

Minutnik_kuchenny
v1

Generated by Doxygen 1.9.6

Chapter 1

Minutnik_kuchenny

Autorzy projektu

- **Marcin Ludyga**
- **Adrian Nowak**

Zalozenia projektu

- Nastawianie czasu w zakresie od 1 minuty do 9 godzin przy uzyciu enkodera
 - Zmiana czasu
 - co 1 minute jezeli nie przekroczono godziny
 - co 10 minut jezeli przekroczono godzine
 - Wyszwietlanie na wyswietlaczu czasu w formacie
 - mm:ss jesli nie przekroczono godziny
 - hh:mm jesli przekroczono godzine
 - Zmiana pracy urzadzenie (ustawianie/odliczanie) przy pomocy przycisku
 - Odmierzenie czasu sygnalizowane dzwiekowo
-

Pliki projektu

- [main.c](#)
 - [vMain\(\)](#) zadanie glowne
 - [vEncoder\(\)](#) zadanie enkodera
 - [vDisp\(\)](#) zadanie wyswietlacza
 - [stm32f1xx_hal_timebase_TIM.c](#)
 - [stm32f1xx_it.c](#)
-

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

STM32F7xx_HAL_Examples	9
HAL_TimeBase	9

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

[display](#)

Elementy wchodzace w sklad kolejki pomiedzy zadaniem glownym a wyswietlaczem [13](#)

Chapter 4

File Index

4.1 File List

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

main.c	Plik zawierajacy glowne funkcje oraz zadania programu	15
stm32f1xx_hal_timebase_TIM.c	HAL time base based on the hardware TIM	??
stm32f1xx_it.c	Interrupt Service Routines	??

Chapter 5

Module Documentation

5.1 STM32F7xx_HAL_Examples

Modules

- [HAL_TimeBase](#)

5.1.1 Detailed Description

5.2 HAL_TimeBase

Functions

- HAL_StatusTypeDef [HAL_InitTick](#) (uint32_t TickPriority)
This function configures the TIM6 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.
- void [HAL_SuspendTick](#) (void)
Suspend Tick increment.
- void [HAL_ResumeTick](#) (void)
Resume Tick increment.
- void [HAL_TIM_PeriodElapsedCallback](#) (TIM_HandleTypeDef *htim)
Period elapsed callback in non blocking mode.

Variables

- TIM_HandleTypeDef **htim6**
- uint32_t **uwIncrementState** = 0

5.2.1 Detailed Description

5.2.2 Function Documentation

5.2.2.1 HAL_InitTick()

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

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

Note

This function is called automatically at the beginning of program after reset by HAL_Init() or at any time when clock is configured, by HAL_RCC_ClockConfig().

Parameters

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

Return values

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

5.2.2.2 HAL_ResumeTick()

```
void HAL_ResumeTick (
    void )
```

Resume Tick increment.

Note

Enable the tick increment by Enabling TIM6 update interrupt.

Parameters

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

5.2.2.3 HAL_SuspendTick()

```
void HAL_SuspendTick (
    void )
```

Suspend Tick increment.

Note

Disable the tick increment by disabling TIM6 update interrupt.

Parameters

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

5.2.2.4 HAL_TIM_PeriodElapsedCallback()

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

Period elapsed callback in non blocking mode.

Note

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

Parameters

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

Chapter 6

Data Structure Documentation

6.1 display Struct Reference

elementy wchodzace w sklad kolejki pomiedzy zadaniem glownym a wyswietlaczem

Data Fields

- `uint8_t minute`
- `uint8_t hour`
- `uint8_t time`
- `uint8_t state`

6.1.1 Detailed Description

elementy wchodzace w sklad kolejki pomiedzy zadaniem glownym a wyswietlaczem

Parameters

<i>minute</i>	jest to zmienna okreslajaca minuty
<i>hour</i>	jest to zmienna okreslajaca godziny
<i>time</i>	jest to zmienna okreslajaca sekundy
<i>state</i>	jest to zmienna okreslajaca stan w jakim aktualnie sie znajduje (ustawianie/odliczanie)

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

- [main.c](#)

Chapter 7

File Documentation

7.1 main.c File Reference

plik zawierający główne funkcje oraz zadania programu

```
#include "stm32f1xx_hal.h"
#include "cmsis_os.h"
#include "gpio.h"
```

Data Structures

- struct [display](#)
elementy wchodzące w skład kolejki pomiędzy zadaniem głównym a wyświetlaczem

Macros

- #define [main_TASK_PRIORITY](#) (tskIDLE_PRIORITY + 1)
ustawienie priorytetu zadania które pozwala na "rownoległe" wykonywanie zadań jeśli jedno zadanie nic nie robi

Functions

- void [SystemClock_Config](#) (void)
Ustawienie parametrów zegara systemowego w poprawnego działania systemu freeRtos.
- void [vMain](#) (void *pvParameters)
Zadanie główne.
- void [vEncoder](#) (void *pvParameters)
Zadanie enkodera.
- void [vDisp](#) (void *pvParameters)
Zadanie wyświetlacza.
- void [vApplicationTickHook](#) (void)
Ustawienie parametrów zegara systemowego w celu odliczania 1 sekundy oraz sterowanie wyświetlaczami.
- int [main](#) (void)
Ładowanie całego systemu FreeRtos oraz inicjalizacja wykorzystywanych funkcji.

Variables

- xQueueHandle [queue_encoder](#)
Kolejka z enkodera przynosząca zmienna enc (więcej w [vMain](#))
- xQueueHandle [queue_wyswietlacz](#)
Kolejka z zadania vMain przynosząca strukturę [display](#).
- const unsigned char [seg7](#) [] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90}

Tablica konwersji wartosci BIN na kod wskaźnika siedmiosegmentowego.

- xSemaphoreHandle [Tim1s](#)

Semafor zależny od przerwania co 1 sekunde w celu odmierzenia czasu.

- volatile uint8_t [LED_buf](#) [4] = {0,0,0,0}

Wartosci poszczególnych wyświetlaczy.

- volatile uint8_t [LED_ptr](#)

Konkretny wyświetlacz

7.1.1 Detailed Description

plik zawierający główne funkcje oraz zadania programu

7.1.2 Macro Definition Documentation

7.1.2.1 main_TASK_PRIORITY

```
#define main_TASK_PRIORITY (tskIDLE_PRIORITY + 1)
```

ustawienie priorytetu zadania ktore pozwala na "rownloegle" wykonywanie zadan jesli jedno zadanie nic nie robi

- oczywiscie "rownloegle" nie oznacza ze zadania wykonuja sie rownolegle, tylko kolejno po sobie w niewielkich odstepach czasowych dajac złudne wrazenie rownoleglego wykonywania ustawienie takiej wartosci pozwala na wykonywanie kolejnych zadan kiedy aktualne zadanie nie wykonuje zadnych zadan

7.1.3 Function Documentation

7.1.3.1 main()

```
int main (
    void )
```

Ladowanie calego systemu FreeRtos oraz inicjalizacja wykorzystywanych funkcji.

Reset of all peripherals, Initializes the Flash interface and the SysTick.

Configure the system clock

Initialize all configured peripherals

Stworzenie kolejek

- Stworzenie pierwszej kolejki [queue_ekoder](#)
- Stworzenie drugiej kolejki [queue_wyswietlacz](#)

Stworzenie semafora

- Stworzenie semafora [Tim1s](#)

Stworzenie zadan

- Stworzenie pierwszego zadania [vMain](#)
- Stworzenie drugiego zadania [vEncoder](#)
- Stworzenie trzeciego zadania [vDisplay](#)

Start scheduler

7.1.3.2 SystemClock_Config()

```
void SystemClock_Config (
    void )
```

Ustawienie parametrow zegara systemowego w poprawnego dzialania systemu freeRtos.
System Clock Configuration

7.1.3.3 vApplicationTickHook()

```
void vApplicationTickHook (
    void )
```

Ustawienie parametrow zegara systemowego w celu odliczania 1 sekundy oraz sterowanie wyswietlaczami.

7.1.3.4 vDisp()

```
void vDisp (
    void * pvParameters )
```

Zadanie wyswietlacza.

Parameters

<i>time</i>	jest to czas w sekundach
<i>minute</i>	jest to czas w minutach
<i>hour</i>	jest to czas w godzinach

```
*/
    if (hour==0){
        LED_buf[3] = seg7[(minute %100) /10];
        LED_buf[2] = seg7[minute % 10];
        LED_buf[1] = seg7[(time %100) /10];
        LED_buf[0] = seg7[time % 10];
    }
    else
    {
        LED_buf[3] = seg7[0];
        LED_buf[2] = seg7[hour];
        LED_buf[1] = seg7[(minute %100) /10];
        LED_buf[0] = seg7[minute % 10];
    }
}
```

Mozna zauwazyc sposob zmiany zakresu wyswietlania

7.1.3.5 vEncoder()

```
void vEncoder (
    void * pvParameters )
```

Zadanie enkodera.

Parameters

<i>rotete1</i>	jest to pierwsza pozycja diagramu oraz jest to pierwszy styk enkodera
<i>rotete2</i>	jest to druga pozycja diagramu oraz jest to drugi styk enkodera
<i>state</i>	jest to stan w ktorym aktualnie sie znajdujemy
<i>enc</i>	jest to zmienna wpisywana do kolejki i idaca do zadania vMain <ul style="list-style-type: none"> W przypadku przejscia ze stanu 3 do 0 nastepuje wpisanie do zmiennej enc wartosci +1 W przypadku przejscia ze stanu 6 do 0 nastepuje wpisanie do zmiennej enc wartosci -1

7.1.3.6 vMain()

```
void vMain (
    void * pvParameters )
```

Zadanie glowne.

Parameters

<i>state</i>	
--------------	--

jesli parametr state jest rowny

- 0 to pobierana jest wartosc +1 lub -1 z kolejki a nastepnie obsluga nastawiania czasu
- 1 to odliczanie czasu pobranego z case 0 i wysylanie na wyswietlacz, w przypadku 00:00 przechodzi do case 2
- 2 to nastepuje generowanie sygnalu dzwiekowego i przejście do case 0

Parameters

<i>enc</i>	jest to wartosc mowiaca o inkrementacji lub dekrementacji podczas ustawiania czasu
<i>queue_wyswietlacz</i>	jest to kolejka opuszczajaca vMain i idaca do vDisp

7.1.4 Variable Documentation

7.1.4.1 LED_buf

```
volatile uint8_t LED_buf[4] = {0,0,0,0}
```

Wartosci poszczegolnych wyswietlaczy.

Table 7.5 Tabela uzytych wyjsc

Segment	Wyjscie	Wyswietlacz	Wyjscie
pd	PD7	LED4	PB14
g	PD6	LED3	PB9
f	PD5	LED2	PB5
e	PD4	LED1	PB1
d	PD7		
c	PD2		
b	PD1		
a	PD0		

Zmienne wykorzystywane do obslugi wyswietlacza LED

Parameters

<i>LED_buf</i>	jako wartosc
----------------	--------------

7.1.4.2 LED_ptr

```
volatile uint8_t LED_ptr
```

Konkretny wyświetlacz

Parameters

<code>LED_ptr</code>	jako numer wyświetlacza
----------------------	-------------------------

7.1.4.3 queue_ekoder

```
xQueueHandle queue_ekoder
```

Kolejka z enkodera przenoszaca zmienna enc (wiecej w [vMain](#))

7.1.4.4 queue_wyswietlacz

```
xQueueHandle queue_wyswietlacz
```

Kolejka z zadania vMain przenoszaca strukture [display](#).

7.1.4.5 seg7

```
const unsigned char seg7[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90}
```

Tablica konwersji wartosci BIN na kod wskaznika siedmiosegmentowego.

Table 7.8 Tabela znakow do wyswietlacza

0xC0	0xF9	0xA4	0xB0	0x99	0x92	0x82	0xF8	0x80	0x90
0	1	2	3	4	5	6	7	8	9

7.1.4.6 Tim1s

```
xSemaphoreHandle Tim1s
```

Semafor zalezny od przerwania co 1 sekunde w celu odmierzenia czasu.

7.2 stm32f1xx_hal_timebase_TIM.c File Reference

HAL time base based on the hardware TIM.

```
#include "stm32f1xx_hal.h"
```

```
#include "stm32f1xx_hal_tim.h"
```

Functions

- HAL_StatusTypeDef [HAL_InitTick](#) (uint32_t TickPriority)
This function configures the TIM6 as a time base source. The time source is configured to have 1ms time base with a dedicated Tick interrupt priority.
- void [HAL_SuspendTick](#) (void)
Suspend Tick increment.
- void [HAL_ResumeTick](#) (void)
Resume Tick increment.
- void [HAL_TIM_PeriodElapsedCallback](#) (TIM_HandleTypeDef *htim)

Period elapsed callback in non blocking mode.

Variables

- TIM_HandleTypeDef **htim6**
- uint32_t **uwIncrementState** = 0

7.2.1 Detailed Description

HAL time base based on the hardware TIM.

COPYRIGHT(c) 2016 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.

7.3 stm32f1xx_it.c File Reference

Interrupt Service Routines.

```
#include "stm32f1xx_hal.h"
#include "stm32f1xx.h"
#include "stm32f1xx_it.h"
#include "cmsis_os.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 **DebugMon_Handler** (void)
This function handles Debug monitor.
- void **SysTick_Handler** (void)

This function handles System tick timer.

- void **TIM6_IRQHandler** (void)

This function handles TIM6 global interrupt.

Variables

- TIM_HandleTypeDef **htim6**

7.3.1 Detailed Description

Interrupt Service Routines.

COPYRIGHT(c) 2016 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.

Index

display, [13](#)

HAL_InitTick

HAL_TimeBase, [9](#)

HAL_ResumeTick

HAL_TimeBase, [11](#)

HAL_SuspendTick

HAL_TimeBase, [11](#)

HAL_TIM_PeriodElapsedCallback

HAL_TimeBase, [11](#)

HAL_TimeBase, [9](#)

HAL_InitTick, [9](#)

HAL_ResumeTick, [11](#)

HAL_SuspendTick, [11](#)

HAL_TIM_PeriodElapsedCallback, [11](#)

LED_buf

main.c, [18](#)

LED_ptr

main.c, [18](#)

main

main.c, [16](#)

main.c, [15](#)

LED_buf, [18](#)

LED_ptr, [18](#)

main, [16](#)

main_TASK_PRIORITY, [16](#)

SystemClock_Config, [16](#)

vApplicationTickHook, [17](#)

vDisp, [17](#)

vEncoder, [17](#)

vMain, [18](#)

main_TASK_PRIORITY

main.c, [16](#)

STM32F7xx_HAL_Examples, [9](#)

SystemClock_Config

main.c, [16](#)

vApplicationTickHook

main.c, [17](#)

vDisp

main.c, [17](#)

vEncoder

main.c, [17](#)

vMain

main.c, [18](#)