

ECSE426 Lab 2 - Group 19

19

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	File Index	3
2.1	File List	3
3	Module Documentation	5
3.1	CMSIS	5
3.1.1	Detailed Description	5
3.2	Stm32f4xx_system	6
3.2.1	Detailed Description	6
3.3	STM32F4xx_System_Private_Includes	7
3.3.1	Detailed Description	7
3.3.2	Macro Definition Documentation	7
3.3.2.1	HSE_VALUE	7
3.3.2.2	HSI_VALUE	7
3.4	STM32F4xx_System_Private_TypesDefinitions	8
3.5	STM32F4xx_System_Private_Defines	9
3.5.1	Detailed Description	9
3.5.2	Macro Definition Documentation	9
3.5.2.1	VECT_TAB_OFFSET	9
3.6	STM32F4xx_System_Private_Macros	10
3.7	STM32F4xx_System_Private_Variables	11
3.7.1	Detailed Description	11
3.8	STM32F4xx_System_Private_FunctionPrototypes	12
3.9	STM32F4xx_System_Private_Functions	13
3.9.1	Detailed Description	13
3.9.2	Function Documentation	13
3.9.2.1	SystemCoreClockUpdate()	13
3.9.2.2	SystemInit()	14

4 File Documentation	15
4.1 main.c File Reference	15
4.1.1 Detailed Description	16
4.1.2 Function Documentation	17
4.1.2.1 _Error_Handler()	17
4.1.2.2 display_num()	17
4.1.2.3 HAL_ADC_ConvCpltCallback()	17
4.1.2.4 SystemClock_Config()	18
4.2 stm32f4xx_it.c File Reference	18
4.2.1 Detailed Description	19
4.3 system_stm32f4xx.c File Reference	19
4.3.1 Detailed Description	20
4.4 voltmeter.c File Reference	21
4.4.1 Detailed Description	21
4.4.2 Function Documentation	22
4.4.2.1 FIR_C()	22
4.4.2.2 get_display_leds()	22
4.4.2.3 plot_point()	22
Index	25

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

CMSIS	5
Stm32f4xx_system	6
STM32F4xx_System_Private_Includes	7
STM32F4xx_System_Private_TypesDefinitions	8
STM32F4xx_System_Private_Defines	9
STM32F4xx_System_Private_Macros	10
STM32F4xx_System_Private_Variables	11
STM32F4xx_System_Private_FunctionPrototypes	12
STM32F4xx_System_Private_Functions	13

Chapter 2

File Index

2.1 File List

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

main.c	Main program body	15
stm32f4xx_it.c	Interrupt Service Routines	18
system_stm32f4xx.c	CMSIS Cortex-M4 Device Peripheral Access Layer System Source File	19
voltmeter.c	Handles Voltmeter Utilities	21

Chapter 3

Module Documentation

3.1 CMSIS

Modules

- [Stm32f4xx_system](#)

3.1.1 Detailed Description

3.2 Stm32f4xx_system

Modules

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

3.2.1 Detailed Description

3.3 STM32F4xx_System_Private_Includes

Macros

- #define [HSE_VALUE](#) ((uint32_t)25000000)
- #define [HSI_VALUE](#) ((uint32_t)16000000)

3.3.1 Detailed Description

3.3.2 Macro Definition Documentation

3.3.2.1 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)25000000)
```

Default value of the External oscillator in Hz

3.3.2.2 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)16000000)
```

Value of the Internal oscillator in Hz

3.4 STM32F4xx_System_Private_TypesDefinitions

3.5 STM32F4xx_System_Private_Defines

Macros

- #define [VECT_TAB_OFFSET](#) 0x00

3.5.1 Detailed Description

3.5.2 Macro Definition Documentation

3.5.2.1 VECT_TAB_OFFSET

```
#define VECT_TAB_OFFSET 0x00
```

< Uncomment the following line if you need to use external SRAM or SDRAM as data memory

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

3.6 STM32F4xx_System_Private_Macros

3.7 STM32F4xx_System_Private_Variables

Variables

- uint32_t **SystemCoreClock** = 16000000
- const uint8_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

3.7.1 Detailed Description

3.8 STM32F4xx_System_Private_FunctionPrototypes

3.9 STM32F4xx_System_Private_Functions

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

3.9.1 Detailed Description

3.9.2 Function Documentation

3.9.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (  
    void )
```

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

Note

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

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

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

(*) HSI_VALUE is a constant defined in stm32f4xx_hal_conf.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(**) HSE_VALUE is a constant defined in stm32f4xx_hal_conf.h file (its value depends on the application requirements), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

Parameters

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

Return values

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

3.9.2.2 SystemInit()

```
void SystemInit (  
                void )
```

Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.

Parameters

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

Return values

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

Chapter 4

File Documentation

4.1 main.c File Reference

Main program body.

```
#include "main.h"
#include "voltmeter.h"
#include "stm32f4xx_hal.h"
```

Macros

- `#define SYSTICK_FREQUENCY 200`
- `#define ADC_RES 8`

Functions

- void [SystemClock_Config](#) (void)
System Clock Configuration.
- void [HAL_ADC_ConvCpltCallback](#) (ADC_HandleTypeDef *hadc)
Callback function invoked when ADC has finished converting a reading.
- void [display_num](#) (char code)
Turns on appropriate display segments.
- int **main** (void)
- void [_Error_Handler](#) (char *file, int line)
This function is executed in case of error occurrence.

Variables

- ADC_HandleTypeDef [hadc1](#)
Handle to the analog to digital converter interface.
- DAC_HandleTypeDef [hdac](#)
Handle to the digital to analog converter interface.
- int [adc_val](#)
Stores latest ADC reading (digital)
- int [dac_val](#)
Stores latest digital value sent to the DAC.
- float [results](#) [3]
Stores the latest values for each display mode (RMS, min, max)
- int [change_mode](#) = 0
Flag to change display mode, handled in [EXTI0_IRQHandler\(\)](#) and [SysTick_Handler\(\)](#)
- int [display_mode](#) = RMS_MODE
Dictates which value from results[] to display.
- int [counter](#) = 0
Keeps track of SysTick interrupts for use as a software counter.

4.1.1 Detailed Description

Main program body.

This notice applies to any and all portions of this file that are not between comment pairs USER CODE BEGIN and USER CODE END. Other portions of this file, whether inserted by the user or by software development tools are owned by their respective copyright owners.

COPYRIGHT(c) 2018 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.

4.1.2 Function Documentation

4.1.2.1 _Error_Handler()

```
void _Error_Handler (
    char * file,
    int line )
```

This function is executed in case of error occurrence.

Parameters

None	
------	--

Return values

None	
------	--

4.1.2.2 display_num()

```
void display_num (
    char code )
```

Turns on appropriate display segments.

Parameters

<i>code</i>	Byte corresponding to the segments to turn on
-------------	---

Return values

None	
------	--

4.1.2.3 HAL_ADC_ConvCpltCallback()

```
void HAL_ADC_ConvCpltCallback (
    ADC_HandleTypeDef * hadc )
```

Callback function invoked when ADC has finished converting a reading.

Parameters

<i>hadc</i>	Handle to the ADC unit that has finished converting a value
-------------	---

Return values

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

4.1.2.4 SystemClock_Config()

```
void SystemClock_Config (
    void )
```

System Clock Configuration.

Configure the main internal regulator output voltage

Initializes the CPU, AHB and APB busses clocks

Initializes the CPU, AHB and APB busses clocks

Configure the SysTick interrupt time

Configure the SysTick

4.2 stm32f4xx_it.c File Reference

Interrupt Service Routines.

```
#include "stm32f4xx_hal.h"
#include "stm32f4xx.h"
#include "stm32f4xx_it.h"
#include "voltmeter.h"
```

Functions

- void [SysTick_Handler](#) (void)
This function handles System tick timer.
- void [EXTI0_IRQHandler](#) (void)
This function handles EXTI line0 interrupt.
- void [ADC_IRQHandler](#) (void)
This function handles ADC1, ADC2 and ADC3 global interrupts.

Variables

- ADC_HandleTypeDef [hadc1](#)
Handle to the analog to digital converter interface.
- int [adc_val](#)
Stores latest ADC reading (digital)
- int [dac_val](#)
Stores latest digital value sent to the DAC.
- int [change_mode](#)
Flag to change display mode, handled in [EXTI0_IRQHandler\(\)](#) and [SysTick_Handler\(\)](#)
- int [display_mode](#)
Dictates which value from results[] to display.
- int [counter](#)
Keeps track of SysTick interrupts for use as a software counter.

4.2.1 Detailed Description

Interrupt Service Routines.

COPYRIGHT(c) 2018 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.

4.3 system_stm32f4xx.c File Reference

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

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

Macros

- #define **HSE_VALUE** ((uint32_t)25000000)
- #define **HSI_VALUE** ((uint32_t)16000000)
- #define **VECT_TAB_OFFSET** 0x00

Functions

- void **SystemInit** (void)
Setup the microcontroller system Initialize the FPU setting, vector table location and External memory configuration.
- void **SystemCoreClockUpdate** (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

- uint32_t **SystemCoreClock** = 16000000
- const uint8_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

4.3.1 Detailed Description

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

Author

MCD Application Team

Version

V2.6.1

Date

- 14-February-2017 This file provides two functions and one global variable to be called from user application:
- **SystemInit()**: This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32f4xx.s" file.
 - **SystemCoreClock** variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
 - **SystemCoreClockUpdate()**: Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.

Attention

© COPYRIGHT 2017 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.

4.4 voltmeter.c File Reference

Handles Voltmeter Utilities.

```
#include "voltmeter.h"
#include "math.h"
```

Functions

- void [FIR_C](#) (int Input, float *Output)
FIR filter 4th order filter that outputs the average over the past 5 inputs.
- char [get_display_leds](#) (int c)
Get byte to indicate which segments should be turned on for a given integer Should be used as input to [display_num\(\)](#)
- void [plot_point](#) (float input, float *output)
Processes ADC readings Used to keep track of RMS, min, and max over the past 10 seconds.

4.4.1 Detailed Description

Handles Voltmeter Utilities.

4.4.2 Function Documentation

4.4.2.1 FIR_C()

```
void FIR_C (
    int Input,
    float * Output )
```

FIR filter 4th order filter that outputs the average over the past 5 inputs.

Parameters

<i>Input</i>	Digital input reading
<i>Output</i>	Reference to float to store the filtered output

Return values

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

4.4.2.2 get_display_leds()

```
char get_display_leds (
    int c )
```

Get byte to indicate which segments should be turned on for a given integer Should be used as input to [display_num\(\)](#)

Parameters

<i>c</i>	Integer to get byte code for
----------	------------------------------

Return values

<i>Byte</i>	indicating which segments should be turned on
-------------	---

4.4.2.3 plot_point()

```
void plot_point (
    float input,
    float * output )
```

Processes ADC readings Used to keep track of RMS, min, and max over the past 10 seconds.

Parameters

<i>input</i>	Latest analog ADC reading
<i>output</i>	Array to store the three display values (RMS, min, max)

Return values

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

Index

- `_Error_Handler`
main.c, [17](#)
- CMSIS, [5](#)
- display_num
main.c, [17](#)
- FIR_C
voltmeter.c, [22](#)
- get_display_leds
voltmeter.c, [22](#)
- HAL_ADC_ConvCpltCallback
main.c, [17](#)
- HSE_VALUE
STM32F4xx_System_Private_Includes, [7](#)
- HSI_VALUE
STM32F4xx_System_Private_Includes, [7](#)
- main.c, [15](#)
 - `_Error_Handler`, [17](#)
 - display_num, [17](#)
 - HAL_ADC_ConvCpltCallback, [17](#)
 - SystemClock_Config, [18](#)
- plot_point
voltmeter.c, [22](#)
- STM32F4xx_System_Private_Defines, [9](#)
 - VECT_TAB_OFFSET, [9](#)
- STM32F4xx_System_Private_FunctionPrototypes, [12](#)
- STM32F4xx_System_Private_Functions, [13](#)
 - SystemCoreClockUpdate, [13](#)
 - SystemInit, [14](#)
- STM32F4xx_System_Private_Includes, [7](#)
 - HSE_VALUE, [7](#)
 - HSI_VALUE, [7](#)
- STM32F4xx_System_Private_Macros, [10](#)
- STM32F4xx_System_Private_TypesDefinitions, [8](#)
- STM32F4xx_System_Private_Variables, [11](#)
- stm32f4xx_it.c, [18](#)
- Stm32f4xx_system, [6](#)
- system_stm32f4xx.c, [19](#)
- SystemClock_Config
main.c, [18](#)
- SystemCoreClockUpdate
STM32F4xx_System_Private_Functions, [13](#)
- SystemInit
STM32F4xx_System_Private_Functions, [14](#)
- VECT_TAB_OFFSET
STM32F4xx_System_Private_Defines, [9](#)
- voltmeter.c, [21](#)
 - FIR_C, [22](#)
 - get_display_leds, [22](#)
 - plot_point, [22](#)