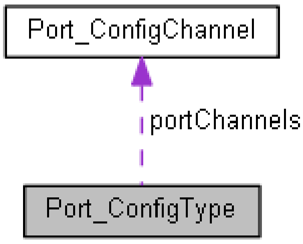
- **Port.h**

Port_ConfigType Struct Reference

This structure holds an array of structures that holds the configurations of the pins.

```
#include <Port.h>
```

Collaboration diagram for Port_ConfigType:



Data Fields

- **Port_ConfigChannel portChannels** [**PORT_CONFIGURED_CHANNLES**]

Detailed Description

This structure holds an array of structures that holds the configurations of the pins.

Note

Unused pins shall configured with the initial value which got from TM4C123GH6PM data sheet, GPIO chapter so that, PORT_CONFIGURED_CHANNELS shall equal 39 in TM4C123GH6PM

Field Documentation

Port_ConfigType::portChannels

member 'portChannels' used as an array to hold the configurations of the pins

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

- **Port.h**

Std_VersionInfoType Struct Reference

```
#include <Std_Types.h>
```

Data Fields

- **uint16 vendorID**
- **uint16 moduleID**
- **uint8 sw_major_version**
- **uint8 sw_minor_version**
- **uint8 sw_patch_version**

Field Documentation

uint16 Std_VersionInfoType::moduleID

uint8 Std_VersionInfoType::sw_major_version

uint8 Std_VersionInfoType::sw_minor_version

uint8 Std_VersionInfoType::sw_patch_version

uint16 Std_VersionInfoType::vendorID

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

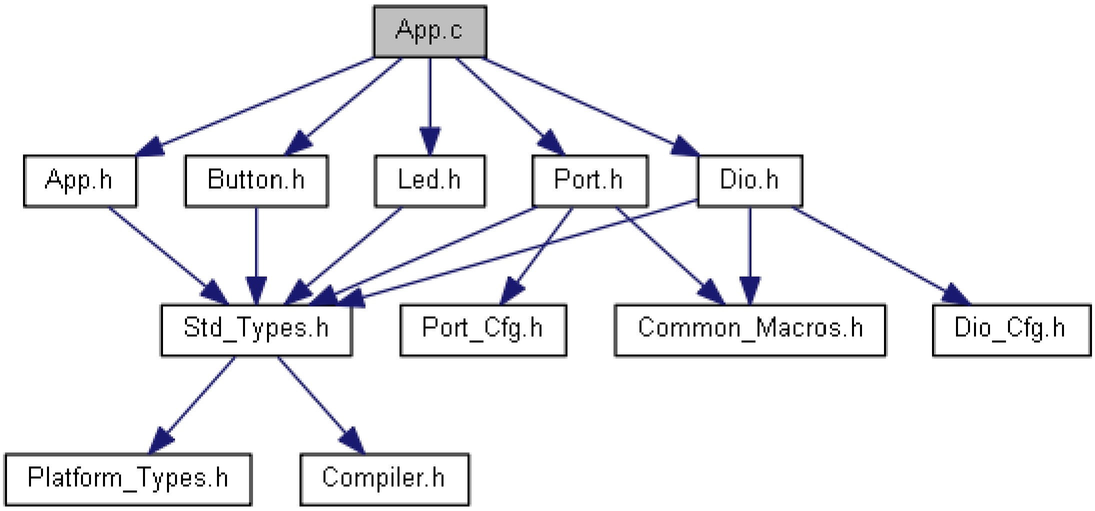
- **Std_Types.h**

File Documentation

App.c File Reference

```
#include "App.h"
#include "Button.h"
#include "Led.h"
#include "Port.h"
#include "Dio.h"
```

Include dependency graph for App.c:



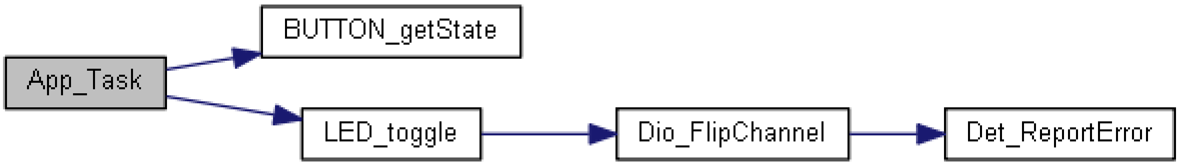
Functions

- void **Init_Task** (void)
- void **Button_Task** (void)
- void **Led_Task** (void)
- void **App_Task** (void)

Function Documentation

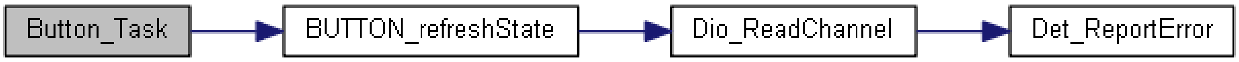
void App_Task (void)

Here is the call graph for this function:



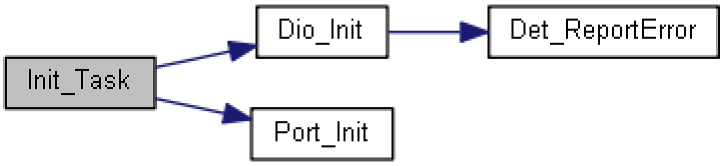
void Button_Task (void)

Here is the call graph for this function:

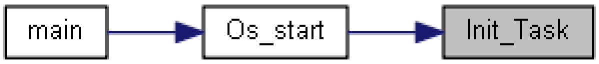


void Init_Task (void)

Here is the call graph for this function:

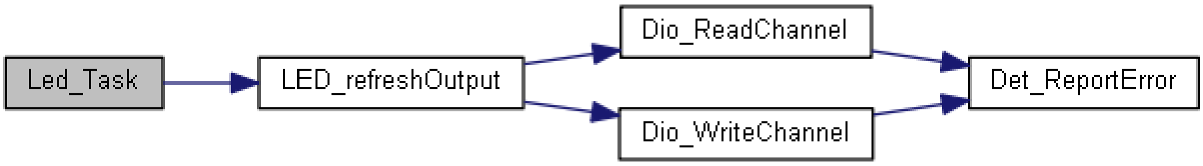


Here is the caller graph for this function:



void Led_Task (void)

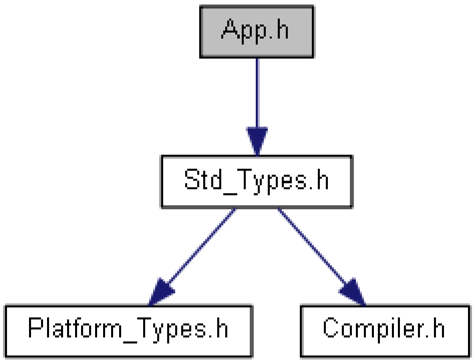
Here is the call graph for this function:



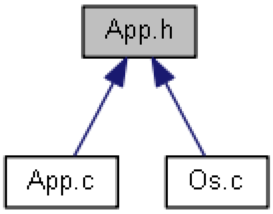
App.h File Reference

`#include "Std_Types.h"`

Include dependency graph for App.h:



This graph shows which files directly or indirectly include this file:



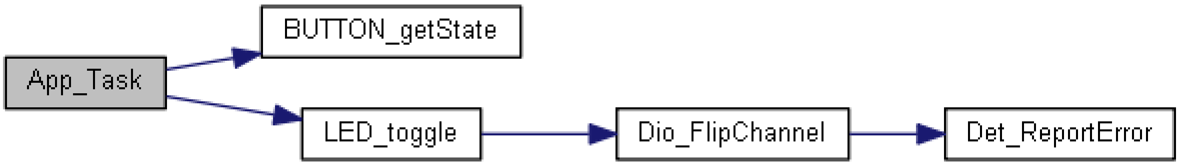
Functions

- void **Init_Task** (void)
- void **Button_Task** (void)
- void **Led_Task** (void)
- void **App_Task** (void)

Function Documentation

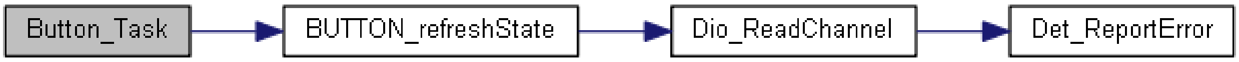
void App_Task (void)

Here is the call graph for this function:



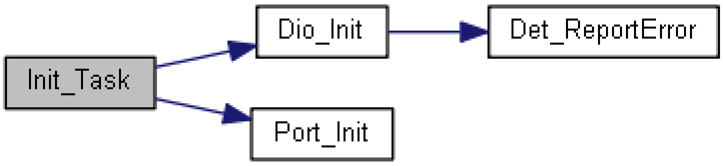
void Button_Task (void)

Here is the call graph for this function:

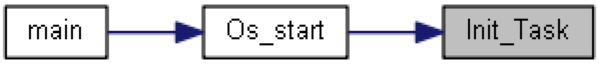


void Init_Task (void)

Here is the call graph for this function:

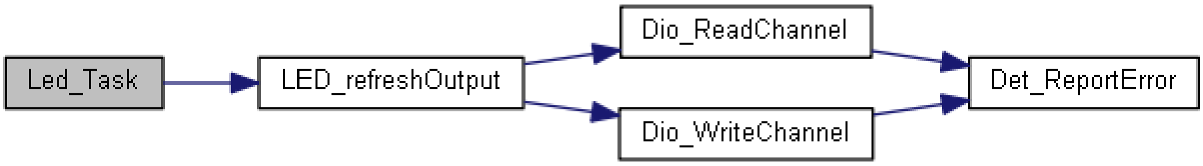


Here is the caller graph for this function:



void Led_Task (void)

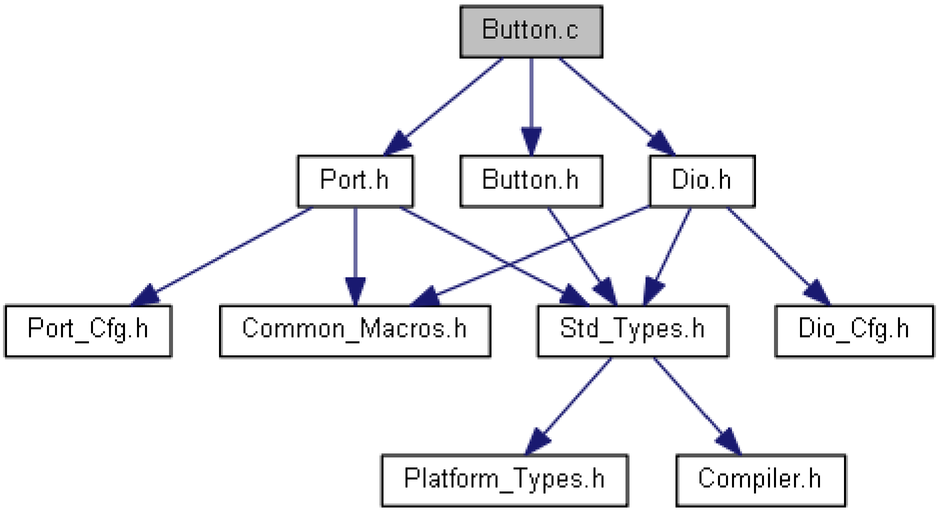
Here is the call graph for this function:



Button.c File Reference

```
#include "Dio.h"
#include "Port.h"
#include "Button.h"
```

Include dependency graph for Button.c:



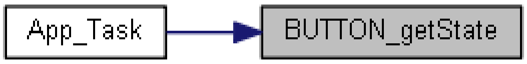
Functions

- **uint8 BUTTON_getState** (void)
- void **BUTTON_refreshState** (void)

Function Documentation

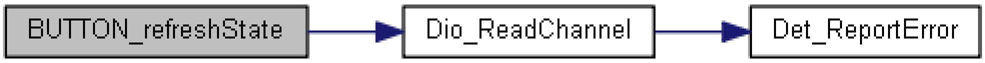
uint8 BUTTON_getState (void)

Here is the caller graph for this function:

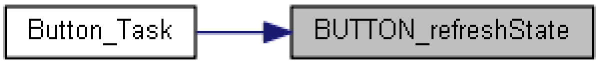


void BUTTON_refreshState (void)

Here is the call graph for this function:



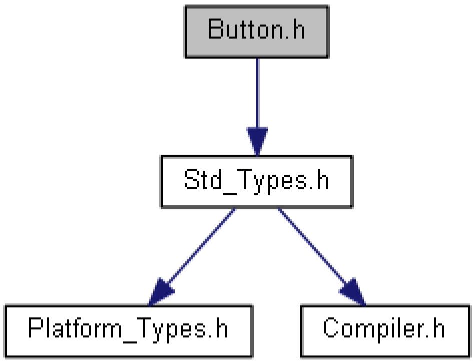
Here is the caller graph for this function:



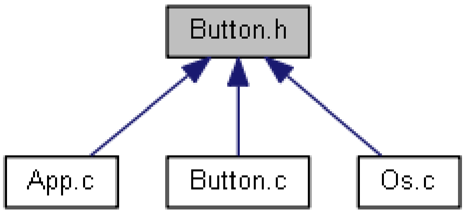
Button.h File Reference

`#include "Std_Types.h"`

Include dependency graph for Button.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **BUTTON_PRESSED** **STD_LOW**
- #define **BUTTON_RELEASED** **STD_HIGH**

Functions

- **uint8 BUTTON_getState** (void)
- void **BUTTON_refreshState** (void)

Macro Definition Documentation

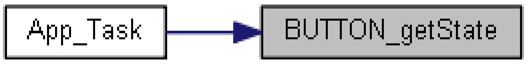
#define BUTTON_PRESSED   STD_LOW

#define BUTTON_RELEASED   STD_HIGH

Function Documentation

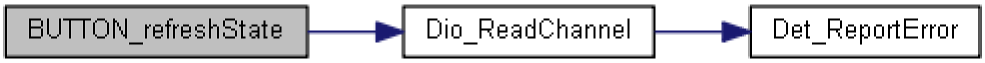
uint8 BUTTON_getState (void)

Here is the caller graph for this function:

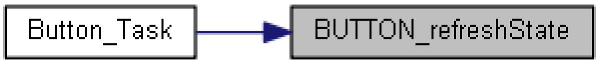


void BUTTON_refreshState (void)

Here is the call graph for this function:

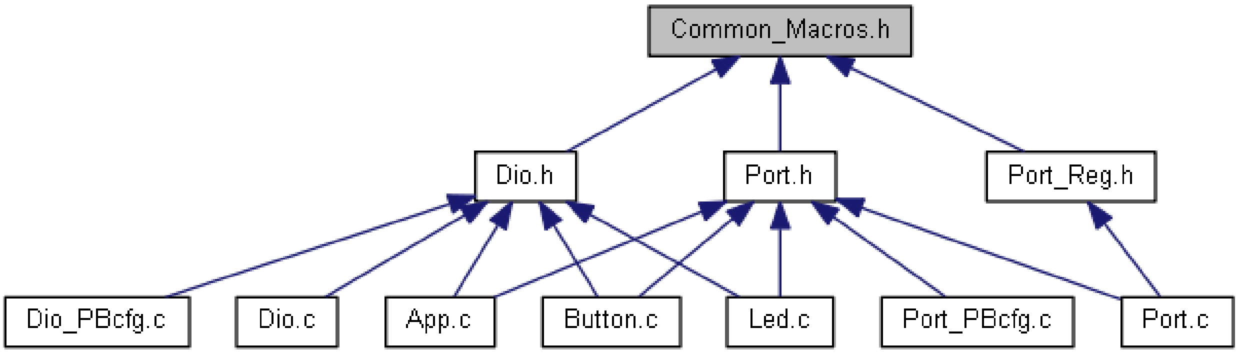


Here is the caller graph for this function:



Common_Macros.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define **SET_BIT**(REG, BIT) (REG|=(1<<BIT))
- #define **CLEAR_BIT**(REG, BIT) (REG&=(~(1<<BIT)))
- #define **TOGGLE_BIT**(REG, BIT) (REG^=(1<<BIT))
- #define **ROR**(REG, num) (REG = (REG>>num) | (REG << ((sizeof(REG) * 8)-num)))
- #define **ROL**(REG, num) (REG = (REG<<num) | (REG >> ((sizeof(REG) * 8)-num)))
- #define **BIT_IS_SET**(REG, BIT) (REG & (1<<BIT))
- #define **BIT_IS_CLEAR**(REG, BIT) (!(REG & (1<<BIT)))

Macro Definition Documentation

#define BIT_IS_CLEAR(REG, BIT) (!(REG & (1<<BIT)))

#define BIT_IS_SET(REG, BIT) (REG & (1<<BIT))

#define CLEAR_BIT(REG, BIT) (REG&=(~(1<<BIT)))

#define ROL(REG, num) (REG = (REG<<num) | (REG >> ((sizeof(REG) * 8)-num)))

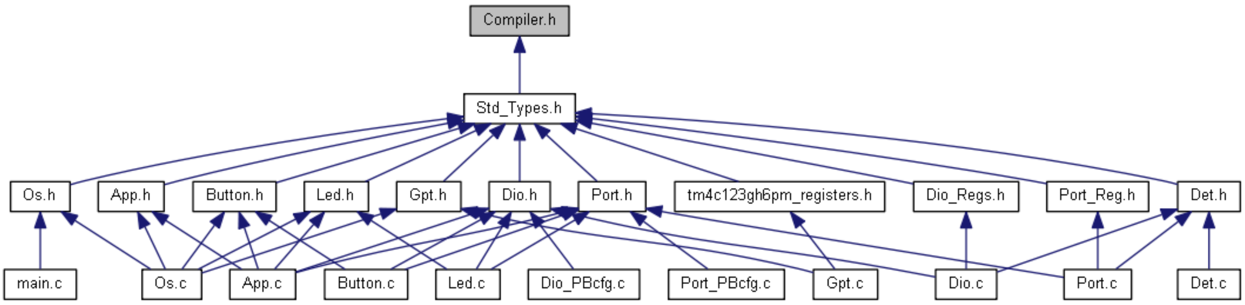
#define ROR(REG, num) (REG = (REG>>num) | (REG << ((sizeof(REG) * 8)-num)))

#define SET_BIT(REG, BIT) (REG|=(1<<BIT))

#define TOGGLE_BIT(REG, BIT) (REG^=(1<<BIT))

Compiler.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define **COMPILER_VENDOR_ID** (1000U)
- #define **COMPILER_SW_MAJOR_VERSION** (1U)
- #define **COMPILER_SW_MINOR_VERSION** (0U)
- #define **COMPILER_SW_PATCH_VERSION** (0U)
- #define **COMPILER_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **COMPILER_AR_RELEASE_MINOR_VERSION** (0U)
- #define **COMPILER_AR_RELEASE_PATCH_VERSION** (3U)
- #define **AUTOMATIC**
- #define **TYPEDEF**
- #define **NULL_PTR** ((void *)0)
- #define **INLINE** inline
- #define **LOCAL_INLINE** static inline
- #define **STATIC** static

Macro Definition Documentation

#define AUTOMATIC

#define COMPILER_AR_RELEASE_MAJOR_VERSION   (4U)

#define COMPILER_AR_RELEASE_MINOR_VERSION   (0U)

#define COMPILER_AR_RELEASE_PATCH_VERSION   (3U)

#define COMPILER_SW_MAJOR_VERSION   (1U)

#define COMPILER_SW_MINOR_VERSION   (0U)

#define COMPILER_SW_PATCH_VERSION   (0U)

#define COMPILER_VENDOR_ID   (1000U)

#define INLINE   inline

#define LOCAL_INLINE   static inline

#define NULL_PTR   ((void *)0)

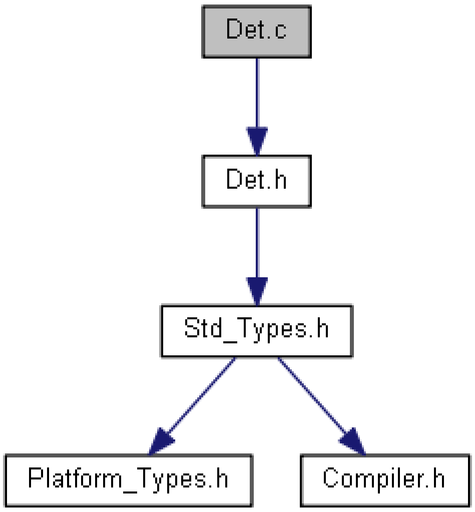
#define STATIC   static

#define TYPEDEF

Det.c File Reference

`#include "Det.h"`

Include dependency graph for Det.c:



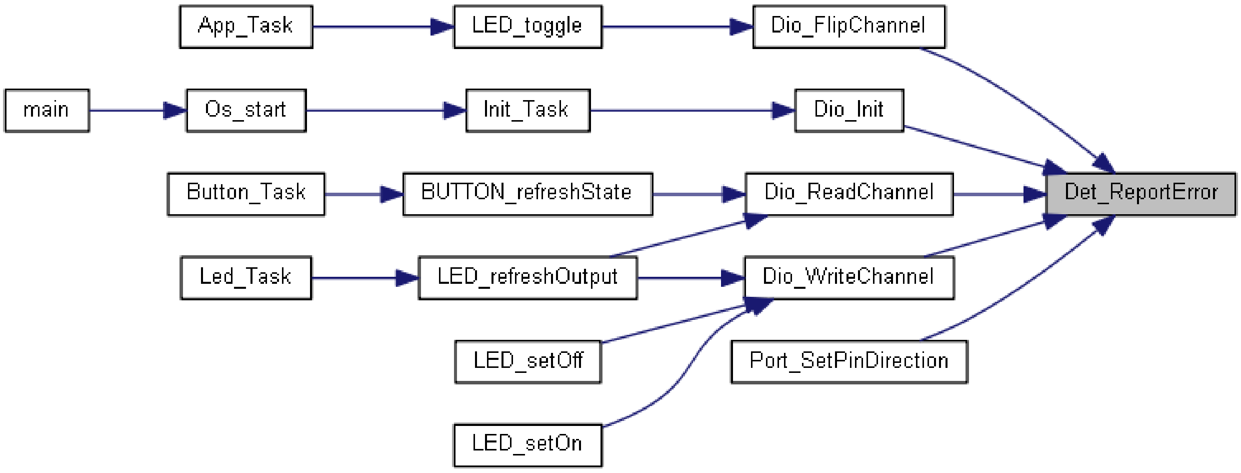
Functions

- **Std_ReturnType Det_ReportError** (**uint16** ModuleId, **uint8** InstanceId, **uint8** ApiId, **uint8** ErrorId)

Function Documentation

Std_ReturnType Det_ReportError (uint16 *ModuleId*, uint8 *InstanceId*, uint8 *ApiId*, uint8 *ErrorId*)

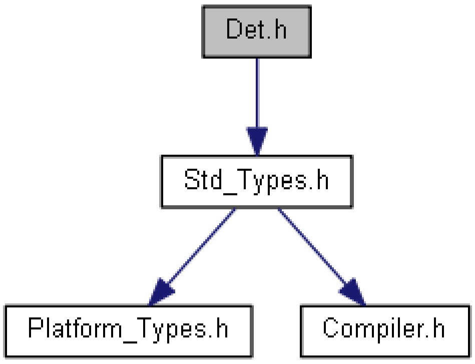
Here is the caller graph for this function:



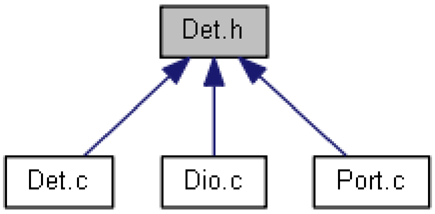
Det.h File Reference

`#include "Std_Types.h"`

Include dependency graph for Det.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **DET_VENDOR_ID** (1000U)
- #define **DET_MODULE_ID** (15U)
- #define **DET_INSTANCE_ID** (0U)
- #define **DET_SW_MAJOR_VERSION** (1U)
- #define **DET_SW_MINOR_VERSION** (0U)
- #define **DET_SW_PATCH_VERSION** (0U)
- #define **DET_AR_MAJOR_VERSION** (4U)
- #define **DET_AR_MINOR_VERSION** (0U)
- #define **DET_AR_PATCH_VERSION** (3U)

Functions

- **Std_ReturnType Det_ReportError** (**uint16** ModuleId, **uint8** InstanceId, **uint8** ApiId, **uint8** ErrorId)

Macro Definition Documentation

#define DET_AR_MAJOR_VERSION (4U)

#define DET_AR_MINOR_VERSION (0U)

#define DET_AR_PATCH_VERSION (3U)

#define DET_INSTANCE_ID (0U)

#define DET_MODULE_ID (15U)

#define DET_SW_MAJOR_VERSION (1U)

#define DET_SW_MINOR_VERSION (0U)

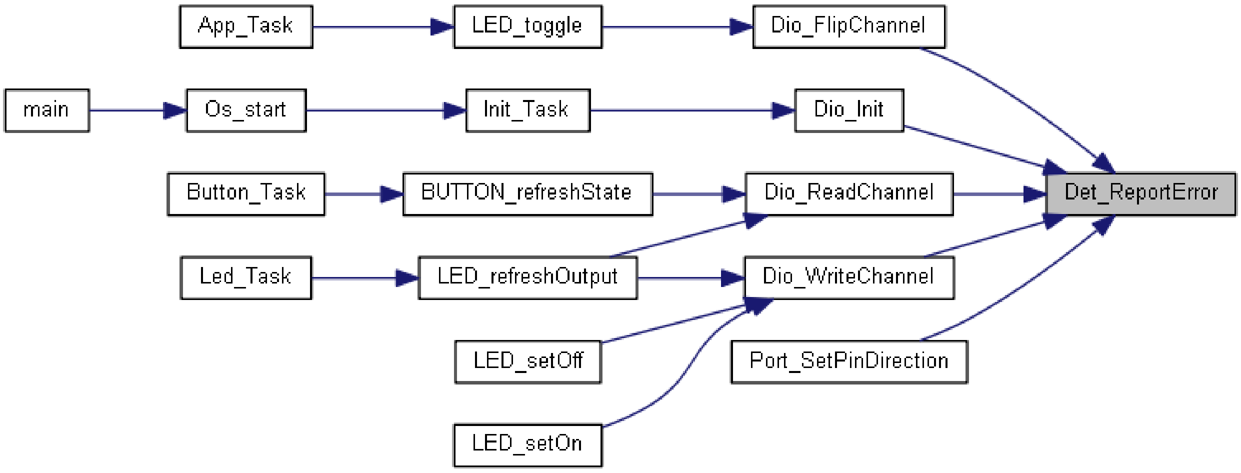
#define DET_SW_PATCH_VERSION (0U)

#define DET_VENDOR_ID (1000U)

Function Documentation

Std_ReturnType Det_ReportError (uint16 *ModuleId*, uint8 *InstanceId*, uint8 *ApiId*, uint8 *ErrorId*)

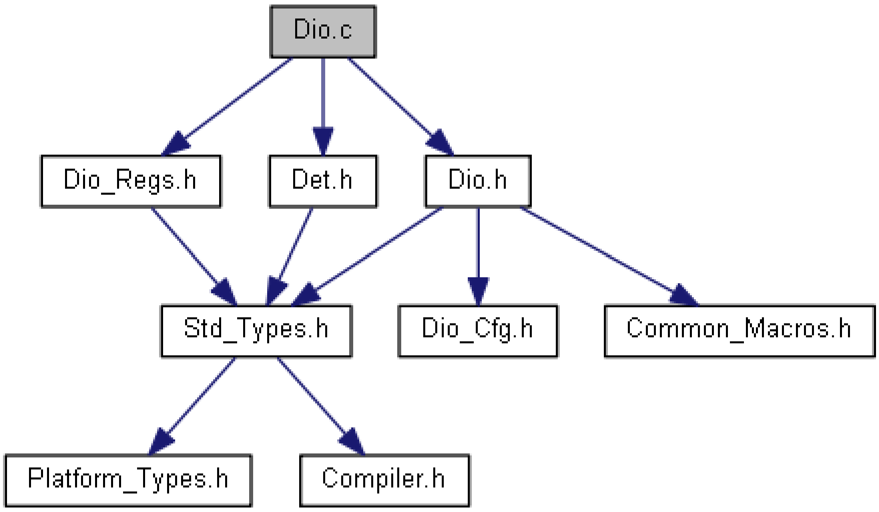
Here is the caller graph for this function:



Dio.c File Reference

```
#include "Dio.h"
#include "Dio_Regs.h"
#include "Det.h"
```

Include dependency graph for Dio.c:



Functions

- void **Dio_Init** (const **Dio_ConfigType** *ConfigPtr)
- void **Dio_WriteChannel** (**Dio_ChannelType** ChannelId, **Dio_LevelType** Level)
- **Dio_LevelType Dio_ReadChannel** (**Dio_ChannelType** ChannelId)
- **Dio_LevelType Dio_FlipChannel** (**Dio_ChannelType** ChannelId)

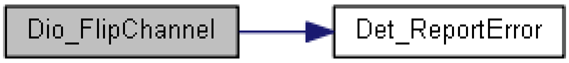
Variables

- const **STATIC Dio_ConfigChannel** * **Dio_PortChannels** = **NULL_PTR**
- **STATIC uint8 Dio_Status** = **DIO_NOT_INITIALIZED**

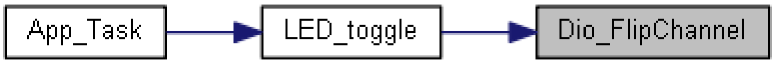
Function Documentation

Dio_LevelType Dio_FlipChannel (Dio_ChannelType *ChannelId*)

Here is the call graph for this function:

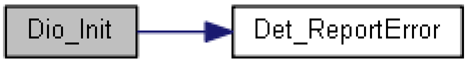


Here is the caller graph for this function:

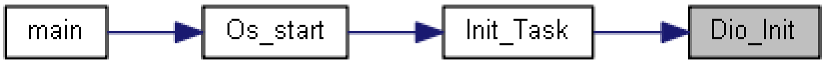


void Dio_Init (const Dio_ConfigType * *ConfigPtr*)

Here is the call graph for this function:

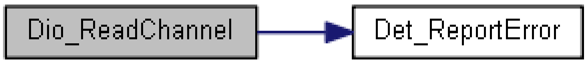


Here is the caller graph for this function:

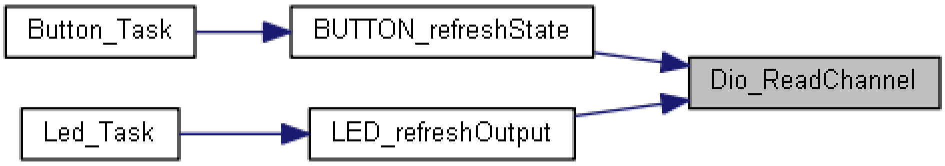


Dio_LevelType Dio_ReadChannel (Dio_ChannelType ***ChannelId*****)**

Here is the call graph for this function:

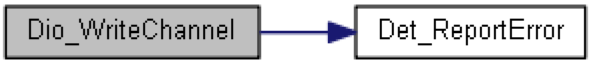


Here is the caller graph for this function:

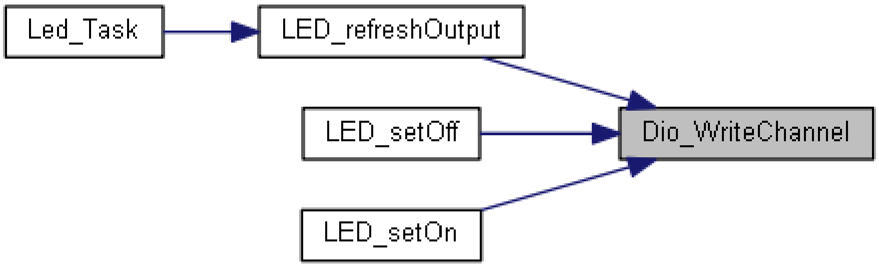


void Dio_WriteChannel (Dio_ChannelType ***ChannelId*****, Dio_LevelType** ***Level*****)**

Here is the call graph for this function:



Here is the caller graph for this function:



Variable Documentation

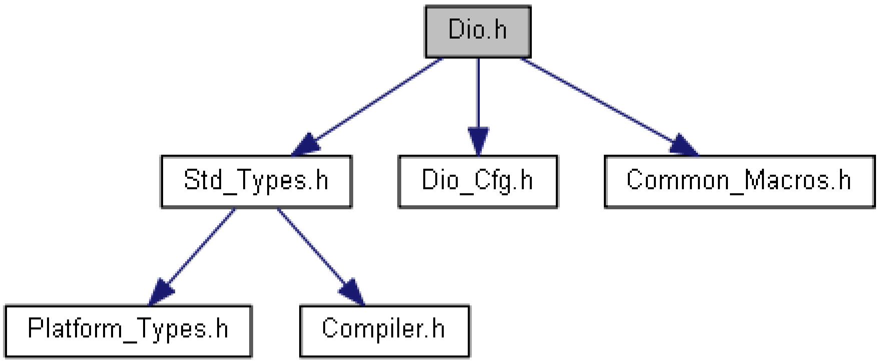
const STATIC Dio_ConfigChannel* Dio_PortChannels = NULL_PTR

STATIC uint8 Dio_Status = DIO_NOT_INITIALIZED

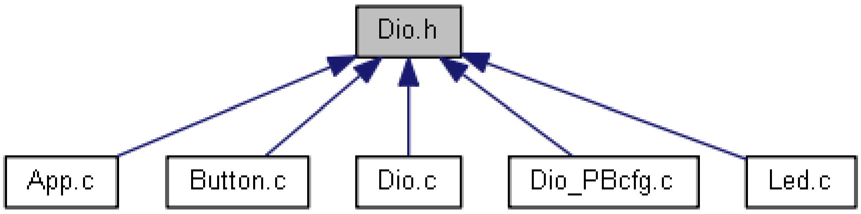
Dio.h File Reference

```
#include "Std_Types.h"
#include "Dio_Cfg.h"
#include "Common_Macros.h"
```

Include dependency graph for Dio.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Dio_ChannelGroupType**
- struct **Dio_ConfigChannel**
- struct **Dio_ConfigType**

Macros

- #define **DIO_VENDOR_ID** (1000U)
- #define **DIO_MODULE_ID** (120U)
- #define **DIO_INSTANCE_ID** (0U)
- #define **DIO_SW_MAJOR_VERSION** (1U)
- #define **DIO_SW_MINOR_VERSION** (0U)
- #define **DIO_SW_PATCH_VERSION** (0U)
- #define **DIO_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **DIO_AR_RELEASE_MINOR_VERSION** (0U)
- #define **DIO_AR_RELEASE_PATCH_VERSION** (3U)
- #define **DIO_INITIALIZED** (1U)
- #define **DIO_NOT_INITIALIZED** (0U)
- #define **DIO_READ_CHANNEL_SID** (**uint8**)0x00
- #define **DIO_WRITE_CHANNEL_SID** (**uint8**)0x01
- #define **DIO_READ_PORT_SID** (**uint8**)0x02
- #define **DIO_WRITE_PORT_SID** (**uint8**)0x03
- #define **DIO_READ_CHANNEL_GROUP_SID** (**uint8**)0x04
- #define **DIO_WRITE_CHANNEL_GROUP_SID** (**uint8**)0x05
- #define **DIO_GET_VERSION_INFO_SID** (**uint8**)0x12
- #define **DIO_INIT_SID** (**uint8**)0x10
- #define **DIO_FLIP_CHANNEL_SID** (**uint8**)0x11

- #define **DIO_E_PARAM_INVALID_CHANNEL_ID** (**uint8**)0x0A
- #define **DIO_E_PARAM_CONFIG** (**uint8**)0x10
- #define **DIO_E_PARAM_INVALID_PORT_ID** (**uint8**)0x14
- #define **DIO_E_PARAM_INVALID_GROUP** (**uint8**)0x1F
- #define **DIO_E_PARAM_POINTER** (**uint8**)0x20
- #define **DIO_E_UNINIT** (**uint8**)0xF0

Typedefs

- typedef **uint8 Dio_ChannelType**
- typedef **uint8 Dio_PortType**
- typedef **uint8 Dio_LevelType**
- typedef **uint8 Dio_PortLevelType**
- typedef struct **Dio_ConfigType Dio_ConfigType**

Functions

- **Dio_PortLevelType Dio_ReadPort** (**Dio_PortType** PortId)
- void **Dio_WritePort** (**Dio_PortType** PortId, **Dio_PortLevelType** Level)
- **Dio_LevelType Dio_ReadChannel** (**Dio_ChannelType** ChannelId)
- void **Dio_WriteChannel** (**Dio_ChannelType** ChannelId, **Dio_LevelType** Level)
- void **Dio_Init** (const **Dio_ConfigType** *ConfigPtr)
- **Dio_LevelType Dio_FlipChannel** (**Dio_ChannelType** ChannelId)

Variables

- const **Dio_ConfigType Dio_Configuration**

Macro Definition Documentation

#define DIO_AR_RELEASE_MAJOR_VERSION   (4U)

#define DIO_AR_RELEASE_MINOR_VERSION   (0U)

#define DIO_AR_RELEASE_PATCH_VERSION   (3U)

#define DIO_E_PARAM_CONFIG   (uint8)0x10

#define DIO_E_PARAM_INVALID_CHANNEL_ID   (uint8)0x0A

#define DIO_E_PARAM_INVALID_GROUP   (uint8)0x1F

#define DIO_E_PARAM_INVALID_PORT_ID   (uint8)0x14

#define DIO_E_PARAM_POINTER   (uint8)0x20

#define DIO_E_UNINIT   (uint8)0xF0

#define DIO_FLIP_CHANNEL_SID   (uint8)0x11

#define DIO_GET_VERSION_INFO_SID   (uint8)0x12

#define DIO_INIT_SID   (uint8)0x10

#define DIO_INITIALIZED   (1U)

#define DIO_INSTANCE_ID   (0U)

#define DIO_MODULE_ID   (120U)

#define DIO_NOT_INITIALIZED   (0U)

#define DIO_READ_CHANNEL_GROUP_SID   (uint8)0x04

#define DIO_READ_CHANNEL_SID   (uint8)0x00

#define DIO_READ_PORT_SID   (uint8)0x02

#define DIO_SW_MAJOR_VERSION   (1U)

#define DIO_SW_MINOR_VERSION   (0U)

#define DIO_SW_PATCH_VERSION   (0U)

#define DIO_VENDOR_ID   (1000U)

#define DIO_WRITE_CHANNEL_GROUP_SID   (uint8)0x05

#define DIO_WRITE_CHANNEL_SID   (uint8)0x01

#define DIO_WRITE_PORT_SID (uint8)0x03

Typedef Documentation

typedef uint8 Dio_ChannelType

typedef struct Dio_ConfigType Dio_ConfigType

typedef uint8 Dio_LevelType

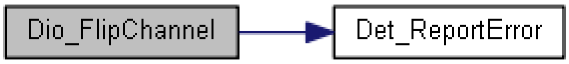
typedef uint8 Dio_PortLevelType

typedef uint8 Dio_PortType

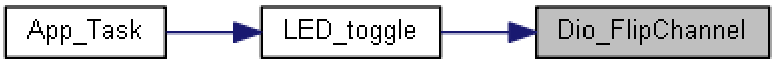
Function Documentation

Dio_LevelType Dio_FlipChannel (Dio_ChannelType *ChannelId*)

Here is the call graph for this function:

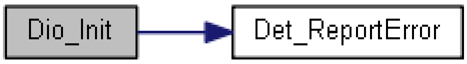


Here is the caller graph for this function:

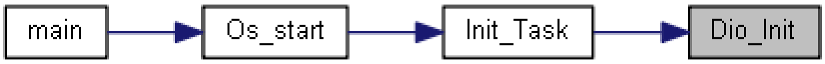


void Dio_Init (const Dio_ConfigType * *ConfigPtr*)

Here is the call graph for this function:

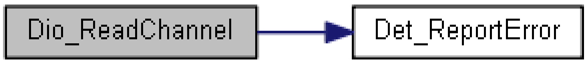


Here is the caller graph for this function:

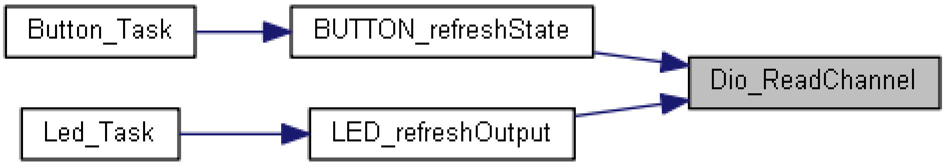


Dio_LevelType Dio_ReadChannel (Dio_ChannelType *ChannelId*)

Here is the call graph for this function:



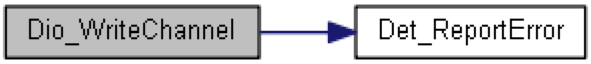
Here is the caller graph for this function:



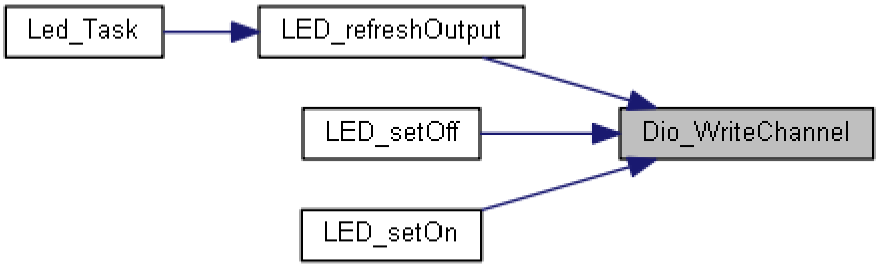
Dio_PortLevelType Dio_ReadPort (Dio_PortType *PortId*)

void Dio_WriteChannel (Dio_ChannelType *ChannelId*, Dio_LevelType *Level*)

Here is the call graph for this function:



Here is the caller graph for this function:



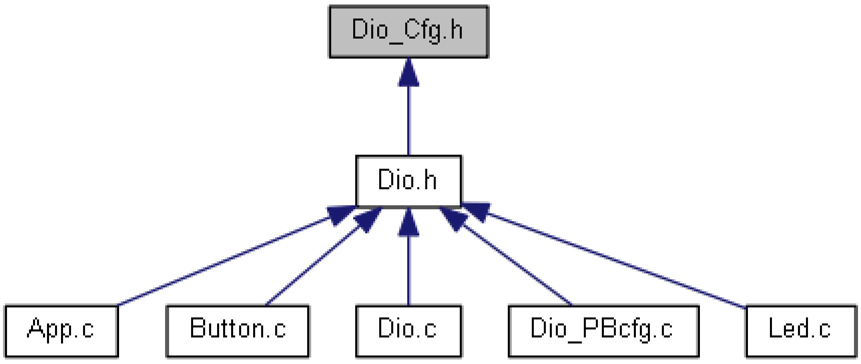
void Dio_WritePort (Dio_PortType *PortId*, Dio_PortLevelType *Level*)

Variable Documentation

const Dio_ConfigType Dio_Configuration

Dio_Cfg.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define **DIO_CFG_SW_MAJOR_VERSION**   (1U)
- #define **DIO_CFG_SW_MINOR_VERSION**   (0U)
- #define **DIO_CFG_SW_PATCH_VERSION**   (0U)
- #define **DIO_CFG_AR_RELEASE_MAJOR_VERSION**   (4U)
- #define **DIO_CFG_AR_RELEASE_MINOR_VERSION**   (0U)
- #define **DIO_CFG_AR_RELEASE_PATCH_VERSION**   (3U)
- #define **DIO_DEV_ERROR_DETECT**   (**STD_ON**)
- #define **DIO_VERSION_INFO_API**   (**STD_OFF**)
- #define **DIO_FLIP_CHANNEL_API**   (**STD_ON**)
- #define **DIO_CONFIGURED_CHANNLES**   (2U)
- #define **DioConf_LED1_CHANNEL_ID_INDEX**   (**uint8**)0x00
- #define **DioConf_SW1_CHANNEL_ID_INDEX**   (**uint8**)0x01
- #define **DioConf_LED1_PORT_NUM**   (**Dio_PortType**)5 /* **PORTF** */
- #define **DioConf_SW1_PORT_NUM**   (**Dio_PortType**)5 /* **PORTF** */
- #define **DioConf_LED1_CHANNEL_NUM**   (**Dio_ChannelType**)2 /* Pin 1 in **PORTF** */
- #define **DioConf_SW1_CHANNEL_NUM**   (**Dio_ChannelType**)4 /* Pin 4 in **PORTF** */

Macro Definition Documentation

#define DIO_CFG_AR_RELEASE_MAJOR_VERSION   (4U)

#define DIO_CFG_AR_RELEASE_MINOR_VERSION   (0U)

#define DIO_CFG_AR_RELEASE_PATCH_VERSION   (3U)

#define DIO_CFG_SW_MAJOR_VERSION   (1U)

#define DIO_CFG_SW_MINOR_VERSION   (0U)

#define DIO_CFG_SW_PATCH_VERSION   (0U)

#define DIO_CONFIGURED_CHANNLES   (2U)

#define DIO_DEV_ERROR_DETECT   (STD_ON)

#define DIO_FLIP_CHANNEL_API   (STD_ON)

#define DIO_VERSION_INFO_API   (STD_OFF)

#define DioConf_LED1_CHANNEL_ID_INDEX   (uint8)0x00

#define DioConf_LED1_CHANNEL_NUM   (Dio_ChannelType)2 /* Pin 1 in PORTF */

#define DioConf_LED1_PORT_NUM   (Dio_PortType)5 /* PORTF */

#define DioConf_SW1_CHANNEL_ID_INDEX   (uint8)0x01

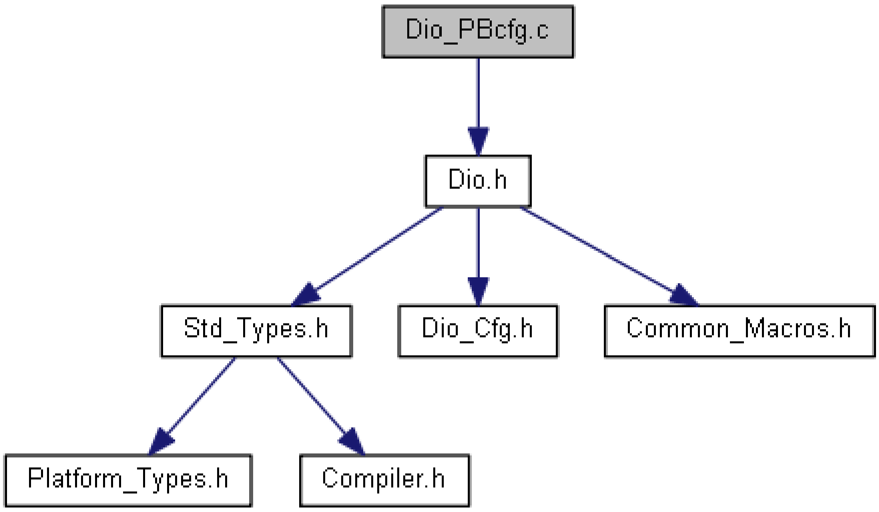
#define DioConf_SW1_CHANNEL_NUM   (Dio_ChannelType)4 /* Pin 4 in PORTF */

#define DioConf_SW1_PORT_NUM   (Dio_PortType)5 /* PORTF */

Dio_PBcfg.c File Reference

`#include "Dio.h"`

Include dependency graph for Dio_PBcfg.c:



Macros

- #define **DIO_PBCFG_SW_MAJOR_VERSION** (1U)
- #define **DIO_PBCFG_SW_MINOR_VERSION** (0U)
- #define **DIO_PBCFG_SW_PATCH_VERSION** (0U)
- #define **DIO_PBCFG_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **DIO_PBCFG_AR_RELEASE_MINOR_VERSION** (0U)
- #define **DIO_PBCFG_AR_RELEASE_PATCH_VERSION** (3U)

Variables

- const **Dio_ConfigType Dio_Configuration**

Macro Definition Documentation

#define DIO_PBCFG_AR_RELEASE_MAJOR_VERSION (4U)

#define DIO_PBCFG_AR_RELEASE_MINOR_VERSION (0U)

#define DIO_PBCFG_AR_RELEASE_PATCH_VERSION (3U)

#define DIO_PBCFG_SW_MAJOR_VERSION (1U)

#define DIO_PBCFG_SW_MINOR_VERSION (0U)

#define DIO_PBCFG_SW_PATCH_VERSION (0U)

Variable Documentation

const Dio_ConfigType Dio_Configuration

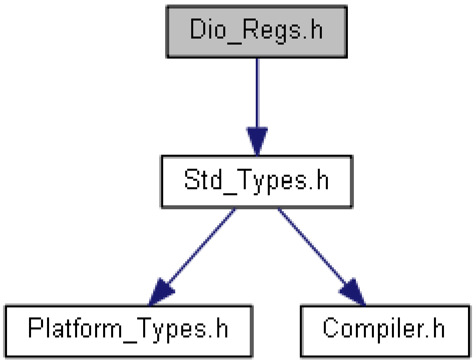
```
Initial value:= {
```

```
DioConf_LED1_PORT_NUM,DioConf_LED1_CHANNEL_NUM,
                                   DioConf_SW1_PORT_NUM,DioConf_SW1_CHANNEL_NUM
                              }
```

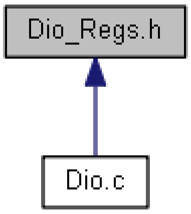
Dio_Regs.h File Reference

```
#include "Std_Types.h"
```

Include dependency graph for Dio_Regs.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **GPIO_PORTA_DATA_REG** (*((volatile **uint32** *)0x400043FC))
- #define **GPIO_PORTB_DATA_REG** (*((volatile **uint32** *)0x400053FC))
- #define **GPIO_PORTC_DATA_REG** (*((volatile **uint32** *)0x400063FC))
- #define **GPIO_PORTD_DATA_REG** (*((volatile **uint32** *)0x400073FC))
- #define **GPIO_PORTE_DATA_REG** (*((volatile **uint32** *)0x400243FC))
- #define **GPIO_PORTF_DATA_REG** (*((volatile **uint32** *)0x400253FC))

Macro Definition Documentation

#define GPIO_PORTA_DATA_REG (*((volatile uint32 *)0x400043FC))

#define GPIO_PORTB_DATA_REG (*((volatile uint32 *)0x400053FC))

#define GPIO_PORTC_DATA_REG (*((volatile uint32 *)0x400063FC))

#define GPIO_PORTD_DATA_REG (*((volatile uint32 *)0x400073FC))

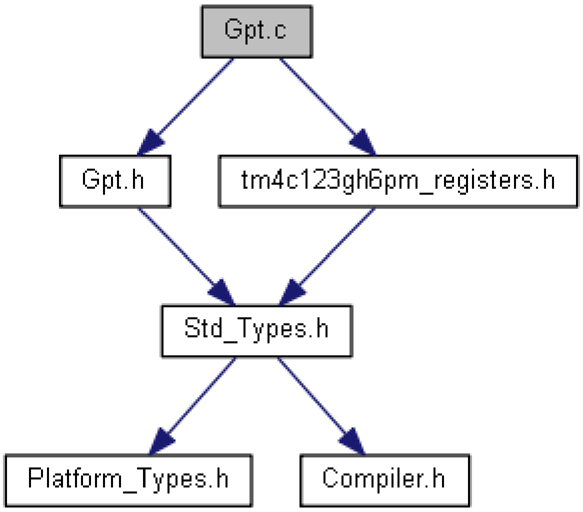
#define GPIO_PORTE_DATA_REG (*((volatile uint32 *)0x400243FC))

#define GPIO_PORTF_DATA_REG (*((volatile uint32 *)0x400253FC))

Gpt.c File Reference

```
#include "Gpt.h"
#include "tm4c123gh6pm_registers.h"
```

Include dependency graph for Gpt.c:



Macros

- #define **SYSTICK_PRIORITY_MASK**   0x1FFFFFFF
- #define **SYSTICK_INTERRUPT_PRIORITY**   3
- #define **SYSTICK_PRIORITY_BITS_POS**   29

Functions

- void **SysTick_Handler** (void)
- void **SysTick_Start** (**uint16** Tick_Time)
- void **SysTick_Stop** (void)
- void **SysTick_SetCallBack** (void(*Ptr2Func)(void))

Macro Definition Documentation

#define SYSTICK_INTERRUPT_PRIORITY   3

#define SYSTICK_PRIORITY_BITS_POS   29

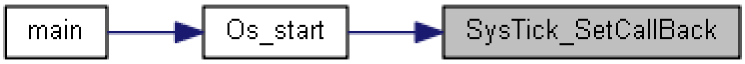
#define SYSTICK_PRIORITY_MASK   0x1FFFFFFF

Function Documentation

void SysTick_Handler (void)

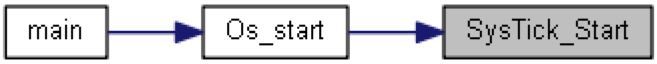
void SysTick_SetCallBack (void(*)(void)   *Ptr2Func*)

Here is the caller graph for this function:



void SysTick_Start (uint16 *Tick_Time*)

Here is the caller graph for this function:

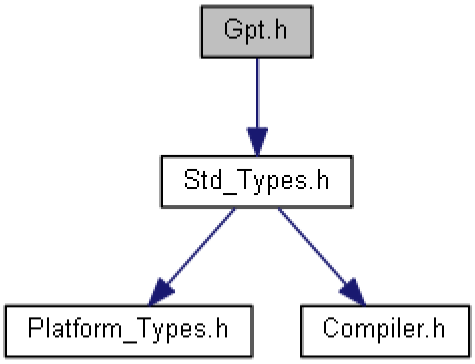


void SysTick_Stop (void)

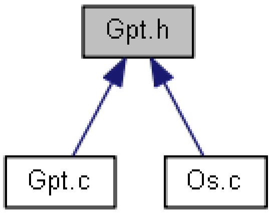
Gpt.h File Reference

```
#include "Std_Types.h"
```

Include dependency graph for Gpt.h:



This graph shows which files directly or indirectly include this file:



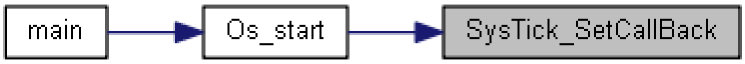
Functions

- void **SysTick_Start** (**uint16** Tick_Time)
- void **SysTick_Stop** (void)
- void **SysTick_SetCallBack** (void(*Ptr2Func)(void))

Function Documentation

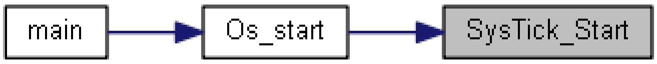
void SysTick_SetCallBack (void(*)(void) *Ptr2Func*)

Here is the caller graph for this function:



void SysTick_Start (uint16 *Tick_Time*)

Here is the caller graph for this function:

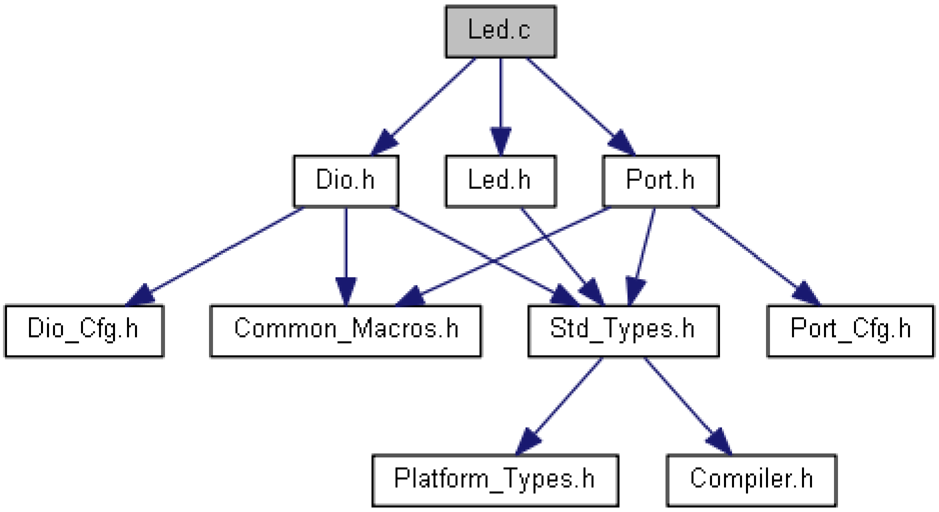


void SysTick_Stop (void)

Led.c File Reference

```
#include "Port.h"
#include "Dio.h"
#include "Led.h"
```

Include dependency graph for Led.c:



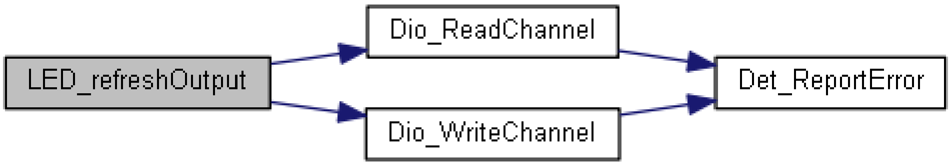
Functions

- void **LED_setOn** (void)
- void **LED_setOff** (void)
- void **LED_refreshOutput** (void)
- void **LED_toggle** (void)

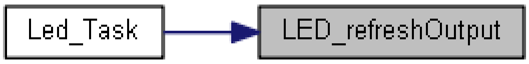
Function Documentation

void LED_refreshOutput (void)

Here is the call graph for this function:

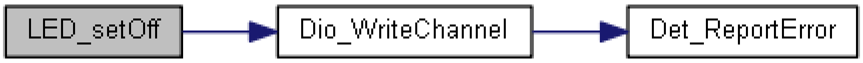


Here is the caller graph for this function:



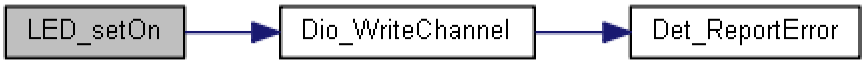
void LED_setOff (void)

Here is the call graph for this function:



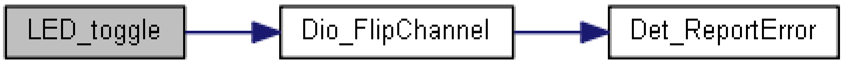
void LED_setOn (void)

Here is the call graph for this function:

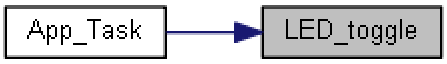


void LED_toggle (void)

Here is the call graph for this function:



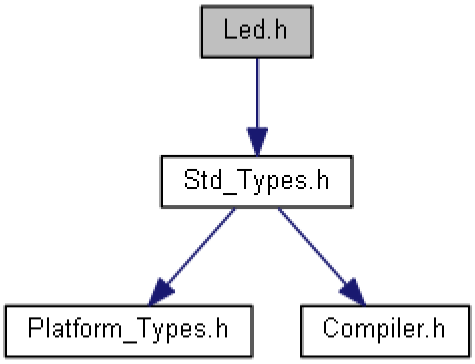
Here is the caller graph for this function:



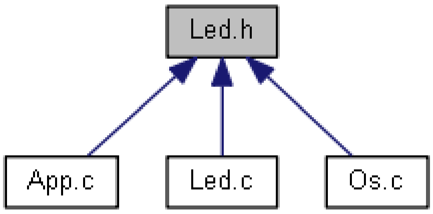
Led.h File Reference

`#include "Std_Types.h"`

Include dependency graph for Led.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **LED_ON   STD_HIGH**
- #define **LED_OFF   STD_LOW**
- #define **LED_PORT   DioConf_LED1_PORT_NUM**
- #define **LED_PIN_NUM   DioConf_LED1_CHANNEL_NUM**

Functions

- void **LED_setOn** (void)
- void **LED_setOff** (void)
- void **LED_toggle** (void)
- void **LED_refreshOutput** (void)

Macro Definition Documentation

#define LED_OFF   STD_LOW

#define LED_ON   STD_HIGH

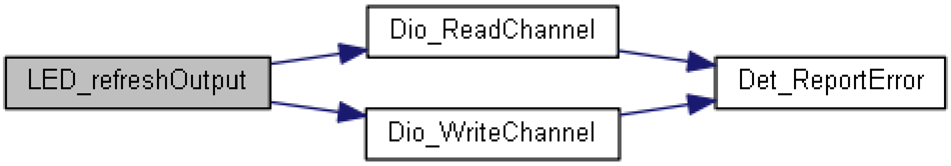
#define LED_PIN_NUM   DioConf_LED1_CHANNEL_NUM

#define LED_PORT   DioConf_LED1_PORT_NUM

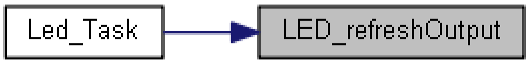
Function Documentation

void LED_refreshOutput (void)

Here is the call graph for this function:

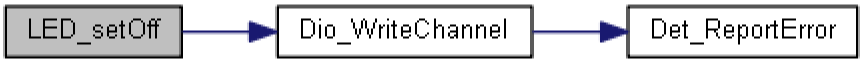


Here is the caller graph for this function:



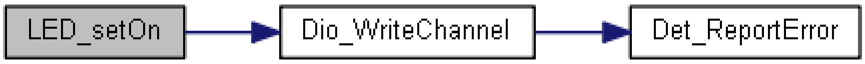
void LED_setOff (void)

Here is the call graph for this function:



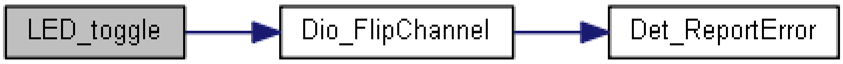
void LED_setOn (void)

Here is the call graph for this function:

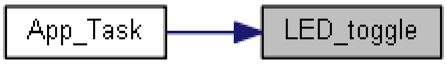


void LED_toggle (void)

Here is the call graph for this function:



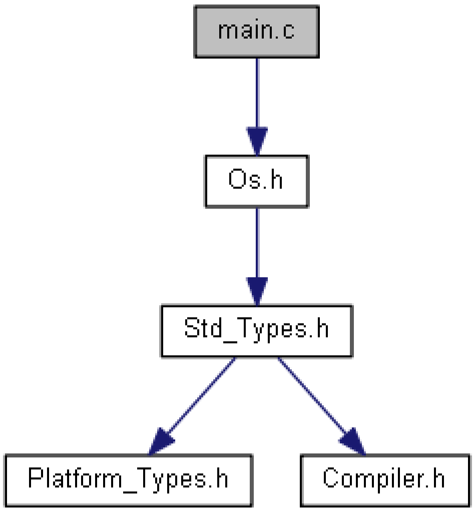
Here is the caller graph for this function:



main.c File Reference

`#include "Os.h"`

Include dependency graph for main.c:



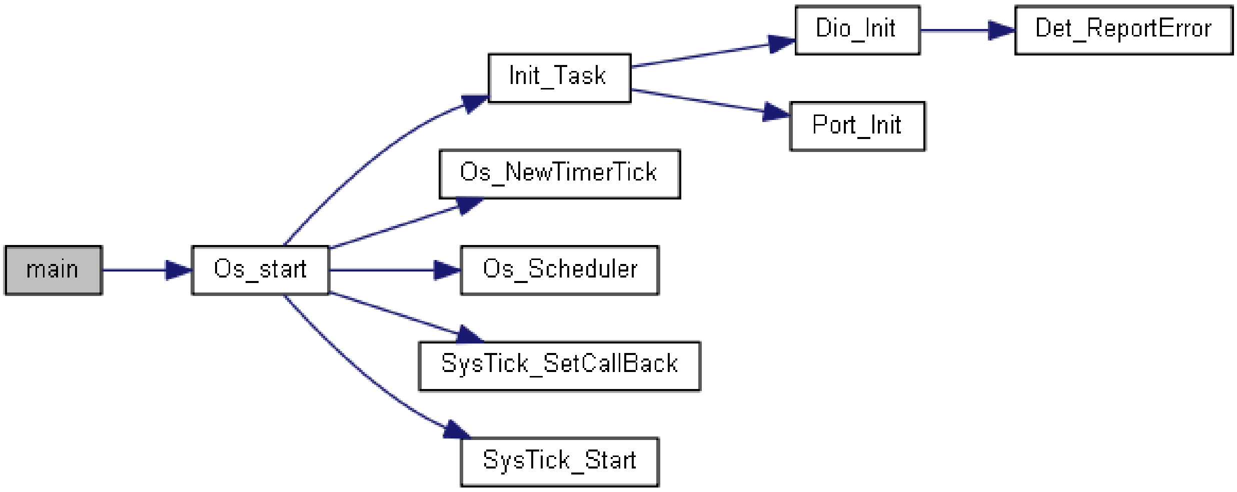
Functions

- int **main** (void)

Function Documentation

int main (void)

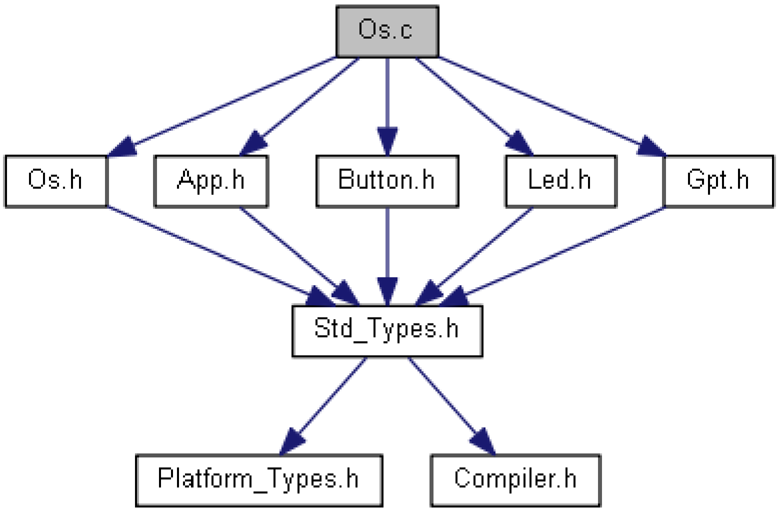
Here is the call graph for this function:



Os.c File Reference

```
#include "Os.h"
#include "App.h"
#include "Button.h"
#include "Led.h"
#include "Gpt.h"
```

Include dependency graph for Os.c:



Macros

- #define **Enable_Interrupts**() __asm(" CPSIE I")
- #define **Disable_Interrupts**() __asm(" CPSID I")

Functions

- void **Os_start** (void)
- void **Os_NewTimerTick** (void)
- void **Os_Scheduler** (void)

Macro Definition Documentation

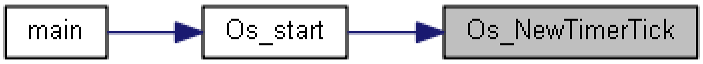
#define Disable_Interrupts() __asm(" CPSID I")

#define Enable_Interrupts() __asm(" CPSIE I")

Function Documentation

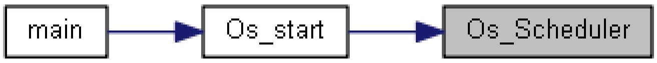
void Os_NewTimerTick (void)

Here is the caller graph for this function:



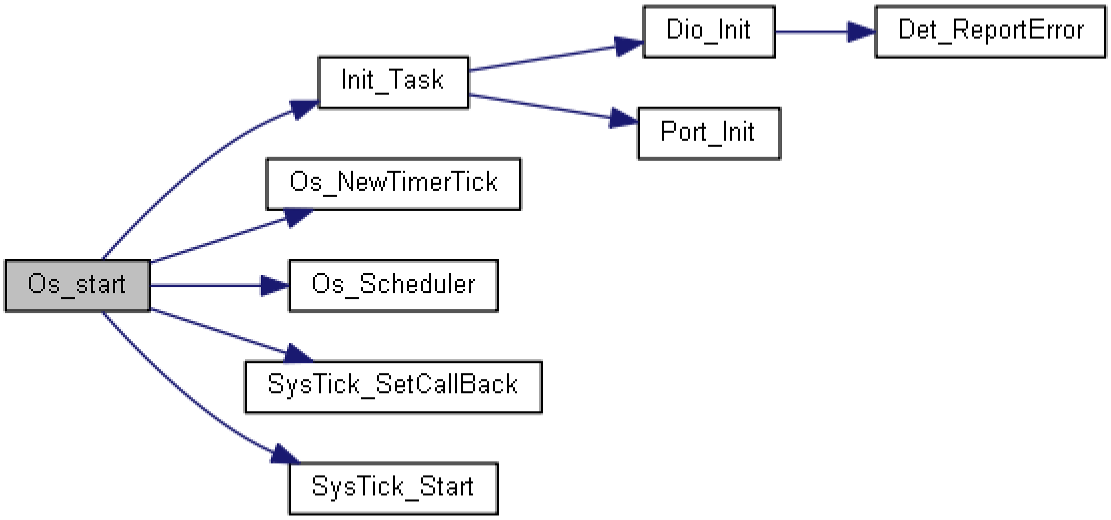
void Os_Scheduler (void)

Here is the caller graph for this function:

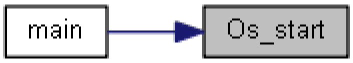


void Os_start (void)

Here is the call graph for this function:



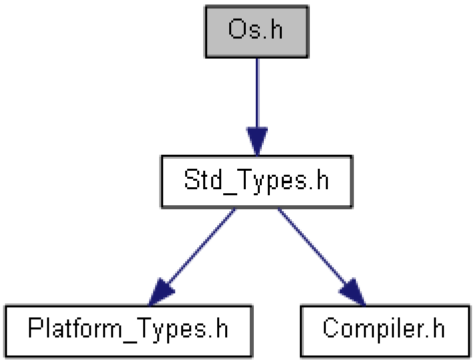
Here is the caller graph for this function:



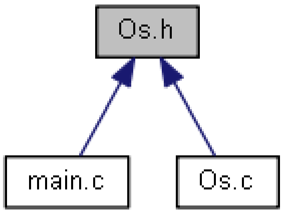
Os.h File Reference

`#include "Std_Types.h"`

Include dependency graph for Os.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **OS_BASE_TIME** 20

Functions

- void **Os_start** (void)
- void **Os_Scheduler** (void)
- void **Os_NewTimerTick** (void)

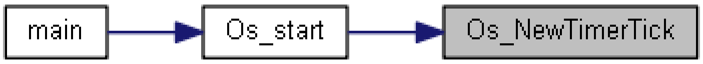
Macro Definition Documentation

#define OS_BASE_TIME 20

Function Documentation

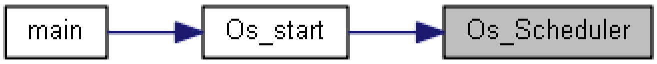
void Os_NewTimerTick (void)

Here is the caller graph for this function:



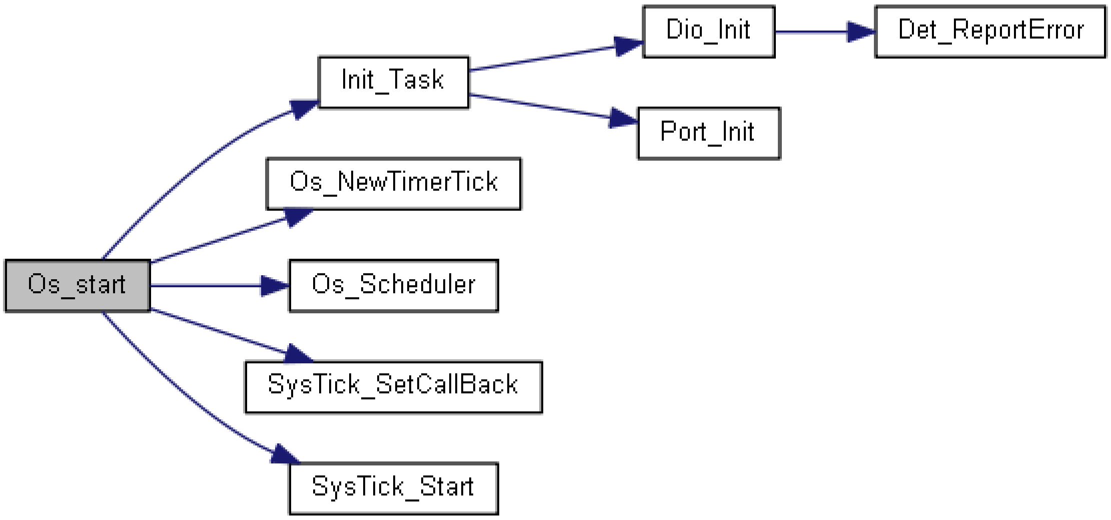
void Os_Scheduler (void)

Here is the caller graph for this function:

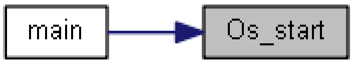


void Os_start (void)

Here is the call graph for this function:

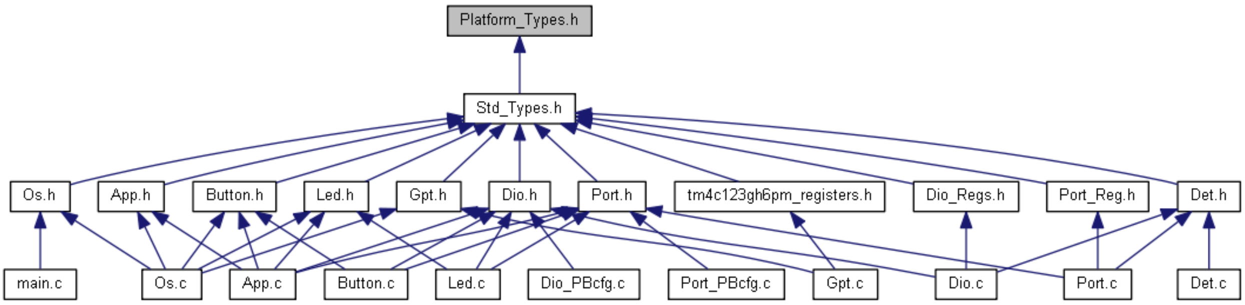


Here is the caller graph for this function:



Platform_Types.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define **PLATFORM_VENDOR_ID** (1000U)
- #define **PLATFORM_SW_MAJOR_VERSION** (1U)
- #define **PLATFORM_SW_MINOR_VERSION** (0U)
- #define **PLATFORM_SW_PATCH_VERSION** (0U)
- #define **PLATFORM_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **PLATFORM_AR_RELEASE_MINOR_VERSION** (0U)
- #define **PLATFORM_AR_RELEASE_PATCH_VERSION** (3U)
- #define **CPU_TYPE_8** (8U)
- #define **CPU_TYPE_16** (16U)
- #define **CPU_TYPE_32** (32U)
- #define **MSB_FIRST** (0u) /* Big endian bit ordering */
- #define **LSB_FIRST** (1u) /* Little endian bit ordering */
- #define **HIGH_BYTE_FIRST** (0u) /* Big endian byte ordering */
- #define **LOW_BYTE_FIRST** (1u) /* Little endian byte ordering */
- #define **CPU_TYPE** **CPU_TYPE_32**
- #define **CPU_BIT_ORDER** **LSB_FIRST**
- #define **CPU_BYTE_ORDER** **LOW_BYTE_FIRST**
- #define **FALSE** (0u)
- #define **TRUE** (1u)

Typedefs

- typedef unsigned char **boolean**
- typedef unsigned char **uint8**
- typedef signed char **sint8**
- typedef unsigned short **uint16**
- typedef signed short **sint16**
- typedef unsigned long **uint32**
- typedef signed long **sint32**
- typedef unsigned long long **uint64**
- typedef signed long long **sint64**
- typedef float **float32**
- typedef double **float64**

Macro Definition Documentation

#define CPU_BIT_ORDER   LSB_FIRST

#define CPU_BYTE_ORDER   LOW_BYTE_FIRST

#define CPU_TYPE   CPU_TYPE_32

#define CPU_TYPE_16   (16U)

#define CPU_TYPE_32   (32U)

#define CPU_TYPE_8   (8U)

#define FALSE   (0u)

#define HIGH_BYTE_FIRST   (0u)   /* Big endian byte ordering   */

#define LOW_BYTE_FIRST   (1u)   /* Little endian byte ordering   */

#define LSB_FIRST   (1u)   /* Little endian bit ordering   */

#define MSB_FIRST   (0u)   /* Big endian bit ordering   */

#define PLATFORM_AR_RELEASE_MAJOR_VERSION   (4U)

#define PLATFORM_AR_RELEASE_MINOR_VERSION   (0U)

#define PLATFORM_AR_RELEASE_PATCH_VERSION   (3U)

#define PLATFORM_SW_MAJOR_VERSION   (1U)

#define PLATFORM_SW_MINOR_VERSION   (0U)

#define PLATFORM_SW_PATCH_VERSION   (0U)

#define PLATFORM_VENDOR_ID   (1000U)

#define TRUE   (1u)

Typedef Documentation

typedef unsigned char boolean

typedef float float32

typedef double float64

typedef signed short sint16

typedef signed long sint32

typedef signed long long sint64

typedef signed char sint8

typedef unsigned short uint16

typedef unsigned long uint32

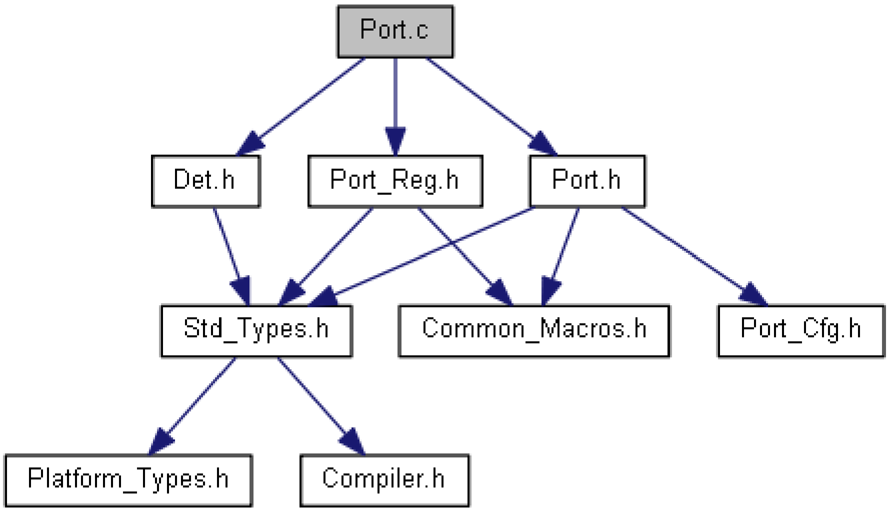
typedef unsigned long long uint64

typedef unsigned char uint8

Port.c File Reference

```
#include "Port.h"
#include "Port_Reg.h"
#include "Det.h"
```

Include dependency graph for Port.c:



Functions

- void **Port_Init** (const **Port_ConfigType** *ConfigPtr)
  Function to initialize the port driver Module.

- void **Port_SetPinDirection** (**Port_PinType** Pin, **Port_PinDirectionType** Direction)
  Function that sets the port pin direction.

- void **Port_RefreshPortDirection** (void)
  Function used to refresh the direction of all configured ports to configured direction.

- void **Port_GetVersionInfo** (**Std_VersionInfoType** *versioninfo)
  Function to get the version information of this module.

- void **Port_SetPinMode** (**Port_PinType** Pin, **Port_PinModeType** Mode)
  Function used to set port pin mode during run time.

Variables

- **STATIC uint8 Port_Status** = **PORT_NOT_INITIALIZED**
- const **STATIC Port_ConfigChannel** * **Port_Channels** = **NULL_PTR**

Function Documentation

void Port_GetVersionInfo (Std_VersionInfoType * *versioninfo*)

Function to get the version information of this module.

Parameters

(in)	VersionInfo - Pointer to where to store the version information of this module.
(inout)	None
(out)	None

Returns

None

Note

This function can be deleted if PORT_VERSION_INFO_API != STD_ON

void Port_Init (const Port_ConfigType * *ConfigPtr*)

Function to initialize the port driver Module.

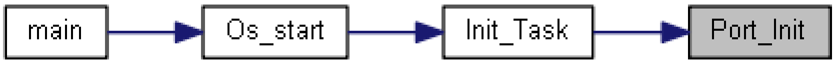
Parameters

(in)	ConfigPtr - Pointer to configuration set
(inout)	None
(out)	None

Returns

none

Here is the caller graph for this function:



void Port_RefreshPortDirection (void)

Function used to refresh the direction of all configured ports to configured direction.

Parameters

(in)	None
(inout)	None
(out)	None

Returns

None

void Port_SetPinDirection (Port_PinType *Pin*, Port_PinDirectionType *Direction*)

Function that sets the port pin direction.

Parameters

(in)	Pin - Port Pin ID Number
(in)	Direction - Port Pin Direction
(inout)	None
(out)	None

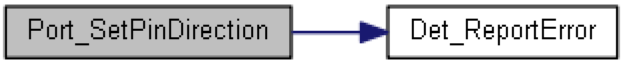
Returns

None

Note

This function can be deleted if PORT_SET_PIN_DIRECTION_API != STD_ON

Here is the call graph for this function:



void Port_SetPinMode (Port_PinType *Pin*, Port_PinModeType *Mode*)

Function used to set port pin mode during run time.

Parameters

(in)	Pin - Port Pin ID Number
(in)	Mode - New Port pin Mode to be set in port pin
(inout)	None
(out)	None

Returns

None

Variable Documentation

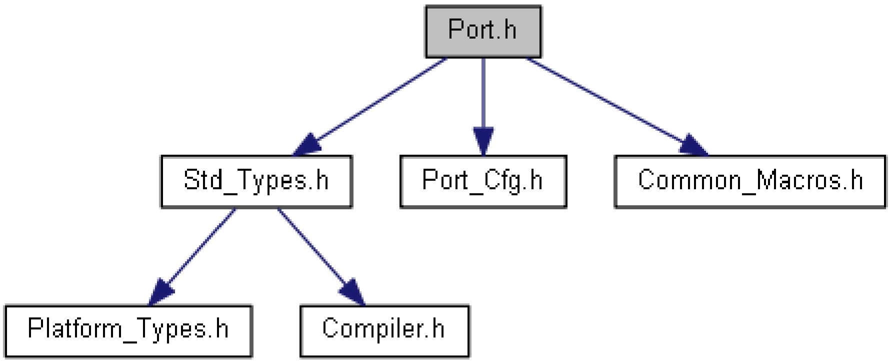
const STATIC Port_ConfigChannel* Port_Channels = NULL_PTR

STATIC uint8 Port_Status = PORT_NOT_INITIALIZED

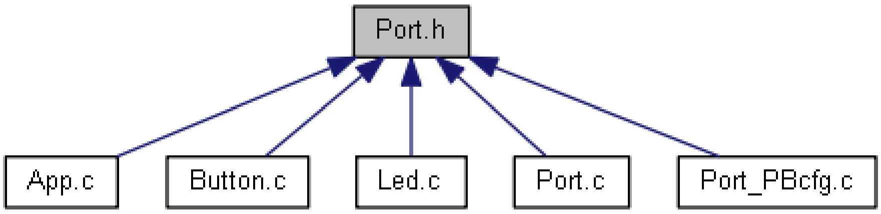
Port.h File Reference

```
#include "Std_Types.h"
#include "Port_Cfg.h"
#include "Common_Macros.h"
```

Include dependency graph for Port.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Port_ConfigChannel**
  Used to set the configurations of a Specific pin in Port Module.
- struct **Port_ConfigType**
  This structure holds an array of structures that holds the configurations of the pins.

Macros

- #define **PORT_VENDOR_ID** (1024U)
- #define **PORT_MODULE_ID** (124U)
- #define **PORT_INSTANCE_ID** (0U)
- #define **PORT_SW_MAJOR_VERSION** (1U)
- #define **PORT_SW_MINOR_VERSION** (1U)
- #define **PORT_SW_PATCH_VERSION** (2U)
- #define **PORT_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **PORT_AR_RELEASE_MINOR_VERSION** (0U)
- #define **PORT_AR_RELEASE_PATCH_VERSION** (3U)
- #define **PORT_INITIALIZED** (1U)
- #define **PORT_NOT_INITIALIZED** (0U)
- #define **PORT_INIT_SID** (**uint8**)0x00
- #define **PORT_SET_PIN_DIRECTION_SID** (**uint8**)0x01
- #define **PORT_REFRESH_PORT_DIRECTION_SID** (**uint8**)0x02
- #define **PORT_GET_VERSION_INFO_SID** (**uint8**)0x03
- #define **PORT_SET_PIN_MODE_SID** (**uint8**)0x04
- #define **PORT_E_PARAM_PIN** (**uint8**)0x0A
- #define **PORT_E_DIRECTION_UNCHANGEABLE** (**uint8**)0x0B
- #define **PORT_E_PARAM_CONFIG** (**uint8**)0x0C

- #define **PORT_E_PARAM_INVALID_MODE** (**uint8**)0x0D
- #define **PORT_E_MODE_UNCHANGEABLE** (**uint8**)0x0E
- #define **PORT_E_UNINIT** (**uint8**)0x0F
- #define **PORT_E_PARAM_POINTER** (**uint8**)0x10
- #define **PORT_E_DIRECTON_UNCHANGEABLE** (**uint8**)0x12
- #define **PIN_DIRECTION_CHANGEABILITY_ON** **STD_ON**
- #define **PIN_DIRECTION_CHANGEABILITY_OFF** **STD_OFF**
- #define **PIN_MODE_CHANGEABILITY_ON** **STD_ON**
- #define **PIN_MODE_CHANGEABILITY_OFF** **STD_OFF**
- #define **INITIAL_VALUE_HIGH** **STD_HIGH**
- #define **INITIAL_VALUE_LOW** **STD_LOW**
- #define **ANALOG_MODE_SELECTED** 15
- #define **SLEW_RATE_ENABLE** **STD_ON**
- #define **SLEW_RATE_DISABLE** **STD_OFF**
- #define **PORT_C_PIN_0_JTAG** (100U)
- #define **PORT_C_PIN_1_JTAG** (101U)
- #define **PORT_C_PIN_2_JTAG** (102U)
- #define **PORT_C_PIN_3_JTAG** (103U)
- #define **PIN_NUM_MAX_CHECK** (40U)
- #define **LOCK_REGISTER_UNLOCK_VALUE** (0x4C4F434B)
- #define **PORT_CONTROL_MASK_VALUE** (0x0000000F)
- #define **PORT_CONTROL_BIT_NUMBERS** (4U)

Typedefs

- typedef **uint8 Port_PinType**
  Type definition for Port_PinType used by PORT APIs Which used to decide which pin number is used.

- typedef **uint8 Port_PinModeType**
  Type definition for Port_PinModeType used by PORT APIs Which used to decide the mode of the Pin.

Enumerations

- enum **Port_PinDirectionType** { **PORT_PIN_IN**, **PORT_PIN_OUT**, **PORT_PIN_DEFAULT** }
  Type definition for Port_PinDirectionType used by PORT APIs.
- enum **Port_InternalResistor** { **OFF**, **PULL_UP**, **PULL_DOWN** }
  Used to decide the mode of internal pull up or to disable it at all.
- enum **Port_PinModeEnum** { **DIO_PIN** = 0, **PORT_A_PIN_0_AF1_U0RX** = 1, **PORT_A_PIN_1_AF1_U0TX** = 1, **PORT_B_PIN_0_AF1_U1RX** = 1, **PORT_B_PIN_1_AF1_U1TX** = 1, **PORT_C_PIN_4_AF1_U4RX** = 1, **PORT_C_PIN_5_AF1_U4TX** = 1, **PORT_C_PIN_6_AF1_U3RX** = 1, **PORT_C_PIN_7_AF1_U3TX** = 1, **PORT_D_PIN_0_AF1_SSI3CLK** = 1, **PORT_D_PIN_1_AF1_SSI3FSS** = 1, **PORT_D_PIN_2_AF1_SSI3RX** = 1, **PORT_D_PIN_3_AF1_SSI3TX** = 1, **PORT_D_PIN_4_AF1_U6RX** = 1, **PORT_D_PIN_5_AF1_U6TX** = 1, **PORT_D_PIN_6_AF1_U2RX** = 1, **PORT_D_PIN_7_AF1_U2TX** = 1, **PORT_E_PIN_0_AF1_U7RX** = 1, **PORT_E_PIN_1_AF1_U7TX** = 1, **PORT_E_PIN_4_AF1_U5RX** = 1, **PORT_E_PIN_5_AF1_U5TX** = 1, **PORT_F_PIN_0_AF1_U1RTS** = 1, **PORT_F_PIN_1_AF1_U1CTS** = 1, **PORT_A_PIN_2_AF2_SSI0CLK** = 2, **PORT_A_PIN_3_AF2_SSI0FSS** = 2, **PORT_A_PIN_4_AF2_SSI0RX** = 2, **PORT_A_PIN_5_AF2_SSI0TX** = 2, **PORT_B_PIN_4_AF2_SSI2CLK** = 2, **PORT_B_PIN_5_AF2_SSI2FSS** = 2, **PORT_B_PIN_6_AF2_SSI2RX** = 2, **PORT_B_PIN_7_AF2_SSI2TX** = 2, **PORT_C_PIN_4_AF2_U1RX** = 2,

PORT_C_PIN_5_AF2_U1TX = 2, **PORT_D_PIN_0_AF2_SSI1CLK** = 2,
PORT_D_PIN_1_AF2_SSI1FSS = 2, **PORT_D_PIN_2_AF2_SSI1RX** = 2,
PORT_D_PIN_3_AF2_SSI1TX = 2, **PORT_F_PIN_0_AF2_SSI1RX** = 2,
PORT_F_PIN_1_AF2_SSI1TX = 2, **PORT_F_PIN_2_AF2_SSI1CLK** = 2,
PORT_F_PIN_3_AF2_SSI1FSS = 2, **PORT_A_PIN_6_AF3_I2C1SCL** = 3,
PORT_A_PIN_7_AF3_I2C1SDA = 3, **PORT_B_PIN_2_AF3_I2C0SCL** = 3,
PORT_B_PIN_3_AF3_I2C0SDA = 3, **PORT_D_PIN_0_AF3_I2C3SCL** = 3,
PORT_D_PIN_1_AF3_I2C3SDA = 3, **PORT_E_PIN_4_AF3_I2C2SCL** = 3,
PORT_E_PIN_5_AF3_I2C2SDA = 3, **PORT_F_PIN_0_AF3_CAN0RX** = 3,
PORT_F_PIN_3_AF3_CAN0TX = 3, **PORT_B_PIN_4_AF4_M0PWM2** = 4,
PORT_B_PIN_5_AF4_M0PWM3 = 4, **PORT_B_PIN_6_AF4_M0PWM0** = 4,
PORT_B_PIN_7_AF4_M0PWM1 = 4, **PORT_C_PIN_4_AF4_M0PWM6** = 4,
PORT_C_PIN_5_AF4_M0PWM7 = 4, **PORT_D_PIN_0_AF4_M0PWM6** = 4,
PORT_D_PIN_1_AF4_M0PWM7 = 4, **PORT_D_PIN_2_AF4_M0FAULT0** = 4,
PORT_D_PIN_6_AF4_M0FAULT0 = 4, **PORT_E_PIN_4_AF4_M0PWM4** = 4,
PORT_E_PIN_5_AF4_M0PWM5 = 4, **PORT_F_PIN_2_AF4_M0FAULT0** = 4,
PORT_A_PIN_6_AF5_M1PWM2 = 5, **PORT_A_PIN_7_AF5_M1PWM3** = 5,
PORT_D_PIN_0_AF5_M1PWM0 = 5, **PORT_D_PIN_1_AF5_M1PWM1** = 5,
PORT_E_PIN_4_AF5_M1PWM2 = 5, **PORT_E_PIN_5_AF5_M1PWM3** = 5,
PORT_F_PIN_0_AF5_M1PWM4 = 5, **PORT_F_PIN_1_AF5_M1PWM5** = 5,
PORT_F_PIN_2_AF5_M1PWM6 = 5, **PORT_F_PIN_3_AF5_M1PWM7** = 5,
PORT_F_PIN_4_AF5_M1FAULT0 = 5, **PORT_C_PIN_4_AF6_IDX1** = 6,
PORT_C_PIN_5_AF6_PHA1 = 6, **PORT_C_PIN_6_AF6_PHB1** = 6,
PORT_D_PIN_3_AF6_IDX0 = 6, **PORT_D_PIN_6_AF6_PHA0** = 6,
PORT_D_PIN_7_AF6_PHB0 = 6, **PORT_F_PIN_0_AF6_PHA0** = 6,
PORT_F_PIN_1_AF6_PHB0 = 6, **PORT_F_PIN_4_AF6_IDX0** = 6,
PORT_B_PIN_0_AF7_T2CCP0 = 7, **PORT_B_PIN_1_AF7_T2CCP1** = 7,
PORT_B_PIN_2_AF7_T3CCP0 = 7, **PORT_B_PIN_3_AF7_T3CCP1** = 7,
PORT_B_PIN_4_AF7_T1CCP0 = 7, **PORT_B_PIN_5_AF7_T1CCP1** = 7,
PORT_B_PIN_6_AF7_T0CCP0 = 7, **PORT_B_PIN_7_AF7_T0CCP1** = 7,
PORT_C_PIN_4_AF7_WT0CCP0 = 7, **PORT_C_PIN_5_AF7_WT0CCP1** = 7,
PORT_C_PIN_6_AF7_WT1CCP0 = 7, **PORT_C_PIN_7_AF7_WT1CCP1** = 7,
PORT_D_PIN_0_AF7_WT2CCP0 = 7, **PORT_D_PIN_1_AF7_WT2CCP1** = 7,
PORT_D_PIN_2_AF7_WT3CCP0 = 7, **PORT_D_PIN_3_AF7_WT3CCP1** = 7,
PORT_D_PIN_4_AF7_WT4CCP0 = 7, **PORT_D_PIN_5_AF7_WT4CCP1** = 7,
PORT_D_PIN_6_AF7_WT5CCP0 = 7, **PORT_D_PIN_7_AF7_WT5CCP1** = 7,
PORT_F_PIN_0_AF7_T0CCP0 = 7, **PORT_F_PIN_1_AF7_T0CCP1** = 7,
PORT_F_PIN_2_AF7_T1CCP0 = 7, **PORT_F_PIN_3_AF7_T1CCP1** = 7,
PORT_F_PIN_4_AF7_T2CCP0 = 7, **PORT_A_PIN_0_AF8_CAN1RX** = 8,
PORT_A_PIN_1_AF8_CAN1TX = 8, **PORT_B_PIN_4_AF8_CAN0RX** = 8,
PORT_B_PIN_5_AF8_CAN0TX = 8, **PORT_C_PIN_4_AF8_U1RTS** = 8,
PORT_C_PIN_5_AF8_U1CTS = 8, **PORT_C_PIN_6_AF8_USB0EPEN** = 8,
PORT_C_PIN_7_AF8_USB0PFLT = 8, **PORT_D_PIN_2_AF8_USB0EPEN** = 8,
PORT_D_PIN_3_AF8_USB0PFLT = 8, **PORT_D_PIN_7_AF8_NMI** = 8,
PORT_E_PIN_4_AF8_CAN0RX = 8, **PORT_E_PIN_5_AF8_CAN0TX** = 8,
PORT_F_PIN_0_AF8_NMI = 8, **PORT_F_PIN_4_AF8_USB0EPEN** = 8,
PORT_F_PIN_0_AF9_C0O = 9, **PORT_F_PIN_1_AF9_C01** = 9,
PORT_F_PIN_1_AF14_TRD1 = 14, **PORT_F_PIN_2_AF14_TRD0** = 14,
PORT_F_PIN_3_AF14_TRCLK = 14 }

This enum has all the alternative function values of the TM4C123GH6PM to be used with Port_PinModeType data type.

- enum **Port_PinNumEnum** { **PORT_A_PIN_0** = 0, **PORT_A_PIN_1** = 1, **PORT_A_PIN_2** = 2,
PORT_A_PIN_3 = 3, **PORT_A_PIN_4** = 4, **PORT_A_PIN_5** = 5, **PORT_A_PIN_6** = 6,
PORT_A_PIN_7 = 7, **PORT_B_PIN_0** = 0, **PORT_B_PIN_1** = 1, **PORT_B_PIN_2** = 2,
PORT_B_PIN_3 = 3, **PORT_B_PIN_4** = 4, **PORT_B_PIN_5** = 5, **PORT_B_PIN_6** = 6,
PORT_B_PIN_7 = 7, **PORT_C_PIN_4** = 4, **PORT_C_PIN_5** = 5, **PORT_C_PIN_6** = 6,
PORT_C_PIN_7 = 7, **PORT_D_PIN_0** = 0, **PORT_D_PIN_1** = 1, **PORT_D_PIN_2** = 2,
PORT_D_PIN_3 = 3, **PORT_D_PIN_4** = 4, **PORT_D_PIN_5** = 5, **PORT_D_PIN_6** = 6,
PORT_D_PIN_7 = 7, **PORT_E_PIN_0** = 0, **PORT_E_PIN_1** = 1, **PORT_E_PIN_2** = 2,

PORT_E_PIN_3 = 3, **PORT_E_PIN_4** = 4, **PORT_E_PIN_5** = 5, **PORT_F_PIN_0** = 0, **PORT_F_PIN_1** = 1, **PORT_F_PIN_2** = 2, **PORT_F_PIN_3** = 3, **PORT_F_PIN_4** = 4 }
This enum used to numbering the pins on the TM4C123GH6PM.

- enum **Port_PortNumEnum** { **PORTA**, **PORTB**, **PORTC**, **PORTD**, **PORTE**, **PORTF** }
  Used to decide the Port Number.

Functions

- void **Port_Init** (const **Port_ConfigType** *ConfigPtr)
  Function to initialize the port driver Module.

- void **Port_SetPinDirection** (**Port_PinType** Pin, **Port_PinDirectionType** Direction)
  Function that sets the port pin direction.

- void **Port_RefreshPortDirection** (void)
  Function used to refresh the direction of all configured ports to configured direction.

- void **Port_GetVersionInfo** (**Std_VersionInfoType** *versioninfo)
  Function to get the version information of this module.

- void **Port_SetPinMode** (**Port_PinType** Pin, **Port_PinModeType** Mode)
  Function used to set port pin mode during run time.

Variables

- const **Port_ConfigType Port_Configuration**

Macro Definition Documentation

#define ANALOG_MODE_SELECTED   15

#define INITIAL_VALUE_HIGH   STD_HIGH

#define INITIAL_VALUE_LOW   STD_LOW

#define LOCK_REGISTER_UNLOCK_VALUE   (0x4C4F434B)

#define PIN_DIRECTION_CHANGEABILITY_OFF   STD_OFF

#define PIN_DIRECTION_CHANGEABILITY_ON   STD_ON

#define PIN_MODE_CHANGEABILITY_OFF   STD_OFF

#define PIN_MODE_CHANGEABILITY_ON   STD_ON

#define PIN_NUM_MAX_CHECK   (40U)

#define PORT_AR_RELEASE_MAJOR_VERSION   (4U)

#define PORT_AR_RELEASE_MINOR_VERSION   (0U)

#define PORT_AR_RELEASE_PATCH_VERSION   (3U)

#define PORT_C_PIN_0_JTAG   (100U)

#define PORT_C_PIN_1_JTAG   (101U)

#define PORT_C_PIN_2_JTAG   (102U)

#define PORT_C_PIN_3_JTAG   (103U)

#define PORT_CONTROL_BIT_NUMBERS   (4U)

#define PORT_CONTROL_MASK_VALUE   (0x0000000F)

#define PORT_E_DIRECTION_UNCHANGEABLE   (uint8)0x0B

#define PORT_E_DIRECTON_UNCHANGEABLE   (uint8)0x12

#define PORT_E_MODE_UNCHANGEABLE   (uint8)0x0E

#define PORT_E_PARAM_CONFIG   (uint8)0x0C

#define PORT_E_PARAM_INVALID_MODE   (uint8)0x0D

#define PORT_E_PARAM_PIN   (uint8)0x0A

#define PORT_E_PARAM_POINTER   (uint8)0x10

#define PORT_E_UNINIT (uint8)0x0F

#define PORT_GET_VERSION_INFO_SID (uint8)0x03

#define PORT_INIT_SID (uint8)0x00

#define PORT_INITIALIZED (1U)

#define PORT_INSTANCE_ID (0U)

#define PORT_MODULE_ID (124U)

#define PORT_NOT_INITIALIZED (0U)

#define PORT_REFRESH_PORT_DIRECTION_SID (uint8)0x02

#define PORT_SET_PIN_DIRECTION_SID (uint8)0x01

#define PORT_SET_PIN_MODE_SID (uint8)0x04

#define PORT_SW_MAJOR_VERSION (1U)

#define PORT_SW_MINOR_VERSION (1U)

#define PORT_SW_PATCH_VERSION (2U)

#define PORT_VENDOR_ID (1024U)

#define SLEW_RATE_DISABLE STD_OFF

#define SLEW_RATE_ENABLE STD_ON

Typedef Documentation

Port_PinModeType

Type definition for Port_PinModeType used by PORT APIs Which used to decide the mode of the Pin.

Note

You can find these values **Port_PinModeEnum**

Port_PinType

Type definition for Port_PinType used by PORT APIs Which used to decide which pin number is used.

Note

You can find these values @Port_PinNumEnum

Enumeration Type Documentation

enum Port_InternalResistor

Used to decide the mode of internal pull up or to disable it at all.

Enumerator:

OFF	OFF.
PULL_UP	PULL_UP.
PULL_DOWN	PULL_DOWN.

enum Port_PinDirectionType

Type definition for Port_PinDirectionType used by PORT APIs.

Enumerator:

PORT_PIN_IN	PORT_PIN_IN.
PORT_PIN_OUT	PORT_PIN_OUT.
PORT_PIN_DEFAULT	PORT_PIN_DEFAULT.

enum Port_PinModeEnum

This enum has all the alternative function values of the TM4C123GH6PM to be used with Port_PinModeType data type.

Enumerator:

DIO_PIN	DIO_PIN.
PORT_A_PIN_0_AF1_U0RX	PORT_A_PIN_0_AF1_U0RX.
PORT_A_PIN_1_AF1_U0TX	PORT_A_PIN_1_AF1_U0TX.
PORT_B_PIN_0_AF1_U1RX	PORT_B_PIN_0_AF1_U1RX.
PORT_B_PIN_1_AF1_U1TX	PORT_B_PIN_1_AF1_U1TX.
PORT_C_PIN_4_AF1_U4RX	PORT_C_PIN_4_AF1_U4RX.
PORT_C_PIN_5_	PORT_C_PIN_5_AF1_U4TX.

AF1_U4TX	
PORT_C_PIN_6_AF1_U3RX	PORT_C_PIN_6_AF1_U3RX.
PORT_C_PIN_7_AF1_U3TX	PORT_C_PIN_7_AF1_U3TX.
PORT_D_PIN_0_AF1_SSI3CLK	PORT_D_PIN_0_AF1_SSI3CLK.
PORT_D_PIN_1_AF1_SSI3FSS	PORT_D_PIN_1_AF1_SSI3FSS.
PORT_D_PIN_2_AF1_SSI3RX	PORT_D_PIN_2_AF1_SSI3RX.
PORT_D_PIN_3_AF1_SSI3TX	PORT_D_PIN_3_AF1_SSI3TX.
PORT_D_PIN_4_AF1_U6RX	PORT_D_PIN_4_AF1_U6RX.
PORT_D_PIN_5_AF1_U6TX	PORT_D_PIN_5_AF1_U6TX.
PORT_D_PIN_6_AF1_U2RX	PORT_D_PIN_6_AF1_U2RX.
PORT_D_PIN_7_AF1_U2TX	PORT_D_PIN_7_AF1_U2TX.
PORT_E_PIN_0_AF1_U7RX	PORT_E_PIN_0_AF1_U7RX.
PORT_E_PIN_1_AF1_U7TX	PORT_E_PIN_1_AF1_U7TX.
PORT_E_PIN_4_AF1_U5RX	PORT_E_PIN_4_AF1_U5RX.
PORT_E_PIN_5_AF1_U5TX	PORT_E_PIN_5_AF1_U5TX.
PORT_F_PIN_0_AF1_U1RTS	PORT_F_PIN_0_AF1_U1RTS.
PORT_F_PIN_1_AF1_U1CTS	PORT_F_PIN_1_AF1_U1CTS.
PORT_A_PIN_2_AF2_SSI0CLK	PORT_A_PIN_2_AF2_SSI0CLK.
PORT_A_PIN_3_AF2_SSI0FSS	PORT_A_PIN_3_AF2_SSI0FSS.
PORT_A_PIN_4_AF2_SSI0RX	PORT_A_PIN_4_AF2_SSI0RX.
PORT_A_PIN_5_AF2_SSI0TX	PORT_A_PIN_5_AF2_SSI0TX.
PORT_B_PIN_4_AF2_SSI2CLK	PORT_B_PIN_4_AF2_SSI2CLK.
PORT_B_PIN_5_AF2_SSI2FSS	PORT_B_PIN_5_AF2_SSI2FSS.
PORT_B_PIN_6_AF2_SSI2RX	PORT_B_PIN_6_AF2_SSI2RX.

PORT_B_PIN_7_AF2_SSI2TX	PORT_B_PIN_7_AF2_SSI2TX.
PORT_C_PIN_4_AF2_U1RX	PORT_C_PIN_4_AF2_U1RX.
PORT_C_PIN_5_AF2_U1TX	PORT_C_PIN_5_AF2_U1TX.
PORT_D_PIN_0_AF2_SSI1CLK	PORT_D_PIN_0_AF2_SSI1CLK.
PORT_D_PIN_1_AF2_SSI1FSS	PORT_D_PIN_1_AF2_SSI1FSS.
PORT_D_PIN_2_AF2_SSI1RX	PORT_D_PIN_2_AF2_SSI1RX.
PORT_D_PIN_3_AF2_SSI1TX	PORT_D_PIN_3_AF2_SSI1TX.
PORT_F_PIN_0_AF2_SSI1RX	PORT_F_PIN_0_AF2_SSI1RX.
PORT_F_PIN_1_AF2_SSI1TX	PORT_F_PIN_1_AF2_SSI1TX.
PORT_F_PIN_2_AF2_SSI1CLK	PORT_F_PIN_2_AF2_SSI1CLK.
PORT_F_PIN_3_AF2_SSI1FSS	PORT_F_PIN_3_AF2_SSI1FSS.
PORT_A_PIN_6_AF3_I2C1SCL	PORT_A_PIN_6_AF3_I2C1SCL.
PORT_A_PIN_7_AF3_I2C1SDA	PORT_A_PIN_7_AF3_I2C1SDA.
PORT_B_PIN_2_AF3_I2C0SCL	PORT_B_PIN_2_AF3_I2C0SCL.
PORT_B_PIN_3_AF3_I2C0SDA	PORT_B_PIN_3_AF3_I2C0SDA.
PORT_D_PIN_0_AF3_I2C3SCL	PORT_D_PIN_0_AF3_I2C3SCL.
PORT_D_PIN_1_AF3_I2C3SDA	PORT_D_PIN_1_AF3_I2C3SDA.
PORT_E_PIN_4_AF3_I2C2SCL	PORT_E_PIN_4_AF3_I2C2SCL.
PORT_E_PIN_5_AF3_I2C2SDA	PORT_E_PIN_5_AF3_I2C2SDA.
PORT_F_PIN_0_AF3_CAN0RX	PORT_F_PIN_0_AF3_CAN0RX.
PORT_F_PIN_3_AF3_CAN0TX	PORT_F_PIN_3_AF3_CAN0TX.
PORT_B_PIN_4_AF4_M0PWM2	PORT_B_PIN_4_AF4_M0PWM2.
PORT_B_PIN_5_AF4_M0PWM3	PORT_B_PIN_5_AF4_M0PWM3.
PORT_B_PIN_6_AF4_M0PWM0	PORT_B_PIN_6_AF4_M0PWM0.

PORT_B_PIN_7_AF4_M0PWM1	PORT_B_PIN_7_AF4_M0PWM1.
PORT_C_PIN_4_AF4_M0PWM6	PORT_C_PIN_4_AF4_M0PWM6.
PORT_C_PIN_5_AF4_M0PWM7	PORT_C_PIN_5_AF4_M0PWM7.
PORT_D_PIN_0_AF4_M0PWM6	PORT_D_PIN_0_AF4_M0PWM6.
PORT_D_PIN_1_AF4_M0PWM7	PORT_D_PIN_1_AF4_M0PWM7.
PORT_D_PIN_2_AF4_M0FAULT0	PORT_D_PIN_2_AF4_M0FAULT0.
PORT_D_PIN_6_AF4_M0FAULT0	PORT_D_PIN_6_AF4_M0FAULT0.
PORT_E_PIN_4_AF4_M0PWM4	PORT_E_PIN_4_AF4_M0PWM4.
PORT_E_PIN_5_AF4_M0PWM5	PORT_E_PIN_5_AF4_M0PWM5.
PORT_F_PIN_2_AF4_M0FAULT0	PORT_F_PIN_2_AF4_M0FAULT0.
PORT_A_PIN_6_AF5_M1PWM2	PORT_A_PIN_6_AF5_M1PWM2.
PORT_A_PIN_7_AF5_M1PWM3	PORT_A_PIN_7_AF5_M1PWM3.
PORT_D_PIN_0_AF5_M1PWM0	PORT_D_PIN_0_AF5_M1PWM0.
PORT_D_PIN_1_AF5_M1PWM1	PORT_D_PIN_1_AF5_M1PWM1.
PORT_E_PIN_4_AF5_M1PWM2	PORT_E_PIN_4_AF5_M1PWM2.
PORT_E_PIN_5_AF5_M1PWM3	PORT_E_PIN_5_AF5_M1PWM3.
PORT_F_PIN_0_AF5_M1PWM4	PORT_F_PIN_0_AF5_M1PWM4.
PORT_F_PIN_1_AF5_M1PWM5	PORT_F_PIN_1_AF5_M1PWM5.
PORT_F_PIN_2_AF5_M1PWM6	PORT_F_PIN_2_AF5_M1PWM6.
PORT_F_PIN_3_AF5_M1PWM7	PORT_F_PIN_3_AF5_M1PWM7.
PORT_F_PIN_4_AF5_M1FAULT0	PORT_F_PIN_4_AF5_M1FAULT0.
PORT_C_PIN_4_AF6_IDX1	PORT_C_PIN_4_AF6_IDX1.
PORT_C_PIN_5_AF6_PHA1	PORT_C_PIN_5_AF6_PHA1.

PORT_C_PIN_6_AF6_PHB1	PORT_C_PIN_6_AF6_PHB1.
PORT_D_PIN_3_AF6_IDX0	PORT_D_PIN_3_AF6_IDX0.
PORT_D_PIN_6_AF6_PHA0	PORT_D_PIN_6_AF6_PHA0.
PORT_D_PIN_7_AF6_PHB0	PORT_D_PIN_7_AF6_PHB0.
PORT_F_PIN_0_AF6_PHA0	PORT_F_PIN_0_AF6_PHA0.
PORT_F_PIN_1_AF6_PHB0	PORT_F_PIN_1_AF6_PHB0.
PORT_F_PIN_4_AF6_IDX0	PORT_F_PIN_4_AF6_IDX0.
PORT_B_PIN_0_AF7_T2CCP0	PORT_B_PIN_0_AF7_T2CCP0.
PORT_B_PIN_1_AF7_T2CCP1	PORT_B_PIN_1_AF7_T2CCP1.
PORT_B_PIN_2_AF7_T3CCP0	PORT_B_PIN_2_AF7_T3CCP0.
PORT_B_PIN_3_AF7_T3CCP1	PORT_B_PIN_3_AF7_T3CCP1.
PORT_B_PIN_4_AF7_T1CCP0	PORT_B_PIN_4_AF7_T1CCP0.
PORT_B_PIN_5_AF7_T1CCP1	PORT_B_PIN_5_AF7_T1CCP1.
PORT_B_PIN_6_AF7_T0CCP0	PORT_B_PIN_6_AF7_T0CCP0.
PORT_B_PIN_7_AF7_T0CCP1	PORT_B_PIN_7_AF7_T0CCP1.
PORT_C_PIN_4_AF7_WT0CCP0	PORT_C_PIN_4_AF7_WT0CCP0.
PORT_C_PIN_5_AF7_WT0CCP1	PORT_C_PIN_5_AF7_WT0CCP1.
PORT_C_PIN_6_AF7_WT1CCP0	PORT_C_PIN_6_AF7_WT1CCP0.
PORT_C_PIN_7_AF7_WT1CCP1	PORT_C_PIN_7_AF7_WT1CCP1.
PORT_D_PIN_0_AF7_WT2CCP0	PORT_D_PIN_0_AF7_WT2CCP0.
PORT_D_PIN_1_AF7_WT2CCP1	PORT_D_PIN_1_AF7_WT2CCP1.
PORT_D_PIN_2_AF7_WT3CCP0	PORT_D_PIN_2_AF7_WT3CCP0.
PORT_D_PIN_3_AF7_WT3CCP1	PORT_D_PIN_3_AF7_WT3CCP1.
PORT_D_PIN_4_AF7_WT4CCP0	PORT_D_PIN_4_AF7_WT4CCP0.

PORT_D_PIN_5_AF7_WT4CCP1	PORT_D_PIN_5_AF7_WT4CCP1.
PORT_D_PIN_6_AF7_WT5CCP0	PORT_D_PIN_6_AF7_WT5CCP0.
PORT_D_PIN_7_AF7_WT5CCP1	PORT_D_PIN_7_AF7_WT5CCP1.
PORT_F_PIN_0_AF7_T0CCP0	PORT_F_PIN_0_AF7_T0CCP0.
PORT_F_PIN_1_AF7_T0CCP1	PORT_F_PIN_1_AF7_T0CCP1.
PORT_F_PIN_2_AF7_T1CCP0	PORT_F_PIN_2_AF7_T1CCP0.
PORT_F_PIN_3_AF7_T1CCP1	PORT_F_PIN_3_AF7_T1CCP1.
PORT_F_PIN_4_AF7_T2CCP0	PORT_F_PIN_4_AF7_T2CCP0.
PORT_A_PIN_0_AF8_CAN1RX	PORT_A_PIN_0_AF8_CAN1RX.
PORT_A_PIN_1_AF8_CAN1TX	PORT_A_PIN_1_AF8_CAN1TX.
PORT_B_PIN_4_AF8_CAN0RX	PORT_B_PIN_4_AF8_CAN0RX.
PORT_B_PIN_5_AF8_CAN0TX	PORT_B_PIN_5_AF8_CAN0TX.
PORT_C_PIN_4_AF8_U1RTS	PORT_C_PIN_4_AF8_U1RTS.
PORT_C_PIN_5_AF8_U1CTS	PORT_C_PIN_5_AF8_U1CTS.
PORT_C_PIN_6_AF8_USB0EPEN	PORT_C_PIN_6_AF8_USB0EPEN.
PORT_C_PIN_7_AF8_USB0PFLT	PORT_C_PIN_7_AF8_USB0PFLT.
PORT_D_PIN_2_AF8_USB0EPEN	PORT_D_PIN_2_AF8_USB0EPEN.
PORT_D_PIN_3_AF8_USB0PFLT	PORT_D_PIN_3_AF8_USB0PFLT.
PORT_D_PIN_7_AF8_NMI	PORT_D_PIN_7_AF8_NMI.
PORT_E_PIN_4_AF8_CAN0RX	PORT_E_PIN_4_AF8_CAN0RX.
PORT_E_PIN_5_AF8_CAN0TX	PORT_E_PIN_5_AF8_CAN0TX.
PORT_F_PIN_0_AF8_NMI	PORT_F_PIN_0_AF8_NMI.
PORT_F_PIN_4_AF8_USB0EPEN	PORT_F_PIN_4_AF8_USB0EPEN.

PORT_F_PIN_0_AF9_C0O	PORT_F_PIN_0_AF9_C0O.
PORT_F_PIN_1_AF9_C01	PORT_F_PIN_1_AF9_C01.
PORT_F_PIN_1_AF14_TRD1	PORT_F_PIN_1_AF14_TRD1.
PORT_F_PIN_2_AF14_TRD0	PORT_F_PIN_2_AF14_TRD0.
PORT_F_PIN_3_AF14_TRCLK	PORT_F_PIN_3_AF14_TRCLK.

enum Port_PinNumEnum

This enum used to numbering the pins on the TM4C123GH6PM.

Enumerator:

PORT_A_PIN_0	PORT_A_PIN_0.
PORT_A_PIN_1	PORT_A_PIN_1.
PORT_A_PIN_2	PORT_A_PIN_2.
PORT_A_PIN_3	PORT_A_PIN_3.
PORT_A_PIN_4	PORT_A_PIN_4.
PORT_A_PIN_5	PORT_A_PIN_5.
PORT_A_PIN_6	PORT_A_PIN_6.
PORT_A_PIN_7	PORT_A_PIN_7.
PORT_B_PIN_0	PORT_B_PIN_0.
PORT_B_PIN_1	PORT_B_PIN_1.
PORT_B_PIN_2	PORT_B_PIN_2.
PORT_B_PIN_3	PORT_B_PIN_3.
PORT_B_PIN_4	PORT_B_PIN_4.
PORT_B_PIN_5	PORT_B_PIN_5.
PORT_B_PIN_6	PORT_B_PIN_6.
PORT_B_PIN_7	PORT_B_PIN_7.

PORT_C_PIN_4	PORT_C_PIN_4.
PORT_C_PIN_5	PORT_C_PIN_5.
PORT_C_PIN_6	PORT_C_PIN_6.
PORT_C_PIN_7	PORT_C_PIN_7.
PORT_D_PIN_0	PORT_D_PIN_0.
PORT_D_PIN_1	PORT_D_PIN_1.
PORT_D_PIN_2	PORT_D_PIN_2.
PORT_D_PIN_3	PORT_D_PIN_3.
PORT_D_PIN_4	PORT_D_PIN_4.
PORT_D_PIN_5	PORT_D_PIN_5.
PORT_D_PIN_6	PORT_D_PIN_6.
PORT_D_PIN_7	PORT_D_PIN_7.
PORT_E_PIN_0	PORT_E_PIN_0.
PORT_E_PIN_1	PORT_E_PIN_1.
PORT_E_PIN_2	PORT_E_PIN_2.
PORT_E_PIN_3	PORT_E_PIN_3.
PORT_E_PIN_4	PORT_E_PIN_4.
PORT_E_PIN_5	PORT_E_PIN_5.
PORT_F_PIN_0	PORT_F_PIN_0.
PORT_F_PIN_1	PORT_F_PIN_1.
PORT_F_PIN_2	PORT_F_PIN_2.
PORT_F_PIN_3	PORT_F_PIN_3.
PORT_F_PIN_4	PORT_F_PIN_4.

enum Port_PortNumEnum

Used to decide the Port Number.

Enumerator:

PORTA	PORTA.
PORTB	PORTB.
PORTC	PORTC.
PORTD	PORTD.
PORTE	PORTE.
PORTF	PORTF.

Function Documentation

void Port_GetVersionInfo (Std_VersionInfoType * *versioninfo*)

Function to get the version information of this module.

Parameters

(in)	VersionInfo - Pointer to where to store the version information of this module.
(inout)	None
(out)	None

Returns

None

Note

This function can be deleted if PORT_VERSION_INFO_API != STD_ON

void Port_Init (const Port_ConfigType * *ConfigPtr*)

Function to initialize the port driver Module.

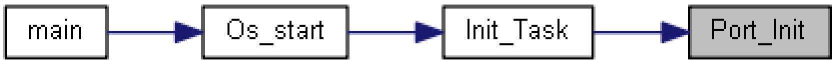
Parameters

(in)	ConfigPtr - Pointer to configuration set
(inout)	None
(out)	None

Returns

none

Here is the caller graph for this function:



void Port_RefreshPortDirection (void)

Function used to refresh the direction of all configured ports to configured direction.

Parameters

(in)	None
(inout)	None
(out)	None

Returns

None

void Port_SetPinDirection (Port_PinType *Pin*, Port_PinDirectionType *Direction*)

Function that sets the port pin direction.

Parameters

(in)	Pin - Port Pin ID Number
(in)	Direction - Port Pin Direction
(inout)	None
(out)	None

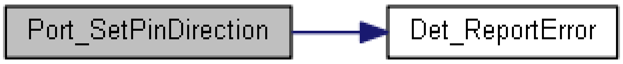
Returns

None

Note

This function can be deleted if PORT_SET_PIN_DIRECTION_API != STD_ON

Here is the call graph for this function:



void Port_SetPinMode (Port_PinType *Pin*, Port_PinModeType *Mode*)

Function used to set port pin mode during run time.

Parameters

(in)	Pin - Port Pin ID Number
(in)	Mode - New Port pin Mode to be set in port pin
(inout)	None
(out)	None

Returns

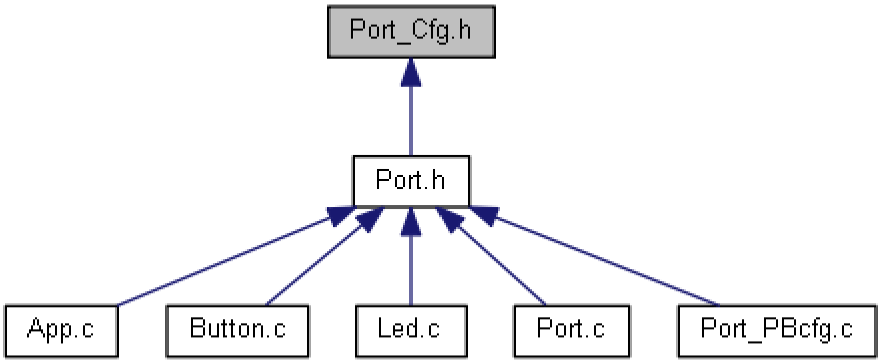
None

Variable Documentation

const Port_ConfigType Port_Configuration

Port_Cfg.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define **PORT_CFG_SW_MAJOR_VERSION**   (1U)
- #define **PORT_CFG_SW_MINOR_VERSION**   (1U)
- #define **PORT_CFG_SW_PATCH_VERSION**   (2U)
- #define **PORT_CFG_AR_RELEASE_MAJOR_VERSION**   (4U)
- #define **PORT_CFG_AR_RELEASE_MINOR_VERSION**   (0U)
- #define **PORT_CFG_AR_RELEASE_PATCH_VERSION**   (3U)
- #define **PORT_DEV_ERROR_DETECT**   (**STD_ON**)
- #define **PORT_SET_PIN_DIRECTION_API**   (**STD_ON**)
- #define **PORT_VERSION_INFO_API**   (**STD_ON**)
- #define **PORT_CONFIGURED_CHANNLES**   (39U)
- #define **PortConf_LED1_PORT_NUM**   (**Port_PortNumEnum**)**PORTF**
- #define **PortConf_SW1_PORT_NUM**   (**Port_PortNumEnum**)**PORTF**
- #define **PortConf_LED1_PIN_NUM**   (**Port_PinType**)**PORT_F_PIN_2** /* Pin 3 in **PORTF** */
- #define **PortConf_SW1_PIN_NUM**   (**Port_PinType**)**PORT_F_PIN_4** /* Pin 4 in **PORTF** */

Macro Definition Documentation

#define PORT_CFG_AR_RELEASE_MAJOR_VERSION   (4U)

#define PORT_CFG_AR_RELEASE_MINOR_VERSION   (0U)

#define PORT_CFG_AR_RELEASE_PATCH_VERSION   (3U)

#define PORT_CFG_SW_MAJOR_VERSION   (1U)

#define PORT_CFG_SW_MINOR_VERSION   (1U)

#define PORT_CFG_SW_PATCH_VERSION   (2U)

#define PORT_CONFIGURED_CHANNLES   (39U)

#define PORT_DEV_ERROR_DETECT   (STD_ON)

#define PORT_SET_PIN_DIRECTION_API   (STD_ON)

#define PORT_VERSION_INFO_API   (STD_ON)

#define PortConf_LED1_PIN_NUM   (Port_PinType)PORT_F_PIN_2 /* Pin 3 in PORTF */

#define PortConf_LED1_PORT_NUM   (Port_PortNumEnum)PORTF

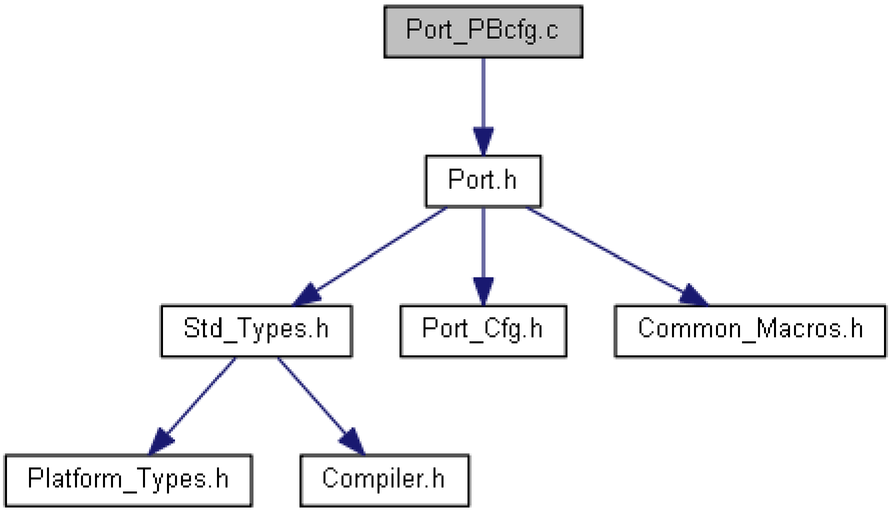
#define PortConf_SW1_PIN_NUM   (Port_PinType)PORT_F_PIN_4 /* Pin 4 in PORTF */

#define PortConf_SW1_PORT_NUM   (Port_PortNumEnum)PORTF

Port_PBcfg.c File Reference

`#include "Port.h"`

Include dependency graph for Port_PBcfg.c:



Macros

- #define **PORT_PBCFG_SW_MAJOR_VERSION** (1U)
- #define **PORT_PBCFG_SW_MINOR_VERSION** (1U)
- #define **PORT_PBCFG_SW_PATCH_VERSION** (2U)
- #define **PORT_PBCFG_AR_RELEASE_MAJOR_VERSION** (4U)
- #define **PORT_PBCFG_AR_RELEASE_MINOR_VERSION** (0U)
- #define **PORT_PBCFG_AR_RELEASE_PATCH_VERSION** (3U)

Variables

- const **Port_ConfigType Port_Configuration**

Macro Definition Documentation

#define PORT_PBCFG_AR_RELEASE_MAJOR_VERSION (4U)

#define PORT_PBCFG_AR_RELEASE_MINOR_VERSION (0U)

#define PORT_PBCFG_AR_RELEASE_PATCH_VERSION (3U)

#define PORT_PBCFG_SW_MAJOR_VERSION (1U)

#define PORT_PBCFG_SW_MINOR_VERSION (1U)

#define PORT_PBCFG_SW_PATCH_VERSION (2U)

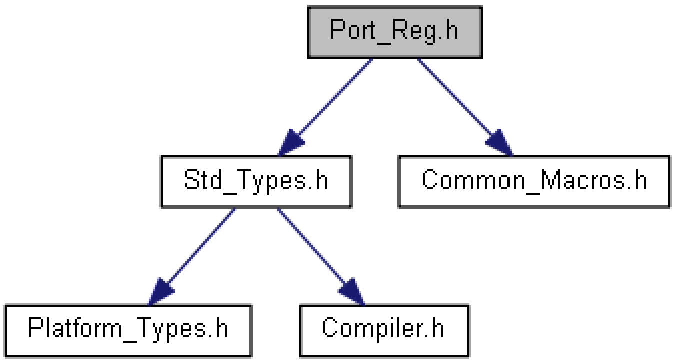
Variable Documentation

const Port_ConfigType Port_Configuration

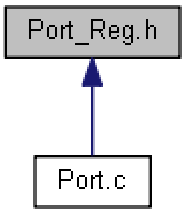
Port_Reg.h File Reference

```
#include "Std_Types.h"
#include "Common_Macros.h"
```

Include dependency graph for Port_Reg.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **GPIO_PORTA_BASE_ADDRESS** 0x40004000 /* GPIOA(APB) Base Address */
- #define **GPIO_PORTB_BASE_ADDRESS** 0x40005000 /* GPIOB(APB) Base Address */
- #define **GPIO_PORTC_BASE_ADDRESS** 0x40006000 /* GPIOC(APB) Base Address */
- #define **GPIO_PORTD_BASE_ADDRESS** 0x40007000 /* GPIOD(APB) Base Address */
- #define **GPIO_PORTE_BASE_ADDRESS** 0x40024000 /* GPIOE(APB) Base Address */
- #define **GPIO_PORTF_BASE_ADDRESS** 0x40025000 /* GPIOF(APB) Base Address */
- #define **PORT_DATA_REG_OFFSET** 0x3FC /* Data Register Offset */
- #define **PORT_DIR_REG_OFFSET** 0x400 /* Direction Register Offset */
- #define **PORT_ALT_FUNC_REG_OFFSET** 0x420 /* Alternative Function Register Offset */
- #define **PORT_PULL_UP_REG_OFFSET** 0x510 /* Pull Up Register Offset */
- #define **PORT_PULL_DOWN_REG_OFFSET** 0x514 /* Pull Down Register Offset */
- #define **PORT_SLEW_RATE_CONTROL_OFFSET** 0x518 /* Slew Rate control register offset */
- #define **PORT_DIGITAL_ENABLE_REG_OFFSET** 0x51C /* Digital Enable Register Offset */

- #define **PORT_LOCK_REG_OFFSET** 0x520 /* Lock Register Offset */
- #define **PORT_COMMIT_REG_OFFSET** 0x524 /* Commit Register Offset */
- #define **PORT_ANALOG_MODE_SEL_REG_OFFSET** 0x528 /* Analog mode selection Register Offset */
- #define **PORT_CTL_REG_OFFSET** 0x52C /* Control Register Offset */
- #define **SYSCTL_REGCGC2_REG** (*((volatile **uint32** *)0x400FE108))

Macro Definition Documentation

#define GPIO_PORTA_BASE_ADDRESS 0x40004000 /* GPIOA(APB) Base Address */

#define GPIO_PORTB_BASE_ADDRESS 0x40005000 /* GPIOB(APB) Base Address */

#define GPIO_PORTC_BASE_ADDRESS 0x40006000 /* GPIOC(APB) Base Address */

#define GPIO_PORTD_BASE_ADDRESS 0x40007000 /* GPIOD(APB) Base Address */

#define GPIO_PORTE_BASE_ADDRESS 0x40024000 /* GPIOE(APB) Base Address */

#define GPIO_PORTF_BASE_ADDRESS 0x40025000 /* GPIOF(APB) Base Address */

#define PORT_ALT_FUNC_REG_OFFSET 0x420 /* Alternative Function Register Offset */

#define PORT_ANALOG_MODE_SEL_REG_OFFSET 0x528 /* Analog mode selection Register Offset */

#define PORT_COMMIT_REG_OFFSET 0x524 /* Commit Register Offset */

#define PORT_CTL_REG_OFFSET 0x52C /* Control Register Offset */

#define PORT_DATA_REG_OFFSET 0x3FC /* Data Register Offset */

#define PORT_DIGITAL_ENABLE_REG_OFFSET 0x51C /* Digital Enable Register Offset */

#define PORT_DIR_REG_OFFSET 0x400 /* Direction Register Offset */

#define PORT_LOCK_REG_OFFSET 0x520 /* Lock Register Offset */

#define PORT_PULL_DOWN_REG_OFFSET 0x514 /* Pull Down Register Offset */

#define PORT_PULL_UP_REG_OFFSET 0x510 /* Pull Up Register Offset */

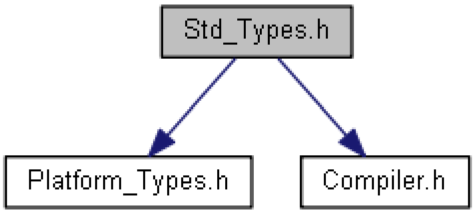
#define PORT_SLEW_RATE_CONTROL_OFFSET 0x518 /* Slew Rate control register offset */

#define SYSCTL_REGCGC2_REG (*((volatile uint32 *)0x400FE108))

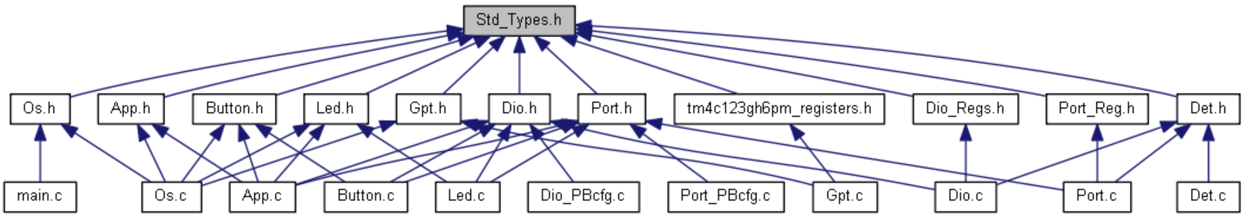
Std_Types.h File Reference

```
#include "Platform_Types.h"
#include "Compiler.h"
```

Include dependency graph for Std_Types.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Std_VersionInfoType**

Macros

- #define **STD_TYPES_VENDOR_ID**   (1000U)
- #define **STD_TYPES_SW_MAJOR_VERSION**   (1U)
- #define **STD_TYPES_SW_MINOR_VERSION**   (0U)
- #define **STD_TYPES_SW_PATCH_VERSION**   (0U)
- #define **STD_TYPES_AR_RELEASE_MAJOR_VERSION**   (4U)
- #define **STD_TYPES_AR_RELEASE_MINOR_VERSION**   (0U)
- #define **STD_TYPES_AR_RELEASE_PATCH_VERSION**   (3U)
- #define **STD_HIGH**   0x01U   /* Standard HIGH */
- #define **STD_LOW**   0x00U   /* Standard LOW */
- #define **STD_ACTIVE**   0x01U   /* Logical state active */
- #define **STD_IDLE**   0x00U   /* Logical state idle */
- #define **STD_ON**   0x01U   /* Standard ON */
- #define **STD_OFF**   0x00U   /* Standard **OFF** */
- #define **E_OK**   ((**Std_ReturnType**)0x00U)   /* Function Return OK */
- #define **E_NOT_OK**   ((**Std_ReturnType**)0x01U)   /* Function Return NOT OK */

Typedefs

- typedef **uint8 Std_ReturnType**

Macro Definition Documentation

#define E_NOT_OK ((Std_ReturnType)0x01U) /* Function Return NOT OK */

#define E_OK ((Std_ReturnType)0x00U) /* Function Return OK */

#define STD_ACTIVE 0x01U /* Logical state active */

#define STD_HIGH 0x01U /* Standard HIGH */

#define STD_IDLE 0x00U /* Logical state idle */

#define STD_LOW 0x00U /* Standard LOW */

#define STD_OFF 0x00U /* Standard OFF */

#define STD_ON 0x01U /* Standard ON */

#define STD_TYPES_AR_RELEASE_MAJOR_VERSION (4U)

#define STD_TYPES_AR_RELEASE_MINOR_VERSION (0U)

#define STD_TYPES_AR_RELEASE_PATCH_VERSION (3U)

#define STD_TYPES_SW_MAJOR_VERSION (1U)

#define STD_TYPES_SW_MINOR_VERSION (0U)

#define STD_TYPES_SW_PATCH_VERSION (0U)

#define STD_TYPES_VENDOR_ID (1000U)

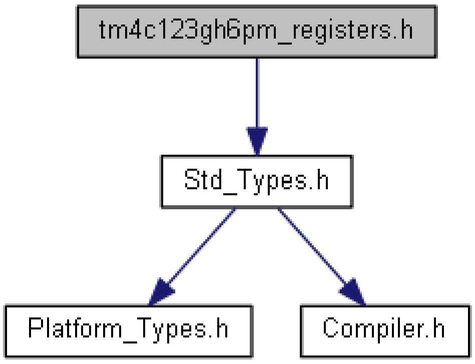
Typedef Documentation

typedef uint8 Std_ReturnType

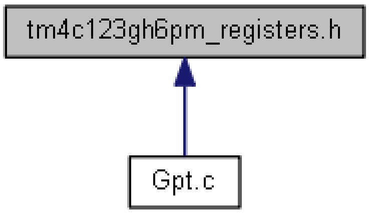
tm4c123gh6pm_registers.h File Reference

`#include "Std_Types.h"`

Include dependency graph for tm4c123gh6pm_registers.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **GPIO_PORTA_DATA_REG** (*((volatile **uint32** *)0x400043FC))
- #define **GPIO_PORTA_DIR_REG** (*((volatile **uint32** *)0x40004400))
- #define **GPIO_PORTA_AFSEL_REG** (*((volatile **uint32** *)0x40004420))
- #define **GPIO_PORTA_PUR_REG** (*((volatile **uint32** *)0x40004510))
- #define **GPIO_PORTA_PDR_REG** (*((volatile **uint32** *)0x40004514))
- #define **GPIO_PORTA_DEN_REG** (*((volatile **uint32** *)0x4000451C))
- #define **GPIO_PORTA_LOCK_REG** (*((volatile **uint32** *)0x40004520))
- #define **GPIO_PORTA_CR_REG** (*((volatile **uint32** *)0x40004524))
- #define **GPIO_PORTA_AMSEL_REG** (*((volatile **uint32** *)0x40004528))
- #define **GPIO_PORTA_PCTL_REG** (*((volatile **uint32** *)0x4000452C))
- #define **GPIO_PORTA_IS_REG** (*((volatile **uint32** *)0x40004404))
- #define **GPIO_PORTA_IBE_REG** (*((volatile **uint32** *)0x40004408))
- #define **GPIO_PORTA_IEV_REG** (*((volatile **uint32** *)0x4000440C))
- #define **GPIO_PORTA_IM_REG** (*((volatile **uint32** *)0x40004410))
- #define **GPIO_PORTA_RIS_REG** (*((volatile **uint32** *)0x40004414))
- #define **GPIO_PORTA_ICR_REG** (*((volatile **uint32** *)0x4000441C))
- #define **GPIO_PORTB_DATA_REG** (*((volatile **uint32** *)0x400053FC))
- #define **GPIO_PORTB_DIR_REG** (*((volatile **uint32** *)0x40005400))
- #define **GPIO_PORTB_AFSEL_REG** (*((volatile **uint32** *)0x40005420))
- #define **GPIO_PORTB_PUR_REG** (*((volatile **uint32** *)0x40005510))
- #define **GPIO_PORTB_PDR_REG** (*((volatile **uint32** *)0x40005514))
- #define **GPIO_PORTB_DEN_REG** (*((volatile **uint32** *)0x4000551C))
- #define **GPIO_PORTB_LOCK_REG** (*((volatile **uint32** *)0x40005520))
- #define **GPIO_PORTB_CR_REG** (*((volatile **uint32** *)0x40005524))
- #define **GPIO_PORTB_AMSEL_REG** (*((volatile **uint32** *)0x40005528))
- #define **GPIO_PORTB_PCTL_REG** (*((volatile **uint32** *)0x4000552C))
- #define **GPIO_PORTB_IS_REG** (*((volatile **uint32** *)0x40005404))
- #define **GPIO_PORTB_IBE_REG** (*((volatile **uint32** *)0x40005408))

- #define **GPIO_PORTB_IEV_REG** (*((volatile **uint32** *)0x4000540C))
- #define **GPIO_PORTB_IM_REG** (*((volatile **uint32** *)0x40005410))
- #define **GPIO_PORTB_RIS_REG** (*((volatile **uint32** *)0x40005414))
- #define **GPIO_PORTB_ICR_REG** (*((volatile **uint32** *)0x4000541C))
- #define **GPIO_PORTC_DATA_REG** (*((volatile **uint32** *)0x400063FC))
- #define **GPIO_PORTC_DIR_REG** (*((volatile **uint32** *)0x40006400))
- #define **GPIO_PORTC_AFSEL_REG** (*((volatile **uint32** *)0x40006420))
- #define **GPIO_PORTC_PUR_REG** (*((volatile **uint32** *)0x40006510))
- #define **GPIO_PORTC_PDR_REG** (*((volatile **uint32** *)0x40006514))
- #define **GPIO_PORTC_DEN_REG** (*((volatile **uint32** *)0x4000651C))
- #define **GPIO_PORTC_LOCK_REG** (*((volatile **uint32** *)0x40006520))
- #define **GPIO_PORTC_CR_REG** (*((volatile **uint32** *)0x40006524))
- #define **GPIO_PORTC_AMSEL_REG** (*((volatile **uint32** *)0x40006528))
- #define **GPIO_PORTC_PCTL_REG** (*((volatile **uint32** *)0x4000652C))
- #define **GPIO_PORTC_IS_REG** (*((volatile **uint32** *)0x40006404))
- #define **GPIO_PORTC_IBE_REG** (*((volatile **uint32** *)0x40006408))
- #define **GPIO_PORTC_IEV_REG** (*((volatile **uint32** *)0x4000640C))
- #define **GPIO_PORTC_IM_REG** (*((volatile **uint32** *)0x40006410))
- #define **GPIO_PORTC_RIS_REG** (*((volatile **uint32** *)0x40006414))
- #define **GPIO_PORTC_ICR_REG** (*((volatile **uint32** *)0x4000641C))
- #define **GPIO_PORTD_DATA_REG** (*((volatile **uint32** *)0x400073FC))
- #define **GPIO_PORTD_DIR_REG** (*((volatile **uint32** *)0x40007400))
- #define **GPIO_PORTD_AFSEL_REG** (*((volatile **uint32** *)0x40007420))
- #define **GPIO_PORTD_PUR_REG** (*((volatile **uint32** *)0x40007510))
- #define **GPIO_PORTD_PDR_REG** (*((volatile **uint32** *)0x40007514))
- #define **GPIO_PORTD_DEN_REG** (*((volatile **uint32** *)0x4000751C))
- #define **GPIO_PORTD_LOCK_REG** (*((volatile **uint32** *)0x40007520))
- #define **GPIO_PORTD_CR_REG** (*((volatile **uint32** *)0x40007524))
- #define **GPIO_PORTD_AMSEL_REG** (*((volatile **uint32** *)0x40007528))
- #define **GPIO_PORTD_PCTL_REG** (*((volatile **uint32** *)0x4000752C))
- #define **GPIO_PORTD_IS_REG** (*((volatile **uint32** *)0x40007404))
- #define **GPIO_PORTD_IBE_REG** (*((volatile **uint32** *)0x40007408))
- #define **GPIO_PORTD_IEV_REG** (*((volatile **uint32** *)0x4000740C))
- #define **GPIO_PORTD_IM_REG** (*((volatile **uint32** *)0x40007410))
- #define **GPIO_PORTD_RIS_REG** (*((volatile **uint32** *)0x40007414))
- #define **GPIO_PORTD_ICR_REG** (*((volatile **uint32** *)0x4000741C))
- #define **GPIO_PORTE_DATA_REG** (*((volatile **uint32** *)0x400243FC))
- #define **GPIO_PORTE_DIR_REG** (*((volatile **uint32** *)0x40024400))
- #define **GPIO_PORTE_AFSEL_REG** (*((volatile **uint32** *)0x40024420))
- #define **GPIO_PORTE_PUR_REG** (*((volatile **uint32** *)0x40024510))
- #define **GPIO_PORTE_PDR_REG** (*((volatile **uint32** *)0x40024514))
- #define **GPIO_PORTE_DEN_REG** (*((volatile **uint32** *)0x4002451C))
- #define **GPIO_PORTE_LOCK_REG** (*((volatile **uint32** *)0x40024520))
- #define **GPIO_PORTE_CR_REG** (*((volatile **uint32** *)0x40024524))
- #define **GPIO_PORTE_AMSEL_REG** (*((volatile **uint32** *)0x40024528))
- #define **GPIO_PORTE_PCTL_REG** (*((volatile **uint32** *)0x4002452C))
- #define **GPIO_PORTE_IS_REG** (*((volatile **uint32** *)0x40024404))
- #define **GPIO_PORTE_IBE_REG** (*((volatile **uint32** *)0x40024408))
- #define **GPIO_PORTE_IEV_REG** (*((volatile **uint32** *)0x4002440C))
- #define **GPIO_PORTE_IM_REG** (*((volatile **uint32** *)0x40024410))
- #define **GPIO_PORTE_RIS_REG** (*((volatile **uint32** *)0x40024414))
- #define **GPIO_PORTE_ICR_REG** (*((volatile **uint32** *)0x4002441C))
- #define **GPIO_PORTF_DATA_REG** (*((volatile **uint32** *)0x400253FC))
- #define **GPIO_PORTF_DIR_REG** (*((volatile **uint32** *)0x40025400))
- #define **GPIO_PORTF_AFSEL_REG** (*((volatile **uint32** *)0x40025420))
- #define **GPIO_PORTF_PUR_REG** (*((volatile **uint32** *)0x40025510))
- #define **GPIO_PORTF_PDR_REG** (*((volatile **uint32** *)0x40025514))

- #define **GPIO_PORTF_DEN_REG** (*((volatile **uint32** *)0x4002551C))
- #define **GPIO_PORTF_LOCK_REG** (*((volatile **uint32** *)0x40025520))
- #define **GPIO_PORTF_CR_REG** (*((volatile **uint32** *)0x40025524))
- #define **GPIO_PORTF_AMSEL_REG** (*((volatile **uint32** *)0x40025528))
- #define **GPIO_PORTF_PCTL_REG** (*((volatile **uint32** *)0x4002552C))
- #define **GPIO_PORTF_IS_REG** (*((volatile **uint32** *)0x40025404))
- #define **GPIO_PORTF_IBE_REG** (*((volatile **uint32** *)0x40025408))
- #define **GPIO_PORTF_IEV_REG** (*((volatile **uint32** *)0x4002540C))
- #define **GPIO_PORTF_IM_REG** (*((volatile **uint32** *)0x40025410))
- #define **GPIO_PORTF_RIS_REG** (*((volatile **uint32** *)0x40025414))
- #define **GPIO_PORTF_ICR_REG** (*((volatile **uint32** *)0x4002541C))
- #define **SYSCTL_REGCGC2_REG** (*((volatile **uint32** *)0x400FE108))
- #define **SYSTICK_CTRL_REG** (*((volatile **uint32** *)0xE000E010))
- #define **SYSTICK_RELOAD_REG** (*((volatile **uint32** *)0xE000E014))
- #define **SYSTICK_CURRENT_REG** (*((volatile **uint32** *)0xE000E018))
- #define **SYSCTL_RIS_REG** (*((volatile **uint32** *)0x400FE050))
- #define **SYSCTL_RCC_REG** (*((volatile **uint32** *)0x400FE060))
- #define **SYSCTL_RCC2_REG** (*((volatile **uint32** *)0x400FE070))
- #define **NVIC_PRI0_REG** (*((volatile **uint32** *)0xE000E400))
- #define **NVIC_PRI1_REG** (*((volatile **uint32** *)0xE000E404))
- #define **NVIC_PRI2_REG** (*((volatile **uint32** *)0xE000E408))
- #define **NVIC_PRI3_REG** (*((volatile **uint32** *)0xE000E40C))
- #define **NVIC_PRI4_REG** (*((volatile **uint32** *)0xE000E410))
- #define **NVIC_PRI5_REG** (*((volatile **uint32** *)0xE000E414))
- #define **NVIC_PRI6_REG** (*((volatile **uint32** *)0xE000E418))
- #define **NVIC_PRI7_REG** (*((volatile **uint32** *)0xE000E41C))
- #define **NVIC_PRI8_REG** (*((volatile **uint32** *)0xE000E420))
- #define **NVIC_PRI9_REG** (*((volatile **uint32** *)0xE000E424))
- #define **NVIC_PRI10_REG** (*((volatile **uint32** *)0xE000E428))
- #define **NVIC_PRI11_REG** (*((volatile **uint32** *)0xE000E42C))
- #define **NVIC_PRI12_REG** (*((volatile **uint32** *)0xE000E430))
- #define **NVIC_PRI13_REG** (*((volatile **uint32** *)0xE000E434))
- #define **NVIC_PRI14_REG** (*((volatile **uint32** *)0xE000E438))
- #define **NVIC_PRI15_REG** (*((volatile **uint32** *)0xE000E43C))
- #define **NVIC_PRI16_REG** (*((volatile **uint32** *)0xE000E440))
- #define **NVIC_PRI17_REG** (*((volatile **uint32** *)0xE000E444))
- #define **NVIC_PRI18_REG** (*((volatile **uint32** *)0xE000E448))
- #define **NVIC_PRI19_REG** (*((volatile **uint32** *)0xE000E44C))
- #define **NVIC_PRI20_REG** (*((volatile **uint32** *)0xE000E450))
- #define **NVIC_PRI21_REG** (*((volatile **uint32** *)0xE000E454))
- #define **NVIC_PRI22_REG** (*((volatile **uint32** *)0xE000E458))
- #define **NVIC_PRI23_REG** (*((volatile **uint32** *)0xE000E45C))
- #define **NVIC_PRI24_REG** (*((volatile **uint32** *)0xE000E460))
- #define **NVIC_PRI25_REG** (*((volatile **uint32** *)0xE000E464))
- #define **NVIC_PRI26_REG** (*((volatile **uint32** *)0xE000E468))
- #define **NVIC_PRI27_REG** (*((volatile **uint32** *)0xE000E46C))
- #define **NVIC_PRI28_REG** (*((volatile **uint32** *)0xE000E470))
- #define **NVIC_PRI29_REG** (*((volatile **uint32** *)0xE000E474))
- #define **NVIC_PRI30_REG** (*((volatile **uint32** *)0xE000E478))
- #define **NVIC_PRI31_REG** (*((volatile **uint32** *)0xE000E47C))
- #define **NVIC_PRI32_REG** (*((volatile **uint32** *)0xE000E480))
- #define **NVIC_PRI33_REG** (*((volatile **uint32** *)0xE000E484))
- #define **NVIC_PRI34_REG** (*((volatile **uint32** *)0xE000E488))
- #define **NVIC_EN0_REG** (*((volatile **uint32** *)0xE000E100))
- #define **NVIC_EN1_REG** (*((volatile **uint32** *)0xE000E104))
- #define **NVIC_EN2_REG** (*((volatile **uint32** *)0xE000E108))
- #define **NVIC_EN3_REG** (*((volatile **uint32** *)0xE000E10C))

- #define **NVIC_EN4_REG** (*((volatile **uint32** *)0xE000E110))
- #define **NVIC_DIS0_REG** (*((volatile **uint32** *)0xE000E180))
- #define **NVIC_DIS1_REG** (*((volatile **uint32** *)0xE000E184))
- #define **NVIC_DIS2_REG** (*((volatile **uint32** *)0xE000E188))
- #define **NVIC_DIS3_REG** (*((volatile **uint32** *)0xE000E18C))
- #define **NVIC_DIS4_REG** (*((volatile **uint32** *)0xE000E190))
- #define **NVIC_SYSTEM_PRI1_REG** (*((volatile **uint32** *)0xE000ED18))
- #define **NVIC_SYSTEM_PRI2_REG** (*((volatile **uint32** *)0xE000ED1C))
- #define **NVIC_SYSTEM_PRI3_REG** (*((volatile **uint32** *)0xE000ED20))
- #define **NVIC_SYSTEM_SYSHNDCTRL** (*((volatile **uint32** *)0xE000ED24))

Macro Definition Documentation

#define GPIO_PORTA_AFSEL_REG   (*((volatile uint32 *)0x40004420))

#define GPIO_PORTA_AMSEL_REG   (*((volatile uint32 *)0x40004528))

#define GPIO_PORTA_CR_REG   (*((volatile uint32 *)0x40004524))

#define GPIO_PORTA_DATA_REG   (*((volatile uint32 *)0x400043FC))

#define GPIO_PORTA_DEN_REG   (*((volatile uint32 *)0x4000451C))

#define GPIO_PORTA_DIR_REG   (*((volatile uint32 *)0x40004400))

#define GPIO_PORTA_IBE_REG   (*((volatile uint32 *)0x40004408))

#define GPIO_PORTA_ICR_REG   (*((volatile uint32 *)0x4000441C))

#define GPIO_PORTA_IEV_REG   (*((volatile uint32 *)0x4000440C))

#define GPIO_PORTA_IM_REG   (*((volatile uint32 *)0x40004410))

#define GPIO_PORTA_IS_REG   (*((volatile uint32 *)0x40004404))

#define GPIO_PORTA_LOCK_REG   (*((volatile uint32 *)0x40004520))

#define GPIO_PORTA_PCTL_REG   (*((volatile uint32 *)0x4000452C))

#define GPIO_PORTA_PDR_REG   (*((volatile uint32 *)0x40004514))

#define GPIO_PORTA_PUR_REG   (*((volatile uint32 *)0x40004510))

#define GPIO_PORTA_RIS_REG   (*((volatile uint32 *)0x40004414))

#define GPIO_PORTB_AFSEL_REG   (*((volatile uint32 *)0x40005420))

#define GPIO_PORTB_AMSEL_REG   (*((volatile uint32 *)0x40005528))

#define GPIO_PORTB_CR_REG   (*((volatile uint32 *)0x40005524))

#define GPIO_PORTB_DATA_REG   (*((volatile uint32 *)0x400053FC))

#define GPIO_PORTB_DEN_REG   (*((volatile uint32 *)0x4000551C))

#define GPIO_PORTB_DIR_REG   (*((volatile uint32 *)0x40005400))

#define GPIO_PORTB_IBE_REG   (*((volatile uint32 *)0x40005408))

#define GPIO_PORTB_ICR_REG   (*((volatile uint32 *)0x4000541C))

#define GPIO_PORTB_IEV_REG   (*((volatile uint32 *)0x4000540C))

```
#define GPIO_PORTB_IM_REG   (*((volatile uint32 *)0x40005410))

#define GPIO_PORTB_IS_REG   (*((volatile uint32 *)0x40005404))

#define GPIO_PORTB_LOCK_REG   (*((volatile uint32 *)0x40005520))

#define GPIO_PORTB_PCTL_REG   (*((volatile uint32 *)0x4000552C))

#define GPIO_PORTB_PDR_REG   (*((volatile uint32 *)0x40005514))

#define GPIO_PORTB_PUR_REG   (*((volatile uint32 *)0x40005510))

#define GPIO_PORTB_RIS_REG   (*((volatile uint32 *)0x40005414))

#define GPIO_PORTC_AFSEL_REG   (*((volatile uint32 *)0x40006420))

#define GPIO_PORTC_AMSEL_REG   (*((volatile uint32 *)0x40006528))

#define GPIO_PORTC_CR_REG   (*((volatile uint32 *)0x40006524))

#define GPIO_PORTC_DATA_REG   (*((volatile uint32 *)0x400063FC))

#define GPIO_PORTC_DEN_REG   (*((volatile uint32 *)0x4000651C))

#define GPIO_PORTC_DIR_REG   (*((volatile uint32 *)0x40006400))

#define GPIO_PORTC_IBE_REG   (*((volatile uint32 *)0x40006408))

#define GPIO_PORTC_ICR_REG   (*((volatile uint32 *)0x4000641C))

#define GPIO_PORTC_IEV_REG   (*((volatile uint32 *)0x4000640C))

#define GPIO_PORTC_IM_REG   (*((volatile uint32 *)0x40006410))

#define GPIO_PORTC_IS_REG   (*((volatile uint32 *)0x40006404))

#define GPIO_PORTC_LOCK_REG   (*((volatile uint32 *)0x40006520))

#define GPIO_PORTC_PCTL_REG   (*((volatile uint32 *)0x4000652C))

#define GPIO_PORTC_PDR_REG   (*((volatile uint32 *)0x40006514))

#define GPIO_PORTC_PUR_REG   (*((volatile uint32 *)0x40006510))

#define GPIO_PORTC_RIS_REG   (*((volatile uint32 *)0x40006414))

#define GPIO_PORTD_AFSEL_REG   (*((volatile uint32 *)0x40007420))

#define GPIO_PORTD_AMSEL_REG   (*((volatile uint32 *)0x40007528))

#define GPIO_PORTD_CR_REG   (*((volatile uint32 *)0x40007524))
```

```
#define GPIO_PORTD_DATA_REG   (*((volatile uint32 *)0x400073FC))

#define GPIO_PORTD_DEN_REG   (*((volatile uint32 *)0x4000751C))

#define GPIO_PORTD_DIR_REG   (*((volatile uint32 *)0x40007400))

#define GPIO_PORTD_IBE_REG   (*((volatile uint32 *)0x40007408))

#define GPIO_PORTD_ICR_REG   (*((volatile uint32 *)0x4000741C))

#define GPIO_PORTD_IEV_REG   (*((volatile uint32 *)0x4000740C))

#define GPIO_PORTD_IM_REG   (*((volatile uint32 *)0x40007410))

#define GPIO_PORTD_IS_REG   (*((volatile uint32 *)0x40007404))

#define GPIO_PORTD_LOCK_REG   (*((volatile uint32 *)0x40007520))

#define GPIO_PORTD_PCTL_REG   (*((volatile uint32 *)0x4000752C))

#define GPIO_PORTD_PDR_REG   (*((volatile uint32 *)0x40007514))

#define GPIO_PORTD_PUR_REG   (*((volatile uint32 *)0x40007510))

#define GPIO_PORTD_RIS_REG   (*((volatile uint32 *)0x40007414))

#define GPIO_PORTE_AFSEL_REG   (*((volatile uint32 *)0x40024420))

#define GPIO_PORTE_AMSEL_REG   (*((volatile uint32 *)0x40024528))

#define GPIO_PORTE_CR_REG   (*((volatile uint32 *)0x40024524))

#define GPIO_PORTE_DATA_REG   (*((volatile uint32 *)0x400243FC))

#define GPIO_PORTE_DEN_REG   (*((volatile uint32 *)0x4002451C))

#define GPIO_PORTE_DIR_REG   (*((volatile uint32 *)0x40024400))

#define GPIO_PORTE_IBE_REG   (*((volatile uint32 *)0x40024408))

#define GPIO_PORTE_ICR_REG   (*((volatile uint32 *)0x4002441C))

#define GPIO_PORTE_IEV_REG   (*((volatile uint32 *)0x4002440C))

#define GPIO_PORTE_IM_REG   (*((volatile uint32 *)0x40024410))

#define GPIO_PORTE_IS_REG   (*((volatile uint32 *)0x40024404))

#define GPIO_PORTE_LOCK_REG   (*((volatile uint32 *)0x40024520))

#define GPIO_PORTE_PCTL_REG   (*((volatile uint32 *)0x4002452C))
```

```
#define GPIO_PORTE_PDR_REG   (*((volatile uint32 *)0x40024514))

#define GPIO_PORTE_PUR_REG   (*((volatile uint32 *)0x40024510))

#define GPIO_PORTE_RIS_REG   (*((volatile uint32 *)0x40024414))

#define GPIO_PORTF_AFSEL_REG   (*((volatile uint32 *)0x40025420))

#define GPIO_PORTF_AMSEL_REG   (*((volatile uint32 *)0x40025528))

#define GPIO_PORTF_CR_REG   (*((volatile uint32 *)0x40025524))

#define GPIO_PORTF_DATA_REG   (*((volatile uint32 *)0x400253FC))

#define GPIO_PORTF_DEN_REG   (*((volatile uint32 *)0x4002551C))

#define GPIO_PORTF_DIR_REG   (*((volatile uint32 *)0x40025400))

#define GPIO_PORTF_IBE_REG   (*((volatile uint32 *)0x40025408))

#define GPIO_PORTF_ICR_REG   (*((volatile uint32 *)0x4002541C))

#define GPIO_PORTF_IEV_REG   (*((volatile uint32 *)0x4002540C))

#define GPIO_PORTF_IM_REG   (*((volatile uint32 *)0x40025410))

#define GPIO_PORTF_IS_REG   (*((volatile uint32 *)0x40025404))

#define GPIO_PORTF_LOCK_REG   (*((volatile uint32 *)0x40025520))

#define GPIO_PORTF_PCTL_REG   (*((volatile uint32 *)0x4002552C))

#define GPIO_PORTF_PDR_REG   (*((volatile uint32 *)0x40025514))

#define GPIO_PORTF_PUR_REG   (*((volatile uint32 *)0x40025510))

#define GPIO_PORTF_RIS_REG   (*((volatile uint32 *)0x40025414))

#define NVIC_DIS0_REG   (*((volatile uint32 *)0xE000E180))

#define NVIC_DIS1_REG   (*((volatile uint32 *)0xE000E184))

#define NVIC_DIS2_REG   (*((volatile uint32 *)0xE000E188))

#define NVIC_DIS3_REG   (*((volatile uint32 *)0xE000E18C))

#define NVIC_DIS4_REG   (*((volatile uint32 *)0xE000E190))

#define NVIC_EN0_REG   (*((volatile uint32 *)0xE000E100))

#define NVIC_EN1_REG   (*((volatile uint32 *)0xE000E104))
```

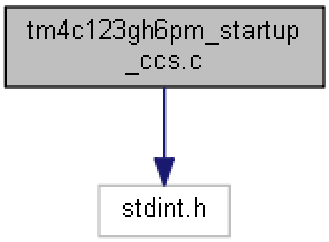
```
#define NVIC_EN2_REG   (*((volatile uint32 *)0xE000E108))

#define NVIC_EN3_REG   (*((volatile uint32 *)0xE000E10C))

#define NVIC_EN4_REG   (*((volatile uint32 *)0xE000E110))

#define NVIC_PRI0_REG   (*((volatile uint32 *)0xE000E400))

#define NVIC_PRI10_REG   (*((volatile uint32 *)0xE000E428))

#define NVIC_PRI11_REG   (*((volatile uint32 *)0xE000E42C))

#define NVIC_PRI12_REG   (*((volatile uint32 *)0xE000E430))

#define NVIC_PRI13_REG   (*((volatile uint32 *)0xE000E434))

#define NVIC_PRI14_REG   (*((volatile uint32 *)0xE000E438))

#define NVIC_PRI15_REG   (*((volatile uint32 *)0xE000E43C))

#define NVIC_PRI16_REG   (*((volatile uint32 *)0xE000E440))

#define NVIC_PRI17_REG   (*((volatile uint32 *)0xE000E444))

#define NVIC_PRI18_REG   (*((volatile uint32 *)0xE000E448))

#define NVIC_PRI19_REG   (*((volatile uint32 *)0xE000E44C))

#define NVIC_PRI1_REG   (*((volatile uint32 *)0xE000E404))

#define NVIC_PRI20_REG   (*((volatile uint32 *)0xE000E450))

#define NVIC_PRI21_REG   (*((volatile uint32 *)0xE000E454))

#define NVIC_PRI22_REG   (*((volatile uint32 *)0xE000E458))

#define NVIC_PRI23_REG   (*((volatile uint32 *)0xE000E45C))

#define NVIC_PRI24_REG   (*((volatile uint32 *)0xE000E460))

#define NVIC_PRI25_REG   (*((volatile uint32 *)0xE000E464))

#define NVIC_PRI26_REG   (*((volatile uint32 *)0xE000E468))

#define NVIC_PRI27_REG   (*((volatile uint32 *)0xE000E46C))

#define NVIC_PRI28_REG   (*((volatile uint32 *)0xE000E470))

#define NVIC_PRI29_REG   (*((volatile uint32 *)0xE000E474))

#define NVIC_PRI2_REG   (*((volatile uint32 *)0xE000E408))
```

```
#define NVIC_PRI30_REG   (*((volatile uint32 *)0xE000E478))

#define NVIC_PRI31_REG   (*((volatile uint32 *)0xE000E47C))

#define NVIC_PRI32_REG   (*((volatile uint32 *)0xE000E480))

#define NVIC_PRI33_REG   (*((volatile uint32 *)0xE000E484))

#define NVIC_PRI34_REG   (*((volatile uint32 *)0xE000E488))

#define NVIC_PRI3_REG   (*((volatile uint32 *)0xE000E40C))

#define NVIC_PRI4_REG   (*((volatile uint32 *)0xE000E410))

#define NVIC_PRI5_REG   (*((volatile uint32 *)0xE000E414))

#define NVIC_PRI6_REG   (*((volatile uint32 *)0xE000E418))

#define NVIC_PRI7_REG   (*((volatile uint32 *)0xE000E41C))

#define NVIC_PRI8_REG   (*((volatile uint32 *)0xE000E420))

#define NVIC_PRI9_REG   (*((volatile uint32 *)0xE000E424))

#define NVIC_SYSTEM_PRI1_REG   (*((volatile uint32 *)0xE000ED18))

#define NVIC_SYSTEM_PRI2_REG   (*((volatile uint32 *)0xE000ED1C))

#define NVIC_SYSTEM_PRI3_REG   (*((volatile uint32 *)0xE000ED20))

#define NVIC_SYSTEM_SYSHNDCTRL   (*((volatile uint32 *)0xE000ED24))

#define SYSCTL_RCC2_REG   (*((volatile uint32 *)0x400FE070))

#define SYSCTL_RCC_REG   (*((volatile uint32 *)0x400FE060))

#define SYSCTL_REGCGC2_REG   (*((volatile uint32 *)0x400FE108))

#define SYSCTL_RIS_REG   (*((volatile uint32 *)0x400FE050))

#define SYSTICK_CTRL_REG   (*((volatile uint32 *)0xE000E010))

#define SYSTICK_CURRENT_REG   (*((volatile uint32 *)0xE000E018))

#define SYSTICK_RELOAD_REG   (*((volatile uint32 *)0xE000E014))
```

tm4c123gh6pm_startup_ccs.c File Reference

`#include <stdint.h>`

Include dependency graph for tm4c123gh6pm_startup_ccs.c:



Functions

- void **ResetISR** (void)
- void **_c_int00** (void)

Variables

- uint32_t **__STACK_TOP**
- void(*const **g_pfnVectors** [])(void)

Function Documentation

void _c_int00 (void)

void ResetISR (void)

Variable Documentation

uint32_t __STACK_TOP

void(* const g_pfnVectors[])(void)

Index

INDEX