

Inteligentna żarówka LED

AUTHOR

Aleksandra Antoszevska

Jagienka Wójcik

Version 1

Wed Jan 27 2021

Table of Contents

Table of contents

Module Index

Modules

Here is a list of all modules:

BSP	11
STM32L4XX_NUCLEO	12
STM32L4XX_NUCLEO BUS	4
BUS Exported Variables	5
BUS Private Variables	6
BUS Private Function	7
STM32L4XX_NUCLEO LOW LEVEL Private Functions	8
STM32L4XX_NUCLEO_BUS Exported Functions	9
 CMSIS	13
Stm32l4xx_system	14
STM32L4xx_System_Private_Includes	15
STM32L4xx_System_Private_TypesDefinitions	16
STM32L4xx_System_Private_Defines	17
STM32L4xx_System_Private_Macros	18
STM32L4xx_System_Private_Variables	19
STM32L4xx_System_Private_FunctionPrototypes	20
STM32L4xx_System_Private_Functions	21

File Index

File List

Here is a list of all files with brief descriptions:

projekt_pmik/Core/Src/adc.c (This file provides code for the configuration of the ADC instances)	23
projekt_pmik/Core/Src/bsp.c	25
projekt_pmik/Core/Src/gpio.c (This file provides code for the configuration of all used GPIO pins)	26
projekt_pmik/Core/Src/i2c.c (This file provides code for the configuration of the I2C instances)	27
projekt_pmik/Core/Src/lcd_i2c.c	29
projekt_pmik/Core/Src/main.c (: Main program body)	30
projekt_pmik/Core/Src/stm32l4xx_hal_msp.c (This file provides code for the MSP Initialization and de-Initialization codes)	36
projekt_pmik/Core/Src/stm32l4xx_it.c (Interrupt Service Routines)	37
projekt_pmik/Core/Src/stm32l4xx_nucleo_bus.c (: source file for the BSP BUS IO driver)	41
projekt_pmik/Core/Src/syscalls.c (STM32CubeIDE Minimal System calls file)	42
projekt_pmik/Core/Src/systemem.c (STM32CubeIDE System Memory calls file)	45
projekt_pmik/Core/Src/system_stm32l4xx.c (CMSIS Cortex-M4 Device Peripheral Access Layer System Source File)	47
projekt_pmik/Core/Src/tim.c (This file provides code for the configuration of the TIM instances)	50

Module Documentation

STM32L4XX_NUCLEO BUS

Modules

- **BUS Exported Variables**
 - **BUS Private Variables**
 - **BUS Private Function**
 - **STM32L4XX_NUCLEO LOW LEVEL Private Functions**
-

Detailed Description

BUS Exported Variables

Variables

- SPI_HandleTypeDef **hspl1**
-

Detailed Description

Variable Documentation

SPI_HandleTypeDef **hspl1**

Definition at line 40 of file stm32l4xx_nucleo_bus.c.

BUS Private Variables

Variables

- `static uint32_t SPI1InitCounter = 0`
-

Detailed Description

Variable Documentation

`uint32_t SPI1InitCounter = 0 [static]`

Definition at line 52 of file `stm32l4xx_nucleo_bus.c`.

BUS Private Function

Functions

- static void **SPI1_MspInit** (SPI_HandleTypeDef *hSPI)
 - static void **SPI1_MspDeInit** (SPI_HandleTypeDef *hSPI)
-

Detailed Description

Function Documentation

static void SPI1_MspDeInit (SPI_HandleTypeDef * *hSPI*)*[static]*

SPI1 GPIO Configuration PA6 ----> SPI1_MISO PA7 ----> SPI1_MOSI PB3
(JTDO-TRACESWO) ----> SPI1_SCK

Definition at line 331 of file stm32l4xx_nucleo_bus.c.

static void SPI1_MspInit (SPI_HandleTypeDef * *hSPI*)*[static]*

SPI1 GPIO Configuration PA6 ----> SPI1_MISO PA7 ----> SPI1_MOSI PB3
(JTDO-TRACESWO) ----> SPI1_SCK

Definition at line 289 of file stm32l4xx_nucleo_bus.c.

STM32L4XX_NUCLEO LOW LEVEL Private Functions

Modules

- **STM32L4XX_NUCLEO_BUS Exported Functions**
-

Detailed Description

STM32L4XX_NUCLEO_BUS Exported Functions

Functions

- `int32_t BSP_SPI1_Init` (void)
Initializes SPI HAL.
- `int32_t BSP_SPI1_DeInit` (void)
DeInitializes SPI HAL.
- `int32_t BSP_SPI1_Send` (uint8_t *pData, uint16_t Length)
Write Data through SPI BUS.
- `int32_t BSP_SPI1_Recv` (uint8_t *pData, uint16_t Length)
Receive Data from SPI BUS.
- `int32_t BSP_SPI1_SendRecv` (uint8_t *pTxData, uint8_t *pRxData, uint16_t Length)
Send and Receive data to/from SPI BUS (Full duplex)
- `int32_t BSP_GetTick` (void)
Return system tick in ms.
- `__weak HAL_StatusTypeDef MX_SPI1_Init` (SPI_HandleTypeDef *hspi)

Detailed Description

Function Documentation

`int32_t BSP_GetTick` (void)

Return system tick in ms.

Return values

<i>Current</i>	HAL time base time stamp
----------------	--------------------------

Definition at line 258 of file stm32l4xx_nucleo_bus.c.

`int32_t BSP_SPI1_DeInit` (void)

DeInitializes SPI HAL.

Return values

<i>None</i>	
<i>BSP</i>	status

Definition at line 129 of file stm32l4xx_nucleo_bus.c.

int32_t BSP_SPI1_Init (void)

Initializes SPI HAL.

Return values

<i>BSP</i>	status
------------	--------

Definition at line 88 of file stm32l4xx_nucleo_bus.c.

int32_t BSP_SPI1_Recv (uint8_t * *pData*, uint16_t *Length*)

Receive Data from SPI BUS.

Parameters

<i>pData</i>	Pointer to data buffer to receive
<i>Length</i>	Length of data in byte

Return values

<i>BSP</i>	status
------------	--------

Definition at line 172 of file stm32l4xx_nucleo_bus.c.

int32_t BSP_SPI1_Send (uint8_t * *pData*, uint16_t *Length*)

Write Data through SPI BUS.

Parameters

<i>pData</i>	Pointer to data buffer to send
<i>Length</i>	Length of data in byte

Return values

<i>BSP</i>	status
------------	--------

Definition at line 155 of file stm32l4xx_nucleo_bus.c.

int32_t BSP_SPI1_SendRecv (uint8_t * *pTxData*, uint8_t * *pRxData*, uint16_t *Length*)

Send and Receive data to/from SPI BUS (Full duplex)

Parameters

<i>pData</i>	Pointer to data buffer to send/receive
<i>Length</i>	Length of data in byte

Return values

<i>BSP</i>	status
------------	--------

Definition at line 189 of file stm32l4xx_nucleo_bus.c.

__weak HAL_StatusTypeDef MX_SPI1_Init (SPI_HandleTypeDef * *hspi*)

Definition at line 264 of file stm32l4xx_nucleo_bus.c.

BSP

Modules

- STM32L4XX_NUCLEO
-

Detailed Description

STM32L4XX_NUCLEO

Modules

- STM32L4XX_NUCLEO BUS
-

Detailed Description

CMSIS

Modules

- Stm32l4xx_system

Detailed Description

Stm32l4xx_system

Modules

- STM32L4xx_System_Private_Includes
 - STM32L4xx_System_Private_TypesDefinitions
 - STM32L4xx_System_Private_Defines
 - STM32L4xx_System_Private_Macros
 - STM32L4xx_System_Private_Variables
 - STM32L4xx_System_Private_FunctionPrototypes
 - STM32L4xx_System_Private_Functions
-

Detailed Description

STM32L4xx_System_Private_Includes

Macros

- `#define HSE_VALUE 8000000U`
 - `#define MSI_VALUE 4000000U`
 - `#define HSI_VALUE 16000000U`
-

Detailed Description

Macro Definition Documentation

#define HSE_VALUE 8000000U

Value of the External oscillator in Hz

Definition at line 95 of file system_stm32l4xx.c.

#define HSI_VALUE 16000000U

Value of the Internal oscillator in Hz

Definition at line 103 of file system_stm32l4xx.c.

#define MSI_VALUE 4000000U

Value of the Internal oscillator in Hz

Definition at line 99 of file system_stm32l4xx.c.

STM32L4xx_System_Private_TypesDefinitions

STM32L4xx_System_Private_Defines

Macros

- `#define VECT_TAB_OFFSET 0x00`
-

Detailed Description

Macro Definition Documentation

#define VECT_TAB_OFFSET 0x00

< 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.

Definition at line 127 of file system_stm32l4xx.c.

STM32L4xx_System_Private_Macros

STM32L4xx_System_Private_Variables

Variables

- `uint32_t SystemCoreClock = 4000000U`
- `const uint8_t AHBPrescTable [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}`
- `const uint8_t APBPrescTable [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}`
- `const uint32_t MSIRangeTable [12]`

Detailed Description

Variable Documentation

`const uint8_t AHBPrescTable[16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}`

Definition at line 154 of file `system_stm32l4xx.c`.

`const uint8_t APBPrescTable[8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}`

Definition at line 155 of file `system_stm32l4xx.c`.

`const uint32_t MSIRangeTable[12]`

```
Initial value:= {1000000U, 2000000U, 4000000U, 8000000U, 10000000U, 20000000U,
40000000U, 80000000U, 160000000U, 240000000U,
320000000U, 480000000U}
```

Definition at line 156 of file `system_stm32l4xx.c`.

`uint32_t SystemCoreClock = 4000000U`

Definition at line 152 of file `system_stm32l4xx.c`.

STM32L4xx_System_Private_FunctionPrototypes

STM32L4xx_System_Private_Functions

Functions

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

Detailed Description

Function Documentation

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 MSI, SystemCoreClock will contain the **MSI_VALUE**(*)
- 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**(*) or **MSI_VALUE**(*) multiplied/divided by the PLL factors.

(*) **MSI_VALUE** is a constant defined in stm32l4xx_hal.h file (default value 4 MHz) but the real value may vary depending on the variations in voltage and temperature.

(**) **HSI_VALUE** is a constant defined in stm32l4xx_hal.h file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(***) **HSE_VALUE** is a constant defined in stm32l4xx_hal.h file (default value 8 MHz), user has to ensure that **HSE_VALUE** is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

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

Parameters

None	
------	--

Return values

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

Definition at line 256 of file system_stm32l4xx.c.

void SystemInit (void)

Setup the microcontroller system.

Parameters

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

Return values

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

Definition at line 180 of file system_stm32l4xx.c.

File Documentation

projekt_pmik/Core/Src/adc.c File Reference

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

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

Functions

- void **MX_ADC1_Init** (void)
- void **MX_ADC2_Init** (void)
- void **HAL_ADC_MspInit** (ADC_HandleTypeDef *adcHandle)
- void **HAL_ADC_MspDeInit** (ADC_HandleTypeDef *adcHandle)

Variables

- ADC_HandleTypeDef **hadc1**
- ADC_HandleTypeDef **hadc2**
- static uint32_t **HAL_RCC_ADC_CLK_ENABLED** =0

Detailed Description

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

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: www.st.com/SLA0044

Function Documentation

void HAL_ADC_MspDeInit (ADC_HandleTypeDef * *adcHandle*)

ADC1 GPIO Configuration PC0 ----> ADC1_IN1 PB0 ----> ADC1_IN15

ADC2 GPIO Configuration PC1 ----> ADC2_IN2 PA4 ----> ADC2_IN9

Definition at line 189 of file adc.c.

void HAL_ADC_MspInit (ADC_HandleTypeDef * *adcHandle*)

ADC1 GPIO Configuration PC0 ----> ADC1_IN1 PB0 ----> ADC1_IN15

ADC2 GPIO Configuration PC1 ----> ADC2_IN2 PA4 ----> ADC2_IN9

Definition at line 121 of file adc.c.

void MX_ADC1_Init (void)

Common config
Configure the ADC multi-mode
Configure Regular Channel
Definition at line 31 of file adc.c.

void MX_ADC2_Init (void)

Common config
Configure Regular Channel
Definition at line 79 of file adc.c.

Variable Documentation

ADC_HandleTypeDef hadc1

Definition at line 27 of file adc.c.

ADC_HandleTypeDef hadc2

Definition at line 28 of file adc.c.

uint32_t HAL_RCC_ADC_CLK_ENABLED =0[static]

Definition at line 119 of file adc.c.

projekt_pmik/Core/Src/bsp.c File Reference

projekt_pmik/Core/Src/gpio.c File Reference

This file provides code for the configuration of all used GPIO pins.
`#include "gpio.h"`

Functions

- `void MX_GPIO_Init (void)`
-

Detailed Description

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

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: www.st.com/SLA0044

Function Documentation

`void MX_GPIO_Init (void)`

Configure pins as Analog Input Output EVENT_OUT EXTI

Definition at line 41 of file `gpio.c`.

projekt_pmik/Core/Src/i2c.c File Reference

This file provides code for the configuration of the I2C instances.

```
#include "i2c.h"
```

Functions

- void **MX_I2C1_Init** (void)
- void **HAL_I2C_MspInit** (I2C_HandleTypeDef *i2cHandle)
- void **HAL_I2C_MspDeInit** (I2C_HandleTypeDef *i2cHandle)

Variables

- I2C_HandleTypeDef **hi2c1**

Detailed Description

This file provides code for the configuration of the I2C instances.

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: www.st.com/SLA0044

Function Documentation

void HAL_I2C_MspDeInit (I2C_HandleTypeDef * *i2cHandle*)

I2C1 GPIO Configuration PB8 ----> I2C1_SCL PB9 ----> I2C1_SDA
Definition at line 91 of file i2c.c.

void HAL_I2C_MspInit (I2C_HandleTypeDef * *i2cHandle*)

I2C1 GPIO Configuration PB8 ----> I2C1_SCL PB9 ----> I2C1_SDA
Definition at line 61 of file i2c.c.

void MX_I2C1_Init (void)

Configure Analogue filter
Configure Digital filter
Definition at line 30 of file i2c.c.

Variable Documentation

I2C_HandleTypeDef hi2c1

Definition at line 27 of file i2c.c.

projekt_pmik/Core/Src/lcd_i2c.c File Reference

```
#include "lcd_i2c.h"
#include "stm32l4xx_hal.h"
#include "i2c.h"
```

Functions

- void **lcd_init** (struct lcd_disp *lcd)
 - void **lcd_write** (uint8_t addr, uint8_t data, uint8_t xpin)
 - void **lcd_display** (struct lcd_disp *lcd)
 - void **lcd_clear** (struct lcd_disp *lcd)
-

Function Documentation

void lcd_clear (struct lcd_disp * *lcd*)

set backlight

clear display

Definition at line 112 of file lcd_i2c.c.

void lcd_display (struct lcd_disp * *lcd*)

set backlight

send first line data

send second line data

Definition at line 82 of file lcd_i2c.c.

void lcd_init (struct lcd_disp * *lcd*)

set backlight I2C P3 (LCD BL)

init sequence

set 4-bit mode

set cursor mode

clear

Definition at line 30 of file lcd_i2c.c.

void lcd_write (uint8_t *addr*, uint8_t *data*, uint8_t *xpin*)

split data

send data via i2c

Definition at line 66 of file lcd_i2c.c.

projekt_pmik/Core/Src/main.c File Reference

```
: Main program body
#include "main.h"
#include "adc.h"
#include "i2c.h"
#include "tim.h"
#include "gpio.h"
#include "app_bluenrg_ms.h"
#include "lcd_i2c.h"
#include <string.h>
#include <stdio.h>
#include <math.h>
#include <stdbool.h>
#include <unistd.h>
```

Macros

- `#define timer_freq 80.0`
- `#define T0H 0.35`
- `#define T1H 0.9`
- `#define T0L 0.9`
- `#define T1L 0.35`
- `#define Treset 50`

Functions

- void **SystemClock_Config** (void)
System Clock Configuration.
- void **Neopixel_setup** (void)
- void **show_neopixels** ()
- void **set_colour** (int x, int y)
- void **print_measurement** (uint16_t raw, uint16_t raw2)
- void **light_correctness** (uint16_t raw2, int j)
- uint8_t **neopixel_transmitting** ()
- void **measurement_light** (int z)
- void **HAL_TIM_PeriodElapsedCallback** (TIM_HandleTypeDef *htim)
- int **main** (void)
The application entry point.
- void **TIM4_IRQHandler** (void)
- void **Error_Handler** (void)
This function is executed in case of error occurrence.

Variables

- ADC_HandleTypeDef **hadc1**
- ADC_HandleTypeDef **hadc2**
- uint8_t **LED_data** [24]
- uint16_t **pos**
- uint8_t **mask** = 0B10000000
- uint8_t **lastbit**
- long double **period**

- uint16_t **low_CCR1**
- uint16_t **low_ARR**
- uint16_t **high_CCR1**
- uint16_t **high_ARR**
- uint16_t **treset_ARR**
- uint16_t **raw**
- uint16_t **raw2**
- int **j** = 2
- int **z** = 0

Detailed Description

: Main program body

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

Macro Definition Documentation

#define T0H 0.35

each different clone can have their own timings timing here are in us
Definition at line 54 of file main.c.

#define T0L 0.9

each different clone can have their own timings timing here are in us
Definition at line 64 of file main.c.

#define T1H 0.9

each different clone can have their own timings timing here are in us
Definition at line 59 of file main.c.

#define T1L 0.35

each different clone can have their own timings timing here are in us
Definition at line 69 of file main.c.

#define timer_freq 80.0

timer clock freq in MHz
Definition at line 49 of file main.c.

#define Treset 50

each different clone can have their own timings timing here are in us
Definition at line 74 of file main.c.

Function Documentation

void Error_Handler (void)

This function is executed in case of error occurrence.

Return values

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

Definition at line 546 of file main.c.

void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim)

This function handles two timers: TIM17 is used to change red's color brightness while printing measurement, TIM17 is used to make measurements and change LED's color in party mode

set GPIO pin high - start timing

start adc conversion

set GPIO pin low - stop timing

white light when it's bright and party mode when it's dark

setting different color in next iteration

Definition at line 306 of file main.c.

void light_correctness (uint16_t raw2, int j)

This function is used to correct LEDs' brightness if needed. First input argument is measurement from sensor, the other one is current LED's color Things done in this function: setting party mode when it's dark setting white light when it's bright

Definition at line 273 of file main.c.

int main (void)

The application entry point.

Return values

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

Adding variables for adc and measurement display

Starting LEDs

address

brightness

Starting lcd

First line of first message

Second line of first message

Display first and second text line in lcd

In while loop program checks if button is pressed. If so, measurements are printed and LEDs turn red.

Definition at line 366 of file main.c.

void measurement_light (int z)

This function turns light to red on different level of brightness Things done in this function: setting red light while printing measurement

Definition at line 297 of file main.c.

void Neopixel_setup (void)

This function is used to setup LEDs Things done by this function: calculate all the timings enable port D clock setup pin 12 on port d to AF mode setup pin 12 on port D to AF timer 2-5 enable the timer4 clock set prescale to zero as timer has to go as fast as possible set PWM mode 110 set to zero so that the pin stay low until transmission set to timing for reset LEDs enable output to pin. Disable channel 1. This bit is used to start and stop transmission. buffer ARR buffer CCR1 ensure we are not enabling interrupt flag to be generated this bit is used to start/stop transmission enable channel 1. Enable interrupt(NVIC level)

Definition at line 119 of file main.c.

uint8_t neopixel_transmitting ()

This function is used for LEDs' action

Definition at line 289 of file main.c.

void print_measurement (uint16_t raw, uint16_t raw2)

This function is used to print measurement based on data given from sensors. Things done in this function: calculating temperature using given algorithm calculating power using given algorithm setting light to "dark" or "light" mode displaying measurement on lcd

Definition at line 243 of file main.c.

void set_colour (int x, int y)

This function is used to set color number x (first input argument) at brightness y (second input argument). Things done in this function: setting color of LEDs sending data to LEDs

Definition at line 182 of file main.c.

void show_neopixels ()

This function proceeds data to LEDs Things done by this function: set the interrupt to start at first byte set the interrupt to start at second bit clear UIF flag enable interrupt flag to be generated to start transmission

Definition at line 166 of file main.c.

void SystemClock_Config (void)

System Clock Configuration.

Return values

None	
------	--

Initializes the RCC Oscillators according to the specified parameters in the RCC_OscInitTypeDef structure.

Initializes the CPU, AHB and APB buses clocks

Configure the main internal regulator output voltage

Definition at line 450 of file main.c.

void TIM4_IRQHandler (void)

This function handles TIM4. Things done in this function: clearing UIF flag setting CCR1 to zero so that pin stays low setting ARR to timing for reset LEDs disabling interrupt flag to end transmission

Definition at line 511 of file main.c.

Variable Documentation

ADC_HandleTypeDef hadc1

Definition at line 85 of file main.c.

ADC_HandleTypeDef hadc2

Definition at line 85 of file main.c.

uint16_t high_ARR

Definition at line 103 of file main.c.

uint16_t high_CCR1

Definition at line 103 of file main.c.

int j = 2

led variable for changing colors

Definition at line 112 of file main.c.

uint8_t lastbit

Definition at line 101 of file main.c.

uint8_t LED_data[24]

variables needed for LEDs

Definition at line 98 of file main.c.

uint16_t low_ARR

Definition at line 103 of file main.c.

uint16_t low_CCR1

Definition at line 103 of file main.c.

uint8_t mask = 0B10000000

Definition at line 100 of file main.c.

long double period

Definition at line 102 of file main.c.

uint16_t pos

Definition at line 99 of file main.c.

uint16_t raw

sensor data

Definition at line 108 of file main.c.

uint16_t raw2

Definition at line 108 of file main.c.

uint16_t treset_ARR

Definition at line 103 of file main.c.

int z = 0

led variable for red light

Definition at line 116 of file main.c.

projekt_pmik/Core/Src/stm32l4xx_hal_msp.c File Reference

This file provides code for the MSP Initialization and de-Initialization codes.
`#include "main.h"`

Functions

- `void HAL_MspInit (void)`
-

Detailed Description

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

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

Function Documentation

`void HAL_MspInit (void)`

Initializes the Global MSP.

Definition at line 64 of file `stm32l4xx_hal_msp.c`.

projekt_pmik/Core/Src/stm32l4xx_it.c File Reference

Interrupt Service Routines.

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

```
#include "stm32l4xx_it.h"
```

Functions

- void **NMI_Handler** (void)
This function handles Non maskable interrupt.
- void **HardFault_Handler** (void)
This function handles Hard fault interrupt.
- void **MemManage_Handler** (void)
This function handles Memory management fault.
- void **BusFault_Handler** (void)
This function handles Prefetch fault, memory access fault.
- void **UsageFault_Handler** (void)
This function handles Undefined instruction or illegal state.
- void **SVC_Handler** (void)
This function handles System service call via SWI instruction.
- void **DebugMon_Handler** (void)
This function handles Debug monitor.
- void **PendSV_Handler** (void)
This function handles Pendable request for system service.
- void **SysTick_Handler** (void)
This function handles System tick timer.
- void **EXTI0_IRQHandler** (void)
This function handles EXTI line0 interrupt.
- void **TIM1_UP_TIM16_IRQHandler** (void)
This function handles TIM1 update interrupt and TIM16 global interrupt.
- void **TIM1_TRG_COM_TIM17_IRQHandler** (void)
This function handles TIM1 trigger and commutation interrupts and TIM17 global interrupt.
- void **EXTI15_10_IRQHandler** (void)
This function handles EXTI line[15:10] interrupts.

Variables

- TIM_HandleTypeDef **htim16**
 - TIM_HandleTypeDef **htim17**
-

Detailed Description

Interrupt Service Routines.

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

Function Documentation

void BusFault_Handler (void)

This function handles Prefetch fault, memory access fault.

Definition at line 116 of file stm32l4xx_it.c.

void DebugMon_Handler (void)

This function handles Debug monitor.

Definition at line 159 of file stm32l4xx_it.c.

void EXTI0_IRQHandler (void)

This function handles EXTI line0 interrupt.

Definition at line 206 of file stm32l4xx_it.c.

void EXTI15_10_IRQHandler (void)

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

Definition at line 248 of file stm32l4xx_it.c.

void HardFault_Handler (void)

This function handles Hard fault interrupt.

Definition at line 86 of file stm32l4xx_it.c.

void MemManage_Handler (void)

This function handles Memory management fault.

Definition at line 101 of file stm32l4xx_it.c.

void NMI_Handler (void)

This function handles Non maskable interrupt.

Definition at line 71 of file stm32l4xx_it.c.

void PendSV_Handler (void)

This function handles Pendable request for system service.

Definition at line 172 of file stm32l4xx_it.c.

void SVC_Handler (void)

This function handles System service call via SWI instruction.

Definition at line 146 of file stm32l4xx_it.c.

void SysTick_Handler (void)

This function handles System tick timer.

Definition at line 185 of file stm32l4xx_it.c.

void TIM1_TRG_COM_TIM17_IRQHandler (void)

This function handles TIM1 trigger and commutation interrupts and TIM17 global interrupt.

Definition at line 234 of file stm32l4xx_it.c.

void TIM1_UP_TIM16_IRQHandler (void)

This function handles TIM1 update interrupt and TIM16 global interrupt.

Definition at line 220 of file stm32l4xx_it.c.

void UsageFault_Handler (void)

This function handles Undefined instruction or illegal state.

Definition at line 131 of file stm32l4xx_it.c.

Variable Documentation

TIM_HandleTypeDef htim16 [extern]

Definition at line 28 of file tim.c.

TIM_HandleTypeDef htim17 [extern]

Definition at line 29 of file tim.c.

projekt_pmik/Core/Src/stm32l4xx_nucleo_bus.c File Reference

: source file for the BSP BUS IO driver
#include "stm32l4xx_nucleo_bus.h"

Functions

- `__weak HAL_StatusTypeDef MX_SPI1_Init (SPI_HandleTypeDef *hspi)`
- `static void SPI1_MspInit (SPI_HandleTypeDef *hSPI)`
- `static void SPI1_MspDeInit (SPI_HandleTypeDef *hSPI)`
- `int32_t BSP_SPI1_Init (void)`
Initializes SPI HAL.
- `int32_t BSP_SPI1_DeInit (void)`
DeInitializes SPI HAL.
- `int32_t BSP_SPI1_Send (uint8_t *pData, uint16_t Length)`
Write Data through SPI BUS.
- `int32_t BSP_SPI1_Recv (uint8_t *pData, uint16_t Length)`
Receive Data from SPI BUS.
- `int32_t BSP_SPI1_SendRecv (uint8_t *pTxData, uint8_t *pRxData, uint16_t Length)`
Send and Receive data to/from SPI BUS (Full duplex)
- `int32_t BSP_GetTick (void)`
Return system tick in ms.

Variables

- `SPI_HandleTypeDef hspi1`
 - `static uint32_t SPI1InitCounter = 0`
-

Detailed Description

: source file for the BSP BUS IO driver

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: www.st.com/SLA0044

projekt_pmik/Core/Src/syscalls.c File Reference

STM32CubeIDE Minimal System calls file.

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

Functions

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

Variables

- `int errno`
- `char ** environ = __env`

Detailed Description

STM32CubeIDE Minimal System calls file.

Author

Auto-generated by STM32CubeIDE

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

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

Function Documentation

__attribute__ ((weak))

Definition at line 69 of file syscalls.c.

int __io_getchar (void)

Definition at line 39 of file syscalls.c.

int __io_putchar (int *ch*)

int _close (int *file*)

Definition at line 92 of file syscalls.c.

int _execve (char * *name*, char ** *argv*, char ** *env*)

Definition at line 155 of file syscalls.c.

void _exit (int *status*)

Definition at line 63 of file syscalls.c.

int _fork (void)

Definition at line 149 of file syscalls.c.

int _fstat (int *file*, struct stat * *st*)

Definition at line 98 of file syscalls.c.

int _getpid (void)

Definition at line 52 of file syscalls.c.

int _isatty (int *file*)

Definition at line 104 of file syscalls.c.

int _kill (int *pid*, int *sig*)

Definition at line 57 of file syscalls.c.

int _link (char * *old*, char * *new*)

Definition at line 143 of file syscalls.c.

int _lseek (int *file*, int *ptr*, int *dir*)

Definition at line 109 of file syscalls.c.

int _open (char * *path*, int *flags*, ...)

Definition at line 114 of file syscalls.c.

int _stat (char * *file*, struct stat * *st*)

Definition at line 137 of file syscalls.c.

int _times (struct tms * *buf*)

Definition at line 132 of file syscalls.c.

int _unlink (char * *name*)

Definition at line 126 of file syscalls.c.

int _wait (int * *status*)

Definition at line 120 of file syscalls.c.

void initialise_monitor_handles ()

Definition at line 48 of file syscalls.c.

Variable Documentation

char environ = __env**

Definition at line 44 of file syscalls.c.

int errno [extern]

projekt_pmik/Core/Src/sysmem.c File Reference

STM32CubeIDE System Memory calls file.

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

Functions

- void * **_sbrk** (ptrdiff_t incr)
_sbrk() allocates memory to the newlib heap and is used by malloc and others from the C library

Variables

- static uint8_t * **__sbrk_heap_end** = NULL

Detailed Description

STM32CubeIDE System Memory calls file.

Author

Generated by STM32CubeIDE

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

Attention

© Copyright (c) 2020 STMicroelectronics. All rights reserved.

This software component is licensed by ST under BSD 3-Clause license, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/BSD-3-Clause

Function Documentation

void* _sbrk (ptrdiff_t incr)

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

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

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

Parameters

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

Returns

Pointer to allocated memory

Definition at line 54 of file sysmem.c.

Variable Documentation

uint8_t* __sbrk_heap_end = NULL[static]

Pointer to the current high watermark of the heap usage

Definition at line 31 of file sysmem.c.

projekt_pmik/Core/Src/system_stm32l4xx.c File Reference

CMSIS Cortex-M4 Device Peripheral Access Layer System Source File.
`#include "stm32l4xx.h"`

Macros

- `#define HSE_VALUE 8000000U`
- `#define MSI_VALUE 4000000U`
- `#define HSI_VALUE 16000000U`
- `#define VECT_TAB_OFFSET 0x00`

Functions

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

Variables

- `uint32_t SystemCoreClock = 4000000U`
- `const uint8_t AHBPrescTable [16] = {0U, 0U, 0U, 0U, 0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U, 6U, 7U, 8U, 9U}`
- `const uint8_t APBPrescTable [8] = {0U, 0U, 0U, 0U, 1U, 2U, 3U, 4U}`
- `const uint32_t MSIRangeTable [12]`

Detailed Description

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

Author

MCD Application Team

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

- **SystemInit()**: This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32l4xx.s" file.
- **SystemCoreClock** variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
- **SystemCoreClockUpdate()**: Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.

After each device reset the MSI (4 MHz) is used as system clock source. Then **SystemInit()** function is called, in "startup_stm32l4xx.s" file, to configure the system clock before to branch to main program.

This file configures the system clock as follows:

System Clock source	 MSI
SYSClk(Hz)	 4000000
HCLK(Hz)	 4000000
AHB Prescaler	 1
APB1 Prescaler	 1
APB2 Prescaler	 1
PLL_M	 1
PLL_N	 8
PLL_P	 7
PLL_Q	 2
PLL_R	 2
PLLSAI1_P	 NA
PLLSAI1_Q	 NA
PLLSAI1_R	 NA
PLLSAI2_P	 NA

PLLSAI2_Q | **NA**

PLLSAI2_R | **NA**

Require 48MHz for USB OTG FS, | Disabled

SDIO and RNG clock |

=====

Attention

© Copyright (c) 2017 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Apache License, Version 2.0, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: opensource.org/licenses/Apache-2.0

projekt_pmik/Core/Src/tim.c File Reference

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

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

Functions

- void **MX_TIM4_Init** (void)
- void **MX_TIM16_Init** (void)
- void **MX_TIM17_Init** (void)
- void **HAL_TIM_Base_MspInit** (TIM_HandleTypeDef *tim_baseHandle)
- void **HAL_TIM_MspPostInit** (TIM_HandleTypeDef *timHandle)
- void **HAL_TIM_Base_MspDeInit** (TIM_HandleTypeDef *tim_baseHandle)

Variables

- TIM_HandleTypeDef **htim4**
- TIM_HandleTypeDef **htim16**
- TIM_HandleTypeDef **htim17**

Detailed Description

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

Attention

© Copyright (c) 2021 STMicroelectronics. All rights reserved.

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License. You may obtain a copy of the License at: www.st.com/SLA0044

Function Documentation

void HAL_TIM_Base_MspDeInit (TIM_HandleTypeDef * *tim_baseHandle*)

Definition at line 182 of file tim.c.

void HAL_TIM_Base_MspInit (TIM_HandleTypeDef * *tim_baseHandle*)

Definition at line 109 of file tim.c.

void HAL_TIM_MspPostInit (TIM_HandleTypeDef * *timHandle*)

TIM4 GPIO Configuration PB6 ----> TIM4_CH1

Definition at line 154 of file tim.c.

void MX_TIM16_Init (void)

Definition at line 75 of file tim.c.

void MX_TIM17_Init (void)

Definition at line 92 of file tim.c.

void MX_TIM4_Init (void)

Definition at line 32 of file tim.c.

Variable Documentation

TIM_HandleTypeDef htim16

Definition at line 28 of file tim.c.

TIM_HandleTypeDef htim17

Definition at line 29 of file tim.c.

TIM_HandleTypeDef htim4

Definition at line 27 of file tim.c.

Index

INDEX