



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Modules

Here is a list of all modules:

[detail level [1](#) [2](#) [3](#) [4](#) [5](#)]

▼ DRIVERS	Driver Layer
▼ BSP	BSP Layer
▼ COMPONENTS	Componer
STSPIN230_Motor_Driver_handler	Handler for STSPIN23 driver
▼ STSPIN230	STSPIN23 section
STSPIN230MotorDriver	API pointer for STSPIN23
EnableInput_CH1_E_CH2_E_CH3_D	Enable Input for CH1 and CH2 for STSPIN23
EnableInput_CH1_E_CH2_D_CH3_E	Enable Input for CH1 and CH3 for STSPIN23
EnableInput_CH1_D_CH2_E_CH3_E	Enable Input for CH2 and CH3 for STSPIN23
DisableInput_CH1_D_CH2_D_CH3_D	Enable Input for CH1 and CH2 for STSPIN23
Start_PWM_driving	Enable PWM driving

		for STSPIN
Stop_PWM_driving		Disable PV channels for STSPIN23
HF_TIMx_SetDutyCycle_CH1		Set the Duty value for Channel 1
HF_TIMx_SetDutyCycle_CH2		Set the Duty value for Channel 2
HF_TIMx_SetDutyCycle_CH3		Set the Duty value for Channel 3
Current_Reference_Start		Enable the Reference
Current_Reference_Stop		Disable the Reference
Current_Reference_Setvalue		Set the value for Current Reference
▼ X-NUCLEO-IHM11M1		X-Nucleo evaluation board
STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D		Enable Input for STSPIN23
STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E		Enable Input for STSPIN23
STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E		Enable Input for STSPIN23
STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D		Disable All channels for STSPIN23
STSPIN230_Start_PWM_driving		Enable the generation channels for STSPIN23
STSPIN230_Stop_PWM_driving		Disable the generation channels for STSPIN23

<code>STSPIN230_HF_TIMx_SetDutyCycle_CH1</code>	Set the Du value for C STSPIN23
<code>STSPIN230_HF_TIMx_SetDutyCycle_CH2</code>	Set the Du value for C STSPIN23
<code>STSPIN230_HF_TIMx_SetDutyCycle_CH3</code>	Set the Du value for C STSPIN23
<code>STSPIN230_Current_Reference_Start</code>	Enable the Reference for STSPIN
<code>STSPIN230_Current_Reference_Stop</code>	Disable the Reference for STSPIN
<code>STSPIN230_Current_Reference_Setvalue</code>	Set the val Current Re STSPIN23
<code>BSP_X_NUCLEO_FAULT_LED_ON</code>	Turns sele On
<code>BSP_X_NUCLEO_FAULT_LED_OFF</code>	Turns sele Off
▼ MIDDLEWARES	Middlewar
▼ MC_6-STEP_LIB	Motor Con
<code>Exported_types</code>	
<code>Exported_function_6StepLib</code>	
<code>MC_SixStep_TABLE</code>	Set the per (TIMx, GPI each step
<code>MC_SixStep_NEXT_step</code>	Generate t number ac the directio CCW)
<code>MC_SixStep_RESET</code>	Reset all v used for 6s

		algorithm
MC_SixStep_Ramp_Motor_calc		Calculate the acceleration step by step during start-up
MC_SixStep_ARR_step		Generate the value for L Frequency start-up
MC_SixStep_Alignment		Generate the alignment
MC_SixStep_Speed_Val_target_potentiometer		Calculate the Speed value threshold according with the potentiometer value
MC_SixStep_Speed_Potentiometer		Calculate the potentiometer set the Motor
MC_Set_PI_param		Set all parameters for the PI regulator
MC_PI_Controller		Compute the torque for the Current Reference
MC_Task_Speed		Main task: with PI regulation
MC_Set_Speed		Set the new speed value
MC_Bemf_Delay		Take the delay after each commutation
MC_StartMotor		Start the Motor
MC_StopMotor		Stop the Motor
MC_GetEISpeedHz		Get the Electrical Speed from

		of LF TIM
MC_GetMechSpeedRPM	Get the Me	Motor Spe
MC_SixStep_Init_main_data	Init the ma	for motor d
MC_SixStep_INIT	Initialitatio	SixStep lib
MC_TIMx_SixStep_timebase	Low Frequ	Callback -
	Call back -	next step a
		the filtered
		value
MC_Speed_Filter	Calculate t	filtered
MC_Potentiometer_filter	Calculate t	potentiome
MC_SysTick_SixStep_MediumFrequencyTask	Systick Ca	the Speed
MC_SixStep_ARR_Bemf	Calculate t	Autoreload
	(ARR) for I	(ARR) for I
	Frequency	Frequency
MC_ADCx_SixStep_Bemf	Compute t	crossing de
MC_EXT_button_SixStep	GPIO EXT	Start or Sto
	Start or Sto	through the
		button on
		STM32Nuc
Main_Motor_parameters	All motor p	for 6Step c
▼ stm32F401_nucleo_ihm11m1	Interface fi	STM32F40
	STM32F40	Motor Con
		figuration

MC_SixStep_ADC_Channel	Select the Channel
MC_SixStep_Nucleo_Init	Init the STI
START_DAC	Start DAC
STOP_DAC	Stop DAC
SET_DAC_value	Set DAC value debug
HAL_ADC_ConvCpltCallback	ADC callback
HAL_TIM_PeriodElapsedCallback	Htim callback
HAL_SYSTICK_Callback	Systick callback
HAL_GPIO_EXTI_Callback	EXT callback
EnableInput_CH1_E_CH2_E_CH3_D	Enable Input CH1 and C STSPIN23
EnableInput_CH1_E_CH2_D_CH3_E	Enable Input CH1 and C STSPIN23
EnableInput_CH1_D_CH2_E_CH3_E	Enable Input CH2 and C STSPIN23
DisableInput_CH1_D_CH2_D_CH3_D	Enable Input CH2 and C STSPIN23
Start_PWM_driving	Enable PWM for STSPIN
Stop_PWM_driving	Disable PWM channels for STSPIN23
HF_TIMx_SetDutyCycle_CH1	Set the Duty value for C
HF_TIMx_SetDutyCycle_CH2	Set the Duty value for C
HF_TIMx_SetDutyCycle_CH3	Set the Duty value for C
Current_Reference_Start	Enable the

		Reference
Current_Reference_Stop		Disable the Reference
Current_Reference_Setvalue		Set the value of Current Reference
Bemf_delay_calc		Bemf delay calculation
Get_UART_data		Get the UART data from DR register
Exported_function_F401		
▼ UART_UI		Serial communication through PC terminal
Exported_function_Uart		

Generated by  1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

DRIVERS

Driver Layer. [More...](#)

Modules

BSP

BSP Layer.

Detailed Description

Driver Layer.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

BSP

[DRIVERS](#)

BSP Layer. [More...](#)

Modules

COMPONENTS

Components.

X-NUCLEO-IHM11M1

X-Nucleo expansion board.

Detailed Description

BSP Layer.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

COMPONENTS

[DRIVERS](#) » [BSP](#)

Components. More...

Modules

STSPIN230_Motor_Driver_handler

Handler for STSPIN230 Motor driver.

STSPIN230

STSPIN230 driver section.

Detailed Description

Components.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Data Structures

STSPIN230_Motor_Driver_handler

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#)

Handler for STSPIN230 Motor driver. [More...](#)

Data Structures

```
struct STSPIN230_MotorDriver_TypeDef
```

Detailed Description

Handler for STSPIN230 Motor driver.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
Data Structures	Data Structure Index	Data Fields	Data Fields
			STSPIN230_MotorDriver_TypeDef Struct Reference

#include <[MC_Common.h](#)>

Data Fields

```
void(* EnableInput_CH1_E_CH2_E_CH3_D )(void)
```

```
void(* EnableInput_CH1_E_CH2_D_CH3_E )(void)
```

```
void(* EnableInput_CH1_D_CH2_E_CH3_E )(void)
```

```
void(* DisableInput_CH1_D_CH2_D_CH3_D )(void)
```

```
void(* Start_PWM_driving )(void)
```

```
void(* Stop_PWM_driving )(void)
```

```
void(* HF_TIMx_SetDutyCycle_CH1 )(uint16_t)
```

```
void(* HF_TIMx_SetDutyCycle_CH2 )(uint16_t)
```

```
void(* HF_TIMx_SetDutyCycle_CH3 )(uint16_t)
```

```
void(* Current_Reference_Start )(void)
```

```
void(* Current_Reference_Stop )(void)
```

```
void(* Current_Reference_Setvalue )(uint16_t)
```

Detailed Description

Definition at line **63** of file **MC_Common.h**.

Field Documentation

```
void(*  
STSPIN230_MotorDriver_TypeDef::Current_Reference_Setvalue)  
(uint16_t)
```

Set current reference value for closed loop control

Definition at line [76](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_Current_Reference_Setvalue\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::Current_Reference_Start)  
(void)
```

Start current reference generation for closed loop control

Definition at line [74](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_Current_Reference_Start\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::Current_Reference_Stop)  
(void)
```

Stop current reference generation for closed loop control

Definition at line [75](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_Current_Reference_Stop\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::DisableInput_CH1_D_CH2_D_C  
(void)
```

Disable all channels

Definition at line [68](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_D_CH2_E_CI  
(void)
```

Enable the channel 2,3 and Disable the channel 1

Definition at line [67](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_D_CI  
(void)
```

Enable the channel 1,3 and Disable the channel 2

Definition at line [66](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_E_CI  
(void)
```

Enable the channel 1,2 and Disable the channel 3

Definition at line [65](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH1)  
(uint16_t)
```

High Frequency Timer - Change DutyCycle value for CH1

Definition at line [71](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_HF_TIMx_SetDutyCycle_CH1\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH2)  
(uint16_t)
```

High Frequency Timer - Change DutyCycle value for CH2

Definition at line [72](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_HF_TIMx_SetDutyCycle_CH2\(\)](#).

```
void(*  
STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH3)  
(uint16_t)
```

High Frequency Timer - Change DutyCycle value for CH3

Definition at line [73](#) of file [MC_Common.h](#).

Referenced by [MC_SixStep_HF_TIMx_SetDutyCycle_CH3\(\)](#).

```
void(* STSPIN230_MotorDriver_TypeDef::Start_PWM_driving)
```

(void)

Start PWM generation

Definition at line **69** of file **MC_Common.h**.

Referenced by **MC_SixStep_Start_PWM_driving()**.

void(* STSPIN230_MotorDriver_TypeDef::Stop_PWM_driving)(void)

Stop PWM generation

Definition at line **70** of file **MC_Common.h**.

Referenced by **MC_SixStep_Stop_PWM_driving()**.

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

- **MC_Common.h**



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

STSPIN230

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#)

STSPIN230 driver section. More...

Modules

STSPIN230MotorDriver

API pointer for STSPIN230.

EnableInput_CH1_E_CH2_E_CH3_D

Enable Input channel CH1 and CH2 for STSPIN230.

EnableInput_CH1_E_CH2_D_CH3_E

Enable Input channel CH1 and CH3 for STSPIN230.

EnableInput_CH1_D_CH2_E_CH3_E

Enable Input channel CH2 and CH3 for STSPIN230.

DisableInput_CH1_D_CH2_D_CH3_D

Enable Input channel CH2 and CH3 for STSPIN230.

Start_PWM_driving

Enable PWM channels for STSPIN230.

Stop_PWM_driving

Disable PWM channels for STSPIN230.

HF_TIMx_SetDutyCycle_CH1

Set the Duty Cycle value for CH1.

HF_TIMx_SetDutyCycle_CH2

Set the Duty Cycle value for CH2.

HF_TIMx_SetDutyCycle_CH3

Set the Duty Cycle value for CH3.

Current_Reference_Start

Enable the Current Reference generation.

Current_Reference_Stop

Disable the Current Reference generation.

Current_Reference_Setvalue

Set the value for Current Reference.

Detailed Description

STSPIN230 driver section.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Variables

STSPIN230MotorDriver

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230](#)

API pointer for STSPIN230. [More...](#)

Variables

STSPIN230_MotorDriver_TypeDef STSPIN230MotorDriver
It handles all API functions for
STSPIN230 MC Driver. [More...](#)

Detailed Description

API pointer for STSPIN230.

Variable Documentation

STSPIN230_MotorDriver_TypeDef STSPIN230MotorDriver

Initial value:

```
=  
{  
    EnableInput_CH1_E_CH2_E_CH3_D,  
    EnableInput_CH1_E_CH2_D_CH3_E,  
    EnableInput_CH1_D_CH2_E_CH3_E,  
    DisableInput_CH1_D_CH2_D_CH3_D,  
    Start_PWM_driving,  
    Stop_PWM_driving,  
    HF_TIMx_SetDutyCycle_CH1,  
    HF_TIMx_SetDutyCycle_CH2,  
    HF_TIMx_SetDutyCycle_CH3,  
    Current_Reference_Start,  
    Current_Reference_Stop,  
    Current_Reference_Setvalue,  
}
```

It handles all API functions for STSPIN230 MC Driver.

Return values

None

Definition at line **70** of file **STSPIN230.c**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

EnableInput_CH1_E_CH2_E_CH3_D

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

Functions

void **EnableInput_CH1_E_CH2_E_CH3_D ()**

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D ()**

Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel CH1 and CH2 for STSPIN230.

Enable Input channel for STSPIN230.

Return values

None

Function Documentation

void EnableInput_CH1_E_CH2_E_CH3_D (void)

Definition at line [96](#) of file [STSPIN230.c](#).

References [STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#).

void MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D (void)

Enable Input channel CH1 and CH2 for STSPIN230.

Return values

None

Definition at line [273](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_E_CI](#)

Referenced by [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

EnableInput_CH1_E_CH2_D_CH3_E

[DRIVERS »](#) [BSP »](#) [COMPONENTS »](#)[STSPIN230MIDDLEWARES »](#)[MC_6-STEP_LIB »](#) |[stm32F401_nucleo_ihm11m1](#)

Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

Functions

void **EnableInput_CH1_E_CH2_D_CH3_E ()**

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E ()**

Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel CH1 and CH3 for STSPIN230.

Enable Input channel for STSPIN230.

Return values

None

Function Documentation

void EnableInput_CH1_E_CH2_D_CH3_E (void)

Definition at line [112](#) of file [STSPIN230.c](#).

References [STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#).

void MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E (void)

Enable Input channel CH1 and CH3 for STSPIN230.

Return values

None

Definition at line [291](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_D_CI](#)

Referenced by [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

EnableInput_CH1_D_CH2_E_CH3_E

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »
[STSPIN230MIDDLEWARES](#) »
[MC_6-STEP_LIB](#) » |
[stm32F401_nucleo_ihm11m1](#)

Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

Functions

void **EnableInput_CH1_D_CH2_E_CH3_E ()**

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E ()**

Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel CH2 and CH3 for STSPIN230.

Enable Input channel for STSPIN230.

Return values

None

Function Documentation

void EnableInput_CH1_D_CH2_E_CH3_E (void)

Definition at line [127](#) of file [STSPIN230.c](#).

References [STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#).

void MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [309](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_D_CH2_E_CI](#)

Referenced by [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Functions](#)

DisableInput_CH1_D_CH2_D_CH3_D

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

Functions

void **DisableInput_CH1_D_CH2_D_CH3_D ()**

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D ()**

Enable Input channel CH2 and CH3 for STSPIN230. More...

Detailed Description

Enable Input channel CH2 and CH3 for STSPIN230.

Disable All Input channels for STSPIN230.

Return values

None

Function Documentation

void DisableInput_CH1_D_CH2_D_CH3_D (void)

Definition at line [143](#) of file [STSPIN230.c](#).

References [STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#).

void MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [327](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::DisableInput_CH1_D_CH2_D_C](#)

Referenced by [MC_StopMotor\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Start_PWM_driving

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »
[STSPIN230MIDDLEWARES](#) »
[MC_6-STEP_LIB](#) » |
[stm32F401_nucleo_ihm11m1](#)

Enable PWM channels for STSPIN230. [More...](#)

Functions

void **Start_PWM_driving ()**

void **MC_SixStep_Start_PWM_driving ()**

Enable PWM channels for STSPIN230. [More...](#)

Detailed Description

Enable PWM channels for STSPIN230.

Enable the PWM generation on Input channels.

Return values

None

Function Documentation

void MC_SixStep_Start_PWM_driving (void)

Enable PWM channels for STSPIN230.

Return values

None

Definition at line **345** of file **stm32F401_nucleo_ihm11m1.c**.

References

STSPIN230_MotorDriver_TypeDef::Start_PWM_driving.

Referenced by **MC_SixStep_NEXT_step()**.

void Start_PWM_driving (void)

Definition at line **158** of file **STSPIN230.c**.

References **STSPIN230_Start_PWM_driving()**.

Referenced by **STSPIN230_Current_Reference_Start()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Stop_PWM_driving

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Disable PWM channels for STSPIN230. More...

Functions

void **Stop_PWM_driving ()**

void **MC_SixStep_Stop_PWM_driving ()**

Disable PWM channels for STSPIN230. More...

Detailed Description

Disable PWM channels for STSPIN230.

Disable the PWM generation on Input channels.

Return values

None

Function Documentation

void MC_SixStep_Stop_PWM_driving (void)

Disable PWM channels for STSPIN230.

Return values

None

Definition at line **363** of file **stm32F401_nucleo_ihm11m1.c**.

References

STSPIN230_MotorDriver_TypeDef::Stop_PWM_driving.

Referenced by **MC_StopMotor()**.

void Stop_PWM_driving (void)

Definition at line **173** of file **STSPIN230.c**.

References **STSPIN230_Stop_PWM_driving()**.

Referenced by **STSPIN230_Current_Reference_Stop()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Functions](#)

HF_TIMx_SetDutyCycle_CH1

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Set the Duty Cycle value for CH1. More...

Functions

void **HF_TIMx_SetDutyCycle_CH1** (uint16_t CCR_value)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1** (uint16_t
CCR_value)

Set the Duty Cycle value for CH1. More...

Detailed Description

Set the Duty Cycle value for CH1.

Return values

None

Function Documentation

void HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)

Definition at line [188](#) of file [STSPIN230.c](#).

References [STSPIN230_HF_TIMx_SetDutyCycle_CH1\(\)](#).

**void
MC_SixStep_HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)**

Set the Duty Cycle value for CH1.

Return values

None

Definition at line [381](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH1](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Functions](#)

HF_TIMx_SetDutyCycle_CH2

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Set the Duty Cycle value for CH2. More...

Functions

```
void HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)
```

```
void MC_SixStep_HF_TIMx_SetDutyCycle_CH2 (uint16_t  
      CCR_value)
```

Set the Duty Cycle value for CH2. More...

Detailed Description

Set the Duty Cycle value for CH2.

Return values

None

Function Documentation

void HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)

Definition at line [204](#) of file [STSPIN230.c](#).

References [STSPIN230_HF_TIMx_SetDutyCycle_CH2\(\)](#).

**void
MC_SixStep_HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)**

Set the Duty Cycle value for CH2.

Return values

None

Definition at line [400](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH2](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Functions](#)

HF_TIMx_SetDutyCycle_CH3

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Set the Duty Cycle value for CH3. More...

Functions

void **HF_TIMx_SetDutyCycle_CH3** (uint16_t CCR_value)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3** (uint16_t
CCR_value)

Set the Duty Cycle value for CH3. More...

Detailed Description

Set the Duty Cycle value for CH3.

Return values

None

Function Documentation

void HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)

Definition at line [219](#) of file [STSPIN230.c](#).

References [STSPIN230_HF_TIMx_SetDutyCycle_CH3\(\)](#).

**void
MC_SixStep_HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)**

Set the Duty Cycle value for CH3.

Return values

None

Definition at line [423](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH3](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Current_Reference_Start

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Enable the Current Reference generation. [More...](#)

Functions

void **Current_Reference_Start ()**

void **MC_SixStep_Current_Reference_Start ()**

Enable the Current Reference generation. [More...](#)

Detailed Description

Enable the Current Reference generation.

Return values

None

Function Documentation

void Current_Reference_Start (void)

Definition at line [237](#) of file [STSPIN230.c](#).

References [STSPIN230_Current_Reference_Start\(\)](#).

void MC_SixStep_Current_Reference_Start (void)

Enable the Current Reference generation.

Return values

None

Definition at line [441](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Start](#).

Referenced by [MC_SixStep_RESET\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Current_Reference_Stop

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Disable the Current Reference generation. [More...](#)

Functions

void **Current_Reference_Stop ()**

void **MC_SixStep_Current_Reference_Stop ()**

Disable the Current Reference generation. [More...](#)

Detailed Description

Disable the Current Reference generation.

Return values

None

Function Documentation

void Current_Reference_Stop (void)

Definition at line [253](#) of file [STSPIN230.c](#).

References [STSPIN230_Current_Reference_Stop\(\)](#).

void MC_SixStep_Current_Reference_Stop (void)

Disable the Current Reference generation.

Return values

None

Definition at line [460](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Stop](#).

Referenced by [MC_StopMotor\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Current_Reference_Setvalue

[DRIVERS](#) » [BSP](#) » [COMPONENTS](#) »

[STSPIN230MIDDLEWARES](#) »

[MC_6-STEP_LIB](#) » |

[stm32F401_nucleo_ihm11m1](#)

Set the value for Current Reference. [More...](#)

Functions

void **Current_Reference_Setvalue** (uint16_t lref)

void **MC_SixStep_Current_Reference_Setvalue** (uint16_t lref)

Set the value for Current Reference. More...

Detailed Description

Set the value for Current Reference.

Return values

None

Function Documentation

void Current_Reference_Setvalue (uint16_t Iref)

Definition at line **269** of file [STSPIN230.c](#).

References [STSPIN230_Current_Reference_Setvalue\(\)](#).

void MC_SixStep_Current_Reference_Setvalue (uint16_t Iref)

Set the value for Current Reference.

Return values

None

Definition at line **480** of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Setvalue](#).

Referenced by [MC_SixStep_INIT\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), [MC_SixStep_RESET\(\)](#), and
[MC_Task_Speed\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

[Modules](#) | [Variables](#)

X-NUCLEO-IHM11M1

[DRIVERS](#) » [BSP](#)

X-Nucleo expansion board. More...

Modules

STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D

Enable Input channel for STSPIN230.

STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E

Enable Input channel for STSPIN230.

STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E

Enable Input channel for STSPIN230.

STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D

Disable All Input channels for STSPIN230.

STSPIN230_Start_PWM_driving

Enable the PWM generation on Input channels for STSPIN230.

STSPIN230_Stop_PWM_driving

Disable the PWM generation on Input channels for STSPIN230.

STSPIN230_HF_TIMx_SetDutyCycle_CH1

Set the Duty Cycle value for CH1 for STSPIN230.

STSPIN230_HF_TIMx_SetDutyCycle_CH2

Set the Duty Cycle value for CH2 for STSPIN230.

STSPIN230_HF_TIMx_SetDutyCycle_CH3

Set the Duty Cycle value for CH3 for STSPIN230.

STSPIN230_Current_Reference_Start

Enable the Current Reference generation for STSPIN230.

STSPIN230_Current_Reference_Stop

Disable the Current Reference generation for STSPIN230.

STSPIN230_Current_Reference_Setvalue

Set the value for Current Reference for STSPIN230.

BSP_X_NUCLEOFAULT_LED_ON

Turns selected LED On.

BSP_X_NUCLEOFAULT_LED_OFF

Turns selected LED Off.

Variables

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

Detailed Description

X-Nucleo expansion board.

Variable Documentation

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

Main SixStep structure

Definition at line **73** of file **6Step_Lib.c**.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Enable Input channel for STSPIN230. [More...](#)

Functions

void **STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D ()**

Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel for STSPIN230.

Function Documentation

void STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D (void)

Enable Input channel CH1 and CH2 for STSPIN230.

Return values

None

Definition at line **79** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **EnableInput_CH1_E_CH2_E_CH3_D()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Enable Input channel for STSPIN230. [More...](#)

Functions

void **STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E ()**

Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel for STSPIN230.

Function Documentation

void STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E (void)

Enable Input channel CH1 and CH3 for STSPIN230.

Return values

None

Definition at line **104** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **EnableInput_CH1_E_CH2_D_CH3_E()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Enable Input channel for STSPIN230. [More...](#)

Functions

void **STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E ()**

Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

Detailed Description

Enable Input channel for STSPIN230.

Function Documentation

void STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line **129** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **EnableInput_CH1_D_CH2_E_CH3_E()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_DisableInput_CH1_D_CH2_D_CH3_I

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Disable All Input channels for STSPIN230. [More...](#)

Functions

void **STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D ()**

Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

Detailed Description

Disable All Input channels for STSPIN230.

Function Documentation

void STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line **154** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **DisableInput_CH1_D_CH2_D_CH3_D()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_Start_PWM_driving

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Enable the PWM generation on Input channels for STSPIN230. [More...](#)

Functions

void **STSPIN230_Start_PWM_driving ()**

Enable PWM channels for STSPIN230. [More...](#)

Detailed Description

Enable the PWM generation on Input channels for STSPIN230.

Function Documentation

void STSPIN230_Start_PWM_driving (void)

Enable PWM channels for STSPIN230.

Return values

None

Definition at line **179** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **Start_PWM_driving()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_Stop_PWM_driving

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Disable the PWM generation on Input channels for STSPIN230.

[More...](#)

Functions

void **STSPIN230_Stop_PWM_driving ()**

Disable PWM channels for STSPIN230. More...

Detailed Description

Disable the PWM generation on Input channels for STSPIN230.

Function Documentation

void STSPIN230_Stop_PWM_driving (void)

Disable PWM channels for STSPIN230.

Return values

None

Definition at line **204** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**, **HF_TIMx_CH1**, **HF_TIMx_CH2**, and
HF_TIMx_CH3.

Referenced by **Stop_PWM_driving()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_HF_TIMx_SetDutyCycle_CH1

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Set the Duty Cycle value for CH1 for STSPIN230. [More...](#)

Functions

```
void STSPIN230_HF_TIMx_SetDutyCycle_CH1 (uint16_t  
    CCR_value)  
Set the Duty Cycle value for CH1. More...
```

Detailed Description

Set the Duty Cycle value for CH1 for STSPIN230.

Function Documentation

void

STSPIN230_HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)

Set the Duty Cycle value for CH1.

Return values

None

Definition at line **230** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**.

Referenced by **HF_TIMx_SetDutyCycle_CH1()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_HF_TIMx_SetDutyCycle_CH2

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Set the Duty Cycle value for CH2 for STSPIN230. [More...](#)

Functions

```
void STSPIN230_HF_TIMx_SetDutyCycle_CH2 (uint16_t  
    CCR_value)  
Set the Duty Cycle value for CH2. More...
```

Detailed Description

Set the Duty Cycle value for CH2 for STSPIN230.

Function Documentation

void

STSPIN230_HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)

Set the Duty Cycle value for CH2.

Return values

None

Definition at line **250** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**.

Referenced by **HF_TIMx_SetDutyCycle_CH2()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_HF_TIMx_SetDutyCycle_CH3

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Set the Duty Cycle value for CH3 for STSPIN230. [More...](#)

Functions

```
void STSPIN230_HF_TIMx_SetDutyCycle_CH3 (uint16_t  
    CCR_value)  
Set the Duty Cycle value for CH3. More...
```

Detailed Description

Set the Duty Cycle value for CH3 for STSPIN230.

Function Documentation

void

STSPIN230_HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)

Set the Duty Cycle value for CH3.

Return values

None

Definition at line **274** of file **X-NUCLEO-IHM11M1.c**.

References **HF_TIMx**.

Referenced by **HF_TIMx_SetDutyCycle_CH3()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_Current_Reference_Start

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Enable the Current Reference generation for STSPIN230. [More...](#)

Functions

void **STSPIN230_Current_Reference_Start ()**

Enable the Current Reference generation. [More...](#)

Detailed Description

Enable the Current Reference generation for STSPIN230.

Function Documentation

void STSPIN230_Current_Reference_Start(void)

Enable the Current Reference generation.

Return values

None

Definition at line **293** of file **X-NUCLEO-IHM11M1.c**.

References **SIXSTEP_Base_InitTypeDef::pulse_value**,
Start_PWM_driving(), and **STARTUP_DUTY_CYCLE**.

Referenced by **Current_Reference_Start()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_Current_Reference_Stop

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Disable the Current Reference generation for STSPIN230. [More...](#)

Functions

void **STSPIN230_Current_Reference_Stop ()**

Disable the Current Reference generation. [More...](#)

Detailed Description

Disable the Current Reference generation for STSPIN230.

Function Documentation

void STSPIN230_Current_Reference_Stop (void)

Disable the Current Reference generation.

Return values

None

Definition at line **313** of file **X-NUCLEO-IHM11M1.c**.

References **SIXSTEP_Base_InitTypeDef::pulse_value**,
STARTUP_DUTY_CYCLE, and **Stop_PWM_driving()**.

Referenced by **Current_Reference_Stop()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STSPIN230_Current_Reference_Setvalue

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Set the value for Current Reference for STSPIN230. [More...](#)

Functions

void **STSPIN230_Current_Reference_Setvalue** (uint16_t lref)

Set the value for Current Reference. More...

Detailed Description

Set the value for Current Reference for STSPIN230.

Function Documentation

void STSPIN230_Current_Reference_Setvalue (uint16_t *Iref*)

Set the value for Current Reference.

Return values

None

Definition at line **334** of file **X-NUCLEO-IHM11M1.c**.

References **SIXSTEP_Base_InitTypeDef::pulse_value**.

Referenced by **Current_Reference_Setvalue()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

BSP_X_NUCLEOFAULT_LED_ON

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Turns selected LED On. [More...](#)

Functions

```
void BSP_X_NUCLEOFAULT_LED_ON()
```

Detailed Description

Turns selected LED On.

Return values

None

Function Documentation

void BSP_X_NUCLEOFAULTLEDON(void)

Definition at line **351** of file **X-NUCLEO-IHM11M1.c**.

Referenced by **MC_StartMotor()**.

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

BSP_X_NUCLEOFAULT_LED_OFF

[DRIVERS](#) » [BSP](#) » [X-NUCLEO-IHM11M1](#)

Turns selected LED Off. [More...](#)

Functions

```
void BSP_X_NUCLEOFAULT_LED_OFF()
```

Detailed Description

Turns selected LED Off.

Return values

None

Function Documentation

void BSP_X_NUCLEOFAULTLED OFF(void)

Definition at line **364** of file **X-NUCLEO-IHM11M1.c**.

Referenced by **MC_StopMotor()**.

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

MIDDLEWARES

Middlewares Layer. [More...](#)

Modules

MC_6-STEP_LIB

Motor Control driver.

UART_UI

Serial communication through PC serial terminal.

Detailed Description

Middlewares Layer.

Main functions for 6-Step algorithm

The main function are the following:

- 1) MC_SixStep_TABLE(...) -> Set the peripherals (TIMx, GPIO etc.) for each step
- 2) **MC_SixStep_ARR_step()** -> Generate the ARR value for Low Frequency TIM during start-up
- 3) **MC_SixStep_INIT()** -> Init the main variables for motor driving from **MC_SixStep_param.h**
- 4) **MC_SixStep_RESET()** -> Reset all variables used for 6Step control algorithm
- 5) **MC_SixStep_Ramp_Motor_calc()** -> Calculate the acceleration profile step by step for motor during start-up
- 6) **MC_SixStep_NEXT_step()**-> Generate the next step number according with the direction (CW or CCW)
- 7) **MC_Task_Speed()** -> Speed Loop with PI regulator
- 8) MC_Set_Speed(...) -> Set the new motor speed value
- 9) **MC_StartMotor()** -> Start the Motor
- 10)**MC_StopMotor()** -> Stop the Motor



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

[Modules](#) | [Functions](#) | [Variables](#)

MC_6-STEP_LIB

MIDDLEWARES

Motor Control driver. More...

Modules

Exported_types

Exported_function_6StepLib

MC_SixStep_TABLE

Set the peripherals (TIMx, GPIO etc.) for each step.

MC_SixStep_NEXT_step

Generate the next step number according with the direction (CW or CCW)

MC_SixStep_RESET

Reset all variables used for 6Step control algorithm.

MC_SixStep_Ramp_Motor_calc

Calculate the acceleration profile step by step for motor during start-up.

MC_SixStep_ARR_step

Generate the ARR value for Low Frequency TIM during start-up.

MC_SixStep_Alignment

Generate the motor alignment.

MC_SixStep_Speed_Val_target_potentiometer

Calculate the Motor Speed validation threshold according with the potentiometer value.

MC_SixStep_Speed_Potentiometer

Calculate the potentiometer value to set the Motor Speed.

MC_Set_PI_param

Set all parameters for PI regulator.

MC_PI_Controller

Compute the PI output for the Current Reference.

MC_Task_Speed

Main task: Speed Loop with PI regulator.

MC_Set_Speed

Set the new motor speed value.

MC_Bemf_Delay

Take the delay time after each new 6-step commutation.

MC_StartMotor

Start the Motor.

MC_StopMotor

Stop the Motor.

MC_GetElSpeedHz

Get the Elelctrical Motor Speed from ARR value of LF TIM.

MC_GetMechSpeedRPM

Get the Mechanical Motor Speed (RPM)

MC_SixStep_Init_main_data

Init the main variables for motor driving from

MC_SixStep_param.h.

MC_SixStep_INIT

Initialitation function for SixStep library.

MC_TIMx_SixStep_timebase

Low Frequency Timer Callback - Call the next step and request the filtered speed value.

MC_Speed_Filter

Calculate the speed filtered.

MC_Potentiometer_filter

Calculate the filtered potentiometer value.

MC_SysTick_SixStep_MediumFrequencyTask

Systick Callback - Call the Speed loop.

MC_SixStep_ARR_Bemf

Calculate the new Autoreload value (ARR) for Low Frequency timer.

MC_ADCx_SixStep_Bemf

Compute the zero crossing detection.

MC_EXT_button_SixStep

GPIO EXT Callback - Start or Stop the motor through the Blue push button on STM32Nucleo.

Main_Motor_parameters

All motor parameters for 6Step driving.

stm32F401_nucleo_ihm11m1

Interface file for STM32F401 and Motor Control Library configuration.

Functions

uint64_t **MCM_Sqrt** (uint64_t wInput)

It calculates the square root of a non-negative s64. [More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D** (void)

Enable Input channel CH1 and CH2 for STSPIN230.

[More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E** (void)

Enable Input channel CH1 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E** (void)

Enable Input channel CH2 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D** (void)

Enable Input channel CH2 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_Start_PWM_driving** (void)

Enable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_Stop_PWM_driving** (void)

Disable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1** (uint16_t)

Set the Duty Cycle value for CH1. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH2** (uint16_t)

Set the Duty Cycle value for CH2. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3** (uint16_t)

Set the Duty Cycle value for CH3. [More...](#)

void **MC_SixStep_Current_Reference_Start** (void)
Enable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Setvalue** (uint16_t)
Set the value for Current Reference. [More...](#)

void **MC_UI_INIT** (void)

void **UART_Set_Value** (void)

void **UART_Communication_Task** (void)

void **CMD_Parser** (char *pCommandString)

void **HAL_IncTick** (void)
This function is called to increment a global variable
"uwTick" used as application time base. [More...](#)

uint32_t **HAL_GetTick** (void)
Provides a tick value in millisecond. [More...](#)

Variables

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

SIXSTEP_PI_PARAM_InitTypeDef_t PI_parameters

uint16_t Rotor_poles_pairs

uint32_t mech_accel_hz = 0

uint32_t constant_k = 0

uint32_t Time_vector_tmp = 0

uint32_t Time_vector_prev_tmp = 0

uint32_t T_single_step = 0

uint32_t T_single_step_first_value =
0

int32_t delta = 0

uint16_t index_array = 1

int16_t speed_tmp_array
[FILTER_DEEP]

uint16_t speed_tmp_buffer
[FILTER_DEEP]

uint16_t HFBuffer [HFBUFFERSIZE]

uint16_t HFBufferIndex = 0

uint8_t array_completed = FALSE

```
    uint8_t buffer_completed = FALSE
```

```
    uint8_t UART_FLAG_RECEIVE =  
        FALSE
```

```
    uint32_t ARR_LF = 0
```

```
    int32_t Mech_Speed_RPM = 0
```

```
    int32_t El_Speed_Hz = 0
```

```
    uint16_t index_adc_chn = 0
```

```
    uint16_t index_motor_run = 0
```

```
    uint16_t test_motor_run = 1
```

```
    uint8_t Enable_start_button = TRUE
```

```
    uint16_t index_ARR_step = 1
```

```
    uint32_t n_zcr_startup = 0
```

```
    uint16_t index_startup_motor = 1
```

```
    uint16_t target_speed =  
        TARGET_SPEED
```

```
    uint16_t shift_n_sqrt = 14
```

```
    uint16_t cnt_bemf_event = 0
```

```
    uint8_t startup_bemf_failure = 0
```

```
    uint8_t speed_fdbk_error = 0
```

__IO uint32_t uwTick = 0

uint8_t dac_status = DAC_ENABLE

uint16_t index_align = 1

int32_t speed_sum_sp_filt = 0

int32_t speed_sum_pot_filt = 0

uint16_t index_pot_filt = 1

int16_t potent_filtered = 0

uint32_t Tick_cnt = 0

uint32_t counter.ARR_Bemf = 0

uint64_t constant_multiplier_tmp = 0

Detailed Description

Motor Control driver.

Function Documentation

void CMD_Parser (char * pCommandString)

uint32_t HAL_GetTick (void)

Povides a tick value in millisecond.

Note

The function is declared as __Weak to be overwritten in case of other implementations in user file.

Return values

tick value

Definition at line [1618](#) of file [6Step.Lib.c](#).

References [uwTick](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

void HAL_IncTick (void)

This function is called to increment a global variable "uwTick" used as application time base.

Note

In the default implementation, this variable is incremented each 1ms in Systick ISR.

This function is declared as __weak to be overwritten in case of other implementations in user file.

Return values

None

Definition at line [1607](#) of file [6Step_Lib.c](#).

References [uwTick](#).

Referenced by [SysTick_Handler\(\)](#).

void MC_SixStep_Current_Reference_Setvalue (uint16_t lref)

Set the value for Current Reference.

Return values

None

Definition at line [480](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_INIT\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), [MC_SixStep_RESET\(\)](#), and
[MC_Task_Speed\(\)](#).

void MC_SixStep_Current_Reference_Start ()

Enable the Current Reference generation.

Return values

None

Definition at line [441](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_RESET\(\)](#).

void MC_SixStep_Current_Reference_Stop ()

Disable the Current Reference generation.

Return values

None

Definition at line [460](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_StopMotor\(\)](#).

void MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D()

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [327](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_StopMotor\(\)](#).

void MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E()

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [309](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_TABLE\(\)](#).

void MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E()

Enable Input channel CH1 and CH3 for STSPIN230.

Return values

None

Definition at line [291](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_TABLE\(\)](#).

void MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D()

Enable Input channel CH1 and CH2 for STSPIN230.

Return values

None

Definition at line [273](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_TABLE\(\)](#).

void MC_SixStep_HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)

Set the Duty Cycle value for CH1.

Return values

None

Definition at line [381](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).

void MC_SixStep_HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)

Set the Duty Cycle value for CH2.

Return values

None

Definition at line [400](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).

```
void  
MC_SixStep_HF_TIMx_SetDutyCycle_CH3 ( uint16_t CCR_value )
```

Set the Duty Cycle value for CH3.

Return values

None

Definition at line [423](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).

```
void MC_SixStep_Start_PWM_driving ( )
```

Enable PWM channels for STSPIN230.

Return values

None

Definition at line [345](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#).

```
void MC_SixStep_Stop_PWM_driving ( )
```

Disable PWM channels for STSPIN230.

Return values

None

Definition at line [363](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_StopMotor\(\)](#).

```
void MC_UI_INIT( void )
```

Referenced by [MC_SixStep_INIT\(\)](#).

```
uint64_t MCM_Sqrt( uint64_t wInput )
```

It calculates the square root of a non-negative s64.

It returns 0 for negative s64.

Parameters

Input uint64_t number

Return values

int32_t Square root of Input (0 if Input<0)

Definition at line [533](#) of file [6Step.Lib.c](#).

References [shift_n_sqrt](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#).

```
void UART_Communication_Task( void )
```

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
void UART_Set_Value( void )
```

Variable Documentation

uint32_t ARR_LF = 0

Autoreload LF TIM variable

Definition at line [93](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint8_t array_completed = FALSE

Speed filter variable

Definition at line [90](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_Speed_Filter\(\)](#).

uint8_t buffer_completed = FALSE

Potentiometer filter variable

Definition at line [91](#) of file [6Step.Lib.c](#).

Referenced by [MC_Potentiometer_filter\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t cnt_bemf_event = 0

Definition at line [105](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint32_t constant_k = 0

1/3*mech_accel_hz

Definition at line [79](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint64_t constant_multiplier_tmp = 0

Definition at line [117](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint32_t counter_ARR_Bemf = 0

Definition at line [116](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint8_t dac_status = DAC_ENABLE

Definition at line [109](#) of file [6Step.Lib.c](#).

Referenced by [MC_StartMotor\(\)](#), and [MC_Task_Speed\(\)](#).

int32_t delta = 0

Startup variable

Definition at line [84](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

int32_t El_Speed_Hz = 0

Electrical motor speed

Definition at line [95](#) of file [6Step.Lib.c](#).

Referenced by [MC_GetElSpeedHz\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint8_t Enable_start_button = TRUE

Start/stop button filter to avoid double command

Definition at line [99](#) of file [6Step.Lib.c](#).

Referenced by [MC_EXT_button_SixStep\(\)](#),
[MC_SixStep_RESET\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint16_t HFBuffer[HFBUFFERSIZE]

Buffer for Potentiometer Value Filtering at the High-Frequency ADC conversion

Definition at line [88](#) of file [6Step.Lib.c](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_RESET\(\)](#), and [MC_SixStep_Speed_Potentiometer\(\)](#).

uint16_t HFBufferIndex = 0

High-Frequency Buffer Index

Definition at line [89](#) of file [6Step_Lib.c](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t index_adc_chn = 0

Index of ADC channel selector for measuring

Definition at line [96](#) of file [6Step_Lib.c](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t index_align = 1

Definition at line [110](#) of file [6Step_Lib.c](#).

Referenced by [MC_SixStep_Alignment\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t index.ARR_step = 1

Definition at line [100](#) of file [6Step_Lib.c](#).

Referenced by [MC_SixStep_ARR_step\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t index_array = 1

Speed filter variable

Definition at line [85](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_Speed_Filter\(\)](#).

uint16_t index_motor_run = 0

Tmp variable for DEMO mode

Definition at line [97](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint16_t index_pot_filt = 1

Definition at line [113](#) of file [6Step.Lib.c](#).

Referenced by [MC_Potentiometer_filter\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t index_startup_motor = 1

Definition at line [102](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_ARR_step\(\)](#), [MC_SixStep_Ramp_Motor_calc\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint32_t mech_accel_hz = 0

Hz – Mechanical acceleration rate

Definition at line [78](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

int32_t Mech_Speed_RPM = 0

Mechanical motor speed

Definition at line [94](#) of file [6Step.Lib.c](#).

Referenced by [MC_GetMechSpeedRPM\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint32_t n_zcr_startup = 0

Definition at line [101](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

SIXSTEP_PI_PARAM_InitTypeDef_t PI_parameters

SixStep PI regulator structure

Definition at line [74](#) of file [6Step.Lib.c](#).

int16_t potent_filtered = 0

Definition at line [114](#) of file [6Step.Lib.c](#).

Referenced by [MC_Potentiometer_filter\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint16_t Rotor_poles_pairs

Number of pole pairs of the motor

Definition at line [77](#) of file [6Step.Lib.c](#).

Referenced by [MC_GetMechSpeedRPM\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint16_t shift_n_sqrt = 14

Definition at line [104](#) of file [6Step.Lib.c](#).

Referenced by [MCM_Sqrt\(\)](#).

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

Main SixStep structure

Definition at line [73](#) of file [6Step.Lib.c](#).

uint8_t speed_fdbk_error = 0

Definition at line [107](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

int32_t speed_sum_pot_filt = 0

Definition at line [112](#) of file [6Step.Lib.c](#).

Referenced by [MC_Potentiometer_filter\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
int32_t speed_sum_sp_filt = 0
```

Definition at line [111](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_Speed_Filter\(\)](#).

```
int16_t speed_tmp_array[FILTER_DEEP]
```

Speed filter variable

Definition at line [86](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_Speed_Filter\(\)](#).

```
uint16_t speed_tmp_buffer[FILTER_DEEP]
```

Potentiometer filter variable

Definition at line [87](#) of file [6Step.Lib.c](#).

Referenced by [MC_Potentiometer_filter\(\)](#), and [MC_SixStep_RESET\(\)](#).

```
uint8_t startup_bemf_failure = 0
```

Definition at line [106](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
uint32_t T_single_step = 0
```

Startup variable

Definition at line [82](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

```
uint32_t T_single_step_first_value = 0
```

Startup variable

Definition at line [83](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

```
uint16_t target_speed = TARGET_SPEED
```

Target speed for closed loop control

Definition at line [103](#) of file [6Step.Lib.c](#).

Referenced by [MC_Set_PI_param\(\)](#), [MC_SixStep_RESET\(\)](#), [MC_SixStep_Speed_Val_target_potentiometer\(\)](#), and [MC_Task_Speed\(\)](#).

```
uint16_t test_motor_run = 1
```

Tmp variable for DEMO mode

Definition at line [98](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
uint32_t Tick_cnt = 0
```

Definition at line [115](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_RESET\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint32_t Time_vector_prev_tmp = 0

Startup variable

Definition at line [81](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint32_t Time_vector_tmp = 0

Startup variable

Definition at line [80](#) of file [6Step.Lib.c](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint8_t UART_FLAG_RECEIVE = FALSE

UART communication flag

Definition at line [92](#) of file [6Step.Lib.c](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

__IO uint32_t uwTick = 0

Tick counter - 1msec updated

Definition at line **108** of file **6Step_Lib.c**.

Referenced by **HAL_GetTick()**, **HAL_IncTick()**,
MC_SixStep_RESET(), **MC_StartMotor()**, and **MC_StopMotor()**.

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

[Data Structures](#) | [Typedefs](#) | [Enumerations](#)

Exported_types

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Data Structures

struct **SIXSTEP_Base_InitTypeDef**

Six Step parameters. More...

struct **SIXSTEP_PI_PARAM_InitTypeDef_t**

Six PI regulator parameters. More...

TypeDefs

```
typedef struct SIXSTEP_PI_PARAM_InitTypeDef_t * SIXSTEP_pi_PA
```

Enumerations

```
enum SIXSTEP_Base_SystStatus_t {
    IDLE, STARTUP, VALIDATION, STOP,
    START, RUN, ALIGNMENT, SPEEDFBKERROR,
    OVERCURRENT, STARTUP_FAILURE,
    STARTUP_BEMF_FAILURE
}
Six Step parameters. More...
```

Detailed Description

Typedef Documentation

```
typedef struct SIXSTEP_PI_PARAM_InitTypeDef_t *  
SIXSTEP_pi_PARAM_InitTypeDef_t
```

PI Data Structure

Enumeration Type Documentation

enum SIXSTEP_Base_SystStatus_t

Six Step parameters.

Enumerator
IDLE
STARTUP
VALIDATION
STOP
START
RUN
ALIGNMENT
SPEEDFBKERROR
OVERCURRENT
STARTUP_FAILURE
STARTUP_BEMF_FAILURE

Definition at line **68** of file **6Step_Lib.h**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files			
Data Structures	Data Structure Index	Data Fields	Data Fields			
SIXSTEP_Base_InitTypeDef Struct Reference						
MIDDLEWARES » MC_6-STEP_LIB »						
Exported_types						

Six Step parameters. More...

#include <6Step_Lib.h>

Data Fields

uint32_t **LF_TIMx_PSC**

uint32_t **LF_TIMx_ARR**

uint32_t **HF_TIMx_PSC**

uint32_t **HF_TIMx_ARR**

uint32_t **HF_TIMx_CCR**

uint8_t **step_position**

SIXSTEP_Base_SystStatus_t STATUS

uint8_t **status_prev**

uint16_t **pulse_value**

uint16_t **ARR_value**

uint32_t **Regular_channel** [4]

uint32_t **CurrentRegular_BEMF_ch**

uint32_t **prescaler_value**

uint16_t **numberofitemArr**

uint32_t **ADC_BUFFER** [4]

uint32_t **ADC_SEQ_CHANNEL** [4]

uint32_t **ADC_Regular_Buffer** [5]

`uint16_t ADC_BEMF_threshold_UP`

`uint16_t ADC_BEMF_threshold_DOWN`

`uint16_t demagn_counter`

`uint16_t demagn_value`

`int16_t speed_fdbk`

`int16_t speed_fdbk_filtered`

`int16_t filter_depth`

`uint16_t Current_Reference`

`uint16_t Ireference`

`int32_t Integral_Term_sum`

`uint8_t CMD`

`uint8_t ALIGN_OK`

`uint8_t ALIGNMENT`

`uint8_t bemf_state_1`

`uint8_t bemf_state_2`

`uint8_t bemf_state_3`

`uint8_t bemf_state_4`

`uint8_t bemf_state_5`

`uint8_t bemf_state_6`

`uint16_t Speed_Loop_Time`

`uint16_t Speed_Ref_filtered`

`uint16_t RUN_Motor`

`uint8_t ARR_OK`

`uint8_t VALIDATION_OK`

`uint8_t SPEED_VALIDATED`

`uint16_t Speed_target_ramp`

`uint16_t Speed_target_time`

`uint16_t Ramp_Start`

`uint16_t Bemf_delay_start`

`uint16_t MediumFrequencyTask_flag`

`uint32_t SYSCLK_frequency`

`uint32_t Uart_cmd_to_set`

`uint32_t Uart_value_to_set`

`uint8_t Button_ready`

`uint8_t BEMF_OK`

`uint8_t CL_READY`

`uint8_t BEMF_Tdown_count`

`uint16_t IREFERENCE`

`uint16_t NUMPOLESPAIRS`

`uint32_t ACCEL`

`uint16_t KP`

`uint16_t KI`

`uint8_t CW_CCW`

`uint8_t Potentiometer`

Detailed Description

Six Step parameters.

Definition at line **94** of file **6Step_Lib.h**.

Field Documentation

uint32_t SIXSTEP_Base_InitTypeDef::ACCEL

Acceleration start-up parameter

Definition at line [152](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_step\(\)](#),
[MC_SixStep_Init_main_data\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

**uint16_t
SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_DOWN**

Voltage threshold for BEMF detection in down direction

Definition at line [114](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_UP

Voltage threshold for BEMF detection in up direction

Definition at line [113](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::ADC_BUFFER[4]

Buffer for ADC regular channel

Definition at line [110](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::ADC-Regular_Buffer[5]

Buffer for ADC regular channel

Definition at line [112](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_Speed_Val_target_potentiometer\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::ADC_SEQ_CHANNEL[4]

Buffer for ADC regular channel

Definition at line [111](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::ALIGN_OK

Flag control for Motor Alignment

Definition at line [124](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_ARR_step\(\)](#), [MC_SixStep_NEXT_step\(\)](#),
[MC_SixStep_RESET\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::ALIGNMENT

Flag control for Motor Alignment ongoing

Definition at line [125](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_step\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::ARR_OK

ARR flag control for Accell status

Definition at line [135](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_step\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_TIMx_SixStep_timebase\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::ARR_value

ARR vector for Accell compute

Definition at line [105](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_ARR_step\(\)](#), and [MC_SixStep_Ramp_Motor_calc\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Bemf_delay_start

Bemf variable

Definition at line [141](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::BEMF_OK

Definition at line [147](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_Task_Speed\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_1

Bemf variable

Definition at line [126](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_2

Bemf variable

Definition at line [127](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_3

Bemf variable

Definition at line [128](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_4

Bemf variable

Definition at line [129](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_5

Bemf variable

Definition at line [130](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::bemf_state_6

Bemf variable

Definition at line [131](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::BEMF_Tdown_count

BEMF Consecutive Threshold Falling Crossings Counter

Definition at line [149](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::Button_ready

Definition at line [146](#) of file [6Step_Lib.h](#).

Referenced by [MC_EXT_button_SixStep\(\)](#), and
[MC_SixStep_INIT\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::CL_READY

Definition at line [148](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_Task_Speed\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::CMD

Flag control for Motor Start/Stop

Definition at line [123](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Current_Reference

Current reference for SixStep algorithm

Definition at line [120](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_RESET\(\)](#), and [MC_Task_Speed\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch

ADC regular channel to select

Definition at line [107](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), [MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::CW_CCW

Set the motor direction

Definition at line [155](#) of file [6Step_Lib.h](#).

Referenced by [MC_Set_PI_param\(\)](#), [MC_Set_Speed\(\)](#), and [MC_SixStep_Init_main_data\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::demagn_counter

Demagnetization counter

Definition at line [115](#) of file [6Step_Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::demagn_value

Demagnetization value

Definition at line [116](#) of file [6Step_Lib.h](#).

Referenced by [Bemf_delay_calc\(\)](#), [MC_ADCx_SixStep_Bemf\(\)](#),
and [MC_SixStep_RESET\(\)](#).

int16_t SIXSTEP_Base_InitTypeDef::filter_depth

Filter depth for speed measuring

Definition at line [119](#) of file [6Step_Lib.h](#).

uint32_t SIXSTEP_Base_InitTypeDef::HF_TIMx_ARR

ARR variable for high frequency timer

Definition at line [99](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::HF_TIMx_CCR

CCR variable for high frequency timer

Definition at line [100](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#), and [MC_SixStep_RESET\(\)](#).

`uint32_t SIXSTEP_Base_InitTypeDef::HF_TIMx_PSC`

Prescaler variable for high frequency timer

Definition at line [98](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#), and [MC_SixStep_RESET\(\)](#).

`int32_t SIXSTEP_Base_InitTypeDef::Integral_Term_sum`

Global Integral part for PI

Definition at line [122](#) of file [6Step_Lib.h](#).

Referenced by [MC_PI_Controller\(\)](#), and [MC_SixStep_RESET\(\)](#).

`uint16_t SIXSTEP_Base_InitTypeDef::lreference`

Current reference for SixStep algorithm

Definition at line [121](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#),
[MC_SixStep_Init_main_data\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

`uint16_t SIXSTEP_Base_InitTypeDef::lREFERENCE`

Current reference

Definition at line [150](#) of file [6Step_Lib.h](#).

`uint16_t SIXSTEP_Base_InitTypeDef::KI`

KI parameter for PI regulator

Definition at line [154](#) of file [6Step_Lib.h](#).

Referenced by [MC_Set_PI_param\(\)](#), and
[MC_SixStep_Init_main_data\(\)](#).

`uint16_t SIXSTEP_Base_InitTypeDef::KP`

KP parameter for PI regulator

Definition at line [153](#) of file [6Step_Lib.h](#).

Referenced by [MC_Set_PI_param\(\)](#), and
[MC_SixStep_Init_main_data\(\)](#).

`uint32_t SIXSTEP_Base_InitTypeDef::LF_TIMx_ARR`

ARR variable for low frequency timer

Definition at line [97](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#), and [MC_SixStep_RESET\(\)](#).

`uint32_t SIXSTEP_Base_InitTypeDef::LF_TIMx_PSC`

Prescaler variable for low frequency timer

Definition at line [96](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_INIT\(\)](#), and [MC_SixStep_RESET\(\)](#).

**uint16_t
SIXSTEP_Base_InitTypeDef::MediumFrequencyTask_flag**

Flag for Medium Task Frequency

Definition at line [142](#) of file [6Step_Lib.h](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::numberofitemArr

Number of elements

Definition at line [109](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_step\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::NUMPOLESPAIRS

Number of motor pole pairs

Definition at line [151](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#), and
[MC_SixStep_RESET\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::Potentiometer

Enable/Disable potentiometer for speed control

Definition at line [156](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#), and [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

`uint32_t SIXSTEP_Base_InitTypeDef::prescaler_value`

Prescaler value for low freq timer

Definition at line [108](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_Alignment\(\)](#), and [MC_SixStep_Ramp_Motor_calc\(\)](#).

`uint16_t SIXSTEP_Base_InitTypeDef::pulse_value`

CCR value for SixStep algorithm

Definition at line [104](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_RESET\(\)](#), [MC_SixStep_TABLE\(\)](#), [STSPIN230_Current_Reference_Setvalue\(\)](#), [STSPIN230_Current_Reference_Start\(\)](#), and [STSPIN230_Current_Reference_Stop\(\)](#).

`uint16_t SIXSTEP_Base_InitTypeDef::Ramp_Start`

Ramp time start

Definition at line [140](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

`uint32_t SIXSTEP_Base_InitTypeDef::Regular_channel[4]`

Buffer for ADC regular channel

Definition at line [106](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_SixStep_TABLE\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::RUN_Motor

Flag for Motor status

Definition at line [134](#) of file [6Step.Lib.h](#).

Referenced by [MC_EXT_button_SixStep\(\)](#), [MC_SixStep_RESET\(\)](#), [MC_StartMotor\(\)](#), and [MC_StopMotor\(\)](#).

int16_t SIXSTEP_Base_InitTypeDef::speed_fdbk

Motor speed variable

Definition at line [117](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_Speed_Filter\(\)](#).

int16_t SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered

Filtered Motor speed variable

Definition at line [118](#) of file [6Step.Lib.h](#).

Referenced by [Bemf_delay_calc\(\)](#), [MC_SixStep_RESET\(\)](#), [MC_Speed_Filter\(\)](#), and [MC_Task_Speed\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Speed_Loop_Time

Speed loop variable for timing

Definition at line [132](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_RESET\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered

Filtered Reference Motor Speed variable

Definition at line [133](#) of file [6Step.Lib.h](#).

Referenced by [MC_Set_Speed\(\)](#), [MC_SixStep_RESET\(\)](#), and
[MC_SixStep_Speed_Potentiometer\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Speed_target_ramp

Target Motor Speed

Definition at line [138](#) of file [6Step.Lib.h](#).

Referenced by [MC_Set_Speed\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint16_t SIXSTEP_Base_InitTypeDef::Speed_target_time

Target Motor Ramp time

Definition at line [139](#) of file [6Step.Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::SPEED_VALIDATED

Validation flag for Speed before closed loop control

Definition at line [137](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_Task_Speed\(\)](#).

**SIXSTEP_Base_SystStatus_t
SIXSTEP_Base_InitTypeDef::STATUS**

Status variable for SixStep algorithm

Definition at line [102](#) of file [6Step.Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), [MC_SixStep_Alignment\(\)](#), [MC_SixStep_ARR_step\(\)](#), [MC_SixStep_Ramp_Motor_calc\(\)](#), [MC_StartMotor\(\)](#), [MC_StopMotor\(\)](#), [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#), [MC_Task_Speed\(\)](#), and [TIM1_BRK_TIM9_IRQHandler\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::status_prev

Previous status variable for SixStep algorithm

Definition at line [103](#) of file [6Step.Lib.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint8_t SIXSTEP_Base_InitTypeDef::step_position

Step number variable for SixStep algorithm

Definition at line [101](#) of file [6Step.Lib.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_Alignment\(\)](#), [MC_SixStep_ARR_Bemf\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::SYSCLK_frequency

System clock main frequency

Definition at line [143](#) of file [6Step_Lib.h](#).

Referenced by [MC_GetElSpeedHz\(\)](#),
[MC_SixStep_Ramp_Motor_calc\(\)](#), and [MC_SixStep_RESET\(\)](#).

uint32_t SIXSTEP_Base_InitTypeDef::Uart_cmd_to_set

Definition at line [144](#) of file [6Step_Lib.h](#).

uint32_t SIXSTEP_Base_InitTypeDef::Uart_value_to_set

Definition at line [145](#) of file [6Step_Lib.h](#).

uint8_t SIXSTEP_Base_InitTypeDef::VALIDATION_OK

Validation flag for Closed loop control begin

Definition at line [136](#) of file [6Step_Lib.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#),
[MC_SixStep_ARR_step\(\)](#), [MC_SixStep_NEXT_step\(\)](#),
[MC_SixStep_RESET\(\)](#),
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#), and
[MC_Task_Speed\(\)](#).

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

- **6Step_Lib.h**
-

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
Data Structures	Data Structure Index	Data Fields	
			Data Fields

SIXSTEP_PI_PARAM_InitTypeDef_t Struct Reference

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) » [Exported_types](#)

Six PI regulator parameters. [More...](#)

```
#include <6Step_Lib.h>
```

Data Fields

`int16_t Reference`

`int16_t Kp_Gain`

`int16_t Ki_Gain`

`int16_t Lower_Limit_Output`

`int16_t Upper_Limit_Output`

`int8_t Max_PID_Output`

`int8_t Min_PID_Output`

Detailed Description

Six PI regulator parameters.

Definition at line **170** of file **6Step_Lib.h**.

Field Documentation

int16_t SIXSTEP_PI_PARAM_InitTypeDef_t::Ki_Gain

Ki value for PI regulator

Definition at line [174](#) of file [6Step.Lib.h](#).

Referenced by [MC_PI_Controller\(\)](#), and [MC_Set_PI_param\(\)](#).

int16_t SIXSTEP_PI_PARAM_InitTypeDef_t::Kp_Gain

Kp value for PI regulator

Definition at line [173](#) of file [6Step.Lib.h](#).

Referenced by [MC_PI_Controller\(\)](#), and [MC_Set_PI_param\(\)](#).

int16_t SIXSTEP_PI_PARAM_InitTypeDef_t::Lower_Limit_Output

Min output value for PI regulator

Definition at line [175](#) of file [6Step.Lib.h](#).

Referenced by [MC_PI_Controller\(\)](#), and [MC_Set_PI_param\(\)](#).

int8_t SIXSTEP_PI_PARAM_InitTypeDef_t::Max_PID_Output

Max Saturation indicator flag

Definition at line [177](#) of file [6Step.Lib.h](#).

Referenced by [MC_Set_PI_param\(\)](#).

int8_t SIXSTEP_PI_PARAM_InitTypeDef_t::Min_PID_Output

Min Saturation indicator flag

Definition at line [178](#) of file [6Step_Lib.h](#).

Referenced by [MC_Set_PI_param\(\)](#).

int16_t SIXSTEP_PI_PARAM_InitTypeDef_t::Reference

Refence value for PI regulator

Definition at line [172](#) of file [6Step_Lib.h](#).

Referenced by [Bemf_delay_calc\(\)](#), [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_GetEISpeedHz\(\)](#), [MC_PI_Controller\(\)](#), [MC_Set_PI_param\(\)](#),
[MC_Set_Speed\(\)](#), [MC_SixStep_ARR_step\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#), and
[MC_Task_Speed\(\)](#).

int16_t SIXSTEP_PI_PARAM_InitTypeDef_t::Upper_Limit_Output

Max output value for PI regulator

Definition at line [176](#) of file [6Step_Lib.h](#).

Referenced by [MC_PI_Controller\(\)](#), and [MC_Set_PI_param\(\)](#).

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

- [6Step_Lib.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Exported_function_6StepLib

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Functions

```
void MC_SixStep_INIT (void)
```

```
void MC_SixStep_RESET (void)
```

```
void MC_StartMotor (void)
```

```
void MC_StopMotor (void)
```

```
void MC_Set_Speed (uint16_t)
```

```
void MC_EXT_button_SixStep (void)
```

Detailed Description

Function Documentation

void MC_EXT_button_SixStep (void)

Definition at line [1577](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::Button_ready](#), [Enable_start_button](#), [FALSE](#), [MC_StartMotor\(\)](#), [MC_StopMotor\(\)](#), [SIXSTEP_Base_InitTypeDef::RUN_Motor](#), and [TRUE](#).

Referenced by [HAL_GPIO_EXTI_Callback\(\)](#).

void MC_Set_Speed (uint16_t)

Definition at line [889](#) of file [6Step.Lib.c](#).

References [ADC_SPEED_TH](#), [SIXSTEP_Base_InitTypeDef::CW_CCW](#), [MAX_POT_SPEED](#), [MIN_POT_SPEED](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), [SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered](#), and [SIXSTEP_Base_InitTypeDef::Speed_target_ramp](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

void MC_SixStep_INIT (void)

Definition at line [1087](#) of file [6Step.Lib.c](#).

References `SIXSTEP_Base_InitTypeDef::Button_ready`, `FALSE`, `HF_TIMx`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_ARR`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_CCR`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_PSC`, `SIXSTEP_Base_InitTypeDef::Ireference`, `LF_TIMx`, `SIXSTEP_Base_InitTypeDef::LF_TIMx_ARR`, `SIXSTEP_Base_InitTypeDef::LF_TIMx_PSC`, `MC_SixStep_Current_Reference_Setvalue()`, `MC_SixStep_Init_main_data()`, `MC_SixStep_Nucleo_Init()`, `MC_SixStep_RESET()`, `MC_UI_INIT()`, and `TRUE`.

Referenced by `main()`.

`void MC_SixStep_RESET (void)`

Definition at line 353 of file `6Step_Lib.c`.

References `ADC_Bemf_CH1`, `ADC_Bemf_CH2`, `ADC_Bemf_CH3`, `SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_DOWN`, `SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_UP`, `ADC_CH_1`, `ADC_CH_2`, `ADC_CH_3`, `ADC_CH_4`, `SIXSTEP_Base_InitTypeDef::ADC_SEQ_CHANNEL`, `SIXSTEP_Base_InitTypeDef::ALIGN_OK`, `SIXSTEP_Base_InitTypeDef::ALIGNMENT`, `ARR_LF`, `SIXSTEP_Base_InitTypeDef::ARR_OK`, `array_completed`, `SIXSTEP_Base_InitTypeDef::BEMF_OK`, `SIXSTEP_Base_InitTypeDef::BEMF_Tdown_count`, `BEMF_THRSLD_DOWN`, `BEMF_THRSLD_UP`, `buffer_completed`, `SIXSTEP_Base_InitTypeDef::CL_READY`, `SIXSTEP_Base_InitTypeDef::CMD`, `cnt_bemf_event`, `constant_k`, `constant_multiplier_tmp`, `counter_ARR_Bemf`, `SIXSTEP_Base_InitTypeDef::Current_Reference`, `SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch`, `delta`, `SIXSTEP_Base_InitTypeDef::demagn_counter`, `SIXSTEP_Base_InitTypeDef::demagn_value`, `EI_Speed_Hz`, `Enable_start_button`, `FALSE`, `FILTER_DEEP`, `HF_TIMx`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_ARR`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_CCR`, `SIXSTEP_Base_InitTypeDef::HF_TIMx_PSC`, `HFBuffer`,

HFBufferIndex, HFBUFFERSIZE, index_adc_chn, index_align,
index_ARR_step, index_array, index_motor_run, index_pot_filt,
index_startup_motor, INITIAL_DEMAGN_DELAY,
SIXSTEP_Base_InitTypeDef::Integral_Term_sum,
SIXSTEP_Base_InitTypeDef::Ireference, LF_TIMx,
SIXSTEP_Base_InitTypeDef::LF_TIMx_ARR,
SIXSTEP_Base_InitTypeDef::LF_TIMx_PSC, MAX_POT_SPEED,
MC_Set_PI_param(),
MC_SixStep_Current_Reference_Setvalue(),
MC_SixStep_Current_Reference_Start(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH1(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH2(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH3(),
MC_SixStep_Ramp_Motor_calc(), mech_accel_hz,
Mech_Speed_RPM, n_zcr_startup, NUMBER_OF_STEPS,
SIXSTEP_Base_InitTypeDef::numberofitemArr,
SIXSTEP_Base_InitTypeDef::NUMPOLESPAIRS, potent_filtered,
SIXSTEP_Base_InitTypeDef::pulse_value,
SIXSTEP_Base_InitTypeDef::Ramp_Start,
SIXSTEP_PI_PARAM_InitTypeDef_t::Reference,
SIXSTEP_Base_InitTypeDef::Regular_channel,
Rotor_poles_pairs, SIXSTEP_Base_InitTypeDef::RUN_Motor,
SIXSTEP_Base_InitTypeDef::speed_fdbk, speed_fdbk_error,
SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered,
SPEED_LOOP_TIME,
SIXSTEP_Base_InitTypeDef::Speed_Loop_Time,
SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered,
speed_sum_pot_filt, speed_sum_sp_filt,
SIXSTEP_Base_InitTypeDef::Speed_target_ramp,
speed_tmp_array, speed_tmp_buffer,
SIXSTEP_Base_InitTypeDef::SPEED_VALIDATED,
startup_bemf_failure, STARTUP_CURRENT_REFERENCE,
SIXSTEP_Base_InitTypeDef::status_prev,
SIXSTEP_Base_InitTypeDef::step_position,
SIXSTEP_Base_InitTypeDef::SYSCLK_frequency,
T_single_step, T_single_step_first_value, TARGET_SPEED,
target_speed, test_motor_run, Tick_cnt,
Time_vector_prev_tmp, Time_vector_tmp, TRUE, uwTick, and
SIXSTEP_Base_InitTypeDef::VALIDATION_OK.

Referenced by **MC_SixStep_INIT()**, and **MC_StopMotor()**.

void MC_StartMotor (void)

Definition at line **978** of file **6Step_Lib.c**.

References **ADCx**, **BSP_X_NUCLEOFAULTLEDON()**, **dac_status**, **LF_TIMx**, **SIXSTEP_Base_InitTypeDef::RUN_Motor**, **START**, **START_DAC()**, **SIXSTEP_Base_InitTypeDef::STATUS**, **TRUE**, and **uwTick**.

Referenced by **MC_EXT_button_SixStep()**, and **MC_SysTick_SixStep_MediumFrequencyTask()**.

void MC_StopMotor (void)

Definition at line **1000** of file **6Step_Lib.c**.

References **ADCx**, **BSP_X_NUCLEOFAULTLEDOFF()**, **HF_TIMx**, **LF_TIMx**, **MC_SixStep_Current_Reference_Stop()**, **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D()**, **MC_SixStep_RESET()**, **MC_SixStep_Stop_PWM_driving()**, **SIXSTEP_Base_InitTypeDef::RUN_Motor**, **SIXSTEP_Base_InitTypeDef::STATUS**, **STOP**, and **uwTick**.

Referenced by **MC_EXT_button_SixStep()**, **MC_SixStep_ARR_step()**, **MC_SysTick_SixStep_MediumFrequencyTask()**, and **TIM1_BRK_TIM9_IRQHandler()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_TABLE

MIDDLEWARES » MC_6-STEP_LIB

Set the peripherals (TIMx, GPIO etc.) for each step. [More...](#)

Functions

```
void MC_SixStep_TABLE (uint8_t)
```

Detailed Description

Set the peripherals (TIMx, GPIO etc.) for each step.

Parameters

step_number step number selected

Return values

None

Function Documentation

void MC_SixStep_TABLE (uint8_t step_number)

Definition at line [164](#) of file [6Step_Lib.c](#).

References

[SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch](#),
[GPIO_CH_COMM](#), [GPIO_COMM](#), [GPIO_PORT_COMM](#),
[MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#),
[MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#),
[MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#),
[MC_SixStep_HF_TIMx_SetDutyCycle_CH1\(\)](#),
[MC_SixStep_HF_TIMx_SetDutyCycle_CH2\(\)](#),
[MC_SixStep_HF_TIMx_SetDutyCycle_CH3\(\)](#),
[SIXSTEP_Base_InitTypeDef::pulse_value](#), and
[SIXSTEP_Base_InitTypeDef::Regular_channel](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_NEXT_step

MIDDLEWARES » MC_6-STEP_LIB

Generate the next step number according with the direction (CW or CCW) [More...](#)

Functions

```
void MC_SixStep_NEXT_step (void)
```

Detailed Description

Generate the next step number according with the direction (CW or CCW)

Return values

`uint8_t SIXSTEP_parameters.status`

Function Documentation

void MC_SixStep_NEXT_step (void)

Definition at line [239](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::ALIGN_OK](#), [ARR_LF](#), [BEMF_CONSEC_DOWN_MAX](#), [SIXSTEP_Base_InitTypeDef::BEMF_Tdown_count](#), [SIXSTEP_Base_InitTypeDef::CL_READY](#), [SIXSTEP_Base_InitTypeDef::CMD](#), [SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch](#), [SIXSTEP_Base_InitTypeDef::demagn_counter](#), [FALSE](#), [HF_TIMx](#), [LF_TIMx](#), [MC_GetMechSpeedRPM\(\)](#), [MC_SixStep_ADC_Channel\(\)](#), [MC_SixStep_Start_PWM_driving\(\)](#), [MC_SixStep_TABLE\(\)](#), [n_zcr_startup](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), [SIXSTEP_Base_InitTypeDef::Regular_channel](#), [SIXSTEP_Base_InitTypeDef::speed_fdbk](#), [speed_fdbk_error](#), [SIXSTEP_Base_InitTypeDef::status_prev](#), [SIXSTEP_Base_InitTypeDef::step_position](#), [TRUE](#), and [SIXSTEP_Base_InitTypeDef::VALIDATION_OK](#).

Referenced by [MC_TIMx_SixStep_timebase\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_RESET

MIDDLEWARES » MC_6-STEP_LIB

Reset all variables used for 6Step control algorithm. [More...](#)

Functions

```
void MC_SixStep_RESET()
```

Detailed Description

Reset all variables used for 6Step control algorithm.

Return values

None

Function Documentation

void MC_SixStep_RESET (void)

Definition at line [353](#) of file [6Step.Lib.c](#).

References [ADC_Bemf_CH1](#), [ADC_Bemf_CH2](#), [ADC_Bemf_CH3](#), [SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_DOWN](#), [SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_UP](#), [ADC_CH_1](#), [ADC_CH_2](#), [ADC_CH_3](#), [ADC_CH_4](#), [SIXSTEP_Base_InitTypeDef::ADC_SEQ_CHANNEL](#), [SIXSTEP_Base_InitTypeDef::ALIGN_OK](#), [SIXSTEP_Base_InitTypeDef::ALIGNMENT](#), [ARR_LF](#), [SIXSTEP_Base_InitTypeDef::ARR_OK](#), [array_completed](#), [SIXSTEP_Base_InitTypeDef::BEMF_OK](#), [SIXSTEP_Base_InitTypeDef::BEMF_Tdown_count](#), [BEMF_THRSLD_DOWN](#), [BEMF_THRSLD_UP](#), [buffer_completed](#), [SIXSTEP_Base_InitTypeDef::CL_READY](#), [SIXSTEP_Base_InitTypeDef::CMD](#), [cnt_bemf_event](#), [constant_k](#), [constant_multiplier_tmp](#), [counter_ARR_Bemf](#), [SIXSTEP_Base_InitTypeDef::Current_Reference](#), [SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch](#), [delta](#), [SIXSTEP_Base_InitTypeDef::demagn_counter](#), [SIXSTEP_Base_InitTypeDef::demagn_value](#), [El_Speed_Hz](#), [Enable_start_button](#), [FALSE](#), [FILTER_DEEP](#), [HF_TIMx](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_ARR](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_CCR](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_PSC](#), [HFBuffer](#), [HFBufferIndex](#), [HFBUFFERSIZE](#), [index_adc_chn](#), [index_align](#), [index_ARR_step](#), [index_array](#), [index_motor_run](#), [index_pot_filt](#), [index_startup_motor](#), [INITIAL_DEMAGN_DELAY](#), [SIXSTEP_Base_InitTypeDef::Integral_Term_sum](#), [SIXSTEP_Base_InitTypeDef::Ireference](#), [LF_TIMx](#), [SIXSTEP_Base_InitTypeDef::LF_TIMx_ARR](#), [SIXSTEP_Base_InitTypeDef::LF_TIMx_PSC](#), [MAX_POT_SPEED](#), [MC_Set_PI_param\(\)](#),

```
MC_SixStep_Current_Reference_Setvalue(),
MC_SixStep_Current_Reference_Start(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH1(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH2(),
MC_SixStep_HF_TIMx_SetDutyCycle_CH3(),
MC_SixStep_Ramp_Motor_calc(), mech_accel_hz,
Mech_Speed_RPM, n_zcr_startup, NUMBER_OF_STEPS,
SIXSTEP_Base_InitTypeDef::numberofitemArr,
SIXSTEP_Base_InitTypeDef::NUMPOLESPAIRS, potent_filtered,
SIXSTEP_Base_InitTypeDef::pulse_value,
SIXSTEP_Base_InitTypeDef::Ramp_Start,
SIXSTEP_PI_PARAM_InitTypeDef_t::Reference,
SIXSTEP_Base_InitTypeDef::Regular_channel,
Rotor_poles_pairs, SIXSTEP_Base_InitTypeDef::RUN_Motor,
SIXSTEP_Base_InitTypeDef::speed_fdbk, speed_fdbk_error,
SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered,
SPEED_LOOP_TIME,
SIXSTEP_Base_InitTypeDef::Speed_Loop_Time,
SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered,
speed_sum_pot_filt, speed_sum_sp_filt,
SIXSTEP_Base_InitTypeDef::Speed_target_ramp,
speed_tmp_array, speed_tmp_buffer,
SIXSTEP_Base_InitTypeDef::SPEED_VALIDATED,
startup_bemf_failure, STARTUP_CURRENT_REFERENCE,
SIXSTEP_Base_InitTypeDef::status_prev,
SIXSTEP_Base_InitTypeDef::step_position,
SIXSTEP_Base_InitTypeDef::SYSCLK_frequency,
T_single_step, T_single_step_first_value, TARGET_SPEED,
target_speed, test_motor_run, Tick_cnt,
Time_vector_prev_tmp, Time_vector_tmp, TRUE, uwTick, and
SIXSTEP_Base_InitTypeDef::VALIDATION_OK.
```

Referenced by **MC_SixStep_INIT()**, and **MC_StopMotor()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_Ramp_Motor_calc

MIDDLEWARES » MC_6-STEP_LIB

Calculate the acceleration profile step by step for motor during start-up.

More...

Functions

```
void MC_SixStep_Ramp_Motor_calc (void)
```

Detailed Description

Calculate the acceleration profile step by step for motor during start-up.

Return values

None

Function Documentation

void MC_SixStep_Ramp_Motor_calc (void)

Definition at line **479** of file **6Step.Lib.c**.

References **SIXSTEP_Base_InitTypeDef::ACCEL**, **ALIGNMENT**, **SIXSTEP_Base_InitTypeDef::ARR_value**, **constant_k**, **constant_multiplier_tmp**, **delta**, **index_startup_motor**, **SIXSTEP_Base_InitTypeDef::Ireference**, **MC_SixStep_Current_Reference_Setvalue()**, **MCM_Sqrt()**, **mech_accel_hz**, **NUMBER_OF_STEPS**, **SIXSTEP_Base_InitTypeDef::prescaler_value**, **Rotor_poles_pairs**, **START**, **SIXSTEP_Base_InitTypeDef::STATUS**, **SIXSTEP_Base_InitTypeDef::SYSCLK_frequency**, **T_single_step**, **T_single_step_first_value**, **Time_vector_prev_tmp**, and **Time_vector_tmp**.

Referenced by **MC_SixStep_Alignment()**, **MC_SixStep_ARR_step()**, and **MC_SixStep_RESET()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_ARR_step

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Generate the ARR value for Low Frequency TIM during start-up.

[More...](#)

Functions

```
void MC_SixStep_ARR_step (void)
```

Detailed Description

Generate the ARR value for Low Frequency TIM during start-up.

Return values

None

Function Documentation

void MC_SixStep.ARR_step (void)

Definition at line **573** of file **6Step.Lib.c**.

References **SIXSTEP_Base_InitTypeDef::ACCEL**, **SIXSTEP_Base_InitTypeDef::ALIGN_OK**, **SIXSTEP_Base_InitTypeDef::ALIGNMENT**, **SIXSTEP_Base_InitTypeDef::ARR_OK**, **SIXSTEP_Base_InitTypeDef::ARR_value**, **FALSE**, **index.ARR_step**, **index_startup_motor**, **LF_TIMx**, **MC_SixStep_Ramp_Motor_calc()**, **MC_StopMotor()**, **MINIMUM_ACC**, **SIXSTEP_Base_InitTypeDef::numberofitemArr**, **SIXSTEP_PI_PARAM_InitTypeDef_t::Reference**, **STARTUP**, **STARTUP_FAILURE**, **SIXSTEP_Base_InitTypeDef::STATUS**, **TRUE**, and **SIXSTEP_Base_InitTypeDef::VALIDATION_OK**.

Referenced by **MC_TIMx_SixStep_timebase()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_Alignment

MIDDLEWARES » MC_6-STEP_LIB

Generate the motor alignment. [More...](#)

Functions

```
void MC_SixStep_Alignment(void)
```

Detailed Description

Generate the motor alignment.

Return values

None

Function Documentation

void MC_SixStep_Alignment(void)

Definition at line [657](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::ALIGN_OK](#), [ALIGNMENT](#), [SIXSTEP_Base_InitTypeDef::ARR_value](#), [index_align](#), [index_startup_motor](#), [LF_TIMx](#), [MC_SixStep_Ramp_Motor_calc\(\)](#), [MC_SixStep_Speed_Val_target_potentiometer\(\)](#), [SIXSTEP_Base_InitTypeDef::prescaler_value](#), [STARTUP](#), [SIXSTEP_Base_InitTypeDef::STATUS](#), [SIXSTEP_Base_InitTypeDef::step_position](#), [TIME_FOR_ALIGN](#), and [TRUE](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_Speed_Val_target_potentiometer

MIDDLEWARES » MC_6-STEP_LIB

Calculate the Motor Speed validation threshold according with the potentiometer value. More...

Functions

```
void MC_SixStep_Speed_Val_target_potentiometer(void)
```

Detailed Description

Calculate the Motor Speed validation threshold according with the potentiometer value.

Return values

None

Function Documentation

void MC_SixStep_Speed_Val_target_potentiometer (void)

Definition at line [687](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::ADC-Regular_Buffer](#), [MAX_POT_SPEED](#), [MIN_POT_SPEED](#), [target_speed](#), and [VAL_POT_SPEED_DIV](#).

Referenced by [MC_SixStep_Alignment\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_Speed_Potentiometer

MIDDLEWARES » MC_6-STEP_LIB

Calculate the potentiometer value to set the Motor Speed. [More...](#)

Functions

```
void MC_SixStep_Speed_Potentiometer (void)
```

Detailed Description

Calculate the potentiometer value to set the Motor Speed.

Return values

None

Function Documentation

void MC_SixStep_Speed_Potentiometer (void)

Definition at line [707](#) of file [6Step.Lib.c](#).

References [HBuffer](#), [HBUFSIZE](#),
[MC_Potentiometer_filter\(\)](#), and
[SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Set_PI_param

MIDDLEWARES » MC_6-STEP_LIB

Set all parameters for PI regulator. [More...](#)

Functions

```
void MC_Set_PI_param (SIXSTEP_PI_PARAM_InitTypeDef_t *)
```

Detailed Description

Set all parameters for PI regulator.

Parameters

PI_PARAM

Return values

None

Function Documentation

```
void  
MC_Set_PI_param ( SIXSTEP_PI_PARAM_InitTypeDef_t * PI PARA
```

Definition at line [740](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::CW_CCW](#), [FALSE](#), [SIXSTEP_Base_InitTypeDef::KI](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Ki_Gain](#), [SIXSTEP_Base_InitTypeDef::KP](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Kp_Gain](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Lower_Limit_Output](#), [LOWER_OUT_LIMIT](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Max_PID_Output](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Min_PID_Output](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), [target_speed](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Upper_Limit_Output](#), and [UPPER_OUT_LIMIT](#).

Referenced by [MC_SixStep_RESET\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_PI_Controller

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Compute the PI output for the Current Reference. [More...](#)

Functions

```
int16_t MC_PI_Controller (SIXSTEP_PI_PARAM_InitTypeDef_t *,  
                           int16_t)
```

Detailed Description

Compute the PI output for the Current Reference.

Parameters

PI_PARAM PI parameters structure

speed_fdb motor_speed_value

Return values

int16_t Currente reference

Function Documentation

```
int16_t  
MC_PI_Controller ( SIXSTEP_PI_PARAM_InitTypeDef_t * PI_PARAM,  
                    int16_t speed_fd  
                    )
```

Definition at line [768](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::Integral_Term_sum](#), [KI_DIV](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Ki_Gain](#), [KP_DIV](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Kp_Gain](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Lower_Limit_Output](#), [SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), and [SIXSTEP_PI_PARAM_InitTypeDef_t::Upper_Limit_Output](#).

Referenced by [MC_Task_Speed\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Task_Speed

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Main task: Speed Loop with PI regulator. More...

Functions

```
void MC_Task_Speed (void)
```

Detailed Description

Main task: Speed Loop with PI regulator.

Return values

None

Function Documentation

void MC_Task_Speed (void)

Definition at line [838](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::BEMF_OK](#),
[SIXSTEP_Base_InitTypeDef::CL_READY](#),
[SIXSTEP_Base_InitTypeDef::Current_Reference](#), [dac_status](#),
[MC_Bemf_Delay\(\)](#), [MC_PI_Controller\(\)](#),
[MC_SixStep_Current_Reference_Setvalue\(\)](#),
[SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), [RUN](#),
[SET_DAC_value\(\)](#),
[SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered](#),
[SIXSTEP_Base_InitTypeDef::SPEED_VALIDATED](#),
[SIXSTEP_Base_InitTypeDef::STATUS](#), [target_speed](#), [TRUE](#),
[VALIDATION](#), and
[SIXSTEP_Base_InitTypeDef::VALIDATION_OK](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Set_Speed

MIDDLEWARES » MC_6-STEP_LIB

Set the new motor speed value. [More...](#)

Functions

```
void MC_Set_Speed (uint16_t speed_value)
```

Detailed Description

Set the new motor speed value.

Parameters

speed_value set new motor speed

Return values

None

Function Documentation

void MC_Set_Speed (uint16_t speed_value)

Definition at line [889](#) of file [6Step.Lib.c](#).

References [ADC_SPEED_TH](#),
[SIXSTEP_Base_InitTypeDef::CW_CCW](#), [MAX_POT_SPEED](#),
[MIN_POT_SPEED](#),
[SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#),
[SIXSTEP_Base_InitTypeDef::Speed_Ref_filtered](#), and
[SIXSTEP_Base_InitTypeDef::Speed_target_ramp](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Bemf_Delay

MIDDLEWARES » MC_6-STEP_LIB

Take the delay time after each new 6-step commutation. [More...](#)

Functions

```
void MC_Bemf_Delay (void)
```

Detailed Description

Take the delay time after each new 6-step commutation.

Return values

None

Function Documentation

void MC_Bemf_Delay (void)

Definition at line **965** of file **6Step.Lib.c**.

References **Bemf_delay_calc()**.

Referenced by **MC_Task_Speed()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_StartMotor

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Start the Motor. [More...](#)

Functions

```
void MC_StartMotor()
```

Detailed Description

Start the Motor.

Return values

None

Function Documentation

void MC_StartMotor (void)

Definition at line **978** of file **6Step.Lib.c**.

References **ADCx**, **BSP_X_NUCLEOFAULTLEDON()**, **dac_status**, **LF_TIMx**, **SIXSTEP_Base_InitTypeDef::RUN_Motor**, **START**, **START_DAC()**, **SIXSTEP_Base_InitTypeDef::STATUS**, **TRUE**, and **uwTick**.

Referenced by **MC_EXT_button_SixStep()**, and **MC_SysTick_SixStep_MediumFrequencyTask()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_StopMotor

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Stop the Motor. [More...](#)

Functions

```
void MC_StopMotor ()
```

Detailed Description

Stop the Motor.

Return values

None

Function Documentation

void MC_StopMotor (void)

Definition at line **1000** of file **6Step.Lib.c**.

References [ADCx](#), [BSP_X_NUCLEOFAULT_LED_OFF\(\)](#), [HF_TIMx](#), [LF_TIMx](#), [MC_SixStep_Current_Reference_Stop\(\)](#), [MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#), [MC_SixStep_RESET\(\)](#), [MC_SixStep_Stop_PWM_driving\(\)](#), [SIXSTEP_Base_InitTypeDef::RUN_Motor](#), [SIXSTEP_Base_InitTypeDef::STATUS](#), [STOP](#), and [uwTick](#).

Referenced by [MC_EXT_button_SixStep\(\)](#), [MC_SixStep_ARR_step\(\)](#), [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#), and [TIM1_BRK_TIM9_IRQHandler\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_GetElSpeedHz

MIDDLEWARES » MC_6-STEP_LIB

Get the Electrical Motor Speed from ARR value of LF TIM. [More...](#)

Functions

int32_t **MC_GetElSpeedHz** (void)

Detailed Description

Get the Electrical Motor Speed from ARR value of LF TIM.

Return values

int32_t Return the electrical motor speed

Function Documentation

int32_t MC_GetElSpeedHz (void)

Definition at line **1025** of file **6Step.Lib.c**.

References **El_Speed_Hz**, **LF_TIMx**,
SIXSTEP_PI_PARAM_InitTypeDef_t::Reference, and
SIXSTEP_Base_InitTypeDef::SYSCLK_frequency.

Referenced by **MC_GetMechSpeedRPM()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_GetMechSpeedRPM

MIDDLEWARES » MC_6-STEP_LIB

Get the Mechanical Motor Speed (RPM) More...

Functions

int32_t **MC_GetMechSpeedRPM** (void)

Detailed Description

Get the Mechanical Motor Speed (RPM)

Return values

int32_t Return the mechanical motor speed (RPM)

Function Documentation

int32_t MC_GetMechSpeedRPM (void)

Definition at line **1049** of file **6Step.Lib.c**.

References **MC_GetElSpeedHz()**, **Mech_Speed_RPM**, and **Rotor_poles_pairs**.

Referenced by **MC_SixStep_NEXT_step()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

MC_SixStep_Init_main_data

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Init the main variables for motor driving from [MC_SixStep_param.h](#).

[More...](#)

Functions

```
void MC_SixStep_Init_main_data (void)
```

Detailed Description

Init the main variables for motor driving from [MC_SixStep_param.h](#).

Return values

None

Function Documentation

void MC_SixStep_Init_main_data (void)

Definition at line **1065** of file **6Step.Lib.c**.

References **ACC**, **SIXSTEP_Base_InitTypeDef::ACCEL**, **SIXSTEP_Base_InitTypeDef::CW_CCW**, **DIRECTION**, **SIXSTEP_Base_InitTypeDef::Ireference**, **SIXSTEP_Base_InitTypeDef::KI**, **KI_GAIN**, **SIXSTEP_Base_InitTypeDef::KP**, **KP_GAIN**, **NUM_POLE_PAIRS**, **SIXSTEP_Base_InitTypeDef::NUMPOLESPAIRS**, **POTENTIOMETER**, **SIXSTEP_Base_InitTypeDef::Potentiometer**, and **STARTUP_CURRENT_REFERENCE**.

Referenced by **MC_SixStep_INIT()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_INIT

MIDDLEWARES » MC_6-STEP_LIB

Initialitation function for SixStep library. [More...](#)

Functions

```
void MC_SixStep_INIT()
```

Detailed Description

Initialitation function for SixStep library.

Return values

None

Function Documentation

void MC_SixStep_INIT(void)

Definition at line [1087](#) of file [6Step_Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::Button_ready](#), [FALSE](#), [HF_TIMx](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_ARR](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_CCR](#), [SIXSTEP_Base_InitTypeDef::HF_TIMx_PSC](#), [SIXSTEP_Base_InitTypeDef::Ireference](#), [LF_TIMx](#), [SIXSTEP_Base_InitTypeDef::LF_TIMx_ARR](#), [SIXSTEP_Base_InitTypeDef::LF_TIMx_PSC](#), [MC_SixStep_Current_Reference_Setvalue\(\)](#), [MC_SixStep_Init_main_data\(\)](#), [MC_SixStep_Nucleo_Init\(\)](#), [MC_SixStep_RESET\(\)](#), [MC_UI_INIT\(\)](#), and [TRUE](#).

Referenced by [main\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_TIMx_SixStep_timebase

MIDDLEWARES » MC_6-STEP_LIB

Low Frequency Timer Callback - Call the next step and request the filtered speed value. More...

Functions

```
void MC_TIMx_SixStep_timebase (void)
```

Detailed Description

Low Frequency Timer Callback - Call the next step and request the filtered speed value.

Return values

None

Function Documentation

void MC_TIMx_SixStep_timebase (void)

Definition at line [1123](#) of file [6Step_Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::ARR_OK](#),
[MC_SixStep_ARR_step\(\)](#), [MC_SixStep_NEXT_step\(\)](#), and
[MC_Speed_Filter\(\)](#).

Referenced by [HAL_TIM_PeriodElapsedCallback\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Speed_Filter

MIDDLEWARES » MC_6-STEP_LIB

Calculate the speed filtered. [More...](#)

Functions

```
void MC_Speed_Filter (void)
```

Detailed Description

Calculate the speed filtered.

Return values

None

Function Documentation

void MC_Speed_Filter(void)

Definition at line [1144](#) of file [6Step.Lib.c](#).

References [array_completed](#), [FALSE](#), [FILTER_DEEP](#), [index_array](#), [SIXSTEP_Base_InitTypeDef::speed_fdbk](#), [SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered](#), [speed_sum_sp_filt](#), [speed_tmp_array](#), and [TRUE](#).

Referenced by [MC_TIMx_SixStep_timebase\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_Potentiometer_filter

MIDDLEWARES » MC_6-STEP_LIB

Calculate the filtered potentiometer value. [More...](#)

Functions

`uint16_t MC_Potentiometer_filter (uint16_t)`

Detailed Description

Calculate the filtered potentiometer value.

Return values

`uint16_t` Return the filtered potentiometer value

Function Documentation

`uint16_t
MC_Potentiometer_filter (uint16_t potentiometer_value)`

Definition at line [1189](#) of file [6Step.Lib.c](#).

References `buffer_completed`, `FALSE`, `FILTER_DEEP`,
`index_pot_filt`, `potent_filtered`, `speed_sum_pot_filt`,
`speed_tmp_buffer`, and `TRUE`.

Referenced by [MC_SixStep_Speed_Potentiometer\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SysTick_SixStep_MediumFrequencyTask

MIDDLEWARES » MC_6-STEP_LIB

Systick Callback - Call the Speed loop. [More...](#)

Functions

```
void MC_SysTick_SixStep_MediumFrequencyTask (void)
```

Detailed Description

Systick Callback - Call the Speed loop.

Return values

None

Function Documentation

void MC_SysTick_SixStep_MediumFrequencyTask (void)

Definition at line [1246](#) of file [6Step.Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::ACCEL](#),
[SIXSTEP_Base_InitTypeDef::ALIGN_OK](#),
[SIXSTEP_Base_InitTypeDef::ALIGNMENT](#), [BUTTON_DELAY](#),
[cnt_bemf_event](#), [DEMO_START_TIME](#), [DEMO_STOP_TIME](#),
[Enable_start_button](#), [FALSE](#), [HAL_GetTick\(\)](#), [index_motor_run](#),
[MC_Set_Speed\(\)](#), [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_Speed_Potentiometer\(\)](#), [MC_StartMotor\(\)](#),
[MC_StopMotor\(\)](#), [MC_Task_Speed\(\)](#),
[SIXSTEP_Base_InitTypeDef::MediumFrequencyTask_flag](#),
[MINIMUM_ACC](#), [SIXSTEP_Base_InitTypeDef::Potentiometer](#),
[speed_fdbk_error](#),
[SIXSTEP_Base_InitTypeDef::Speed_Loop_Time](#),
[SPEEDFBKERROR](#), [STARTUP_BEMF_FAILURE](#),
[startup_bemf_failure](#), [SIXSTEP_Base_InitTypeDef::STATUS](#),
[test_motor_run](#), [Tick_cnt](#), [TRUE](#), [UART_Communication_Task\(\)](#),
[UART_FLAG_RECEIVE](#), and
[SIXSTEP_Base_InitTypeDef::VALIDATION_OK](#).

Referenced by [HAL_SYSTICK_Callback\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_ARR_Bemf

MIDDLEWARES » MC_6-STEP_LIB

Calculate the new Autoreload value (ARR) for Low Frequency timer.

More...

Functions

```
void MC_SixStep.ARR_Bemf (uint8_t)
```

Detailed Description

Calculate the new Autoreload value (ARR) for Low Frequency timer.

Return values

None

Function Documentation

void MC_SixStep.ARR_Bemf(uint8_t up_bemf)

Definition at line [1328](#) of file [6Step.Lib.c](#).

References [ARR_LF](#), [BEMF_CNT_EVENT_MAX](#), [SIXSTEP_Base_InitTypeDef::BEMF_OK](#), [cnt_bemf_event](#), [counter_ARR_Bemf](#), [GPIO_CH_ZCR](#), [GPIO_PORT_ZCR](#), [GPIO_ZERO_CROSS](#), [LF_TIMx](#), [n_zcr_startup](#), [NUMBER_ZCR](#), [SIXSTEP_Base_InitTypeDef::SPEED_VALIDATED](#), [startup_bemf_failure](#), [SIXSTEP_Base_InitTypeDef::status_prev](#), [SIXSTEP_Base_InitTypeDef::step_position](#), [TRUE](#), and [SIXSTEP_Base_InitTypeDef::VALIDATION_OK](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_ADCx_SixStep_Bemf

MIDDLEWARES » MC_6-STEP_LIB

Compute the zero crossing detection. [More...](#)

Functions

```
void MC_ADCx_SixStep_Bemf (void)
```

Detailed Description

Compute the zero crossing detection.

Return values

None

Function Documentation

void MC_ADCx_SixStep_Bemf(void)

Definition at line [1381](#) of file [6Step_Lib.c](#).

References

[SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_DOWN](#),
[SIXSTEP_Base_InitTypeDef::ADC_BEMF_threshold_UP](#),
[SIXSTEP_Base_InitTypeDef::ADC_BUFFER](#),
[SIXSTEP_Base_InitTypeDef::ADC_Regular_Buffer](#),
[SIXSTEP_Base_InitTypeDef::ADC_SEQ_CHANNEL](#), [ADCx](#),
[ALIGNMENT](#), [SIXSTEP_Base_InitTypeDef::BEMF_Tdown_count](#),
[SIXSTEP_Base_InitTypeDef::CurrentRegular_BEMF_ch](#),
[SIXSTEP_Base_InitTypeDef::demagn_counter](#),
[SIXSTEP_Base_InitTypeDef::demagn_value](#), [GPIO_CH_COMM](#),
[GPIO_PORT_COMM](#), [HF_TIMx](#), [HFBuffer](#), [HFBufferIndex](#),
[HFBUFFERSIZE](#), [index_adc_chn](#), [MC_SixStep_ADC_Channel\(\)](#),
[MC_SixStep_ARR_Bemf\(\)](#),
[SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), [START](#),
[SIXSTEP_Base_InitTypeDef::STATUS](#), and
[SIXSTEP_Base_InitTypeDef::step_position](#).

Referenced by [HAL_ADC_ConvCpltCallback\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_EXT_button_SixStep

MIDDLEWARES » MC_6-STEP_LIB

GPIO EXT Callback - Start or Stop the motor through the Blue push button on STM32Nucleo. [More...](#)

Functions

```
void MC_EXT_button_SixStep ()
```

Detailed Description

GPIO EXT Callback - Start or Stop the motor through the Blue push button on STM32Nucleo.

Return values

None

Function Documentation

void MC_EXT_button_SixStep (void)

Definition at line [1577](#) of file [6Step_Lib.c](#).

References [SIXSTEP_Base_InitTypeDef::Button_ready](#), [Enable_start_button](#), [FALSE](#), [MC_StartMotor\(\)](#), [MC_StopMotor\(\)](#), [SIXSTEP_Base_InitTypeDef::RUN_Motor](#), and [TRUE](#).

Referenced by [HAL_GPIO_EXTI_Callback\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Macros

Main_Motor_parameters

MIDDLEWARES » MC_6-STEP_LIB

All motor parameters for 6Step driving. [More...](#)

Macros

```
#define NUM_POLE_PAIRS 2  
  
#define DIRECTION 0  
  
#define TARGET_SPEED 4000  
  
#define POTENTIOMETER 1  
  
#define STARTUP_DUTY_CYCLE 600  
  
#define STARTUP_CURRENT_REFERENCE STARTUP_DUTY_CY  
  
#define ACC 1000000  
  
#define MINIMUM_ACC 1000  
  
#define NUMBER_OF_STEPS 20000  
  
#define TIME_FOR_ALIGN 500  
  
#define BUTTON_DELAY 1000  
  
#define NUMBER_ZCR 12  
  
#define SPEED_LOOP_TIME 1  
  
#define KP_GAIN 500  
  
#define KI_GAIN 50  
  
#define KP_DIV 4096  
  
#define KI_DIV 4096
```

```
#define LOWER_OUT_LIMIT 50
#define UPPER_OUT_LIMIT 800
#define MAX_POT_SPEED 20000
#define MIN_POT_SPEED 1700
#define VAL_POT_SPEED_DIV 2
#define INITIAL_DEMAGN_DELAY 10
#define BEMF_THRSLD_DOWN 200
#define BEMF_THRSLD_UP 200
#define FILTER_DEEP 20
#define HFBUFFERSIZE 10
#define ADC_SPEED_TH 82
#define BEMF_CONSEC_DOWN_MAX 10
#define BEMF_CNT_EVENT_MAX 100
#define GPIO_ZERO_CROSS 1
#define GPIO_COMM 1
#define DEMO_START_TIME 5000
#define DEMO_STOP_TIME 2000
#define DEMAGN_VAL_1 1
#define DEMAGN_VAL_2 2
```

```
#define DEMAGN_VAL_3 3  
  
#define DEMAGN_VAL_4 4  
  
#define DEMAGN_VAL_5 5  
  
#define DEMAGN_VAL_6 6  
  
#define DEMAGN_VAL_7 7  
  
#define DEMAGN_VAL_8 8  
  
#define DEMAGN_VAL_9 9  
  
#define DEMAGN_VAL_10 10  
  
#define DEMAGN_VAL_11 11  
  
#define DEMAGN_VAL_12 12  
  
#define DEMAGN_VAL_13 13  
  
#define DEMAGN_VAL_14 14  
  
#define TRUE 1  
  
#define FALSE 0
```

Detailed Description

All motor parameters for 6Step driving.

Macro Definition Documentation

#define ACC 1000000

Mechanical acceleration rate (setting available in manual mode,
LOAD_TYPE = 0)

Definition at line [73](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

#define ADC_SPEED_TH 82

Fixed threshold to change the target speed (t.b.f) Motor stall detection
parameters

Definition at line [99](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Set_Speed\(\)](#).

#define BEMF_CNT_EVENT_MAX 100

Maximum number of BEMF Counter in open loop Debug pin

Definition at line [103](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#).

#define BEMF_CONSEC_DOWN_MAX 10

Maximum value of BEMF Consecutive Threshold Falling Crossings
Counter in closed loop

Definition at line [102](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_NEXT_step\(\)](#).

```
#define BEMF_THRSLD_DOWN 200
```

Zero Crossing threshold

Definition at line [93](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

```
#define BEMF_THRSLD_UP 200
```

Zero Crossing threshold Speed filtering parameters

Definition at line [94](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

```
#define BUTTON_DELAY 1000
```

Delay time to enable push button for new command (1 = 1msec)

Definition at line [77](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
#define DEMAGN_VAL_1 1
```

Look UP table for dynamic demagn control for speed into
(10000,12000] or [-12000,-10000) range

Definition at line [115](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_10 10

Look UP table for dynamic demagn control for speed into (1800, 2600] or [- 2600,- 1800) range

Definition at line [124](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_11 11

Look UP table for dynamic demagn control for speed into (1500, 1800] or [- 1800,- 1500) range

Definition at line [125](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_12 12

Look UP table for dynamic demagn control for speed into (1300, 1500] or [- 1500,- 1300) range

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

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_13 13

Look UP table for dynamic demagn control for speed into (1000, 1300] or [- 1300,- 1000) range

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

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_14 14

Look UP table for dynamic demagn control for speed into [500, 1000] or [- 1000,- 500] range

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

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_2 2

Look UP table for dynamic demagn control for speed into (7800,10000] or [-10000,- 7800) range

Definition at line [116](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_3 3

Look UP table for dynamic demagn control for speed into (6400, 7800] or [- 7800,- 6400) range

Definition at line [117](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_4 4

Look UP table for dynamic demagn control for speed into (5400, 6400] or [- 6400,- 5400) range

Definition at line [118](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_5 5

Look UP table for dynamic demagn control for speed into (4650, 5400] or [- 5400,- 4650) range

Definition at line [119](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_6 6

Look UP table for dynamic demagn control for speed into (4100, 4650] or [- 4650,- 4100) range

Definition at line [120](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_7 7

Look UP table for dynamic demagn control for speed into (3650, 4100] or [- 4100,- 3650) range

Definition at line [121](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_8 8

Look UP table for dynamic demagn control for speed into (3300, 3650] or [- 3650,- 3300) range

Definition at line [122](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMAGN_VAL_9 9

Look UP table for dynamic demagn control for speed into (2600, 3300] or [- 3300,- 2600) range

Definition at line [123](#) of file [MC_SixStep_param.h](#).

Referenced by [Bemf_delay_calc\(\)](#).

#define DEMO_START_TIME 5000

Time (msec) to keep the motor in run mode

Definition at line [111](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

#define DEMO_STOP_TIME 2000

Time (msec) to keep the motor in stop mode Look UP table for dynamic demagn control of speed

Definition at line [112](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

#define DIRECTION 0

Set motor direction CW = 0 and CCW = 1

Definition at line [60](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

```
#define FALSE 0
```

Define FALSE

Definition at line [131](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_EXT_button_SixStep\(\)](#),
[MC_Potentiometer_filter\(\)](#), [MC_Set_PI_param\(\)](#),
[MC_SixStep_ARR_step\(\)](#), [MC_SixStep_INIT\(\)](#),
[MC_SixStep_NEXT_step\(\)](#), [MC_SixStep_RESET\(\)](#),
[MC_Speed_Filter\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
#define FILTER_DEEP 20
```

Number of bits for digital filter

Definition at line [97](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Potentiometer_filter\(\)](#), [MC_SixStep_RESET\(\)](#),
and [MC_Speed_Filter\(\)](#).

```
#define GPIO_COMM 1
```

Enable (1) the GPIO toggling for commutation Demo mode parameters

Definition at line [107](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_TABLE\(\)](#).

```
#define GPIO_ZERO_CROSS 1
```

Enable (1) the GPIO toggling for zero crossing detection

Definition at line [106](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#).

```
#define HFBUFFERSIZE 10
```

Definition at line [98](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_RESET\(\)](#), and [MC_SixStep_Speed_Potentiometer\(\)](#).

```
#define INITIAL_DEMAGN_DELAY 10
```

Initial value for delay time during startup for Bemf detection Zero
Crossissing parameters

Definition at line [90](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

```
#define KI_DIV 4096
```

Ki parameter divider for PI regulator

Definition at line [84](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_PI_Controller\(\)](#).

```
#define KI_GAIN 50
```

Ki parameter for PI regulator

Definition at line [82](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

```
#define KP_DIV 4096
```

Kp parameter divider for PI regulator

Definition at line [83](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_PI_Controller\(\)](#).

```
#define KP_GAIN 500
```

Kp parameter for PI regulator

Definition at line [81](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

```
#define LOWER_OUT_LIMIT 50
```

Low Out value of PI regulator

Definition at line [85](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Set_PI_param\(\)](#).

```
#define MAX_POT_SPEED 20000
```

Maximum Speed regulated by potentiometer

Definition at line [87](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Set_Speed\(\)](#), [MC_SixStep_RESET\(\)](#), and [MC_SixStep_Speed_Val_target_potentiometer\(\)](#).

```
#define MIN_POT_SPEED 1700
```

Minimum Speed regulated by potentiometer

Definition at line [88](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Set_Speed\(\)](#), and
[MC_SixStep_Speed_Val_target_potentiometer\(\)](#).

```
#define MINIMUM_ACC 1000
```

Mechanical acceleration rate for BIG load application

Definition at line [74](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_ARR_step\(\)](#), and
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).

```
#define NUM_POLE_PAIRS 2
```

Number of Motor Pole pairs

Definition at line [59](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

```
#define NUMBER_OF_STEPS 20000
```

Number of elements for motor start-UP (max value 65535)

Definition at line [75](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Ramp_Motor_calc\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define NUMBER_ZCR 12
```

Number of zero crossing event during the startup for closed loop control begin ***** Closed Loop control

Definition at line [78](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_ARR_Bemf\(\)](#).

```
#define POTENTIOMETER 1
```

Enable (1)/Disable (0) the potentiometer

Definition at line [62](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#).

```
#define SPEED_LOOP_TIME 1
```

Speed Loop time (1 = 1msec)

Definition at line [80](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

```
#define  
STARTUP_CURRENT_REFERENCE STARTUP_DUTY_CYCLE
```

Definition at line [72](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Init_main_data\(\)](#), and [MC_SixStep_RESET\(\)](#).

```
#define STARTUP_DUTY_CYCLE 600
```

< ***** Open loop control StartUP Duty Cycle

Definition at line [71](#) of file [MC_SixStep_param.h](#).

Referenced by [STSPIN230_Current_Reference_Start\(\)](#), and [STSPIN230_Current_Reference_Stop\(\)](#).

#define TARGET_SPEED 4000

Target speed in closed loop control

Definition at line [61](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_RESET\(\)](#).

#define TIME_FOR_ALIGN 500

Time for alignment (msec)

Definition at line [76](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Alignment\(\)](#).

#define TRUE 1

Define TRUE

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

Referenced by [MC_EXT_button_SixStep\(\)](#),
[MC_Potentiometer_filter\(\)](#), [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_ARR_step\(\)](#),
[MC_SixStep_INIT\(\)](#), [MC_SixStep_NEXT_step\(\)](#),
[MC_SixStep_RESET\(\)](#), [MC_Speed_Filter\(\)](#), [MC_StartMotor\(\)](#),
[MC_SysTick_SixStep_MediumFrequencyTask\(\)](#), and
[MC_Task_Speed\(\)](#).

```
#define UPPER_OUT_LIMIT 800
```

High Out value of PI regulator

Definition at line [86](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_Set_PI_param\(\)](#).

```
#define VAL_POT_SPEED_DIV 2
```

Validation potentiometer speed divider

Definition at line [89](#) of file [MC_SixStep_param.h](#).

Referenced by [MC_SixStep_Speed_Val_target_potentiometer\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

stm32F401_nucleo_ihm11m1

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#)

Interface file for STM32F401 and Motor Control Library configuration.

[More...](#)

Modules

MC_SixStep_ADC_Channel

Select the new ADC Channel.

MC_SixStep_Nucleo_Init

Init the STM32 register.

START_DAC

Start DAC for debug.

STOP_DAC

Stop DAC for debug.

SET_DAC_value

Set DAC value for debug.

HAL_ADC_ConvCpltCallback

ADC callback.

HAL_TIM_PeriodElapsedCallback

htim callback

HAL_SYSTICK_Callback

Systick callback.

HAL_GPIO_EXTI_Callback

EXT callback.

EnableInput_CH1_E_CH2_E_CH3_D

Enable Input channel CH1 and CH2 for STSPIN230.

EnableInput_CH1_E_CH2_D_CH3_E

Enable Input channel CH1 and CH3 for STSPIN230.

EnableInput_CH1_D_CH2_E_CH3_E

Enable Input channel CH2 and CH3 for STSPIN230.

DisableInput_CH1_D_CH2_D_CH3_D

Enable Input channel CH2 and CH3 for STSPIN230.

Start_PWM_driving

Enable PWM channels for STSPIN230.

Stop_PWM_driving

Disable PWM channels for STSPIN230.

HF_TIMx_SetDutyCycle_CH1

Set the Duty Cycle value for CH1.

HF_TIMx_SetDutyCycle_CH2

Set the Duty Cycle value for CH2.

HF_TIMx_SetDutyCycle_CH3

Set the Duty Cycle value for CH3.

Current_Reference_Start

Enable the Current Reference generation.

Current_Reference_Stop

Disable the Current Reference generation.

Current_Reference_Setvalue

Set the value for Current Reference.

Bemf_delay_calc

Bemf delay calculation.

Get_UART_data

Get the UART value from DR register.

Exported_function_F401

Detailed Description

Interface file for STM32F401 and Motor Control Library configuration.

Interface file for STM32F401 and Library configuration.

Generated by  1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_ADC_Channel

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Select the new ADC Channel. [More...](#)

Functions

void **MC_SixStep_ADC_Channel** (uint32_t adc_ch)

Select the new ADC Channel. [More...](#)

Detailed Description

Select the new ADC Channel.

Function Documentation

void MC_SixStep_ADC_Channel (uint32_t adc_ch)

Select the new ADC Channel.

API function for STM32 instruction.

Parameters

adc_ch

Return values

None

Definition at line **74** of file [stm32F401_nucleo_ihm11m1.c](#).

References [ADCx](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_NEXT_step\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

MC_SixStep_Nucleo_Init

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32f401_nucleo_ihm11m1](#)

Init the STM32 register. [More...](#)

Functions

void **MC_SixStep_Nucleo_Init ()**
Init the STM32 register. More...

Detailed Description

Init the STM32 register.

Function Documentation

void MC_SixStep_Nucleo_Init(void)

Init the STM32 register.

Return values

None

Definition at line 95 of file [stm32F401_nucleo_ihm11m1.c](#).

References [ADC_Bemf_CH1](#), [ADC_Bemf_CH1_ST](#),
[ADC_Bemf_CH2](#), [ADC_Bemf_CH2_ST](#), [ADC_Bemf_CH3](#),
[ADC_Bemf_CH3_ST](#), [ADC_CH_1](#), [ADC_CH_1_ST](#), [ADC_CH_2](#),
[ADC_CH_2_ST](#), [ADC_CH_3](#), [ADC_CH_3_ST](#), [ADC_CH_4](#),
[ADC_CH_4_ST](#), [hadc1](#), [HF_TIMx](#), [HF_TIMx_CH1](#), [HF_TIMx_CH2](#),
[HF_TIMx_CH3](#), and [htim1](#).

Referenced by [MC_SixStep_INIT\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

START_DAC

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Start DAC for debug. More...

Functions

void **START_DAC ()**

Start DAC for debug. More...

Detailed Description

Start DAC for debug.

Function Documentation

void START_DAC (void)

Start DAC for debug.

Return values

None

Definition at line **154** of file **stm32F401_nucleo_ihm11m1.c**.

Referenced by **MC_StartMotor()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

STOP_DAC

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Stop DAC for debug. More...

Functions

void **STOP_DAC ()**

Stop DAC for debug. More...

Detailed Description

Stop DAC for debug.

Function Documentation

void STOP_DAC (void)

Stop DAC for debug.

Return values

None

Definition at line **169** of file **stm32F401_nucleo_ihm11m1.c**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

SET_DAC_value

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Set DAC value for debug. [More...](#)

Functions

void **SET_DAC_value** (uint16_t dac_value)
Set DAC value for debug. [More...](#)

Detailed Description

Set DAC value for debug.

Function Documentation

void SET_DAC_value (uint16_t dac_value)

Set DAC value for debug.

Parameters

dac_value information to plot through DAC

Return values

None

Definition at line **185** of file **stm32F401_nucleo_ihm11m1.c**.

Referenced by **MC_Task_Speed()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)[Functions](#)

HAL_ADC_ConvCpltCallback

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

ADC callback. [More...](#)

Functions

```
void HAL_ADC_ConvCpltCallback (ADC_HandleTypeDef *hadc)  
  ADC callback. More...
```

Detailed Description

ADC callback.

Function Documentation

void HAL_ADC_ConvCpltCallback (ADC_HandleTypeDef * hadc)

ADC callback.

Parameters

hadc

Return values

None

Definition at line [204](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [MC_ADCx_SixStep_Bemf\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

HAL_TIM_PeriodElapsedCallback

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

htim callback More...

Functions

```
void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef  
*htim)  
htim callback More...
```

Detailed Description

htim callback

Function Documentation

void

HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim)

htim callback

Parameters

htim

Return values

None

Definition at line [221](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [MC_TIMx_SixStep_timebase\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

HAL_SYSTICK_Callback

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Systick callback. [More...](#)

Functions

void **HAL_SYSTICK_Callback** ()
Systick callback. More...

Detailed Description

Systick callback.

Function Documentation

void HAL_SYSTICK_Callback()

Systick callback.

Return values

None

Definition at line **238** of file **stm32F401_nucleo_ihm11m1.c**.

References [MC_SysTick_SixStep_MediumFrequencyTask\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

HAL_GPIO_EXTI_Callback

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

EXT callback. More...

Functions

```
void HAL_GPIO_EXTI_Callback (uint16_t GPIO_Pin)
```

EXT callback. More...

Detailed Description

EXT callback.

Function Documentation

void HAL_GPIO_EXTI_Callback (uint16_t GPIO_Pin)

EXT callback.

Parameters

GPIO_Pin

Return values

None

Definition at line 256 of file [stm32F401_nucleo_ihm11m1.c](#).

References [MC_EXT_button_SixStep\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Bemf_delay_calc

MIDDLEWARES » MC_6-STEP_LIB »

[stm32F401_nucleo_ihm11m1](#)

Bemf delay calculation. [More...](#)

Functions

void **Bemf_delay_calc ()**

Bemf delay calculation. [More...](#)

Detailed Description

Bemf delay calculation.

Function Documentation

void Bemf_delay_calc(void)

Bemf delay calculation.

Return values

None

Definition at line **498** of file **stm32F401_nucleo_ihm11m1.c**.

References **DEMAGN_VAL_1**, **DEMAGN_VAL_10**,
DEMAGN_VAL_11, **DEMAGN_VAL_12**, **DEMAGN_VAL_13**,
DEMAGN_VAL_14, **DEMAGN_VAL_2**, **DEMAGN_VAL_3**,
DEMAGN_VAL_4, **DEMAGN_VAL_5**, **DEMAGN_VAL_6**,
DEMAGN_VAL_7, **DEMAGN_VAL_8**, **DEMAGN_VAL_9**,
SIXSTEP_Base_InitTypeDef::demagn_value,
SIXSTEP_PI_PARAM_InitTypeDef_t::Reference, and
SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered.

Referenced by **MC_Bemf_Delay()**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

[Main Page](#)[Modules](#)[Data Structures](#)[Files](#)

Functions

Get_UART_data

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Get the UART value from DR register. [More...](#)

Functions

`uint32_t Get_UART_Data()`

Get the UART value from DR register. [More...](#)

Detailed Description

Get the UART value from DR register.

Function Documentation

uint32_t Get_UART_Data (void)

Get the UART value from DR register.

Return values

uint32_t

Definition at line **629** of file **stm32F401_nucleo_ihm11m1.c**.

References **UART**.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Exported_function_F401

[MIDDLEWARES](#) » [MC_6-STEP_LIB](#) »

[stm32F401_nucleo_ihm11m1](#)

Functions

void **MC_SixStep_ADC_Channel** (uint32_t)
API function for STM32 instruction. [More...](#)

void **MC_SixStep_Nucleo_Init** (void)
Init the STM32 register. [More...](#)

void **START_Ref_Generation** (void)

void **STOP_Ref_Generation** (void)

void **Set_Ref_Generation** (uint16_t)

void **START_DAC** (void)
Start DAC for debug. [More...](#)

void **STOP_DAC** (void)
Stop DAC for debug. [More...](#)

void **SET_DAC_value** (uint16_t)
Set DAC value for debug. [More...](#)

void **Bemf_delay_calc** (void)
Bemf delay calculation. [More...](#)

uint32_t **Get_UART_Data** (void)
Get the UART value from DR register. [More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D** (void)
Enable Input channel CH1 and CH2 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E** (void)
Enable Input channel CH1 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E** (void)
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D** (void)
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_Start_PWM_driving** (void)
Enable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_Stop_PWM_driving** (void)
Disable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1** (uint16_t)
Set the Duty Cycle value for CH1. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH2** (uint16_t)
Set the Duty Cycle value for CH2. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3** (uint16_t)
Set the Duty Cycle value for CH3. [More...](#)

void **MC_SixStep_Current_Reference_Start** (void)
Enable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Setvalue** (uint16_t)
Set the value for Current Reference. [More...](#)

void **BSP_X_NUCLEO_FAULT_LED_ON** (void)

void **BSP_X_NUCLEO_FAULT_LED_OFF** (void)

Detailed Description

Function Documentation

void Bemf_delay_calc(void)

Bemf delay calculation.

Return values

None

Definition at line [498](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [DEMAGN_VAL_1](#), [DEMAGN_VAL_10](#),
[DEMAGN_VAL_11](#), [DEMAGN_VAL_12](#), [DEMAGN_VAL_13](#),
[DEMAGN_VAL_14](#), [DEMAGN_VAL_2](#), [DEMAGN_VAL_3](#),
[DEMAGN_VAL_4](#), [DEMAGN_VAL_5](#), [DEMAGN_VAL_6](#),
[DEMAGN_VAL_7](#), [DEMAGN_VAL_8](#), [DEMAGN_VAL_9](#),
[SIXSTEP_Base_InitTypeDef::demagn_value](#),
[SIXSTEP_PI_PARAM_InitTypeDef_t::Reference](#), and
[SIXSTEP_Base_InitTypeDef::speed_fdbk_filtered](#).

Referenced by [MC_Bemf_Delay\(\)](#).

void BSP_X_NUCLEOFAULTLED OFF(void)

Definition at line [364](#) of file [X-NUCLEO-IHM11M1.c](#).

Referenced by [MC_StopMotor\(\)](#).

void BSP_X_NUCLEOFAULTLED ON(void)

Definition at line [351](#) of file [X-NUCLEO-IHM11M1.c](#).

Referenced by [MC_StartMotor\(\)](#).

uint32_t Get_UART_Data (void)

Get the UART value from DR register.

Return values

uint32_t

Definition at line [629](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [UART](#).

void MC_SixStep_ADC_Channel (uint32_t adc_ch)

API function for STM32 instruction.

API function for STM32 instruction.

Parameters

adc_ch

Return values

None

Definition at line [74](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [ADCx](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#), and
[MC_SixStep_NEXT_step\(\)](#).

void MC_SixStep_Current_Reference_Setvalue (uint16_t lref)

Set the value for Current Reference.

Return values

None

Definition at line [480](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Setvalue](#).

void MC_SixStep_Current_Reference_Start (void)

Enable the Current Reference generation.

Return values

None

Definition at line [441](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Start](#).

void MC_SixStep_Current_Reference_Stop (void)

Disable the Current Reference generation.

Return values

None

Definition at line [460](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Current_Reference_Stop](#).

void MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [327](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::DisableInput_CH1_D_CH2_D_C](#)

void MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E (void)

Enable Input channel CH2 and CH3 for STSPIN230.

Return values

None

Definition at line [309](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_D_CH2_E_CI](#)

void MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E (void)

Enable Input channel CH1 and CH3 for STSPIN230.

Return values

None

Definition at line [291](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_D_CI](#)

void MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D (void)

Enable Input channel CH1 and CH2 for STSPIN230.

Return values

None

Definition at line **273** of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::EnableInput_CH1_E_CH2_E_CI](#)

```
void  
MC_SixStep_HF_TIMx_SetDutyCycle_CH1 ( uint16_t CCR_value )
```

Set the Duty Cycle value for CH1.

Return values

None

Definition at line **381** of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH1](#).

```
void  
MC_SixStep_HF_TIMx_SetDutyCycle_CH2 ( uint16_t CCR_value )
```

Set the Duty Cycle value for CH2.

Return values

None

Definition at line **400** of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH2](#).

```
void  
MC_SixStep_HF_TIMx_SetDutyCycle_CH3( uint16_t CCR_value )
```

Set the Duty Cycle value for CH3.

Return values

None

Definition at line [423](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::HF_TIMx_SetDutyCycle_CH3](#).

```
void MC_SixStep_Nucleo_Init( void )
```

Init the STM32 register.

Return values

None

Definition at line [95](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References [ADC_Bemf_CH1](#), [ADC_Bemf_CH1_ST](#),
[ADC_Bemf_CH2](#), [ADC_Bemf_CH2_ST](#), [ADC_Bemf_CH3](#),
[ADC_Bemf_CH3_ST](#), [ADC_CH_1](#), [ADC_CH_1_ST](#), [ADC_CH_2](#),
[ADC_CH_2_ST](#), [ADC_CH_3](#), [ADC_CH_3_ST](#), [ADC_CH_4](#),
[ADC_CH_4_ST](#), [hadc1](#), [HF_TIMx](#), [HF_TIMx_CH1](#), [HF_TIMx_CH2](#),
[HF_TIMx_CH3](#), and [htim1](#).

Referenced by [MC_SixStep_INIT\(\)](#).

```
void MC_SixStep_Start_PWM_driving( void )
```

Enable PWM channels for STSPIN230.

Return values

None

Definition at line [345](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Start_PWM_driving](#).

void MC_SixStep_Stop_PWM_driving (void)

Disable PWM channels for STSPIN230.

Return values

None

Definition at line [363](#) of file [stm32F401_nucleo_ihm11m1.c](#).

References

[STSPIN230_MotorDriver_TypeDef::Stop_PWM_driving](#).

void SET_DAC_value (uint16_t dac_value)

Set DAC value for debug.

Parameters

dac_value information to plot through DAC

Return values

None

Definition at line [185](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_Task_Speed\(\)](#).

void Set_Ref_Generation (uint16_t)

void START_DAC (void)

Start DAC for debug.

Return values

None

Definition at line [154](#) of file [stm32F401_nucleo_ihm11m1.c](#).

Referenced by [MC_StartMotor\(\)](#).

void START_Ref_Generation (void)

void STOP_DAC (void)

Stop DAC for debug.

Return values

None

Definition at line [169](#) of file [stm32F401_nucleo_ihm11m1.c](#).

void STOP_Ref_Generation (void)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Modules

UART_UI

MIDDLEWARES

Serial communication through PC serial terminal. [More...](#)

Modules

Exported_function_Uart

Detailed Description

Serial communication through PC serial terminal.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

Functions

Exported_function_Uart

MIDDLEWARES » UART_UI

Functions

void **CMD_STARTM** (void)
UART function. More...

void **CMD_STOPMT** (void)

void **CMD_DIRECTION** (void)

void **CMD_SETSPD** (void)

void **CMD_GETSPD** (void)

void **CMD_STATUS** (void)

void **CMD_POTENZ** (void)

void **CMD_HELP** (void)

void **CMD_INIREF** (void)

void **CMD_POLESP** (void)

void **CMD_ACCELE** (void)

void **CMD_KP_PRM** (void)

void **CMD_KI_PRM** (void)

Detailed Description

Function Documentation

void CMD_ACCELE (void)

void CMD_DIRECTION (void)

void CMD_GETSPD (void)

void CMD_HELP (void)

void CMD_INIREF (void)

void CMD_KI_PRM (void)

void CMD_KP_PRM (void)

void CMD_POLESP (void)

void CMD_POTENZ (void)

void CMD_SETSPD (void)

void CMD_STARTM (void)

UART function.

void CMD_STATUS (void)

void CMD_STOPMT (void)

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
Data Structures	Data Structure Index	Data Fields	
<h2>Data Structures</h2>			

Here are the data structures with brief descriptions:

 CMD_T	
 SIXSTEP_Base_InitTypeDef	Six Step parameters
 SIXSTEP_PI_PARAM_InitTypeDef_t	Six PI regulator parameters
 STSPIN230_MotorDriver_TypeDef	



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files	
Data Structures	Data Structure Index	Data Fields	Data Fields	
<h2>CMD_T Struct Reference</h2>				

```
#include <UART_UI.h>
```

Data Fields

char **name** [10]

void(* **pCmdFunc**)(void)

Detailed Description

Definition at line **50** of file [UART_UI.h](#).

Field Documentation

char CMD_T::name[10]

Definition at line [51](#) of file [UART_UI.h](#).

void(* CMD_T::pCmdFunc) (void)

Definition at line [52](#) of file [UART_UI.h](#).

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

- [UART_UI.h](#)
-



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
Data Structures	Data Structure Index	Data Fields	

Data Structure Index

C | S

C

S

SIXSTEP_PI_PARAM_InitT
STSPIN230_MotorDriver_T

CMD_T SIXSTEP_Base_InitTypeDef

C | S

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files													
Data Structures		Data Structure Index	Data Fields													
All	Variables															
a	b	c	d	e	f	h	i	k	l	m	n	p	r	s	u	v

Here is a list of all struct and union fields with links to the structures/unions they belong to:

- a -

- ACCEL : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BEMF_threshold_DOWN : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BEMF_threshold_UP : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BUFFER : [SIXSTEP_Base_InitTypeDef](#)
- ADC-Regular_Buffer : [SIXSTEP_Base_InitTypeDef](#)
- ADC_SEQ_CHANNEL : [SIXSTEP_Base_InitTypeDef](#)
- ALIGN_OK : [SIXSTEP_Base_InitTypeDef](#)
- ALIGNMENT : [SIXSTEP_Base_InitTypeDef](#)
- ARR_OK : [SIXSTEP_Base_InitTypeDef](#)
- ARR_value : [SIXSTEP_Base_InitTypeDef](#)

- b -

- Bemf_delay_start : [SIXSTEP_Base_InitTypeDef](#)
- BEMF_OK : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_1 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_2 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_3 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_4 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_5 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_6 : [SIXSTEP_Base_InitTypeDef](#)
- BEMF_Tdown_count : [SIXSTEP_Base_InitTypeDef](#)

- Button_ready : **SIXSTEP_Base_InitTypeDef**

- c -

- CL_READY : **SIXSTEP_Base_InitTypeDef**
- CMD : **SIXSTEP_Base_InitTypeDef**
- Current_Reference : **SIXSTEP_Base_InitTypeDef**
- Current_Reference_Setvalue :
STSPIN230_MotorDriver_TypeDef
- Current_Reference_Start : **STSPIN230_MotorDriver_TypeDef**
- Current_Reference_Stop : **STSPIN230_MotorDriver_TypeDef**
- CurrentRegular_BEMF_ch : **SIXSTEP_Base_InitTypeDef**
- CW_CCW : **SIXSTEP_Base_InitTypeDef**

- d -

- demagn_counter : **SIXSTEP_Base_InitTypeDef**
- demagn_value : **SIXSTEP_Base_InitTypeDef**
- DisableInput_CH1_D_CH2_D_CH3_D :
STSPIN230_MotorDriver_TypeDef

- e -

- EnableInput_CH1_D_CH2_E_CH3_E :
STSPIN230_MotorDriver_TypeDef
- EnableInput_CH1_E_CH2_D_CH3_E :
STSPIN230_MotorDriver_TypeDef
- EnableInput_CH1_E_CH2_E_CH3_D :
STSPIN230_MotorDriver_TypeDef

- f -

- filter_depth : **SIXSTEP_Base_InitTypeDef**

- h -

- HF_TIMx_ARR : **SIXSTEP_Base_InitTypeDef**
- HF_TIMx_CCR : **SIXSTEP_Base_InitTypeDef**
- HF_TIMx_PSC : **SIXSTEP_Base_InitTypeDef**

- HF_TIMx_SetDutyCycle_CH1 : **STSPIN230_MotorDriver_TypeDef**
- HF_TIMx_SetDutyCycle_CH2 : **STSPIN230_MotorDriver_TypeDef**
- HF_TIMx_SetDutyCycle_CH3 : **STSPIN230_MotorDriver_TypeDef**

- i -

- Integral_Term_sum : **SIXSTEP_Base_InitTypeDef**
- IREFERENCE : **SIXSTEP_Base_InitTypeDef**
- Ireference : **SIXSTEP_Base_InitTypeDef**

- k -

- KI : **SIXSTEP_Base_InitTypeDef**
- Ki_Gain : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- KP : **SIXSTEP_Base_InitTypeDef**
- Kp_Gain : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- l -

- LF_TIMx_ARR : **SIXSTEP_Base_InitTypeDef**
- LF_TIMx_PSC : **SIXSTEP_Base_InitTypeDef**
- Lower_Limit_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- m -

- Max_PID_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- MediumFrequencyTask_flag : **SIXSTEP_Base_InitTypeDef**
- Min_PID_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- n -

- name : **CMD_T**
- numberofitemArr : **SIXSTEP_Base_InitTypeDef**
- NUMPOLESPAIRS : **SIXSTEP_Base_InitTypeDef**

- p -

- pCmdFunc : **CMD_T**
- Potentiometer : **SIXSTEP_Base_InitTypeDef**
- prescaler_value : **SIXSTEP_Base_InitTypeDef**
- pulse_value : **SIXSTEP_Base_InitTypeDef**

- r -

- Ramp_Start : **SIXSTEP_Base_InitTypeDef**
- Reference : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- Regular_channel : **SIXSTEP_Base_InitTypeDef**
- RUN_Motor : **SIXSTEP_Base_InitTypeDef**

- s -

- speed_fdbk : **SIXSTEP_Base_InitTypeDef**
- speed_fdbk_filtered : **SIXSTEP_Base_InitTypeDef**
- Speed_Loop_Time : **SIXSTEP_Base_InitTypeDef**
- Speed_Ref_filtered : **SIXSTEP_Base_InitTypeDef**
- Speed_target_ramp : **SIXSTEP_Base_InitTypeDef**
- Speed_target_time : **SIXSTEP_Base_InitTypeDef**
- SPEED_VALIDATED : **SIXSTEP_Base_InitTypeDef**
- Start_PWM_driving : **STSPIN230_MotorDriver_TypeDef**
- STATUS : **SIXSTEP_Base_InitTypeDef**
- status_prev : **SIXSTEP_Base_InitTypeDef**
- step_position : **SIXSTEP_Base_InitTypeDef**
- Stop_PWM_driving : **STSPIN230_MotorDriver_TypeDef**
- SYSCLK_frequency : **SIXSTEP_Base_InitTypeDef**

- u -

- Uart_cmd_to_set : **SIXSTEP_Base_InitTypeDef**
- Uart_value_to_set : **SIXSTEP_Base_InitTypeDef**
- Upper_Limit_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- v -

- VALIDATION_OK : **SIXSTEP_Base_InitTypeDef**
-

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files													
Data Structures		Data Structure Index	Data Fields													
All	Variables															
a	b	c	d	e	f	h	i	k	l	m	n	p	r	s	u	v

- a -

- ACCEL : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BEMF_threshold_DOWN : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BEMF_threshold_UP : [SIXSTEP_Base_InitTypeDef](#)
- ADC_BUFFER : [SIXSTEP_Base_InitTypeDef](#)
- ADC-Regular_Buffer : [SIXSTEP_Base_InitTypeDef](#)
- ADC_SEQ_CHANNEL : [SIXSTEP_Base_InitTypeDef](#)
- ALIGN_OK : [SIXSTEP_Base_InitTypeDef](#)
- ALIGNMENT : [SIXSTEP_Base_InitTypeDef](#)
- ARR_OK : [SIXSTEP_Base_InitTypeDef](#)
- ARR_value : [SIXSTEP_Base_InitTypeDef](#)

- b -

- Bemf_delay_start : [SIXSTEP_Base_InitTypeDef](#)
- BEMF_OK : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_1 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_2 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_3 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_4 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_5 : [SIXSTEP_Base_InitTypeDef](#)
- bemf_state_6 : [SIXSTEP_Base_InitTypeDef](#)
- BEMF_Tdown_count : [SIXSTEP_Base_InitTypeDef](#)
- Button_ready : [SIXSTEP_Base_InitTypeDef](#)

- c -

- CL_READY : **SIXSTEP_Base_InitTypeDef**
- CMD : **SIXSTEP_Base_InitTypeDef**
- Current_Reference : **SIXSTEP_Base_InitTypeDef**
- Current_Reference_Setvalue :
STSPIN230_MotorDriver_TypeDef
- Current_Reference_Start : **STSPIN230_MotorDriver_TypeDef**
- Current_Reference_Stop : **STSPIN230_MotorDriver_TypeDef**
- CurrentRegular_BEMF_ch : **SIXSTEP_Base_InitTypeDef**
- CW_CCW : **SIXSTEP_Base_InitTypeDef**

- d -

- demagn_counter : **SIXSTEP_Base_InitTypeDef**
- demagn_value : **SIXSTEP_Base_InitTypeDef**
- DisableInput_CH1_D_CH2_D_CH3_D :
STSPIN230_MotorDriver_TypeDef

- e -

- EnableInput_CH1_D_CH2_E_CH3_E :
STSPIN230_MotorDriver_TypeDef
- EnableInput_CH1_E_CH2_D_CH3_E :
STSPIN230_MotorDriver_TypeDef
- EnableInput_CH1_E_CH2_E_CH3_D :
STSPIN230_MotorDriver_TypeDef

- f -

- filter_depth : **SIXSTEP_Base_InitTypeDef**

- h -

- HF_TIMx_ARR : **SIXSTEP_Base_InitTypeDef**
- HF_TIMx_CCR : **SIXSTEP_Base_InitTypeDef**
- HF_TIMx_PSC : **SIXSTEP_Base_InitTypeDef**
- HF_TIMx_SetDutyCycle_CH1 :
STSPIN230_MotorDriver_TypeDef

- HF_TIMx_SetDutyCycle_CH2 : **STSPIN230_MotorDriver_TypeDef**
- HF_TIMx_SetDutyCycle_CH3 : **STSPIN230_MotorDriver_TypeDef**

- i -

- Integral_Term_sum : **SIXSTEP_Base_InitTypeDef**
- IREFERENCE : **SIXSTEP_Base_InitTypeDef**
- Ireference : **SIXSTEP_Base_InitTypeDef**

- k -

- KI : **SIXSTEP_Base_InitTypeDef**
- Ki_Gain : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- KP : **SIXSTEP_Base_InitTypeDef**
- Kp_Gain : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- l -

- LF_TIMx_ARR : **SIXSTEP_Base_InitTypeDef**
- LF_TIMx_PSC : **SIXSTEP_Base_InitTypeDef**
- Lower_Limit_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- m -

- Max_PID_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- MediumFrequencyTask_flag : **SIXSTEP_Base_InitTypeDef**
- Min_PID_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- n -

- name : **CMD_T**
- numberofitemArr : **SIXSTEP_Base_InitTypeDef**
- NUMPOLESPAIRS : **SIXSTEP_Base_InitTypeDef**

- p -

- pCmdFunc : **CMD_T**

- Potentiometer : **SIXSTEP_Base_InitTypeDef**
- prescaler_value : **SIXSTEP_Base_InitTypeDef**
- pulse_value : **SIXSTEP_Base_InitTypeDef**

- r -

- Ramp_Start : **SIXSTEP_Base_InitTypeDef**
- Reference : **SIXSTEP_PI_PARAM_InitTypeDef_t**
- Regular_channel : **SIXSTEP_Base_InitTypeDef**
- RUN_Motor : **SIXSTEP_Base_InitTypeDef**

- s -

- speed_fdbk : **SIXSTEP_Base_InitTypeDef**
- speed_fdbk_filtered : **SIXSTEP_Base_InitTypeDef**
- Speed_Loop_Time : **SIXSTEP_Base_InitTypeDef**
- Speed_Ref_filtered : **SIXSTEP_Base_InitTypeDef**
- Speed_target_ramp : **SIXSTEP_Base_InitTypeDef**
- Speed_target_time : **SIXSTEP_Base_InitTypeDef**
- SPEED_VALIDATED : **SIXSTEP_Base_InitTypeDef**
- Start_PWM_driving : **STSPIN230_MotorDriver_TypeDef**
- STATUS : **SIXSTEP_Base_InitTypeDef**
- status_prev : **SIXSTEP_Base_InitTypeDef**
- step_position : **SIXSTEP_Base_InitTypeDef**
- Stop_PWM_driving : **STSPIN230_MotorDriver_TypeDef**
- SYSCLK_frequency : **SIXSTEP_Base_InitTypeDef**

- u -

- Uart_cmd_to_set : **SIXSTEP_Base_InitTypeDef**
- Uart_value_to_set : **SIXSTEP_Base_InitTypeDef**
- Upper_Limit_Output : **SIXSTEP_PI_PARAM_InitTypeDef_t**

- v -

- VALIDATION_OK : **SIXSTEP_Base_InitTypeDef**



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files		
File List	Globals				
<h2>File List</h2>					
Here is a list of all files with brief descriptions:					
6Step_Lib.c	This file provides the set of functions for Motor Control library				
6Step_Lib.h	This header file provides the set of functions for Motor Control library				
main_F401.c	This file provides a set of functions needed to configure STM32 MCU				
main_F401.h	Main program body for STM32F401xx				
MC_Common.h	This header file is a common file				
MC_SixStep_param.h	This header file provides all parameters to driver a motor with 6Step library				
stm32_nucleo_ihm11m1.h	This file provides the interface between the MC-lib and STM Nucleo				
stm32F401_nucleo_ihm11m1.c	This file provides the interface between the MC-lib and STM Nucleo F401xx				
stm32F401_nucleo_ihm11m1.h	This file provides the interface				

	between the MC-lib and STM Nucleo
stm32f4xx_hal_conf.h	HAL configuration file
stm32f4xx_hal_msp.c	This file provides code for the MSP Initialization and de-Initialization codes
stm32f4xx_it.c	Interrupt Service Routines
stm32f4xx_it.h	This file contains the headers of the interrupt handlers
STSPIN230.c	This file provides a set of functions to manage STSPIN230 driver
STSPIN230.h	This file provides a set of functions to manage STSPIN230 driver
UART_UI.c	This file provides a set of functions needed to manage the UART com
UART_UI.h	This file provides a set of functions needed to manage the UART com
X-NUCLEO-IHM11M1.c	This file provides the set of functions to manage the X-Nucleo expansion board
X-NUCLEO-IHM11M1.h	This file provides the set of functions to manage the X-Nucleo board



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Src

Functions | Variables

6Step_Lib.c File Reference

This file provides the set of functions for Motor Control library. [More...](#)

```
#include "6Step_Lib.h" #include <string.h>
```

[Go to the source code of this file.](#)

Functions

int16_t **MC_PI_Controller** (SIXSTEP_PI_PARAM_InitTypeDef_t *,
int16_t)

uint16_t **MC_Potentiometer_filter** (uint16_t)

uint64_t **MCM_Sqrt** (uint64_t wInput)

It calculates the square root of a non-negative s64. More...

int32_t **MC_GetEISpeedHz** (void)

int32_t **MC_GetMechSpeedRPM** (void)

void **MC_SixStep_NEXT_step** (void)

void **MC_Speed_Filter** (void)

void **MC_SixStep_ARR_step** (void)

void **MC_SixStep_TABLE** (uint8_t)

void **MC_SixStep_Speed_Potentiometer** (void)

void **MC_Set_PI_param** (SIXSTEP_PI_PARAM_InitTypeDef_t *)

void **MC_Task_Speed** (void)

void **MC_SixStep_Alignment** (void)

void **MC_Bemf_Delay** (void)

void **MC_TIMx_SixStep_timebase** (void)

void **MC_ADCx_SixStep_Bemf** (void)

void **MC_SysTick_SixStep_MediumFrequencyTask** (void)

void **MC_SixStep_Ramp_Motor_calc** (void)

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D** (void)

Enable Input channel CH1 and CH2 for STSPIN230.

[More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E** (void)

Enable Input channel CH1 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E** (void)

Enable Input channel CH2 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D** (void)

Enable Input channel CH2 and CH3 for STSPIN230.

[More...](#)

void **MC_SixStep_Start_PWM_driving** (void)

Enable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_Stop_PWM_driving** (void)

Disable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1** (uint16_t)

Set the Duty Cycle value for CH1. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH2** (uint16_t)

Set the Duty Cycle value for CH2. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3** (uint16_t)

Set the Duty Cycle value for CH3. [More...](#)

void **MC_SixStep_Current_Reference_Start** (void)

Enable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Setvalue** (uint16_t)
Set the value for Current Reference. [More...](#)

void **MC_SixStep_ARR_Bemf** (uint8_t)

void **MC_UI_INIT** (void)

void **UART_Set_Value** (void)

void **UART_Communication_Task** (void)

void **MC_SixStep_Init_main_data** (void)

void **CMD_Parser** (char *pCommandString)

void **MC_SixStep_Speed_Val_target_potentiometer** (void)

void **MC_SixStep_RESET** ()

void **MC_Set_Speed** (uint16_t speed_value)

void **MC_StartMotor** ()

void **MC_StopMotor** ()

void **MC_SixStep_INIT** ()

void **MC_EXT_button_SixStep** ()

void **HAL_IncTick** (void)

This function is called to increment a global variable
"uwTick" used as application time base. [More...](#)

```
uint32_t HAL_GetTick (void)
```

Povides a tick value in millisecond. [More...](#)

Variables

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

SIXSTEP_PI_PARAM_InitTypeDef_t PI_parameters

uint16_t Rotor_poles_pairs

uint32_t mech_accel_hz = 0

uint32_t constant_k = 0

uint32_t Time_vector_tmp = 0

uint32_t Time_vector_prev_tmp = 0

uint32_t T_single_step = 0

uint32_t T_single_step_first_value =
0

int32_t delta = 0

uint16_t index_array = 1

int16_t speed_tmp_array
[FILTER_DEEP]

uint16_t speed_tmp_buffer
[FILTER_DEEP]

uint16_t HFBuffer [HFBUFFERSIZE]

uint16_t HFBufferIndex = 0

uint8_t array_completed = FALSE

```
    uint8_t buffer_completed = FALSE
```

```
    uint8_t UART_FLAG_RECEIVE =  
        FALSE
```

```
    uint32_t ARR_LF = 0
```

```
    int32_t Mech_Speed_RPM = 0
```

```
    int32_t El_Speed_Hz = 0
```

```
    uint16_t index_adc_chn = 0
```

```
    uint16_t index_motor_run = 0
```

```
    uint16_t test_motor_run = 1
```

```
    uint8_t Enable_start_button = TRUE
```

```
    uint16_t index_ARR_step = 1
```

```
    uint32_t n_zcr_startup = 0
```

```
    uint16_t index_startup_motor = 1
```

```
    uint16_t target_speed =  
        TARGET_SPEED
```

```
    uint16_t shift_n_sqrt = 14
```

```
    uint16_t cnt_bemf_event = 0
```

```
    uint8_t startup_bemf_failure = 0
```

```
    uint8_t speed_fdbk_error = 0
```

__IO uint32_t uwTick = 0

uint8_t dac_status = DAC_ENABLE

uint16_t index_align = 1

int32_t speed_sum_sp_filt = 0

int32_t speed_sum_pot_filt = 0

uint16_t index_pot_filt = 1

int16_t potent_filtered = 0

uint32_t Tick_cnt = 0

uint32_t counter.ARR_Bemf = 0

uint64_t constant_multiplier_tmp = 0

Detailed Description

This file provides the set of functions for Motor Control library.

Author

System lab - Automation and Motion control team

Version

V1.0.0

Date

06-July-2015

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [6Step_Lib.c](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Inc

Data Structures | Typedefs | Enumerations |
Functions

6Step_Lib.h File Reference

This header file provides the set of functions for Motor Control library.
[More...](#)

```
#include "stm32_nucleo_ihm11m1.h" #include "math.h"  
#include "stdlib.h"  
#include "stdio.h"
```

[Go to the source code of this file.](#)

Data Structures

struct **SIXSTEP_Base_InitTypeDef**

Six Step parameters. More...

struct **SIXSTEP_PI_PARAM_InitTypeDef_t**

Six PI regulator parameters. More...

TypeDefs

```
typedef struct SIXSTEP_PI_PARAM_InitTypeDef_t * SIXSTEP_pi_PA
```

Enumerations

```
enum SIXSTEP_Base_SystStatus_t {
    IDLE, STARTUP, VALIDATION, STOP,
    START, RUN, ALIGNMENT, SPEEDFBKERROR,
    OVERCURRENT, STARTUP_FAILURE,
    STARTUP_BEMF_FAILURE
}
Six Step parameters. More...
```

Functions

```
void MC_SixStep_INIT (void)
```

```
void MC_SixStep_RESET (void)
```

```
void MC_StartMotor (void)
```

```
void MC_StopMotor (void)
```

```
void MC_Set_Speed (uint16_t)
```

```
void MC_EXT_button_SixStep (void)
```

Detailed Description

This header file provides the set of functions for Motor Control library.

Author

System lab

Version

V1.0.0

Date

06-July-2015

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [6Step_Lib.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Src | Functions | Variables

main_F401.c File Reference

This file provides a set of functions needed to configure STM32 MCU.
[More...](#)

```
#include "stm32f4xx_hal.h" #include "6Step_Lib.h"
```

[Go to the source code of this file.](#)

Functions

void **SystemClock_Config** (void)
System Clock Configuration. [More...](#)

static void **MX_ADC1_Init** (void)

static void **MX_TIM1_Init** (void)

static void **MX_TIM3_Init** (void)

static void **MX_TIM4_Init** (void)

static void **MX_USART2_UART_Init** (void)

int **main** (void)

Variables

ADC_HandleTypeDef **hadc1**

TIM_HandleTypeDef **htim1**

TIM_HandleTypeDef **htim2**

TIM_HandleTypeDef **htim3**

TIM_HandleTypeDef **htim4**

UART_HandleTypeDef **huart2**

Detailed Description

This file provides a set of functions needed to configure STM32 MCU.

Author

IPC

Version

V0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [main_F401.c](#).

Function Documentation

int main (void)

How to use the 6Step FW Ex

This workspace contains the middleware layer with Motor Control library performing a 6-step control algorithm allowing the motor speed regulation based on 1shunt current sensing mode and sensorless algorithm for bE STM32Fxx-Nucleo in four different configurations, normal, demo, comm blue button event to start the motor, the "demo" mode starts and stop the communication protocol with external PC terminal and the "boot" mode

A list of APIs is provided to send command to 6Step lib, for instance:

(#) **MC_StartMotor()** -> Start the motor

(#) MC_StopMotor() -> Stop the motor

(#) MC_Set_Speed(...) -> Set the new motor speed

The **MC_SixStep_param.h** contains the full list of MC parameters

USER SPACE

Definition at line [74](#) of file [main_F401.c](#).

References [MC_SixStep_Init\(\)](#), [MX_ADC1_Init\(\)](#), [MX_TIM1_Init\(\)](#), [MX_TIM3_Init\(\)](#), [MX_TIM4_Init\(\)](#), [MX_USART2_UART_Init\(\)](#), and [SystemClock_Config\(\)](#).

void MX_ADC1_Init (void)

static

Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of conversion)

Configure for the selected ADC regular channel its corresponding rank in the sequencer and its sample time.

Definition at line [180](#) of file [main_F401.c](#).

References [hadc1](#).

Referenced by [main\(\)](#).

void MX_TIM1_Init (void)

static

Definition at line [213](#) of file [main_F401.c](#).

References [htim1](#).

Referenced by [main\(\)](#).

void MX_TIM3_Init (void)

static

Definition at line [271](#) of file [main_F401.c](#).

References [htim3](#).

Referenced by [main\(\)](#).

void MX_TIM4_Init (void)

static

Definition at line [304](#) of file [main_F401.c](#).

References [htim4](#).

Referenced by [main\(\)](#).

void MX_USART2_UART_Init (void)

static

Definition at line [327](#) of file [main_F401.c](#).

References [huart2](#).

Referenced by [main\(\)](#).

void SystemClock_Config (void)

System Clock Configuration.

Definition at line [149](#) of file [main_F401.c](#).

Referenced by [main\(\)](#).

Variable Documentation

ADC_HandleTypeDef hadc1

Definition at line [46](#) of file [main_F401.c](#).

Referenced by [ADC_IRQHandler\(\)](#), [MC_SixStep_Nucleo_Init\(\)](#), and [MX_ADC1_Init\(\)](#).

TIM_HandleTypeDef htim1

Definition at line [48](#) of file [main_F401.c](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), [MX_TIM1_Init\(\)](#), and [TIM1_BRK_TIM9_IRQHandler\(\)](#).

TIM_HandleTypeDef htim2

Definition at line [49](#) of file [main_F401.c](#).

TIM_HandleTypeDef htim3

Definition at line [50](#) of file [main_F401.c](#).

Referenced by [MX_TIM3_Init\(\)](#).

TIM_HandleTypeDef htim4

Definition at line [51](#) of file [main_F401.c](#).

Referenced by [MX_TIM4_Init\(\)](#), and [TIM4_IRQHandler\(\)](#).

UART_HandleTypeDef huart2

Definition at line [53](#) of file [main_F401.c](#).

Referenced by [MX_USART2_UART_Init\(\)](#), and [USART2_IRQHandler\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Variables

main_F401.h File Reference

Main program body for STM32F401xx. [More...](#)

[Go to the source code of this file.](#)

Variables

ADC_HandleTypeDef **hadc1**

TIM_HandleTypeDef **htim1**

TIM_HandleTypeDef **htim3**

TIM_HandleTypeDef **htim4**

UART_HandleTypeDef **huart2**

Detailed Description

Main program body for STM32F401xx.

Author

IPC

Version

V0

Date

10/07/2016 COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [main_F401.h](#).

Variable Documentation

ADC_HandleTypeDef hadc1

Definition at line [46](#) of file [main_F401.c](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and [MX_ADC1_Init\(\)](#).

TIM_HandleTypeDef htim1

Definition at line [48](#) of file [main_F401.c](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and [MX_TIM1_Init\(\)](#).

TIM_HandleTypeDef htim3

Definition at line [50](#) of file [main_F401.c](#).

Referenced by [MX_TIM3_Init\(\)](#).

TIM_HandleTypeDef htim4

Definition at line [51](#) of file [main_F401.c](#).

Referenced by [MX_TIM4_Init\(\)](#).

UART_HandleTypeDef huart2

Definition at line [53](#) of file [main_F401.c](#).

Referenced by [MX_USART2_UART_Init\(\)](#).

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	Common

Data Structures

MC_Common.h File Reference

This header file is a common file. More...

[Go to the source code of this file.](#)

Data Structures

```
struct STSPIN230_MotorDriver_TypeDef
```

Detailed Description

This header file is a common file.

Author

System lab - Automation and Motion control team

Version

V1.0.0

Date

06-July-2015

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [MC_Common.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Macros

MC_SixStep_param.h File Reference

This header file provides all parameters to driver a motor with 6Step library. [More...](#)

[Go to the source code of this file.](#)

Macros

```
#define NUM_POLE_PAIRS 2  
  
#define DIRECTION 0  
  
#define TARGET_SPEED 4000  
  
#define POTENTIOMETER 1  
  
#define STARTUP_DUTY_CYCLE 600  
  
#define STARTUP_CURRENT_REFERENCE STARTUP_DUTY_CY  
  
#define ACC 1000000  
  
#define MINIMUM_ACC 1000  
  
#define NUMBER_OF_STEPS 20000  
  
#define TIME_FOR_ALIGN 500  
  
#define BUTTON_DELAY 1000  
  
#define NUMBER_ZCR 12  
  
#define SPEED_LOOP_TIME 1  
  
#define KP_GAIN 500  
  
#define KI_GAIN 50  
  
#define KP_DIV 4096  
  
#define KI_DIV 4096
```

```
#define LOWER_OUT_LIMIT 50
#define UPPER_OUT_LIMIT 800
#define MAX_POT_SPEED 20000
#define MIN_POT_SPEED 1700
#define VAL_POT_SPEED_DIV 2
#define INITIAL_DEMAGN_DELAY 10
#define BEMF_THRSLD_DOWN 200
#define BEMF_THRSLD_UP 200
#define FILTER_DEEP 20
#define HFBUFFERSIZE 10
#define ADC_SPEED_TH 82
#define BEMF_CONSEC_DOWN_MAX 10
#define BEMF_CNT_EVENT_MAX 100
#define GPIO_ZERO_CROSS 1
#define GPIO_COMM 1
#define DEMO_START_TIME 5000
#define DEMO_STOP_TIME 2000
#define DEMAGN_VAL_1 1
#define DEMAGN_VAL_2 2
```

```
#define DEMAGN_VAL_3 3  
  
#define DEMAGN_VAL_4 4  
  
#define DEMAGN_VAL_5 5  
  
#define DEMAGN_VAL_6 6  
  
#define DEMAGN_VAL_7 7  
  
#define DEMAGN_VAL_8 8  
  
#define DEMAGN_VAL_9 9  
  
#define DEMAGN_VAL_10 10  
  
#define DEMAGN_VAL_11 11  
  
#define DEMAGN_VAL_12 12  
  
#define DEMAGN_VAL_13 13  
  
#define DEMAGN_VAL_14 14  
  
#define TRUE 1  
  
#define FALSE 0
```

Detailed Description

This header file provides all parameters to driver a motor with 6Step library.

Author

Version

V0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [MC_SixStep_param.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Inc

stm32_nucleo_ihm11m1.h File Reference

This file provides the interface between the MC-lib and STM Nucleo.
[More...](#)

[Go to the source code of this file.](#)

Detailed Description

This file provides the interface between the MC-lib and STM Nucleo.

Author

System lab - Automation and Motion control team

Version

V1.0.0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32_nucleo_ihm11m1.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Functions | Variables

stm32F401_nucleo_ihm11m1.c File Reference

This file provides the interface between the MC-lib and STM Nucleo F401xx. [More...](#)

```
#include "stm32F401_nucleo_ihm11m1.h" #include "6Step_Lib.h"  
#include "X-NUCLEO-IHM11M1.h"
```

[Go to the source code of this file.](#)

Functions

void **MC_ADCx_SixStep_Bemf** (void)

void **MC_TIMx_SixStep_timebase** (void)

void **MC_SysTick_SixStep_MediumFrequencyTask** (void)

void **MC_SixStep_ADC_Channel** (uint32_t adc_ch)
Select the new ADC Channel. [More...](#)

void **MC_SixStep_Nucleo_Init** ()
Init the STM32 register. [More...](#)

void **START_DAC** ()
Start DAC for debug. [More...](#)

void **STOP_DAC** ()
Stop DAC for debug. [More...](#)

void **SET_DAC_value** (uint16_t dac_value)
Set DAC value for debug. [More...](#)

void **HAL_ADC_ConvCpltCallback** (ADC_HandleTypeDef *hadc)
ADC callback. [More...](#)

void **HAL_TIM_PeriodElapsedCallback** (TIM_HandleTypeDef *htim)
htim callback [More...](#)

void **HAL_SYSTICK_Callback** ()
Systick callback. [More...](#)

void **HAL_GPIO_EXTI_Callback** (uint16_t GPIO_Pin)
EXT callback. [More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D ()**
Enable Input channel CH1 and CH2 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E ()**
Enable Input channel CH1 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E ()**
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D ()**
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_Start_PWM_driving ()**
Enable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_Stop_PWM_driving ()**
Disable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)**
Set the Duty Cycle value for CH1. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)**
Set the Duty Cycle value for CH2. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)**
Set the Duty Cycle value for CH3. [More...](#)

void **MC_SixStep_Current_Reference_Start ()**
Enable the Current Reference generation. [More...](#)

```
void MC_SixStep_Current_Reference_Stop ()
```

Disable the Current Reference generation. More...

```
void MC_SixStep_Current_Reference_Setvalue (uint16_t lref)
```

Set the value for Current Reference. More...

```
void Bemf_delay_calc ()
```

Bemf delay calculation. More...

```
uint32_t Get_UART_Data ()
```

Get the UART value from DR register. More...

Variables

SIXSTEP_Base_InitTypeDef **SIXSTEP_parameters**

SIXSTEP_PI_PARAM_InitTypeDef_t **PI_parameters**

STSPIN230_MotorDriver_TypeDef **STSPIN230MotorDriver**

It handles all API functions for STSPIN230 MC Driver.

[More...](#)

Detailed Description

This file provides the interface between the MC-lib and STM Nucleo F401xx.

Author

IPC

Version

V0

Date

10/07/2016 COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32F401_nucleo_ihm11m1.c](#).

Generated by  doxygen 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Macros | Functions

stm32f401_nucleo_ihm11m1.h File Reference

This file provides the interface between the MC-lib and STM Nucleo.
[More...](#)

```
#include "stm32f4xx_hal.h" #include "main_F401.h"
```

[Go to the source code of this file.](#)

Macros

```
#define HF_TIMx htim1
```

```
#define LF_TIMx htim4
```

```
#define HALL_ENCODER_TIMx htim2
```

```
#define ADCx hadc1
```

```
#define DACx htim3
```

```
#define UART huart2
```

```
#define ADC_CH_1 ADC_CHANNEL_0 /*CURRENT*/
```

```
#define ADC_CH_2 ADC_CHANNEL_12 /*SPEED*/
```

```
#define ADC_CH_3 ADC_CHANNEL_1 /*VBUS*/
```

```
#define ADC_CH_4 ADC_CHANNEL_2 /*TEMP */
```

```
#define ADC_Bemf_CH1 ADC_CHANNEL_13 /*BEMF1*/
```

```
#define ADC_Bemf_CH2 ADC_CHANNEL_8 /*BEMF2*/
```

```
#define ADC_Bemf_CH3 ADC_CHANNEL_7 /*BEMF3*/
```

```
#define ADC_CH_1_ST ADC_SAMPLETIME_3CYCLES  
/*CURRENT sampling time */
```

```
#define ADC_CH_2_ST ADC_SAMPLETIME_84CYCLES /*SPEED  
sampling time*/
```

```
#define ADC_CH_3_ST ADC_SAMPLETIME_84CYCLES /*VBUS  
sampling time*/
```

```
#define ADC_CH_4_ST ADC_SAMPLETIME_84CYCLES /*TEMP  
sampling time*/  
  
#define ADC_Bemf_CH1_ST ADC_SAMPLETIME_28CYCLES  
/*BEMF1 sampling time*/  
  
#define ADC_Bemf_CH2_ST ADC_SAMPLETIME_28CYCLES  
/*BEMF2 sampling time*/  
  
#define ADC_Bemf_CH3_ST ADC_SAMPLETIME_28CYCLES  
/*BEMF3 sampling time*/  
  
#define HF_TIMx_CH1 TIM_CHANNEL_1  
  
#define HF_TIMx_CH2 TIM_CHANNEL_2  
  
#define HF_TIMx_CH3 TIM_CHANNEL_3  
  
#define HF_TIMx_CCR1 CCR1 /*Channel 1*/  
  
#define HF_TIMx_CCR2 CCR2 /*Channel 2*/  
  
#define HF_TIMx_CCR3 CCR3 /*Channel 3*/  
  
#define DAC_ENABLE 0  
  
#define GPIO_PORT_ZCR GPIOC  
  
#define GPIO_CH_ZCR GPIO_PIN_12  
  
#define GPIO_PORT_COMM GPIOC  
  
#define GPIO_CH_COMM GPIO_PIN_10  
  
#define STARTM_CMD 0
```

```
#define STOPMT_CMD 1
```

```
#define SETSPD_CMD 2
```

```
#define GETSPD_CMD 3
```

```
#define INIREF_CMD 4
```

```
#define POLESP_CMD 5
```

```
#define ACCELE_CMD 6
```

```
#define KP_PRM_CMD 7
```

```
#define KI_PRM_CMD 8
```

```
#define POTENZ_CMD 9
```

```
#define HELP_CMD 10
```

```
#define STATUS_CMD 11
```

```
#define DIRECT_CMD 12
```

Functions

void **MC_SixStep_ADC_Channel** (uint32_t)
API function for STM32 instruction. [More...](#)

void **MC_SixStep_Nucleo_Init** (void)
Init the STM32 register. [More...](#)

void **START_Ref_Generation** (void)

void **STOP_Ref_Generation** (void)

void **Set_Ref_Generation** (uint16_t)

void **START_DAC** (void)
Start DAC for debug. [More...](#)

void **STOP_DAC** (void)
Stop DAC for debug. [More...](#)

void **SET_DAC_value** (uint16_t)
Set DAC value for debug. [More...](#)

void **Bemf_delay_calc** (void)
Bemf delay calculation. [More...](#)

uint32_t **Get_UART_Data** (void)
Get the UART value from DR register. [More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D** (void)
Enable Input channel CH1 and CH2 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E** (void)
Enable Input channel CH1 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E** (void)
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D** (void)
Enable Input channel CH2 and CH3 for STSPIN230.
[More...](#)

void **MC_SixStep_Start_PWM_driving** (void)
Enable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_Stop_PWM_driving** (void)
Disable PWM channels for STSPIN230. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH1** (uint16_t)
Set the Duty Cycle value for CH1. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH2** (uint16_t)
Set the Duty Cycle value for CH2. [More...](#)

void **MC_SixStep_HF_TIMx_SetDutyCycle_CH3** (uint16_t)
Set the Duty Cycle value for CH3. [More...](#)

void **MC_SixStep_Current_Reference_Start** (void)
Enable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **MC_SixStep_Current_Reference_Setvalue** (uint16_t)
Set the value for Current Reference. [More...](#)

void **BSP_X_NUCLEO_FAULT_LED_ON** (void)

void **BSP_X_NUCLEO_FAULT_LED_OFF** (void)

Detailed Description

This file provides the interface between the MC-lib and STM Nucleo.

Author

IPC

Version

V0

Date

10/08/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32F401_nucleo_ihm11m1.h](#).

Macro Definition Documentation

#define ACCELE_CMD 6

Set the Accelleration for Start-up of the motor command received

Definition at line [88](#) of file [stm32F401_nucleo_ihm11m1.h](#).

#define ADC_Bemf_CH1 ADC_CHANNEL_13 /*BEMF1*/

Definition at line [56](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

**#define ADC_Bemf_CH1_ST ADC_SAMPLETIME_28CYCLES
/*BEMF1 sampling time*/**

Definition at line [64](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

#define ADC_Bemf_CH2 ADC_CHANNEL_8 /*BEMF2*/

Definition at line [57](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

**#define ADC_Bemf_CH2_ST ADC_SAMPLETIME_28CYCLES
/*BEMF2 sampling time*/**

Definition at line [65](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADC_Bemf_CH3  ADC_CHANNEL_7 /*BEMF3*/
```

Definition at line [58](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define ADC_Bemf_CH3_ST  ADC_SAMPLETIME_28CYCLES  
/*BEMF3 sampling time*/
```

Definition at line [66](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADC_CH_1  ADC_CHANNEL_0 /*CURRENT*/
```

Definition at line [52](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define ADC_CH_1_ST  ADC_SAMPLETIME_3CYCLES  
/*CURRENT sampling time */
```

Definition at line [60](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADC_CH_2 ADC_CHANNEL_12 /*SPEED*/
```

Definition at line [53](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define ADC_CH_2_ST ADC_SAMPLETIME_84CYCLES  
/*SPEED sampling time*/
```

Definition at line [61](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADC_CH_3 ADC_CHANNEL_1 /*VBUS*/
```

Definition at line [54](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define ADC_CH_3_ST ADC_SAMPLETIME_84CYCLES /*VBUS  
sampling time*/
```

Definition at line [62](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADC_CH_4 ADC_CHANNEL_2 /*TEMP */
```

Definition at line [55](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by [MC_SixStep_Nucleo_Init\(\)](#), and
[MC_SixStep_RESET\(\)](#).

```
#define ADC_CH_4_ST  ADC_SAMPLETIME_84CYCLES /*TEMP  
sampling time*/
```

Definition at line [63](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#).

```
#define ADCx  hadc1
```

Definition at line [48](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_ADCx_SixStep_Bemf\(\)](#),
[MC_SixStep_ADC_Channel\(\)](#), [MC_StartMotor\(\)](#), and
[MC_StopMotor\(\)](#).

```
#define DAC_ENABLE  0
```

Enable (1) the DAC peripheral

Definition at line [75](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define DACx  htim3
```

Definition at line [49](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define DIRECT_CMD  12
```

Get the motor direction

Definition at line [94](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define GETSPD_CMD  3
```

Get Mechanical Motor Speed command received

Definition at line [85](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define GPIO_CH_COMM GPIO_PIN_10
```

GPIO pin name for 6Step commutation

Definition at line [80](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [HAL_MspInit\(\)](#), [MC_ADCx_SixStep_Bemf\(\)](#), and [MC_SixStep_TABLE\(\)](#).

```
#define GPIO_CH_ZCR GPIO_PIN_12
```

GPIO pin name for zero crossing detection

Definition at line [78](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [HAL_MspInit\(\)](#), and [MC_SixStep_ARR_Bemf\(\)](#).

```
#define GPIO_PORT_COMM GPIOC
```

GPIO port name for 6Step commutation

Definition at line [79](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [HAL_MspInit\(\)](#), [MC_ADCx_SixStep_Bemf\(\)](#), and [MC_SixStep_TABLE\(\)](#).

```
#define GPIO_PORT_ZCR GPIOC
```

GPIO port name for zero crossing detection

Definition at line [77](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by `HAL_MspInit()`, and `MC_SixStep_ARR_Bemf()`.

```
#define HALL_ENCODER_TIMx htim2
```

Definition at line [47](#) of file `stm32F401_nucleo_ihm11m1.h`.

```
#define HELP_CMD 10
```

Help command received

Definition at line [92](#) of file `stm32F401_nucleo_ihm11m1.h`.

```
#define HF_TIMx htim1
```

Definition at line [45](#) of file `stm32F401_nucleo_ihm11m1.h`.

Referenced by `MC_ADCx_SixStep_Bemf()`, `MC_SixStep_INIT()`,
`MC_SixStep_NEXT_step()`, `MC_SixStep_Nucleo_Init()`,
`MC_SixStep_RESET()`, `MC_StopMotor()`,
`STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D()`,
`STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E()`,
`STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E()`,
`STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D()`,
`STSPIN230_HF_TIMx_SetDutyCycle_CH1()`,
`STSPIN230_HF_TIMx_SetDutyCycle_CH2()`,
`STSPIN230_HF_TIMx_SetDutyCycle_CH3()`,
`STSPIN230_Start_PWM_driving()`, and
`STSPIN230_Stop_PWM_driving()`.

```
#define HF_TIMx_CCR1 CCR1 /*Channel 1*/
```

Definition at line [71](#) of file `stm32F401_nucleo_ihm11m1.h`.

```
#define HF_TIMx_CCR2 CCR2 /*Channel 2*/
```

Definition at line [72](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define HF_TIMx_CCR3 CCR3 /*Channel 3*/
```

Definition at line [73](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define HF_TIMx_CH1 TIM_CHANNEL_1
```

Definition at line [68](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#),
[STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#),
[STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#),
[STSPIN230_Start_PWM_driving\(\)](#), and
[STSPIN230_Stop_PWM_driving\(\)](#).

```
#define HF_TIMx_CH2 TIM_CHANNEL_2
```

Definition at line [69](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#),
[STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#),
[STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#),
[STSPIN230_Start_PWM_driving\(\)](#), and
[STSPIN230_Stop_PWM_driving\(\)](#).

```
#define HF_TIMx_CH3 TIM_CHANNEL_3
```

Definition at line [70](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_SixStep_Nucleo_Init\(\)](#),
[STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D\(\)](#),
[STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E\(\)](#),
[STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D\(\)](#),
[STSPIN230_Start_PWM_driving\(\)](#), and
[STSPIN230_Stop_PWM_driving\(\)](#).

```
#define INIREF_CMD 4
```

Set the new STARUP_CURRENT_REFERENCE value command received

Definition at line [86](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define KI_PRM_CMD 8
```

Set the KI PI param command received

Definition at line [90](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define KP_PRM_CMD 7
```

Set the KP PI param command received

Definition at line [89](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define LF_TIMx htim4
```

Definition at line [46](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [MC_GetElSpeedHz\(\)](#), [MC_SixStep_Alignment\(\)](#),
[MC_SixStep_ARR_Bemf\(\)](#), [MC_SixStep_ARR_step\(\)](#),

MC_SixStep_INIT(), **MC_SixStep_NEXT_step()**,
MC_SixStep_RESET(), **MC_StartMotor()**, and **MC_StopMotor()**.

#define POLESP_CMD 5

Set the Pole Pairs value command received

Definition at line [87](#) of file **stm32F401_nucleo_ihm11m1.h**.

#define POTENZ_CMD 9

Enable Potentiometer command received

Definition at line [91](#) of file **stm32F401_nucleo_ihm11m1.h**.

#define SETSPD_CMD 2

Set the new speed value command received

Definition at line [84](#) of file **stm32F401_nucleo_ihm11m1.h**.

#define STARTM_CMD 0

Start Motor command received

Definition at line [82](#) of file **stm32F401_nucleo_ihm11m1.h**.

#define STATUS_CMD 11

Get the Status of the system command received

Definition at line [93](#) of file **stm32F401_nucleo_ihm11m1.h**.

```
#define STOPMT_CMD 1
```

Stop Motor command received

Definition at line [83](#) of file [stm32F401_nucleo_ihm11m1.h](#).

```
#define UART huart2
```

Definition at line [50](#) of file [stm32F401_nucleo_ihm11m1.h](#).

Referenced by [Get_UART_Data\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Macros

stm32f4xx_hal_conf.h

File Reference

HAL configuration file. [More...](#)

```
#include "stm32f4xx_hal_rcc.h" #include "stm32f4xx_hal_gpio.h"  
#include "stm32f4xx_hal_dma.h"  
#include "stm32f4xx_hal_cortex.h"  
#include "stm32f4xx_hal_adc.h"  
#include "stm32f4xx_hal_flash.h"  
#include "stm32f4xx_hal_pwr.h"  
#include "stm32f4xx_hal_spi.h"  
#include "stm32f4xx_hal_tim.h"  
#include "stm32f4xx_hal_uart.h"
```

[Go to the source code of this file.](#)

Macros

```
#define HAL_MODULE_ENABLED
```

This is the list of modules to be used in the HAL driver.
[More...](#)

```
#define HAL_ADC_MODULE_ENABLED
```

```
#define HAL_SPI_MODULE_ENABLED
```

```
#define HAL_TIM_MODULE_ENABLED
```

```
#define HAL_UART_MODULE_ENABLED
```

```
#define HAL_GPIO_MODULE_ENABLED
```

```
#define HAL_DMA_MODULE_ENABLED
```

```
#define HAL_RCC_MODULE_ENABLED
```

```
#define HAL_FLASH_MODULE_ENABLED
```

```
#define HAL_PWR_MODULE_ENABLED
```

```
#define HAL_CORTEX_MODULE_ENABLED
```

```
#define HSE_VALUE ((uint32_t)8000000)
```

Adjust the value of External High Speed oscillator (HSE)
used in your application. [More...](#)

```
#define HSE_STARTUP_TIMEOUT ((uint32_t)5000)
```

```
#define HSI_VALUE ((uint32_t)16000000)
```

Internal High Speed oscillator (HSI) value. [More...](#)

```
#define LSI_VALUE ((uint32_t)32000)
```

Internal Low Speed oscillator (LSI) value. [More...](#)

```
#define LSE_VALUE ((uint32_t)32768)
```

External Low Speed oscillator (LSE) value. [More...](#)

```
#define EXTERNAL_CLOCK_VALUE ((uint32_t)12288000)
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad. [More...](#)

```
#define VDD_VALUE ((uint32_t)3300)
```

This is the HAL system configuration section. [More...](#)

```
#define TICK_INT_PRIORITY ((uint32_t)2)
```

```
#define USE_RTOS 0
```

```
#define PREFETCH_ENABLE 1
```

```
#define INSTRUCTION_CACHE_ENABLE 1
```

```
#define DATA_CACHE_ENABLE 1
```

```
#define MAC_ADDR0 2
```

Uncomment the line below to expand the "assert_param" macro in the HAL drivers code. [More...](#)

```
#define MAC_ADDR1 0
```

```
#define MAC_ADDR2 0
```

```
#define MAC_ADDR3 0
```

```
#define MAC_ADDR4 0
```

```
#define MAC_ADDR5 0
```

```
#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer  
size for receive */
```

```
#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer  
size for transmit */
```

```
#define ETH_RXBUFN ((uint32_t)4) /* 4 Rx buffers of size  
ETH_RX_BUF_SIZE */
```

```
#define ETH_TXBUFN ((uint32_t)4) /* 4 Tx buffers of size  
ETH_TX_BUF_SIZE */
```

```
#define DP83848_PHY_ADDRESS 0x01
```

```
#define PHY_RESET_DELAY ((uint32_t)0x000000FF)
```

```
#define PHY_CONFIG_DELAY ((uint32_t)0x00000FFF)
```

```
#define PHY_READ_TO ((uint32_t)0x0000FFFF)
```

```
#define PHY_WRITE_TO ((uint32_t)0x0000FFFF)
```

```
#define PHY_BCR ((uint16_t)0x00)
```

```
#define PHY_BSR ((uint16_t)0x01)
```

```
#define PHY_RESET ((uint16_t)0x8000)
```

```
#define PHY_LOOPBACK ((uint16_t)0x4000)
```

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100)
```

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000)
```

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100)
```

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000)

#define PHY_AUTONEGOTIATION ((uint16_t)0x1000)

#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200)

#define PHY_POWERDOWN ((uint16_t)0x0800)

#define PHY_ISOLATE ((uint16_t)0x0400)

#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020)

#define PHY_LINKED_STATUS ((uint16_t)0x0004)

#define PHY_JABBER_DETECTION ((uint16_t)0x0002)

#define PHY_SR ((uint16_t)0x10)

#define PHY_MICR ((uint16_t)0x11)

#define PHY_MISR ((uint16_t)0x12)

#define PHY_LINK_STATUS ((uint16_t)0x0001)

#define PHY_SPEED_STATUS ((uint16_t)0x0002)

#define PHY_DUPLEX_STATUS ((uint16_t)0x0004)

#define PHY_MICR_INT_EN ((uint16_t)0x0002)

#define PHY_MICR_INT_OE ((uint16_t)0x0001)

#define PHY_MISR_LINK_INT_EN ((uint16_t)0x0020)

#define PHY_LINK_INTERRUPT ((uint16_t)0x2000)
```

```
#define assert_param(expr) ((void)0)  
Include module's header file. More...
```

Detailed Description

HAL configuration file.

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32f4xx_hal_conf.h](#).

Macro Definition Documentation

#define assert_param (expr) ((void)0)

Include module's header file.

Definition at line [391](#) of file [stm32f4xx_hal_conf.h](#).

#define DATA_CACHE_ENABLE 1

Definition at line [149](#) of file [stm32f4xx_hal_conf.h](#).

#define DP83848_PHY_ADDRESS 0x01

Definition at line [179](#) of file [stm32f4xx_hal_conf.h](#).

#define ETH_RX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for receive */

Definition at line [171](#) of file [stm32f4xx_hal_conf.h](#).

#define ETH_RXBUFN B ((uint32_t)4) /* 4 Rx buffers of size ETH_RX_BUF_SIZE */

Definition at line [173](#) of file [stm32f4xx_hal_conf.h](#).

#define ETH_TX_BUF_SIZE ETH_MAX_PACKET_SIZE /* buffer size for transmit */

Definition at line [172](#) of file [stm32f4xx_hal_conf.h](#).

```
#define ETH_TXBUFN  ((uint32_t)4) /* 4 Tx buffers of size  
ETH_TX_BUFSIZE */
```

Definition at line [174](#) of file [stm32f4xx_hal_conf.h](#).

```
#define EXTERNAL_CLOCK_VALUE ((uint32_t)12288000)
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.

Value of the External audio frequency in Hz

Definition at line [133](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HAL_ADC_MODULE_ENABLED
```

Definition at line [51](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HAL_CORTEX_MODULE_ENABLED
```

Definition at line [87](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HAL_DMA_MODULE_ENABLED
```

Definition at line [83](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HAL_FLASH_MODULE_ENABLED
```

Definition at line [85](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_GPIO_MODULE_ENABLED

Definition at line [82](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_MODULE_ENABLED

This is the list of modules to be used in the HAL driver.

Definition at line [50](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_PWR_MODULE_ENABLED

Definition at line [86](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_RCC_MODULE_ENABLED

Definition at line [84](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_SPI_MODULE_ENABLED

Definition at line [73](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_TIM_MODULE_ENABLED

Definition at line [74](#) of file [stm32f4xx_hal_conf.h](#).

#define HAL_UART_MODULE_ENABLED

Definition at line [75](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HSE_STARTUP_TIMEOUT ((uint32_t)5000)
```

Time out for HSE start up, in ms

Definition at line [100](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HSE_VALUE ((uint32_t)8000000)
```

Adjust the value of External High Speed oscillator (HSE) used in your application.

This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL). Value of the External oscillator in Hz

Definition at line [96](#) of file [stm32f4xx_hal_conf.h](#).

```
#define HSI_VALUE ((uint32_t)16000000)
```

Internal High Speed oscillator (HSI) value.

This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL). Value of the Internal oscillator in Hz

Definition at line [109](#) of file [stm32f4xx_hal_conf.h](#).

```
#define INSTRUCTION_CACHE_ENABLE 1
```

Definition at line [148](#) of file [stm32f4xx_hal_conf.h](#).

```
#define LSE_VALUE ((uint32_t)32768)
```

External Low Speed oscillator (LSE) value.

< Value of the Internal Low Speed oscillator in Hz The real value may vary depending on the variations in voltage and temperature.Value of the External Low Speed oscillator in Hz

Definition at line [124](#) of file [stm32f4xx_hal_conf.h](#).

```
#define LSI_VALUE ((uint32_t)32000)
```

Internal Low Speed oscillator (LSI) value.

Definition at line [116](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR0 2
```

Uncomment the line below to expand the "assert_param" macro in the HAL drivers code.

Definition at line [163](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR1 0
```

Definition at line [164](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR2 0
```

Definition at line [165](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR3 0
```

Definition at line [166](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR4 0
```

Definition at line [167](#) of file [stm32f4xx_hal_conf.h](#).

```
#define MAC_ADDR5 0
```

Definition at line [168](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020)
```

Auto-Negotiation process completed

Definition at line [204](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000)
```

Enable auto-negotiation function

Definition at line [199](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_BCR ((uint16_t)0x00)
```

Transceiver Basic Control Register

Definition at line [190](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_BSR ((uint16_t)0x01)
```

Transceiver Basic Status Register

Definition at line [191](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_CONFIG_DELAY ((uint32_t)0x00000FFF)
```

Definition at line [183](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_DUPLEX_STATUS ((uint16_t)0x0004)
```

PHY Duplex mask

Definition at line [216](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100)
```

Set the full-duplex mode at 100 Mb/s

Definition at line [195](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100)
```

Set the full-duplex mode at 10 Mb/s

Definition at line [197](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000)
```

Set the half-duplex mode at 100 Mb/s

Definition at line [196](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000)
```

Set the half-duplex mode at 10 Mb/s

Definition at line [198](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_ISOLATE ((uint16_t)0x0400)
```

Isolate PHY from MII

Definition at line [202](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002)
```

Jabber condition detected

Definition at line [206](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_LINK_INTERRUPT ((uint16_t)0x2000)
```

PHY link status interrupt mask

Definition at line [222](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_LINK_STATUS ((uint16_t)0x0001)
```

PHY Link mask

Definition at line [214](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004)
```

Valid link established

Definition at line [205](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_LOOPBACK ((uint16_t)0x4000)
```

Select loop-back mode

Definition at line [194](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_MICR ((uint16_t)0x11)
```

MII Interrupt Control Register

Definition at line [211](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_MICR_INT_EN ((uint16_t)0x0002)
```

PHY Enable interrupts

Definition at line [218](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_MICR_INT_OE ((uint16_t)0x0001)
```

PHY Enable output interrupt events

Definition at line [219](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_MISR ((uint16_t)0x12)
```

MII Interrupt Status and Misc. Control Register

Definition at line [212](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_MISR_LINK_INT_EN ((uint16_t)0x0020)
```

Enable Interrupt on change of link status

Definition at line [221](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_POWERDOWN ((uint16_t)0x0800)
```

Select the power down mode

Definition at line [201](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_READ_TO ((uint32_t)0x0000FFFF)
```

Definition at line [185](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_RESET ((uint16_t)0x8000)
```

PHY Reset

Definition at line [193](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_RESET_DELAY ((uint32_t)0x000000FF)
```

Definition at line [181](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200)
```

Restart auto-negotiation function

Definition at line [200](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_SPEED_STATUS ((uint16_t)0x0002)
```

PHY Speed mask

Definition at line [215](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_SR ((uint16_t)0x10)
```

PHY status register Offset

Definition at line [210](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PHY_WRITE_TO ((uint32_t)0x0000FFFF)
```

Definition at line [186](#) of file [stm32f4xx_hal_conf.h](#).

```
#define PREFETCH_ENABLE 1
```

Definition at line [147](#) of file [stm32f4xx_hal_conf.h](#).

```
#define TICK_INT_PRIORITY ((uint32_t)2)
```

tick interrupt priority

Definition at line [145](#) of file [stm32f4xx_hal_conf.h](#).

```
#define USERTOS 0
```

Definition at line [146](#) of file [stm32f4xx_hal_conf.h](#).

```
#define VDD_VALUE ((uint32_t)3300)
```

This is the HAL system configuration section.

Value of VDD in mv

Definition at line [144](#) of file [stm32f4xx_hal_conf.h](#).

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Functions

stm32f4xx_hal_msp.c

File Reference

This file provides code for the MSP Initialization and de-Initialization codes. [More...](#)

```
#include "stm32f4xx_hal.h" #include "stm32f401_nucleo_ihm11m1.h"
```

[Go to the source code of this file.](#)

Functions

void **HAL_MspInit** (void)

Initializes the Global MSP. [More...](#)

void **HAL_ADC_MspInit** (ADC_HandleTypeDef *hadc)

void **HAL_ADC_MspDeInit** (ADC_HandleTypeDef *hadc)

void **HAL_TIM_Base_MspInit** (TIM_HandleTypeDef *htim_base)

void **HAL_TIM_PWM_MspInit** (TIM_HandleTypeDef *htim)

Initializes the TIM PWM MSP. [More...](#)

void **HAL_TIMEx_HallSensor_MspInit** (TIM_HandleTypeDef *htimex_hallsensor)

void **HAL_TIM_Base_MspDeInit** (TIM_HandleTypeDef *htim_base)

void **HAL_TIMEx_HallSensor_MspDeInit** (TIM_HandleTypeDef *htimex_hallsensor)

void **HAL_UART_MspInit** (UART_HandleTypeDef *huart)

void **HAL_UART_MspDeInit** (UART_HandleTypeDef *huart)

Detailed Description

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

Author

IPC

Version

V0

Date

10/07/2016 COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32f4xx_hal_msp.c](#).

Function Documentation

void HAL_ADC_MspDelInit (ADC_HandleTypeDef * hadc)

ADC1 GPIO Configuration IHM11M1 PC1 ----> ADC1_IN11 PC2 ---
---> ADC1_IN12 PC3 ----> ADC1_IN13 PA1 ----> ADC1_IN1 PA7
----> ADC1_IN7 PB0 ----> ADC1_IN8

Definition at line [160](#) of file [stm32f4xx_hal_msp.c](#).

void HAL_ADC_MspInit (ADC_HandleTypeDef * hadc)

ADC1 GPIO Configuration IHM11M1 PC1 ----> ADC1_IN11 PC2 ---
---> ADC1_IN12 PC3 ----> ADC1_IN13 PA1 ----> ADC1_IN1 PA7
----> ADC1_IN7 PB0 ----> ADC1_IN8

Definition at line [115](#) of file [stm32f4xx_hal_msp.c](#).

void HAL_MspInit (void)

Initializes the Global MSP.

GPIO Configuration IHM11M1

Definition at line [48](#) of file [stm32f4xx_hal_msp.c](#).

References [GPIO_CH_COMM](#), [GPIO_CH_ZCR](#),
[GPIO_PORT_COMM](#), and [GPIO_PORT_ZCR](#).

**void
HAL_TIM_Base_MspDelInit (TIM_HandleTypeDef * htim_base)**

TIM1 GPIO Configuration IHM11M1 PA8 —> TIM1_CH1 PA9 —> TIM1_CH2 PA10 —> TIM1_CH3 PB13 —> TIM1_CH1N PB14 —> TIM1_CH2N PB1 —> TIM1_CH3N

TIM3 GPIO Configuration PB4 —> TIM3_CH1

Definition at line [374](#) of file [stm32f4xx_hal_msp.c](#).

void HAL_TIM_Base_MspInit (TIM_HandleTypeDef * htim_base)

TIM1 GPIO Configuration IHM11M1 PA8 —> TIM1_CH1 PA9 —> TIM1_CH2 PA10 —> TIM1_CH3 PB13 —> TIM1_CH1N PB14 —> TIM1_CH2N PB1 —> TIM1_CH3N

TIM3 GPIO Configuration IHM11M1 PB4 —> TIM3_CH1

Definition at line [195](#) of file [stm32f4xx_hal_msp.c](#).

void HAL_TIM_PWM_MspInit (TIM_HandleTypeDef * htim)

Initializes the TIM PWM MSP.

Parameters

htim pointer to a `TIM_HandleTypeDef` structure that contains the configuration information for TIM module.

Return values

None

TIM1 GPIO Configuration IHM11M1 PA8 —> TIM1_CH1 PA9 —> TIM1_CH2 PA10 —> TIM1_CH3 PB13 —> TIM1_CH1N PB14 —> TIM1_CH2N PB1 —> TIM1_CH3N

Definition at line [287](#) of file [stm32f4xx_hal_msp.c](#).

```
void HAL_TIMEx_HallSensor_MspDeInit ( TIM_HandleTypeDef * htimex )
```

TIM2 GPIO Configuration PA15 ----> TIM2_CH1 PB10 ----> TIM2_C
> TIM2_CH2

Definition at line [440](#) of file [stm32f4xx_hal_msp.c](#).

```
void HAL_TIMEx_HallSensor_MspInit ( TIM_HandleTypeDef * htimex )
```

TIM2 GPIO Configuration IHM11M1 PA15 ----> TIM2_CH1 PB10 ----
TIM2_CH3 PB3 ----> TIM2_CH2

Definition at line [326](#) of file [stm32f4xx_hal_msp.c](#).

```
void HAL_UART_MspDeInit ( UART_HandleTypeDef * huart )
```

USART2 GPIO Configuration PA2 ----> USART2_TX PA3 ---->
USART2_RX

Definition at line [500](#) of file [stm32f4xx_hal_msp.c](#).

```
void HAL_UART_MspInit ( UART_HandleTypeDef * huart )
```

USART2 GPIO Configuration PA2 ----> USART2_TX PA3 ---->
USART2_RX

Definition at line [467](#) of file [stm32f4xx_hal_msp.c](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Src | Functions | Variables

stm32f4xx_it.c File Reference

Interrupt Service Routines. More...

```
#include "stm32f4xx_hal.h" #include "stm32f4xx.h"  
#include "stm32f4xx_it.h"  
#include "6Step_Lib.h"
```

Go to the source code of this file.

Functions

void **UART_Set_Value** (void)

void **ADC_IRQHandler** (void)

This function handles ADC1 global interrupt. [More...](#)

void **SysTick_Handler** (void)

This function handles System tick timer. [More...](#)

void **USART2_IRQHandler** (void)

This function handles USART2 global interrupt. [More...](#)

void **TIM4_IRQHandler** (void)

This function handles TIM4 global interrupt. [More...](#)

void **EXTI15_10_IRQHandler** (void)

This function handles EXTI Line[15:10] interrupts. [More...](#)

void **TIM1_BRK_TIM9_IRQHandler** (void)

This function handles TIM1 Break interrupt and TIM9 global interrupt. [More...](#)

Variables

SIXSTEP_Base_InitTypeDef **SIXSTEP_parameters**

ADC_HandleTypeDef **hadc1**

TIM_HandleTypeDef **htim1**

TIM_HandleTypeDef **htim4**

UART_HandleTypeDef **huart2**

Detailed Description

Interrupt Service Routines.

Date

10/07/2016 COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32f4xx_it.c](#).

Function Documentation

void ADC_IRQHandler (void)

This function handles ADC1 global interrupt.

Definition at line [59](#) of file [stm32f4xx_it.c](#).

References [hadc1](#).

void EXTI15_10_IRQHandler (void)

This function handles EXTI Line[15:10] interrupts.

Definition at line [118](#) of file [stm32f4xx_it.c](#).

void SysTick_Handler (void)

This function handles System tick timer.

Definition at line [73](#) of file [stm32f4xx_it.c](#).

References [HAL_IncTick\(\)](#).

void TIM1_BRK_TIM9_IRQHandler (void)

This function handles TIM1 Break interrupt and TIM9 global interrupt.

Definition at line [132](#) of file [stm32f4xx_it.c](#).

References [htim1](#), [MC_StopMotor\(\)](#), [OVERCURRENT](#), and

[**SIXSTEP_Base_InitTypeDef::STATUS**](#).

void TIM4_IRQHandler (void)

This function handles TIM4 global interrupt.

Definition at line [104](#) of file [**stm32f4xx_it.c**](#).

References [**htim4**](#).

void UART_Set_Value (void)

Referenced by [**USART2_IRQHandler\(\)**](#).

void USART2_IRQHandler (void)

This function handles USART2 global interrupt.

Definition at line [88](#) of file [**stm32f4xx_it.c**](#).

References [**huart2**](#), and [**UART_Set_Value\(\)**](#).

Variable Documentation

ADC_HandleTypeDef hadc1

Definition at line [46](#) of file [main_F401.c](#).

Referenced by [ADC_IRQHandler\(\)](#).

TIM_HandleTypeDef htim1

Definition at line [48](#) of file [main_F401.c](#).

Referenced by [TIM1_BRK_TIM9_IRQHandler\(\)](#).

TIM_HandleTypeDef htim4

Definition at line [51](#) of file [main_F401.c](#).

Referenced by [TIM4_IRQHandler\(\)](#).

UART_HandleTypeDef huart2

Definition at line [53](#) of file [main_F401.c](#).

Referenced by [USART2_IRQHandler\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

Functions

stm32f4xx_it.h File Reference

This file contains the headers of the interrupt handlers. [More...](#)

[Go to the source code of this file.](#)

Functions

void **ADC_IRQHandler** (void)

This function handles ADC1 global interrupt. [More...](#)

void **SysTick_Handler** (void)

This function handles System tick timer. [More...](#)

void **USART2_IRQHandler** (void)

This function handles USART2 global interrupt. [More...](#)

void **TIM4_IRQHandler** (void)

This function handles TIM4 global interrupt. [More...](#)

void **EXTI15_10_IRQHandler** (void)

This function handles EXTI Line[15:10] interrupts. [More...](#)

void **TIM1_BRK_TIM9_IRQHandler** (void)

This function handles TIM1 Break interrupt and TIM9 global interrupt. [More...](#)

Detailed Description

This file contains the headers of the interrupt handlers.

Date

27/04/2015 12:50:22 COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [stm32f4xx_it.h](#).

Function Documentation

void ADC_IRQHandler (void)

This function handles ADC1 global interrupt.

Definition at line [59](#) of file [stm32f4xx_it.c](#).

References [hadc1](#).

void EXTI15_10_IRQHandler (void)

This function handles EXTI Line[15:10] interrupts.

Definition at line [118](#) of file [stm32f4xx_it.c](#).

void SysTick_Handler (void)

This function handles System tick timer.

Definition at line [73](#) of file [stm32f4xx_it.c](#).

References [HAL_IncTick\(\)](#).

void TIM1_BRK_TIM9_IRQHandler (void)

This function handles TIM1 Break interrupt and TIM9 global interrupt.

Definition at line [132](#) of file [stm32f4xx_it.c](#).

References [htim1](#), [MC_StopMotor\(\)](#), [OVERCURRENT](#), and

SIXSTEP_Base_InitTypeDef::STATUS.

void TIM4_IRQHandler (void)

This function handles TIM4 global interrupt.

Definition at line **104** of file [stm32f4xx_it.c](#).

References [htim4](#).

void USART2_IRQHandler (void)

This function handles USART2 global interrupt.

Definition at line **88** of file [stm32f4xx_it.c](#).

References [huart2](#), and [UART_Set_Value\(\)](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	STSPIN230

Functions | Variables

STSPIN230.c File Reference

This file provides a set of functions to manage STSPIN230 driver.
[More...](#)

```
#include "STSPIN230.h"
```

[Go to the source code of this file.](#)

Functions

```
void EnableInput_CH1_E_CH2_E_CH3_D ()
```

```
void EnableInput_CH1_E_CH2_D_CH3_E ()
```

```
void EnableInput_CH1_D_CH2_E_CH3_E ()
```

```
void DisableInput_CH1_D_CH2_D_CH3_D ()
```

```
void Start_PWM_driving ()
```

```
void Stop_PWM_driving ()
```

```
void HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)
```

```
void HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)
```

```
void HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)
```

```
void Current_Reference_Start ()
```

```
void Current_Reference_Stop ()
```

```
void Current_Reference_Setvalue (uint16_t Iref)
```

Variables

STSPIN230_MotorDriver_TypeDef STSPIN230MotorDriver
It handles all API functions for
STSPIN230 MC Driver. [More...](#)

Detailed Description

This file provides a set of functions to manage STSPIN230 driver.

Author

IPC

Version

V0

Date

28-April-2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [STSPIN230.c](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	STSPIN230

Functions

STSPIN230.h File Reference

This file provides a set of functions to manage STSPIN230 driver.
[More...](#)

```
#include "stdint.h" #include "MC_Common.h"
```

[Go to the source code of this file.](#)

Functions

void **STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D** (void)
Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E** (void)
Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E** (void)
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D** (void)
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_Start_PWM_driving** (void)
Enable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_Stop_PWM_driving** (void)
Disable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH1** (uint16_t)
Set the Duty Cycle value for CH1. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH2** (uint16_t)
Set the Duty Cycle value for CH2. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH3** (uint16_t)
Set the Duty Cycle value for CH3. [More...](#)

void **STSPIN230_Current_Reference_Start** (void)
Enable the Current Reference generation. [More...](#)

void **STSPIN230_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **STSPIN230_Current_Reference_Setvalue** (uint16_t)
Set the value for Current Reference. More...

void **STSPIN230_START_Ref_Generation** (void)

void **STSPIN230_STOP_Ref_Generation** (void)

void **STSPIN230_Set_Ref_Generation** (uint16_t)

void **EnableInput_CH1_E_CH2_E_CH3_D** (void)

void **EnableInput_CH1_E_CH2_D_CH3_E** (void)

void **EnableInput_CH1_D_CH2_E_CH3_E** (void)

void **DisableInput_CH1_D_CH2_D_CH3_D** (void)

void **Start_PWM_driving** (void)

void **Stop_PWM_driving** (void)

void **HF_TIMx_SetDutyCycle_CH1** (uint16_t)

void **HF_TIMx_SetDutyCycle_CH2** (uint16_t)

void **HF_TIMx_SetDutyCycle_CH3** (uint16_t)

void **Current_Reference_Start** (void)

void **Current_Reference_Stop** (void)

void **Current_Reference_Setvalue** (uint16_t)

Detailed Description

This file provides a set of functions to manage STSPIN230 driver.

Author

IPC

Version

V0

Date

04-April-2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [STSPIN230.h](#).

Function Documentation

void STSPIN230_Set_Ref_Generation (uint16_t)

void STSPIN230_START_Ref_Generation (void)

void STSPIN230_STOP_Ref_Generation (void)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	UART_serial_com	Src

UART_UI.c File Reference

This file provides a set of functions needed to manage the UART com.
[More...](#)

```
#include "UART_UI.h"
```

[Go to the source code of this file.](#)

Detailed Description

This file provides a set of functions needed to manage the UART com.

Author

IPC

Version

V0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [UART_UI.c](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		

Middlewares › ST › UART_serial_com › Inc ›

Data Structures | Macros | Functions

UART_UI.h File Reference

This file provides a set of functions needed to manage the UART com.
[More...](#)

```
#include "6Step_Lib.h" #include "stdlib.h"  
#include "stdio.h"  
#include <string.h>
```

[Go to the source code of this file.](#)

Data Structures

```
struct CMD_T
```

Macros

```
#define TOKEN "\r"
```

```
#define CMD_NUM 16
```

```
#define COUNTOF(__BUFFER__) (sizeof(__BUFFER__) / sizeof(*  
    __BUFFER__)))
```

```
#define TXBUFFERSIZE (COUNTOF(aTxBuffer) - 1)
```

```
#define RXBUFFERSIZE 8
```

Functions

void **CMD_STARTM** (void)
UART function. More...

void **CMD_STOPMT** (void)

void **CMD_DIRECTION** (void)

void **CMD_SETSPD** (void)

void **CMD_GETSPD** (void)

void **CMD_STATUS** (void)

void **CMD_POTENZ** (void)

void **CMD_HELP** (void)

void **CMD_INIREF** (void)

void **CMD_POLESP** (void)

void **CMD_ACCELE** (void)

void **CMD_KP_PRM** (void)

void **CMD_KI_PRM** (void)

Detailed Description

This file provides a set of functions needed to manage the UART com.

Author

System lab

Version

V1.0.0

Date

06-July-2015

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [**UART_UI.h**](#).

Macro Definition Documentation

#define CMD_NUM 16

Definition at line [44](#) of file [UART_UI.h](#).

#define COUNTOF (__BUFFER__) (__BUFFER__) (__BUFFER__) (sizeof(__BUFFER__) / sizeof(* __BUFFER__))

Definition at line [46](#) of file [UART_UI.h](#).

#define RXBUFFERSIZE 8

Definition at line [48](#) of file [UART_UI.h](#).

#define TOKEN "\r"

Definition at line [43](#) of file [UART_UI.h](#).

#define TXBUFFERSIZE (COUNTOF(aTxBuffer) - 1)

Definition at line [47](#) of file [UART_UI.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	X-NUCLEO-IHM11M1	

Functions | Variables

X-NUCLEO-IHM11M1.c File Reference

This file provides the set of functions to manage the X-Nucleo expansion board. [More...](#)

```
#include "X-NUCLEO-IHM11M1.h" #include "6Step_Lib.h"
```

[Go to the source code of this file.](#)

Functions

void **STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D ()**
Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E ()**
Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E ()**
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D ()**
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_Start_PWM_driving ()**
Enable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_Stop_PWM_driving ()**
Disable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH1 (uint16_t CCR_value)**
Set the Duty Cycle value for CH1. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH2 (uint16_t CCR_value)**
Set the Duty Cycle value for CH2. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH3 (uint16_t CCR_value)**
Set the Duty Cycle value for CH3. [More...](#)

void **STSPIN230_Current_Reference_Start ()**
Enable the Current Reference generation. [More...](#)

```
void STSPIN230_Current_Reference_Stop ()  
    Disable the Current Reference generation. More...
```

```
void STSPIN230_Current_Reference_Setvalue (uint16_t lref)  
    Set the value for Current Reference. More...
```

```
void BSP_X_NUCLEOFAULT_LED_ON ()
```

```
void BSP_X_NUCLEOFAULT_LED_OFF ()
```

Variables

SIXSTEP_Base_InitTypeDef SIXSTEP_parameters

Detailed Description

This file provides the set of functions to manage the X-Nucleo expansion board.

Author

IPC

Version

V0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [X-NUCLEO-IHM11M1.c](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	X-NUCLEO-IHM11M1	Functions

X-NUCLEO-IHM11M1.h File Reference

This file provides the set of functions to manage the X-Nucleo board.
[More...](#)

```
#include "STSPIN230.h"
```

[Go to the source code of this file.](#)

Functions

void **STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D** (void)
Enable Input channel CH1 and CH2 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E** (void)
Enable Input channel CH1 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E** (void)
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D** (void)
Enable Input channel CH2 and CH3 for STSPIN230. [More...](#)

void **STSPIN230_Start_PWM_driving** (void)
Enable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_Stop_PWM_driving** (void)
Disable PWM channels for STSPIN230. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH1** (uint16_t)
Set the Duty Cycle value for CH1. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH2** (uint16_t)
Set the Duty Cycle value for CH2. [More...](#)

void **STSPIN230_HF_TIMx_SetDutyCycle_CH3** (uint16_t)
Set the Duty Cycle value for CH3. [More...](#)

void **STSPIN230_Current_Reference_Start** (void)
Enable the Current Reference generation. [More...](#)

void **STSPIN230_Current_Reference_Stop** (void)
Disable the Current Reference generation. [More...](#)

void **STSPIN230_Current_Reference_Setvalue** (uint16_t)

Set the value for Current Reference. More...

Detailed Description

This file provides the set of functions to manage the X-Nucleo board.

Author

IPC

Version

V0

Date

10/07/2016

Attention

© COPYRIGHT(c) 2015 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [X-NUCLEO-IHM11M1.h](#).



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- a -

- ACC : [MC_SixStep_param.h](#)
- ACCELE_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH1 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH1_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH2 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH2_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH3 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH3_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_1 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_1_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_2 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_2_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_3 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_3_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_4 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_4_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_IRQHandler() : [stm32f4xx_it.c , stm32f4xx_it.h](#)
- ADC_SPEED_TH : [MC_SixStep_param.h](#)
- ADCx : [stm32F401_nucleo_ihm11m1.h](#)
- ALIGNMENT : [6Step_Lib.h](#)

- ARR_LF : **6Step.Lib.c**
 - array_completed : **6Step.Lib.c**
 - assert_param : **stm32f4xx_hal_conf.h**
-

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- b -

- BEMF_CNT_EVENT_MAX : [MC_SixStep_param.h](#)
- BEMF_CONSEC_DOWN_MAX : [MC_SixStep_param.h](#)
- Bemf_delay_calc() : [stm32F401_nucleo_ihm11m1.h](#) , [stm32F401_nucleo_ihm11m1.c](#)
- BEMF_THRSLD_DOWN : [MC_SixStep_param.h](#)
- BEMF_THRSLD_UP : [MC_SixStep_param.h](#)
- BSP_X_NUCLEOFAULT_LED_OFF() :
[stm32F401_nucleo_ihm11m1.h](#) , [X-NUCLEO-IHM11M1.c](#)
- BSP_X_NUCLEOFAULT_LED_ON() :
[stm32F401_nucleo_ihm11m1.h](#) , [X-NUCLEO-IHM11M1.c](#)
- buffer_completed : [6Step_Lib.c](#)
- BUTTON_DELAY : [MC_SixStep_param.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- C -

- CMD_ACCELE() : [UART_UI.h](#)
- CMD_DIRECTION() : [UART_UI.h](#)
- CMD_GETSPD() : [UART_UI.h](#)
- CMD_HELP() : [UART_UI.h](#)
- CMD_INIREF() : [UART_UI.h](#)
- CMD_KI_PRM() : [UART_UI.h](#)
- CMD_KP_PRM() : [UART_UI.h](#)
- CMD_NUM : [UART_UI.h](#)
- CMD_Parser() : [6Step_Lib.c](#)
- CMD_POLESP() : [UART_UI.h](#)
- CMD_POTENZ() : [UART_UI.h](#)
- CMD_SETSPD() : [UART_UI.h](#)
- CMD_STARTM() : [UART_UI.h](#)
- CMD_STATUS() : [UART_UI.h](#)
- CMD_STOPMT() : [UART_UI.h](#)
- cnt_bemf_event : [6Step_Lib.c](#)
- constant_k : [6Step_Lib.c](#)
- constant_multiplier_tmp : [6Step_Lib.c](#)
- counter_ARR_Bemf : [6Step_Lib.c](#)
- COUNTOF : [UART_UI.h](#)

- Current_Reference_Setvalue() : **STSPIN230.h** , **STSPIN230.c**
 - Current_Reference_Start() : **STSPIN230.c** , **STSPIN230.h**
 - Current_Reference_Stop() : **STSPIN230.c** , **STSPIN230.h**
-

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- d -

- DAC_ENABLE : [stm32F401_nucleo_ihm11m1.h](#)
- dac_status : [6Step_Lib.c](#)
- DACx : [stm32F401_nucleo_ihm11m1.h](#)
- DATA_CACHE_ENABLE : [stm32f4xx_hal_conf.h](#)
- delta : [6Step_Lib.c](#)
- DEMAGN_VAL_1 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_10 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_11 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_12 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_13 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_14 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_2 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_3 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_4 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_5 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_6 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_7 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_8 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_9 : [MC_SixStep_param.h](#)
- DEMO_START_TIME : [MC_SixStep_param.h](#)

- DEMO_STOP_TIME : **MC_SixStep_param.h**
 - DIRECT_CMD : **stm32F401_nucleo_ihm11m1.h**
 - DIRECTION : **MC_SixStep_param.h**
 - DisableInput_CH1_D_CH2_D_CH3_D() : **STSPIN230.c** ,
STSPIN230.h
 - DP83848_PHY_ADDRESS : **stm32f4xx_hal_conf.h**
-

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- e -

- El_Speed_Hz : [6Step_Lib.c](#)
- Enable_start_button : [6Step_Lib.c](#)
- EnableInput_CH1_D_CH2_E_CH3_E() : [STSPIN230.h](#) , [STSPIN230.c](#)
- EnableInput_CH1_E_CH2_D_CH3_E() : [STSPIN230.c](#) , [STSPIN230.h](#)
- EnableInput_CH1_E_CH2_E_CH3_D() : [STSPIN230.h](#) , [STSPIN230.c](#)
- ETH_RX_BUF_SIZE : [stm32f4xx_hal_conf.h](#)
- ETH_RXBUFN: [stm32f4xx_hal_conf.h](#)
- ETH_TX_BUF_SIZE : [stm32f4xx_hal_conf.h](#)
- ETH_TXBUFN: [stm32f4xx_hal_conf.h](#)
- EXTERNAL_CLOCK_VALUE : [stm32f4xx_hal_conf.h](#)
- EXTI15_10_IRQHandler() : [stm32f4xx_it.h](#) , [stm32f4xx_it.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- f -

- FALSE : [MC_SixStep_param.h](#)
- FILTER_DEEP : [MC_SixStep_param.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- g -

- Get_UART_Data() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- GETSPD_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_CH_COMM : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_CH_ZCR : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_COMM : [MC_SixStep_param.h](#)
- GPIO_PORT_COMM : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_PORT_ZCR : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_ZERO_CROSS : [MC_SixStep_param.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- h -

- hadc1 : [main_F401.c](#) , [stm32f4xx_it.c](#) , [main_F401.h](#)
- HAL_ADC_ConvCpltCallback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_ADC_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_ADC_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_ADC_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_CORTEX_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_DMA_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_FLASH_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_GetTick() : [6Step_Lib.c](#)
- HAL_GPIO_EXTI_Callback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_GPIO_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_IncTick() : [6Step_Lib.c](#)
- HAL_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_PWR_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_RCC_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_SPI_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_SYSTICK_Callback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_TIM_Base_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIM_Base_MspInit() : [stm32f4xx_hal_msp.c](#)

- HAL_TIM_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_TIM_PeriodElapsedCallback() :
[stm32F401_nucleo_ihm11m1.c](#)
- HAL_TIM_PWM_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIMEx_HallSensor_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIMEx_HallSensor_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_UART_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_UART_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_UART_MspInit() : [stm32f4xx_hal_msp.c](#)
- HALL_ENCODER_TIMx : [stm32F401_nucleo_ihm11m1.h](#)
- HELP_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR1 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR2 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR3 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CH1 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CH2 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CH3 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_SetDutyCycle_CH1() : [STSPIN230.c](#) , [STSPIN230.h](#)
- HF_TIMx_SetDutyCycle_CH2() : [STSPIN230.h](#) , [STSPIN230.c](#)
- HF_TIMx_SetDutyCycle_CH3() : [STSPIN230.h](#) , [STSPIN230.c](#)
- HFBuffer : [6Step_Lib.c](#)
- HFBufferIndex : [6Step_Lib.c](#)
- HFBUFFERSIZE : [MC_SixStep_param.h](#)
- HSE_STARTUP_TIMEOUT : [stm32f4xx_hal_conf.h](#)
- HSE_VALUE : [stm32f4xx_hal_conf.h](#)
- HSI_VALUE : [stm32f4xx_hal_conf.h](#)
- htim1 : [main_F401.h](#) , [main_F401.c](#) , [stm32f4xx_it.c](#)
- htim2 : [main_F401.c](#)
- htim3 : [main_F401.c](#) , [main_F401.h](#)
- htim4 : [main_F401.h](#) , [main_F401.c](#) , [stm32f4xx_it.c](#)
- huart2 : [main_F401.c](#) , [stm32f4xx_it.c](#) , [main_F401.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- i -

- IDLE : [6Step_Lib.h](#)
- index_adc_chn : [6Step_Lib.c](#)
- index_align : [6Step_Lib.c](#)
- index.ARR_step : [6Step_Lib.c](#)
- index_array : [6Step_Lib.c](#)
- index_motor_run : [6Step_Lib.c](#)
- index_pot_filt : [6Step_Lib.c](#)
- index_startup_motor : [6Step_Lib.c](#)
- INIREF_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- INITIAL_DEMAGN_DELAY : [MC_SixStep_param.h](#)
- INSTRUCTION_CACHE_ENABLE : [stm32f4xx_hal_conf.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- k -

- KI_DIV : [MC_SixStep_param.h](#)
- KI_GAIN : [MC_SixStep_param.h](#)
- KI_PRM_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- KP_DIV : [MC_SixStep_param.h](#)
- KP_GAIN : [MC_SixStep_param.h](#)
- KP_PRM_CMD : [stm32F401_nucleo_ihm11m1.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	I	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- I -

- LF_TIMx : [stm32f401_nucleo_ihm11m1.h](#)
- LOWER_OUT_LIMIT : [MC_SixStep_param.h](#)
- LSE_VALUE : [stm32f4xx_hal_conf.h](#)
- LSI_VALUE : [stm32f4xx_hal_conf.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																		
File List	Globals																				
All	Functions	Variables	Typedefs	Enumerations	Enumerator																
Macros																					
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- m -

- MAC_ADDR0 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR1 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR2 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR3 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR4 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR5 : [stm32f4xx_hal_conf.h](#)
- main() : [main_F401.c](#)
- MAX_POT_SPEED : [MC_SixStep_param.h](#)
- MC_ADCx_SixStep_Bemf() : [6Step_Lib.c](#) , [stm32F401_nucleo_ihm11m1.c](#)
- MC_Bemf_Delay() : [6Step_Lib.c](#)
- MC_EXT_button_SixStep() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
- MC_GetElSpeedHz() : [6Step_Lib.c](#)
- MC_GetMechSpeedRPM() : [6Step_Lib.c](#)
- MC_PI_Controller() : [6Step_Lib.c](#)
- MC_Potentiometer_filter() : [6Step_Lib.c](#)
- MC_Set_PI_param() : [6Step_Lib.c](#)
- MC_Set_Speed() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
- MC_SixStep_ADC_Channel() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)

- MC_SixStep_Alignment() : **6Step_Lib.c**
- MC_SixStep_ARR_Bemf() : **6Step_Lib.c**
- MC_SixStep_ARR_step() : **6Step_Lib.c**
- MC_SixStep_Current_Reference_Setvalue() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_Current_Reference_Start() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_Current_Reference_Stop() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH1() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH2() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH3() :
stm32F401_nucleo_ihm11m1.h , **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c
- MC_SixStep_INIT() : **6Step_Lib.h** , **6Step_Lib.c**
- MC_SixStep_Init_main_data() : **6Step_Lib.c**
- MC_SixStep_NEXT_step() : **6Step_Lib.c**
- MC_SixStep_Nucleo_Init() : **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_Ramp_Motor_calc() : **6Step_Lib.c**
- MC_SixStep_RESET() : **6Step_Lib.h** , **6Step_Lib.c**

- MC_SixStep_Speed_Potentiometer() : [6Step.Lib.c](#)
- MC_SixStep_Speed_Val_target_potentiometer() : [6Step.Lib.c](#)
- MC_SixStep_Start_PWM_driving() :
[stm32F401_nucleo_ihm11m1.c](#) , [6Step.Lib.c](#) ,
[stm32F401_nucleo_ihm11m1.h](#)
- MC_SixStep_Stop_PWM_driving() :
[stm32F401_nucleo_ihm11m1.h](#) ,
[stm32F401_nucleo_ihm11m1.c](#) , [6Step.Lib.c](#)
- MC_SixStep_TABLE() : [6Step.Lib.c](#)
- MC_Speed_Filter() : [6Step.Lib.c](#)
- MC_StartMotor() : [6Step.Lib.c](#) , [6Step.Lib.h](#)
- MC_StopMotor() : [6Step.Lib.h](#) , [6Step.Lib.c](#)
- MC_SysTick_SixStep_MediumFrequencyTask() :
[stm32F401_nucleo_ihm11m1.c](#) , [6Step.Lib.c](#)
- MC_Task_Speed() : [6Step.Lib.c](#)
- MC_TIMx_SixStep_timebase() : [6Step.Lib.c](#) ,
[stm32F401_nucleo_ihm11m1.c](#)
- MC_UI_INIT() : [6Step.Lib.c](#)
- MCM_Sqrt() : [6Step.Lib.c](#)
- mech_accel_hz : [6Step.Lib.c](#)
- Mech_Speed_RPM : [6Step.Lib.c](#)
- MIN_POT_SPEED : [MC_SixStep_param.h](#)
- MINIMUM_ACC : [MC_SixStep_param.h](#)
- MX_ADC1_Init() : [main_F401.c](#)
- MX_TIM1_Init() : [main_F401.c](#)
- MX_TIM3_Init() : [main_F401.c](#)
- MX_TIM4_Init() : [main_F401.c](#)
- MX_USART2_UART_Init() : [main_F401.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- n -

- n_zcr_startup : [6Step.Lib.c](#)
- NUM_POLE_PAIRS : [MC_SixStep_param.h](#)
- NUMBER_OF_STEPS : [MC_SixStep_param.h](#)
- NUMBER_ZCR : [MC_SixStep_param.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- O -

- OVERCURRENT : [6Step_Lib.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- p -

- PHY_AUTONEGO_COMPLETE : [stm32f4xx_hal_conf.h](#)
- PHY_AUTONEGOTIATION : [stm32f4xx_hal_conf.h](#)
- PHY_BCR : [stm32f4xx_hal_conf.h](#)
- PHY_BSR : [stm32f4xx_hal_conf.h](#)
- PHY_CONFIG_DELAY : [stm32f4xx_hal_conf.h](#)
- PHY_DUPLEX_STATUS : [stm32f4xx_hal_conf.h](#)
- PHY_FULLDUPLEX_100M : [stm32f4xx_hal_conf.h](#)
- PHY_FULLDUPLEX_10M : [stm32f4xx_hal_conf.h](#)
- PHY_HALFDUPLEX_100M : [stm32f4xx_hal_conf.h](#)
- PHY_HALFDUPLEX_10M : [stm32f4xx_hal_conf.h](#)
- PHY_ISOLATE : [stm32f4xx_hal_conf.h](#)
- PHY_JABBER_DETECTION : [stm32f4xx_hal_conf.h](#)
- PHY_LINK_INTERRUPT : [stm32f4xx_hal_conf.h](#)
- PHY_LINK_STATUS : [stm32f4xx_hal_conf.h](#)
- PHY_LINKED_STATUS : [stm32f4xx_hal_conf.h](#)
- PHY_LOOPBACK : [stm32f4xx_hal_conf.h](#)
- PHY_MICR : [stm32f4xx_hal_conf.h](#)
- PHY_MICR_INT_EN : [stm32f4xx_hal_conf.h](#)
- PHY_MICR_INT_OE : [stm32f4xx_hal_conf.h](#)
- PHY_MISR : [stm32f4xx_hal_conf.h](#)

- `PHY_MISR_LINK_INT_EN` : `stm32f4xx_hal_conf.h`
 - `PHY_POWERDOWN` : `stm32f4xx_hal_conf.h`
 - `PHY_READ_TO` : `stm32f4xx_hal_conf.h`
 - `PHY_RESET` : `stm32f4xx_hal_conf.h`
 - `PHY_RESET_DELAY` : `stm32f4xx_hal_conf.h`
 - `PHY_RESTART_AUTONEGOTIATION` : `stm32f4xx_hal_conf.h`
 - `PHY_SPEED_STATUS` : `stm32f4xx_hal_conf.h`
 - `PHY_SR` : `stm32f4xx_hal_conf.h`
 - `PHY_WRITE_TO` : `stm32f4xx_hal_conf.h`
 - `PI_parameters` : `6Step.Lib.c` , `stm32F401_nucleo_ihm11m1.c`
 - `POLESP_CMD` : `stm32F401_nucleo_ihm11m1.h`
 - `potent_filtered` : `6Step.Lib.c`
 - `POTENTIOMETER` : `MC_SixStep_param.h`
 - `POTENZ_CMD` : `stm32F401_nucleo_ihm11m1.h`
 - `PREFETCH_ENABLE` : `stm32f4xx_hal_conf.h`
-



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- r -

- Rotor_poles_pairs : [6Step_Lib.c](#)
- RUN : [6Step_Lib.h](#)
- RXBUFSIZE : [UART_UI.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- S -

- SET_DAC_value() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- Set_Ref_Generation() : [stm32F401_nucleo_ihm11m1.h](#)
- SETSPD_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- shift_n_sqrt : [6Step_Lib.c](#)
- SIXSTEP_Base_SystStatus_t : [6Step_Lib.h](#)
- SIXSTEP_parameters : [6Step_Lib.c](#) , [stm32F401_nucleo_ihm11m1.c](#) , [stm32f4xx_it.c](#) , [X-NUCLEO-IHM11M1.c](#)
- SIXSTEP_pi_PARAM_InitTypeDef_t : [6Step_Lib.h](#)
- speed_fdbk_error : [6Step_Lib.c](#)
- SPEED_LOOP_TIME : [MC_SixStep_param.h](#)
- speed_sum_pot_filt : [6Step_Lib.c](#)
- speed_sum_sp_filt : [6Step_Lib.c](#)
- speed_tmp_array : [6Step_Lib.c](#)
- speed_tmp_buffer : [6Step_Lib.c](#)
- SPEEDFBKERROR : [6Step_Lib.h](#)
- START : [6Step_Lib.h](#)
- START_DAC() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)

- Start_PWM_driving() : **STSPIN230.c** , **STSPIN230.h**
- START_Ref_Generation() : **stm32F401_nucleo_ihm11m1.h**
- STARTM_CMD : **stm32F401_nucleo_ihm11m1.h**
- STARTUP : **6Step_Lib.h**
- startup_bemf_failure : **6Step_Lib.c**
- STARTUP_BEMF_FAILURE : **6Step_Lib.h**
- STARTUP_CURRENT_REFERENCE : **MC_SixStep_param.h**
- STARTUP_DUTY_CYCLE : **MC_SixStep_param.h**
- STARTUP_FAILURE : **6Step_Lib.h**
- STATUS_CMD : **stm32F401_nucleo_ihm11m1.h**
- STOP : **6Step_Lib.h**
- STOP_DAC() : **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- Stop_PWM_driving() : **STSPIN230.c** , **STSPIN230.h**
- STOP_Ref_Generation() : **stm32F401_nucleo_ihm11m1.h**
- STOPMT_CMD : **stm32F401_nucleo_ihm11m1.h**
- STSPIN230_Current_Reference_Setvalue() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_Current_Reference_Start() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_Current_Reference_Stop() : **X-NUCLEO-IHM11M1.c** , **STSPIN230.h** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D() :
STSPIN230.h , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E() :
STSPIN230.h , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E() :
STSPIN230.h , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D() : **X-NUCLEO-IHM11M1.c** , **STSPIN230.h** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_HF_TIMx_SetDutyCycle_CH1() : **X-NUCLEO-IHM11M1.c** , **STSPIN230.h** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_HF_TIMx_SetDutyCycle_CH2() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
- STSPIN230_HF_TIMx_SetDutyCycle_CH3() : **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h** , **STSPIN230.h**
- STSPIN230_Set_Ref_Generation() : **STSPIN230.h**
- STSPIN230_Start_PWM_driving() : **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h** , **STSPIN230.h**
- STSPIN230_START_Ref_Generation() : **STSPIN230.h**

- STSPIN230_Stop_PWM_driving() : **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h** , **STSPIN230.h**
 - STSPIN230_STOP_Ref_Generation() : **STSPIN230.h**
 - STSPIN230MotorDriver : **stm32F401_nucleo_ihm11m1.c** , **STSPIN230.c**
 - SystemClock_Config() : **main_F401.c**
 - SysTick_Handler() : **stm32f4xx_it.c** , **stm32f4xx_it.h**
-

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations	Enumerator														
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- t -

- T_single_step : [6Step_Lib.c](#)
- T_single_step_first_value : [6Step_Lib.c](#)
- TARGET_SPEED : [MC_SixStep_param.h](#)
- target_speed : [6Step_Lib.c](#)
- test_motor_run : [6Step_Lib.c](#)
- Tick_cnt : [6Step_Lib.c](#)
- TICK_INT_PRIORITY : [stm32f4xx_hal_conf.h](#)
- TIM1_BRK_TIM9_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)
- TIM4_IRQHandler() : [stm32f4xx_it.h](#) , [stm32f4xx_it.c](#)
- TIME_FOR_ALIGN : [MC_SixStep_param.h](#)
- Time_vector_prev_tmp : [6Step_Lib.c](#)
- Time_vector_tmp : [6Step_Lib.c](#)
- TOKEN : [UART_UI.h](#)
- TRUE : [MC_SixStep_param.h](#)
- TXBUFSIZE : [UART_UI.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- u -

- UART : [stm32F401_nucleo_ihm11m1.h](#)
- UART_Communication_Task() : [6Step_Lib.c](#)
- UART_FLAG_RECEIVE : [6Step_Lib.c](#)
- UART_Set_Value() : [6Step_Lib.c](#) , [stm32f4xx_it.c](#)
- UPPER_OUT_LIMIT : [MC_SixStep_param.h](#)
- USART2_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)
- USERTOS : [stm32f4xx_hal_conf.h](#)
- uwTick : [6Step_Lib.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files																
File List	Globals																		
All	Functions	Variables	Typedefs	Enumerations															
Macros																			
a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	v

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- v -

- VAL_POT_SPEED_DIV : [MC_SixStep_param.h](#)
- VALIDATION : [6Step_Lib.h](#)
- VDD_VALUE : [stm32f4xx_hal_conf.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations						
Macros										
a	b	c	d	e	g	h	m	s	t	u

- a -

- ADC_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files		
File List	Globals				
All	Functions	Variables	TypeDefs	Enumerations	Enumerator
Macros					
a	b	c	d	e	g h m s t u

- b -

- Bemf_delay_calc() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- BSP_X_NUCLEOFAULT_LED_OFF() : [stm32F401_nucleo_ihm11m1.h](#) , [X-NUCLEO-IHM11M1.c](#)
- BSP_X_NUCLEOFAULT_LED_ON() : [stm32F401_nucleo_ihm11m1.h](#) , [X-NUCLEO-IHM11M1.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations						
Macros										
a	b	c	d	e	g	h	m	s	t	u

- C -

- CMD_ACCELE() : [UART_UI.h](#)
- CMD_DIRECTION() : [UART_UI.h](#)
- CMD_GETSPD() : [UART_UI.h](#)
- CMD_HELP() : [UART_UI.h](#)
- CMD_INIREF() : [UART_UI.h](#)
- CMD_KI_PRM() : [UART_UI.h](#)
- CMD_KP_PRM() : [UART_UI.h](#)
- CMD_Parser() : [6Step_Lib.c](#)
- CMD_POLESP() : [UART_UI.h](#)
- CMD_POTENZ() : [UART_UI.h](#)
- CMD_SETSPD() : [UART_UI.h](#)
- CMD_STARTM() : [UART_UI.h](#)
- CMD_STATUS() : [UART_UI.h](#)
- CMD_STOPMT() : [UART_UI.h](#)
- Current_Reference_Setvalue() : [STSPIN230.h](#) , [STSPIN230.c](#)
- Current_Reference_Start() : [STSPIN230.c](#) , [STSPIN230.h](#)
- Current_Reference_Stop() : [STSPIN230.h](#) , [STSPIN230.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs							
Macros										
a	b	c	d	e	g	h	m	s	t	u

- d -

- DisableInput_CH1_D_CH2_D_CH3_D() : **STSPIN230.c** , **STSPIN230.h**



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files			
File List	Globals					
All	Functions	Variables	TypeDefs	Enumerations	Enumerator	
Macros						
a	b	c	d	e	g	h
m	s	t	u			

- e -

- EnableInput_CH1_D_CH2_E_CH3_E() : [STSPIN230.c](#) , [STSPIN230.h](#)
- EnableInput_CH1_E_CH2_D_CH3_E() : [STSPIN230.h](#) , [STSPIN230.c](#)
- EnableInput_CH1_E_CH2_E_CH3_D() : [STSPIN230.c](#) , [STSPIN230.h](#)
- EXTI15_10_IRQHandler() : [stm32f4xx_it.h](#) , [stm32f4xx_it.c](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files	
File List	Globals			
All	Functions	Variables	Typedefs	Enumerations
Macros				
a	b	c	d	e
			g	h
			m	s
			t	u

- g -

- Get_UART_Data() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations						
Macros										
a	b	c	d	e	g	h	m	s	t	u

- h -

- HAL_ADC_ConvCpltCallback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_ADC_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_ADC_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_GetTick() : [6Step_Lib.c](#)
- HAL_GPIO_EXTI_Callback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_IncTick() : [6Step_Lib.c](#)
- HAL_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_SYSTICK_Callback() : [stm32F401_nucleo_ihm11m1.c](#)
- HAL_TIM_Base_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIM_Base_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIM_PeriodElapsedCallback() :
[stm32F401_nucleo_ihm11m1.c](#)
- HAL_TIM_PWM_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIMEx_HallSensor_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_TIMEx_HallSensor_MspInit() : [stm32f4xx_hal_msp.c](#)
- HAL_UART_MspDeInit() : [stm32f4xx_hal_msp.c](#)
- HAL_UART_MspInit() : [stm32f4xx_hal_msp.c](#)
- HF_TIMx_SetDutyCycle_CH1() : [STSPIN230.h](#) , [STSPIN230.c](#)
- HF_TIMx_SetDutyCycle_CH2() : [STSPIN230.c](#) , [STSPIN230.h](#)
- HF_TIMx_SetDutyCycle_CH3() : [STSPIN230.h](#) , [STSPIN230.c](#)

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations	Enumerator					
Macros										
a	b	c	d	e	g	h	m	s	t	u

- m -

- main() : [main_F401.c](#)
- MC_ADCx_SixStep_Bemf() : [6Step_Lib.c](#) , [stm32F401_nucleo_ihm11m1.c](#)
- MC_Bemf_Delay() : [6Step_Lib.c](#)
- MC_EXT_button_SixStep() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
- MC_GetElSpeedHz() : [6Step_Lib.c](#)
- MC_GetMechSpeedRPM() : [6Step_Lib.c](#)
- MC_PI_Controller() : [6Step_Lib.c](#)
- MC_Potentiometer_filter() : [6Step_Lib.c](#)
- MC_Set_PI_param() : [6Step_Lib.c](#)
- MC_Set_Speed() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
- MC_SixStep_ADC_Channel() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- MC_SixStep_Alignment() : [6Step_Lib.c](#)
- MC_SixStep_ARR_Bemf() : [6Step_Lib.c](#)
- MC_SixStep_ARR_step() : [6Step_Lib.c](#)
- MC_SixStep_Current_Reference_Setvalue() : [6Step_Lib.c](#) , [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- MC_SixStep_Current_Reference_Start() : [6Step_Lib.c](#) , [stm32F401_nucleo_ihm11m1.c](#) ,

stm32F401_nucleo_ihm11m1.h

- MC_SixStep_Current_Reference_Stop() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E() :
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h , **6Step_Lib.c**
- MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D() :
6Step_Lib.c , **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH1() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH2() : **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_HF_TIMx_SetDutyCycle_CH3() :
stm32F401_nucleo_ihm11m1.c , **6Step_Lib.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_INIT() : **6Step_Lib.h** , **6Step_Lib.c**
- MC_SixStep_Init_main_data() : **6Step_Lib.c**
- MC_SixStep_NEXT_step() : **6Step_Lib.c**
- MC_SixStep_Nucleo_Init() : **stm32F401_nucleo_ihm11m1.c** ,
stm32F401_nucleo_ihm11m1.h
- MC_SixStep_Ramp_Motor_calc() : **6Step_Lib.c**
- MC_SixStep_RESET() : **6Step_Lib.h** , **6Step_Lib.c**
- MC_SixStep_Speed_Potentiometer() : **6Step_Lib.c**
- MC_SixStep_Speed_Val_target_potentiometer() : **6Step_Lib.c**
- MC_SixStep_Start_PWM_driving() :
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h , **6Step_Lib.c**
- MC_SixStep_Stop_PWM_driving() :
stm32F401_nucleo_ihm11m1.c ,
stm32F401_nucleo_ihm11m1.h , **6Step_Lib.c**

- MC_SixStep_TABLE() : [6Step_Lib.c](#)
 - MC_Speed_Filter() : [6Step_Lib.c](#)
 - MC_StartMotor() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
 - MC_StopMotor() : [6Step_Lib.h](#) , [6Step_Lib.c](#)
 - MC_SysTick_SixStep_MediumFrequencyTask() : [6Step_Lib.c](#) ,
[stm32F401_nucleo_ihm11m1.c](#)
 - MC_Task_Speed() : [6Step_Lib.c](#)
 - MC_TIMx_SixStep_timebase() : [stm32F401_nucleo_ihm11m1.c](#)
, [6Step_Lib.c](#)
 - MC_UI_INIT() : [6Step_Lib.c](#)
 - MCM_Sqrt() : [6Step_Lib.c](#)
 - MX_ADC1_Init() : [main_F401.c](#)
 - MX_TIM1_Init() : [main_F401.c](#)
 - MX_TIM3_Init() : [main_F401.c](#)
 - MX_TIM4_Init() : [main_F401.c](#)
 - MX_USART2_UART_Init() : [main_F401.c](#)
-



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations	Enumerator					
Macros										
a	b	c	d	e	g	h	m	s	t	u

- S -

- SET_DAC_value() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- Set_Ref_Generation() : [stm32F401_nucleo_ihm11m1.h](#)
- START_DAC() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- Start_PWM_driving() : [STSPIN230.c](#) , [STSPIN230.h](#)
- START_Ref_Generation() : [stm32F401_nucleo_ihm11m1.h](#)
- STOP_DAC() : [stm32F401_nucleo_ihm11m1.c](#) , [stm32F401_nucleo_ihm11m1.h](#)
- Stop_PWM_driving() : [STSPIN230.c](#) , [STSPIN230.h](#)
- STOP_Ref_Generation() : [stm32F401_nucleo_ihm11m1.h](#)
- STSPIN230_Current_Reference_Setvalue() : [X-NUCLEO-IHM11M1.h](#) , [STSPIN230.h](#) , [X-NUCLEO-IHM11M1.c](#)
- STSPIN230_Current_Reference_Start() : [STSPIN230.h](#) , [X-NUCLEO-IHM11M1.c](#) , [X-NUCLEO-IHM11M1.h](#)
- STSPIN230_Current_Reference_Stop() : [STSPIN230.h](#) , [X-NUCLEO-IHM11M1.c](#) , [X-NUCLEO-IHM11M1.h](#)
- STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D() : [STSPIN230.h](#) , [X-NUCLEO-IHM11M1.c](#) , [X-NUCLEO-IHM11M1.h](#)
- STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E() : [STSPIN230.h](#) , [X-NUCLEO-IHM11M1.c](#) , [X-NUCLEO-IHM11M1.h](#)

- STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E() : **X-NUCLEO-IHM11M1.h** , **STSPIN230.h** , **X-NUCLEO-IHM11M1.c**
 - STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
 - STSPIN230_HF_TIMx_SetDutyCycle_CH1() : **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h** , **STSPIN230.h**
 - STSPIN230_HF_TIMx_SetDutyCycle_CH2() : **X-NUCLEO-IHM11M1.c** , **STSPIN230.h** , **X-NUCLEO-IHM11M1.h**
 - STSPIN230_HF_TIMx_SetDutyCycle_CH3() : **X-NUCLEO-IHM11M1.h** , **X-NUCLEO-IHM11M1.c** , **STSPIN230.h**
 - STSPIN230_Set_Ref_Generation() : **STSPIN230.h**
 - STSPIN230_Start_PWM_driving() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
 - STSPIN230_START_Ref_Generation() : **STSPIN230.h**
 - STSPIN230_Stop_PWM_driving() : **STSPIN230.h** , **X-NUCLEO-IHM11M1.c** , **X-NUCLEO-IHM11M1.h**
 - STSPIN230_STOP_Ref_Generation() : **STSPIN230.h**
 - SystemClock_Config() : **main_F401.c**
 - SysTick_Handler() : **stm32f4xx_it.h** , **stm32f4xx_it.c**
-



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations						
Macros										
a	b	c	d	e	g	h	m	s	t	u

- t -

- TIM1_BRK_TIM9_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)
- TIM4_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files							
File List	Globals									
All	Functions	Variables	Typedefs	Enumerations						
Macros										
a	b	c	d	e	g	h	m	s	t	u

- U -

- UART_Communication_Task() : [6Step_Lib.c](#)
- UART_Set_Value() : [6Step_Lib.c](#) , [stm32f4xx_it.c](#)
- USART2_IRQHandler() : [stm32f4xx_it.c](#) , [stm32f4xx_it.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files										
File List	Globals												
All	Functions	Variables	TypeDefs	Enumerations									
Macros													
a	b	c	d	e	h	i	m	n	p	r	s	t	u

- a -

- ARR_LF : [6Step.Lib.c](#)
- array_completed : [6Step.Lib.c](#)

- b -

- buffer_completed : [6Step.Lib.c](#)

- c -

- cnt_bemf_event : [6Step.Lib.c](#)
- constant_k : [6Step.Lib.c](#)
- constant_multiplier_tmp : [6Step.Lib.c](#)
- counter.ARR_Bemf : [6Step.Lib.c](#)

- d -

- dac_status : [6Step.Lib.c](#)
- delta : [6Step.Lib.c](#)

- e -

- El_Speed_Hz : **6Step_Lib.c**
- Enable_start_button : **6Step_Lib.c**

- h -

- hadc1 : **main_F401.c** , **stm32f4xx_it.c** , **main_F401.h**
- HFBuffer : **6Step_Lib.c**
- HFBufferIndex : **6Step_Lib.c**
- htim1 : **stm32f4xx_it.c** , **main_F401.h** , **main_F401.c**
- htim2 : **main_F401.c**
- htim3 : **main_F401.c** , **main_F401.h**
- htim4 : **main_F401.c** , **stm32f4xx_it.c** , **main_F401.h**
- huart2 : **main_F401.c** , **stm32f4xx_it.c** , **main_F401.h**

- i -

- index_adc_chn : **6Step_Lib.c**
- index_align : **6Step_Lib.c**
- index_ARR_step : **6Step_Lib.c**
- index_array : **6Step_Lib.c**
- index_motor_run : **6Step_Lib.c**
- index_pot_filt : **6Step_Lib.c**
- index_startup_motor : **6Step_Lib.c**

- m -

- mech_accel_hz : **6Step_Lib.c**
- Mech_Speed_RPM : **6Step_Lib.c**

- n -

- n_zcr_startup : **6Step_Lib.c**

- p -

- PI_parameters : **6Step_Lib.c** , **stm32F401_nucleo_ihm11m1.c**
- potent_filtered : **6Step_Lib.c**

- r -

- Rotor_poles_pairs : **6Step_Lib.c**

- **S** -

- shift_n_sqrt : **6Step_Lib.c**
- SIXSTEP_parameters : **X-NUCLEO-IHM11M1.c , stm32f4xx_it.c , 6Step_Lib.c , stm32F401_nucleo_ihm11m1.c**
- speed_fdbk_error : **6Step_Lib.c**
- speed_sum_pot_filt : **6Step_Lib.c**
- speed_sum_sp_filt : **6Step_Lib.c**
- speed_tmp_array : **6Step_Lib.c**
- speed_tmp_buffer : **6Step_Lib.c**
- startup_bemf_failure : **6Step_Lib.c**
- STSPIN230MotorDriver : **STSPIN230.c , stm32F401_nucleo_ihm11m1.c**

- **t** -

- T_single_step : **6Step_Lib.c**
- T_single_step_first_value : **6Step_Lib.c**
- target_speed : **6Step_Lib.c**
- test_motor_run : **6Step_Lib.c**
- Tick_cnt : **6Step_Lib.c**
- Time_vector_prev_tmp : **6Step_Lib.c**
- Time_vector_tmp : **6Step_Lib.c**

- **u** -

- UART_FLAG_RECEIVE : **6Step_Lib.c**
- uwTick : **6Step_Lib.c**



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files		
File List	Globals				
All	Functions	Variables	Typedefs	Enumerations	Enumerator
Macros					

- SIXSTEP_pi_PARAM_InitTypeDef_t : [6Step_Lib.h](#)

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files	
File List	Globals			
All	Functions	Variables	TypeDefs	Enumerations
Macros				

- SIXSTEP_Base_SystStatus_t : [6Step_Lib.h](#)

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files	
File List	Globals			
All	Functions	Variables	Typedefs	Enumerations
Macros				

- ALIGNMENT : [6Step_Lib.h](#)
- IDLE : [6Step_Lib.h](#)
- OVERCURRENT : [6Step_Lib.h](#)
- RUN : [6Step_Lib.h](#)
- SPEEDFBKERROR : [6Step_Lib.h](#)
- START : [6Step_Lib.h](#)
- STARTUP : [6Step_Lib.h](#)
- STARTUP_BEMF_FAILURE : [6Step_Lib.h](#)
- STARTUP_FAILURE : [6Step_Lib.h](#)
- STOP : [6Step_Lib.h](#)
- VALIDATION : [6Step_Lib.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files															
File List	Globals																	
All	Functions	Variables	Typedefs	Enumerations	Enumerator													
Macros																		
a	b	c	d	e	f	g	h	i	k	l	m	n	p	r	s	t	u	v

- a -

- ACC : [MC_SixStep_param.h](#)
- ACCELE_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH1 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH1_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH2 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH2_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH3 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_Bemf_CH3_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_1 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_1_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_2 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_2_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_3 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_3_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_4 : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_CH_4_ST : [stm32F401_nucleo_ihm11m1.h](#)
- ADC_SPEED_TH : [MC_SixStep_param.h](#)
- ADCx : [stm32F401_nucleo_ihm11m1.h](#)
- assert_param : [stm32f4xx_hal_conf.h](#)

- b -

- BEMF_CNT_EVENT_MAX : [MC_SixStep_param.h](#)
- BEMF_CONSEC_DOWN_MAX : [MC_SixStep_param.h](#)
- BEMF_THRSLD_DOWN : [MC_SixStep_param.h](#)
- BEMF_THRSLD_UP : [MC_SixStep_param.h](#)
- BUTTON_DELAY : [MC_SixStep_param.h](#)

- c -

- CMD_NUM : [UART_UI.h](#)
- COUNTOF : [UART_UI.h](#)

- d -

- DAC_ENABLE : [stm32F401_nucleo_ihm11m1.h](#)
- DACx : [stm32F401_nucleo_ihm11m1.h](#)
- DATA_CACHE_ENABLE : [stm32f4xx_hal_conf.h](#)
- DEMAGN_VAL_1 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_10 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_11 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_12 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_13 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_14 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_2 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_3 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_4 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_5 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_6 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_7 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_8 : [MC_SixStep_param.h](#)
- DEMAGN_VAL_9 : [MC_SixStep_param.h](#)
- DEMO_START_TIME : [MC_SixStep_param.h](#)
- DEMO_STOP_TIME : [MC_SixStep_param.h](#)
- DIRECT_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- DIRECTION : [MC_SixStep_param.h](#)
- DP83848_PHY_ADDRESS : [stm32f4xx_hal_conf.h](#)

- e -

- ETH_RX_BUF_SIZE : [stm32f4xx_hal_conf.h](#)
- ETH_RXBUFNFB : [stm32f4xx_hal_conf.h](#)

- ETH_TX_BUF_SIZE : [stm32f4xx_hal_conf.h](#)
- ETH_TXBUFN_B : [stm32f4xx_hal_conf.h](#)
- EXTERNAL_CLOCK_VALUE : [stm32f4xx_hal_conf.h](#)

- f -

- FALSE : [MC_SixStep_param.h](#)
- FILTER_DEEP : [MC_SixStep_param.h](#)

- g -

- GETSPD_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_CH_COMM : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_CH_ZCR : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_COMM : [MC_SixStep_param.h](#)
- GPIO_PORT_COMM : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_PORT_ZCR : [stm32F401_nucleo_ihm11m1.h](#)
- GPIO_ZERO_CROSS : [MC_SixStep_param.h](#)

- h -

- HAL_ADC_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_CORTEX_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_DMA_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_FLASH_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_GPIO_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_PWR_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_RCC_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_SPI_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_TIM_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_UART_MODULE_ENABLED : [stm32f4xx_hal_conf.h](#)
- HAL_ENCODER_TIMx : [stm32F401_nucleo_ihm11m1.h](#)
- HELP_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR1 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR2 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CCR3 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CH1 : [stm32F401_nucleo_ihm11m1.h](#)
- HF_TIMx_CH2 : [stm32F401_nucleo_ihm11m1.h](#)

- HF_TIMx_CH3 : [stm32F401_nucleo_ihm11m1.h](#)
- HFBUFFERSIZE : [MC_SixStep_param.h](#)
- HSE_STARTUP_TIMEOUT : [stm32f4xx_hal_conf.h](#)
- HSE_VALUE : [stm32f4xx_hal_conf.h](#)
- HSI_VALUE : [stm32f4xx_hal_conf.h](#)

- i -

- INIREF_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- INITIAL_DEMAGN_DELAY : [MC_SixStep_param.h](#)
- INSTRUCTION_CACHE_ENABLE : [stm32f4xx_hal_conf.h](#)

- k -

- KI_DIV : [MC_SixStep_param.h](#)
- KI_GAIN : [MC_SixStep_param.h](#)
- KI_PRM_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- KP_DIV : [MC_SixStep_param.h](#)
- KP_GAIN : [MC_SixStep_param.h](#)
- KP_PRM_CMD : [stm32F401_nucleo_ihm11m1.h](#)

- l -

- LF_TIMx : [stm32F401_nucleo_ihm11m1.h](#)
- LOWER_OUT_LIMIT : [MC_SixStep_param.h](#)
- LSE_VALUE : [stm32f4xx_hal_conf.h](#)
- LSI_VALUE : [stm32f4xx_hal_conf.h](#)

- m -

- MAC_ADDR0 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR1 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR2 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR3 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR4 : [stm32f4xx_hal_conf.h](#)
- MAC_ADDR5 : [stm32f4xx_hal_conf.h](#)
- MAX_POT_SPEED : [MC_SixStep_param.h](#)
- MIN_POT_SPEED : [MC_SixStep_param.h](#)
- MINIMUM_ACC : [MC_SixStep_param.h](#)

- n -

- NUM_POLE_PAIRS : **MC_SixStep_param.h**
- NUMBER_OF_STEPS : **MC_SixStep_param.h**
- NUMBER_ZCR : **MC_SixStep_param.h**

- p -

- PHY_AUTONEGO_COMPLETE : **stm32f4xx_hal_conf.h**
- PHY_AUTONEGOTIATION : **stm32f4xx_hal_conf.h**
- PHY_BCR : **stm32f4xx_hal_conf.h**
- PHY_BSR : **stm32f4xx_hal_conf.h**
- PHY_CONFIG_DELAY : **stm32f4xx_hal_conf.h**
- PHY_DUPLEX_STATUS : **stm32f4xx_hal_conf.h**
- PHY_FULLDUPLEX_100M : **stm32f4xx_hal_conf.h**
- PHY_FULLDUPLEX_10M : **stm32f4xx_hal_conf.h**
- PHY_HALFDUPLEX_100M : **stm32f4xx_hal_conf.h**
- PHY_HALFDUPLEX_10M : **stm32f4xx_hal_conf.h**
- PHY_ISOLATE : **stm32f4xx_hal_conf.h**
- PHY_JABBER_DETECTION : **stm32f4xx_hal_conf.h**
- PHY_LINK_INTERRUPT : **stm32f4xx_hal_conf.h**
- PHY_LINK_STATUS : **stm32f4xx_hal_conf.h**
- PHY_LINKED_STATUS : **stm32f4xx_hal_conf.h**
- PHY_LOOPBACK : **stm32f4xx_hal_conf.h**
- PHY_MICR : **stm32f4xx_hal_conf.h**
- PHY_MICR_INT_EN : **stm32f4xx_hal_conf.h**
- PHY_MICR_INT_OE : **stm32f4xx_hal_conf.h**
- PHY_MISR : **stm32f4xx_hal_conf.h**
- PHY_MISR_LINK_INT_EN : **stm32f4xx_hal_conf.h**
- PHY_POWERDOWN : **stm32f4xx_hal_conf.h**
- PHY_READ_TO : **stm32f4xx_hal_conf.h**
- PHY_RESET : **stm32f4xx_hal_conf.h**
- PHY_RESET_DELAY : **stm32f4xx_hal_conf.h**
- PHY_RESTART_AUTONEGOTIATION : **stm32f4xx_hal_conf.h**
- PHY_SPEED_STATUS : **stm32f4xx_hal_conf.h**
- PHY_SR : **stm32f4xx_hal_conf.h**
- PHY_WRITE_TO : **stm32f4xx_hal_conf.h**
- POLESP_CMD : **stm32F401_nucleo_ihm11m1.h**
- POTENTIOMETER : **MC_SixStep_param.h**
- POTENZ_CMD : **stm32F401_nucleo_ihm11m1.h**

- PREFETCH_ENABLE : [stm32f4xx_hal_conf.h](#)

- r -

- RXBUFFERSIZE : [UART_UI.h](#)

- s -

- SETSPD_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- SPEED_LOOP_TIME : [MC_SixStep_param.h](#)
- STARTM_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- STARTUP_CURRENT_REFERENCE : [MC_SixStep_param.h](#)
- STARTUP_DUTY_CYCLE : [MC_SixStep_param.h](#)
- STATUS_CMD : [stm32F401_nucleo_ihm11m1.h](#)
- STOPMT_CMD : [stm32F401_nucleo_ihm11m1.h](#)

- t -

- TARGET_SPEED : [MC_SixStep_param.h](#)
- TICK_INT_PRIORITY : [stm32f4xx_hal_conf.h](#)
- TIME_FOR_ALIGN : [MC_SixStep_param.h](#)
- TOKEN : [UART_UI.h](#)
- TRUE : [MC_SixStep_param.h](#)
- TXBUFFERSIZE : [UART_UI.h](#)

- u -

- UART : [stm32F401_nucleo_ihm11m1.h](#)
- UPPER_OUT_LIMIT : [MC_SixStep_param.h](#)
- USERTOS : [stm32f4xx_hal_conf.h](#)

- v -

- VAL_POT_SPEED_DIV : [MC_SixStep_param.h](#)
- VDD_VALUE : [stm32f4xx_hal_conf.h](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	Common

MC_Common.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      MC_Common.h
4  * @author    System lab - Automation and
Motion control team
5  * @version   V1.0.0
6  * @date     06-July-2015
7  * @brief    This header file is a common
file
8  ****
9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
```

```
15 * 1. Redistributions of source code must  
    retain the above copyright notice,  
16 *      this list of conditions and the  
    following disclaimer.  
17 * 2. Redistributions in binary form must  
    reproduce the above copyright notice,  
18 *      this list of conditions and the  
    following disclaimer in the documentation  
19 *      and/or other materials provided with  
    the distribution.  
20 * 3. Neither the name of  
    STMicroelectronics nor the names of its  
    contributors  
21 *      may be used to endorse or promote  
    products derived from this software  
22 *      without specific prior written  
    permission.  
23 *  
24 * THIS SOFTWARE IS PROVIDED BY THE  
    COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
25 * AND ANY EXPRESS OR IMPLIED WARRANTIES,  
    INCLUDING, BUT NOT LIMITED TO, THE  
26 * IMPLIED WARRANTIES OF MERCHANTABILITY AND  
    FITNESS FOR A PARTICULAR PURPOSE ARE  
27 * DISCLAIMED. IN NO EVENT SHALL THE  
    COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE  
28 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
    SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
29 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
    PROCUREMENT OF SUBSTITUTE GOODS OR  
30 * SERVICES; LOSS OF USE, DATA, OR PROFITS;  
    OR BUSINESS INTERRUPTION) HOWEVER  
31 * CAUSED AND ON ANY THEORY OF LIABILITY,  
    WHETHER IN CONTRACT, STRICT LIABILITY,  
32 * OR TORT (INCLUDING NEGLIGENCE OR  
    OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
33 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

```
POSSIBILITY OF SUCH DAMAGE.

34    *
35
***** *****
***** *****
36    */
37
38
39 /* Define to prevent recursive inclusion ---  
-----*/  

40 #ifndef __MC_COMMON_H  

41 #define __MC_COMMON_H  

42  

43 /** @addtogroup DRIVERS      DRIVERS  

44   * @brief Driver Layer  

45   * @{  

46   */  

47  

48 /** @addtogroup BSP      BSP  

49   * @brief BSP Layer  

50   * @{  

51   */  

52  

53 /** @addtogroup COMPONENTS      COMPONENTS  

54   * @brief Components  

55   * @{  

56   */  

57  

58 /** @addtogroup  

  STSPIN230_Motor_Driver_handler  

  STSPIN230_Motor_Driver_handler  

59   * @brief Handler for STSPIN230 Motor  

  driver  

60   * @{  

61   */  

62  

63 typedef struct
```

```
64|    {
65|        void (*EnableInput_CH1_E_CH2_E_CH3_D)
66|            (void); /*!< Enable the channel 1,2 and
67|            Disable the channel 3 */
68|        void (*EnableInput_CH1_E_CH2_D_CH3_E)
69|            (void); /*!< Enable the channel 1,3 and
70|            Disable the channel 2 */
71|        void (*EnableInput_CH1_D_CH2_E_CH3_E)
72|            (void); /*!< Enable the channel 2,3 and
73|            Disable the channel 1 */
74|        void (*DisableInput_CH1_D_CH2_D_CH3_D)
75|            (void); /*!< Disable all channels */
76|        void (*Start_PWM_driving)(void);
77|        /*!< Start PWM generation */
78|        void (*Stop_PWM_driving)(void);
79|        /*!< Stop PWM generation */
80|        void (*HF_TIMx_SetDutyCycle_CH1)
81|            (uint16_t); /*!< High Frequency Timer -
82|            Change DutyCycle value for CH1 */
83|        void (*HF_TIMx_SetDutyCycle_CH2)
84|            (uint16_t); /*!< High Frequency Timer -
85|            Change DutyCycle value for CH2 */
86|        void (*HF_TIMx_SetDutyCycle_CH3)
87|            (uint16_t); /*!< High Frequency Timer -
88|            Change DutyCycle value for CH3 */
89|        void (*Current_Reference_Start)(void);
90|        /*!< Start current reference generation for
91|        closed loop control */
92|        void (*Current_Reference_Stop)(void);
93|        /*!< Stop current reference generation for
94|        closed loop control */
95|        void (*Current_Reference_Setvalue)
96|            (uint16_t); /*!< Set current reference value
97|            for closed loop control */
98|    } STSPIN230_MotorDriver_TypeDef;
99|    /*!< MC driver handler */
100|
```

```
79  /**
80   * @}    end STSPIN230_Motor_Driver_handler
81   */
82
83 /**
84  * @}    end COMPONENTS
85 */
86
87 /**
88  * @}    end BSP
89 */
90
91 /**
92  * @}    end DRIVERS
93 */
94
95 #endif
96
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Src

stm32F401_nucleo_ihm11m1.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      stm32F401_nucleo_ihm11m1.c
4  * @author    IPC
5  * @version   V0
6  * @date      10/07/2016
7  * @brief     This file provides the interface
8  *             between the MC-lib and STM Nucleo F401xx
9  *
10 * COPYRIGHT(c) 2015 STMicroelectronics
11 *
12 * Redistribution and use in source and
13 * binary forms, with or without modification,
14 * are permitted provided that the
15 * following conditions are met:
16 *   1. Redistributions of source code must
17 *      retain the above copyright notice,
18 *      this list of conditions and the
```

following disclaimer.

16 | * 2. Redistributions in binary form must
reproduce the above copyright notice,
17 | * this list of conditions and the
following disclaimer in the documentation
18 | * and/or other materials provided
with the distribution.

19 | * 3. Neither the name of
STMicroelectronics nor the names of its
contributors

20 | * may be used to endorse or promote
products derived from this software

21 | * without specific prior written
permission.

22 | *

23 | * THIS SOFTWARE IS PROVIDED BY THE
COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
24 | * AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE
25 | * IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE
26 | * DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
27 | * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
28 | * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
29 | * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
30 | * CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY,
31 | * OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
32 | * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

33 | *

34 | *

```
*****
*****
35  */
36
37 #include "stm32F401_nucleo_ihm11m1.h"
38 #include "6Step_Lib.h"
39 #include "X-NUCLEO-IHM11M1.h"
40
41 extern SIXSTEP_Base_InitTypeDef
  SIXSTEP_parameters; /*!< Main SixStep
  structure*/
42 extern SIXSTEP_PI_PARAM_InitTypeDef_t
  PI_parameters; /*!< SixStep PI regulator
  structure*/
43 extern STSPIN230_MotorDriver_TypeDef
  STSPIN230MotorDriver;
44
45 extern void MC_ADCx_SixStep_Bemf(void);
46 extern void
  MC_TIMx_SixStep_timebase(void);
47 extern void
  MC_SysTick_SixStep_MediumFrequencyTask(void);
48
49 /** @addtogroup MIDDLEWARES      MIDDLEWARES
50 * @brief Middlewares Layer
51 * @{
52 */
53
54
55 /** @addtogroup MC_6-STEP_LIB      MC_6-
  STEP_LIB
56 * @brief Motor Control driver
57 * @{
58 */
59
60 /** @addtogroup stm32F401_nucleo_ihm11m1
  stm32F401_nucleo_ihm11m1
```

```
61     * @brief Interface file for STM32F401 and
62     * Motor Control Library configuration
63     */
64
65 /**
66 * @{
67 * @brief Select the new ADC Channel
68 */
69 /**
70 * @brief Select the new ADC Channel
71 * @param adc_ch
72 * @retval None
73 */
74 void MC_SixStep_ADC_Channel(uint32_t
adc_ch)
75 {
76     __HAL_ADC_DISABLE(&ADCx);
77     /* Clear the old SQx bits for the
selected rank */
78     ADCx.Instance->SQR3 &=
~ADC_SQR3_RK(ADC_SQR3_SQ1, 1);
79     /* Set the SQx bits for the selected
rank */
80     ADCx.Instance->SQR3 |=
ADC_SQR3_RK(adc_ch, 1);
81     __HAL_ADC_ENABLE(&ADCx);
82 }
83 /**
84 * @}
85 */
86
87 /**
88 * @{
89 * @brief Init the STM32 register
```

```
90 */
91 /**
92 * @brief Init the STM32 register
93 * @retval None
94 */
95 void MC_SixStep_Nucleo_Init()
96 {
97     TIM_ClearInputConfigTypeDef
98     sClearInputConfig;
99
100    /***** ETR CONFIGURATION
101    *****/
102    sClearInputConfig.ClearInputState = 1;
103    sClearInputConfig.ClearInputSource =
104        TIM_CLEARINPUTSOURCE_ETR;
105    sClearInputConfig.ClearInputPolarity =
106        TIM_CLEARINPUTPOLARITY_NONINVERTED;
107    sClearInputConfig.ClearInputPrescaler =
108        TIM_CLEARINPUTPRESCALER_DIV1;
109    sClearInputConfig.ClearInputFilter = 0;
110
111    HAL_TIM_ConfigOCrefClear(&HF_TIMx,
112        &sClearInputConfig, HF_TIMx_CH1);
113    HAL_TIM_ConfigOCrefClear(&HF_TIMx,
114        &sClearInputConfig, HF_TIMx_CH2);
115    HAL_TIM_ConfigOCrefClear(&HF_TIMx,
116        &sClearInputConfig, HF_TIMx_CH3);
117
118    /*****
119    *****/
120
121    __HAL_FREEZE_TIM1_DBGMCU(); /* Stop TIM
122        during Breakpoint */
123
124    __HAL_TIM_ENABLE_IT(&htim1,
125        TIM_IT_BREAK); /* Enable the TIM Break
126        interrupt */
```

```

114
115     /***** REGULAR CHANNELS
116     CONFIGURATION *****/
116     sConfig.Channel = ADC_CH_1; /* Current
117     feedabck */
117     sConfig.Rank = 1;
118     sConfig.SamplingTime = ADC_CH_1_ST;
119     sConfig.Offset = 0;
120     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
121     sConfig.Channel = ADC_CH_3; /* Bus
121     voltage */
122     sConfig.SamplingTime = ADC_CH_3_ST;
123     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
124     sConfig.Channel = ADC_CH_4; /* 
124     Temperature feedback */
125     sConfig.SamplingTime = ADC_CH_4_ST;
126     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
127     sConfig.Channel = ADC_Bemf_CH1; /* BEMF
127     feedback phase A */
128     sConfig.SamplingTime = ADC_Bemf_CH1_ST;
129     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
130     sConfig.Channel = ADC_Bemf_CH2; /* BEMF
130     feedback phase B */
131     sConfig.SamplingTime = ADC_Bemf_CH2_ST;
132     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
133     sConfig.Channel = ADC_Bemf_CH3; /* BEMF
133     feedback phase C */
134     sConfig.SamplingTime = ADC_Bemf_CH3_ST;
135     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
136     sConfig.Channel = ADC_CH_2; /* 
136     Potentiometer */
137     sConfig.SamplingTime = ADC_CH_2_ST;
138     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
139
139     *****/
140 }
```

```
141  /**
142   * @}
143   */
144
145
146  /** @defgroup START_DAC      START_DAC
147   * @{
148     @brief Start DAC for debug
149   */
150 /**
151   * @brief Start DAC for debug
152   * @retval None
153 */
154 void START_DAC()
155 {
156
157 }
158 /**
159  * @}
160 */
161 /** @defgroup STOP_DAC      STOP_DAC
162  * @{
163    @brief Stop DAC for debug
164  */
165 /**
166  * @brief Stop DAC for debug
167  * @retval None
168 */
169 void STOP_DAC()
170 {
171
172 }
173 /**
174  * @}
175 */
176 /** @defgroup SET_DAC_value  SET_DAC_value
177  * @{
```

```
178     * @brief Set DAC value for debug
179     */
180 /**
181 * @brief Set DAC value for debug
182 * @param dac_value: information to plot
183 * through DAC
184 * @retval None
185 */
186 void SET_DAC_value(uint16_t dac_value)
187 {
188 }
189 /**
190 * @}
191 */
192
193
194
195 /**
196 * @defgroup HAL_ADC_ConvCpltCallback
197 * HAL_ADC_ConvCpltCallback
198 * @{
199 */
200 /**
201 * @brief ADC callback
202 * @param hadc
203 * @retval None
204 */
205 void
206 HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
207 {
208     MC_ADCx_SixStep_Bemf();
209 }
210 /**
211 * @}
212 */
```

```
211
212 /**
213 * @defgroup HAL_TIM_PeriodElapsedCallback
214 * @{
215 * @brief htim callback
216 */
217 /**
218 * @brief htim callback
219 * @param htim
220 * @retval None
221 */
222 void
223 HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
224 {
225     MC_TIMx_SixStep_timebase();
226 }
227 /**
228 */
229 /**
230 * @defgroup HAL_SYSTICK_Callback
231 * @{
232 * @brief Systick callback
233 */
234 /**
235 * @brief Systick callback
236 * @retval None
237 */
238 void HAL_SYSTICK_Callback()
239 {
240     MC_SysTick_SixStep_MediumFrequencyTask();
241 }
242 */
```

```
243     * @}
244     */
245
246 /**
247 * @{
248 * @brief EXTI callback
249 */
250 /**
251 * @brief EXT callback
252 * @param GPIO_Pin
253 * @retval None
254 */
255
256 void HAL_GPIO_EXTI_Callback(uint16_t
257     GPIO_Pin)
258 {
259     MC_EXT_button_SixStep();
260 }
261 /**
262 * @}
263 */
264 /**
265 * @defgroup EnableInput_CH1_E_CH2_E_CH3_D
266 * @brief Enable Input channel for
267 * STSPIN230
268 */
269 /**
270 * @brief Enable Input channel CH1 and CH2
271 * for STSPIN230
272 * @retval None
273 */
274 void
275 MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D()
```

```
274 {
275     STSPIN230MotorDriver.EnableInput_CH1_E_CH2_E_C
276     H3_D();
277 }
278 /**
279 * @}
280 */
281
282 /** @defgroup EnableInput_CH1_E_CH2_D_CH3_E
283  * @{
284  * @brief Enable Input channel for
285  * STSPIN230
286  */
287 /**
288  * @brief Enable Input channel CH1 and CH3
289  * for STSPIN230
290  * @retval None
291 */
292 void
293 MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E()
294 {
295     STSPIN230MotorDriver.EnableInput_CH1_E_CH2_D_C
296     H3_E();
297 }
298 /**
299 * @}
300 */
301 /** @defgroup EnableInput_CH1_D_CH2_E_CH3_E
302  * @{
303  * @brief Enable Input channel for
304  * STSPIN230
305  */
306 /**
307  * @brief Enable Input channel CH1 and CH3
308  * for STSPIN230
309  * @retval None
310 */
311 void
312 MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E()
313 {
314     STSPIN230MotorDriver.EnableInput_CH1_D_CH2_E_C
315     H3_E();
316 }
```

```
302     * @brief Enable Input channel for  
303     STSPIN230  
304     */  
305     /**  
306     * @brief Enable Input channel CH2 and CH3  
307     for STSPIN230  
308     * @retval None  
309     */  
310     void  
311     MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E()  
312     {  
313         STSPIN230MotorDriver.EnableInput_CH1_D_CH2_E_C  
314         H3_E();  
315     }  
316     /**  
317     * @}  
318     /** @defgroup DisableInput_CH1_D_CH2_D_CH3_D  
319     * @{  
320     * @brief Disable All Input channels for  
321     STSPIN230  
322     */  
323     /**  
324     * @brief Enable Input channel CH2 and CH3  
325     for STSPIN230  
326     * @retval None  
327     */  
328     void  
329     MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D()  
330     {  
331 }
```

```
    STSPIN230MotorDriver.DisableInput_CH1_D_CH2_D_
    CH3_D();
330 }
331 /**
332 * @}
333 */
334
335 /**
336 ** @defgroup Start_PWM_driving
337 Start_PWM_driving
338 * @{
339     * @brief Enable the PWM generation on
340     Input channels
341 */
342 /**
343     * @brief Enable PWM channels for
344     STSPIN230
345     * @retval None
346 */
347 void MC_SixStep_Start_PWM_driving()
348 {
349     STSPIN230MotorDriver.Start_PWM_driving();
350 }
351 /**
352 * @}
353 */
354 /**
355 ** @defgroup Stop_PWM_driving
356 Stop_PWM_driving
357 * @{
358     * @brief Disable the PWM generation on
359     Input channels
360 */
361 /**
362     * @brief Disable PWM channels for
363     STSPIN230
364 */
365 void MC_SixStep_Stop_PWM_driving()
366 {
367     STSPIN230MotorDriver.Stop_PWM_driving();
368 }
```

```
STSPIN230
360     * @retval None
361 */
362
363 void MC_SixStep_Stop_PWM_driving()
364 {
365     STSPIN230MotorDriver.Stop_PWM_driving();
366 }
367
368 /**
369 * @}
370 */
371
372 /**
373 * @{
374     * @brief Set the Duty Cycle value for
375     * @}
376 /**
377     * @brief Set the Duty Cycle value for CH1
378     * @retval None
379 */
380
381 void
382     MC_SixStep_HF_TIMx_SetDutyCycle_CH1(uint16_t
383     CCR_value)
384 {
385
386     STSPIN230MotorDriver.HF_TIMx_SetDutyCycle_CH1(
387     CCR_value);
388 }
389 /**
390 * @}
391 */
```

```
390
391 /** @defgroup HF_TIMx_SetDutyCycle_CH2
392   HF_TIMx_SetDutyCycle_CH2
393   * @{
394   * @brief Set the Duty Cycle value for
395   * CH2
396   */
397 /**
398   * @brief Set the Duty Cycle value for CH2
399   * @retval None
400   */
401 void
402 MC_SixStep_HF_TIMx_SetDutyCycle_CH2(uint16_t
403 CCR_value)
404 {
405   STSPIN230MotorDriver.HF_TIMx_SetDutyCycle_CH2(
406   CCR_value);
407 }
408 /**
409 */
410
411
412
413 /** @defgroup HF_TIMx_SetDutyCycle_CH3
414   HF_TIMx_SetDutyCycle_CH3
415   * @{
416   * @brief Set the Duty Cycle value for
417   * CH3
418   */
419 /**
420   * @brief Set the Duty Cycle value for CH3
```

```
419     * @retval None
420 */
421
422
423 void
MC_SixStep_HF_TIMx_SetDutyCycle_CH3(uint16_t
CCR_value)
424 {
425
    STSPIN230MotorDriver.HF_TIMx_SetDutyCycle_CH3(
CCR_value);
426 }
427
428 /**
429 * @}
430 */
431
432 /** @defgroup Current_Reference_Start
Current_Reference_Start
433 * @{
434     * @brief Enable the Current Reference
generation
435 */
436 /**
437     * @brief Enable the Current Reference
generation
438     * @retval None
439 */
440
441 void MC_SixStep_Current_Reference_Start()
442 {
443
    STSPIN230MotorDriver.Current_Reference_Start();
444 }
445
446 /**
```

```
447 * @}
448 */
449
450
451 /** @defgroup Current_Reference_Stop
   Current_Reference_Stop
452 * @{
453 * @brief Disable the Current Reference
   generation
454 */
455 /**
456 * @brief Disable the Current Reference
   generation
457 * @retval None
458 */
459
460 void MC_SixStep_Current_Reference_Stop()
461 {
462     STSPIN230MotorDriver.Current_Reference_Stop();
463 }
464
465 /**
466 * @}
467 */
468
469
470 /** @defgroup Current_Reference_Setvalue
   Current_Reference_Setvalue
471 * @{
472 * @brief Set the value for Current
   Reference
473 */
474 /**
475 * @brief Set the value for Current
   Reference
476 * @retval None
```

```

477  */
478
479
480 void
    MC_SixStep_Current_Reference_Setvalue(uint16_t
        Iref)
481 {
482     STSPIN230MotorDriver.Current_Reference_Setvalue
        (Iref);
483 }
484
485 /**
486 * @}
487 */
488
489
490 /** @defgroup Bemf_delay_calc
    Bemf_delay_calc
491 * @{
492     * @brief Bemf delay calculation
493 */
494 /**
495     * @brief Bemf delay calculation
496     * @retval None
497 */
498 void Bemf_delay_calc()
499 {
500 if(PI_parameters.Reference>=0)
501 {
502
503     if(SIXSTEP_parameters.speed_fdbk_filtered<=120
          00 &&
          SIXSTEP_parameters.speed_fdbk_filtered>10000)
504     {
505         SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_2;

```

```
505     }
506     else
507         if(SIXSTEP_parameters.speed_fdbk_filtered<=100
508             00 &&
509             SIXSTEP_parameters.speed_fdbk_filtered>9400)
510         {
511             SIXSTEP_parameters.demagn_value =
512                 DEMAGN_VAL_3;
513         }
514     else
515         if(SIXSTEP_parameters.speed_fdbk_filtered<=940
516             0 &&
517             SIXSTEP_parameters.speed_fdbk_filtered>7600)
518         {
519             SIXSTEP_parameters.demagn_value =
520                 DEMAGN_VAL_4;
521         }
522     else
523         if(SIXSTEP_parameters.speed_fdbk_filtered<=760
524             0 &&
525             SIXSTEP_parameters.speed_fdbk_filtered>6000)
526         {
527             SIXSTEP_parameters.demagn_value =
528                 DEMAGN_VAL_5;
529         }
530     else
531         if(SIXSTEP_parameters.speed_fdbk_filtered<=600
532             0 &&
533             SIXSTEP_parameters.speed_fdbk_filtered>5400)
534         {
535             SIXSTEP_parameters.demagn_value =
536                 DEMAGN_VAL_6;
537         }
538     else
539         if(SIXSTEP_parameters.speed_fdbk_filtered<=540
540             0 &&
541             SIXSTEP_parameters.speed_fdbk_filtered>4750)
```

```
523     {
524         SIXSTEP_parameters.demagn_value =
525             DEMAGN_VAL_7;
526     }
527     else
528         if(SIXSTEP_parameters.speed_fdbk_filtered<=475
529             0 &&
530             SIXSTEP_parameters.speed_fdbk_filtered>4200)
531     {
532         SIXSTEP_parameters.demagn_value =
533             DEMAGN_VAL_8;
534     }
535     else
536         if(SIXSTEP_parameters.speed_fdbk_filtered<=420
537             0 &&
538             SIXSTEP_parameters.speed_fdbk_filtered>2600)
539     {
540         SIXSTEP_parameters.demagn_value =
541             DEMAGN_VAL_9;
542     }
543     else
544         if(SIXSTEP_parameters.speed_fdbk_filtered<=260
545             0 &&
546             SIXSTEP_parameters.speed_fdbk_filtered>1800)
547     {
548         SIXSTEP_parameters.demagn_value =
549             DEMAGN_VAL_10;
550     }
551     else
552         if(SIXSTEP_parameters.speed_fdbk_filtered<=180
553             0 &&
554             SIXSTEP_parameters.speed_fdbk_filtered>1500)
555     {
556         SIXSTEP_parameters.demagn_value =
557             DEMAGN_VAL_11;
558     }
559     else
```

```
    if(SIXSTEP_parameters.speed_fdbk_filtered<=150
0 &&
SIXSTEP_parameters.speed_fdbk_filtered>1300)
543    {
544        SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_12;
545    }
546 else
if(SIXSTEP_parameters.speed_fdbk_filtered<=130
0 &&
SIXSTEP_parameters.speed_fdbk_filtered>1000)
547    {
548        SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_13;
549    }
550 else
if(SIXSTEP_parameters.speed_fdbk_filtered<=100
0 &&
SIXSTEP_parameters.speed_fdbk_filtered>500)
551    {
552        SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_14;
553    }
554 }
555 else
556 {
557     if(SIXSTEP_parameters.speed_fdbk_filtered>=-12
000 &&
SIXSTEP_parameters.speed_fdbk_filtered<-10000)
558     {
559         SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_1;
560     }
561 else
if(SIXSTEP_parameters.speed_fdbk_filtered>=-10
000 &&
```

```
    SIXSTEP_parameters.speed_fdbk_filtered<-7800)
562    {
563        SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_2;
564    }
565    else
566        if(SIXSTEP_parameters.speed_fdbk_filtered>=-78
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-6400)
567        {
568            SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_3;
569        }
570        else
571            if(SIXSTEP_parameters.speed_fdbk_filtered>=-64
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-5400)
572        {
573            SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_4;
574        }
575        else
576            if(SIXSTEP_parameters.speed_fdbk_filtered>=-54
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-4650)
577        {
578            SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_5;
579        }
580    }
```

```
581 |     else
582 |     if(SIXSTEP_parameters.speed_fdbk_filtered>=-41
583 |         00 &&
584 |         SIXSTEP_parameters.speed_fdbk_filtered<-3650)
585 |     {
586 |         SIXSTEP_parameters.demagn_value =
587 |             DEMAGN_VAL_7;
588 |     }
589 |     else
590 |     if(SIXSTEP_parameters.speed_fdbk_filtered>=-36
591 |         50 &&
592 |         SIXSTEP_parameters.speed_fdbk_filtered<-3300)
593 |     {
594 |         SIXSTEP_parameters.demagn_value =
595 |             DEMAGN_VAL_8;
596 |     }
597 |     else
598 |     if(SIXSTEP_parameters.speed_fdbk_filtered>=-33
599 |         00 &&
600 |         SIXSTEP_parameters.speed_fdbk_filtered<-2650)
601 |     {
602 |         SIXSTEP_parameters.demagn_value =
603 |             DEMAGN_VAL_9;
604 |     }
605 |     else
606 |     if(SIXSTEP_parameters.speed_fdbk_filtered>=-26
607 |         00 &&
608 |         SIXSTEP_parameters.speed_fdbk_filtered<-1800)
609 |     {
610 |         SIXSTEP_parameters.demagn_value =
611 |             DEMAGN_VAL_10;
612 |     }
613 |     else
614 |     if(SIXSTEP_parameters.speed_fdbk_filtered>=-18
615 |         00 &&
616 |         SIXSTEP_parameters.speed_fdbk_filtered<-1500)
617 |     {
618 |         SIXSTEP_parameters.demagn_value =
619 |             DEMAGN_VAL_11;
620 |     }
621 | }
```

```

599     SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_11;
600 }
601 else
602 if(SIXSTEP_parameters.speed_fdbk_filtered>=-15
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-1300)
603 {
604     SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_12;
605 }
606 else
607 if(SIXSTEP_parameters.speed_fdbk_filtered>=-13
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-1000)
608 {
609     SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_13;
610 }
611 else
612 if(SIXSTEP_parameters.speed_fdbk_filtered>=-10
00 &&
SIXSTEP_parameters.speed_fdbk_filtered<-500)
613 {
614     SIXSTEP_parameters.demagn_value =
DEMAGN_VAL_14;
615 }
616
617 /**
618 * @}
619 */
620
621 /**
622 * @defgroup Get_UART_data    Get_UART_data
* @{

```

```
623     * @brief Get the UART value from DR
624     register
625     */
626     /***
627     * @brief Get the UART value from DR
628     * @register
629     * @retval uint32_t
630     */
631     uint32_t Get_UART_Data()
632     {
633         return (UART.Instance->DR);
634     }
635     /**
636
637     /**
638     * @} // end STM32F401_Interface
639     */
640
641 /**
642 * @} end MC_6-STEP_LIB
643 */
644
645 /**
646 * @} // end MIDDLEWARES
647 */
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	STSPIN230

STSPIN230.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      STSPIN230.c
4  * @author    IPC
5  * @version   V0
6  * @date     28-April-2016
7  * @brief    This file provides a set of
functions to manage STSPIN230 driver
8
****

9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
15 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34  *
35 ****
36 */
37
38 #include "STSPIN230.h"
39
40
41 /** @addtogroup DRIVERS      DRIVERS
42  * @brief Driver Layer
43  * @{
44  */
45
46 /** @addtogroup BSP      BSP
47  * @brief BSP Layer
48  * @{
49  */
50
51 /** @addtogroup COMPONENTS    COMPONENTS
52  * @brief Components
53  * @{
54  */
55
56 /** @addtogroup STSPIN230      STSPIN230
57  * @brief STSPIN230 driver section
58  * @{
59  */
60
61 /** @defgroup STSPIN230MotorDriver
62  * STSPIN230MotorDriver
63  * @{
64  * @brief API pointer for STSPIN230
65  */
66 /**
67  * @brief It handles all API functions for
68  * STSPIN230 MC Driver
```

```
67     * @retval None
68 */
69
70 STSPIN230_MotorDriver_TypeDef
71 STSPIN230MotorDriver =
72 {
73     EnableInput_CH1_E_CH2_E_CH3_D,
74     EnableInput_CH1_E_CH2_D_CH3_E,
75     EnableInput_CH1_D_CH2_E_CH3_E,
76     DisableInput_CH1_D_CH2_D_CH3_D,
77     Start_PWM_driving,
78     Stop_PWM_driving,
79     HF_TIMx_SetDutyCycle_CH1,
80     HF_TIMx_SetDutyCycle_CH2,
81     HF_TIMx_SetDutyCycle_CH3,
82     Current_Reference_Start,
83     Current_Reference_Stop,
84     Current_Reference_Setvalue,
85 };
86 /**
87 * @}
88 */
89
90 /** @defgroup EnableInput_CH1_E_CH2_E_CH3_D
91 * @{
92 * @brief Enable Input channel CH1 and CH2
93 * for STSPIN230
94 * @retval None
95 */
96 void EnableInput_CH1_E_CH2_E_CH3_D()
97 {
98     STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D();
99 }
100
```

```
101  /**
102  * @}
103  */
104
105
106 /**
107 * @defgroup EnableInput_CH1_E_CH2_D_CH3_E
108 * @brief Enable Input channel CH1 and CH3
109 for STSPIN230
110 * @retval None
111 */
112 void EnableInput_CH1_E_CH2_D_CH3_E()
113 {
114     STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E();
115 }
116
117 /**
118 * @}
119 */
120
121 /**
122 * @defgroup EnableInput_CH1_D_CH2_E_CH3_E
123 * @brief Enable Input channel CH2 and CH3
124 for STSPIN230
125 * @retval None
126 */
127 void EnableInput_CH1_D_CH2_E_CH3_E()
128 {
129     STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E();
130 }
131
132 /**
133 * @}
```

```
134     */
135
136
137 /**
138 * @defgroup DisableInput_CH1_D_CH2_D_CH3_D
139 * @brief Enable Input channel CH2 and CH3
140 for STSPIN230
141 * @retval None
142 */
143 void DisableInput_CH1_D_CH2_D_CH3_D()
144 {
145     STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D();
146 }
147
148 /**
149 * @}
150 */
151
152 /**
153 * @defgroup Start_PWM_driving
154 * @brief Enable PWM channels for
155 STSPIN230
156 * @retval None
157 */
158 void Start_PWM_driving()
159 {
160     STSPIN230_Start_PWM_driving();
161 }
162
163 /**
164 * @}
165 */
```

```
166
167 /**
168 * @defgroup Stop_PWM_driving
169 * @brief Disable PWM channels for
170 * @retval None
171 */
172
173 void Stop_PWM_driving()
174 {
175     STSPIN230_Stop_PWM_driving();
176 }
177
178 /**
179 * @}
180 */
181
182 /**
183 * @defgroup HF_TIMx_SetDutyCycle_CH1
184 * @brief Set the Duty Cycle value for CH1
185 * @retval None
186 */
187
188 void HF_TIMx_SetDutyCycle_CH1(uint16_t
189     CCR_value)
190 {
191     STSPIN230_HF_TIMx_SetDutyCycle_CH1(CCR_value);
192 }
193
194 /**
195 * @}
196 */
197
```

```
198  /** @defgroup HF_TIMx_SetDutyCycle_CH2
199   HF_TIMx_SetDutyCycle_CH2
200   * @{
201   * @brief Set the Duty Cycle value for CH2
202   * @retval None
203   */
204 void HF_TIMx_SetDutyCycle_CH2(uint16_t
205   CCR_value)
206 {
207
208   STSPIN230_HF_TIMx_SetDutyCycle_CH2(CCR_value);
209 }
210 /**
211  * @}
212 */
213 /** @defgroup HF_TIMx_SetDutyCycle_CH3
214   HF_TIMx_SetDutyCycle_CH3
215   * @{
216   * @brief Set the Duty Cycle value for CH3
217   * @retval None
218   */
219 void HF_TIMx_SetDutyCycle_CH3(uint16_t
220   CCR_value)
221 {
222
223   STSPIN230_HF_TIMx_SetDutyCycle_CH3(CCR_value);
224 }
225 /**
226  * @}
227 */
228
```

```
229
230
231 /** @defgroup Current_Reference_Start
   Current_Reference_Start
232 * @{
233 * @brief Enable the Current Reference
   generation
234 * @retval None
235 */
236
237 void Current_Reference_Start()
238 {
239   STSPIN230_Current_Reference_Start();
240 }
241
242 /**
243 * @}
244 */
245
246
247 /** @defgroup Current_Reference_Stop
   Current_Reference_Stop
248 * @{
249 * @brief Disable the Current Reference
   generation
250 * @retval None
251 */
252
253 void Current_Reference_Stop()
254 {
255   STSPIN230_Current_Reference_Stop();
256 }
257
258 /**
259 * @}
260 */
261
```

```
262
263 /** @defgroup Current_Reference_Setvalue
   Current_Reference_Setvalue
264 * @{
265 * @brief Set the value for Current
   Reference
266 * @retval None
267 */
268
269 void Current_Reference_Setvalue(uint16_t
   Iref)
270 {
271     STSPIN230_Current_Reference_Setvalue(Iref);
272 }
273
274 /**
275 * @}
276 */
277
278
279
280
281
282
283 /**
284 * @} end STSPIN230
285 */
286
287 /**
288 * @} end COMPONENTS
289 */
290
291 /**
292 * @} end BSP
293 */
294
```

```
295  /**
296  * @}    end DRIVERS
297 */
298
```

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	X-NUCLEO-IHM11M1	

X-NUCLEO-IHM11M1.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      X-NUCLEO-IHM11M1.c
4  * @author    IPC
5  * @version   V0
6  * @date     10/07/2016
7  * @brief    This file provides the set of
8  *           functions to manage the X-Nucleo expansion
9  *           board
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
12 * STMicroelectronics</center></h2>
13 *
14 * Redistribution and use in source and
15 * binary forms, with or without modification,
16 * are permitted provided that the following
17 * conditions are met:
```

```
15 * 1. Redistributions of source code must  
    retain the above copyright notice,  
16 *      this list of conditions and the  
    following disclaimer.  
17 * 2. Redistributions in binary form must  
    reproduce the above copyright notice,  
18 *      this list of conditions and the  
    following disclaimer in the documentation  
19 *      and/or other materials provided with  
    the distribution.  
20 * 3. Neither the name of  
    STMicroelectronics nor the names of its  
    contributors  
21 *      may be used to endorse or promote  
    products derived from this software  
22 *      without specific prior written  
    permission.  
23 *  
24 * THIS SOFTWARE IS PROVIDED BY THE  
    COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
25 * AND ANY EXPRESS OR IMPLIED WARRANTIES,  
    INCLUDING, BUT NOT LIMITED TO, THE  
26 * IMPLIED WARRANTIES OF MERCHANTABILITY AND  
    FITNESS FOR A PARTICULAR PURPOSE ARE  
27 * DISCLAIMED. IN NO EVENT SHALL THE  
    COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE  
28 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
    SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
29 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
    PROCUREMENT OF SUBSTITUTE GOODS OR  
30 * SERVICES; LOSS OF USE, DATA, OR PROFITS;  
    OR BUSINESS INTERRUPTION) HOWEVER  
31 * CAUSED AND ON ANY THEORY OF LIABILITY,  
    WHETHER IN CONTRACT, STRICT LIABILITY,  
32 * OR TORT (INCLUDING NEGLIGENCE OR  
    OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
33 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

```
POSSIBILITY OF SUCH DAMAGE.

34    *
35
36    */
37
38 /** @addtogroup DRIVERS      DRIVERS
39 * @brief Driver Layer
40 * @{
41 */
42
43 /** @addtogroup BSP      BSP
44 * @brief BSP Layer
45 * @{
46 */
47
48 /** @addtogroup X-NUCLEO-IHM11M1      X-
49   NUCLEO-IHM11M1
50 * @brief X-Nucleo expansion board
51 * @{
52 */
53
54 /* Includes -----
-----*/
55
56 #include "X-NUCLEO-IHM11M1.h"
57 #include "6Step_Lib.h"
58
59 #ifdef STM32F030x8
60 #include "stm32f0xx_hal.h"
61 #include "stm32F030_nucleo_ihm11m1.h"
62 #endif
63 #ifdef STM32F401xE
64 #include "stm32f4xx_hal.h"
65 #include "stm32F401_nucleo_ihm11m1.h"
```

```
66 #endif
67 extern SIXSTEP_Base_InitTypeDef
  SIXSTEP_parameters; /*!< Main SixStep
  structure*/
68
69 /** @defgroup
  STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D
  STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D
70 * @{
71   * @brief Enable Input channel for
  STSPIN230
72 */
73
74 /**
75   * @brief Enable Input channel CH1 and CH2
  for STSPIN230
76   * @retval None
77 */
78
79 void
  STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D()
80 {
81   HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH1);
  //TIM1_CH1 ENABLE
82   HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH1)
  ;
83
84   HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH2);
  //TIM1_CH2 ENABLE
85   HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH2)
  ;
86
87   HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH3);
  //TIM1_CH3 DISABLE
88   HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH3)
  ;
89 }
```

```
90
91 /**
92 * @}
93 */
94
95 /** @defgroup
96 STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E
97 STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E
98 * @{
99 * @brief Enable Input channel for
100 STSPIN230
101 */
102 /**
103 * @brief Enable Input channel CH1 and CH3
104 for STSPIN230
105 * @retval None
106 */
107
108 void
109 STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E()
110 {
111     HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH1);
//TIM1_CH1 ENABLE
112     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH1)
113 ;
114     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH2);
//TIM1_CH2 DISABLE
115     HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH2)
116 ;
117
118     HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH3);
//TIM1_CH3 ENABLE
119     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH3)
120 ;
121 }
```

```
116  /**
117  * @}
118 */
119
120 /** @defgroup
121   STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E
122   STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E
123   */
124 /**
125   * @brief Enable Input channel for
126   STSPIN230
127   */
128
129 void
130 STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E()
131 {
132     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH1);
133     //TIM1_CH1 DISABLE
134     HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH1)
135     ;
136
137     HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH2);
138     //TIM1_CH2 ENABLE
139     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH2)
140     ;
141
142     HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH3);
143     //TIM1_CH3 ENABLE
144     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH3)
145     ;
146 }
147 /**
148 */
```

```
142     * @}
143     */
144
145 /**
146 * @{
147 * @brief Disable All Input channels for
148 * STSPIN230
149 */
150 * @brief Enable Input channel CH2 and CH3
151 * for STSPIN230
152 * @retval None
153 */
154 void
155 STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D()
156 {
157     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH1);
//TIM1_CH1 DISABLE
158
159     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH2);
//TIM1_CH2 DISABLE
160
161     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH3);
//TIM1_CH3 DISABLE
162
163 }
164 */
165 /**
166 * @}
```

```
168     */
169
170 /**
171 * @{
172 * @brief Enable the PWM generation on
173 * Input channels for STSPIN230
174 */
175 /**
176 * @brief Enable PWM channels for
177 * STSPIN230
178 */
179 void STSPIN230_Start_PWM_driving()
180 {
181     HAL_TIM_PWM_Start(&HF_TIMx,
182                         HF_TIMx_CH1);           //TIM1_CH1 ENABLE
183
184     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH1) ;
185
186     HAL_TIM_PWM_Start(&HF_TIMx, HF_TIMx_CH2);
187     //TIM1_CH2 ENABLE
188
189     HAL_TIMEx_PWMN_Start(&HF_TIMx, HF_TIMx_CH2) ;
190
191 }
192 /**
193 * @}
194 */
195 /**
196 * @defgroup STSPIN230_Stop_PWM_driving
```

```
    STSPIN230_Stop_PWM_driving
196    * @{
197        * @brief Disable the PWM generation on
Input channels for STSPIN230
198    */
199 /**
200     * @brief Disable PWM channels for
STSPIN230
201     * @retval None
202 */
203
204 void STSPIN230_Stop_PWM_driving()
205 {
206     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH1);
//TIM1_CH1 DISABLE
207
    HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH1) ;
208
209     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH2);
//TIM1_CH2 DISABLE
210
    HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH2) ;
211
212     HAL_TIM_PWM_Stop(&HF_TIMx, HF_TIMx_CH3);
//TIM1_CH3 DISABLE
213
    HAL_TIMEx_PWMN_Stop(&HF_TIMx, HF_TIMx_CH3) ;
214
215 }
216
217 /**
218     * @}
219 */
220
221 /** @defgroup
STSPIN230_HF_TIMx_SetDutyCycle_CH1
STSPIN230_HF_TIMx_SetDutyCycle_CH1
```

```
222 * @{
223   * @brief Set the Duty Cycle value for
224   CH1 for STSPIN230
225 /**
226   * @brief Set the Duty Cycle value for CH1
227   * @retval None
228 /**
229
230 void
231   STSPIN230_HF_TIMx_SetDutyCycle_CH1(uint16_t
232     CCR_value)
233 {
234   HF_TIMx.Instance->HF_TIMx_CCR1 =
235     CCR_value;
236
237 /**
238   * @}
239 /**
240
241 /** @defgroup
242   STSPIN230_HF_TIMx_SetDutyCycle_CH2
243   STSPIN230_HF_TIMx_SetDutyCycle_CH2
244   * @{
245   * @brief Set the Duty Cycle value for
246   CH2 for STSPIN230
247 /**
248   * @brief Set the Duty Cycle value for CH2
249   * @retval None
250 /**
251 void
252   STSPIN230_HF_TIMx_SetDutyCycle_CH2(uint16_t
```

```
    CCR_value)
251 {           HF_TIMx.Instance->HF_TIMx_CCR2 =
252     CCR_value;
253
254 }
255 /**
256 * @}
257 */
258
259
260
261
262
263
264 /** @defgroup
265   STSPIN230_HF_TIMx_SetDutyCycle_CH3
266   STSPIN230_HF_TIMx_SetDutyCycle_CH3
267   * @{
268   * @brief Set the Duty Cycle value for
269   * CH3 for STSPIN230
270   */
271 /**
272   * @brief Set the Duty Cycle value for CH3
273   * @retval None
274 */
275
276 void
277 STSPIN230_HF_TIMx_SetDutyCycle_CH3(uint16_t
278 CCR_value)
279 {
280     HF_TIMx.Instance->HF_TIMx_CCR3 =
281     CCR_value;
282 }
```

```
280  /**
281  * @}
282  */
283
284 /** @defgroup
285   STSPIN230_Current_Reference_Start
286   STSPIN230_Current_Reference_Start
287   * @{
288   * @brief Enable the Current Reference
289   * generation for STSPIN230
290   */
291 /**
292   * @brief Enable the Current Reference
293   * generation
294   * @retval None
295   */
296
297 void STSPIN230_Current_Reference_Start()
298 {
299   Start_PWM_driving();
300
301   SIXSTEP_parameters.pulse_value=STARTUP_DUTY_CY
302   CLE;
303
304 /**
305  * @}
306  */
307
308
309 /** @defgroup
310   STSPIN230_Current_Reference_Stop
311   STSPIN230_Current_Reference_Stop
312   * @{
313   * @brief Disable the Current Reference
314   * generation for STSPIN230
315   */
316
```

```
308  /**
309   * @brief  Disable the Current Reference
310   *         generation
311   * @retval None
312
313 void STSPIN230_Current_Reference_Stop()
314 {
315     Stop_PWM_driving();
316
317     SIXSTEP_parameters.pulse_value=STARTUP_DUTY_CY
318     CLE;
319 }
320 /**
321 * @}
322 */
323
324 /**
325 * @defgroup
326 * @brief Set the value for Current
327 * Reference for STSPIN230
328 */
329 /**
330 * @brief Set the value for Current
331 * Reference
332 *
333 */
334 void
335 STSPIN230_Current_Reference_Setvalue(uint16_t
```

```
336     SIXSTEP_parameters.pulse_value=Iref;
337
338 }
339
340 /**
341 * @}
342 */
343
344
345 /** @defgroup BSP_X_NUCLEOFAULTLEDON
BSP_X_NUCLEOFAULTLEDON
346 * @{
347 * @brief Turns selected LED On.
348 * @retval None
349 */
350
351 void BSP_X_NUCLEOFAULTLEDON()
352 {
353 }
354
355 /**
356 * @}
357 */
358
359 /** @defgroup BSP_X_NUCLEOFAULTLEDOFF
BSP_X_NUCLEOFAULTLEDOFF
360 * @{
361 * @brief Turns selected LED Off.
362 * @retval None
363 */
364 void BSP_X_NUCLEOFAULTLEDOFF()
365 {
366 }
367 /**
368 * @}
369 */
370 */
```

```
371     * @}    end X-NUCLEO-IHM11M1
372     */
373
374 /**
375     * @}    end BSP
376     */
377
378 /**
379     * @}    end DRIVERS
380     */
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Src

6Step_Lib.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file    6Step_Lib.c
4  * @author  System lab - Automation and
5  Motion control team
6  * @version V1.0.0
7  * @date    06-July-2015
8  * @brief   This file provides the set of
functions for Motor Control library
9  ****
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
12 STMicroelectronics</center></h2>
13 *
14 * Redistribution and use in source and
binary forms, with or without modification,
15 * are permitted provided that the following
conditions are met:
```

```
15 * 1. Redistributions of source code must  
    retain the above copyright notice,  
16 *      this list of conditions and the  
    following disclaimer.  
17 * 2. Redistributions in binary form must  
    reproduce the above copyright notice,  
18 *      this list of conditions and the  
    following disclaimer in the documentation  
19 *      and/or other materials provided with  
    the distribution.  
20 * 3. Neither the name of  
    STMicroelectronics nor the names of its  
    contributors  
21 *      may be used to endorse or promote  
    products derived from this software  
22 *      without specific prior written  
    permission.  
23 *  
24 * THIS SOFTWARE IS PROVIDED BY THE  
    COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
25 * AND ANY EXPRESS OR IMPLIED WARRANTIES,  
    INCLUDING, BUT NOT LIMITED TO, THE  
26 * IMPLIED WARRANTIES OF MERCHANTABILITY AND  
    FITNESS FOR A PARTICULAR PURPOSE ARE  
27 * DISCLAIMED. IN NO EVENT SHALL THE  
    COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE  
28 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
    SPECIAL, EXEMPLARY, OR CONSEQUENTIAL  
29 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,  
    PROCUREMENT OF SUBSTITUTE GOODS OR  
30 * SERVICES; LOSS OF USE, DATA, OR PROFITS;  
    OR BUSINESS INTERRUPTION) HOWEVER  
31 * CAUSED AND ON ANY THEORY OF LIABILITY,  
    WHETHER IN CONTRACT, STRICT LIABILITY,  
32 * OR TORT (INCLUDING NEGLIGENCE OR  
    OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
33 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

POSSIBILITY OF SUCH DAMAGE.

```
34 *  
35 *****  
36 */  
37 /*!  
38 ======  
39 ======  
40 ##### Main functions for  
6-Step algorithm #####  
41 ======  
42 The main function are the following:  
43  
44 1) MC_SixStep_TABLE(...) -> Set the  
peripherals (TIMx, GPIO etc.) for each step  
45 2) MC_SixStep_ARR_step() -> Generate the ARR  
value for Low Frequency TIM during start-up  
46 3) MC_SixStep_INIT() -> Init the main  
variables for motor driving from  
MC_SixStep_param.h  
47 4) MC_SixStep_RESET() -> Reset all  
variables used for 6Step control algorithm  
48 5) MC_SixStep_Ramp_Motor_calc() -> Calculate  
the acceleration profile step by step for  
motor during start-up  
49 6) MC_SixStep_NEXT_step()-> Generate the  
next step number according with the direction  
(CW or CCW)  
50 7) MC_Task_Speed() -> Speed Loop with  
PI regulator  
51 8) MC_Set_Speed(...) -> Set the new  
motor speed value
```

```

52 | 9) MC_StartMotor()           -> Start the Motor
53 | 10)MC_StopMotor()          -> Stop the Motor
54 | ****
55 |
56 /* Includes -----
57  *-----*/
58
59 #include "6Step_Lib.h"
60
61 /** @addtogroup MIDDLEWARES      MIDDLEWARES
62  * @brief Middlewares Layer
63  * @{
64  */
65
66
67 /** @addtogroup MC_6-STEP_LIB      MC_6-
68  * STEP_LIB
69  * @brief Motor Control driver
70  * @{
71  */
72 /* Data struct -----
73  *-----*/
74 SIXSTEP_Base_InitTypeDef SIXSTEP_parameters;
75 /*!< Main SixStep structure*/
76 SIXSTEP_PI_PARAM_InitTypeDef_t
77 PI_parameters;           /*!< SixStep PI
78 regulator structure*/
79
80 /* Variables -----
81  *-----*/
82
83 uint16_t Rotor_poles_pairs;
84 /*!< Number of pole pairs of the motor */
85 uint32_t mech_accel_hz = 0;
86 /*!< Hz -- Mechanical acceleration rate */

```

```
79| uint32_t constant_k = 0;
/*!< 1/3*mech_accel_hz */
80| uint32_t Time_vector_tmp = 0;
/*!< Startup variable */
81| uint32_t Time_vector_prev_tmp = 0 ;
/*!< Startup variable */
82| uint32_t T_single_step = 0;
/*!< Startup variable */
83| uint32_t T_single_step_first_value = 0;
/*!< Startup variable */
84| int32_t delta = 0;
/*!< Startup variable */
85| uint16_t index_array = 1;
/*!< Speed filter variable */
86| int16_t speed_tmp_array[FILTER_DEEP];
/*!< Speed filter variable */
87| uint16_t speed_tmp_buffer[FILTER_DEEP];
/*!< Potentiometer filter variable */
88| uint16_t HFBuffer[HFBUFFERSIZE];
/*!< Buffer for Potentiometer Value Filtering
at the High-Frequency ADC conversion */
89| uint16_t HFBufferIndex = 0;
/*!< High-Frequency Buffer Index */
90| uint8_t array_completed = FALSE;
/*!< Speed filter variable */
91| uint8_t buffer_completed = FALSE;
/*!< Potentiometer filter variable */
92| uint8_t UART_FLAG_RECEIVE = FALSE;
/*!< UART communication flag */
93| uint32_t ARR_LF = 0;
/*!< Autoreload LF TIM variable */
94| int32_t Mech_Speed_RPM = 0;
/*!< Mechanical motor speed */
95| int32_t El_Speed_Hz = 0;
/*!< Electrical motor speed */
96| uint16_t index_adc_chn = 0;
/*!< Index of ADC channel selector for
```

```

    measuring */
97 | uint16_t index_motor_run = 0;
/*!< Tmp variable for DEMO mode */
98 | uint16_t test_motor_run = 1;
/*!< Tmp variable for DEMO mode */
99 | uint8_t Enable_start_button = TRUE;
/*!< Start/stop button filter to avoid double
   command */
100| uint16_t index.ARR_step = 1;
101| uint32_t n_zcr_startup = 0;
102| uint16_t index_startup_motor = 1;
103| uint16_t target_speed = TARGET_SPEED;
/*!< Target speed for closed loop control */
104| uint16_t shift_n_sqrt = 14;
105| uint16_t cnt_bemf_event = 0;
106| uint8_t startup_bemf_failure = 0;
107| uint8_t speed_fdbk_error = 0;
108| __IO uint32_t uwTick = 0;
/*!< Tick counter - 1msec updated */
109| uint8_t dac_status = DAC_ENABLE;
110| uint16_t index_align = 1;
111| int32_t speed_sum_sp_filt = 0;
112| int32_t speed_sum_pot_filt = 0;
113| uint16_t index_pot_filt = 1;
114| int16_t potent_filtered = 0;
115| uint32_t Tick_cnt = 0;
116| uint32_t counter.ARR_Bemf = 0;
117| uint64_t constant_multiplier_tmp = 0;
118|
119| int16_t
  MC_PI_Controller(SIXSTEP_PI_PARAM_InitTypeDef_
  t *, int16_t);
120| uint16_t MC_Potentiometer_filter(uint16_t );
121| uint64_t MCM_Sqrt(uint64_t );
122| int32_t MC_GetElSpeedHz(void);
123| int32_t MC_GetMechSpeedRPM(void);
124| void MC_SixStep_NEXT_step(void);

```

```
125 void MC_Speed_Filter(void);
126 void MC_SixStep_ARR_step(void);
127 void MC_SixStep_TABLE(uint8_t);
128 void MC_SixStep_Speed_Potentiometer(void);
129 void
    MC_Set_PI_param(SIXSTEP_PI_PARAM_InitTypeDef_t
    *);
130 void MC_Task_Speed(void);
131 void MC_SixStep_Alignment(void);
132 void MC_Bemf_Delay(void);
133 void MC_TIMx_SixStep_timebase(void);
134 void MC_ADCx_SixStep_Bemf(void);
135 void
    MC_SysTick_SixStep_MediumFrequencyTask(void);
136 void MC_SixStep_Ramp_Motor_calc(void);
137 void
    MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D(void);

138 void
    MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E(void);

139 void
    MC_SixStep_DisableInput_CH1_D_CH2_E_CH3_E(void);

140 void
    MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D(void
    );
141 void MC_SixStep_Start_PWM_driving(void);
142 void MC_SixStep_Stop_PWM_driving(void);
143 void
    MC_SixStep_HF_TIMx_SetDutyCycle_CH1(uint16_t);

144 void
    MC_SixStep_HF_TIMx_SetDutyCycle_CH2(uint16_t);

145 void
    MC_SixStep_HF_TIMx_SetDutyCycle_CH3(uint16_t);
```

```
146 void
    MC_SixStep_Current_Reference_Start(void);
147 void
    MC_SixStep_Current_Reference_Stop(void);
148 void
    MC_SixStep_Current_Reference_Setvalue(uint16_t
    );
149 void MC_SixStep_ARR_Bemf(uint8_t);
150 void MC_UI_INIT(void);
151 void UART_Set_Value(void);
152 void UART_Communication_Task(void);
153 void MC_SixStep_Init_main_data(void);
154 void CMD_Parser(char* pCommandString);
155 void
    MC_SixStep_Speed_Val_target_potentiometer(void
    );
156
157 /** @defgroup MC_SixStep_TABLE
    MC_SixStep_TABLE
158 * @{
159 * @brief Set the peripherals (TIMx, GPIO
    etc.) for each step
160 * @param step_number: step number
    selected
161 * @retval None
162 */
163
164 void MC_SixStep_TABLE(uint8_t step_number)
165 {
166     if(GPIO_COMM == 1)
167     {
168         HAL_GPIO_TogglePin(GPIO_PORT_COMM,GPIO_CH_COMM
    );
169     }
170 }
```

```
171     switch (step_number)
172     {
173         case 1:
174             {
175                 MC_SixStep_HF_TIMx_SetDutyCycle_CH1(SIXSTEP_pa
176                     rameters.pulse_value);
177
178                 MC_SixStep_HF_TIMx_SetDutyCycle_CH2(0);
179
180                 MC_SixStep_HF_TIMx_SetDutyCycle_CH3(0);
181
182                 MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D();
183
184                 SIXSTEP_parameters.CurrentRegular_BEMF_ch =
185                     SIXSTEP_parameters.Regular_channel[3];
186
187                 }
188
189                 break;
190
191                 case 2:
192
193                 {
```

```
    MC_SixStep_HF_TIMx_SetDutyCycle_CH2(SIXSTEP_pa
    rameters.pulse_value);
194|
    MC_SixStep_HF_TIMx_SetDutyCycle_CH3(0);
195|
    MC_SixStep_HF_TIMx_SetDutyCycle_CH1(0);
196|
    MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E();
197|
    SIXSTEP_parameters.CurrentRegular_BEMF_ch =
    SIXSTEP_parameters.Regular_channel[1];
198|
    }
199|
    break;
200|
    case 4:
201|
    {
202|
        MC_SixStep_HF_TIMx_SetDutyCycle_CH2(SIXSTEP_pa
        rameters.pulse_value);
203|
        MC_SixStep_HF_TIMx_SetDutyCycle_CH1(0);
204|
        MC_SixStep_HF_TIMx_SetDutyCycle_CH3(0);
205|
        MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D();
206|
        SIXSTEP_parameters.CurrentRegular_BEMF_ch =
        SIXSTEP_parameters.Regular_channel[3];
207|
        }
208|
        break;
209|
        case 5:
210|
        {
211|
            MC_SixStep_HF_TIMx_SetDutyCycle_CH3(SIXSTEP_pa
            rameters.pulse_value);
212|
            MC_SixStep_HF_TIMx_SetDutyCycle_CH1(0);
213|
```

```

        MC_SixStep_HF_TIMx_SetDutyCycle_CH2(0);
214    MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E());
215    SIXSTEP_parameters.CurrentRegular_BEMF_ch =
SIXSTEP_parameters.Regular_channel[2];
216    }
217    break;
218    case 6:
219    {
220
        MC_SixStep_HF_TIMx_SetDutyCycle_CH3(SIXSTEP_pa
rameters.pulse_value);
221
        MC_SixStep_HF_TIMx_SetDutyCycle_CH2(0);
222
        MC_SixStep_HF_TIMx_SetDutyCycle_CH1(0);
223
        MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E());
224
        SIXSTEP_parameters.CurrentRegular_BEMF_ch =
SIXSTEP_parameters.Regular_channel[1];
225    }
226    break;
227}
228}
229/**
230 * @}
231 */
232
233
234 /** @defgroup MC_SixStep_NEXT_step
MC_SixStep_NEXT_step
235 * @{
236     * @brief Generate the next step number
according with the direction (CW or CCW)
237     * @retval uint8_t

```

```

        SIXSTEP_parameters.status
238     */
239 void MC_SixStep_NEXT_step()
240 {
241
242     if(SIXSTEP_parameters.CMD == TRUE)
243     {
244         SIXSTEP_parameters.CMD = FALSE;
245         MC_SixStep_Start_PWM_driving();
246     }
247     ARR_LF =
248     __HAL_TIM_GetAutoreload(&LF_TIMx);
249
250     if(SIXSTEP_parameters.ALIGN_OK == TRUE)
251     {
252         SIXSTEP_parameters.speed_fdbk =
253         MC_GetMechSpeedRPM();
254         SIXSTEP_parameters.demagn_counter = 1;
255         if(SIXSTEP_parameters.status_prev !=
256             SIXSTEP_parameters.step_position)
257         {
258             n_zcr_startup = 0;
259         }
260         if(PI_parameters.Reference>=0)
261         {
262             SIXSTEP_parameters.step_position++;
263             if(SIXSTEP_parameters.CL_READY == TRUE)
264             {
265                 SIXSTEP_parameters.VALIDATION_OK =
266                 TRUE;
267             }
268         }
269     }

```

```

270     {
271         SIXSTEP_parameters.step_position--;
272         if(SIXSTEP_parameters.CL_READY == TRUE)
273         {
274             SIXSTEP_parameters.VALIDATION_OK =
275             TRUE;
276         }
277         if(SIXSTEP_parameters.step_position < 1)
278         {
279             SIXSTEP_parameters.step_position = 6;
280         }
281     }
282
283     if(SIXSTEP_parameters.VALIDATION_OK == 1)
284     {
285         /* Motor Stall condition detection and
286          Speed-Feedback error generation */
287         SIXSTEP_parameters.BEMF_Tdown_count++;
288         if
289             (SIXSTEP_parameters.BEMF_Tdown_count>BEMF_CONS
290             EC_DOWN_MAX)
291         {
292             speed_fdbk_error = 1;
293         }
294         else
295         {
296             __HAL_TIM_SetAutoreload(&LF_TIMx, 0xFFFF);
297         }
298     }
299
300     MC_SixStep_TABLE(SIXSTEP_parameters.step_posit
301     ion);
302
303     /* It controls if the changing step
304      request appears during DOWNcounting

```

```
299     * in this case it changes the ADC
  channel */
300
301     /* UP-COUNTING direction started DIR =
  0*/
302     if(__HAL_TIM_DIRECTION_STATUS(&HF_TIMX))
303     {
304         switch (SIXSTEP_parameters.step_position)
305         {
306             case 1:
307                 {
308                     SIXSTEP_parameters.CurrentRegular_BEMF_ch =
309                     SIXSTEP_parameters.Regular_channel[3];
310                 }
311                 break;
312             case 2:
313                 {
314                     SIXSTEP_parameters.CurrentRegular_BEMF_ch =
315                     SIXSTEP_parameters.Regular_channel[2];
316                 }
317                 break;
318             case 3:
319                 {
320                     SIXSTEP_parameters.CurrentRegular_BEMF_ch =
321                     SIXSTEP_parameters.Regular_channel[1];
322                 }
323                 break;
324             case 4:
325                 {
326                     SIXSTEP_parameters.CurrentRegular_BEMF_ch =
327                     SIXSTEP_parameters.Regular_channel[3];
328                 }
329                 break;
```

```
326     case 5:
327     {
328         SIXSTEP_parameters.CurrentRegular_BEMF_ch =
329             SIXSTEP_parameters.Regular_channel[2];
330     }
331     break;
332     case 6:
333     {
334         SIXSTEP_parameters.CurrentRegular_BEMF_ch =
335             SIXSTEP_parameters.Regular_channel[1];
336     }
337     break;
338 } /* end switch case*/
339
340
341 }
342
343 /**
344 * @}
345 */
346
347 /** @defgroup MC_SixStep_RESET
348  MC_SixStep_RESET
349  * @{
350  * @brief Reset all variables used for
351  * 6Step control algorithm
352  * @retval None
353  */
354 void MC_SixStep_RESET()
355 {
```

```
355 SIXSTEP_parameters.CMD = TRUE;
356 SIXSTEP_parameters.numberofitemArr =
    NUMBER_OF_STEPS;
357 SIXSTEP_parameters.ADC_BEMF_threshold_UP =
    BEMF_THRSLD_UP;
358 SIXSTEP_parameters.ADC_BEMF_threshold_DOWN
    = BEMF_THRSLD_DOWN;
359 SIXSTEP_parameters.Ireference =
    STARTUP_CURRENT_REFERENCE;
360 SIXSTEP_parameters.Speed_Loop_Time =
    SPEED_LOOP_TIME;
361 SIXSTEP_parameters.pulse_value =
    SIXSTEP_parameters.HF_TIMx_CCR;
362 SIXSTEP_parameters.Speed_target_ramp =
    MAX_POT_SPEED;
363 SIXSTEP_parameters.ALIGNMENT = FALSE;
364 SIXSTEP_parameters.Speed_Ref_filtered = 0;
365 SIXSTEP_parameters.demagn_value =
    INITIAL_DEMAGN_DELAY;
366
367 SIXSTEP_parameters.CurrentRegular_BEMF_ch =
    0;
368 SIXSTEP_parameters.status_prev = 0;
369 SIXSTEP_parameters.step_position = 0;
370
371
372 LF_TIMx.Init.Prescaler =
    SIXSTEP_parameters.LF_TIMx_PSC;
373 LF_TIMx.Instance->PSC =
    SIXSTEP_parameters.LF_TIMx_PSC;
374 LF_TIMx.Init.Period =
    SIXSTEP_parameters.LF_TIMx_ARR;
375 LF_TIMx.Instance->ARR =
    SIXSTEP_parameters.LF_TIMx_ARR;
376 HF_TIMx.Init.Prescaler =
    SIXSTEP_parameters.HF_TIMx_PSC;
377 HF_TIMx.Instance->PSC =
```

```

        SIXSTEP_parameters.HF_TIMx_PSC;
378 | HF_TIMx.Init.Period =
        SIXSTEP_parameters.HF_TIMx_ARR;
379 | HF_TIMx.Instance->ARR =
        SIXSTEP_parameters.HF_TIMx_ARR;
380 | HF_TIMx.Instance->HF_TIMx_CCR1 =
        SIXSTEP_parameters.HF_TIMx_CCR;
381 |
382 | Rotor_poles_pairs =
        SIXSTEP_parameters.NUMPOLESPAIRS;
383 | SIXSTEP_parameters.SYCLK_frequency =
        HAL_RCC_GetSysClockFreq();
384 |
385 | MC_SixStep_HF_TIMx_SetDutyCycle_CH1(0);
386 | MC_SixStep_HF_TIMx_SetDutyCycle_CH2(0);
387 | MC_SixStep_HF_TIMx_SetDutyCycle_CH3(0);
388 |
389 | SIXSTEP_parameters.Regular_channel[1] =
        ADC_Bemf_CH1; /*BEMF1*/
390 | SIXSTEP_parameters.Regular_channel[2] =
        ADC_Bemf_CH2; /*BEMF2*/
391 | SIXSTEP_parameters.Regular_channel[3] =
        ADC_Bemf_CH3; /*BEMF3*/
392 | SIXSTEP_parameters.ADC_SEQ_CHANNEL[0] =
        ADC_CH_1; /*CURRENT*/
393 | SIXSTEP_parameters.ADC_SEQ_CHANNEL[1] =
        ADC_CH_2; /*SPEED*/
394 | SIXSTEP_parameters.ADC_SEQ_CHANNEL[2] =
        ADC_CH_3; /*VBUS*/
395 | SIXSTEP_parameters.ADC_SEQ_CHANNEL[3] =
        ADC_CH_4; /*TEMP*/
396 |
397 | SIXSTEP_parameters.step_position = 0;
398 | SIXSTEP_parameters.demagn_counter = 0;
399 | SIXSTEP_parameters.ALIGN_OK = FALSE;
400 | SIXSTEP_parameters.VALIDATION_OK = 0;
401 | SIXSTEP_parameters.ARR_OK = 0;

```

```
402 SIXSTEP_parameters.speed_fdbk_filtered = 0;
403 SIXSTEP_parameters.Integral_Term_sum = 0;
404 SIXSTEP_parameters.Current_Reference = 0;
405 SIXSTEP_parameters.Ramp_Start = 0;
406 SIXSTEP_parameters.RUN_Motor = 0;
407 SIXSTEP_parameters.speed_fdbk = 0;
408 SIXSTEP_parameters.BEMF_OK = FALSE;
409 SIXSTEP_parameters.CL_READY = FALSE;
410 SIXSTEP_parameters.SPEED_VALIDATED = FALSE;
411 SIXSTEP_parameters.BEMF_Tdown_count = 0;
/* Reset of the Counter to detect Stop motor
condition when a stall condition occurs*/
412
413 uwTick = 0;
414 index_motor_run = 0;
415 test_motor_run = 1;
416 T_single_step = 0;
417 T_single_step_first_value = 0;
418 delta = 0;
419 Time_vector_tmp = 0;
420 Time_vector_prev_tmp = 0;
421 Mech_Speed_RPM = 0;
422 El_Speed_Hz = 0;
423 index_adc_chn = 0;
424 mech_accel_hz = 0;
425 constant_k = 0;
426 ARR_LF = 0;
427 index_array = 1;
428 Enable_start_button = TRUE;
429 index_ARR_step = 1;
430 n_zcr_startup = 0;
431 cnt_bemf_event = 0;
432 startup_bemf_failure = 0;
433 speed_fdbk_error = 0;
434
435 index_align = 1;
436 speed_sum_sp_filt = 0;
```

```

437 speed_sum_pot_filt = 0;
438 index_pot_filt = 1;
439 potent_filtered = 0;
440 Tick_cnt = 0;
441 counter.ARR_Bemf = 0;
442 constant_multiplier_tmp = 0;
443
444 HFBufferIndex =0;
445 for(uint16_t i = 0; i < HFBUFFERSIZE;i++)
446 {
447   HFBuffer[i]=0;
448 }
449
450 for(uint16_t i = 0; i < FILTER_DEEP;i++)
451 {
452   speed_tmp_array[i] = 0;
453   speed_tmp_buffer[i]= 0;
454 }
455 array_completed = FALSE;
456 buffer_completed = FALSE;
457
458 if(PI_parameters.Reference < 0)
459 {
460   SIXSTEP_parameters.step_position = 1;
461 }
462 target_speed = TARGET_SPEED;
463 MC_Set_PI_param(&PI_parameters);
464 MC_SixStep_Current_Reference_Start();
465
MC_SixStep_Current_Reference_Setvalue(SIXSTEP_
parameters.Ireference);
466 index_startup_motor = 1;
467 MC_SixStep_Ramp_Motor_calc();
468 }
469
470 /**
471 * @}

```

```

472    */
473
474 /** @defgroup MC_SixStep_Ramp_Motor_calc
   MC_SixStep_Ramp_Motor_calc
475 * @{
476     * @brief Calculate the acceleration
       profile step by step for motor during start-up
477     * @retval None
478 */
479 void MC_SixStep_Ramp_Motor_calc()
480 {
481     uint32_t constant_multiplier = 100;
482     uint32_t constant_multiplier_2 =
4000000000;
483
484     if(index_startup_motor == 1)
485     {
486         mech_accel_hz = SIXSTEP_parameters.ACCEL
        * Rotor_poles_pairs / 60;
487         constant_multiplier_tmp =
        (uint64_t)constant_multiplier*
        (uint64_t)constant_multiplier_2;
488         constant_k =
        constant_multiplier_tmp/(3*mech_accel_hz);
489
        MC_SixStep_Current_Reference_Setvalue(SIXSTEP_
parameters.Ireference);
490         Time_vector_prev_tmp = 0;
491     }
492     if(index_startup_motor < NUMBER_OF_STEPS)
493     {
494         Time_vector_tmp = ((uint64_t) 1000 *
        (uint64_t)1000 * (uint64_t)
        MCM_Sqrt(((uint64_t)index_startup_motor *
        (uint64_t)constant_k)))/632455;
495         delta = Time_vector_tmp -
        Time_vector_prev_tmp;

```

```

496     if(index_startup_motor==1)
497     {
498         T_single_step_first_value = (2 *
3141)*delta/1000;
499         SIXSTEP_parameters.ARR_value =
(uint32_t)(65535);
500     }
501     else
502     {
503         T_single_step = (2 * 3141)*delta/1000;
504         SIXSTEP_parameters.ARR_value =
(uint32_t)(65535 *
T_single_step)/(T_single_step_first_value);
505     }
506 }
507 else index_startup_motor=1;
508
509 if(index_startup_motor==1)
510 {
511     SIXSTEP_parameters.prescaler_value =
(((SIXSTEP_parameters.SYSCLK_frequency/10000000
)*T_single_step_first_value)/65535) - 1;
512 }
513 if(SIXSTEP_parameters.STATUS != ALIGNMENT
&& SIXSTEP_parameters.STATUS != START)
514 {
515     index_startup_motor++;
516 }
517 else Time_vector_tmp = 0;
518 Time_vector_prev_tmp = Time_vector_tmp;
519
520 }
521
522 /**
523 * @}
524 */
525

```

```
526 /**
527  * @brief It calculates the square root of
528  * a non-negative s64.
529  * It returns 0 for negative s64.
530  * @param Input uint64_t number
531  * @retval int32_t Square root of Input (0
532  * if Input<0)
533 */
534 uint64_t MCM_Sqrt(uint64_t wInput)
535 {
536     uint8_t biter = 0u;
537     uint64_t wtemproot;
538     uint64_t wtemprootnew;
539
540     if (wInput <= (uint64_t)
541         ((uint64_t)2097152<<shift_n_sqrt))
542     {
543         wtemproot = (uint64_t)
544             ((uint64_t)128<<shift_n_sqrt);
545     }
546     else
547     {
548         wtemproot = (uint64_t)
549             ((uint64_t)8192<<shift_n_sqrt);
550     }
551
552     do
553     {
554         wtemprootnew = (wtemproot +
555             wInput/wtemproot)>>1;
556         if (wtemprootnew == wtemproot)
557         {
558             biter = (shift_n_sqrt-1);
559         }
560         else
561         {
```

```

557         biter++;
558         wtemproot = wtemprootnew;
559     }
560 }
561 while (biter < (shift_n_sqrt-1));
562
563 return (wtemprootnew);
564 }
565
566
567 /** @defgroup MC_SixStep_ARR_step
568  * MC_SixStep_ARR_step
569  * @brief Generate the ARR value for Low
570  * Frequency TIM during start-up
571  * @retval None
572 */
573 void MC_SixStep_ARR_step()
574 {
575
576 if(SIXSTEP_parameters.ALIGNMENT == FALSE)
577 {
578     SIXSTEP_parameters.ALIGNMENT = TRUE;
579 }
580 if(SIXSTEP_parameters.ALIGN_OK == TRUE)
581 {
582     if(PI_parameters.Reference >= 0)
583     {
584         if(SIXSTEP_parameters.VALIDATION_OK != TRUE)
585         {
586             SIXSTEP_parameters.STATUS = STARTUP;
587             MC_SixStep_Ramp_Motor_calc();
588             if(index_ARR_step <
589                 SIXSTEP_parameters.numberofitemArr)

```

```
590         LF_TIMx.Init.Period =
591             SIXSTEP_parameters.ARR_value;
592         LF_TIMx.Instance->ARR =
593             (uint32_t)LF_TIMx.Init.Period;
594         index_ARR_step++;
595     }
596     else if(SIXSTEP_parameters.ARR_OK ==
597             0)
598     {
599         index_ARR_step = 1;
600         SIXSTEP_parameters.ACCEL>>=1;
601         if(SIXSTEP_parameters.ACCEL <
602             MINIMUM_ACC)
603         {
604             SIXSTEP_parameters.ACCEL =
605                 MINIMUM_ACC;
606         }
607         MC_StopMotor();
608         SIXSTEP_parameters.STATUS =
609             STARTUP_FAILURE;
610     }
611 }
612 }
613 else
614 {
615     if(SIXSTEP_parameters.VALIDATION_OK !=
616         TRUE)
617     {
618         SIXSTEP_parameters.STATUS = STARTUP;
619         MC_SixStep_Ramp_Motor_calc();
620         if(index_ARR_step <
```

```

        SIXSTEP_parameters.numberofitemArr)
620        {
621            LF_TIMx.Init.Period =
SIXSTEP_parameters.ARR_value;
622            LF_TIMx.Instance->ARR =
(uint32_t)LF_TIMx.Init.Period;
623            index_ARR_step++;
624        }
625        else if(SIXSTEP_parameters.ARR_OK==0)
626        {
627            index_ARR_step = 1;
628            SIXSTEP_parameters.ACCEL>>=1;
629            if(SIXSTEP_parameters.ACCEL <
MINIMUM_ACC)
630            {
631                SIXSTEP_parameters.ACCEL =
MINIMUM_ACC;
632            }
633            MC_StopMotor();
634            SIXSTEP_parameters.STATUS =
STARTUP_FAILURE;
635        }
636    }
637    else
638    {
639        SIXSTEP_parameters.ARR_OK = 1;
640        index_startup_motor = 1;
641        index_ARR_step = 1;
642    }
643 }
644 }
645 }
646
647 /**
648 * @}
649 */
650

```

```
651 /** @defgroup MC_SixStep_Alignment
MC_SixStep_Alignment
652 * @{
653 * @brief Generate the motor alignment
654 * @retval None
655 */
656
657 void MC_SixStep_Alignment()
658 {
659     SIXSTEP_parameters.step_position = 6;
660     LF_TIMx.Init.Period =
SIXSTEP_parameters.ARR_value;
661     LF_TIMx.Instance->ARR =
(uint32_t)LF_TIMx.Init.Period;
662     SIXSTEP_parameters.STATUS = ALIGNMENT;
663
MC_SixStep_Speed_Val_target_potentiometer();
664     index_align++;
665     if(index_align >= TIME_FOR_ALIGN+1)
666     {
667         SIXSTEP_parameters.ALIGN_OK = TRUE;
668         SIXSTEP_parameters.STATUS = STARTUP;
669         index_startup_motor = 1;
670         MC_SixStep_Ramp_Motor_calc();
671         LF_TIMx.Init.Prescaler =
SIXSTEP_parameters.prescaler_value;
672         LF_TIMx.Instance->PSC =
LF_TIMx.Init.Prescaler;
673         index_align = 0;
674     }
675 }
676
677 /**
678 * @}
679 */
680
681 /** @defgroup
```

```
    MC_SixStep_Speed_Val_target_potentiometer
    MC_SixStep_Speed_Val_target_potentiometer
682    * @{
683        * @brief Calculate the Motor Speed
        validation threshold according with the
        potentiometer value
684        * @retval None
685    */
686
687 void
    MC_SixStep_Speed_Val_target_potentiometer()
688 {
689     target_speed =
        SIXSTEP_parameters.ADC-Regular_Buffer[1] *
        MAX_POT_SPEED/ 4096;
690
691     if(target_speed < MIN_POT_SPEED)
692         target_speed = MIN_POT_SPEED;
693
694     if(target_speed >
        (MAX_POT_SPEED/VAL_POT_SPEED_DIV))
695         target_speed =
        (MAX_POT_SPEED/VAL_POT_SPEED_DIV);
696 }
697 /**
698 * @}
699 */
700
701 /** @defgroup MC_SixStep_Speed_Potentiometer
    MC_SixStep_Speed_Potentiometer
702 * @{
703     * @brief Calculate the potentiometer
        value to set the Motor Speed
704     * @retval None
705 */
706
707 void MC_SixStep_Speed_Potentiometer()
```

```

708 {
709     uint16_t i=0;
710     uint32_t sum = 0;
711     uint16_t mean = 0;
712     uint16_t max = 0;
713     for (i = 0; i < HFBUFFERSIZE; i++)
714     {
715         uint16_t val = HFBuffer[i];
716         sum += val;
717         if (val > max)
718         {
719             max = val;
720         }
721     }
722     sum -= max;
723     mean = sum / (HFBUFFERSIZE - 1);
724
725     SIXSTEP_parameters.Speed_Ref_filtered =
726         MC_Potentiometer_filter(mean);
727 }
728
729 /**
730 * @}
731 */
732
733 /** @defgroup MC_Set_PI_param
734  MC_Set_PI_param
735  * @{
736  * @brief Set all parameters for PI
737  * regulator
738  * @param PI_PARAM
739  * @retval None
740 */
741 void
742 MC_Set_PI_param(SIXSTEP_PI_PARAM_InitTypeDef_t

```

```

    *PI_PARAM)
741 { 
742     if(SIXSTEP_parameters.CW_CCW == 0)
743         PI_PARAM->Reference = target_speed;
744     else
745         PI_PARAM->Reference = -target_speed;
746
747     PI_PARAM->Kp_Gain = SIXSTEP_parameters.KP;
748     PI_PARAM->Ki_Gain = SIXSTEP_parameters.KI;
749
750     PI_PARAM->Lower_Limit_Output =
751         LOWER_OUT_LIMIT;
752     PI_PARAM->Upper_Limit_Output =
753         UPPER_OUT_LIMIT;
754 }
755 /**
756 * @}
757 */
758
759 /**
760 * @defgroup MC_PI_Controller
761 * @{
762 * @brief Compute the PI output for the
763 * Current Reference
764 * @param PI_PARAM PI parameters
765 * structure
766 * @param speed_fdb motor_speed_value
767 * @return int16_t Current reference
768 */
769 int16_t
    MC_PI_Controller(SIXSTEP_PI_PARAM_InitTypeDef_
    t *PI_PARAM, int16_t speed_fdb)
770 {

```

```
770     int32_t wProportional_Term=0,
    wIntegral_Term=0,
    wOutput_32=0,wIntegral_sum_temp=0;
771     int32_t Error =0;
772
773     Error = (PI_PARAM->Reference - speed_fdb);
774
775     /* Proportional term computation*/
776     wProportional_Term = PI_PARAM->Kp_Gain *
Error;
777
778     /* Integral term computation */
779     if (PI_PARAM->Ki_Gain == 0)
780     {
781         SIXSTEP_parameters.Integral_Term_sum =
0;
782     }
783     else
784     {
785         wIntegral_Term = PI_PARAM->Ki_Gain *
Error;
786         wIntegral_sum_temp =
SIXSTEP_parameters.Integral_Term_sum +
wIntegral_Term;
787         SIXSTEP_parameters.Integral_Term_sum =
wIntegral_sum_temp;
788     }
789
790     if(SIXSTEP_parameters.Integral_Term_sum>
KI_DIV * PI_PARAM->Upper_Limit_Output)
791         SIXSTEP_parameters.Integral_Term_sum =
KI_DIV* PI_PARAM->Upper_Limit_Output;
792
793     if(SIXSTEP_parameters.Integral_Term_sum<-
KI_DIV* PI_PARAM->Upper_Limit_Output)
794         SIXSTEP_parameters.Integral_Term_sum =
-KI_DIV* PI_PARAM->Upper_Limit_Output;
```

```
795
796 /* WARNING: the below instruction is not
   MISRA compliant, user should verify
797      that Cortex-M3 assembly
   instruction ASR (arithmetic shift right)
798      is used by the compiler to
   perform the shifts (instead of LSR
   logical shift right)*/
799
800
801     wOutput_32 = (wProportional_Term/KP_DIV) +
   (SIXSTEP_parameters.Integral_Term_sum/KI_DIV);
802
803     if(PI_PARAM->Reference>0)
804     {
805         if (wOutput_32 > PI_PARAM-
   >Upper_Limit_Output)
806         {
807             wOutput_32 = PI_PARAM-
   >Upper_Limit_Output;
808         }
809         else if (wOutput_32 < PI_PARAM-
   >Lower_Limit_Output)
810         {
811             wOutput_32 = PI_PARAM-
   >Lower_Limit_Output;
812         }
813     }
814     else
815     {
816         if (wOutput_32 < (- PI_PARAM-
   >Upper_Limit_Output) )
817         {
818             wOutput_32 = - (PI_PARAM-
   >Upper_Limit_Output);
819         }
820         else if (wOutput_32 > (-PI_PARAM-
   >Lower_Limit_Output))
```

```

821     {
822         wOutput_32 = (-PI_PARAM-
823 >Lower_Limit_Output);
824     }
825     return((int16_t)(wOutput_32));
826 }
827 /**
828 * @}
829 */
830
831
832 /**
833 * @defgroup MC_Task_Speed      MC_Task_Speed
834 * @{
835     * @brief Main task: Speed Loop with PI
836     * regulator
837     * @retval None
838 */
839 void MC_Task_Speed()
840 {
841     if(dac_status == TRUE)
842     {
843         SET_DAC_value(SIXSTEP_parameters.speed_fdbk_fi
844         ltered);
845     }
846     if((SIXSTEP_parameters.speed_fdbk_filtered
847 > (target_speed) ||
848 SIXSTEP_parameters.speed_fdbk_filtered < (-
849 target_speed)) &&
850 SIXSTEP_parameters.VALIDATION_OK !=TRUE)
851     {
852         SIXSTEP_parameters.STATUS = VALIDATION;
853         SIXSTEP_parameters.SPEED_VALIDATED =

```

```

        TRUE;
850    }
851
852    if(SIXSTEP_parameters.SPEED_VALIDATED ==
     TRUE && SIXSTEP_parameters.BEMF_OK == TRUE &&
     SIXSTEP_parameters.CL_READY != TRUE)
853    {
854        SIXSTEP_parameters.CL_READY = TRUE;
855    }
856
857    if(SIXSTEP_parameters.VALIDATION_OK ==
     TRUE)
858    {
859
     ****
     ****
860    SIXSTEP_parameters.STATUS = RUN;
861
     ****
     ****
862
863    if(PI_parameters.Reference>=0)
864    {
865        SIXSTEP_parameters.Current_Reference =
     (uint16_t)MC_PI_Controller(&PI_parameters,
     (int16_t)SIXSTEP_parameters.speed_fdbk_filtered
     );
866    }
867    else
868    {
869        SIXSTEP_parameters.Current_Reference =
     (uint16_t)(-MC_PI_Controller(&PI_parameters,
     (int16_t)SIXSTEP_parameters.speed_fdbk_filtered
     ));
870    }
871
872

```

```

        MC_SixStep_Current_Reference_Setvalue(SIXSTEP_
parameters.Current_Reference);
873    // SIXSTEP_parameters.pulse_value=SIXSTEP_parameters.Current_Reference;
874
875    }
876    MC_Bemf_Delay();
877 }
878 /**
879 * @}
880 */
881
882 /**
883 * @defgroup MC_Set_Speed      MC_Set_Speed
884 * @{
885 * @brief Set the new motor speed value
886 * @param speed_value: set new motor
887 * @retval None
888 */
889 void MC_Set_Speed(uint16_t speed_value)
890 {
891
892 #if (POTENTIOMETER == 1)
893     uint8_t change_target_speed = 0;
894     int16_t reference_tmp = 0;
895
896     if (SIXSTEP_parameters.Speed_Ref_filtered
897         > SIXSTEP_parameters.Speed_target_ramp)
898     {
899         if
900             ((SIXSTEP_parameters.Speed_Ref_filtered -
SIXSTEP_parameters.Speed_target_ramp) >
ADC_SPEED_TH)
901             {
902                 change_target_speed = 1;

```

```
901     }
902     else
903     {
904         /* Not change target speed because
905            less than threshold */
906     }
907     else
908     {
909         if
910             ((SIXSTEP_parameters.Speed_target_ramp -
911             SIXSTEP_parameters.Speed_Ref_filtered) >
912             ADC_SPEED_TH)
913             {
914                 change_target_speed = 1;
915             }
916             else
917             {
918                 /* Not change target speed because
919                    less than threshold */
920             }
921             if (change_target_speed == 1)
922             {
923                 SIXSTEP_parameters.Speed_target_ramp =
924                     SIXSTEP_parameters.Speed_Ref_filtered;
925                 if(SIXSTEP_parameters.CW_CCW == 0)
926                 {
927                     reference_tmp =
928                         SIXSTEP_parameters.Speed_Ref_filtered *
MAX_POT_SPEED / 4096;
929                     if(reference_tmp <= MIN_POT_SPEED)
930                     {
931                         PI_parameters.Reference =
MIN_POT_SPEED;
932                     }
933                 }
934             }
935         }
936     }
937 }
```

```

929         else
930     {
931         PI_parameters.Reference =
932             reference_tmp;
933     }
934 }
935 else
936 {
937     reference_tmp = -
938         (SIXSTEP_parameters.Speed_Ref_filtered *
939          MAX_POT_SPEED / 4096);
940     if(reference_tmp >= MIN_POT_SPEED)
941     {
942         PI_parameters.Reference = -
943             MIN_POT_SPEED;
944     }
945 }
946
947 }
948 #else
949     if(speed_value != 0)
950         PI_parameters.Reference = speed_value;
951 #endif
952
953 }
954
955 /**
956 * @}
957 */
958
959
960 /** @defgroup MC_Bemf_Delay      MC_Bemf_Delay

```

```

961     * @{
962         * @brief Take the delay time after each
963             new 6-step commutation
964         * @retval None
965     */
966 void MC_Bemf_Delay()
967 {
968     Bemf_delay_calc();
969 }
970 /**
971     * @}
972 */
973 /**
974     * @defgroup MC_StartMotor      MC_StartMotor
975     * @{
976         * @brief Start the Motor
977         * @retval None
978     */
979 void MC_StartMotor()
980 {
981     uwTick = 0;
982     SIXSTEP_parameters.STATUS = START;
983     HAL_TIM_Base_Start_IT(&LF_TIMx);
984     HAL_ADC_Start_IT(&ADCx);
985     SIXSTEP_parameters.RUN_Motor = 1;
986     BSP_X_NUCLEO_FAULT_LED_ON();
987     if(dac_status == TRUE)
988     {
989         START_DAC();
990     }
991 /**
992     * @}
993 */
994
995 /**
996     * @defgroup MC_StopMotor      MC_StopMotor
997     * @{

```

```

997     * @brief Stop the Motor
998     * @retval None
999 */
1000 void MC_StopMotor()
1001 {
1002     uwTick = 0;
1003     SIXSTEP_parameters.STATUS = STOP;
1004     SIXSTEP_parameters.RUN_Motor = 0;
1005     MC_SixStep_Stop_PWM_driving();
1006     HF_TIMx.Instance->CR1 &= ~(TIM_CR1_CEN);
1007     HF_TIMx.Instance->CNT = 0;
1008
1009     MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D();
1010     HAL_TIM_Base_Stop_IT(&LF_TIMx);
1011     HAL_ADC_Stop_IT(&ADCx);
1012     MC_SixStep_Current_Reference_Stop();
1013     BSP_X_NUCLEO_FAULT_LED_OFF();
1014     MC_SixStep_RESET();
1015 }
1016 /**
1017 * @}
1018 */
1019
1020 /**
1021 * @{
1022 * @brief Get the Eletrical Motor Speed
1023 *        from ARR value of LF TIM
1024 * @retval int32_t Return the electrical
1025 *        motor speed
1026 */
1027 int32_t MC_GetElSpeedHz()
1028 {

```

```

1029     uint16_t prsc = LF_TIMx.Instance->PSC;
1030     El_Speed_Hz = (int32_t)
1031         ((SIXSTEP_parameters.SYSCLK_frequency)/(prsc))
1032         /(__HAL_TIM_GetAutoreload(&LF_TIMx)*6);
1033     }
1034     else
1035         El_Speed_Hz = 0;
1036     if(PI_parameters.Reference<0)
1037         return (-El_Speed_Hz);
1038     else
1039         return (El_Speed_Hz);
1040     /**
1041      * @}
1042      */
1043 /**
1044  * @defgroup MC_GetMechSpeedRPM
1045  MC_GetMechSpeedRPM
1046  * @{
1047  * @brief Get the Mechanical Motor Speed
1048  * (RPM)
1049  * @return int32_t Return the mechanical
1050  * motor speed (RPM)
1051  */
1052 int32_t MC_GetMechSpeedRPM()
1053 {
1054     Mech_Speed_RPM = (int32_t)
1055         (MC_GetElSpeedHz() * 60 / Rotor_poles_pairs);
1056     return (Mech_Speed_RPM);
1057 }
1058 /**
1059  * @}
1060 /**
1061  * @defgroup MC_SixStep_Init_main_data

```

```
    MC_SixStep_Init_main_data
1060    * @{
1061        * @brief Init the main variables for
1062            motor driving from MC_SixStep_param.h
1063        * @retval None
1064    */
1065 void MC_SixStep_Init_main_data()
1066 {
1067     SIXSTEP_parameters.Ireference =
1068         STARTUP_CURRENT_REFERENCE;
1069     SIXSTEP_parameters.NUMPOLESPAIRS =
1070         NUM_POLE_PAIRS;
1071     SIXSTEP_parameters.ACCEL = ACC;
1072     SIXSTEP_parameters.KP = KP_GAIN;
1073     SIXSTEP_parameters.KI = KI_GAIN;
1074     SIXSTEP_parameters.CW_CCW = DIRECTION;
1075     SIXSTEP_parameters.Potentiometer =
1076         POTENTIOMETER;
1077 }
1078 /**
1079 * @}
1080 */
1081 /** @defgroup MC_SixStep_INIT
1082     MC_SixStep_INIT
1083     * @{
1084         * @brief Initialitation function for
1085             SixStep library
1086         * @retval None
1087     */
1088 void MC_SixStep_INIT()
1089 {
1090     MC_SixStep_Nucleo_Init();
```

```

1090     SIXSTEP_parameters.HF_TIMx_CCR =  

1091     HF_TIMx.Instance->HF_TIMx_CCR1;  

1092     SIXSTEP_parameters.HF_TIMx_ARR =  

1093     HF_TIMx.Instance->ARR;  

1094     SIXSTEP_parameters.HF_TIMx_PSC =  

1095     HF_TIMx.Instance->PSC;  

1096 //      MC_SixStep_Current_Reference_Start();  

1097  

1098     MC_SixStep_Current_Reference_Setvalue(SIXSTEP_  

1099     parameters.Ireference);  

1100  

1101 #ifdef UART_COMM  

1102     SIXSTEP_parameters.Button_ready = FALSE;  

1103     MC_UI_INIT(); /*!< Start  

1104     the UART Communication Task*/  

1105 #endif  

1106  

1107     MC_SixStep_Init_main_data();  

1108  

1109 #ifndef UART_COMM  

1110     SIXSTEP_parameters.Button_ready = TRUE;  

1111 #endif  

1112     MC_SixStep_RESET();  

1113 }  

1114  

1115  

1116  

1117 /** @defgroup MC_TIMx_SixStep_timebase  

1118     MC_TIMx_SixStep_timebase

```

```

1118 * @{
1119   * @brief Low Frequency Timer Callback -
Call the next step and request the filtered
speed value
1120   * @retval None
1121 */
1122
1123 void MC_TIMx_SixStep_timebase()
1124 {
1125   MC_SixStep_NEXT_step();
/*Change STEP number */
1126   if(SIXSTEP_parameters.ARR_OK == 0)
1127   {
1128     MC_SixStep_ARR_step();
/*BASE TIMER - ARR modification for STEP
frequency changing */
1129   }
1130
1131   MC_Speed_Filter();
/*Calculate SPEED filtered */
1132 }
1133
1134 /**
1135 * @}
1136 */
1137
1138 /** @defgroup MC_Speed_Filter
MC_Speed_Filter
1139 * @{
1140   * @brief Calculate the speed filtered
1141   * @retval None
1142 */
1143
1144 void MC_Speed_Filter()
1145 {
1146   if(array_completed == FALSE)
1147   {

```

```
1148     speed_tmp_array[index_array] =
    SIXSTEP_parameters.speed_fdbk;
1149     speed_sum_sp_filt = 0;
1150     for(uint16_t i = 1; i <=
    index_array;i++)
1151     {
1152         speed_sum_sp_filt = speed_sum_sp_filt
    + speed_tmp_array[i];
1153     }
1154     SIXSTEP_parameters.speed_fdbk_filtered
    = speed_sum_sp_filt/index_array;
1155     index_array++;
1156
1157     if(index_array >= FILTER_DEEP)
1158     {
1159         index_array = 1;
1160         array_completed = TRUE;
1161     }
1162 }
1163 else
1164 {
1165     index_array++;
1166     if(index_array >= FILTER_DEEP)
1167         index_array = 1;
1168
1169     speed_sum_sp_filt = 0;
1170     speed_tmp_array[index_array] =
    SIXSTEP_parameters.speed_fdbk;
1171     for(uint16_t i = 1; i <
    FILTER_DEEP;i++)
1172     {
1173         speed_sum_sp_filt =
    speed_sum_sp_filt + speed_tmp_array[i];
1174     }
1175     SIXSTEP_parameters.speed_fdbk_filtered
    = speed_sum_sp_filt/(FILTER_DEEP-1);
1176 }
```

```
1177 }
1178 /**
1179 * @}
1180 */
1182
1183 /** @defgroup MC_Potentiometer_filter
MC_Potentiometer_filter
1184 * @{
1185     * @brief Calculate the filtered
potentiometer value
1186     * @retval uint16_t Return the filtered
potentiometer value
1187 */
1188
1189 uint16_t MC_Potentiometer_filter(uint16_t
potentiometer_value)
1190 {
1191     if(buffer_completed == FALSE)
1192     {
1193         speed_tmp_buffer[index_pot_filt] =
potentiometer_value;
1194         speed_sum_pot_filt = 0;
1195         for(uint16_t i = 1; i <=
index_pot_filt;i++)
1196         {
1197             speed_sum_pot_filt =
speed_sum_pot_filt + speed_tmp_buffer[i];
1198         }
1199         potent_filtered =
speed_sum_pot_filt/index_pot_filt;
1200         index_pot_filt++;
1201
1202         if(index_pot_filt >= FILTER_DEEP)
1203         {
1204             index_pot_filt = 1;
1205             buffer_completed = TRUE;
```

```
1206         }
1207     }
1208     else
1209     {
1210         index_pot_filt++;
1211         if(index_pot_filt >= FILTER_DEEP)
1212         {
1213             index_pot_filt = 1;
1214         }
1215
1216         speed_sum_pot_filt = 0;
1217         speed_tmp_buffer[index_pot_filt] =
1218             potentiometer_value;
1219         uint16_t speed_max = 0;
1220         for(uint16_t i = 1; i <
1221             FILTER_DEEP;i++)
1222         {
1223             uint16_t val = speed_tmp_buffer[i];
1224             if (val > speed_max)
1225             {
1226                 speed_max = val;
1227             }
1228             speed_sum_pot_filt += val;
1229         }
1230         speed_sum_pot_filt -= speed_max;
1231         potent_filtered =
1232             speed_sum_pot_filt/(FILTER_DEEP-2);
1233     }
1234     if(potent_filtered==0) potent_filtered =
1235         1;
1236
1237     return(potent_filtered);
1238 }
```

```
1239
1240  /** @defgroup
MC_SysTick_SixStep_MediumFrequencyTask
MC_SysTick_SixStep_MediumFrequencyTask
1241  * @{
1242  * @brief Systick Callback - Call the
Speed loop
1243  * @retval None
1244  */
1245
1246 void
    MC_SysTick_SixStep_MediumFrequencyTask()
1247 {
1248  if(SIXSTEP_parameters.ALIGNMENT == TRUE &&
    SIXSTEP_parameters.ALIGN_OK == FALSE)
1249  {
1250      MC_SixStep_Alignment();
1251  }
1252
1253 #ifdef UART_COMM
1254  if(UART_FLAG_RECEIVE == TRUE)
    UART_Communication_Task();
1255 #endif
1256
1257 #ifdef DEMOMODE
1258  index_motor_run++;
1259  if(index_motor_run >= DEMO_START_TIME &&
    test_motor_run == 0)
1260  {
1261      MC_StopMotor();
1262      index_motor_run=0;
1263      test_motor_run=1;
1264  }
1265  if(index_motor_run >= DEMO_STOP_TIME &&
    test_motor_run == 1)
1266  {
1267      MC_StartMotor();
```

```

1268     test_motor_run = 0;
1269     index_motor_run=0;
1270 }
1271 #endif
1272
1273 if(SIXSTEP_parameters.VALIDATION_OK == TRUE
&& SIXSTEP_parameters.Potentiometer == TRUE)
1274 {
1275     MC_SixStep_Speed_Potentiometer();
1276 }
1277 /* Push button delay time to avoid double
   command */
1278 if(HAL_GetTick() == BUTTON_DELAY &&
   Enable_start_button != TRUE)
1279 {
1280     Enable_start_button = TRUE;
1281 }
1282
1283 /* SIXSTEP_parameters.Speed_Loop_Time x
   1msec */
1284 if(Tick_cnt >=
SIXSTEP_parameters.Speed_Loop_Time)
1285 {
1286     if(SIXSTEP_parameters.STATUS !=
SPEEDFBKERROR)
1287     {
1288         MC_Task_Speed();
1289     }
1290
SIXSTEP_parameters.MediumFrequencyTask_flag =
TRUE;
1291 if(SIXSTEP_parameters.VALIDATION_OK ==
TRUE)
1292 {
1293     MC_Set_Speed(0);
1294 }
1295 Tick_cnt=0;

```

```

1296     }
1297     else Tick_cnt++;
1298
1299     if(startup_bemf_failure == 1)
1300     {
1301         SIXSTEP_parameters.ACCEL>>=1;
1302         if(SIXSTEP_parameters.ACCEL <
1303             MINIMUM_ACC)
1304         {
1305             SIXSTEP_parameters.ACCEL =
1306             MINIMUM_ACC;
1307         }
1308         MC_StopMotor();
1309         cnt_bemf_event = 0;
1310         SIXSTEP_parameters.STATUS =
1311             STARTUP_BEMF_FAILURE;
1312     }
1313     if(speed_fdbk_error == 1)
1314     {
1315         MC_StopMotor();
1316         SIXSTEP_parameters.STATUS =
1317             SPEEDFBKERROR;
1318     }
1319 }
1320 */
1321
1322 /**
1323 * @}
1324 */
1325 /**
1326 * @defgroup MC_SixStep_ARR_Bemf
1327 MC_SixStep_ARR_Bemf
1328 * @{
1329 * @brief Calculate the new Autoreload
1330 value (ARR) for Low Frequency timer
1331 * @retval None
1332 */

```

```
1327
1328 void MC_SixStep_ARR_Bemf(uint8_t up_bemf)
1329 {
1330
1331 if(SIXSTEP_parameters.status_prev != SIXSTEP_parameters.step_position)
1332 {
1333 if(SIXSTEP_parameters.SPEED_VALIDATED == TRUE)
1334 {
1335 if(GPIO_ZERO_CROSS == 1)
1336 {
1337 HAL_GPIO_TogglePin(GPIO_PORT_ZCR,GPIO_CH_ZCR);

1338 }
1339 if(cnt_bemf_event> BEMF_CNT_EVENT_MAX)
1340 {
1341     startup_bemf_failure = 1;
1342 }
1343
1344 if(up_bemf == 1 &&
SIXSTEP_parameters.BEMF_OK !=TRUE)
1345 {
1346     n_zcr_startup++;
1347     cnt_bemf_event = 0;
1348 }
1349 else if(SIXSTEP_parameters.BEMF_OK
!=TRUE)
1350 {
1351     cnt_bemf_event++;
1352 }
1353
1354 if(n_zcr_startup>= NUMBER_ZCR &&
SIXSTEP_parameters.BEMF_OK !=TRUE )
1355 {
1356     SIXSTEP_parameters.BEMF_OK = TRUE;
```

```

1357         n_zcr_startup = 0;
1358     }
1359 }
1360 SIXSTEP_parameters.status_prev =
1361     SIXSTEP_parameters.step_position;
1362     if(SIXSTEP_parameters.VALIDATION_OK ==
1363     1)
1364     {
1365         counter_ARR_Bemf =
1366         __HAL_TIM_GetCounter(&LF_TIMx);
1367         __HAL_TIM_SetAutoreload(&LF_TIMx,
1368         (counter_ARR_Bemf+ARR_LF/2));
1369     }
1370 }
1371 /**
1372 * @}
1373 */
1374
1375 /** @defgroup MC_ADCx_SixStep_Bemf
1376   MC_ADCx_SixStep_Bemf
1377   *
1378   * @brief Compute the zero crossing
1379   * detection
1380   * @retval None
1381 */
1382 void MC_ADCx_SixStep_Bemf()
1383 {
1384     if(__HAL_TIM_DIRECTION_STATUS(&HF_TIMx))
1385     {
1386         HAL_GPIO_WritePin(GPIO_PORT_COMM, GPIO_CH_COMM,

```

```

        GPIO_PIN_SET);
1387     /* UP-counting direction started */
1388     /* GET the ADC value (PHASE CURRENT)*/
1389     if(SIXSTEP_parameters.STATUS != START &&
1390       SIXSTEP_parameters.STATUS != ALIGNMENT)
1390     {
1391         switch
1392         (
1393             SIXSTEP_parameters.step_position)
1392         {
1393             case 6:
1394             {
1395                 if(SIXSTEP_parameters.demagn_counter
1396                   >= SIXSTEP_parameters.demagn_value)
1396                 {
1397                     SIXSTEP_parameters.ADC_BUFFER[1] =
1397                     HAL_ADC_GetValue(&ADCx);
1398                     if(PI_parameters.Reference>=0)
1399                     {
1400
1400                     if(SIXSTEP_parameters.ADC_BUFFER[1]>
1401                       SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1401                     {
1402                         MC_SixStep_ARR_Bemf(1);
1403
1403                     SIXSTEP_parameters.BEMF_Tdown_count = 0;
1404                     }
1405                     }
1406                     else
1407                     {
1408                         if(SIXSTEP_parameters.ADC_BUFFER[1]
1409                           < SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1409                         {
1410                             MC_SixStep_ARR_Bemf(0);
1411                         }
1412                         }
1413                     }
1414                     else

```

```
    SIXSTEP_parameters.demagn_counter++;

1415    }
1416    break;
1417    case 3:
1418    {
1419        if(SIXSTEP_parameters.demagn_counter
1420            >= SIXSTEP_parameters.demagn_value)
1421        {
1422            SIXSTEP_parameters.ADC_BUFFER[1] =
1423                HAL_ADC_GetValue(&ADCx);
1424            if(PI_parameters.Reference>=0)
1425            {
1426                if(SIXSTEP_parameters.ADC_BUFFER[1]<
1427                    SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1428                {
1429                    MC_SixStep_ARR_Bemf(0);
1430                }
1431            }
1432            if(SIXSTEP_parameters.ADC_BUFFER[1]>
1433                SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1434            {
1435                MC_SixStep_ARR_Bemf(1);
1436            }
1437        }
1438        else
1439        {
1440            SIXSTEP_parameters.demagn_counter++;
1441        }
1442        break;
1443        case 5:
1444        {
```

```

        >= SIXSTEP_parameters.demagn_value)
1444     {
1445         SIXSTEP_parameters.ADC_BUFFER[2] =
1446             HAL_ADC_GetValue(&ADCx);
1447             if(PI_parameters.Reference>=0)
1448             {
1449                 if(SIXSTEP_parameters.ADC_BUFFER[2]<
1450                     SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1451                     {
1452                         MC_SixStep_ARR_Bemf(0);
1453                     }
1454             }
1455             if(SIXSTEP_parameters.ADC_BUFFER[2]>
1456                 SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1457                 {
1458                     MC_SixStep_ARR_Bemf(1);
1459                 }
1460             }
1461         }
1462     else
1463         SIXSTEP_parameters.demagn_counter++;
1464     break;
1465 case 2:
1466 {
1467     if(SIXSTEP_parameters.demagn_counter
1468         >= SIXSTEP_parameters.demagn_value)
1469         {
1470             SIXSTEP_parameters.ADC_BUFFER[2] =
1471                 HAL_ADC_GetValue(&ADCx);
1472                 if(PI_parameters.Reference>=0)
1473                 {
1474                     if(SIXSTEP_parameters.ADC_BUFFER[2]>

```

```

        SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1473    {
1474        MC_SixStep_ARR_Bemf(1);
1475
1476    }
1477}
1478else
1479{
1480    if(SIXSTEP_parameters.ADC_BUFFER[2]<
SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1481    {
1482        MC_SixStep_ARR_Bemf(0);
1483    }
1484}
1485}
1486else
1487    SIXSTEP_parameters.demagn_counter++;
1488}
1489break;
1490case 4:
1491{
1492    if(SIXSTEP_parameters.demagn_counter
1493    >= SIXSTEP_parameters.demagn_value)
1494    {
1495        SIXSTEP_parameters.ADC_BUFFER[3] =
1496        HAL_ADC_GetValue(&ADCx);
1497        if(PI_parameters.Reference>=0)
1498        {
1499            if(SIXSTEP_parameters.ADC_BUFFER[3]>
SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1500            {
1501                MC_SixStep_ARR_Bemf(1);
1502
1503            SIXSTEP_parameters.BEMF_Tdown_count = 0;
1504        }
1505    }

```

```

1502     else
1503     {
1504         if(SIXSTEP_parameters.ADC_BUFFER[3]<
1505             SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1506         {
1507             MC_SixStep_ARR_Bemf(0);
1508         }
1509     }
1510     else
1511     {
1512         break;
1513     case 1:
1514     {
1515         if(SIXSTEP_parameters.demagn_counter
1516             >= SIXSTEP_parameters.demagn_value)
1517         {
1518             SIXSTEP_parameters.ADC_BUFFER[3] =
1519                 HAL_ADC_GetValue(&ADCx);
1520             if(PI_parameters.Reference>=0)
1521             {
1522                 if(SIXSTEP_parameters.ADC_BUFFER[3]<
1523                     SIXSTEP_parameters.ADC_BEMF_threshold_DOWN)
1524                 {
1525                     MC_SixStep_ARR_Bemf(0);
1526                 }
1527                 else
1528                 {
1529                     if(SIXSTEP_parameters.ADC_BUFFER[3]>
1530                         SIXSTEP_parameters.ADC_BEMF_threshold_UP)
1531                     {
1532                         MC_SixStep_ARR_Bemf(1);
1533                         SIXSTEP_parameters.BEMF_Tdown_count = 0;
1534                     }

```

```

1532         }
1533     }
1534     else
1535         SIXSTEP_parameters.demagn_counter++;
1536     }
1537     break;
1538   }
1539   /***** SET ADC CHANNEL FOR
1540   SPEED/CURRENT/VBUS *****/
1540   /* Set the channel for next ADC Regular
1541   reading */
1541
1541   MC_SixStep_ADC_Channel(SIXSTEP_parameters.ADC_
1542   SEQ_CHANNEL[index_adc_chn]);
1542
1542   *****
1542   *****
1543
1543   HAL_GPIO_WritePin(GPIO_PORT_COMM, GPIO_CH_COMM,
1544   GPIO_PIN_RESET);
1544
1545   else
1546   {
1547     /* Down-counting direction started */
1548     /* Set the channel for next ADC Regular
1549     reading */
1550
1550     SIXSTEP_parameters.ADC-Regular_Buffer[index_ad
1551     c_chn] = HAL_ADC_GetValue(&ADCx);
1552
1552     if (index_adc_chn == 1)
1553     {
1554       HBuffer[HBufferIndex++] =
1554       HAL_ADC_GetValue(&ADCx);
1555       if (HBufferIndex >= HBUFFERSIZE)

```

```

1556     {
1557         HFBufferIndex = 0;
1558     }
1559 }
1560 index_adc_chn++;
1561 if(index_adc_chn>3) index_adc_chn = 0;
1562
    MC_SixStep_ADC_Channel(SIXSTEP_parameters.CurrentRegular_BEMF_ch);
1563 }
1564 }
1565 }
1566
1567 /**
1568 * @}
1569 */
1570
1571 /** @defgroup MC_EXT_button_SixStep
    MC_EXT_button_SixStep
1572 * @{
1573 * @brief GPIO EXT Callback - Start or
    Stop the motor through the Blue push button on
    STM32Nucleo
        * @retval None
1574 */
1575
1576
1577 void MC_EXT_button_SixStep()
1578 {
1579     if(Enable_start_button == TRUE)
1580     {
1581         if(SIXSTEP_parameters.RUN_Motor == 0 &&
            SIXSTEP_parameters.Button_ready == TRUE)
1582         {
1583             MC_StartMotor();
1584             Enable_start_button = FALSE;
1585         }
1586     else

```

```
1587     {
1588         MC_StopMotor();
1589         Enable_start_button = FALSE;
1590     }
1591 }
1592 }
1593 /**
1594 * @}
1595 */
1596 /**
1597 /**
1598 * @brief This function is called to
1599 increment a global variable "uwTick"
1600 * used as application time base.
1601 * @note In the default implementation,
1602 this variable is incremented each 1ms
1603 * in Systick ISR.
1604 * @note This function is declared as
1605 __weak to be overwritten in case of other
1606 * implementations in user file.
1607 * @retval None
1608 */
1609 void HAL_IncTick(void)
1610 {
1611     uwTick++;
1612 }
1613 /**
1614 * @brief Provides a tick value in
1615 millisecond.
1616 * @note The function is declared as
1617 __weak to be overwritten in case of other
1618 * implementations in user file.
1619 * @retval tick value
1620 */
1621 uint32_t HAL_GetTick(void)
```

```
1619 {  
1620     return uwTick;  
1621 }  
1622  
1623  
1624 /**  
 * @}    end MC_6-STEP_LIB  
 */  
1627  
1628 /**  
 * @}    end MIDDLEWARES  
 */  
1630
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Inc

stm32F401_nucleo_ihm11m1.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      stm32F401_nucleo_ihm11m1.h
4  * @author    IPC
5  * @version   V0
6  * @date      10/08/2016
7  * @brief     This file provides the interface
8  *             between the MC-lib and STM Nucleo
9  *
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
12 * STMicroelectronics</center></h2>
13 *
14 * Redistribution and use in source and
15 * binary forms, with or without modification,
16 * are permitted provided that the following
17 * conditions are met:
18 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34    *
35 ****
36 */
37
38 /* Define to prevent recursive inclusion ---
39 #ifndef __STM32F401_NUCLEO_IHM11M1_H
40 #define __STM32F401_NUCLEO_IHM11M1_H
41
42 #include "stm32f4xx_hal.h"
43 #include "main_F401.h"
44
45 #define HF_TIMx          htim1
46 #define LF_TIMx          htim4
47 #define HALL_ENCODER_TIMx htim2
48 #define ADCx             hadc1
49 #define DACx             htim3
50 #define UART            huart2
51
52 #define ADC_CH_1
      ADC_CHANNEL_0 /*CURRENT*/
53 #define ADC_CH_2
      ADC_CHANNEL_12 /*SPEED*/
54 #define ADC_CH_3
      ADC_CHANNEL_1 /*VBUS*/
55 #define ADC_CH_4
      ADC_CHANNEL_2 /*TEMP */
56 #define ADC_Bemf_CH1
      ADC_CHANNEL_13 /*BEMF1*/
57 #define ADC_Bemf_CH2
      ADC_CHANNEL_8 /*BEMF2*/
58 #define ADC_Bemf_CH3
      ADC_CHANNEL_7 /*BEMF3*/
59
60 #define ADC_CH_1_ST
```

```
    ADC_SAMPLETIME_3CYCLES /*CURRENT sampling
time */
61| #define ADC_CH_2_ST
ADC_SAMPLETIME_84CYCLES /*SPEED sampling
time*/
62| #define ADC_CH_3_ST
ADC_SAMPLETIME_84CYCLES /*VBUSS sampling
time*/
63| #define ADC_CH_4_ST
ADC_SAMPLETIME_84CYCLES /*TEMP sampling
time*/
64| #define ADC_Bemf_CH1_ST
ADC_SAMPLETIME_28CYCLES /*BEMF1 sampling
time*/
65| #define ADC_Bemf_CH2_ST
ADC_SAMPLETIME_28CYCLES /*BEMF2 sampling
time*/
66| #define ADC_Bemf_CH3_ST
ADC_SAMPLETIME_28CYCLES /*BEMF3 sampling
time*/
67|
68| #define HF_TIMx_CH1
TIM_CHANNEL_1
69| #define HF_TIMx_CH2
TIM_CHANNEL_2
70| #define HF_TIMx_CH3
TIM_CHANNEL_3
71| #define HF_TIMx_CCR1 CCR1
/*Channel 1*/
72| #define HF_TIMx_CCR2 CCR2
/*Channel 2*/
73| #define HF_TIMx_CCR3 CCR3
/*Channel 3*/
74|
75| #define DAC_ENABLE 0
/*!< Enable (1) the DAC peripheral */
76|
```

```

77 | #define GPIO_PORT_ZCR           GPIOC
    /*!< GPIO port name for zero crossing
       detection */
78 | #define GPIO_CH_ZCR            GPIO_PIN_12
    /*!< GPIO pin name for zero crossing
       detection */
79 | #define GPIO_PORT_COMM          GPIOC
    /*!< GPIO port name for 6Step commutation */
80 | #define GPIO_CH_COMM            GPIO_PIN_10
    /*!< GPIO pin name for 6Step commutation */
81 |
82 | #define STARTM_CMD              0      /*!<
   Start Motor command received */
83 | #define STOPMT_CMD               1      /*!<
   Stop Motor command received */
84 | #define SETSPD_CMD               2      /*!<
   Set the new speed value command received */
85 | #define GETSPD_CMD               3      /*!<
   Get Mechanical Motor Speed command received */
86 | #define INIREF_CMD               4      /*!<
   Set the new STARUP_CURRENT_REFERENCE value
   command received */
87 | #define POLESP_CMD               5      /*!<
   Set the Pole Pairs value command received */
88 | #define ACCELE_CMD               6      /*!<
   Set the Accelleration for Start-up of the
   motor command received */
89 | #define KP_PRM_CMD               7      /*!<
   Set the KP PI param command received */
90 | #define KI_PRM_CMD               8      /*!<
   Set the KI PI param command received */
91 | #define POTENZ_CMD               9      /*!<
   Enable Potentiometer command received */
92 | #define HELP_CMD                 10     /*!<
   Help command received */
93 | #define STATUS_CMD                11     /*!<
   Get the Status of the system command received

```

```
* /  
94 |     #define DIRECT_CMD           12      /* !<  
   |     Get the motor direction */  
95 |  
96 |     /** @addtogroup stm32F401_nucleo_ihm11m1  
   |     stm32F401_nucleo_ihm11m1  
97 |     * @brief Interface file for STM32F401 and  
   |     Library configuration  
98 |     * @{  
99 |     */  
100|  
101|     /** @defgroup Exported_function_F401  
   |     Exported_function_F401  
102|     * @{  
103|     */  
104|     /**  
105|     * @brief API function for STM32  
   |     instruction  
106|     */  
107|     void MC_SixStep_ADC_Channel(uint32_t);  
108|     void MC_SixStep_Nucleo_Init(void);  
109|     void START_Ref_Generation(void);  
110|     void STOP_Ref_Generation(void);  
111|     void Set_Ref_Generation(uint16_t);  
112|     void START_DAC(void);  
113|     void STOP_DAC(void);  
114|     void SET_DAC_value(uint16_t);  
115|     void Bemf_delay_calc(void);  
116|     uint32_t Get_UART_Data(void);  
117|     void  
   |     MC_SixStep_EnableInput_CH1_E_CH2_E_CH3_D(void);  
  
118|     void  
   |     MC_SixStep_EnableInput_CH1_E_CH2_D_CH3_E(void);  
  
119|     void  
   |     MC_SixStep_EnableInput_CH1_D_CH2_E_CH3_E(void);
```

```
120 |     void
|     MC_SixStep_DisableInput_CH1_D_CH2_D_CH3_D(void
| );
121 |     void MC_SixStep_Start_PWM_driving(void);
122 |     void MC_SixStep_Stop_PWM_driving(void);
123 |     void
|     MC_SixStep_HF_TIMx_SetDutyCycle_CH1(uint16_t);
124 |     void
|     MC_SixStep_HF_TIMx_SetDutyCycle_CH2(uint16_t);
125 |     void
|     MC_SixStep_HF_TIMx_SetDutyCycle_CH3(uint16_t);
126 |     void
|     MC_SixStep_Current_Reference_Start(void);
127 |     void
|     MC_SixStep_Current_Reference_Stop(void);
128 |     void
|     MC_SixStep_Current_Reference_Setvalue(uint16_t
| );
129 |     void BSP_X_NUCLEO_FAULT_LED_ON(void);
130 |     void BSP_X_NUCLEO_FAULT_LED_OFF(void);
131 |     /**
132 |     * @}
133 |     */
134 |
135 |     /**
136 |     * @}
137 |     */
138 | #endif
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Inc

6Step_Lib.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      6Step_Lib.h
4  * @author    System lab
5  * @version   V1.0.0
6  * @date     06-July-2015
7  * @brief    This header file provides the
8  *           set of functions for Motor Control
9  *           library
10 *
11 *
12 * <h2><center>&copy; COPYRIGHT(c) 2015
13 * STMicroelectronics</center></h2>
14 *
15 * Redistribution and use in source and
 * binary forms, with or without modification,
 * are permitted provided that the following
 * conditions are met:
```

```
16 *      1. Redistributions of source code must
17 *         retain the above copyright notice,
18 *         this list of conditions and the
19 *         following disclaimer.
20 *      2. Redistributions in binary form must
21 *         reproduce the above copyright notice,
22 *         this list of conditions and the
23 *         following disclaimer in the documentation
24 *         and/or other materials provided with
25 *         the distribution.
26 *      3. Neither the name of
27 *         STMicroelectronics nor the names of its
28 *         contributors
29 *         may be used to endorse or promote
30 *         products derived from this software
31 *         without specific prior written
32 *         permission.
33 *
34 * THIS SOFTWARE IS PROVIDED BY THE
35 * COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
36 * AND ANY EXPRESS OR IMPLIED WARRANTIES,
37 * INCLUDING, BUT NOT LIMITED TO, THE
38 * IMPLIED WARRANTIES OF MERCHANTABILITY AND
39 * FITNESS FOR A PARTICULAR PURPOSE ARE
40 * DISCLAIMED. IN NO EVENT SHALL THE
41 * COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
42 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
43 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
44 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
45 * PROCUREMENT OF SUBSTITUTE GOODS OR
46 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
47 * OR BUSINESS INTERRUPTION) HOWEVER
48 * CAUSED AND ON ANY THEORY OF LIABILITY,
49 * WHETHER IN CONTRACT, STRICT LIABILITY,
50 * OR TORT (INCLUDING NEGLIGENCE OR
51 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
52 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

POSSIBILITY OF SUCH DAMAGE.

```
35 *  
36 ****  
37 */  
38  
39  
40 /* Define to prevent recursive inclusion ---  
-----*/  
41 #ifndef __6STEP_LIB_H  
42 #define __6STEP_LIB_H  
43  
44  
45 #include "stm32_nucleo_ihm11m1.h"  
46  
47 #include "math.h"  
48 #include "stdlib.h"  
49 #include "stdio.h"  
50  
51 /** @addtogroup MIDDLEWARES      MIDDLEWARES  
52 * @brief Middlewares Layer  
53 * @{  
54 *}  
55  
56  
57 /** @addtogroup MC_6-STEP_LIB      MC_6-  
STEP_LIB  
58 * @brief Motor Control driver  
59 * @{  
60 *}  
61  
62 /** @defgroup Exported_types  Exported_types  
63 * @{  
64 *}  
65 /**  
66 * @brief Six Step parameters
```

```
67 */  
68 typedef enum  
69 {  
70     IDLE, /* 0  
71 */  
71     STARTUP, /* 1  
72 */  
72     VALIDATION, /* 2  
73 */  
73     STOP, /* 3  
74 */  
74     START, /* 4  
75 */  
75     RUN, /* 5  
76 */  
76     ALIGNMENT, /* 6  
77 */  
77     SPEEDFBKERROR, /* 7  
78 */  
78     OVERCURRENT, /* 8  
79 */  
79     STARTUP_FAILURE, /* 9  
80     STARTUP_BEMF_FAILURE /* 10 */  
81 } SIXSTEP_Base_SystStatus_t;  
82  
83 /**  
84 * @}  
85 */  
86  
87  
88 /** @defgroup Exported_types Exported_types  
89 * @{  
90 */  
91 /**  
92 * @brief Six Step parameters  
93 */
```

```

94 |     typedef struct
95 |     {
96 |         uint32_t LF_TIMx_PSC;           /* !
97 |         < Prescaler variable for low frequency timer */
98 |         uint32_t LF_TIMx_ARR;          /* !
99 |         < ARR variable for low frequency timer */
100|         uint32_t HF_TIMx_PSC;          /* !
101|         < Prescaler variable for high frequency
102|             timer */
103|         uint32_t HF_TIMx_ARR;          /* !
104|         < ARR variable for high frequency timer */
105|         uint32_t HF_TIMx_CCR;          /* !
106|         < CCR variable for high frequency timer */
107|         uint8_t step_position;          /* !
108|         < Step number variable for SixStep algorithm */
109|         SIXSTEP_Base_SystStatus_t STATUS; /* !
110|         < Status variable for SixStep algorithm */
111|         uint8_t status_prev;           /* !
112|         < Previous status variable for SixStep
113|             algorithm */
114|         uint16_t pulse_value;           /* !
115|         < CCR value for SixStep algorithm */
116|         uint16_t ARR_value;            /* !
117|         < ARR vector for Accell compute */
118|         uint32_t Regular_channel[4];   /* !
119|         < Buffer for ADC regular channel */
120|         uint32_t CurrentRegular_BEMF_ch; /* !
121|         < ADC regular channel to select */
122|         uint32_t prescaler_value;       /* !
123|         < Prescaler value for low freq timer */
124|         uint16_t numberofitemArr;      /* !
125|         < Number of elements */
126|         uint32_t ADC_BUFFER[4];        /* !
127|         < Buffer for ADC regular channel */
128|         uint32_t ADC_SEQ_CHANNEL[4];   /* !
129|         < Buffer for ADC regular channel */
130|         uint32_t ADC_Regular_Buffer[5]; /* !

```

```

    < Buffer for ADC regular channel */
113|   uint16_t ADC_BEMF_threshold_UP;           /*!
    < Voltage threshold for BEMF detection in up
    direction*/
114|   uint16_t ADC_BEMF_threshold_DOWN;          /*!
    < Voltage threshold for BEMF detection in down
    direction*/
115|   uint16_t demagn_counter;                   /*!
    < Demagnetization counter*/
116|   uint16_t demagn_value;                    /*!
    < Demagnetization value*/
117|   int16_t speed_fdbk;                      /*!
    < Motor speed variable*/
118|   int16_t speed_fdbk_filtered;             /*!
    < Filtered Motor speed variable*/
119|   int16_t filter_depth;                    /*!
    < Filter depth for speed measuring*/
120|   uint16_t Current_Reference;              /*!
    < Current reference for SixStep algorithm*/
121|   uint16_t Ireference;                     /*!
    < Current reference for SixStep algorithm*/
122|   int32_t Integral_Term_sum;              /*!
    < Global Integral part for PI*/
123|   uint8_t CMD;                           /*!
    < Flag control for Motor Start/Stop*/
124|   uint8_t ALIGN_OK;                      /*!
    < Flag control for Motor Alignment*/
125|   uint8_t ALIGNMENT;                     /*!
    < Flag control for Motor Alignment ongoing*/
126|   uint8_t bemf_state_1;                  /*!
    < Bemf variable */
127|   uint8_t bemf_state_2;                  /*!
    < Bemf variable */
128|   uint8_t bemf_state_3;                  /*!
    < Bemf variable */
129|   uint8_t bemf_state_4;                  /*!
    < Bemf variable */

```

```

130 |     uint8_t bemf_state_5;          /* !
   |     < Bemf variable */
131 |     uint8_t bemf_state_6;          /* !
   |     < Bemf variable */
132 |     uint16_t Speed_Loop_Time;      /* !
   |     < Speed loop variable for timing */
133 |     uint16_t Speed_Ref_filtered;    /* !
   |     < Filtered Reference Motor Speed variable */
134 |     uint16_t RUN_Motor;            /* !
   |     < Flag for Motor status */
135 |     uint8_t ARR_OK;               /* !
   |     < ARR flag control for Accell status */
136 |     uint8_t VALIDATION_OK;        /* !
   |     < Validation flag for Closed loop control
begin */

137 |     uint8_t SPEED_VALIDATED;       /* !
   |     < Validation flag for Speed before closed loop
control */
138 |     uint16_t Speed_target_ramp;     /* !
   |     < Target Motor Speed */
139 |     uint16_t Speed_target_time;    /* !
   |     < Target Motor Ramp time */
140 |     uint16_t Ramp_Start;          /* !
   |     < Ramp time start */
141 |     uint16_t Bemf_delay_start;    /* !
   |     < Bemf variable */
142 |     uint16_t MediumFrequencyTask_flag; /* !
   |     < Flag for Medium Task Frequency */
143 |     uint32_t SYSCLK_frequency;    /* !
   |     < System clock main frequency */
144 |     uint32_t Uart_cmd_to_set;     /* !
   |     < */
145 |     uint32_t Uart_value_to_set;    /* !
   |     < */
146 |     uint8_t Button_ready;         /* !
   |     < */
147 |     uint8_t BEMF_OK;              /* !

```

```

    <   */
148 |     uint8_t CL_READY;                      /* !
    < */
149 |     uint8_t BEMF_Tdown_count;                /* !
< BEMF Consecutive Threshold Falling Crossings
Counter */
150 |     uint16_t IREFERENCE;                    /* !
< Currrent reference*/
151 |     uint16_t NUMPOLESPAIRS;                  /* !
< Number of motor pole pairs */
152 |     uint32_t ACCEL;                        /* !
< Acceleration start-up parameter*/
153 |     uint16_t KP;                           /* !
< KP parameter for PI regulator */
154 |     uint16_t KI;                           /* !
< KI parameter for PI regulator */
155 |     uint8_t CW_CCW;                       /* !
< Set the motor direction */
156 |     uint8_t Potentiometer;                 /* !
< Enable/Disable potentiometer for speed
control */
157 | } SIXSTEP_Base_InitTypeDef;                /* !
< Six Step Data Structure */
158 |
159 /**
160 * @}
161 */
162
163 /** @defgroup Exported_types Exported_types
164 * @{
165 */
166 /**
167 * @brief Six PI regulator parameters
168 */
169
170 typedef struct
171 {

```

```

172     int16_t Reference; /* !<
   Reference value for PI regulator */
173     int16_t Kp_Gain; /* !<
   Kp value for PI regulator */
174     int16_t Ki_Gain; /* !<
   Ki value for PI regulator */
175     int16_t Lower_Limit_Output; /* !<
   Min output value for PI regulator */
176     int16_t Upper_Limit_Output; /* !<
   Max output value for PI regulator */
177     int8_t Max_PID_Output; /* !<
   Max Saturation indicator flag */
178     int8_t Min_PID_Output; /* !<
   Min Saturation indicator flag */
179 } SIXSTEP_PI_PARAM_InitTypeDef_t,
 *SIXSTEP_pi_PARAM_InitTypeDef_t; /* !< PI Data
 Structure */

180 /**
181  * @}
182  */
183
184
185 /** @defgroup Exported_function_6StepLib
   Exported_function_6StepLib
186 * @{
187 */
188
189 void MC_SixStep_INIT(void);
190 void MC_SixStep_RESET(void);
191 void MC_StartMotor(void);
192 void MC_StopMotor(void);
193 void MC_Set_Speed(uint16_t);
194 void MC_EXT_button_SixStep(void);
195
196 /**
197  * @}
198 */

```

```
199
200  /** MC_6-STEP_LIB
201  * @}
202  */
203
204
205  /** MIDDLEWARES
206  * @}
207  */
208
209 #endif
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Inc

MC_SixStep_param.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      MC_SixStep_param.h
4  * @author
5  * @version   V0
6  * @date      10/07/2016
7  * @brief     This header file provides all
8  *             parameters to driver a motor with 6Step
9  *             library
10 *
11 *
12 * <h2><center>&copy; COPYRIGHT(c) 2015
13 * STMicroelectronics</center></h2>
14 *
15 * Redistribution and use in source and
 * binary forms, with or without modification,
 * are permitted provided that the following
 * conditions are met:
```

```
16 *      1. Redistributions of source code must
17 *         retain the above copyright notice,
18 *         this list of conditions and the
19 *         following disclaimer.
20 *      2. Redistributions in binary form must
21 *         reproduce the above copyright notice,
22 *         this list of conditions and the
23 *         following disclaimer in the documentation
24 *         and/or other materials provided with
25 *         the distribution.
26 *      3. Neither the name of
27 *         STMicroelectronics nor the names of its
28 *         contributors
29 *            may be used to endorse or promote
30 *            products derived from this software
31 *            without specific prior written
32 *            permission.
33 *
34 * THIS SOFTWARE IS PROVIDED BY THE
35 * COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
36 * AND ANY EXPRESS OR IMPLIED WARRANTIES,
37 * INCLUDING, BUT NOT LIMITED TO, THE
38 * IMPLIED WARRANTIES OF MERCHANTABILITY AND
39 * FITNESS FOR A PARTICULAR PURPOSE ARE
40 * DISCLAIMED. IN NO EVENT SHALL THE
41 * COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
42 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
43 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
44 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
45 * PROCUREMENT OF SUBSTITUTE GOODS OR
46 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
47 * OR BUSINESS INTERRUPTION) HOWEVER
48 * CAUSED AND ON ANY THEORY OF LIABILITY,
49 * WHETHER IN CONTRACT, STRICT LIABILITY,
50 * OR TORT (INCLUDING NEGLIGENCE OR
51 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
52 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

```
POSSIBILITY OF SUCH DAMAGE.

35    *
36
***** ****
***** ****
37    */
38
39 /** @addtogroup MIDDLEWARES      MIDDLEWARES
40   * @brief Middlewares Layer
41   * @{
42 */
43
44 /** @addtogroup MC_6-STEP_LIB      MC_6-
STEP_LIB
45   * @brief Motor Control driver
46   * @{
47 */
48
49 /** @defgroup Main_Motor_parameters
Main_Motor_parameters
50   * @{
51   * @brief All motor parameters for 6Step
driving
52 */
53
54 /*
***** ****
***** ****
55
=====
=====
56          ##### BASIC
PARAMETERS #####
57
=====
=====
58 ****
```

```

***** ****
59 | #define NUM_POLE_PAIRS
  2   /*!< Number of Motor Pole pairs */
60 | #define DIRECTION
  0   /*!< Set motor direction CW = 0 and CCW
= 1*/
61 | #define TARGET_SPEED
  4000  /*!< Target speed in closed loop
control */
62 | #define POTENTIOMETER
  1   /*!< Enable (1)/Disable (0) the
potentiometer */
63 |
64 |
65 | /*
***** ****
***** ****
66 |
===== =====
===== =====
67 | ##### ADVANCED
PARAMETERS #####
68 |
===== =====
===== =====
69 | ****
***** ****
***** ****
***** */
70 | /*!< **** Open loop control
***** */
71 | #define STARTUP_DUTY_CYCLE
  600  /*!< StartUP Duty Cycle*/
72 | #define STARTUP_CURRENT_REFERENCE
  STARTUP_DUTY_CYCLE
73 | #define ACC
  10000000  /*!< Mechanical acceleration rate
(setting available in manual mode, LOAD_TYPE =
0) */

```

```
74 | #define MINIMUM_ACC
    1000      /*!< Mechanical acceleration rate
for BIG load application */
75 | #define NUMBER_OF_STEPS
    20000     /*!< Number of elements for motor
start-UP (max value 65535)*/
76 | #define TIME_FOR_ALIGN
    500       /*!< Time for alignment (msec)*/
77 | #define BUTTON_DELAY
    1000      /*!< Delay time to enable push
button for new command (1 = 1msec)*/
78 | #define NUMBER_ZCR
    12        /*!< Number of zero crossing event
during the startup for closed loop control
begin */
79 | /*!< *****
* Closed Loop
* control *****/
80 | #define SPEED_LOOP_TIME
    1         /*!< Speed Loop time (1 = 1msec) */
81 | #define KP_GAIN
    500      /*!< Kp parameter for PI regulator */
82 | #define KI_GAIN
    50        /*!< Ki parameter for PI regulator */
83 | #define KP_DIV
    4096     /*!< Kp parameter divider for PI
regulator */
84 | #define KI_DIV
    4096     /*!< Ki parameter divider for PI
regulator */
85 | #define LOWER_OUT_LIMIT
    50        /*!< Low Out value of PI regulator */
86 | #define UPPER_OUT_LIMIT
    800      /*!< High Out value of PI regulator
*/
87 | #define MAX_POT_SPEED
    20000    /*!< Maximum Speed regulated by
potentiometer */
```

```

88 #define MIN_POT_SPEED
1700 /*!< Minimum Speed regulated by
potentiometer */
89 #define VAL_POT_SPEED_DIV
2 /*!< Validation potentiometer speed
divider */
90 #define INITIAL_DEMAGN_DELAY
10 /*!< Initial value for delay time
during startup for Bemf detection */
91
92 /*!< Zero Crossission parameters */
93 #define BEMF_THRSLD_DOWN
200 /*!< Zero Crossing threshold */
94 #define BEMF_THRSLD_UP
200 /*!< Zero Crossing threshold */
95
96 /*!< Speed filtering parameters */
97 #define FILTER_DEEP
20 /*!< Number of bits for digital filter
*/
98 #define HFBUFFERSIZE
10
99 #define ADC_SPEED_TH
82 /*!<Fixed treshold to change the
target speed (t.b.f) */
100
101 /*!< Motor stall detection parameters */
102 #define BEMF_CONSEC_DOWN_MAX
10 /*!< Maximum value of BEMF Consecutive
Threshold Falling Crossings Counter in closed
loop */
103 #define BEMF_CNT_EVENT_MAX
100 /*!< Maximum number of BEMF Counter in
open loop*/
104
105 /*!< Debug pin */
106 #define GPIO_ZERO_CROSS

```

```
1      /*!< Enable (1) the GPIO toggling for
zero crossing detection */
107 | #define GPIO_COMM
1      /*!< Enable (1) the GPIO toggling for
commutation */

108 |
109 |
110 | /*!< Demo mode parameters */
111 | #define DEMO_START_TIME
    5000      /*!< Time (msec) to keep the motor in
run mode */
112 | #define DEMO_STOP_TIME
    2000      /*!< Time (msec) to keep the motor in
stop mode */

113 |
114 | /*!< Look UP table for dynamic demagn
control of speed */
115 | #define DEMAGN_VAL_1
    1      /*!< Look UP table for dynamic demagn
control for speed into (10000,12000] or
[-12000,-10000) range */
116 | #define DEMAGN_VAL_2
    2      /*!< Look UP table for dynamic demagn
control for speed into ( 7800,10000] or
[-10000,- 7800) range */
117 | #define DEMAGN_VAL_3
    3      /*!< Look UP table for dynamic demagn
control for speed into ( 6400, 7800] or [-
7800,- 6400) range */
118 | #define DEMAGN_VAL_4
    4      /*!< Look UP table for dynamic demagn
control for speed into ( 5400, 6400] or [-
6400,- 5400) range */
119 | #define DEMAGN_VAL_5
    5      /*!< Look UP table for dynamic demagn
control for speed into ( 4650, 5400] or [-
5400,- 4650) range */
```

```
120 #define DEMAGN_VAL_6
6     /*!< Look UP table for dynamic demagn
control for speed into ( 4100, 4650] or [-
4650, - 4100) range */
121 #define DEMAGN_VAL_7
7     /*!< Look UP table for dynamic demagn
control for speed into ( 3650, 4100] or [-
4100, - 3650) range */
122 #define DEMAGN_VAL_8
8     /*!< Look UP table for dynamic demagn
control for speed into ( 3300, 3650] or [-
3650, - 3300) range */
123 #define DEMAGN_VAL_9
9     /*!< Look UP table for dynamic demagn
control for speed into ( 2600, 3300] or [-
3300, - 2600) range */
124 #define DEMAGN_VAL_10
10    /*!< Look UP table for dynamic demagn
control for speed into ( 1800, 2600] or [-
2600, - 1800) range */
125 #define DEMAGN_VAL_11
11    /*!< Look UP table for dynamic demagn
control for speed into ( 1500, 1800] or [-
1800, - 1500) range */
126 #define DEMAGN_VAL_12
12    /*!< Look UP table for dynamic demagn
control for speed into ( 1300, 1500] or [-
1500, - 1300) range */
127 #define DEMAGN_VAL_13
13    /*!< Look UP table for dynamic demagn
control for speed into ( 1000, 1300] or [-
1300, - 1000) range */
128 #define DEMAGN_VAL_14
14    /*!< Look UP table for dynamic demagn
control for speed into [ 500, 1000] or [-
1000, - 500] range */
129
```

```
130 #define TRUE  
1     /*!< Define TRUE */  
131 #define FALSE  
0     /*!< Define FALSE */  
132  
133 /**  
134 * @} Main_Motor_parameters  
135 */  
136  
137  
138 /**  
139 * @} end MC_6-STEP_LIB  
140 */  
141  
142 /**  
143 * @} end MIDDLEWARES  
144 */
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Src

stm32f4xx_it.c

Go to the documentation of this file.

```
1  /**
2
3  ****
4  * @file      stm32f4xx_it.c
5  * @date      10/07/2016
6  * @brief     Interrupt Service Routines.
7
8  ****
9
10 * Redistribution and use in source and
11 * binary forms, with or without modification,
12 * are permitted provided that the
13 * following conditions are met:
14 *   1. Redistributions of source code must
15 *      retain the above copyright notice,
16 *      this list of conditions and the
17 *      following disclaimer.
18 *   2. Redistributions in binary form must
19 *      reproduce the above copyright notice,
```

```
15 *      this list of conditions and the
16 *      following disclaimer in the documentation
17 *      and/or other materials provided
18 *      with the distribution.
19 *      3. Neither the name of
20 *      STMicroelectronics nor the names of its
21 *      contributors
22 *      may be used to endorse or promote
23 *      products derived from this software
24 *      without specific prior written
25 *      permission.
26 *
27 *      THIS SOFTWARE IS PROVIDED BY THE
28 *      COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
29 *      AND ANY EXPRESS OR IMPLIED WARRANTIES,
30 *      INCLUDING, BUT NOT LIMITED TO, THE
31 *      IMPLIED WARRANTIES OF MERCHANTABILITY
32 *      AND FITNESS FOR A PARTICULAR PURPOSE ARE
33 *      DISCLAIMED. IN NO EVENT SHALL THE
34 *      COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
35 *      FOR ANY DIRECT, INDIRECT, INCIDENTAL,
36 *      SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
37 *      DAMAGES (INCLUDING, BUT NOT LIMITED TO,
38 *      PROCUREMENT OF SUBSTITUTE GOODS OR
39 *      SERVICES; LOSS OF USE, DATA, OR PROFITS;
40 *      OR BUSINESS INTERRUPTION) HOWEVER
41 *      CAUSED AND ON ANY THEORY OF LIABILITY,
42 *      WHETHER IN CONTRACT, STRICT LIABILITY,
43 *      OR TORT (INCLUDING NEGLIGENCE OR
44 *      OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
45 *      OF THIS SOFTWARE, EVEN IF ADVISED OF THE
46 *      POSSIBILITY OF SUCH DAMAGE.
47 *
48 *
49 ****
50 ****
51 */
52
```

```
34 /* Includes -----*/
-----*/
35 #include "stm32f4xx_hal.h"
36 #include "stm32f4xx.h"
37 #include "stm32f4xx_it.h"
38 /* USER CODE BEGIN 0 */
39 #include "6Step_Lib.h"
40
41 extern SIXSTEP_Base_InitTypeDef
  SIXSTEP_parameters; /*!< Main SixStep
  structure*/
42 extern void UART_Set_Value(void);
43 /* USER CODE END 0 */
44 /* External variables -----*/
-----*/
45
46 extern ADC_HandleTypeDef hadc1;
47 extern TIM_HandleTypeDef htim1;
48 extern TIM_HandleTypeDef htim4;
49 extern UART_HandleTypeDef huart2;
50
51
52 /*****
*****/
53 /*          Cortex-M4 Processor
   Interruption and Exception Handlers */
54 /*****
*****/
55
56 /**
57 * @brief This function handles ADC1 global
  interrupt.
58 */
59 void ADC_IRQHandler(void)
60 {
61 /* USER CODE BEGIN ADC_IRQHandler 0 */
62
```

```
63  /* USER CODE END ADC_IRQHandler_0 */  
64  HAL_ADC_IRQHandler(&hadc1);  
65  /* USER CODE BEGIN ADC_IRQHandler_1 */  
66  
67  /* USER CODE END ADC_IRQHandler_1 */  
68 }  
69  
70 /**
71 * @brief This function handles System tick
72 timer.  
73 */  
74 void SysTick_Handler(void)  
75 {  
76  /* USER CODE BEGIN SysTick_IRQHandler_0 */  
77  
78  /* USER CODE END SysTick_IRQHandler_0 */  
79  HAL_IncTick();  
80  HAL_SYSTICK_IRQHandler();  
81  /* USER CODE BEGIN SysTick_IRQHandler_1 */  
82  
83  /* USER CODE END SysTick_IRQHandler_1 */  
84 }  
85 /**
86 * @brief This function handles USART2 global
87 interrupt.  
88 */  
89 void USART2_IRQHandler(void)  
90 {  
91  /* USER CODE BEGIN USART2_IRQHandler_0 */  
92  
93  /* USER CODE END USART2_IRQHandler_0 */  
94  HAL_UART_IRQHandler(&huart2);  
95  /* USER CODE BEGIN USART2_IRQHandler_1 */  
96  #ifdef UART_COMM  
97    UART_Set_Value();  
  #endif
```

```
98     /* USER CODE END USART2_IRQHandler 1 */
99 }
100
101 /**
102 * @brief This function handles TIM4 global
103 * interrupt.
104 */
105 void TIM4_IRQHandler(void)
106 {
107     /* USER CODE BEGIN TIM4_IRQHandler 0 */
108
109     /* USER CODE END TIM4_IRQHandler 0 */
110     HAL_TIM_IRQHandler(&htim4);
111     /* USER CODE BEGIN TIM4_IRQHandler 1 */
112
113     /* USER CODE END TIM4_IRQHandler 1 */
114 }
115 /**
116 * @brief This function handles EXTI
117 * Line[15:10] interrupts.
118 */
119 void EXTI15_10_IRQHandler(void)
120 {
121     /* USER CODE BEGIN EXTI15_10_IRQHandler 0 */
122
123     /* USER CODE END EXTI15_10_IRQHandler 0 */
124     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_13);
125     /* USER CODE BEGIN EXTI15_10_IRQHandler 1 */
126
127     /* USER CODE END EXTI15_10_IRQHandler 1 */
128 }
129 /**
130 * @brief This function handles TIM1 Break
131 * interrupt and TIM9 global interrupt.
132 */
133
```

```
132 void TIM1_BRK_TIM9_IRQHandler(void)
133 {
134     /* USER CODE BEGIN TIM1_BRK_TIM9_IRQHandler_0 */
135     if(__HAL_TIM_GET_FLAG(&htim1,
136         TIM_FLAG_BREAK) != RESET)
137     {
138         MC_StopMotor();
139         SIXSTEP_parameters.STATUS = OVERCURRENT;
140     }
141     /* USER CODE END TIM1_BRK_TIM9_IRQHandler_0 */
142     HAL_TIM_IRQHandler(&htim1);
143     /* USER CODE BEGIN TIM1_BRK_TIM9_IRQHandler_1 */
144
145     /* USER CODE END TIM1_BRK_TIM9_IRQHandler_1 */
146
147     /* USER CODE BEGIN 1 */
148
149     /* USER CODE END 1 */
150     /***** (C) COPYRIGHT
STMicroelectronics *****END OF FILE****/
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Src

main_F401.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      main_F401.c
4  * @author    IPC
5  * @version   V0
6  * @date     10/07/2016
7  * @brief    This file provides a set of
functions needed to configure STM32 MCU.
8
****

9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
15 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34  *
35 ****
36 */
37 /* Includes -----
38 */-----*/
39 #include "stm32f4xx_hal.h"
40
41 /* USER CODE BEGIN Includes */
42 #include "6Step_Lib.h"
43 /* USER CODE END Includes */
44
45 /* Private variables -----
46 */-----*/
47 ADC_HandleTypeDef hadc1;
48 TIM_HandleTypeDef htim1;
49 TIM_HandleTypeDef htim2;
50 TIM_HandleTypeDef htim3;
51 TIM_HandleTypeDef htim4;
52
53 UART_HandleTypeDef huart2;
54
55 /* USER CODE BEGIN PV */
56 /* USER CODE END PV */
57
58 /* Private function prototypes -----
59 */-----*/
60 void SystemClock_Config(void);
61 static void MX_ADC1_Init(void);
62 static void MX_TIM1_Init(void);
63 static void MX_TIM3_Init(void);
64 static void MX_TIM4_Init(void);
65 static void MX_USART2_UART_Init(void);
```

```
66  /* USER CODE BEGIN PFP */
67
68  /* USER CODE END PFP */
69
70  /* USER CODE BEGIN 0 */
71
72  /* USER CODE END 0 */
73
74 int main(void)
75 {
76
77  /* USER CODE BEGIN 1 */
78
79  /* USER CODE END 1 */
80
81  /* MCU Configuration-----*/
-----*/
82
83  /* Reset of all peripherals, Initializes
   the Flash interface and the Systick. */
84  HAL_Init();
85
86  /* Configure the system clock */
87  SystemClock_Config();
88
89  /* Initialize all configured peripherals
 */
90  MX_ADC1_Init();
91  MX_TIM1_Init();
92  MX_TIM3_Init();
93  MX_TIM4_Init();
94  MX_USART2_UART_Init();
95
96
97  /* USER CODE BEGIN 2 */
98  /*
```

```
*****
99 |
=====
=====
100      ##### This function initializes
       6-Step lib #####
101 |
=====
=====
102 |
*****
103     MC_SixStep_INIT();
104
104 /* **** */
105 /* USER CODE END 2 */
106
107 /* USER CODE BEGIN 3 */
108 /* Infinite loop */
109 while (1)
110 {
111 /* !
111 ****
112
=====
=====
113      ##### How to use the 6Step FW
       Example project #####
114
=====
=====
115 This workspace contains the middleware
       layer with Motor Control library to drive
116 a motor connected on X-Nucleo board
       performing a 6-step control algorithm
```

```
117    allowing the motor speed regulation  
118    through a potentiometer. The 6-step algorithm  
119    is based on 1shunt current sensing mode  
120    and sensorless algorithm for bEmf detection.  
121    The workspace is provided for STM32Fxx-  
122    Nucleo in four different configurations,  
123    normal, demo, comm mode, boot mode. The  
124    "normal" mode waits the blue button event  
125    to start the motor, the "demo" mode starts  
126    and stop the motor automatically, the  
127    "comm" mode enables the communication  
128    protocol with external PC terminal and the  
129    "boot" mode enables the FW for external  
130    boot loader.  
131  
132    A list of APIs is provided to send  
133    command to 6Step lib, for instance:  
134  
135  
136    ###### USER SPACE  
137  
138
```

```
*****
***** /  
139  
140  
141  
***** /  
***** /  
142 }  
143 /* USER CODE END 3 */  
144  
145 }  
146  
147 /** System Clock Configuration  
148 */  
149 void SystemClock_Config(void)  
150 {  
151  
152     RCC_OscInitTypeDef RCC_OscInitStruct;  
153     RCC_ClkInitTypeDef RCC_ClkInitStruct;  
154  
155     __PWR_CLK_ENABLE();  
156  
157     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_  
VOLTAGE_SCALE2);  
158  
159     RCC_OscInitStruct.OscillatorType =  
RCC_OSCILLATORTYPE_HSI;  
160     RCC_OscInitStruct.HSISState = RCC_HSI_ON;  
161     RCC_OscInitStruct.HSICalibrationValue = 6;  
162     RCC_OscInitStruct.PLL.PLLState =  
RCC_PLL_ON;  
163     RCC_OscInitStruct.PLL.PLLSource =  
RCC_PLLSOURCE_HSI;  
164     RCC_OscInitStruct.PLL.PLLM = 16;  
165     RCC_OscInitStruct.PLL.PLLN = 336;  
166     RCC_OscInitStruct.PLL.PLLP =
```

```
    RCC_PLLP_DIV4;
167    RCC_OscInitStruct.PLL.PLLQ = 7;
168    HAL_RCC_OscConfig(&RCC_OscInitStruct);
169
170    RCC_ClkInitStruct.ClockType =
171        RCC_CLOCKTYPE_SYSCLK|RCC_CLOCKTYPE_PCLK1;
172    RCC_ClkInitStruct.SYSCLKSource =
173        RCC_SYSCLKSOURCE_PLLCLK;
174    RCC_ClkInitStruct.AHCLKDivider =
175        RCC_SYSCLK_DIV1;
176    RCC_ClkInitStruct.APB1CLKDivider =
177        RCC_HCLK_DIV2;
178    RCC_ClkInitStruct.APB2CLKDivider =
179        RCC_HCLK_DIV1;
180    HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
181                      FLASH_LATENCY_2);
182
183    /*
184     * ADC1 init function
185     */
186
187    void MX_ADC1_Init(void)
188    {
189
190        /*Configure the global features of the
191         *ADC (Clock, Resolution, Data Alignment and
192         *number of conversion)
193         */
194
195        hadc1.Instance = ADC1;
196        hadc1.Init.ClockPrescaler =
197            ADC_CLOCKPRESCALER_PCLK_DIV2;
198        hadc1.Init.Resolution = ADC_RESOLUTION12b;
199        hadc1.Init.ScanConvMode = DISABLE;
200        hadc1.Init.ContinuousConvMode = DISABLE;
201        hadc1.Init.DiscontinuousConvMode =
202            DISABLE;
```

```

193     hadc1.Init.ExternalTrigConvEdge =
194         ADC_EXTERNALTRIGCONVEDGE_RISING;
195     hadc1.Init.ExternalTrigConv =
196         ADC_EXTERNALTRIGCONV_T3_TRGO;
197     hadc1.Init.DataAlign =
198         ADC_DATAALIGN_RIGHT;
199     hadc1.Init.NbrOfConversion = 1;
200     hadc1.Init.DMAContinuousRequests =
201         DISABLE;
202     hadc1.Init.EOCSelection = EOC_SINGLE_CONV;
203     HAL_ADC_Init(&hadc1);
204
205     /*Configure for the selected ADC
206      regular channel its corresponding rank in the
207      sequencer and its sample time.
208 */
209     sConfig.Channel = ADC_CHANNEL_2;
210     sConfig.Rank = 1;
211     sConfig.SamplingTime =
212         ADC_SAMPLETIME_3CYCLES;
213     HAL_ADC_ConfigChannel(&hadc1, &sConfig);
214
215 }
216
217
218 /* TIM1 init function */
219 void MX_TIM1_Init(void)
220 {
221     TIM_OC_InitTypeDef sConfig;
222     TIM_SlaveConfigTypeDef sSlaveConfig;
223     TIM_MasterConfigTypeDef sMasterConfig;
224
225     /*##-1- Configure the TIM peripheral
226      ######*/
227
228     /* Initialize TIMx peripheral as follow:
229      + Prescaler = 0
230      + Period = uwPeriod (to have an

```

```

        output frequency equal to 17.57 KHz)
222            + ClockDivision = 0
223            + Counter direction = Up
224        */
225
226    htim1.Instance = TIM1;
227
228    htim1.Init.Period      = 839;
229    htim1.Init.Prescaler   = 0;
230    htim1.Init.ClockDivision =
231        TIM_CLOCKDIVISION_DIV1;
232    htim1.Init.CounterMode = TIM_COUNTERMODE_CENTERALIGNED1;
233    htim1.Init.RepetitionCounter = 0;
234
235
236
237    sSlaveConfig.SlaveMode =
238        TIM_SLAVERESET_TRIGGER;
239    sSlaveConfig.InputTrigger = TIM_TS_ITR3;
240    HAL_TIM_SlaveConfigSynchronization(&htim1,
241        &sSlaveConfig);
242
243    sMasterConfig.MasterOutputTrigger =
244        TIM_TRGO_RESET;
245    sMasterConfig.MasterSlaveMode =
246        TIM_MASTERSLAVEMODE_DISABLE;
247
248    HAL_TIMEx_MasterConfigSynchronization(&htim1,
249        &sMasterConfig);
250
251    /*##-2- Configure the PWM channels
252    #####
253    */
254
255    /* Common configuration for all channels
256    */
257
258    sConfig.OCMode      = TIM_OCMODE_PWM2;

```

```
248     sConfig.OCFastMode  = TIM_OCFAST_DISABLE;
249     sConfig.OCPolarity = TIM_OCPOLARITY_LOW;
250     sConfig.OCNPolarity = TIM_OCNPOLARITY_LOW;
251     sConfig.OCIdleState =
252         TIM_OCIDLESTATE_RESET;
253     sConfig.OCNIdleState=
254         TIM_OCNIDLESTATE_RESET;
255     sConfig.Pulse = 100;
256
257     /* Set the pulse value for channel 1 */
258     HAL_TIM_PWM_ConfigChannel(&htim1,
259                               &sConfig, TIM_CHANNEL_1) ;
260
261     /* Set the pulse value for channel 2 */
262     HAL_TIM_PWM_ConfigChannel(&htim1,
263                               &sConfig, TIM_CHANNEL_2) ;
264
265
266
267
268
269
270     /* TIM3 init function */
271 void MX_TIM3_Init(void)
272 {
273
274     TIM_SlaveConfigTypeDef sSlaveConfig;
275     TIM_MasterConfigTypeDef sMasterConfig;
276     TIM_OC_InitTypeDef sConfigOC;
277
278     htim3.Instance = TIM3;
279     htim3.Init.Prescaler = 0;
```

```
280     htim3.Init.CounterMode =
281         TIM_COUNTERMODE_CENTERALIGNED1;
282     htim3.Init.Period = 839;
283     htim3.Init.ClockDivision =
284         TIM_CLOCKDIVISION_DIV1;
285     HAL_TIM_Base_Init(&htim3);
286
287     sSlaveConfig.SlaveMode =
288         TIM_SLAVEMODE_TRIGGER;
289     sSlaveConfig.InputTrigger = TIM_TS_ITR3;
290     HAL_TIM_SlaveConfigSynchronization(&htim3,
291                                         &sSlaveConfig);
292
293     sMasterConfig.MasterOutputTrigger =
294         TIM_TRGO_UPDATE;
295     sMasterConfig.MasterSlaveMode =
296         TIM_MASTERSLAVEMODE_ENABLE;
297
298     HAL_TIMEx_MasterConfigSynchronization(&htim3,
299                                         &sMasterConfig);
300
301     sConfigOC.OCMode = TIM_OCMODE_PWM1;
302     sConfigOC.Pulse = 671;
303     sConfigOC.OCPolarity =
304         TIM_OCPOLARITY_HIGH;
305     sConfigOC.OCFastMode = TIM_OCFAST_ENABLE;
306     HAL_TIM_PWM_ConfigChannel(&htim3,
307                               &sConfigOC, TIM_CHANNEL_1);
308
309 }
310
311 /* TIM4 init function */
312 void MX_TIM4_Init(void)
313 {
314 }
```

```
307     TIM_ClockConfigTypeDef sClockSourceConfig;
308     TIM_MasterConfigTypeDef sMasterConfig;
309
310     htim4.Instance = TIM4;
311     htim4.Init.Prescaler = 0;
312     htim4.Init.CounterMode =
313         TIM_COUNTERMODE_UP;
314     htim4.Init.Period = 24000;
315     htim4.Init.ClockDivision =
316         TIM_CLOCKDIVISION_DIV1;
317     HAL_TIM_Base_Init(&htim4);
318
319     sClockSourceConfig.ClockSource =
320         TIM_CLOCKSOURCE_INTERNAL;
321     HAL_TIM_ConfigClockSource(&htim4,
322         &sClockSourceConfig);
323
324 }
325
326 /* USART2 init function */
327 void MX_USART2_UART_Init(void)
328 {
329
330     huart2.Instance = USART2;
331     huart2.Init.BaudRate = 19200;
332     huart2.Init.WordLength =
333         UART_WORDLENGTH_9B;
334     huart2.Init.StopBits = UART_STOPBITS_1;
335     huart2.Init.Parity = UART_PARITY_ODD;
```

```
335     huart2.Init.Mode = UART_MODE_TX_RX;
336     huart2.Init.HwFlowCtl =
337         UART_HWCONTROL_NONE;
338     huart2.Init.OverSampling =
339         UART_OVERSAMPLING_16;
340     HAL_UART_Init(&huart2);
341
342
343
344 /* USER CODE BEGIN 4 */
345
346 /* USER CODE END 4 */
347
348 #ifdef USE_FULL_ASSERT
349
350 /**
351  * @brief Reports the name of the source
352  * file and the source line number
353  * where the assert_param error has
354  * occurred.
355  * @param file: pointer to the source file
356  * name
357  * @param line: assert_param error line
358  * source number
359  * @retval None
360  */
361 void assert_failed(uint8_t* file, uint32_t
362 line)
363 {
364     /* USER CODE BEGIN 6 */
365     /* User can add his own implementation to
366      report the file name and line number,
367      ex: printf("Wrong parameters value: file
368          %s on line %d\r\n", file, line) */
369
370     /* USER CODE END 6 */
```

```
363 }
364 }
365 #endif
366 /**
367 * @}
368 */
369 /**
370 * @}
371 */
372 /**
373 * @}
374 */
375
376 /***** (C) COPYRIGHT
    STMicroelectronics *****END OF FILE****/
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	UART_serial_com	Inc

UART_UI.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      UART_UI.h
4  * @author    System lab
5  * @version   V1.0.0
6  * @date     06-July-2015
7  * @brief    This file provides a set of
functions needed to manage the UART com.
8
****

9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
15 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34    *
35 ****
36    */
37
38 #include "6Step_Lib.h"
39 #include "stdlib.h"
40 #include "stdio.h"
41 #include <string.h>
42
43 #define TOKEN "\r"
44 #define CMD_NUM 16
45
46 #define COUNTOF(__BUFFER__)
47 (sizeof(__BUFFER__) / sizeof(*(__BUFFER__)))
48 #define TXBUFFERSIZE
49 (COUNTOF(aTxBuffer) - 1)
50 #define RXBUFFERSIZE 8
51
52 typedef struct {
53     char name[10];
54     void (*pCmdFunc)(void);
55 } CMD_T;
56
57 /**
58 * @addtogroup MIDDLEWARES      MIDDLEWARES
59 * @brief Middlewares Layer
60 * @{
61 */
62
63 /**
64 * @addtogroup UART_UI  UART_UI
65 * @brief Serial communication through PC
66 * serial terminal
67 * @{
68 */
69
```

```
66  /** @defgroup Exported_function_Uart
67  * @{
68  */
69 /**
70 * @brief  UART function
71 */
72
73 void CMD_STARTM( void );
74 void CMD_STOPMT( void );
75 void CMD_DIRECTION( void );
76 void CMD_SETSPD( void );
77 void CMD_GETSPD( void );
78 void CMD_STATUS( void );
79 void CMD_POTENZ( void );
80 void CMD_HELP(void);
81 void CMD_INIREF(void);
82 void CMD_POLESP(void);
83 void CMD_ACCELE(void);
84 void CMD_KP_PRM(void);
85 void CMD_KI_PRM(void);
86
87 /**
88 * @}
89 */
90
91 /**
92 * @}
93 */
94
95 /**
96 * @}
97 */
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

main_F401.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      main_F401.h
4  * @author    IPC
5  * @version   V0
6  * @date     10/07/2016
7  * @brief    Main program body for
8  *           STM32F401xx
9  *
10 * COPYRIGHT(c) 2015 STMicroelectronics
11 *
12 * Redistribution and use in source and
13 * binary forms, with or without modification,
14 * are permitted provided that the
15 * following conditions are met:
16 *   1. Redistributions of source code must
17 *      retain the above copyright notice,
18 *      this list of conditions and the
```

following disclaimer.

16 | * 2. Redistributions in binary form must
reproduce the above copyright notice,
17 | * this list of conditions and the
following disclaimer in the documentation
18 | * and/or other materials provided
with the distribution.

19 | * 3. Neither the name of
STMicroelectronics nor the names of its
contributors

20 | * may be used to endorse or promote
products derived from this software

21 | * without specific prior written
permission.

22 | *

23 | * THIS SOFTWARE IS PROVIDED BY THE
COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
24 | * AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE
25 | * IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE
26 | * DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
27 | * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
28 | * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR
29 | * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER
30 | * CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY,
31 | * OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
32 | * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

33 | *

34 | *

```
*****  
*****  
35 */  
36  
37 extern ADC_HandleTypeDef hadc1;  
38 extern TIM_HandleTypeDef htim1;  
39 extern TIM_HandleTypeDef htim3;  
40 extern TIM_HandleTypeDef htim4;  
41 extern UART_HandleTypeDef huart2;
```

Generated by  1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	Inc
stm32_nucleo_ihm11m1.h			

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      stm32_nucleo_ihm11m1.h
4  * @author    System lab - Automation and
5  Motion control team
6  * @version   V1.0.0
7  * @date     10/07/2016
8  * @brief    This file provides the interface
9  between the MC-lib and STM Nucleo
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
12 STMicroelectronics</center></h2>
13 *
14 * Redistribution and use in source and
binary forms, with or without modification,
15 * are permitted provided that the following
conditions are met:
```

```
15 *      1. Redistributions of source code must
16 *         retain the above copyright notice,
17 *         this list of conditions and the
18 *         following disclaimer.
19 *      2. Redistributions in binary form must
20 *         reproduce the above copyright notice,
21 *         this list of conditions and the
22 *         following disclaimer in the documentation
23 *         and/or other materials provided with
24 *         the distribution.
25 *      3. Neither the name of
26 *         STMicroelectronics nor the names of its
27 *         contributors
28 *            may be used to endorse or promote
29 *            products derived from this software
30 *            without specific prior written
31 *            permission.
32 *
33 * THIS SOFTWARE IS PROVIDED BY THE
34 * COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
35 * AND ANY EXPRESS OR IMPLIED WARRANTIES,
36 * INCLUDING, BUT NOT LIMITED TO, THE
37 * IMPLIED WARRANTIES OF MERCHANTABILITY AND
38 * FITNESS FOR A PARTICULAR PURPOSE ARE
39 * DISCLAIMED. IN NO EVENT SHALL THE
40 * COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
41 * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
42 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
43 * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
44 * PROCUREMENT OF SUBSTITUTE GOODS OR
45 * SERVICES; LOSS OF USE, DATA, OR PROFITS;
46 * OR BUSINESS INTERRUPTION) HOWEVER
47 * CAUSED AND ON ANY THEORY OF LIABILITY,
48 * WHETHER IN CONTRACT, STRICT LIABILITY,
49 * OR TORT (INCLUDING NEGLIGENCE OR
50 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
51 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
```

POSSIBILITY OF SUCH DAMAGE.

```
34 *  
35 ****  
36 */  
37  
38 #ifdef STM32F030x8  
39 #include "stm32F030_nucleo_ihm11m1.h"  
40 #include "MC_SixStep_param_F030.h"  
41 #endif  
42 #ifdef STM32F401xE  
43 #include "stm32F401_nucleo_ihm11m1.h"  
44 #include "MC_SixStep_param.h"  
45 #endif  
46  
47  
48
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Inc

stm32f4xx_hal_conf.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      stm32f4xx_hal_conf.h
4  * @brief     HAL configuration file.
5  ****
6  * @attention
7  *
8  * <h2><center>&copy; COPYRIGHT(c) 2015
9  STMicroelectronics</center></h2>
10 * Redistribution and use in source and
11 * binary forms, with or without modification,
12 * are permitted provided that the
13 * following conditions are met:
14 *   1. Redistributions of source code must
15 *      retain the above copyright notice,
16 *   2. Redistributions in binary form must
```

reproduce the above copyright notice,

15 | * this list of conditions and the
following disclaimer in the documentation

16 | * and/or other materials provided
with the distribution.

17 | * 3. Neither the name of
STMicroelectronics nor the names of its
contributors

18 | * may be used to endorse or promote
products derived from this software

19 | * without specific prior written
permission.

20 | *

21 | * THIS SOFTWARE IS PROVIDED BY THE
COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

22 | * AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE

23 | * IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE

24 | * DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

25 | * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

26 | * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR

27 | * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER

28 | * CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY,

29 | * OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

30 | * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

31 | *

32 | *

```
33  */
34
35 /* Define to prevent recursive inclusion --- */
36 #ifndef __STM32F4xx_HAL_CONF_H
37 #define __STM32F4xx_HAL_CONF_H
38
39 #ifdef __cplusplus
40   extern "C" {
41 #endif
42
43 /* Exported types ----- */
44 /* Exported constants ----- */
45
46 /* ##### Module Selection ##### */
47 /**
48  * @brief This is the list of modules to be
49  *        used in the HAL driver
50 */
51 #define HAL_MODULE_ENABLED
52 // #define HAL_CAN_MODULE_ENABLED
53 // #define HAL_CRC_MODULE_ENABLED
54 // #define HAL_CRYP_MODULE_ENABLED
55 // #define HAL_DAC_MODULE_ENABLED
56 // #define HAL_DCMI_MODULE_ENABLED
57 // #define HAL_DMA2D_MODULE_ENABLED
58 // #define HAL_ETH_MODULE_ENABLED
59 // #define HAL_NAND_MODULE_ENABLED
60 // #define HAL_NOR_MODULE_ENABLED
61 // #define HAL_PCCARD_MODULE_ENABLED
62 // #define HAL_SRAM_MODULE_ENABLED
63 // #define HAL_SDRAM_MODULE_ENABLED
64 // #define HAL_HASH_MODULE_ENABLED
```

```
65 // #define HAL_I2C_MODULE_ENABLED
66 // #define HAL_I2S_MODULE_ENABLED
67 // #define HAL_IWDG_MODULE_ENABLED
68 // #define HAL_LTDC_MODULE_ENABLED
69 // #define HAL_RNG_MODULE_ENABLED
70 // #define HAL_RTC_MODULE_ENABLED
71 // #define HAL_SAI_MODULE_ENABLED
72 // #define HAL_SD_MODULE_ENABLED
73 #define HAL_SPI_MODULE_ENABLED
74 #define HAL_TIM_MODULE_ENABLED
75 #define HAL_UART_MODULE_ENABLED
76 // #define HAL_USART_MODULE_ENABLED
77 // #define HAL_IRDA_MODULE_ENABLED
78 // #define HAL_SMARTCARD_MODULE_ENABLED
79 // #define HAL_WWDG_MODULE_ENABLED
80 // #define HAL_PCD_MODULE_ENABLED
81 // #define HAL_HCD_MODULE_ENABLED
82 #define HAL_GPIO_MODULE_ENABLED
83 #define HAL_DMA_MODULE_ENABLED
84 #define HAL_RCC_MODULE_ENABLED
85 #define HAL_FLASH_MODULE_ENABLED
86 #define HAL_PWR_MODULE_ENABLED
87 #define HAL_CORTEX_MODULE_ENABLED
88
89 /* ##### HSE/HSI Values adaptation ##### */
90 /**
91 * @brief Adjust the value of External High Speed oscillator (HSE) used in your application.
92 * This value is used by the RCC HAL module to compute the system frequency
93 * (when HSE is used as system clock source, directly or through the PLL).
94 */
95 #if !defined (HSE_VALUE)
96 #define HSE_VALUE ((uint32_t)8000000)
```

```
/*!< Value of the External oscillator in Hz */
97 #endif /* HSE_VALUE */
98
99 #if !defined (HSE_STARTUP_TIMEOUT)
100 #define HSE_STARTUP_TIMEOUT
((uint32_t)5000) /*!< Time out for HSE start
up, in ms */
101#endif /* HSE_STARTUP_TIMEOUT */
102
103 /**
104 * @brief Internal High Speed oscillator
(HSI) value.
105 * This value is used by the RCC HAL
module to compute the system frequency
106 * (when HSI is used as system clock
source, directly or through the PLL).
107 */
108 #if !defined (HSI_VALUE)
109 #define HSI_VALUE ((uint32_t)16000000)
/*!< Value of the Internal oscillator in Hz*/
110#endif /* HSI_VALUE */
111
112 /**
113 * @brief Internal Low Speed oscillator
(LSI) value.
114 */
115 #if !defined (LSI_VALUE)
116 #define LSI_VALUE ((uint32_t)32000)
117#endif /* LSI_VALUE */ /*!< Value of the
Internal Low Speed oscillator in Hz
118
The real value may vary depending on the
variations
119
in voltage and temperature. */
120 /**
121 * @brief External Low Speed oscillator
```

```
        (LSE) value.
122    */
123 #if !defined (LSE_VALUE)
124 #define LSE_VALUE ((uint32_t)32768) /*!
< Value of the External Low Speed oscillator
in Hz */
125#endif /* LSE_VALUE */
126
127 /**
128 * @brief External clock source for I2S
peripheral
129 *          This value is used by the I2S HAL
module to compute the I2S clock source
130 *          frequency, this source is
inserted directly through I2S_CKIN pad.
131 */
132 #if !defined (EXTERNAL_CLOCK_VALUE)
133 #define EXTERNAL_CLOCK_VALUE
((uint32_t)12288000) /*!< Value of the
External audio frequency in Hz*/
134#endif /* EXTERNAL_CLOCK_VALUE */
135
136 /* Tip: To avoid modifying this file each
time you need to use different HSE,
137 === you can define the HSE value in your
toolchain compiler preprocessor. */
138
139 /* ##### System
Configuration ##### */
140 /**
141 * @brief This is the HAL system
configuration section
142 */
143
144#define VDD_VALUE
((uint32_t)3300) /*!< Value of VDD in mv */
145#define TICK_INT_PRIORITY
```

```

        ((uint32_t)2) /*!< tick interrupt priority
*/
146 #define USE_RTOS          0
147 #define PREFETCH_ENABLE    1
148 #define INSTRUCTION_CACHE_ENABLE 1
149 #define DATA_CACHE_ENABLE   1
150
151 /* ##### Assert
Selection ##### */
152 /**
153 * @brief Uncomment the line below to
expance the "assert_param" macro in the
154 *           HAL drivers code
155 */
156 /* #define USE_FULL_ASSERT    1 */
157
158 /* ##### Ethernet peripheral
configuration ##### */
159
160 /* Section 1 : Ethernet peripheral
configuration */
161
162 /* MAC ADDRESS:
MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_AD
DR4:MAC_ADDR5 */
163 #define MAC_ADDR0    2
164 #define MAC_ADDR1    0
165 #define MAC_ADDR2    0
166 #define MAC_ADDR3    0
167 #define MAC_ADDR4    0
168 #define MAC_ADDR5    0
169
170 /* Definition of the Ethernet driver buffers
size and count */
171 #define ETH_RX_BUF_SIZE
ETH_MAX_PACKET_SIZE /* buffer size for receive
*/

```

```
172 #define ETH_TX_BUF_SIZE  
    ETH_MAX_PACKET_SIZE /* buffer size for  
    transmit */  
173 #define ETH_RXBUFN  
    ((uint32_t)4) /* 4 Rx buffers of size  
    ETH_RX_BUF_SIZE */  
174 #define ETH_TXBUFN  
    ((uint32_t)4) /* 4 Tx buffers of size  
    ETH_TX_BUF_SIZE */  
175  
176 /* Section 2: PHY configuration section */  
177  
178 /* DP83848 PHY Address */  
179 #define DP83848_PHY_ADDRESS 0x01  
180 /* PHY Reset delay these values are based on  
   a 1 ms Systick interrupt */  
181 #define PHY_RESET_DELAY  
    ((uint32_t)0x000000FF)  
182 /* PHY Configuration delay */  
183 #define PHY_CONFIG_DELAY  
    ((uint32_t)0x00000FFF)  
184  
185 #define PHY_READ_TO  
    ((uint32_t)0x0000FFFF)  
186 #define PHY_WRITE_TO  
    ((uint32_t)0x00000FFF)  
187  
188 /* Section 3: Common PHY Registers */  
189  
190 #define PHY_BCR  
    ((uint16_t)0x00) /* !< Transceiver Basic  
    Control Register */  
191 #define PHY_BSR  
    ((uint16_t)0x01) /* !< Transceiver Basic  
    Status Register */  
192  
193 #define PHY_RESET
```

```
    ((uint16_t)0x8000) /*!< PHY Reset */  
194 | #define PHY_LOOPBACK  
     ((uint16_t)0x4000) /*!< Select loop-back mode */  
     /*/  
195 | #define PHY_FULLDUPLEX_100M  
     ((uint16_t)0x2100) /*!< Set the full-duplex  
     mode at 100 Mb/s */  
196 | #define PHY_HALFDUPLEX_100M  
     ((uint16_t)0x2000) /*!< Set the half-duplex  
     mode at 100 Mb/s */  
197 | #define PHY_FULLDUPLEX_10M  
     ((uint16_t)0x0100) /*!< Set the full-duplex  
     mode at 10 Mb/s */  
198 | #define PHY_HALFDUPLEX_10M  
     ((uint16_t)0x0000) /*!< Set the half-duplex  
     mode at 10 Mb/s */  
199 | #define PHY_AUTONEGOTIATION  
     ((uint16_t)0x1000) /*!< Enable auto-  
     negotiation function */  
200 | #define PHY_RESTART_AUTONEGOTIATION  
     ((uint16_t)0x0200) /*!< Restart auto-  
     negotiation function */  
201 | #define PHY_POWERDOWN  
     ((uint16_t)0x0800) /*!< Select the power down  
     mode */  
202 | #define PHY_ISOLATE  
     ((uint16_t)0x0400) /*!< Isolate PHY from MII  
     */  
203 |  
204 | #define PHY_AUTONEGO_COMPLETE  
     ((uint16_t)0x0020) /*!< Auto-Negotiation  
     process completed */  
205 | #define PHY_LINKED_STATUS  
     ((uint16_t)0x0004) /*!< Valid link  
     established */  
206 | #define PHY_JABBER_DETECTION  
     ((uint16_t)0x0002) /*!< Jabber condition
```

```
detected */  
207 /* Section 4: Extended PHY Registers */  
209  
210 #define PHY_SR  
((uint16_t)0x10) /*!< PHY status register  
Offset */  
211 #define PHY_MICR  
((uint16_t)0x11) /*!< MII Interrupt Control  
Register */  
212 #define PHY_MISR  
((uint16_t)0x12) /*!< MII Interrupt Status  
and Misc. Control Register */  
213  
214 #define PHY_LINK_STATUS  
((uint16_t)0x0001) /*!< PHY Link mask  
*/  
215 #define PHY_SPEED_STATUS  
((uint16_t)0x0002) /*!< PHY Speed mask  
*/  
216 #define PHY_DUPLEX_STATUS  
((uint16_t)0x0004) /*!< PHY Duplex mask  
*/  
217  
218 #define PHY_MICR_INT_EN  
((uint16_t)0x0002) /*!< PHY Enable interrupts  
*/  
219 #define PHY_MICR_INT_OE  
((uint16_t)0x0001) /*!< PHY Enable output  
interrupt events */  
220  
221 #define PHY_MISR_LINK_INT_EN  
((uint16_t)0x0020) /*!< Enable Interrupt on  
change of link status */  
222 #define PHY_LINK_INTERRUPT  
((uint16_t)0x2000) /*!< PHY link status  
interrupt mask */
```

```
223
224 /* Includes -----
225 /**
226 * @brief Include module's header file
227 */
228
229 #ifdef HAL_RCC_MODULE_ENABLED
230     #include "stm32f4xx_hal_rcc.h"
231 #endif /* HAL_RCC_MODULE_ENABLED */
232
233 #ifdef HAL_GPIO_MODULE_ENABLED
234     #include "stm32f4xx_hal_gpio.h"
235 #endif /* HAL_GPIO_MODULE_ENABLED */
236
237 #ifdef HAL_DMA_MODULE_ENABLED
238     #include "stm32f4xx_hal_dma.h"
239 #endif /* HAL_DMA_MODULE_ENABLED */
240
241 #ifdef HAL_CORTEX_MODULE_ENABLED
242     #include "stm32f4xx_hal_cortex.h"
243 #endif /* HAL_CORTEX_MODULE_ENABLED */
244
245 #ifdef HAL_ADC_MODULE_ENABLED
246     #include "stm32f4xx_hal_adc.h"
247 #endif /* HAL_ADC_MODULE_ENABLED */
248
249 #ifdef HAL_CAN_MODULE_ENABLED
250     #include "stm32f4xx_hal_can.h"
251 #endif /* HAL_CAN_MODULE_ENABLED */
252
253 #ifdef HAL_CRC_MODULE_ENABLED
254     #include "stm32f4xx_hal_crc.h"
255 #endif /* HAL_CRC_MODULE_ENABLED */
256
257 #ifdef HAL_CRYP_MODULE_ENABLED
258     #include "stm32f4xx_hal_cryp.h"
```

```
259 #endif /* HAL_CRYPT_MODULE_ENABLED */
260
261 #ifdef HAL_DMA2D_MODULE_ENABLED
262     #include "stm32f4xx_hal_dma2d.h"
263 #endif /* HAL_DMA2D_MODULE_ENABLED */
264
265 #ifdef HAL_DAC_MODULE_ENABLED
266     #include "stm32f4xx_hal_dac.h"
267 #endif /* HAL_DAC_MODULE_ENABLED */
268
269 #ifdef HAL_DCMI_MODULE_ENABLED
270     #include "stm32f4xx_hal_dcmi.h"
271 #endif /* HAL_DCMI_MODULE_ENABLED */
272
273 #ifdef HAL_ETH_MODULE_ENABLED
274     #include "stm32f4xx_hal_eth.h"
275 #endif /* HAL_ETH_MODULE_ENABLED */
276
277 #ifdef HAL_FLASH_MODULE_ENABLED
278     #include "stm32f4xx_hal_flash.h"
279 #endif /* HAL_FLASH_MODULE_ENABLED */
280
281 #ifdef HAL_SRAM_MODULE_ENABLED
282     #include "stm32f4xx_hal_sram.h"
283 #endif /* HAL_SRAM_MODULE_ENABLED */
284
285 #ifdef HAL_NOR_MODULE_ENABLED
286     #include "stm32f4xx_hal_nor.h"
287 #endif /* HAL_NOR_MODULE_ENABLED */
288
289 #ifdef HAL_NAND_MODULE_ENABLED
290     #include "stm32f4xx_hal_nand.h"
291 #endif /* HAL_NAND_MODULE_ENABLED */
292
293 #ifdef HAL_PCCARD_MODULE_ENABLED
294     #include "stm32f4xx_hal_pccard.h"
295 #endif /* HAL_PCCARD_MODULE_ENABLED */
```

```
296
297 #ifdef HAL_SDRAM_MODULE_ENABLED
298     #include "stm32f4xx_hal_sdram.h"
299 #endif /* HAL_SDRAM_MODULE_ENABLED */
300
301 #ifdef HAL_HASH_MODULE_ENABLED
302     #include "stm32f4xx_hal_hash.h"
303 #endif /* HAL_HASH_MODULE_ENABLED */
304
305 #ifdef HAL_I2C_MODULE_ENABLED
306     #include "stm32f4xx_hal_i2c.h"
307 #endif /* HAL_I2C_MODULE_ENABLED */
308
309 #ifdef HAL_I2S_MODULE_ENABLED
310     #include "stm32f4xx_hal_i2s.h"
311 #endif /* HAL_I2S_MODULE_ENABLED */
312
313 #ifdef HAL_IWDG_MODULE_ENABLED
314     #include "stm32f4xx_hal_iwdg.h"
315 #endif /* HAL_IWDG_MODULE_ENABLED */
316
317 #ifdef HAL_LTDC_MODULE_ENABLED
318     #include "stm32f4xx_hal_ltdc.h"
319 #endif /* HAL_LTDC_MODULE_ENABLED */
320
321 #ifdef HAL_PWR_MODULE_ENABLED
322     #include "stm32f4xx_hal_pwr.h"
323 #endif /* HAL_PWR_MODULE_ENABLED */
324
325 #ifdef HAL_RNG_MODULE_ENABLED
326     #include "stm32f4xx_hal_rng.h"
327 #endif /* HAL_RNG_MODULE_ENABLED */
328
329 #ifdef HAL_RTC_MODULE_ENABLED
330     #include "stm32f4xx_hal_rtc.h"
331 #endif /* HAL_RTC_MODULE_ENABLED */
332
```

```
333 #ifdef HAL_SAI_MODULE_ENABLED  
334     #include "stm32f4xx_hal_sai.h"  
335 #endif /* HAL_SAI_MODULE_ENABLED */  
336  
337 #ifdef HAL_SD_MODULE_ENABLED  
338     #include "stm32f4xx_hal_sd.h"  
339 #endif /* HAL_SD_MODULE_ENABLED */  
340  
341 #ifdef HAL_SPI_MODULE_ENABLED  
342     #include "stm32f4xx_hal_spi.h"  
343 #endif /* HAL_SPI_MODULE_ENABLED */  
344  
345 #ifdef HAL_TIM_MODULE_ENABLED  
346     #include "stm32f4xx_hal_tim.h"  
347 #endif /* HAL_TIM_MODULE_ENABLED */  
348  
349 #ifdef HAL_UART_MODULE_ENABLED  
350     #include "stm32f4xx_hal_uart.h"  
351 #endif /* HAL_UART_MODULE_ENABLED */  
352  
353 #ifdef HAL_USART_MODULE_ENABLED  
354     #include "stm32f4xx_hal_usart.h"  
355 #endif /* HAL_USART_MODULE_ENABLED */  
356  
357 #ifdef HAL_IRDA_MODULE_ENABLED  
358     #include "stm32f4xx_hal_irda.h"  
359 #endif /* HAL_IRDA_MODULE_ENABLED */  
360  
361 #ifdef HAL_SMARTCARD_MODULE_ENABLED  
362     #include "stm32f4xx_hal_smartcard.h"  
363 #endif /* HAL_SMARTCARD_MODULE_ENABLED */  
364  
365 #ifdef HAL_WWDG_MODULE_ENABLED  
366     #include "stm32f4xx_hal_wwdg.h"  
367 #endif /* HAL_WWDG_MODULE_ENABLED */  
368  
369 #ifdef HAL_PCD_MODULE_ENABLED
```

```
370 #include "stm32f4xx_hal_pcd.h"
371 #endif /* HAL_PCD_MODULE_ENABLED */
372
373 #ifdef HAL_HCD_MODULE_ENABLED
374     #include "stm32f4xx_hal_hcd.h"
375 #endif /* HAL_HCD_MODULE_ENABLED */
376
377 /* Exported macro -----
   -----*/
378 #ifdef USE_FULL_ASSERT
379 /**
380 * @brief The assert_param macro is used
381 * for function's parameters check.
382 * @param expr: If expr is false, it calls
383 * assert_failed function
384 *           which reports the name of the
385 *           source file and the source
386 *           line number of the call that
387 *           failed.
388 *           If expr is true, it returns no
389 *           value.
390 * @retval None
391 */
392 #define assert_param(expr) ((expr) ?
393     (void)0 : assert_failed((uint8_t *)__FILE__,
394     __LINE__))
395
396 /* Exported functions -----
   -----*/
397 void assert_failed(uint8_t* file, uint32_t
398 line);
399 #else
400 #define assert_param(expr) ((void)0)
401 #endif /* USE_FULL_ASSERT */
402
403 #ifdef __cplusplus
404 }
405#endif
```

```
397  
398 #endif /* __STM32F4xx_HAL_CONF_H */  
399  
400  
401 /***** (C) COPYRIGHT  
STMicroelectronics *****END OF FILE****/
```

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Src

stm32f4xx_hal_msp.c

Go to the documentation of this file.

```
1  /**
2
3  ****
4  * @file      stm32f4xx_hal_msp.c
5  * @author    IPC
6  * @version   V0
7  * @date      10/07/2016
8  * @brief     This file provides code for the
9  *             MSP Initialization
10 *             and de-Initialization codes.
11 *
12 * COPYRIGHT(c) 2015 STMicroelectronics
13 *
14 * Redistribution and use in source and
15 * binary forms, with or without modification,
16 * are permitted provided that the
17 * following conditions are met:
18 *   1. Redistributions of source code must
19 *      retain the above copyright notice,
```

```
16 *      this list of conditions and the
17 *      following disclaimer.
18 *      2. Redistributions in binary form must
19 *      reproduce the above copyright notice,
20 *      this list of conditions and the
21 *      following disclaimer in the documentation
22 *      and/or other materials provided
23 *      with the distribution.
24 *      3. Neither the name of
25 *      STMicroelectronics nor the names of its
26 *      contributors
27 *      may be used to endorse or promote
28 *      products derived from this software
29 *      without specific prior written
30 *      permission.
31 *
32 *      THIS SOFTWARE IS PROVIDED BY THE
33 *      COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
34 *      AND ANY EXPRESS OR IMPLIED WARRANTIES,
35 *      INCLUDING, BUT NOT LIMITED TO, THE
36 *      IMPLIED WARRANTIES OF MERCHANTABILITY
37 *      AND FITNESS FOR A PARTICULAR PURPOSE ARE
38 *      DISCLAIMED. IN NO EVENT SHALL THE
39 *      COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
40 *      FOR ANY DIRECT, INDIRECT, INCIDENTAL,
41 *      SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
42 *      DAMAGES (INCLUDING, BUT NOT LIMITED TO,
43 *      PROCUREMENT OF SUBSTITUTE GOODS OR
44 *      SERVICES; LOSS OF USE, DATA, OR PROFITS;
45 *      OR BUSINESS INTERRUPTION) HOWEVER
46 *      CAUSED AND ON ANY THEORY OF LIABILITY,
47 *      WHETHER IN CONTRACT, STRICT LIABILITY,
48 *      OR TORT (INCLUDING NEGLIGENCE OR
49 *      OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
50 *      OF THIS SOFTWARE, EVEN IF ADVISED OF THE
51 *      POSSIBILITY OF SUCH DAMAGE.
```

```
35 ****
36 */
37 /* Includes -----
   -----*/
38 #include "stm32f4xx_hal.h"
39 #include "stm32F401_nucleo_ihm11m1.h"
40
41 /* USER CODE BEGIN 0 */
42
43 /* USER CODE END 0 */
44
45 /**
46 * Initializes the Global MSP.
47 */
48 void HAL_MspInit(void)
49 {
50     GPIO_InitTypeDef GPIO_InitStruct;
51
52     /* GPIO Ports Clock Enable */
53     __GPIOC_CLK_ENABLE();
54     __GPIOH_CLK_ENABLE();
55     __GPIOA_CLK_ENABLE();
56     __GPIOB_CLK_ENABLE();
57     /** GPIO Configuration      IHM11M1*/
58
59     /*Configure GPIO pin : PC13 */
60     GPIO_InitStruct.Pin = GPIO_PIN_13;
61     GPIO_InitStruct.Mode =
62         GPIO_MODE_IT_FALLING;
63     GPIO_InitStruct.Pull = GPIO_PULLUP;
64     HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
65
66     GPIO_InitStruct.Pin = GPIO_CH_COMM;
67     GPIO_InitStruct.Mode =
68         GPIO_MODE_OUTPUT_PP;
```

```
67     GPIO_InitStruct.Pull = GPIO_NOPULL;
68     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
69     HAL_GPIO_Init(GPIO_PORT_COMM,
70                   &GPIO_InitStruct);
71     GPIO_InitStruct.Pin = GPIO_CH_ZCR;
72     GPIO_InitStruct.Mode =
73         GPIO_MODE_OUTPUT_PP;
74     GPIO_InitStruct.Pull = GPIO_NOPULL;
75     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
76     HAL_GPIO_Init(GPIO_PORT_ZCR,
77                   &GPIO_InitStruct);
78     /* EXTI interrupt init*/
79     HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0,
80     0);
81     HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
82     /* USER CODE BEGIN MspInit 0 */
83     /* EN PIN */
84     GPIO_InitStruct.Pin = GPIO_PIN_5;
85     GPIO_InitStruct.Mode =
86         GPIO_MODE_OUTPUT_PP;
87     GPIO_InitStruct.Pull = GPIO_NOPULL;
88     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
89     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5,
90                       GPIO_PIN_SET);
91     /* FAULT PIN */
92     GPIO_InitStruct.Pin = GPIO_PIN_4;
93     GPIO_InitStruct.Mode =
94         GPIO_MODE_IT_FALLING;
95     GPIO_InitStruct.Pull = GPIO_NOPULL;
96     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
97     /* SBY&RESET PIN */
98     GPIO_InitStruct.Pin = GPIO_PIN_7;
```

```
97     GPIO_InitStruct.Mode =
98         GPIO_MODE_OUTPUT_OD;
99     HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
100    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_7,
101        GPIO_PIN_SET);
102    /* USER CODE END MspInit 0 */
103
104
105    HAL_NVIC_SetPriorityGrouping(NVIC_PRIORITYGROUP_4);
106    /* System interrupt init*/
107    /* SysTick_IRQn interrupt configuration */
108    HAL_NVIC_SetPriority(SysTick_IRQn, 2, 0);
109
110    /* USER CODE BEGIN MspInit 1 */
111
112    /* USER CODE END MspInit 1 */
113}
114
115 void HAL_ADC_MspInit(ADC_HandleTypeDef* hadc)
116{
117
118    GPIO_InitTypeDef GPIO_InitStruct;
119    if(hadc->Instance==ADC1)
120    {
121        /* USER CODE BEGIN ADC1_MspInit 0 */
122
123        /* USER CODE END ADC1_MspInit 0 */
124        /* Peripheral clock enable */
125        __ADC1_CLK_ENABLE();
126
127        /**ADC1 GPIO Configuration      IHM11M1
128          PC1      -----> ADC1_IN11
```

```
129     PC2      -----> ADC1_IN12
130     PC3      -----> ADC1_IN13
131     PA1      -----> ADC1_IN1
132     PA7      -----> ADC1_IN7
133     PB0      -----> ADC1_IN8
134     */
135     GPIO_InitStruct.Pin =
136         GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3;
137     GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
138     GPIO_InitStruct.Pull = GPIO_NOPULL;
139     HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
140
141     GPIO_InitStruct.Pin =
142         GPIO_PIN_1|GPIO_PIN_7;
143     GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
144     GPIO_InitStruct.Pull = GPIO_NOPULL;
145     HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
146
147     GPIO_InitStruct.Pin = GPIO_PIN_0;
148     GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
149     GPIO_InitStruct.Pull = GPIO_NOPULL;
150     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
151
152     /* System interrupt init*/
153     HAL_NVIC_SetPriority(ADC_IRQn, 0, 0);
154     HAL_NVIC_EnableIRQ(ADC_IRQn);
155     /* USER CODE BEGIN ADC1_MspInit 1 */
156 }
157
158 }
159
160 void HAL_ADC_MspDeInit(ADC_HandleTypeDef* hadc)
161 {
162 }
```

```

163     if(hadc->Instance==ADC1)
164     {
165     /* USER CODE BEGIN ADC1_MspDeInit_0 */
166
167     /* USER CODE END ADC1_MspDeInit_0 */
168     /* Peripheral clock disable */
169     __ADC1_CLK_DISABLE();
170
171     /**ADC1 GPIO Configuration      IHM11M1
172     PC1      -----> ADC1_IN11
173     PC2      -----> ADC1_IN12
174     PC3      -----> ADC1_IN13
175     PA1      -----> ADC1_IN1
176     PA7      -----> ADC1_IN7
177     PB0      -----> ADC1_IN8
178     */
179     HAL_GPIO_DeInit(GPIOC,
180                   GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
180
181     HAL_GPIO_DeInit(GPIOA,
182                   GPIO_PIN_1|GPIO_PIN_7);
182
183     HAL_GPIO_DeInit(GPIOB, GPIO_PIN_0);
184
185     /* Peripheral interrupt DeInit*/
186     HAL_NVIC_DisableIRQ(ADC_IRQn);
187
188     /* USER CODE BEGIN ADC1_MspDeInit_1 */
189
190     /* USER CODE END ADC1_MspDeInit_1 */
191 }
192
193 }
194
195 void HAL_TIM_Base_MspInit(TIM_HandleTypeDef*
196 htim_base)
196 {

```

```
197
198     GPIO_InitTypeDef GPIO_InitStruct;
199     if(htim_base->Instance==TIM1)
200     {
201         /* USER CODE BEGIN TIM1_MspInit_0 */
202
203         /* USER CODE END TIM1_MspInit_0 */
204         /* Peripheral clock enable */
205         __TIM1_CLK_ENABLE();
206
207         /* Enable GPIO Channels Clock */
208         __HAL_RCC_GPIOA_CLK_ENABLE();
209         __HAL_RCC_GPIOB_CLK_ENABLE();
210
211         /**TIM1 GPIO Configuration      IHM11M1
212         PA8      -----> TIM1_CH1
213         PA9      -----> TIM1_CH2
214         PA10     -----> TIM1_CH3
215         PB13     -----> TIM1_CH1N
216         PB14     -----> TIM1_CH2N
217         PB1      -----> TIM1_CH3N
218     */
219
220     GPIO_InitStruct.Pin =
221         GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10 ;
222     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
223     GPIO_InitStruct.Pull = GPIO_PULLDOWN;
224     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
225     GPIO_InitStruct.Alternate =
226         GPIO_AF1_TIM1;
227
228     GPIO_InitStruct.Pin =
229         GPIO_PIN_1|GPIO_PIN_13|GPIO_PIN_14 ;
230     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
231     GPIO_InitStruct.Pull = GPIO_PULLDOWN;
```

```
231     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
232     GPIO_InitStruct.Alternate =
233         GPIO_AF1_TIM1;
234     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
235 
236     /* System interrupt init*/
237     //
238     HAL_NVIC_SetPriority(TIM1_BRK_TIM9_IRQn, 0,
239     0);
240     //
241     HAL_NVIC_EnableIRQ(TIM1_BRK_TIM9_IRQn);
242     /* USER CODE BEGIN TIM1_MspInit_1 */
243 }
244 /* USER CODE END TIM1_MspInit_1 */
245 }
246 else if(htim_base->Instance==TIM3)
247 {
248     /* USER CODE BEGIN TIM3_MspInit_0 */
249 
250     /**TIM3 GPIO Configuration      IHM11M1
251     PB4      -----> TIM3_CH1
252     */
253 //     GPIO_InitStruct.Pin = GPIO_PIN_4;
254 //     GPIO_InitStruct.Mode =
255 //         GPIO_MODE_AF_PP;
256 //     GPIO_InitStruct.Pull = GPIO_NOPULL;
257 //     GPIO_InitStruct.Speed =
258 //         GPIO_SPEED_LOW;
259 //     GPIO_InitStruct.Alternate =
260 //         GPIO_AF2_TIM3;
261 //     HAL_GPIO_Init(GPIOB,
262 //     &GPIO_InitStruct);
263 }
```

```

260  /* USER CODE BEGIN TIM3_MspInit 1 */
261
262  /* USER CODE END TIM3_MspInit 1 */
263 }
264 else if(htim_base->Instance==TIM4)
265 {
266  /* USER CODE BEGIN TIM4_MspInit 0 */
267
268  /* USER CODE END TIM4_MspInit 0 */
269  /* Peripheral clock enable */
270  __TIM4_CLK_ENABLE();           //IHM11M1
271  /* System interrupt init*/
272  HAL_NVIC_SetPriority(TIM4_IRQn, 1, 0);
273  HAL_NVIC_EnableIRQ(TIM4_IRQn);
274  /* USER CODE BEGIN TIM4_MspInit 1 */
275
276  /* USER CODE END TIM4_MspInit 1 */
277 }
278
279 }
280
281 /**
282 * @brief Initializes the TIM PWM MSP.
283 * @param htim: pointer to a
284 *             TIM_HandleTypeDef structure that contains
285 *             the configuration
286 *             information for TIM module.
287 * @retval None
288 */
289 void HAL_TIM_PWM_MspInit(TIM_HandleTypeDef
290 *htim)
291 {
292     GPIO_InitTypeDef GPIO_InitStruct;
293
294     /*##-1- Enable peripherals and GPIO
295      Clocks #####*/

```

```

293     /* TIMx Peripheral clock enable */
294     __HAL_RCC_TIM1_CLK_ENABLE();
295
296     /* Enable GPIO Channels Clock */
297     __HAL_RCC_GPIOA_CLK_ENABLE();
298     __HAL_RCC_GPIOB_CLK_ENABLE();
299
300     /***TIM1 GPIO Configuration      IHM11M1
301     PA8      -----> TIM1_CH1
302     PA9      -----> TIM1_CH2
303     PA10     -----> TIM1_CH3
304     PB13     -----> TIM1_CH1N
305     PB14     -----> TIM1_CH2N
306     PB1      -----> TIM1_CH3N
307     */
308
309     GPIO_InitStruct.Pin =
310         GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10 ;
311     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
312     GPIO_InitStruct.Pull = GPIO_PULLDOWN;
313     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
314     GPIO_InitStruct.Alternate =
315         GPIO_AF1_TIM1;
316
317     GPIO_InitStruct.Pin =
318         GPIO_PIN_1|GPIO_PIN_13|GPIO_PIN_14 ;
319     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
320     GPIO_InitStruct.Pull = GPIO_PULLDOWN;
321     GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
322     GPIO_InitStruct.Alternate =
323         GPIO_AF1_TIM1;
324
325 }
```

```
326 void
327 HAL_TIMEx_HallSensor_MspInit(TIM_HandleTypeDef
328 * htimex_hallsensor)
329 {
330     GPIO_InitTypeDef GPIO_InitStruct;
331     if(htimex_hallsensor->Instance==TIM2)
332     {
333         /* USER CODE BEGIN TIM2_MspInit_0 */
334
335         /* USER CODE END TIM2_MspInit_0 */
336         /* Peripheral clock enable */
337         __TIM2_CLK_ENABLE();
338
339         /* Enable GPIO Channels Clock */
340         __HAL_RCC_GPIOA_CLK_ENABLE();
341         __HAL_RCC_GPIOB_CLK_ENABLE();
342
343         /**TIM2 GPIO Configuration      IHM11M1
344          PA15      -----> TIM2_CH1
345          PB10      -----> TIM2_CH3
346          PB3       -----> TIM2_CH2
347          */
348         GPIO_InitStruct.Pin = GPIO_PIN_15;
349         GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
350         GPIO_InitStruct.Pull = GPIO_NOPULL;
351         GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
352         GPIO_InitStruct.Alternate =
353             GPIO_AF1_TIM2;
354         HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
355
356         GPIO_InitStruct.Pin =
357             GPIO_PIN_10|GPIO_PIN_3;
358         GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
359         GPIO_InitStruct.Pull = GPIO_NOPULL;
360         GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
```

```

359     GPIO_InitStruct.Alternate =
360     GPIO_AF1_TIM2;
361
362     /* System interrupt init */
363     HAL_NVIC_SetPriority(TIM2_IRQn, 1, 0);
364     HAL_NVIC_EnableIRQ(TIM2_IRQn);
365
366
367     /* USER CODE BEGIN TIM2_MspInit 1 */
368
369     /* USER CODE END TIM2_MspInit 1 */
370 }
371
372 }
373
374 void
HAL_TIM_Base_MspDeInit(TIM_HandleTypeDef* htim_base)
375 {
376
377     if(htim_base->Instance==TIM1)
378     {
379         /* USER CODE BEGIN TIM1_MspDeInit 0 */
380
381         /* USER CODE END TIM1_MspDeInit 0 */
382         /* Peripheral clock disable */
383         __TIM1_CLK_DISABLE();
384
385         /**
386          **TIM1 GPIO Configuration      IHM11M1
387          PA8      -----> TIM1_CH1
388          PA9      -----> TIM1_CH2
389          PA10     -----> TIM1_CH3
390          PB13     -----> TIM1_CH1N
391          PB14     -----> TIM1_CH2N
392          PB1      -----> TIM1_CH3N
393     */

```

```
393
394     HAL_GPIO_DeInit(GPIOA,
395             GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10 );
396     HAL_GPIO_DeInit(GPIOB,
397             GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_1 );
398
399     /* Peripheral interrupt DeInit*/
400     HAL_NVIC_DisableIRQ(TIM1_BRK_TIM9 IRQn);
401
402     /* USER CODE BEGIN TIM1_MspDeInit 1 */
403
404     /* USER CODE END TIM1_MspDeInit 1 */
405 }
406 else if(htim_base->Instance==TIM3)
407 {
408     /* USER CODE BEGIN TIM3_MspDeInit 0 */
409
410     /* USER CODE END TIM3_MspDeInit 0 */
411     /* Peripheral clock disable */
412     __TIM3_CLK_DISABLE();
413
414     /**TIM3 GPIO Configuration
415      PB4      -----> TIM3_CH1
416      */
417     // HAL_GPIO_DeInit(GPIOB, GPIO_PIN_4);
418
419     /* USER CODE BEGIN TIM3_MspDeInit 1 */
420
421     /* USER CODE END TIM3_MspDeInit 1 */
422 }
423 else if(htim_base->Instance==TIM4)
424 {
425     /* USER CODE BEGIN TIM4_MspDeInit 0 */
426
427     /* USER CODE END TIM4_MspDeInit 0 */
428     /* Peripheral clock disable */
```

```
428     __TIM4_CLK_DISABLE();
429
430     /* Peripheral interrupt DeInit */
431     HAL_NVIC_DisableIRQ(TIM4_IRQn);
432
433     /* USER CODE BEGIN TIM4_MspDeInit_1 */
434
435     /* USER CODE END TIM4_MspDeInit_1 */
436 }
437
438 }
439
440 void
441 HAL_TIMEx_HallSensor_MspDeInit(TIM_HandleTypeDef* htimex_hallsensor)
442 {
443     if(htimex_hallsensor->Instance==TIM2)
444     {
445         /* USER CODE BEGIN TIM2_MspDeInit_0 */
446
447         /* USER CODE END TIM2_MspDeInit_0 */
448         /* Peripheral clock disable */
449         __TIM2_CLK_DISABLE();
450
451         /**TIM2 GPIO Configuration
452          PA15      -----> TIM2_CH1
453          PB10      -----> TIM2_CH3
454          PB3       -----> TIM2_CH2
455          */
456         HAL_GPIO_DeInit(GPIOA, GPIO_PIN_15);
457
458         HAL_GPIO_DeInit(GPIOB,
459                         GPIO_PIN_10|GPIO_PIN_3);
460
461     /* USER CODE BEGIN TIM2_MspDeInit_1 */
462 }
```

```
462     /* USER CODE END TIM2_MspDeInit 1 */
463 }
464 }
465 }
466
467 void HAL_UART_MspInit(UART_HandleTypeDef* huart)
468 {
469
470     GPIO_InitTypeDef GPIO_InitStruct;
471     if(huart->Instance==USART2)
472     {
473         /* USER CODE BEGIN USART2_MspInit 0 */
474
475         /* USER CODE END USART2_MspInit 0 */
476         /* Peripheral clock enable */
477         __USART2_CLK_ENABLE();
478
479         /**USART2 GPIO Configuration
480          PA2      -----> USART2_TX
481          PA3      -----> USART2_RX
482         */
483         GPIO_InitStruct.Pin =
484             GPIO_PIN_2|GPIO_PIN_3;
485         GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
486         GPIO_InitStruct.Pull = GPIO_NOPULL;
487         GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
488         GPIO_InitStruct.Alternate =
489             GPIO_AF7_USART2;
490         HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
491
492         /* System interrupt init*/
493         HAL_NVIC_SetPriority(USART2_IRQn, 3, 0);
494         HAL_NVIC_EnableIRQ(USART2_IRQn);
495         /* USER CODE BEGIN USART2_MspInit 1 */
496
497         /* USER CODE END USART2_MspInit 1 */
```

```
496     }
497
498 }
499
500 void HAL_UART_MspDeInit(UART_HandleTypeDef* huart)
501 {
502
503     if(huart->Instance==USART2)
504     {
505         /* USER CODE BEGIN USART2_MspDeInit_0 */
506
507         /* USER CODE END USART2_MspDeInit_0 */
508         /* Peripheral clock disable */
509         __USART2_CLK_DISABLE();
510
511         /**USART2 GPIO Configuration
512          PA2      -----> USART2_TX
513          PA3      -----> USART2_RX
514         */
515         HAL_GPIO_DeInit(GPIOA,
516                         GPIO_PIN_2|GPIO_PIN_3);
517
518         /* Peripheral interrupt DeInit*/
519         HAL_NVIC_DisableIRQ(USART2_IRQn);
520
521         /* USER CODE BEGIN USART2_MspDeInit_1 */
522
523     }
524
525 }
526
527 /* USER CODE BEGIN 1 */
528
529 /* USER CODE END 1 */
530
```

```
531  /**
532  * @}
533  */
534
535 /**
536 * @}
537 */
538
539 /***** (C) COPYRIGHT
      STMicroelectronics *****END OF FILE****/
```

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

stm32f4xx_it.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      stm32f4xx_it.h
4  * @date      27/04/2015 12:50:22
5  * @brief     This file contains the headers
6  *            of the interrupt handlers.
7  ****
8  * COPYRIGHT(c) 2015 STMicroelectronics
9  *
10 * Redistribution and use in source and
11 * binary forms, with or without modification,
12 * are permitted provided that the
13 * following conditions are met:
14 *   1. Redistributions of source code must
15 *      retain the above copyright notice,
16 *      this list of conditions and the
17 *      following disclaimer.
18 *   2. Redistributions in binary form must
```

reproduce the above copyright notice,

15 | * this list of conditions and the
following disclaimer in the documentation

16 | * and/or other materials provided
with the distribution.

17 | * 3. Neither the name of
STMicroelectronics nor the names of its
contributors

18 | * may be used to endorse or promote
products derived from this software

19 | * without specific prior written
permission.

20 | *

21 | * THIS SOFTWARE IS PROVIDED BY THE
COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

22 | * AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE

23 | * IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE

24 | * DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

25 | * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

26 | * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR

27 | * SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER

28 | * CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY,

29 | * OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

30 | * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

31 | *

32 | *

```
33 */  
34 /* Define to prevent recursive inclusion ---  
35 -----*/  
36 #ifndef __STM32F4xx_IT_H  
37 #define __STM32F4xx_IT_H  
38  
39 #ifdef __cplusplus  
40   extern "C" {  
41 #endif  
42  
43 /* Includes -----  
44 -----*/  
44 /* Exported types -----  
45 -----*/  
45 /* Exported constants -----  
46 -----*/  
46 /* Exported macro -----  
47 -----*/  
47 /* Exported functions -----  
48 -----*/  
48 void ADC_IRQHandler(void);  
49 void SysTick_Handler(void);  
50 void USART2_IRQHandler(void);  
51 void TIM4_IRQHandler(void);  
52 void EXTI15_10_IRQHandler(void);  
53 void TIM1_BRK_TIM9_IRQHandler(void);  
54  
55  
56 #ifdef __cplusplus  
57 }  
58 #endif  
59  
60 #endif /* __STM32F4xx_IT_H */  
61  
62 /***** (C) COPYRIGHT  
STMicroelectronics *****END OF FILE****/
```

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	STSPIN230

STSPIN230.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      STSPIN230.h
4  * @author    IPC
5  * @version   V0
6  * @date     04-April-2016
7  * @brief    This file provides a set of
functions to manage STSPIN230 driver
8
****

9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
15 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34    *
35 ****
36 */
37 /* Define to prevent recursive inclusion ---
38 */
39 #ifndef __STSPIN230_H
40 #define __STSPIN230_H
41
42 #include "stdint.h"
43 #include "MC_Common.h"
44
45
46
47 extern void
    STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D(void);
48 extern void
    STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E(void);
49 extern void
    STSPIN230_EnableInput_CH1_D_CH2_E_CH3_E(void);
50 extern void
    STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D(void);

51 extern void
    STSPIN230_Start_PWM_driving(void);
52 extern void
    STSPIN230_Stop_PWM_driving(void);
53 extern void
    STSPIN230_HF_TIMx_SetDutyCycle_CH1(uint16_t);
54 extern void
    STSPIN230_HF_TIMx_SetDutyCycle_CH2(uint16_t);
55 extern void
    STSPIN230_HF_TIMx_SetDutyCycle_CH3(uint16_t);
56 extern void
    STSPIN230_Current_Reference_Start(void);
```

```

57    extern void
      STSPIN230_Current_Reference_Stop(void);
58    extern void
      STSPIN230_Current_Reference_Setvalue(uint16_t)
      ;
59    extern void
      STSPIN230_START_Ref_Generation(void);
60    extern void
      STSPIN230_STOP_Ref_Generation(void);
61    extern void
      STSPIN230_Set_Ref_Generation(uint16_t);
62
63    void EnableInput_CH1_E_CH2_E_CH3_D(void);
64    void EnableInput_CH1_E_CH2_D_CH3_E(void);
65    void EnableInput_CH1_D_CH2_E_CH3_E(void);
66    void DisableInput_CH1_D_CH2_D_CH3_D(void);
67    void Start_PWM_driving(void);
68    void Stop_PWM_driving(void);
69    void HF_TIMx_SetDutyCycle_CH1(uint16_t);
70    void HF_TIMx_SetDutyCycle_CH2(uint16_t);
71    void HF_TIMx_SetDutyCycle_CH3(uint16_t);
72    void Current_Reference_Start(void);
73    void Current_Reference_Stop(void);
74    void Current_Reference_Setvalue(uint16_t);
75
76
77 /** @addtogroup DRIVERS      DRIVERS
78 * @brief Driver Layer
79 * @{
80 */
81
82 /** @addtogroup BSP      BSP
83 * @brief BSP Layer
84 * @{
85 */
86
87 /** @addtogroup COMPONENTS    COMPONENTS

```

```
88 * @brief Components
89 * @{
90 */
91
92 /** @addtogroup STSPIN230      STSPIN230
93 * @brief STSPIN230 driver section
94 * @{
95 */
96
97
98
99 /**
100 * @}
101 */
102
103 #endif
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	UART_serial_com	Src

UART_UI.c

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      UART_UI.c
4  * @author    IPC
5  * @version   V0
6  * @date     10/07/2016
7  * @brief    This file provides a set of
functions needed to manage the UART com.
8
****

9  * @attention
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
STMicroelectronics</center></h2>
12 *
13 * Redistribution and use in source and
binary forms, with or without modification,
14 * are permitted provided that the following
conditions are met:
15 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34    *
35 ****
36    */
37
38 /* Includes -----
   -----*/
39 #include "UART_UI.h"
40
41 #ifdef UART_COMM
42
43 extern SIXSTEP_Base_InitTypeDef
  SIXSTEP_parameters;      /* !< Main SixStep
  structure*/
44 extern SIXSTEP_PI_PARAM_InitTypeDef_t
  PI_parameters;          /* !< SixStep PI regulator
  structure*/
45 extern uint8_t Enable_start_button;
46 extern void CMD_Parser(char*
  pCommandString);
47 extern void
  MC_Set_PI_param(SIXSTEP_PI_PARAM_InitTypeDef_t
  *);
48
49 static uint16_t Uart_cmd_flag = 0;
50 const CMD_T CmdTable[] = {
51 {"STARTM",  CMD_STARTM},
52 {"STOPMT",  CMD_STOPMT},
53 {"DIRECT",  CMD_DIRECTION},
54 {"SETSPD",  CMD_SETSPD},
55 {"STATUS",  CMD_STATUS},
56 {"GETSPD",  CMD_GETSPD},
57 {"POTENZ",  CMD_POTENZ},
58 {"INIREF",  CMD_INIREF},
59 {"POLESP",  CMD_POLESP},
60 {"ACCELE",  CMD_ACCELE},
```

```

61 {"KP-PRM", CMD_KP_PRM},
62 {"KI-PRM", CMD_KI_PRM},
63 {"HELP", CMD_HELP},
64 {"", NULL}
65 };
66
67 uint8_t BUFF_RCV = RXBUFFERSIZE;
68 uint8_t aRxBuffer[8] = {'0'}; /* !<
   Buffer used for reception */
69 uint8_t row0TxBuffer[] = "\033[2J"; /* !<
   Buffer used for transmission */
70 uint8_t row1TxBuffer[] =
  ****
  ****\n\r";
71 uint8_t row2TxBuffer[] = "*          X-
  NUCLEO-IHM11M1 - 6-Step Motor Control
  Expansion board      *\n\r";
72 uint8_t row3TxBuffer[] =
  ****
  ****\n\r";
73 uint8_t row4TxBuffer[] = " List of
  commands:\n\r\n\r";
74 uint8_t row5TxBuffer[] = " <STARTM> -- 
  Start Motor\n\r";
75 uint8_t row6TxBuffer[] = " <STOPMT> -- Stop
  Motor\n\r";
76 uint8_t row7aTxBuffer[] = " <DIRECT> -- Set
  the Motor direction CW or CCW \n\r";
77 uint8_t row7bTxBuffer[] = " <SETSPD> -- Set
  the Motor Speed\n\r";
78 uint8_t row7cTxBuffer[] = " <GETSPD> -- Get
  the Motor Speed\n\r";
79 uint8_t row7dTxBuffer[] = " <STATUS> -- Get
  the Status of system \n\r";
80 uint8_t row7eTxBuffer[] = " <POTENZ> -- 
  Enable/Disable the potentiometer\n\r";
81 uint8_t row7fTxBuffer[] = " <HELP>    --

```

```

    Show the help menu \n\r";
82| uint8_t rowVTxBuffer[] = " >>> Insert the
      value: ";
83| uint8_t rowVTBxBuffer[] = " >>> ENABLE <1>
      DISABLE <0>; ";
84| uint8_t rowV1TBxBuffer[] = " >>> CW <0> CCW
      <1>; ";
85| uint8_t rowMxBuffer[] = " >>> START MOTOR
      COMMAND RECEIVED ! <<< \n\r >>> ";
86| uint8_t rowSxBuffer[] = " >>> STOP MOTOR
      COMMAND RECEIVED ! <<< \n\r >>> ";
87| uint8_t rowETxBuffer[] = " >>> ERROR -
      PLEASE TYPE AGAIN ! <<< \n\r >>> ";
88| uint8_t row8TxBuffer[] = "
***** MAIN MOTOR PARAMETERS
***** \n\r",
89| uint8_t row9TxBuffer[] = " <INIREF> --
      Start-up current reference (0-4095) \n\r";
90| uint8_t row10TxBuffer[] = " <ACCELE> --
      Motor acceleration value during startup \n\r";
91| uint8_t row11TxBuffer[] = " <POLESP> -- Set
      the Motor pole pairs \n\r";
92| uint8_t row14TxBuffer[] = " ***** --
      PI REGULATOR PARAMETERS - SPEED LOOP
      *****\n\r";
93| uint8_t row15TxBuffer[] = " <KP-PRM> -- Set
      the PI proportional term \n\r";
94| uint8_t row16TxBuffer[] = " <KI-PRM> -- Set
      the PI integral term \n\r\n\r >";
95| uint8_t rowLxBuffer[] = " --- OK ---\n\r >";
96| extern uint8_t UART_FLAG_RECEIVE;
97| extern uint8_t UART_FLAG_POTENZ;
98|
99|
100| /**
101| * @addtogroup MIDDLEWARES      MIDDLEWARES
102| * @brief   Middlewares Layer
103| * @{

```

```

103    */
104
105
106 /** @addtogroup UART_UI  UART_UI
107  * @brief Serial communication through PC
108  * serial terminal
109  * @{
110
111
112 /** @defgroup UART_Communication_Task
113  * @{
114  * @brief UART start receive function
115  */
116 void UART_Communication_Task()
117 {
118     if(HAL_UART_Receive_IT(&huart2, (uint8_t
119     *)aRxBuffer, 10) != HAL_OK)
120     {
121         huart2.State = HAL_UART_STATE_READY;
122     }
123     else
124     {
125     /**
126     * @}
127     */
128
129 /**
130  * @brief UART User Interface init
131 */
132 void MC_UI_INIT()
133 {
134     HAL_UART_Transmit(&huart2, (uint8_t
135     *)row0TxBuffer, (COUNTOF(row0TxBuffer) -
136     1),5000);
137     while (HAL_UART_GetState(&huart2) !=
138     HAL_UART_STATE_READY)

```

```
134     {
135     }
136     HAL_UART_Transmit(&huart2, (uint8_t
137     *)row1TxBuffer, (COUNTOF(row1TxBuffer) -
138     1),5000);
139     while (HAL_UART_GetState(&huart2) !=
140     HAL_UART_STATE_READY)
141     {
142     }
143     }
144     HAL_UART_Transmit(&huart2, (uint8_t
145     *)row2TxBuffer, (COUNTOF(row2TxBuffer) -
146     1),5000);
147     while (HAL_UART_GetState(&huart2) !=
148     HAL_UART_STATE_READY)
149     {
150     }
151     }
152     HAL_UART_Transmit(&huart2, (uint8_t
153     *)row4TxBuffer, (COUNTOF(row4TxBuffer) -
154     1),5000);
155     while (HAL_UART_GetState(&huart2) !=
```

```
156     HAL_UART_Transmit(&huart2, (uint8_t
 * )row6TxBuffer, (COUNTOF(row6TxBuffer) -
 1),5000);
157     while (HAL_UART_GetState(&huart2) !=
 HAL_UART_STATE_READY)
158     {
159     }
160     HAL_UART_Transmit(&huart2, (uint8_t
 * )row7aTxBuffer, (COUNTOF(row7aTxBuffer) -
 1),5000);
161     while (HAL_UART_GetState(&huart2) !=
 HAL_UART_STATE_READY)
162     {
163     }
164     HAL_UART_Transmit(&huart2, (uint8_t
 * )row7TxBuffer, (COUNTOF(row7TxBuffer) -
 1),5000);
165     while (HAL_UART_GetState(&huart2) !=
 HAL_UART_STATE_READY)
166     {
167     }
168     HAL_UART_Transmit(&huart2, (uint8_t
 * )row7bTxBuffer, (COUNTOF(row7bTxBuffer) -
 1),5000);
169     while (HAL_UART_GetState(&huart2) !=
 HAL_UART_STATE_READY)
170     {
171     }
172     HAL_UART_Transmit(&huart2, (uint8_t
 * )row7cTxBuffer, (COUNTOF(row7cTxBuffer) -
 1),5000);
173     while (HAL_UART_GetState(&huart2) !=
 HAL_UART_STATE_READY)
174     {
175     }
176     HAL_UART_Transmit(&huart2, (uint8_t
 * )row71TxBuffer, (COUNTOF(row71TxBuffer) -
```

```
    1),5000);
177    while (HAL_UART_GetState(&huart2) !=  
    HAL_UART_STATE_READY)  
178    {  
179    }  
180    HAL_UART_Transmit(&huart2, (uint8_t  
    *)row71bTxBuffer, (COUNTOF(row71bTxBuffer) -  
    1),5000);  
181    while (HAL_UART_GetState(&huart2) !=  
    HAL_UART_STATE_READY)  
182    {  
183    }  
184    HAL_UART_Transmit(&huart2, (uint8_t  
    *)row8TxBuffer, (COUNTOF(row8TxBuffer) -  
    1),5000);  
185    while (HAL_UART_GetState(&huart2) !=  
    HAL_UART_STATE_READY)  
186    {  
187    }  
188    HAL_UART_Transmit(&huart2, (uint8_t  
    *)row9TxBuffer, (COUNTOF(row9TxBuffer) -  
    1),5000);  
189    while (HAL_UART_GetState(&huart2) !=  
    HAL_UART_STATE_READY)  
190    {  
191    }  
192    HAL_UART_Transmit(&huart2, (uint8_t  
    *)row10TxBuffer, (COUNTOF(row10TxBuffer) -  
    1),5000);  
193    while (HAL_UART_GetState(&huart2) !=  
    HAL_UART_STATE_READY)  
194    {  
195    }  
196    HAL_UART_Transmit(&huart2, (uint8_t  
    *)row11TxBuffer, (COUNTOF(row11TxBuffer) -  
    1),5000);  
197    while (HAL_UART_GetState(&huart2) !=
```

```

        HAL_UART_STATE_READY)
198    {
199    }
200    HAL_UART_Transmit(&huart2, (uint8_t
 * )row14TxBuffer, (COUNTOF(row14TxBuffer) -
1),5000);
201    while (HAL_UART_GetState(&huart2) !=
HAL_UART_STATE_READY)
202    {
203    }
204    HAL_UART_Transmit(&huart2, (uint8_t
 * )row15TxBuffer, (COUNTOF(row15TxBuffer) -
1),5000);
205    while (HAL_UART_GetState(&huart2) !=
HAL_UART_STATE_READY)
206    {
207    }
208    HAL_UART_Transmit(&huart2, (uint8_t
 * )row16TxBuffer, (COUNTOF(row16TxBuffer) -
1),5000);
209    while (HAL_UART_GetState(&huart2) !=
HAL_UART_STATE_READY)
210    {
211    }
212    UART_Communication_Task();
213    SIXSTEP_parameters.Button_ready = TRUE;
214}
215 /**
216 * @}
217 */
218
219
220 /** @defgroup UART_num_decode
UART_num_decode
221 * @{
222 * @brief UART Value decoding function
223 */

```

```

224 uint32_t UART_num_decode()
225 {
226     static char Value_Buffer[RXBUFFERSIZE];
227
228     for(uint8_t i=0;i<BUFF_RCV;i++)
229     {
230         Value_Buffer[i] = aRxBuffer[i];
231     }
232     return(atoi(Value_Buffer));
233 }
234 /**
235 * @}
236 */
237
238 /**
239 * @{
240 * @brief UART Main function
241 */
242 void UART_Set_Value()
243 {
244     if(Get_UART_Data() == 266 && (huart2.State
245     != HAL_UART_STATE_BUSY_TX && huart2.State !=
246     HAL_UART_STATE_BUSY_RX))
247     {
248         if(Uart_cmd_flag == 0)
249         {
250             CMD_Parser((char*)aRxBuffer);
251         }
252         else
253         {
254             SIXSTEP_parameters.Uart_value_to_set =
255             UART_num_decode();
256
257             switch(SIXSTEP_parameters.Uart_cmd_to_set)
258             {

```

```

256     case SETSPD_CMD: /*!< Set the new
257     speed value command received */
258         PI_parameters.Reference =
259             SIXSTEP_parameters.Uart_value_to_set;
260         SIXSTEP_parameters.Ramp_Start = 1;
261         BUFF_RCV = RXBUFFERSIZE;
262         Uart_cmd_flag = 0;
263         HAL_UART_Transmit(&huart2, (uint8_t
264             *)rowLxBuffer, (COUNTOF(rowLxBuffer) -
265             1), 5000);
266         UART_FLAG_RECEIVE = TRUE;
267         huart2.State = HAL_UART_STATE_READY;
268         break;
269     case INIREF_CMD: /*!< Set the new
270     STARUP_CURRENT_REFERENCE value command
271     received */
272         SIXSTEP_parameters.Ireference =
273             SIXSTEP_parameters.Uart_value_to_set;
274         SIXSTEP_parameters.pulse_value=
275             SIXSTEP_parameters.Uart_value_to_set;
276         BUFF_RCV = RXBUFFERSIZE;
277         Uart_cmd_flag = 0;
278         huart2.State = HAL_UART_STATE_READY;
279         HAL_UART_Transmit(&huart2, (uint8_t
280             *)rowLxBuffer, (COUNTOF(rowLxBuffer) -
281             1), 5000);
282         UART_FLAG_RECEIVE = TRUE;
283         break;
284     case POLESP_CMD: /*!< Set the Pole
285     Pairs value command received */
286         SIXSTEP_parameters.NUMPOLESPAIRS =
287             SIXSTEP_parameters.Uart_value_to_set;
288         BUFF_RCV = RXBUFFERSIZE;
289         Uart_cmd_flag = 0;
290         huart2.State = HAL_UART_STATE_READY;
291         HAL_UART_Transmit(&huart2, (uint8_t
292             *)rowLxBuffer, (COUNTOF(rowLxBuffer) -

```

```
    1),5000);
280        UART_FLAG_RECEIVE = TRUE;
281        break;
282    case ACCELE_CMD: /*!< Set the
Accelleration for Start-up of the motor
command received */
283        SIXSTEP_parameters.ACCEL =
SIXSTEP_parameters.Uart_value_to_set;
284        BUFF_RCV = RXBUFFERSIZE;
285        Uart_cmd_flag = 0;
286        huart2.State = HAL_UART_STATE_READY;
287        HAL_UART_Transmit(&huart2, (uint8_t
*)rowLxBuffer, (COUNTOF(rowLxBuffer) -
1),5000);
288        UART_FLAG_RECEIVE = TRUE;
289        break;
290    case DIRECT_CMD: /*!< Set the motor
direction */
291        SIXSTEP_parameters.CW_CCW =
SIXSTEP_parameters.Uart_value_to_set;
292        MC_Set_PI_param(&PI_parameters);
293        BUFF_RCV = RXBUFFERSIZE;
294        Uart_cmd_flag = 0;
295        huart2.State = HAL_UART_STATE_READY;
296        HAL_UART_Transmit(&huart2, (uint8_t
*)rowLxBuffer, (COUNTOF(rowLxBuffer) -
1),5000);
297        UART_FLAG_RECEIVE = TRUE;
298        break;
299    case KP_PRM_CMD: /*!< Set the KP PI
param command received */
300        PI_parameters.Kp_Gain =
SIXSTEP_parameters.Uart_value_to_set;
301        BUFF_RCV = RXBUFFERSIZE;
302        Uart_cmd_flag = 0;
303        huart2.State = HAL_UART_STATE_READY;
304        HAL_UART_Transmit(&huart2, (uint8_t
```

```

        *)rowLxBuffer, (COUNTOF(rowLxBuffer) -
1),5000);
305         UART_FLAG_RECEIVE = TRUE;
306         break;
307         case KI_PRM_CMD: /*!< Set the KI PI
param command received */
308         PI_parameters.Ki_Gain =
SIXSTEP_parameters.Uart_value_to_set;
309         BUFF_RCV = RXBUFFERSIZE;
310         Uart_cmd_flag = 0;
311         huart2.State = HAL_UART_STATE_READY;
312         HAL_UART_Transmit(&huart2, (uint8_t
*)rowLxBuffer, (COUNTOF(rowLxBuffer) -
1),5000);
313         UART_FLAG_RECEIVE = TRUE;
314         break;
315         case POTENZ_CMD: /*!< Enable
Potentiometer command received */
316         BUFF_RCV = RXBUFFERSIZE;
317         Uart_cmd_flag = 0;
318         huart2.State = HAL_UART_STATE_READY;
319         HAL_UART_Transmit(&huart2, (uint8_t
*)rowLxBuffer, (COUNTOF(rowLxBuffer) -
1),5000);
320         UART_FLAG_RECEIVE = TRUE;
321         SIXSTEP_parameters.Potentiometer =
SIXSTEP_parameters.Uart_value_to_set;
322         break;
323         } /* switch case */
324     } /* else */
325 } /* if */
326 }
327 /**
328 * @}
329 */
330
331

```

```

332 /**
333 * @{
334 * @brief UART Transmit standard message
335 */
336 void CMD_MSG()
337 {
338     HAL_UART_Transmit(&huart2, (uint8_t
339     *)rowVTxBuffer, (COUNTOF(rowVTxBuffer) -
340     1),5000);
341     BUFF_RCV = RXBUFFERSIZE - 1;
342     Uart_cmd_flag = 1;
343     UART_FLAG_RECEIVE = TRUE;
344     huart2.State = HAL_UART_STATE_READY;
345 }
346 /**
347 * @}
348 */
349 /**
350 * @brief UART Transmit standard message
351 * for input data
352 */
353 void CMD_MSG_BOOL()
354 {
355     huart2.State = HAL_UART_STATE_READY;
356     HAL_UART_Transmit(&huart2, (uint8_t
357     *)rowVTBxBuffer, (COUNTOF(rowVTBxBuffer) -
358     1),5000);
359     BUFF_RCV = RXBUFFERSIZE - 1;
360     Uart_cmd_flag = 1;
361     UART_FLAG_RECEIVE = TRUE;
362 }
363 /**
364 * @}
365 */

```

```

364  /** @defgroup CMD_MSG_BOOL_DIRECT
365   * @{
366   * @brief UART Transmit CW or CCW message
367   * for input data
368   */
369 void CMD_MSG_BOOL_DIRECT()
370 {
371     huart2.State = HAL_UART_STATE_READY;
372     HAL_UART_Transmit(&huart2, (uint8_t
373     *)rowV1TBxBuffer, (COUNTOF(rowV1TBxBuffer) -
374     1),5000);
375     BUFF_RCV = RXBUFFERSIZE - 1;
376     Uart_cmd_flag = 1;
377     UART_FLAG_RECEIVE = TRUE;
378 }
379 /**
380  * @}
381  */
382 /**
383  * @defgroup CMD_STARTM      CMD_STARTM
384  * @{
385  * @brief UART Transmit Start motor
386  * message
387  */
388 void CMD_STARTM()
389 {
390     /*!< Start Motor command received */
391     huart2.State = HAL_UART_STATE_READY;
392     HAL_UART_Transmit(&huart2, (uint8_t
393     *)rowMxBuffer, (COUNTOF(rowMxBuffer) -
394     1),5000);
395     MC_StartMotor();
396     Enable_start_button = FALSE;
397     UART_FLAG_RECEIVE = TRUE;
398 }
399 /**

```

```

394     * @}
395     */
396
397 /** @defgroup CMD_STOPMT      CMD_STOPMT
398 *  @{
399   * @brief UART Transmit Stop motor
400   message
401 */
402 void CMD_STOPMT()
403 {
404   /*!< Stop Motor command received */
405   huart2.State = HAL_UART_STATE_READY;
406   HAL_UART_Transmit(&huart2, (uint8_t
407 *)rowSxBuffer, (COUNTOF(rowSxBuffer) -
408 1),5000);
409   MC_StopMotor();
410   Enable_start_button = FALSE;
411   UART_FLAG_RECEIVE = TRUE;
412 }
413
414 /** @defgroup CMD_SETSPD      CMD_SETSPD
415 *  @{
416   * @brief UART Change the motor speed
417 */
418 void CMD_SETSPD()
419 {
420   /*!< Set the new speed value command
421   received */
422   CMD_MSG();
423   SIXSTEP_parameters.Uart_cmd_to_set =
424   SETSPD_CMD;
425 }
426 /**
427 * @}

```

```

426     */
427
428 /** @defgroup CMD_GETSPD      CMD_GETSPD
429 * @{
430   * @brief UART Get the motor speed
431 */
432 void CMD_GETSPD()
433 {
434 /*!< Get Mechanical Motor Speed command
435 received */
436     static char strLineMessage[40];
437
438     huart2.State = HAL_UART_STATE_READY;
439     sprintf(strLineMessage, "-- The Motor
Speed is: %d RPM -- \r\n >",
440             SIXSTEP_parameters.speed_fdbk_filtered);
441     HAL_UART_Transmit(&huart2, (uint8_t*)
442                         strLineMessage, sizeof(strLineMessage), 5000);
443     UART_FLAG_RECEIVE = TRUE;
444 }
445 /**
446 * @{
447 */
448
449 /** @defgroup CMD_POTENZ      CMD_POTENZ
450 * @{
451   * @brief UART Enable the potentiometer
452 */
453 void CMD_POTENZ()
454 {
455 /*!< Enable Potentiometer command received
456 */
457     CMD_MSG_BOOL();
458     SIXSTEP_parameters.Uart_cmd_to_set =
459         POTENZ_CMD;
460 }
461 /**

```

```
457     * @}
458     */
459
460 /** @defgroup CMD_HELP      CMD_HELP
461  *  @{
462  *  @brief UART Help command
463  */
464 void CMD_HELP()
465 {
466 /*!< Help command received */
467     huart2.State = HAL_UART_STATE_READY;
468     MC_UI_INIT();
469     UART_FLAG_RECEIVE = TRUE;
470 }
471 /**
472  * @}
473 */
474
475
476 /** @defgroup CMD_INIREF    CMD_INIREF
477  *  @{
478  *  @brief UART Set the current reference
479  */
480 void CMD_INIREF()
481 {
482 /*!< Set the new STARUP_CURRENT_REFERENCE
483 value command received */
484     CMD_MSG();
485     SIXSTEP_parameters.Uart_cmd_to_set =
486     INIREF_CMD;
487 }
488 /**
489  * @}
490 */
491 /** @defgroup CMD_POLESP    CMD_POLESP
```

```
492     * @{
493     * @brief UART Set the motor poles pairs
494     */
495 void CMD_POLESP()
496 {
497     /*!< Set the Pole Pairs value command
498      received */
499     CMD_MSG();
500     SIXSTEP_parameters.Uart_cmd_to_set =
501         POLESP_CMD;
502 }
503 /**
504 * @}
505 */
506 /**
507 * @defgroup CMD_ACCELE    CMD_ACCELE
508 * @{
509     * @brief UART Set the accelleration of
510      the motor at start-up
511 */
512 void CMD_ACCELE()
513 {
514     /*!< Set the Accelleration for Start-up of
515      the motor command received */
516     CMD_MSG();
517     SIXSTEP_parameters.Uart_cmd_to_set =
518         ACCELE_CMD;
519 }
520 /**
521 * @}
522 */
523 /**
524 * @defgroup CMD_DIRECTION  CMD_DIRECTION
525 * @{
526     * @brief UART Set the motor direction
527 */
528 void CMD_DIRECTION()
```

```
524 {
525     /*!< Enable the DEMAG dynamic control
526     command received */
527     CMD_MSG_BOOL_DIRECT();
528     SIXSTEP_parameters.Uart_cmd_to_set =
529     DIRECT_CMD;
530 /**
531 * @}
532 */
533 /** @defgroup CMD_KP_PRM      CMD_KP_PRM
534 * @{
535 * @brief UART Set the KP PI param
536 */
537 void CMD_KP_PRM()
538 {
539     /*!< Set the KP PI param command received
540 */
541     CMD_MSG();
542     SIXSTEP_parameters.Uart_cmd_to_set =
543     KP_PRM_CMD;
544 /**
545 * @}
546 */
547 /** @defgroup CMD_KI_PRM      CMD_KI_PRM
548 * @{
549 * @brief UART Set the KI PI param
550 */
551 void CMD_KI_PRM()
552 {
553     /*!< Set the KI PI param command received
554 */
555     CMD_MSG();
556     SIXSTEP_parameters.Uart_cmd_to_set =
```

```
    KI_PRM_CMD;
556 }
557 /**
558 * @}
559 */
560
561 /** @defgroup CMD_STATUS      CMD_STATUS
562 * @{
563 * @brief UART View the STATUS
564 */
565 void CMD_STATUS()
566 {
567     static char strLineMessage1[30];
568     static char strLineMessage2[30];
569     static char strLineMessage3[30];
570     static char strLineMessage4[30];
571     static char strLineMessage5[30];
572     static char strLineMessage6[30];
573     static char strLineMessage7[30];
574     static char strLineMessage8[30];
575
576     huart2.State = HAL_UART_STATE_READY;
577     switch (SIXSTEP_parameters.STATUS)
578     {
579         case STARTUP:
580             sprintf(strLineMessage1, " --
The status is: STARTUP --\r\n >");
581             HAL_UART_Transmit(&huart2,
582                               (uint8_t*)
583                               strLineMessage1, sizeof(strLineMessage1), 5000);
584
585         break;
586         case VALIDATION:
587             sprintf(strLineMessage2, " --
The status is: VALIDATION --\r\n >");
588             HAL_UART_Transmit(&huart2,
589                               (uint8_t*)
```

```
strLineMessage2, sizeof(strLineMessage2), 5000);

586         break;
587     case START:
588         sprintf(strLineMessage3, " --
The status is: START --\r\n >");
589         HAL_UART_Transmit(&huart2,
(uint8_t*)
strLineMessage3, sizeof(strLineMessage3), 5000);

590         break;
591     case STOP:
592         sprintf(strLineMessage4, " --
The status is: STOP --\r\n >");
593         HAL_UART_Transmit(&huart2,
(uint8_t*)
strLineMessage4, sizeof(strLineMessage4), 5000);

594         break;
595     case RUN:
596         sprintf(strLineMessage5, " --
The status is: RUN --\r\n >");
597         HAL_UART_Transmit(&huart2,
(uint8_t*)
strLineMessage5, sizeof(strLineMessage5), 5000);

598         break;
599     case ALIGNMENT:
600         sprintf(strLineMessage6, " --
The status is: ALIGNMENT --\r\n >");
601         HAL_UART_Transmit(&huart2,
(uint8_t*)
strLineMessage6, sizeof(strLineMessage6), 5000);

602         break;
603     case SPEEDFBKERROR:
604         sprintf(strLineMessage7, " --
```

```

        The status is: SPEEDFBKERROR --\r\n >");
605             HAL_UART_Transmit(&huart2,
        (uint8_t*)
        strLineMessage7, sizeof(strLineMessage7), 5000);

606         break;
607     default:
608         sprintf(strLineMessage8, " --
The status is: IDLE --\r\n >");
609             HAL_UART_Transmit(&huart2,
        (uint8_t*)
        strLineMessage8, sizeof(strLineMessage8), 5000);

610         break;
611     }
612     UART_FLAG_RECEIVE = TRUE;
613 }
614 /**
615 * @}
616 */
617
618 /** @defgroup CMD_Parser      CMD_Parser
619 * @{
620 * @brief UART String parser
621 */
622 void CMD_Parser(char* pCommandString)
623 {
624     static uint8_t CmdListIndex;
625     char sCmd[16];
626     #define TOKEN2 "\n"

628     strcpy(sCmd, pCommandString);
629     strtok (sCmd, TOKEN);
630     strtok (sCmd, TOKEN2);

631
632     /* Command Callback identification */
633

```

```
    for(CmdListIndex=0;CmdTable[CmdListIndex].pCmd
        Func!=NULL; CmdListIndex++) {
634        if
        (strcmp(CmdTable[CmdListIndex].name, sCmd) ==
         0 )
635        {
636            break;
637        }
638    }
639    if ( CmdListIndex < CMD_NUM ){
640        // CMD OK --> extract parameters
641        /* Check for valid callback */
642
643        if(CmdTable[CmdListIndex].pCmdFunc!=NULL) {
644            CmdTable[CmdListIndex].pCmdFunc();
645        }
646        else{
647            huart2.State =
648            HAL_UART_STATE_READY;
649            HAL_UART_Transmit(&huart2,
650                (uint8_t *)rowETxBUFFER,
651                (COUNTOF(rowETxBUFFER) - 1),5000);
652            BUFF_RCV = RXBUFFERSIZE;
653            Uart_cmd_flag = 0;
654            UART_FLAG_RECEIVE = TRUE;
655            return;
656        }
657    /**
658     * @}
659     */
660
661 /**

```

```
662     * @}  end UART_UI
663     */
664
665 /**
666     * @}  end MIDDLEWARES
667     */
668 #endif
```



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	X-NUCLEO-IHM11M1	

X-NUCLEO-IHM11M1.h

Go to the documentation of this file.

```
1  /**
2  ****
3  * @file      X-NUCLEO-IHM11M1.h
4  * @author    IPC
5  * @version   V0
6  * @date     10/07/2016
7  * @brief    This file provides the set of
8  *           functions to manage the X-Nucleo board
9  *
10 *
11 * <h2><center>&copy; COPYRIGHT(c) 2015
12 * STMicroelectronics</center></h2>
13 *
14 * Redistribution and use in source and
15 * binary forms, with or without modification,
16 * are permitted provided that the following
17 * conditions are met:
18 *
19 *   1. Redistributions of source code must
```

```
    retain the above copyright notice,
16| *      this list of conditions and the
   following disclaimer.
17| *      2. Redistributions in binary form must
   reproduce the above copyright notice,
18| *      this list of conditions and the
   following disclaimer in the documentation
19| *      and/or other materials provided with
   the distribution.
20| *      3. Neither the name of
   STMicroelectronics nor the names of its
   contributors
21| *      may be used to endorse or promote
   products derived from this software
22| *      without specific prior written
   permission.
23| *
24| * THIS SOFTWARE IS PROVIDED BY THE
   COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
25| * AND ANY EXPRESS OR IMPLIED WARRANTIES,
   INCLUDING, BUT NOT LIMITED TO, THE
26| * IMPLIED WARRANTIES OF MERCHANTABILITY AND
   FITNESS FOR A PARTICULAR PURPOSE ARE
27| * DISCLAIMED. IN NO EVENT SHALL THE
   COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
28| * FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
29| * DAMAGES (INCLUDING, BUT NOT LIMITED TO,
   PROCUREMENT OF SUBSTITUTE GOODS OR
30| * SERVICES; LOSS OF USE, DATA, OR PROFITS;
   OR BUSINESS INTERRUPTION) HOWEVER
31| * CAUSED AND ON ANY THEORY OF LIABILITY,
   WHETHER IN CONTRACT, STRICT LIABILITY,
32| * OR TORT (INCLUDING NEGLIGENCE OR
   OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
33| * OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.
```

```
34    *
35 ****
36    */
37
38 /* Includes -----
   ----- */
39
40 #include "STSPIN230.h"
41
42 void
43     STSPIN230_EnableInput_CH1_E_CH2_E_CH3_D(void);
44 void
45     STSPIN230_EnableInput_CH1_E_CH2_D_CH3_E(void);
46 void
47     STSPIN230_DisableInput_CH1_D_CH2_D_CH3_D(void);

48 void
49     STSPIN230_Start_PWM_driving(void);
50 void
51     STSPIN230_Stop_PWM_driving(void);
52 void
53     STSPIN230_HF_TIMx_SetDutyCycle_CH1(uint16_t);
54 void
55     STSPIN230_HF_TIMx_SetDutyCycle_CH2(uint16_t);
56 void
57     STSPIN230_HF_TIMx_SetDutyCycle_CH3(uint16_t);
58 void
59     STSPIN230_Current_Reference_Start(void);
60 void
61     STSPIN230_Current_Reference_Stop(void);
62 void
63     STSPIN230_Current_Reference_Setvalue(uint16_t);
```

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares

Middlewares Directory Reference

Directories

directory **ST**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST		
<h2>ST Directory Reference</h2>			

Directories

directory **MC_6Step_Lib**

directory **UART_serial_com**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Middlewares	ST	MC_6Step_Lib	

MC_6Step_Lib Directory Reference

Directories

directory [**Inc**](#)

directory [**Src**](#)

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares › ST › MC_6Step_Lib › Src ›

Src Directory Reference

Files

file [**6Step_Lib.c \[code\]**](#)

This file provides the set of functions for Motor Control library.

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares › ST › MC_6Step_Lib › Inc ›

Inc Directory Reference

Files

file [**6Step_Lib.h \[code\]**](#)

This header file provides the set of functions for Motor Control library.

file [**stm32_nucleo_ihm11m1.h \[code\]**](#)

This file provides the interface between the MC-lib and STM Nucleo.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects			
<h2>Projects Directory Reference</h2>			

Directories

directory **Multi**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi		

Multi Directory Reference

Directories

directory **Applications**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	

Applications Directory Reference

Directories

directory **MotorControl**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl

MotorControl Directory Reference

Directories

directory [**Inc**](#)

directory [**Src**](#)

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
			Src

Src Directory Reference

Files

file **[main_F401.c \[code\]](#)**

This file provides a set of functions needed to configure STM32 MCU.

file **[stm32F401_nucleo_ihm11m1.c \[code\]](#)**

This file provides the interface between the MC-lib and STM Nucleo F401xx.

file **[stm32f4xx_hal_msp.c \[code\]](#)**

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

file **[stm32f4xx_it.c \[code\]](#)**

Interrupt Service Routines.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Projects	Multi	Applications	MotorControl
Inc			

Inc Directory Reference

Files

file [**main_F401.h \[code\]**](#)

Main program body for STM32F401xx.

file [**MC_SixStep_param.h \[code\]**](#)

This header file provides all parameters to driver a motor with 6Step library.

file [**stm32F401_nucleo_ihm11m1.h \[code\]**](#)

This file provides the interface between the MC-lib and STM Nucleo.

file [**stm32f4xx_hal_conf.h \[code\]**](#)

HAL configuration file.

file [**stm32f4xx_it.h \[code\]**](#)

This file contains the headers of the interrupt handlers.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Drivers

Drivers Directory Reference

Directories

directory **BSP**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP		
<h2>BSP Directory Reference</h2>			

Directories

directory **Components**

directory **X-NUCLEO-IHM11M1**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	

Components Directory Reference

Directories

directory **Common**

directory **STSPIN230**

Generated by **doxygen** 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page	Modules	Data Structures	Files
File List	Globals		
Drivers	BSP	Components	Common

Common Directory Reference

Files

file [**MC_Common.h \[code\]**](#)

This header file is a common file.

Generated by [doxygen](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Drivers

BSP

Components

STSPIN230

STSPIN230 Directory Reference

Files

file **[STSPIN230.c \[code\]](#)**

This file provides a set of functions to manage STSPIN230 driver.

file **[STSPIN230.h \[code\]](#)**

This file provides a set of functions to manage STSPIN230 driver.



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares

ST

UART_serial_com

UART_serial_com Directory Reference

Directories

directory [**Inc**](#)

directory [**Src**](#)

Generated by [**doxygen**](#) 1.8.11



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares › ST › UART_serial_com › Src ›

Src Directory Reference

Files

file [**UART_UI.c \[code\]**](#)

This file provides a set of functions needed to manage the UART com.

Generated by [doxygen 1.8.11](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Middlewares › ST › UART_serial_com › Inc ›

Inc Directory Reference

Files

file [**UART_UI.h \[code\]**](#)

This file provides a set of functions needed to manage the UART com.

Generated by [doxygen 1.8.11](#)



X-CUBE-SPN11 for X-NUCLEO-IHM11M1

Main Page

Modules

Data Structures

Files

File List

Globals

Drivers

BSP

X-NUCLEO-IHM11M1

X-NUCLEO-IHM11M1 Directory Reference

Files

file **X-NUCLEO-IHM11M1.c** [code]

This file provides the set of functions to manage the X-Nucleo expansion board.

file **X-NUCLEO-IHM11M1.h** [code]

This file provides the set of functions to manage the X-Nucleo board.