

Micromouse2019

Generated by Doxygen 1.8.16

1 Module Index	1
1.1 Modules	1
2 File Index	3
2.1 File List	3
3 Module Documentation	5
3.1 Template_Project	5
3.1.1 *	5
3.1.2 Detailed Description	5
3.1.3 Function Documentation	5
3.1.3.1 BusFault_Handler()	5
3.1.3.2 DebugMon_Handler()	6
3.1.3.3 HardFault_Handler()	6
3.1.3.4 MemManage_Handler()	6
3.1.3.5 NMI_Handler()	7
3.1.3.6 PendSV_Handler()	7
3.1.3.7 SVC_Handler()	7
3.1.3.8 SysTick_Handler()	8
3.1.3.9 UsageFault_Handler()	8
3.2 CMSIS	9
3.2.1 *	9
3.2.2 Detailed Description	9
3.3 Stm32f4xx_system	10
3.3.1 *	10
3.3.2 Detailed Description	10
3.4 STM32F4xx_System_Private_Includes	11
3.5 STM32F4xx_System_Private_TypesDefinitions	12
3.6 STM32F4xx_System_Private_Defines	13
3.6.1 *	13
3.6.2 Detailed Description	13
3.6.3 Macro Definition Documentation	13
3.6.3.1 VECT_TAB_OFFSET	13
3.7 STM32F4xx_System_Private_Macros	14
3.8 STM32F4xx_System_Private_Variables	15
3.8.1 *	15
3.8.2 Detailed Description	15
3.8.3 Variable Documentation	15
3.8.3.1 AHBPrescTable	15
3.9 STM32F4xx_System_Private_FunctionPrototypes	16
3.9.1 Detailed Description	16
3.10 STM32F4xx_System_Private_Functions	17
3.10.1 *	17

3.10.2 Detailed Description	17
3.10.3 Function Documentation	17
3.10.3.1 SystemCoreClockUpdate()	17
3.10.3.2 SystemInit()	18
4 File Documentation	19
4.1 src/main.c File Reference	19
4.1.1 *	19
4.1.2 Detailed Description	20
4.1.3 Function Documentation	20
4.1.3.1 function()	20
4.1.3.2 main()	21
4.2 src/stm32f4xx_it.c File Reference	21
4.2.1 *	21
4.2.2 Detailed Description	22
4.2.2.1 *	22
4.3 src/system_stm32f4xx.c File Reference	22
4.3.1 *	23
4.3.2 *	23
4.3.3 *	23
4.3.4 Detailed Description	23
4.3.5 *	24
4.3.5.1 *	24
4.3.5.2 *	24
4.3.5.3 *	24
4.3.5.4 *	24
4.3.5.5 *	24
4.3.5.6 *	24
4.3.5.7 *	24
4.3.5.8 *	24
4.3.5.9 *	24
4.3.5.10 *	24
4.3.5.11 *	24
4.3.5.12 *	24
4.3.5.13 *	24
4.3.5.14 *	24
4.3.5.15 *	24
4.3.5.16 *	24
4.3.5.17 *	24
4.3.5.18 *	24
4.3.5.19 *	24
4.3.5.20 *	24

4.3.5.21 *	24
4.3.5.22 *	25
4.3.5.23 *	25
Index	27

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Template_Project	5
CMSIS	9
Stm32f4xx_system	10
STM32F4xx_System_Private_Includes	11
STM32F4xx_System_Private_TypesDefinitions	12
STM32F4xx_System_Private_Defines	13
STM32F4xx_System_Private_Macros	14
STM32F4xx_System_Private_Variables	15
STM32F4xx_System_Private_FunctionPrototypes	16
STM32F4xx_System_Private_Functions	17

Chapter 2

File Index

2.1 File List

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

lib/ test_lib.h	??
src/ main.c	
Main program	19
src/ stm32f4xx_conf.h	??
src/ stm32f4xx_it.c	
Main Interrupt Service Routines. This file provides template for all exceptions handler and peripherals interrupt service routine	21
src/ stm32f4xx_it.h	??
src/ system_stm32f4xx.c	
CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices, and is generated by the clock configuration tool stm32f4xx_Clock_Configuration_V1.1.0.xls	22

Chapter 3

Module Documentation

3.1 Template_Project

3.1.1 *

Functions

- void [NMI_Handler](#) (void)
This function handles NMI exception.
- void [HardFault_Handler](#) (void)
This function handles Hard Fault exception.
- void [MemManage_Handler](#) (void)
This function handles Memory Manage exception.
- void [BusFault_Handler](#) (void)
This function handles Bus Fault exception.
- void [UsageFault_Handler](#) (void)
This function handles Usage Fault exception.
- void [SVC_Handler](#) (void)
This function handles SVCcall exception.
- void [DebugMon_Handler](#) (void)
This function handles Debug Monitor exception.
- void [PendSV_Handler](#) (void)
This function handles PendSVC exception.
- void [SysTick_Handler](#) (void)
This function handles SysTick Handler.

3.1.2 Detailed Description

3.1.3 Function Documentation

3.1.3.1 BusFault_Handler()

```
void BusFault_Handler (  
    void )
```

This function handles Bus Fault exception.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.1.3.2 DebugMon_Handler()

```
void DebugMon_Handler (  
    void )
```

This function handles Debug Monitor exception.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.1.3.3 HardFault_Handler()

```
void HardFault_Handler (  
    void )
```

This function handles Hard Fault exception.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.1.3.4 MemManage_Handler()

```
void MemManage_Handler (  
    void )
```

This function handles Memory Manage exception.

Parameters

None	
------	--

Return values

None	
------	--

3.1.3.5 NMI_Handler()

```
void NMI_Handler (  
                void )
```

This function handles NMI exception.

Parameters

None	
------	--

Return values

None	
------	--

3.1.3.6 PendSV_Handler()

```
void PendSV_Handler (  
                void )
```

This function handles PendSVC exception.

Parameters

None	
------	--

Return values

None	
------	--

3.1.3.7 SVC_Handler()

```
void SVC_Handler (  
                void )
```

This function handles SVCcall exception.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.1.3.8 SysTick_Handler()

```
void SysTick_Handler (
    void )
```

This function handles SysTick Handler.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.1.3.9 UsageFault_Handler()

```
void UsageFault_Handler (
    void )
```

This function handles Usage Fault exception.

Parameters

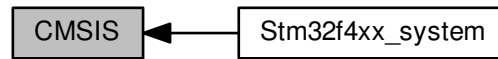
<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.2 CMSIS

Collaboration diagram for CMSIS:



3.2.1 *

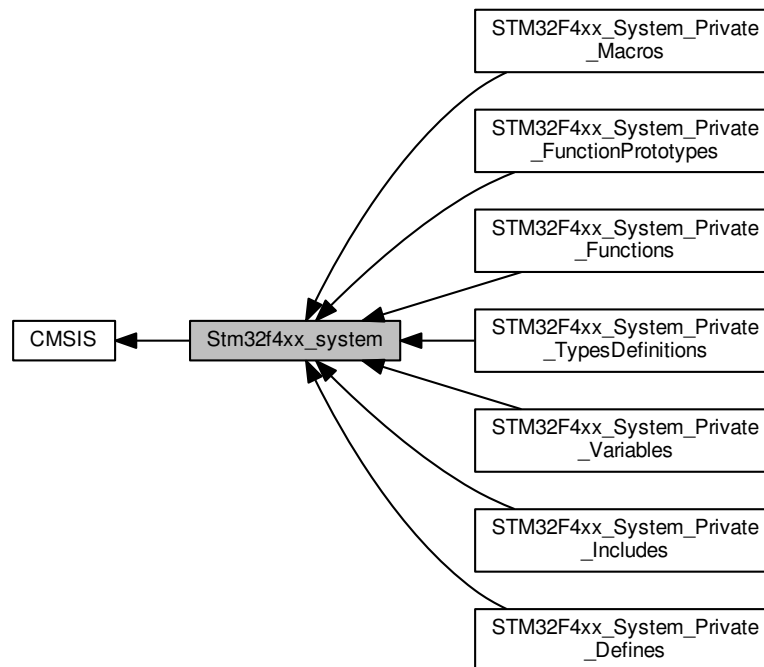
Modules

- [Stm32f4xx_system](#)

3.2.2 Detailed Description

3.3 Stm32f4xx_system

Collaboration diagram for Stm32f4xx_system:



3.3.1 *

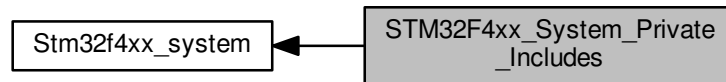
Modules

- [STM32F4xx_System_Private_Includes](#)
- [STM32F4xx_System_Private_TypesDefinitions](#)
- [STM32F4xx_System_Private_Defines](#)
- [STM32F4xx_System_Private_Macros](#)
- [STM32F4xx_System_Private_Variables](#)
- [STM32F4xx_System_Private_FunctionPrototypes](#)
- [STM32F4xx_System_Private_Functions](#)

3.3.2 Detailed Description

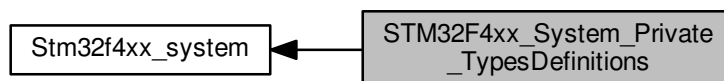
3.4 STM32F4xx_System_Private_Includes

Collaboration diagram for STM32F4xx_System_Private_Includes:



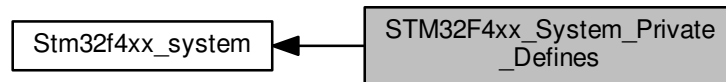
3.5 STM32F4xx_System_Private_TypesDefinitions

Collaboration diagram for STM32F4xx_System_Private_TypesDefinitions:



3.6 STM32F4xx_System_Private_Defines

Collaboration diagram for STM32F4xx_System_Private_Defines:



3.6.1 *

Macros

- `#define VECT_TAB_OFFSET 0x00`
- `#define PLL_M 16 /* 25 */`
- `#define PLL_N 336`
- `#define PLL_P 2`
- `#define PLL_Q 7`

3.6.2 Detailed Description

3.6.3 Macro Definition Documentation

3.6.3.1 VECT_TAB_OFFSET

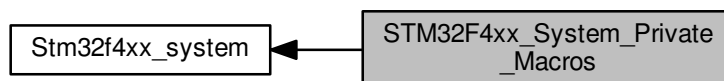
```
#define VECT_TAB_OFFSET 0x00
```

< Uncomment the following line if you need to use external SRAM mounted on STM324xG_EVAL/STM324x7I_↔ EVAL boards as data memory

< Uncomment the following line if you need to relocate your vector Table in Internal SRAM. Vector Table base offset field. This value must be a multiple of 0x200.

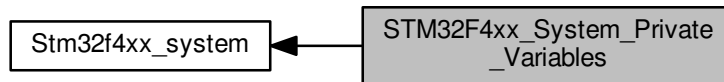
3.7 STM32F4xx_System_Private_Macros

Collaboration diagram for STM32F4xx_System_Private_Macros:



3.8 STM32F4xx_System_Private_Variables

Collaboration diagram for STM32F4xx_System_Private_Variables:



3.8.1 *

Variables

- uint32_t **SystemCoreClock** = 168000000
- __I uint8_t **AHBPrescTable** [16]

3.8.2 Detailed Description

3.8.3 Variable Documentation

3.8.3.1 AHBPrescTable

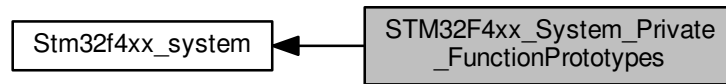
```
__I uint8_t AHBPrescTable[16]
```

Initial value:

```
=
    { 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9 }
```

3.9 STM32F4xx_System_Private_FunctionPrototypes

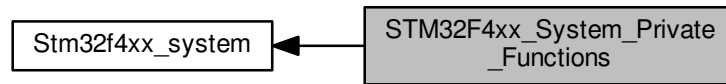
Collaboration diagram for STM32F4xx_System_Private_FunctionPrototypes:



3.9.1 Detailed Description

3.10 STM32F4xx_System_Private_Functions

Collaboration diagram for STM32F4xx_System_Private_Functions:



3.10.1 *

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

3.10.2 Detailed Description

3.10.3 Function Documentation

3.10.3.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
    void )
```

Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Note

Each time the core clock (HCLK) changes, this function must be called to update SystemCoreClock variable value. Otherwise, any configuration based on this variable will be incorrect.

- The system frequency computed by this function is not the real frequency in the chip. It is calculated based on the predefined constant and the selected clock source:

- If SYSCLK source is HSI, SystemCoreClock will contain the HSI_VALUE(*)
- If SYSCLK source is HSE, SystemCoreClock will contain the HSE_VALUE(**)
- If SYSCLK source is PLL, SystemCoreClock will contain the HSE_VALUE(**) or HSI_VALUE(*) multiplied/divided by the PLL factors.

(*) HSI_VALUE is a constant defined in stm32f4xx.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(**) HSE_VALUE is a constant defined in stm32f4xx.h file (default value 25 MHz), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

- The result of this function could be not correct when using fractional value for HSE crystal.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

3.10.3.2 SystemInit()

```
void SystemInit (  
                void )
```

Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Chapter 4

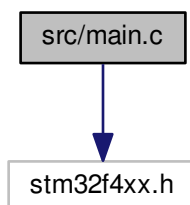
File Documentation

4.1 src/main.c File Reference

main program

```
#include "stm32f4xx.h"
```

Include dependency graph for main.c:



4.1.1 *

Functions

- int `function` (int a, int b)
- int `main` (void)

4.1.2 Detailed Description

main program

Author

Daniel Tar, Zoltar Resi, Andras Lauko

Version

V0.1

Date

04-03-2019

Attention

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.1.3 Function Documentation

4.1.3.1 function()

```
int function (
    int a,
    int b )
```

Parameters

<i>a</i>	- interger 1
<i>b</i>	- interger 2

Returns

the sum of intergers

4.1.3.2 main()

```
int main (
    void )
```

Abstract: main program

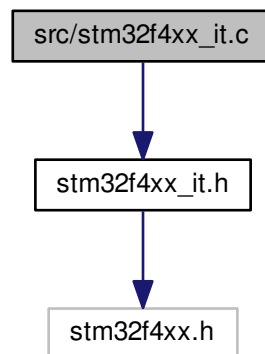
IMPORTANT NOTE! The symbol VECT_TAB_SRAM needs to be defined when building the project if code has been located to RAM and interrupts are used. Otherwise the interrupt table located in flash will be used. See also the <system_*.c> file and how the [SystemInit\(\)](#) function updates SCB->VTOR register. E.g. SCB->VTOR = 0x20000000;

4.2 src/stm32f4xx_it.c File Reference

Main Interrupt Service Routines. This file provides template for all exceptions handler and peripherals interrupt service routine.

```
#include "stm32f4xx_it.h"
```

Include dependency graph for stm32f4xx_it.c:



4.2.1 *

Functions

- void [NMI_Handler](#) (void)
This function handles NMI exception.
- void [HardFault_Handler](#) (void)
This function handles Hard Fault exception.
- void [MemManage_Handler](#) (void)
This function handles Memory Manage exception.
- void [BusFault_Handler](#) (void)
This function handles Bus Fault exception.
- void [UsageFault_Handler](#) (void)
This function handles Usage Fault exception.
- void [SVC_Handler](#) (void)

This function handles SVCcall exception.

- void [DebugMon_Handler](#) (void)

This function handles Debug Monitor exception.

- void [PendSV_Handler](#) (void)

This function handles PendSVC exception.

- void [SysTick_Handler](#) (void)

This function handles SysTick Handler.

4.2.2 Detailed Description

Main Interrupt Service Routines. This file provides template for all exceptions handler and peripherals interrupt service routine.

Author

MCD Application Team

Version

V1.1.0

Date

18-January-2013

Attention

4.2.2.1 *

© COPYRIGHT 2013 STMicroelectronics

Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.st.com/software_license_agreement_liberty_v2

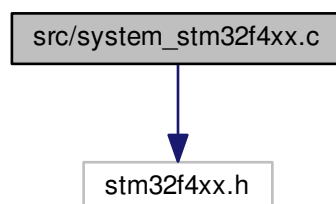
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

4.3 src/system_stm32f4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices, and is generated by the clock configuration tool `stm32f4xx_Clock_Configuration_V1.1.0.xls`.

```
#include "stm32f4xx.h"
```

Include dependency graph for `system_stm32f4xx.c`:



4.3.1 *

Macros

- #define `VECT_TAB_OFFSET` 0x00
- #define `PLL_M` 16 /* 25 */
- #define `PLL_N` 336
- #define `PLL_P` 2
- #define `PLL_Q` 7

4.3.2 *

Functions

- void `SystemInit` (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void `SystemCoreClockUpdate` (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

4.3.3 *

Variables

- uint32_t `SystemCoreClock` = 168000000
- __I uint8_t `AHBPrescTable` [16]

4.3.4 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File. This file contains the system clock configuration for STM32F4xx devices, and is generated by the clock configuration tool stm32f4xx_Clock_Configuration↔ V1.1.0.xls.

Author

MCD Application Team

Version

V1.1.0

Date

18-January-2013

1. This file provides two functions and one global variable to be called from user application:
 - `SystemInit()`: Setups the system clock (System clock source, PLL Multiplier and Divider factors, A↔ HB/APBx prescalers and Flash settings), depending on the configuration made in the clock xls tool. This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32f4xx.s" file.
 - SystemCoreClock variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
 - `SystemCoreClockUpdate()`: Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.
2. After each device reset the HSI (16 MHz) is used as system clock source. Then `SystemInit()` function is called, in "startup_stm32f4xx.s" file, to configure the system clock before to branch to main program.
3. If the system clock source selected by user fails to startup, the `SystemInit()` function will do nothing and HSI still used as system clock source. User can add some code to deal with this issue inside the SetSysClock() function.
4. The default value of HSE crystal is set to 25MHz, refer to "HSE_VALUE" define in "stm32f4xx.h" file. When HSE is used as system clock source, directly or through PLL, and you are using different crystal you have to adapt the HSE value to your own configuration.

4.3.5 *

5. This file configures the system clock as follows:

=====

4.3.5.1 *

Supported STM32F40xx/41xx/427x/437x devices

4.3.5.2 *

System Clock source | PLL (HSI) // (HSE)

4.3.5.3 *

SYSCLK(Hz) | 168000000

4.3.5.4 *

HCLK(Hz) | 168000000

4.3.5.5 *

AHB Prescaler | 1

4.3.5.6 *

APB1 Prescaler | 4

4.3.5.7 *

APB2 Prescaler | 2

4.3.5.8 *

HSI Frequency (Hz) // HSE Frequency(Hz) | 16000000 // 25000000

4.3.5.9 *

PLL_M | 16 // 25

4.3.5.10 *

PLL_N | 336

4.3.5.11 *

PLL_P | 2

4.3.5.12 *

PLL_Q | 7

4.3.5.13 *

PLLI2S_N | NA

4.3.5.14 *

PLLI2S_R | NA

4.3.5.15 *

I2S input clock | NA

4.3.5.16 *

VDD(V) | 3.3

4.3.5.17 *

Main regulator output voltage | Scale1 mode

4.3.5.18 *

Flash Latency(WS) | 5

4.3.5.19 *

Prefetch Buffer | ON

4.3.5.20 *

Instruction cache | ON

4.3.5.21 *

Data cache | ON

Require 48MHz for USB OTG FS, | Disabled

4.3.5.22 *

SDIO and RNG clock |

=====
Attention

4.3.5.23 *

© COPYRIGHT 2013 STMicroelectronics

Licensed under MCD-ST Liberty SW License Agreement V2, (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.st.com/software_license_agreement_liberty_v2

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Index

- AHBPrescTable
 - STM32F4xx_System_Private_Variables, [15](#)
- BusFault_Handler
 - Template_Project, [5](#)
- CMSIS, [9](#)
- DebugMon_Handler
 - Template_Project, [6](#)
- function
 - main.c, [20](#)
- HardFault_Handler
 - Template_Project, [6](#)
- main
 - main.c, [21](#)
- main.c
 - function, [20](#)
 - main, [21](#)
- MemManage_Handler
 - Template_Project, [6](#)
- NMI_Handler
 - Template_Project, [7](#)
- PendSV_Handler
 - Template_Project, [7](#)
- src/main.c, [19](#)
- src/stm32f4xx_it.c, [21](#)
- src/system_stm32f4xx.c, [22](#)
- Stm32f4xx_system, [10](#)
- STM32F4xx_System_Private_Defines, [13](#)
 - VECT_TAB_OFFSET, [13](#)
- STM32F4xx_System_Private_FunctionPrototypes, [16](#)
- STM32F4xx_System_Private_Functions, [17](#)
 - SystemCoreClockUpdate, [17](#)
 - SystemInit, [18](#)
- STM32F4xx_System_Private_Includes, [11](#)
- STM32F4xx_System_Private_Macros, [14](#)
- STM32F4xx_System_Private_TypesDefinitions, [12](#)
- STM32F4xx_System_Private_Variables, [15](#)
 - AHBPrescTable, [15](#)
- SVC_Handler
 - Template_Project, [7](#)
- SystemCoreClockUpdate
 - STM32F4xx_System_Private_Functions, [17](#)
- SystemInit
 - STM32F4xx_System_Private_Functions, [18](#)
- SysTick_Handler
 - Template_Project, [8](#)
- Template_Project, [5](#)
 - BusFault_Handler, [5](#)
 - DebugMon_Handler, [6](#)
 - HardFault_Handler, [6](#)
 - MemManage_Handler, [6](#)
 - NMI_Handler, [7](#)
 - PendSV_Handler, [7](#)
 - SVC_Handler, [7](#)
 - SysTick_Handler, [8](#)
 - UsageFault_Handler, [8](#)
- UsageFault_Handler
 - Template_Project, [8](#)
- VECT_TAB_OFFSET
 - STM32F4xx_System_Private_Defines, [13](#)