

NVIC Driver for M3 Guide

Generated by Doxygen 1.8.18

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 NVIC_t Struct Reference	5
4 File Documentation	7
4.1 DNVIC/DNVIC.c File Reference	7
4.1.1 Detailed Description	8
4.1.2 Function Documentation	9
4.1.2.1 DNVIC_ClearPendingIRQ()	9
4.1.2.2 DNVIC_DisableIRQ()	9
4.1.2.3 DNVIC_EnableIRQ()	10
4.1.2.4 DNVIC_GetActive()	10
4.1.2.5 DNVIC_GetPendingIRQ()	10
4.1.2.6 DNVIC_GetPriority()	11
4.1.2.7 DNVIC_SetPendingIRQ()	11
4.1.2.8 DNVIC_SetPriority()	12
4.1.2.9 DNVIC_SetPriorityGrouping()	12
4.1.2.10 DNVIC_voidChangeVectorOffset()	12
4.1.2.11 DNVIC_voidDisableAllFaults()	13
4.1.2.12 DNVIC_voidDisableAllPeripherals()	13
4.1.2.13 DNVIC_voidEnableAllFaults()	13
4.1.2.14 DNVIC_voidEnableAllPeripherals()	14
4.1.2.15 DNVIC_voidSetBASEPRI()	14
4.1.2.16 DNVIC_voidSysReset()	14
4.2 DNVIC/DNVIC.h File Reference	16
4.2.1 Detailed Description	18
4.2.2 Function Documentation	18
4.2.2.1 DNVIC_ClearPendingIRQ()	18
4.2.2.2 DNVIC_DisableIRQ()	19
4.2.2.3 DNVIC_EnableIRQ()	19
4.2.2.4 DNVIC_GetActive()	19
4.2.2.5 DNVIC_GetPendingIRQ()	20
4.2.2.6 DNVIC_GetPriority()	20
4.2.2.7 DNVIC_SetPendingIRQ()	21
4.2.2.8 DNVIC_SetPriority()	21
4.2.2.9 DNVIC_SetPriorityGrouping()	21
4.2.2.10 DNVIC_voidChangeVectorOffset()	22
4.2.2.11 DNVIC_voidDisableAllFaults()	22

4.2.2.12 DNVIC_voidDisableAllPeripherals()	23
4.2.2.13 DNVIC_voidEnableAllFaults()	23
4.2.2.14 DNVIC_voidEnableAllPeripherals()	23
4.2.2.15 DNVIC_voidSetBASEPRI()	24
4.2.2.16 DNVIC_voidSysReset()	24

Index	25
--------------	-----------

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

NVIC_t	5
----------------------------------	---

Chapter 2

File Index

2.1 File List

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

DNVIC/ DNVIC.c	
This file is the Implementation for NVIC Driver for Cortex M3	7
DNVIC/ DNVIC.h	
This file is the user interface for NVIC Driver for Cortex M3	16

Chapter 3

Data Structure Documentation

3.1 NVIC_t Struct Reference

Data Fields

- uint_32t **ISER** [8]
- uint_32t **Reserved** [24]
- uint_32t **ICER** [8]
- uint_32t **Reserved1** [24]
- uint_32t **ISPR** [8]
- uint_32t **Reserved2** [24]
- uint_32t **ICPR** [8]
- uint_32t **Reserved3** [24]
- uint_32t **IABR** [8]
- uint_32t **Reserved4** [56]
- uint_8t **IPR** [240]

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

- DNVIC/[DNVIC.c](#)

Chapter 4

File Documentation

4.1 DNVIC/DNVIC.c File Reference

This file is the Implementation for NVIC Driver for Cortex M3.

```
#include "STD_TYPES.h"
#include "DNVIC.h"
```

Data Structures

- struct [NVIC_t](#)

Macros

- #define [NVIC_BASE_ADDRESS](#) (([NVIC_t*](#)) 0XE000E100)
NVIC peripheral base address.
- #define [SCB_AIRCR](#) *((volatile uint_32t*) 0XE000ED0C)
NVIC Application Interrupt and Reset Control Register.
- #define [SCB_VTOR](#) *((volatile uint_32t*) 0xE000ED08)
Vector Table Offset register.
- #define [PASSWORD_MASK](#) 0X05FA0000
Password to make Software reset.
- #define [FLASH_BASE_ADDRESS](#) 0x08000000
Base Address of Flash.
- #define [OFFSET_POSITION](#) 0x08
used in shifting offset value to be corresponding for offset bits in Register
- #define [RESET_MASK](#) 0x04
Base Address of Flash.

Functions

- uint_8t [DNVIC_EnableIRQ](#) (uint_8t IRQn)
Function to Enable peripheral Interrupt.
- uint_8t [DNVIC_DisableIRQ](#) (uint_8t IRQn)
Function to Disable peripheral Interrupt.
- uint_8t [DNVIC_SetPendingIRQ](#) (uint_8t IRQn)
Function to Set Peripheral Pending Interrupt Flag By Software.
- uint_8t [DNVIC_ClearPendingIRQ](#) (uint_8t IRQn)
Function to Reset Peripheral Pending Interrupt Flag By Software.
- uint_8t [DNVIC_GetPendingIRQ](#) (uint_8t IRQn, uint_8t *Val)
Function to get Pending Flag Value.
- uint_8t [DNVIC_GetActive](#) (uint_8t IRQn, uint_8t *Val)
Function to get Active Flag Value.
- uint_8t [DNVIC_SetPriorityGrouping](#) (uint_32t priority_grouping)
Function provides priority grouping control for the exception model.
- uint_8t [DNVIC_SetPriority](#) (uint_8t IRQn, uint_8t priority)
Function to Set Interrupt Priority.
- uint_8t [DNVIC_GetPriority](#) (uint_8t IRQn, uint_8t *priority)
Function to Read peripheral Priority.
- void [DNVIC_voidDisableAllPeripherals](#) (void)
Function to Disable All peripherals interrupt.
- void [DNVIC_voidEnableAllPeripherals](#) (void)
Function to Enable All peripherals Interrupt.
- void [DNVIC_voidDisableAllFaults](#) (void)
Function to Disable all faults Interrupts.
- void [DNVIC_voidEnableAllFaults](#) (void)
Function to Enable all faults Interrupts.
- void [DNVIC_voidSetBASEPRI](#) (uint_8t priority)
Function to disable interrupts only with priority lower than certain level.
- void [DNVIC_voidChangeVectorOffset](#) (uint_32t offset)
Function to change vector Table offset
- void [DNVIC_voidSysReset](#) (void)
Function to make Software reset.

Variables

- [NVIC_t * NV](#) = [NVIC_BASE_ADDRESS](#)
NV pointer to struct [NVIC_t](#).

4.1.1 Detailed Description

This file is the Implementation for NVIC Driver for Cortex M3.

Author

Ahmed Qandeel (Ahmed.qandee196@gmail.com)

Version

0.1

Date

2020-06-05

Copyright

Copyright (c) 2020

4.1.2 Function Documentation**4.1.2.1 DNVIC_ClearPendingIRQ()**

```
uint_8t DNVIC_ClearPendingIRQ (
    uint_8t IRQn )
```

Function to Reset Peripheral Pending Interrupt Flag By Software.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.1.2.2 DNVIC_DisableIRQ()

```
uint_8t DNVIC_DisableIRQ (
    uint_8t IRQn )
```

Function to Disable peripheral Interrupt.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.1.2.3 DNVIC_EnableIRQ()

```
uint_8t DNVIC_EnableIRQ (
    uint_8t IRQn )
```

Function to Enable peripheral Interrupt.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.1.2.4 DNVIC_GetActive()

```
uint_8t DNVIC_GetActive (
    uint_8t IRQn,
    uint_8t * Val )
```

Function to get Active Flag Value.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
<i>Val</i>	pointer of uint_8t , Active Flag which to be read assigned to pointer

Returns

OK | NOT_OK

4.1.2.5 DNVIC_GetPendingIRQ()

```
uint_8t DNVIC_GetPendingIRQ (
    uint_8t IRQn,
    uint_8t * Val )
```

Function to get Pending Flag Value.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
<i>Val</i>	pointer of uint_8t , Pending Flag which to be read assigned to pointer

Returns

OK | NOT_OK

4.1.2.6 DNVIC_GetPriority()

```
uint_8t DNVIC_GetPriority (
    uint_8t IRQn,
    uint_8t * priority )
```

Function to Read peripheral Priority.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
<i>priority</i>	Pointer to uint_8t , Priority which to be read

Returns

OK | NOT_OK

4.1.2.7 DNVIC_SetPendingIRQ()

```
uint_8t DNVIC_SetPendingIRQ (
    uint_8t IRQn )
```

Function to Set Peripheral Pending Interrupt Flag By Software.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.1.2.8 DNVIC_SetPriority()

```
uint_8t DNVIC_SetPriority (
    uint_8t IRQn,
    uint_8t priority )
```

Function to Set Interrupt Priority.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
<i>priority</i>	Variable of uint_8t , Priority (0..16)

Returns

OK | NOT_OK

4.1.2.9 DNVIC_SetPriorityGrouping()

```
uint_8t DNVIC_SetPriorityGrouping (
    uint_32t priority_grouping )
```

Function provides priority grouping control for the exception model.

Parameters

<i>priority_grouping</i>	Variable of uint_32t , Priority Type (ALL_PREEMPTION , THREE_PREEMPTION_ONE_SUBGROUP)
--------------------------	---

Returns

OK | NOT_OK

4.1.2.10 DNVIC_voidChangeVectorOffset()

```
void DNVIC_voidChangeVectorOffset (
    uint_32t offset )
```

Function to change vector Table offset

Parameters

<i>offset</i>	Variable of uint_32t , Change vector Table offset
---------------	---

Returns

NA

4.1.2.11 DNVIC_voidDisableAllFaults()

```
void DNVIC_voidDisableAllFaults (  
    void )
```

Function to Disable all faults Interrupts.

Parameters

NA	
----	--

Returns

NA

4.1.2.12 DNVIC_voidDisableAllPeripherals()

```
void DNVIC_voidDisableAllPeripherals (  
    void )
```

Function to Disable All peripherals interrupt.

Parameters

NA	
----	--

Returns

NA

4.1.2.13 DNVIC_voidEnableAllFaults()

```
void DNVIC_voidEnableAllFaults (  
    void )
```

Function to Enable all faults Interrupts.

Parameters

NA	
----	--

Returns

NA

4.1.2.14 DNVIC_voidEnableAllPeripherals()

```
void DNVIC_voidEnableAllPeripherals (
    void )
```

Function to Enable All peripherals Interrupt.

Parameters

NA	
----	--

Returns

NA

4.1.2.15 DNVIC_voidSetBASEPRI()

```
void DNVIC_voidSetBASEPRI (
    uint_8t priority )
```

Function to disable interrupts only with priority lower than certain level.

Parameters

<i>priority</i>	Variable of uint_8t , Priority (0..16)
-----------------	--

Returns

NA

4.1.2.16 DNVIC_voidSysReset()

```
void DNVIC_voidSysReset (
    void )
```

Function to make Software reset.

Parameters

NA	
----	--

Returns

NA

4.2 DNVIC/DNVIC.h File Reference

This file is the user interface for NVIC Driver for Cortex M3.

Macros

- #define **WWDG** 0
- #define **PVD** 1
- #define **TAMPER** 2
- #define **RTC** 3
- #define **FLASH** 4
- #define **RCCINT** 5
- #define **EXTI0** 6
- #define **EXTI1** 7
- #define **EXTI2** 8
- #define **EXTI3** 9
- #define **EXTI4** 10
- #define **DMA1_CHANNEL1** 11
- #define **DMA1_CHANNEL2** 12
- #define **DMA1_CHANNEL3** 13
- #define **DMA1_CHANNEL4** 14
- #define **DMA1_CHANNEL5** 15
- #define **DMA1_CHANNEL6** 16
- #define **DMA1_CHANNEL7** 17
- #define **ADC1_2** 18
- #define **USB_HP_CAN_TX** 19
- #define **USB_LP_CAN_RX0** 20
- #define **CAN_RX1** 21
- #define **CAN_SCE** 22
- #define **EXTI9_5** 23
- #define **TIM1_BRK** 24
- #define **TIM1_UP** 25
- #define **TIM1_TRG_COM** 26
- #define **TIM1_CC** 27
- #define **TIM2** 28
- #define **TIM3** 29
- #define **TIM4** 30
- #define **I2C1_EV** 31
- #define **I2C1_ER** 32
- #define **I2C2_EV** 33
- #define **I2C2_ER** 34
- #define **SPI1** 35
- #define **SPI2** 36

- `#define USART1` 37
- `#define USART2` 38
- `#define USART3` 39
- `#define EXTI15_10` 40
- `#define RTCALARM` 41
- `#define USBWAKEUP` 42
- `#define TIM8_BRK` 43
- `#define TIM8_UP` 44
- `#define TIM8_TRG_COM` 45
- `#define TIM8_CC` 46
- `#define ADC3` 47
- `#define FSMC` 48
- `#define SDIO` 49
- `#define TIM5` 50
- `#define SPI3` 51
- `#define UART4` 52
- `#define UART5` 53
- `#define TIM6` 54
- `#define TIM7` 55
- `#define DMA2_CHANNEL1` 56
- `#define DMA2_CHANNEL2` 57
- `#define DMA2_CHANNEL3` 58
- `#define DMA2_CHANNEL4_5` 59
- `#define ALL_PREEMPTION` 0X00000300
- `#define THREE_PREEMPTION_ONE_SUBGROUP` 0X00000400
- `#define TWO_PREEMPTION_TWO_SUBGROUP` 0X00000500
- `#define ONE_PREEMPTION_THREE_SUBGROUP` 0X00000600
- `#define ALL_SUBGROUP` 0X00000700

Functions

- `uint_8t DNVIC_EnableIRQ` (`uint_8t IRQn`)
Function to Enable peripheral Interrupt.
- `uint_8t DNVIC_DisableIRQ` (`uint_8t IRQn`)
Function to Disable peripheral Interrupt.
- `uint_8t DNVIC_SetPendingIRQ` (`uint_8t IRQn`)
Function to Set Peripheral Pending Interrupt Flag By Software.
- `uint_8t DNVIC_ClearPendingIRQ` (`uint_8t IRQn`)
Function to Reset Peripheral Pending Interrupt Flag By Software.
- `uint_8t DNVIC_GetPendingIRQ` (`uint_8t IRQn`, `uint_8t *Val`)
Function to get Pending Flag Value.
- `uint_8t DNVIC_GetActive` (`uint_8t IRQn`, `uint_8t *Val`)
Function to get Active Flag Value.
- `uint_8t DNVIC_SetPriorityGrouping` (`uint_32t priority_grouping`)
Function provides priority grouping control for the exception model.
- `uint_8t DNVIC_SetPriority` (`uint_8t IRQn`, `uint_8t priority`)
Function to Set Interrupt Priority.
- `uint_8t DNVIC_GetPriority` (`uint_8t IRQn`, `uint_8t *priority`)
Function to Read peripheral Priority.
- `void DNVIC_voidDisableAllPeripherals` (`void`)
Function to Disable All peripherals interrupt.
- `void DNVIC_voidEnableAllPeripherals` (`void`)

- Function to Enable All peripherals Interrupt.*
 - void [DNVIC_voidDisableAllFaults](#) (void)
- Function to Disable all faults Interrupts.*
 - void [DNVIC_voidEnableAllFaults](#) (void)
- Function to Enable all faults Interrupts.*
 - void [DNVIC_voidSetBASEPRI](#) (uint_8t priority)
- Function to disable interrupts only with priority lower than certain level.*
 - void [DNVIC_voidChangeVectorOffset](#) (uint_32t offset)
- Function to change vector Table offset*
- void [DNVIC_voidSysReset](#) (void)
- Function to make Software reset.*

4.2.1 Detailed Description

This file is the user interface for NVIC Driver for Cortex M3.

Author

Ahmed Qandeel (Ahmed.qandeel96@gmail.com)

Version

0.1

Date

2020-06-05

Copyright

Copyright (c) 2020

4.2.2 Function Documentation

4.2.2.1 DNVIC_ClearPendingIRQ()

```
uint_8t DNVIC_ClearPendingIRQ (
    uint_8t IRQn )
```

Function to Reset Peripheral Pending Interrupt Flag By Software.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.2.2.2 DNVIC_DisableIRQ()

```
uint_8t DNVIC_DisableIRQ (
    uint_8t IRQn )
```

Function to Disable peripheral Interrupt.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.2.2.3 DNVIC_EnableIRQ()

```
uint_8t DNVIC_EnableIRQ (
    uint_8t IRQn )
```

Function to Enable peripheral Interrupt.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.2.2.4 DNVIC_GetActive()

```
uint_8t DNVIC_GetActive (
    uint_8t IRQn,
    uint_8t * Val )
```

Function to get Active Flag Value.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
<i>Val</i>	pointer of uint_8t , Active Flag which to be read assigned to pointer

Returns

OK | NOT_OK

4.2.2.5 DNVIC_GetPendingIRQ()

```
uint_8t DNVIC_GetPendingIRQ (
    uint_8t IRQn,
    uint_8t * Val )
```

Function to get Pending Flag Value.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
<i>Val</i>	pointer of uint_8t , Pending Flag which to be read assigned to pointer

Returns

OK | NOT_OK

4.2.2.6 DNVIC_GetPriority()

```
uint_8t DNVIC_GetPriority (
    uint_8t IRQn,
    uint_8t * priority )
```

Function to Read peripheral Priority.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offest of them in vector table array ex (WWDG , TIM2)
<i>priority</i>	Pointer to uint_8t , Priority which to be read

Returns

OK | NOT_OK

4.2.2.7 DNVIC_SetPendingIRQ()

```
uint_8t DNVIC_SetPendingIRQ (
    uint_8t IRQn )
```

Function to Set Peripheral Pending Interrupt Flag By Software.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
-------------	--

Returns

OK | NOT_OK

4.2.2.8 DNVIC_SetPriority()

```
uint_8t DNVIC_SetPriority (
    uint_8t IRQn,
    uint_8t priority )
```

Function to Set Interrupt Priority.

Parameters

<i>IRQn</i>	Variable of uint_8t , Interrupt Request Number is number assigned to peripheral as it is offset of them in vector table array ex (WWDG , TIM2)
<i>priority</i>	Variable of uint_8t , Priority (0..16)

Returns

OK | NOT_OK

4.2.2.9 DNVIC_SetPriorityGrouping()

```
uint_8t DNVIC_SetPriorityGrouping (
    uint_32t priority_grouping )
```

Function provides priority grouping control for the exception model.

Parameters

<i>priority_grouping</i>	Variable of uint_32t , Priority Type (ALL_PREEMPTION , THREE_PREEMPTION_ONE_SUBGROUP)
--------------------------	---

Returns

OK | NOT_OK

4.2.2.10 DNVIC_voidChangeVectorOffset()

```
void DNVIC_voidChangeVectorOffset (
    uint_32t offset )
```

Function to change vector Table offest

Parameters

<i>offset</i>	Variable of uint_32t , Change vector Table offest
---------------	---

Returns

NA

4.2.2.11 DNVIC_voidDisableAllFaults()

```
void DNVIC_voidDisableAllFaults (
    void )
```

Function to Disable all faults Interrupts.

Parameters

NA	
----	--

Returns

NA

4.2.2.12 DNVIC_voidDisableAllPeripherals()

```
void DNVIC_voidDisableAllPeripherals (
    void )
```

Function to Disable All peripherals interrupt.

Parameters

NA	
----	--

Returns

NA

4.2.2.13 DNVIC_voidEnableAllFaults()

```
void DNVIC_voidEnableAllFaults (
    void )
```

Function to Enable all faults Interrupts.

Parameters

NA	
----	--

Returns

NA

4.2.2.14 DNVIC_voidEnableAllPeripherals()

```
void DNVIC_voidEnableAllPeripherals (
    void )
```

Function to Enable All peripherals Interrupt.

Parameters

NA	
----	--

Returns

NA

4.2.2.15 DNVIC_voidSetBASEPRI()

```
void DNVIC_voidSetBASEPRI (
    uint_8t priority )
```

Function to disable interrupts only with priority lower than certain level.

Parameters

<i>priority</i>	Variable of uint_8t , Priority (0..16)
-----------------	--

Returns

NA

4.2.2.16 DNVIC_voidSysReset()

```
void DNVIC_voidSysReset (
    void )
```

Function to make Software reset.

Parameters

NA	
----	--

Returns

NA

Index

DNVIC.c

- DNVIC_ClearPendingIRQ, [9](#)
- DNVIC_DisableIRQ, [9](#)
- DNVIC_EnableIRQ, [10](#)
- DNVIC_GetActive, [10](#)
- DNVIC_GetPendingIRQ, [10](#)
- DNVIC_GetPriority, [11](#)
- DNVIC_SetPendingIRQ, [11](#)
- DNVIC_SetPriority, [11](#)
- DNVIC_SetPriorityGrouping, [12](#)
- DNVIC_voidChangeVectorOffset, [12](#)
- DNVIC_voidDisableAllFaults, [13](#)
- DNVIC_voidDisableAllPeripherals, [13](#)
- DNVIC_voidEnableAllFaults, [13](#)
- DNVIC_voidEnableAllPeripherals, [14](#)
- DNVIC_voidSetBASEPRI, [14](#)
- DNVIC_voidSysReset, [14](#)

DNVIC.h

- DNVIC_ClearPendingIRQ, [18](#)
- DNVIC_DisableIRQ, [19](#)
- DNVIC_EnableIRQ, [19](#)
- DNVIC_GetActive, [19](#)
- DNVIC_GetPendingIRQ, [20](#)
- DNVIC_GetPriority, [20](#)
- DNVIC_SetPendingIRQ, [21](#)
- DNVIC_SetPriority, [21](#)
- DNVIC_SetPriorityGrouping, [21](#)
- DNVIC_voidChangeVectorOffset, [22](#)
- DNVIC_voidDisableAllFaults, [22](#)
- DNVIC_voidDisableAllPeripherals, [22](#)
- DNVIC_voidEnableAllFaults, [23](#)
- DNVIC_voidEnableAllPeripherals, [23](#)
- DNVIC_voidSetBASEPRI, [24](#)
- DNVIC_voidSysReset, [24](#)

DNVIC/DNVIC.c, [7](#)

DNVIC/DNVIC.h, [16](#)

DNVIC_ClearPendingIRQ

- DNVIC.c, [9](#)
- DNVIC.h, [18](#)

DNVIC_DisableIRQ

- DNVIC.c, [9](#)
- DNVIC.h, [19](#)

DNVIC_EnableIRQ

- DNVIC.c, [10](#)
- DNVIC.h, [19](#)

DNVIC_GetActive

- DNVIC.c, [10](#)
- DNVIC.h, [19](#)

DNVIC_GetPendingIRQ

DNVIC.c, [10](#)

DNVIC.h, [20](#)

DNVIC_GetPriority

- DNVIC.c, [11](#)
- DNVIC.h, [20](#)

DNVIC_SetPendingIRQ

- DNVIC.c, [11](#)
- DNVIC.h, [21](#)

DNVIC_SetPriority

- DNVIC.c, [11](#)
- DNVIC.h, [21](#)

DNVIC_SetPriorityGrouping

- DNVIC.c, [12](#)
- DNVIC.h, [21](#)

DNVIC_voidChangeVectorOffset

- DNVIC.c, [12](#)
- DNVIC.h, [22](#)

DNVIC_voidDisableAllFaults

- DNVIC.c, [13](#)
- DNVIC.h, [22](#)

DNVIC_voidDisableAllPeripherals

- DNVIC.c, [13](#)
- DNVIC.h, [22](#)

DNVIC_voidEnableAllFaults

- DNVIC.c, [13](#)
- DNVIC.h, [23](#)

DNVIC_voidEnableAllPeripherals

- DNVIC.c, [14](#)
- DNVIC.h, [23](#)

DNVIC_voidSetBASEPRI

- DNVIC.c, [14](#)
- DNVIC.h, [24](#)

DNVIC_voidSysReset

- DNVIC.c, [14](#)
- DNVIC.h, [24](#)

NVIC_t, [5](#)