

# My Project

Generated by Doxygen 1.12.0



<b>1 Topic Index</b>	<b>1</b>
1.1 Topics	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Topic Documentation</b>	<b>5</b>
3.1 CMSIS	5
3.1.1 Detailed Description	5
3.1.2 Stm32g4xx_system	5
3.1.2.1 Detailed Description	5
3.1.2.2 STM32G4xx_System_Private_Includes	5
3.1.2.3 STM32G4xx_System_Private_TypesDefinitions	6
3.1.2.4 STM32G4xx_System_Private_Defines	6
3.1.2.5 STM32G4xx_System_Private_Macros	6
3.1.2.6 STM32G4xx_System_Private_Variables	6
3.1.2.7 STM32G4xx_System_Private_FunctionPrototypes	6
3.1.2.8 STM32G4xx_System_Private_Functions	6
<b>4 File Documentation</b>	<b>9</b>
4.1 adc.c File Reference	9
4.1.1 Detailed Description	9
4.1.2 Function Documentation	10
4.1.2.1 HAL_ADC_MspDeInit()	10
4.1.2.2 HAL_ADC_MspInit()	10
4.1.2.3 MX_ADC1_Init()	10
4.1.2.4 MX_ADC2_Init()	10
4.2 dma.c File Reference	11
4.2.1 Detailed Description	11
4.2.2 Function Documentation	11
4.2.2.1 MX_DMA_Init()	11
4.3 gpio.c File Reference	11
4.3.1 Detailed Description	12
4.3.2 Function Documentation	12
4.3.2.1 MX_GPIO_Init()	12
4.4 main.c File Reference	12
4.4.1 Detailed Description	13
4.4.2 Function Documentation	13
4.4.2.1 Error_Handler()	13
4.4.2.2 HAL_TIM_PeriodElapsedCallback()	13
4.4.2.3 main()	14
4.4.2.4 SystemClock_Config()	14
4.4.3 Variable Documentation	14

4.4.3.1 ADC_buffer	14
4.5 process.c File Reference	15
4.5.1 Detailed Description	15
4.5.2 Macro Definition Documentation	16
4.5.2.1 ADC_BUF_SIZE	16
4.5.2.2 ASCII_ENTER	16
4.5.2.3 CMD_BUFFER_SIZE	16
4.5.2.4 MAX_ARGS	16
4.5.2.5 MAX_SPEED	16
4.5.2.6 UART_RX_BUFFER_SIZE	16
4.5.2.7 UART_TX_BUFFER_SIZE	16
4.5.3 Function Documentation	17
4.5.3.1 ADC()	17
4.5.3.2 HAL_UART_RxCpltCallback()	17
4.5.3.3 processCommand()	17
4.5.3.4 shell()	17
4.5.3.5 speed()	17
4.5.4 Variable Documentation	18
4.5.4.1 ADC_buffer	18
4.5.4.2 argc	18
4.5.4.3 argv	18
4.5.4.4 ch_MCC	18
4.5.4.5 cmdBuffer	18
4.5.4.6 former_speed	18
4.5.4.7 idx_cmd	18
4.6 stm32g4xx_hal_msp.c File Reference	19
4.6.1 Detailed Description	19
4.6.2 Function Documentation	19
4.6.2.1 HAL_Msplnit()	19
4.7 stm32g4xx_hal_timebase_tim.c File Reference	19
4.7.1 Detailed Description	20
4.7.2 Function Documentation	20
4.7.2.1 HAL_InitTick()	20
4.7.2.2 HAL_ResumeTick()	20
4.7.2.3 HAL_SuspendTick()	21
4.8 stm32g4xx_it.c File Reference	21
4.8.1 Detailed Description	22
4.9 syscalls.c File Reference	22
4.9.1 Detailed Description	23
4.10 sysmem.c File Reference	23
4.10.1 Detailed Description	24
4.10.2 Function Documentation	24

4.10.2.1 _sbrk()	24
4.11 system_stm32g4xx.c File Reference	25
4.11.1 Detailed Description	25
4.11.2 This file configures the system clock as follows:	26
4.11.2.1 System Clock source   HSI	26
4.11.2.2 SYSCLK(Hz)   16000000	26
4.11.2.3 HCLK(Hz)   16000000	26
4.11.2.4 AHB Prescaler   1	26
4.11.2.5 APB1 Prescaler   1	26
4.11.2.6 APB2 Prescaler   1	26
4.11.2.7 PLL_M   1	26
4.11.2.8 PLL_N   16	26
4.11.2.9 PLL_P   7	26
4.11.2.10 PLL_Q   2	26
4.11.2.11 PLL_R   2	26
4.11.2.12 Require 48MHz for RNG   Disabled	26
4.12 tim.c File Reference	26
4.12.1 Detailed Description	26
4.12.2 Function Documentation	27
4.12.2.1 HAL_TIM_Encoder_MspDeInit()	27
4.12.2.2 HAL_TIM_Encoder_MspInit()	27
4.12.2.3 HAL_TIM_MspPostInit()	27
4.13 usart.c File Reference	27
4.13.1 Detailed Description	27
4.13.2 Function Documentation	28
4.13.2.1 HAL_UART_MspDeInit()	28
4.13.2.2 HAL_UART_MspInit()	28
<b>Index</b>	<b>29</b>



# Chapter 1

## Topic Index

### 1.1 Topics

Here is a list of all topics with brief descriptions:

CMSIS . . . . .	5
Stm32g4xx_system . . . . .	5
STM32G4xx_System_Private_Includes . . . . .	5
STM32G4xx_System_Private_TypesDefinitions . . . . .	6
STM32G4xx_System_Private_Defines . . . . .	6
STM32G4xx_System_Private_Macros . . . . .	6
STM32G4xx_System_Private_Variables . . . . .	6
STM32G4xx_System_Private_FunctionPrototypes . . . . .	6
STM32G4xx_System_Private_Functions . . . . .	6





## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">adc.c</a>	This file provides code for the configuration of the ADC instances . . . . .	9
<a href="#">dma.c</a>	This file provides code for the configuration of all the requested memory to memory DMA transfers . . . . .	11
<a href="#">gpio.c</a>	This file provides code for the configuration of all used GPIO pins . . . . .	11
<a href="#">main.c</a>	: Main program body . . . . .	12
<a href="#">process.c</a>	Implémentation des fonctions pour la gestion des commandes via un shell UART sur STM32 . . . . .	15
<a href="#">stm32g4xx_hal_msp.c</a>	This file provides code for the MSP Initialization and de-Initialization codes . . . . .	19
<a href="#">stm32g4xx_hal_timebase_tim.c</a>	HAL time base based on the hardware TIM . . . . .	19
<a href="#">stm32g4xx_it.c</a>	Interrupt Service Routines . . . . .	21
<a href="#">syscalls.c</a>	STM32CubeIDE Minimal System calls file . . . . .	22
<a href="#">systemem.c</a>	STM32CubeIDE System Memory calls file . . . . .	23
<a href="#">system_stm32g4xx.c</a>	CMSIS Cortex-M4 Device Peripheral Access Layer System Source File . . . . .	25
<a href="#">tim.c</a>	This file provides code for the configuration of the TIM instances . . . . .	26
<a href="#">usart.c</a>	This file provides code for the configuration of the USART instances . . . . .	27



# Chapter 3

## Topic Documentation

### 3.1 CMSIS

#### Topics

- [Stm32g4xx\\_system](#)

#### 3.1.1 Detailed Description

#### 3.1.2 Stm32g4xx\_system

#### Topics

- [STM32G4xx\\_System\\_Private\\_Includes](#)
- [STM32G4xx\\_System\\_Private\\_TypesDefinitions](#)
- [STM32G4xx\\_System\\_Private\\_Defines](#)
- [STM32G4xx\\_System\\_Private\\_Macros](#)
- [STM32G4xx\\_System\\_Private\\_Variables](#)
- [STM32G4xx\\_System\\_Private\\_FunctionPrototypes](#)
- [STM32G4xx\\_System\\_Private\\_Functions](#)

#### 3.1.2.1 Detailed Description

#### 3.1.2.2 STM32G4xx\_System\_Private\_Includes

#### Macros

- #define [HSE\\_VALUE](#) 24000000U
- #define [HSI\\_VALUE](#) 16000000U

### 3.1.2.2.1 Detailed Description

### 3.1.2.2.2 Macro Definition Documentation

#### 3.1.2.2.2.1 HSE\_VALUE

```
#define HSE_VALUE 24000000U
```

Value of the External oscillator in Hz

#### 3.1.2.2.2.2 HSI\_VALUE

```
#define HSI_VALUE 16000000U
```

Value of the Internal oscillator in Hz

### 3.1.2.3 STM32G4xx\_System\_Private\_TypeDefinitions

### 3.1.2.4 STM32G4xx\_System\_Private\_Defines

### 3.1.2.5 STM32G4xx\_System\_Private\_Macros

### 3.1.2.6 STM32G4xx\_System\_Private\_Variables

#### Variables

- uint32\_t **SystemCoreClock** = [HSI\\_VALUE](#)
- const uint8\_t **AHBPrescTable** [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8\_t **APBPrescTable** [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

### 3.1.2.6.1 Detailed Description

### 3.1.2.7 STM32G4xx\_System\_Private\_FunctionPrototypes

### 3.1.2.8 STM32G4xx\_System\_Private\_Functions

#### Functions

- void [SystemInit](#) (void)  
*Setup the microcontroller system.*
- 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.*

### 3.1.2.8.1 Detailed Description

### 3.1.2.8.2 Function Documentation

#### 3.1.2.8.2.1 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 stm32g4xx\_hal.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 stm32g4xx\_hal.h file (default value 24 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

None	
------	--

#### Return values

None	
------	--

#### 3.1.2.8.2.2 SystemInit()

```
void SystemInit (
    void )
```

Setup the microcontroller system.

**Parameters**

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

**Return values**

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

# Chapter 4

## File Documentation

### 4.1 adc.c File Reference

This file provides code for the configuration of the ADC instances.

```
#include "adc.h"
```

#### Functions

- void [MX\\_ADC1\\_Init](#) (void)
- void [MX\\_ADC2\\_Init](#) (void)
- void [HAL\\_ADC\\_MspInit](#) (ADC\_HandleTypeDef \*adcHandle)
- void [HAL\\_ADC\\_MspDeInit](#) (ADC\_HandleTypeDef \*adcHandle)

#### Variables

- ADC\_HandleTypeDef **hadc1**
- ADC\_HandleTypeDef **hadc2**
- DMA\_HandleTypeDef **hdma\_adc1**

#### 4.1.1 Detailed Description

This file provides code for the configuration of the ADC instances.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 4.1.2 Function Documentation

### 4.1.2.1 HAL\_ADC\_MspDeInit()

```
void HAL_ADC_MspDeInit (
    ADC_HandleTypeDef * adcHandle)
```

ADC1 GPIO Configuration PC2 -----> ADC1\_IN8 PA1 -----> ADC1\_IN2 PB0 -----> ADC1\_IN15 PB1 -----> ADC1\_IN12

ADC2 GPIO Configuration PC0 -----> ADC2\_IN6 PC1 -----> ADC2\_IN7 PC3 -----> ADC2\_IN9 PA0 -----> ADC2\_IN1

### 4.1.2.2 HAL\_ADC\_MspInit()

```
void HAL_ADC_MspInit (
    ADC_HandleTypeDef * adcHandle)
```

Initializes the peripherals clocks

ADC1 GPIO Configuration PC2 -----> ADC1\_IN8 PA1 -----> ADC1\_IN2 PB0 -----> ADC1\_IN15 PB1 -----> ADC1\_IN12

Initializes the peripherals clocks

ADC2 GPIO Configuration PC0 -----> ADC2\_IN6 PC1 -----> ADC2\_IN7 PC3 -----> ADC2\_IN9 PA0 -----> ADC2\_IN1

### 4.1.2.3 MX\_ADC1\_Init()

```
void MX_ADC1_Init (
    void )
```

Common config

Configure the ADC multi-mode

Configure Regular Channel

### 4.1.2.4 MX\_ADC2\_Init()

```
void MX_ADC2_Init (
    void )
```

Common config

Configure Regular Channel



## 4.2 dma.c File Reference

This file provides code for the configuration of all the requested memory to memory DMA transfers.

```
#include "dma.h"
```

### Functions

- void [MX\\_DMA\\_Init](#) (void)

### 4.2.1 Detailed Description

This file provides code for the configuration of all the requested memory to memory DMA transfers.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.2.2 Function Documentation

#### 4.2.2.1 MX\_DMA\_Init()

```
void MX_DMA_Init (  
    void )
```

Enable DMA controller clock

## 4.3 gpio.c File Reference

This file provides code for the configuration of all used GPIO pins.

```
#include "gpio.h"
```

### Functions

- void [MX\\_GPIO\\_Init](#) (void)

### 4.3.1 Detailed Description

This file provides code for the configuration of all used GPIO pins.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.3.2 Function Documentation

#### 4.3.2.1 MX\_GPIO\_Init()

```
void MX_GPIO_Init (  
    void )
```

Configure pins as Analog Input Output EVENT\_OUT EXTI PC10 -----> USART3\_TX PC11 -----> USART3\_RX

## 4.4 main.c File Reference

: Main program body

```
#include "main.h"  
#include "adc.h"  
#include "dma.h"  
#include "tim.h"  
#include "usart.h"  
#include "gpio.h"
```

#### Macros

- **#define UART\_TX\_BUFFER\_SIZE** 64
- **#define UART\_RX\_BUFFER\_SIZE** 1
- **#define CMD\_BUFFER\_SIZE** 64
- **#define MAX\_ARGS** 9
- **#define ASCII\_ENTER** 0x0D
- **#define MAX\_SPEED** 1024
- **#define ADC\_BUF\_SIZE** 8
- **#define PULSES\_PER\_REVOLUTION** 1000
- **#define SAMPLE\_PERIOD** 0.01

## Functions

- void [SystemClock\\_Config](#) (void)  
*System Clock Configuration.*
- int [main](#) (void)  
*The application entry point.*
- void [HAL\\_TIM\\_PeriodElapsedCallback](#) (TIM\_HandleTypeDef \*htim)  
*Period elapsed callback in non blocking mode.*
- void [Error\\_Handler](#) (void)  
*This function is executed in case of error occurrence.*

## Variables

- uint16\_t [ADC\\_buffer](#) [ADC\_BUF\_SIZE]
- char [Speed\\_buf](#) [5]

### 4.4.1 Detailed Description

: Main program body

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.4.2 Function Documentation

#### 4.4.2.1 Error\_Handler()

```
void Error_Handler (
    void )
```

This function is executed in case of error occurrence.

#### Return values

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

#### 4.4.2.2 HAL\_TIM\_PeriodElapsedCallback()

```
void HAL_TIM_PeriodElapsedCallback (
    TIM_HandleTypeDef * htim)
```

Period elapsed callback in non blocking mode.

#### Note

This function is called when TIM6 interrupt took place, inside HAL\_TIM\_IRQHandler(). It makes a direct call to HAL\_IncTick() to increment a global variable "uwTick" used as application time base.

**Parameters**

<i>htim</i>	: TIM handle
-------------	--------------

**Return values**

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

**4.4.2.3 main()**

```
int main (
    void )
```

The application entry point.

**Return values**

<i>int</i>	
------------	--

< Démarrage de la conversion ADC en mode DMA.

< Activation du Timer 1.

< Gestion des commandes utilisateur via le shell MCC.

**4.4.2.4 SystemClock\_Config()**

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

**Return values**

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

Configure the main internal regulator output voltage

Initializes the RCC Oscillators according to the specified parameters in the RCC\_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

**4.4.3 Variable Documentation****4.4.3.1 ADC\_buffer**

```
uint16_t ADC_buffer[ADC_BUF_SIZE]
```

Buffer pour les données ADC

## 4.5 process.c File Reference

Implémentation des fonctions pour la gestion des commandes via un shell UART sur STM32.

```
#include "process.h"
#include "usart.h"
#include "tim.h"
#include "adc.h"
```

### Macros

- `#define UART_TX_BUFFER_SIZE 64`
- `#define UART_RX_BUFFER_SIZE 1`
- `#define CMD_BUFFER_SIZE 64`
- `#define MAX_ARGS 9`
- `#define ASCII_ENTER 0x0D`
- `#define MAX_SPEED 1024`
- `#define ADC_BUF_SIZE 8`

### Functions

- void `HAL_UART_RxCpltCallback` (UART\_HandleTypeDef \*huart)  
*Callback pour la réception UART.*
- void `shell` ()  
*Fonction principale de gestion du shell UART.*
- void `speed` ()  
*Calcule et affiche la vitesse actuelle.*
- void `ADC` ()  
*Lit les données du capteur ADC et calcule le courant correspondant.*
- void `processCommand` (char \*cmd)  
*Traite une commande reçue via UART et exécute l'action correspondante.*

### Variables

- uint16\_t `ADC_buffer` [ADC\_BUF\_SIZE]
- char `cmdBuffer` [CMD\_BUFFER\_SIZE]
- int `idx_cmd`
- char \* `argv` [MAX\_ARGS]
- int `argc` = 0
- int32\_t `ch_MCC` = 0
- uint8\_t `former_speed` = 0

### 4.5.1 Detailed Description

Implémentation des fonctions pour la gestion des commandes via un shell UART sur STM32.

Ce fichier contient le code permettant de gérer des commandes utilisateur pour contrôler divers périphériques via un shell interactif.

#### Date

27 novembre 2024

#### Author

MAMOUR SARR

## 4.5.2 Macro Definition Documentation

### 4.5.2.1 ADC\_BUF\_SIZE

```
#define ADC_BUF_SIZE 8
```

Taille du buffer ADC

### 4.5.2.2 ASCII\_ENTER

```
#define ASCII_ENTER 0x0D
```

Code ASCII pour la touche ENTER

### 4.5.2.3 CMD\_BUFFER\_SIZE

```
#define CMD_BUFFER_SIZE 64
```

Taille maximale du buffer de commande

### 4.5.2.4 MAX\_ARGS

```
#define MAX_ARGS 9
```

Nombre maximal d'arguments pour une commande

### 4.5.2.5 MAX\_SPEED

```
#define MAX_SPEED 1024
```

Vitesse maximale pour le moteur

### 4.5.2.6 UART\_RX\_BUFFER\_SIZE

```
#define UART_RX_BUFFER_SIZE 1
```

Taille du buffer de réception UART

### 4.5.2.7 UART\_TX\_BUFFER\_SIZE

```
#define UART_TX_BUFFER_SIZE 64
```

Taille du buffer de transmission UART

### 4.5.3 Function Documentation

#### 4.5.3.1 ADC()

```
void ADC ()
```

Lit les données du capteur ADC et calcule le courant correspondant.

Cette fonction convertit les données brutes de l'ADC en tension, puis en courant en utilisant les spécifications du capteur (sensibilité et offset).

#### 4.5.3.2 HAL\_UART\_RxCpltCallback()

```
void HAL_UART_RxCpltCallback (  
    UART_HandleTypeDef * huart)
```

Callback pour la réception UART.

Cette fonction est appelée automatiquement lorsqu'une donnée est reçue via UART.

##### Parameters

<i>huart</i>	Pointeur vers la structure UART utilisée.
--------------	---

#### 4.5.3.3 processCommand()

```
void processCommand (  
    char * cmd)
```

Traite une commande reçue via UART et exécute l'action correspondante.

##### Parameters

<i>cmd</i>	La commande saisie par l'utilisateur sous forme de chaîne de caractères.
------------	--

#### 4.5.3.4 shell()

```
void shell ()
```

Fonction principale de gestion du shell UART.

Cette fonction traite les entrées utilisateur, exécute les commandes correspondantes et renvoie les résultats via UART.

#### 4.5.3.5 speed()

```
void speed ()
```

Calcule et affiche la vitesse actuelle.

Cette fonction lit un encodeur pour déterminer la vitesse du moteur et affiche les résultats via UART.

## 4.5.4 Variable Documentation

### 4.5.4.1 ADC\_buffer

```
uint16_t ADC_buffer[ADC_BUF_SIZE] [extern]
```

Buffer pour les données ADC

### 4.5.4.2 argc

```
int argc = 0
```

Nombre d'arguments pour la commande

### 4.5.4.3 argv

```
char* argv[MAX_ARGS]
```

Tableau des arguments d'une commande

### 4.5.4.4 ch\_MCC

```
int32_t ch_MCC = 0
```

État du module MCC

### 4.5.4.5 cmdBuffer

```
char cmdBuffer[CMD_BUFFER_SIZE]
```

Buffer pour les commandes utilisateur

### 4.5.4.6 former\_speed

```
uint8_t former_speed = 0
```

Dernière vitesse définie

### 4.5.4.7 idx\_cmd

```
int idx_cmd
```

Index courant dans le buffer de commande



## 4.6 stm32g4xx\_hal\_msp.c File Reference

This file provides code for the MSP Initialization and de-Initialization codes.

```
#include "main.h"
```

### Functions

- void [HAL\\_MspInit](#) (void)

### 4.6.1 Detailed Description

This file provides code for the MSP Initialization and de-Initialization codes.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.6.2 Function Documentation

#### 4.6.2.1 HAL\_MspInit()

```
void HAL_MspInit (  
    void )
```

Initializes the Global MSP. Disable the internal Pull-Up in Dead Battery pins of UCPD peripheral

## 4.7 stm32g4xx\_hal\_timebase\_tim.c File Reference

HAL time base based on the hardware TIM.

```
#include "stm32g4xx_hal.h"  
#include "stm32g4xx_hal_tim.h"
```

### Functions

- HAL\_StatusTypeDef [HAL\\_InitTick](#) (uint32\_t TickPriority)  
*This function configures the TIM6 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.*
- void [HAL\\_SuspendTick](#) (void)  
*Suspend Tick increment.*
- void [HAL\\_ResumeTick](#) (void)  
*Resume Tick increment.*

## Variables

- TIM\_HandleTypeDef **htim6**

### 4.7.1 Detailed Description

HAL time base based on the hardware TIM.

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.7.2 Function Documentation

#### 4.7.2.1 HAL\_InitTick()

```
HAL_StatusTypeDef HAL_InitTick (  
    uint32_t TickPriority)
```

This function configures the TIM6 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.

#### Note

This function is called automatically at the beginning of program after reset by HAL\_Init() or at any time when clock is configured, by HAL\_RCC\_ClockConfig().

#### Parameters

<i>TickPriority</i>	Tick interrupt priority.
---------------------	--------------------------

#### Return values

<i>HAL</i>	status
------------	--------

#### 4.7.2.2 HAL\_ResumeTick()

```
void HAL_ResumeTick (  
    void )
```

Resume Tick increment.

#### Note

Enable the tick increment by Enabling TIM6 update interrupt.

## Parameters

None	
------	--

## Return values

None	
------	--

### 4.7.2.3 HAL\_SuspendTick()

```
void HAL_SuspendTick (  
    void )
```

Suspend Tick increment.

## Note

Disable the tick increment by disabling TIM6 update interrupt.

## Parameters

None	
------	--

## Return values

None	
------	--

## 4.8 stm32g4xx\_it.c File Reference

Interrupt Service Routines.

```
#include "main.h"  
#include "stm32g4xx_it.h"
```

### Functions

- void **NMI\_Handler** (void)  
*This function handles Non maskable interrupt.*
- void **HardFault\_Handler** (void)  
*This function handles Hard fault interrupt.*
- void **MemManage\_Handler** (void)  
*This function handles Memory management fault.*
- void **BusFault\_Handler** (void)  
*This function handles Prefetch fault, memory access fault.*
- void **UsageFault\_Handler** (void)

*This function handles Undefined instruction or illegal state.*

- void **SVC\_Handler** (void)

*This function handles System service call via SWI instruction.*

- void **DebugMon\_Handler** (void)

*This function handles Debug monitor.*

- void **PendSV\_Handler** (void)

*This function handles Pendable request for system service.*

- void **SysTick\_Handler** (void)

*This function handles System tick timer.*

- void **DMA1\_Channel2\_IRQHandler** (void)

*This function handles DMA1 channel2 global interrupt.*

- void **USART2\_IRQHandler** (void)

*This function handles USART2 global interrupt / USART2 wake-up interrupt through EXTI line 26.*

- void **EXTI15\_10\_IRQHandler** (void)

*This function handles EXTI line[15:10] interrupts.*

- void **TIM6\_DAC\_IRQHandler** (void)

*This function handles TIM6 global interrupt, DAC1 and DAC3 channel underrun error interrupts.*

## Variables

- DMA\_HandleTypeDef **hdma\_adc1**
- UART\_HandleTypeDef **huart2**
- TIM\_HandleTypeDef **htim6**

## 4.8.1 Detailed Description

Interrupt Service Routines.

Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 4.9 syscalls.c File Reference

STM32CubeIDE Minimal System calls file.

```
#include <sys/stat.h>
#include <stdlib.h>
#include <errno.h>
#include <stdio.h>
#include <signal.h>
#include <time.h>
#include <sys/time.h>
#include <sys/times.h>
```

## Functions

- `int __io_putchar (int ch) __attribute__((weak))`
- `int __io_getchar (void)`
- `void initialise_monitor_handles ()`
- `int _getpid (void)`
- `int _kill (int pid, int sig)`
- `void _exit (int status)`
- `__attribute__((weak))`
- `int _close (int file)`
- `int _fstat (int file, struct stat *st)`
- `int _isatty (int file)`
- `int _lseek (int file, int ptr, int dir)`
- `int _open (char *path, int flags,...)`
- `int _wait (int *status)`
- `int _unlink (char *name)`
- `int _times (struct tms *buf)`
- `int _stat (char *file, struct stat *st)`
- `int _link (char *old, char *new)`
- `int _fork (void)`
- `int _execve (char *name, char **argv, char **env)`

## Variables

- `char ** environ = __env`

### 4.9.1 Detailed Description

STM32CubeIDE Minimal System calls file.

#### Author

Auto-generated by STM32CubeIDE

```
For more information about which c-functions
need which of these lowlevel functions
please consult the Newlib libc-manual
```

#### Attention

Copyright (c) 2020-2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 4.10 systemem.c File Reference

STM32CubeIDE System Memory calls file.

```
#include <errno.h>
#include <stdint.h>
```

## Functions

- void \* [\\_sbrk](#) (ptrdiff\_t incr)  
[\\_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

### 4.10.1 Detailed Description

STM32CubeIDE System Memory calls file.

#### Author

Generated by STM32CubeIDE

For more information about which C functions  
need which of these lowlevel functions  
please consult the newlib libc manual

#### Attention

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

### 4.10.2 Function Documentation

#### 4.10.2.1 [\\_sbrk\(\)](#)

```
void * _sbrk (
    ptrdiff_t incr)
```

[\\_sbrk\(\)](#) allocates memory to the newlib heap and is used by malloc and others from the C library

```
* #####
* # .data # .bss #          newlib heap          #          MSP stack          #
* #          #          #          # Reserved by _Min_Stack_Size #
* #####
* ^-- RAM start          ^-- _end          _estack, RAM end --^
*
```

This implementation starts allocating at the '\_end' linker symbol The '\_Min\_Stack\_Size' linker symbol reserves a memory for the MSP stack The implementation considers '\_estack' linker symbol to be RAM end NOTE: If the MSP stack, at any point during execution, grows larger than the reserved size, please increase the '\_Min\_Stack\_Size'.

#### Parameters

<i>incr</i>	Memory size
-------------	-------------

#### Returns

Pointer to allocated memory

## 4.11 system\_stm32g4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

```
#include "stm32g4xx.h"
```

### Macros

- #define [HSE\\_VALUE](#) 24000000U
- #define [HSI\\_VALUE](#) 16000000U

### Functions

- void [SystemInit](#) (void)  
*Setup the microcontroller system.*
- 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.*

### Variables

- uint32\_t [SystemCoreClock](#) = [HSI\\_VALUE](#)
- const uint8\_t [AHBPrescTable](#) [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}
- const uint8\_t [APBPrescTable](#) [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}

### 4.11.1 Detailed Description

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.

#### Author

MCD Application Team

This file provides two functions and one global variable to be called from user application:

- [SystemInit\(\)](#): This function is called at startup just after reset and before branch to main program. This call is made inside the "startup\_stm32g4xx.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.

After each device reset the HSI (16 MHz) is used as system clock source. Then [SystemInit\(\)](#) function is called, in "startup\_stm32g4xx.s" file, to configure the system clock before to branch to main program.

### 4.11.2 This file configures the system clock as follows:

4.11.2.1 System Clock source | HSI

4.11.2.2 SYSCCLK(Hz) | 16000000

4.11.2.3 HCLK(Hz) | 16000000

4.11.2.4 AHB Prescaler | 1

4.11.2.5 APB1 Prescaler | 1

4.11.2.6 APB2 Prescaler | 1

4.11.2.7 PLL\_M | 1

4.11.2.8 PLL\_N | 16

4.11.2.9 PLL\_P | 7

4.11.2.10 PLL\_Q | 2

4.11.2.11 PLL\_R | 2

4.11.2.12 Require 48MHz for RNG | Disabled

=====

Attention

Copyright (c) 2019 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 4.12 tim.c File Reference

This file provides code for the configuration of the TIM instances.

```
#include "tim.h"
```

### Functions

- void **MX\_TIM1\_Init** (void)
- void **MX\_TIM3\_Init** (void)
- void **HAL\_TIM\_Base\_MspInit** (TIM\_HandleTypeDef \*tim\_baseHandle)
- void **HAL\_TIM\_Encoder\_MspInit** (TIM\_HandleTypeDef \*tim\_encoderHandle)
- void **HAL\_TIM\_MspPostInit** (TIM\_HandleTypeDef \*timHandle)
- void **HAL\_TIM\_Base\_MspDeInit** (TIM\_HandleTypeDef \*tim\_baseHandle)
- void **HAL\_TIM\_Encoder\_MspDeInit** (TIM\_HandleTypeDef \*tim\_encoderHandle)

### Variables

- TIM\_HandleTypeDef **htim1**
- TIM\_HandleTypeDef **htim3**

### 4.12.1 Detailed Description

This file provides code for the configuration of the TIM instances.



**Attention**

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

**4.12.2 Function Documentation****4.12.2.1 HAL\_TIM\_Encoder\_MspDeInit()**

```
void HAL_TIM_Encoder_MspDeInit (
    TIM_HandleTypeDef * tim_encoderHandle)
TIM3 GPIO Configuration PA4 -----> TIM3_CH2 PA6 -----> TIM3_CH1 PC8 -----> TIM3_CH3
```

**4.12.2.2 HAL\_TIM\_Encoder\_MspInit()**

```
void HAL_TIM_Encoder_MspInit (
    TIM_HandleTypeDef * tim_encoderHandle)
TIM3 GPIO Configuration PA4 -----> TIM3_CH2 PA6 -----> TIM3_CH1 PC8 -----> TIM3_CH3
```

**4.12.2.3 HAL\_TIM\_MspPostInit()**

```
void HAL_TIM_MspPostInit (
    TIM_HandleTypeDef * timHandle)
TIM1 GPIO Configuration PB13 -----> TIM1_CH1N PB14 -----> TIM1_CH2N PB15 -----> TIM1_CH3N PA8 -----
> TIM1_CH1 PA9 -----> TIM1_CH2 PA10 -----> TIM1_CH3
```

**4.13 usart.c File Reference**

This file provides code for the configuration of the USART instances.

```
#include "usart.h"
```

**Functions**

- void **MX\_USART2\_UART\_Init** (void)
- void [HAL\\_UART\\_MspInit](#) (UART\_HandleTypeDef \*uartHandle)
- void [HAL\\_UART\\_MspDeInit](#) (UART\_HandleTypeDef \*uartHandle)

**Variables**

- UART\_HandleTypeDef **huart2**

**4.13.1 Detailed Description**

This file provides code for the configuration of the USART instances.

**Attention**

Copyright (c) 2024 STMicroelectronics. All rights reserved.

This software is licensed under terms that can be found in the LICENSE file in the root directory of this software component. If no LICENSE file comes with this software, it is provided AS-IS.

## 4.13.2 Function Documentation

### 4.13.2.1 HAL\_UART\_MspDeInit()

```
void HAL_UART_MspDeInit (  
    UART_HandleTypeDef * uartHandle)  
USART2 GPIO Configuration PA2 -----> USART2_TX PA3 -----> USART2_RX
```

### 4.13.2.2 HAL\_UART\_MspInit()

```
void HAL_UART_MspInit (  
    UART_HandleTypeDef * uartHandle)  
Initializes the peripherals clocks  
USART2 GPIO Configuration PA2 -----> USART2_TX PA3 -----> USART2_RX
```

# Index

- [\\_sbrk](#)
    - [sysmem.c, 24](#)
- ADC
  - [process.c, 17](#)
- [adc.c, 9](#)
  - [HAL\\_ADC\\_MspDeInit, 10](#)
  - [HAL\\_ADC\\_MspInit, 10](#)
  - [MX\\_ADC1\\_Init, 10](#)
  - [MX\\_ADC2\\_Init, 10](#)
- ADC\_BUF\_SIZE
  - [process.c, 16](#)
- ADC\_buffer
  - [main.c, 14](#)
  - [process.c, 18](#)
- [argc](#)
  - [process.c, 18](#)
- [argv](#)
  - [process.c, 18](#)
- ASCII\_ENTER
  - [process.c, 16](#)
- ch\_MCC
  - [process.c, 18](#)
- CMD\_BUFFER\_SIZE
  - [process.c, 16](#)
- cmdBuffer
  - [process.c, 18](#)
- CMSIS, 5
- [dma.c, 11](#)
  - [MX\\_DMA\\_Init, 11](#)
- Error\_Handler
  - [main.c, 13](#)
- former\_speed
  - [process.c, 18](#)
- [gpio.c, 11](#)
  - [MX\\_GPIO\\_Init, 12](#)
- HAL\_ADC\_MspDeInit
  - [adc.c, 10](#)
- HAL\_ADC\_MspInit
  - [adc.c, 10](#)
- HAL\_InitTick
  - [stm32g4xx\\_hal\\_timebase\\_tim.c, 20](#)
- HAL\_MspInit
  - [stm32g4xx\\_hal\\_msp.c, 19](#)
- HAL\_ResumeTick
  - [stm32g4xx\\_hal\\_timebase\\_tim.c, 20](#)
- HAL\_SuspendTick
  - [stm32g4xx\\_hal\\_timebase\\_tim.c, 21](#)
- HAL\_TIM\_Encoder\_MspDeInit
  - [tim.c, 27](#)
- HAL\_TIM\_Encoder\_MspInit
  - [tim.c, 27](#)
- HAL\_TIM\_MspPostInit
  - [tim.c, 27](#)
- HAL\_TIM\_PeriodElapsedCallback
  - [main.c, 13](#)
- HAL\_UART\_MspDeInit
  - [usart.c, 28](#)
- HAL\_UART\_MspInit
  - [usart.c, 28](#)
- HAL\_UART\_RxCpltCallback
  - [process.c, 17](#)
- HSE\_VALUE
  - [STM32G4xx\\_System\\_Private\\_Includes, 6](#)
- HSI\_VALUE
  - [STM32G4xx\\_System\\_Private\\_Includes, 6](#)
- idx\_cmd
  - [process.c, 18](#)
- main
  - [main.c, 14](#)
- [main.c, 12](#)
  - [ADC\\_buffer, 14](#)
  - [Error\\_Handler, 13](#)
  - [HAL\\_TIM\\_PeriodElapsedCallback, 13](#)
  - [main, 14](#)
  - [SystemClock\\_Config, 14](#)
- MAX\_ARGS
  - [process.c, 16](#)
- MAX\_SPEED
  - [process.c, 16](#)
- MX\_ADC1\_Init
  - [adc.c, 10](#)
- MX\_ADC2\_Init
  - [adc.c, 10](#)
- MX\_DMA\_Init
  - [dma.c, 11](#)
- MX\_GPIO\_Init
  - [gpio.c, 12](#)
- [process.c, 15](#)
  - [ADC, 17](#)
  - [ADC\\_BUF\\_SIZE, 16](#)
  - [ADC\\_buffer, 18](#)

- argc, [18](#)
- argv, [18](#)
- ASCII\_ENTER, [16](#)
- ch\_MCC, [18](#)
- CMD\_BUFFER\_SIZE, [16](#)
- cmdBuffer, [18](#)
- former\_speed, [18](#)
- HAL\_UART\_RxCpltCallback, [17](#)
- idx\_cmd, [18](#)
- MAX\_ARGS, [16](#)
- MAX\_SPEED, [16](#)
- processCommand, [17](#)
- shell, [17](#)
- speed, [17](#)
- UART\_RX\_BUFFER\_SIZE, [16](#)
- UART\_TX\_BUFFER\_SIZE, [16](#)
- processCommand
  - process.c, [17](#)
- shell
  - process.c, [17](#)
- speed
  - process.c, [17](#)
- stm32g4xx\_hal\_msp.c, [19](#)
  - HAL\_MspInit, [19](#)
- stm32g4xx\_hal\_timebase\_tim.c, [19](#)
  - HAL\_InitTick, [20](#)
  - HAL\_ResumeTick, [20](#)
  - HAL\_SuspendTick, [21](#)
- stm32g4xx\_it.c, [21](#)
- Stm32g4xx\_system, [5](#)
- STM32G4xx\_System\_Private\_Defines, [6](#)
- STM32G4xx\_System\_Private\_FunctionPrototypes, [6](#)
- STM32G4xx\_System\_Private\_Functions, [6](#)
  - SystemCoreClockUpdate, [7](#)
  - SystemInit, [7](#)
- STM32G4xx\_System\_Private\_Includes, [5](#)
  - HSE\_VALUE, [6](#)
  - HSI\_VALUE, [6](#)
- STM32G4xx\_System\_Private\_Macros, [6](#)
- STM32G4xx\_System\_Private\_TypesDefinitions, [6](#)
- STM32G4xx\_System\_Private\_Variables, [6](#)
- syscalls.c, [22](#)
- sysmem.c, [23](#)
  - \_sbrk, [24](#)
- system\_stm32g4xx.c, [25](#)
- SystemClock\_Config
  - main.c, [14](#)
- SystemCoreClockUpdate
  - STM32G4xx\_System\_Private\_Functions, [7](#)
- SystemInit
  - STM32G4xx\_System\_Private\_Functions, [7](#)
- tim.c, [26](#)
  - HAL\_TIM\_Encoder\_MspDeInit, [27](#)
  - HAL\_TIM\_Encoder\_MspInit, [27](#)
  - HAL\_TIM\_MspPostInit, [27](#)
- UART\_RX\_BUFFER\_SIZE
  - process.c, [16](#)
- UART\_TX\_BUFFER\_SIZE
  - process.c, [16](#)
- usart.c, [27](#)
  - HAL\_UART\_MspDeInit, [28](#)
  - HAL\_UART\_MspInit, [28](#)