

Biblioteka SysTimer

wersja 1

Wygenerowano za pomocą Doxygen 1.12.0

1 Opis projektu	1
2 Indeks plików	3
2.1 Lista plików	3
3 Dokumentacja plików	5
3.1 Dokumentacja pliku SysTimer/main.c	5
3.1.1 Opis szczegółowy	6
3.1.2 Dokumentacja definicji	6
3.1.2.1 F_CPU	6
3.1.3 Dokumentacja funkcji	6
3.1.3.1 main()	6
3.1.3.2 systimer_callback()	6
3.2 main.c	7
3.3 Dokumentacja pliku SysTimer/systimer.c	7
3.3.1 Opis szczegółowy	8
3.3.2 Dokumentacja funkcji	8
3.3.2.1 ISR()	8
3.3.2.2 systimer_init()	8
3.3.3 Dokumentacja zmiennych	9
3.3.3.1 PROGMEM	9
3.4 systimer.c	9
3.5 Dokumentacja pliku SysTimer/systimer.h	9
3.5.1 Opis szczegółowy	11
3.5.2 Dokumentacja definicji	11
3.5.2.1 F_CPU	11
3.5.2.2 SYSTIMER_ERROR_1	11
3.5.2.3 SYSTIMER_ERROR_2	11
3.5.2.4 SYSTIMER_NO_ERROR	11
3.5.3 Dokumentacja funkcji	12
3.5.3.1 systimer_callback()	12
3.5.3.2 systimer_init()	12
3.6 systimer.h	13
Skorowidz	15

Rozdział 1

Opis projektu

Tutaj wpisać autorów, treść projektu, założenia itp.

Formatowanie wybranych elementów jak w html (lub znacznikami DoxyGena). Szczegóły można znaleźć w dokumentacji DoxyGena, odnośnik tutaj [Dokumentacja DoxyGena](#).

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików wraz z ich krótkimi opisami:

SysTimer/ main.c	
Plik glowny programu	5
SysTimer/ systimer.c	
Plik implementacyjny biblioteki SysTimer	7
SysTimer/ systimer.h	
Plik naglowkowy biblioteki SysTimer	9

Rozdział 3

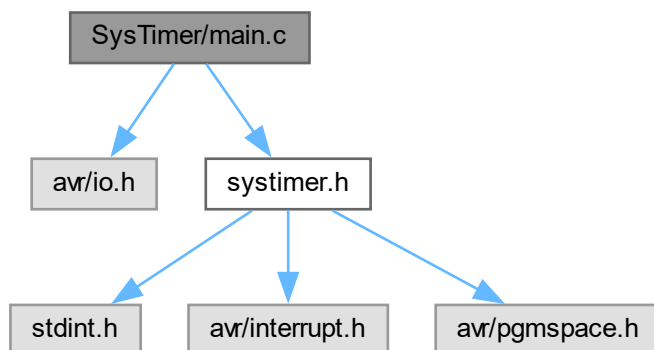
Dokumentacja plików

3.1 Dokumentacja pliku SysTimer/main.c

Plik główny programu.

```
#include <avr/io.h>
#include "systimer.h"
```

Wykres zależności załączania dla main.c:



Definicje

- `#define F_CPU 14745600UL`
definicja częstotliwości rezonatora kwarcowego

Funkcje

- `void systimer_callback (void)`
Funkcja wywoływana w programie obsługi przerwania od Timer1.
- `int main (void)`

3.1.1 Opis szczegółowy

Plik glowny programu.

Autor

B.W.

Data

2024/11/20

Definicja w pliku [main.c](#).

3.1.2 Dokumentacja definicji

3.1.2.1 F_CPU

```
#define F_CPU 14745600UL
```

definicja czestotliwosci rezonatora kwarcowego

Definicja w linii 11 pliku [main.c](#).

3.1.3 Dokumentacja funkcji

3.1.3.1 main()

```
int main (  
    void )
```

Definicja w linii 19 pliku [main.c](#).

3.1.3.2 systimer_callback()

```
void systimer_callback (  
    void )
```

Funkcja wywoływana w programie obsługi przerwania od Timer1.

Funkcja obsługi przerwania od porównania licznika Timer1. Zawiera wywołanie tzw. funkcji zwrotnej, która zdefiniowana w aplikacji będzie wywoływana cyklicznie co czas `systick`.

Definicja w linii 14 pliku [main.c](#).

3.2 main.c

[Idź do dokumentacji tego pliku.](#)

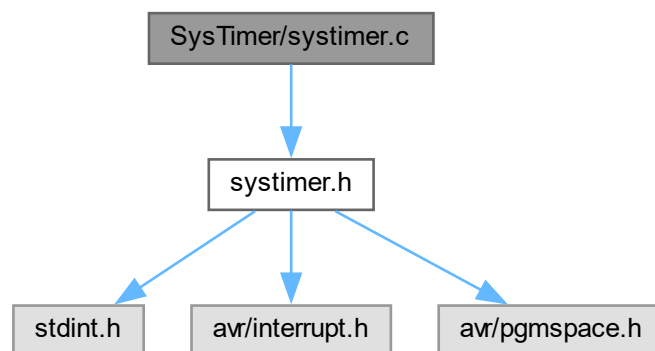
```
00001
00010 #include <avr/io.h>
00011 #define F_CPU 14745600UL
00012 #include "systimer.h"
00013
00014 void systimer_callback(void) /* wywołanie funkcji callback */
00015 {
00016     PORTD ^= 0b11000000; /* przy każdym wywołaniu negowane są bity PD7 i PD6 */
00017 }
00018
00019 int main(void)
00020 {
00021
00022     DDRD |= 0b11000000; /* dla testów wykorzystane zostaną wyprowadzenia PD7 i PD6 */
00023     PORTD |= 0b11000000; /* można do nich podłączyć np. diody LED */
00024
00025     systimer_init(231); /* wywołanie funkcji inicjalizującej SysTimer
00026                          z przykładową wartością (określa co jaki czas w ms
00027                          wywoływana będzie funkcja systimer_callback()) */
00028
00029     while (1) /* pętla nieskończona */
00030     {
00031     }
00032 }
```

3.3 Dokumentacja pliku SysTimer/systimer.c

Plik implementacyjny biblioteki SysTimer.

```
#include "systimer.h"
```

Wykres zależności załączania dla systimer.c:



Funkcje

- `uint8_t systimer_init(uint16_t systick)`
Funkcja pozwalająca skonfigurować licznik Timer1 tak by pozwalał na generację przerwania o określony czas.
- `ISR (TIMER1_COMPA_vect)`
Funkcja przerwania od licznika Timer1.

Zmienne

- `const uint16_t prescaler_div[5] PROGMEM = {1, 8, 64, 256, 1024}`

Tablica wartosci dzielnika czestotliwosci sygnalu zegarowego dla licznika Timer1.

3.3.1 Opis szczegółowy

Plik implementacyjny biblioteki SysTimer.

W pliku zawarto implementacje wszystkich funkcji biblioteki SysTimer. Inicjalizacja licznika odbywa sie automatycznie w oparciu o wartosc podanego czasu - interwalu pomiedzy wystapieniem kolejnych przerwan.

Definicja w pliku [systimer.c](#).

3.3.2 Dokumentacja funkcji

3.3.2.1 ISR()

```
ISR (
    TIMER1_COMPA_vect )
```

Funkcja przerwania od licznika Timer1.

Funkcja obslugi przerwania od porownania licznika Timer1. Zawiera wywołanie tzw. funkcji zwrotnej, ktora zdefiniowana w aplikacji bedzie wywoływana cyklicznie co czas `systick`.

Definicja w linii 63 pliku [systimer.c](#).

3.3.2.2 systimer_init()

```
uint8_t systimer_init (
    uint16_t systick)
```

Funkcja pozwalajaca skonfigurowac licznik Timer1 tak by pozwalal na generacje przerwania o okreslony czas.

Parametry

<code>systick</code>	wartosc czasu podana w [ms]
----------------------	-----------------------------

Zwraca

zwraca kod okreslajacy, czy licznik udalo sie poprawnie skonfigurowac i z jaka dokladnoscia bedzie odmierzany czas

Przykład wykorzystania

```
#include "systimer.h"

void main(void)
{
    systimer_init(100); // odmierzany interwal czasu 100ms

    while(1)
    {
    }
}

void systimer_callback()
{
    // tutaj wpisac instrukcje, ktore beda wykonywane co 100ms
}
```

Definicja w linii 23 pliku [systimer.c](#).

3.3.3 Dokumentacja zmiennych

3.3.3.1 PROGMEM

```
const uint16_t prescaler_div [5] PROGMEM = {1, 8, 64, 256, 1024}
```

Tablica wartosci dzielnika czestotliwosci sygnalu zegarowego dla licznika Timer1.

Stablicowane wartosci dzielnika czestotliwosci licznika Timer1. Wartosci umieszczone jako stale w pamieci programu FLAS.

Definicja w linii 21 pliku [systimer.c](#).

3.4 systimer.c

[Idź do dokumentacji tego pliku.](#)

```
00001
00011 #include "systimer.h"
00012
00021 const uint16_t prescaler_div[5] PROGMEM = {1, 8, 64, 256, 1024};
00022
00023 uint8_t systimer_init(uint16_t systick)
00024 {
00025     uint64_t _OCR1A;
00026     uint8_t index = 0;
00027
00028     /* wyszukiwanie wartosci dzielnika */
00029     do
00030     {
00031         _OCR1A = (((F_CPU / pgm_read_word(&prescaler_div[index]))) * systick) / 1000ULL;
00032     } while ((_OCR1A > 65535) && (index++ < 5));
00033
00034     /* sprawdzenie jak dokladnie udalo sie dobrac parametry licznika */
00035     uint8_t error_code = 0;
00036     if (index >= 5)
00037         error_code = SYSTIMER_ERROR_2; /* nie udalo sie znalezc dzielnika prescalera do odmierzania czasu */
00038
00039     else if (((F_CPU / pgm_read_word(&prescaler_div[index]))) * systick) / 1000ULL != 0)
00040         error_code = SYSTIMER_ERROR_1; /* dla podanych wartosci czas odmierzany z przyblizeniem */
00041     else
00042         error_code = SYSTIMER_NO_ERROR; /* licznik skonfigurowany poprawnie */
00043
00044     if (error_code < SYSTIMER_ERROR_2) /* wlasciwa konfiguracja licznika Timer1 */
00045     {
00046         OCR1A = (uint16_t) _OCR1A;
00047         TIMSK |= (1 << OCIE1A);
00048         sei();
00049         TCCR1B = (1 << WGM12) | (index + 1); /* start timer at index (1-5) prescaler */
00050     }
00051     return error_code;
00052 }
00053
00063 ISR(TIMER1_COMPA_vect)
00064 {
00065     systimer_callback();
00066 }
```

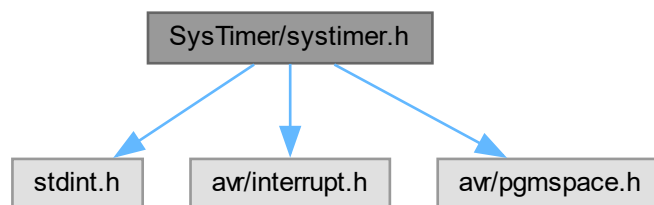
3.5 Dokumentacja pliku SysTimer/systimer.h

Plik naglowkowy biblioteki SysTimer.

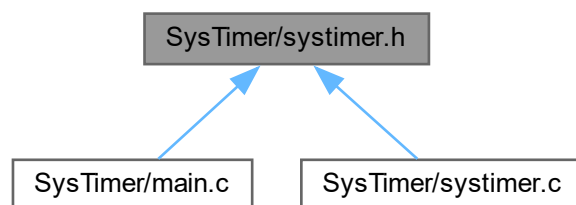
```
#include <stdint.h>
#include <avr/interrupt.h>
```

```
#include <avr/pgmspace.h>
```

Wykres zależności załączania dla systimer.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Definicje

- `#define F_CPU 14745600UL`
Definicja częstotliwości F_CPU.
- `#define SYSTIMER_NO_ERROR 0`
Licznik skonfigurowany poprawnie.
- `#define SYSTIMER_ERROR_1 1`
Licznik nie będzie dokładnie odmierzać czasu.
- `#define SYSTIMER_ERROR_2 2`
Nie udało się skonfigurować licznika.

Funkcje

- `uint8_t systimer_init (uint16_t systick)`
Funkcja pozwalająca skonfigurować licznik Timer1 tak by pozwalał na generację przerwania o określony czas.
- `void systimer_callback (void) __attribute__((weak))`
Funkcja wywoływana w programie obsługi przerwania od Timer1.

3.5.1 Opis szczegółowy

Plik naglowkowy biblioteki SysTimer.

W bibliotece zawarte zostały podstawowe funkcji pozwalające skonfigurować licznik Timer 1 mikrokontrolera w trybie CTC umożliwiając generowanie przerwania co określony interwał czasu (system tick).

Nota

wersja testowa

Definicja w pliku [systimer.h](#).

3.5.2 Dokumentacja definicji

3.5.2.1 F_CPU

```
#define F_CPU 14745600UL
```

Definicja częstotliwości F_CPU.

Definicja w linii 23 pliku [systimer.h](#).

3.5.2.2 SYSTIMER_ERROR_1

```
#define SYSTIMER_ERROR_1 1
```

Licznik nie będzie dokładnie odmierzać czasu.

Definicja w linii 28 pliku [systimer.h](#).

3.5.2.3 SYSTIMER_ERROR_2

```
#define SYSTIMER_ERROR_2 2
```

Nie udało się skonfigurować licznika.

Definicja w linii 29 pliku [systimer.h](#).

3.5.2.4 SYSTIMER_NO_ERROR

```
#define SYSTIMER_NO_ERROR 0
```

Licznik skonfigurowany poprawnie.

Definicja w linii 27 pliku [systimer.h](#).

3.5.3 Dokumentacja funkcji

3.5.3.1 systimer_callback()

```
void systimer_callback (
    void )
```

Funkcja wywoływana w programie obsługi przerwania od Timer1.

Funkcja obsługi przerwania od porównania licznika Timer1. Zawiera wywołanie tzw. funkcji zwrotnej, która zdefiniowana w aplikacji będzie wywoływana cyklicznie co czas `systick`.

Definicja w linii 14 pliku [main.c](#).

3.5.3.2 systimer_init()

```
uint8_t systimer_init (
    uint16_t systick)
```

Funkcja pozwalająca skonfigurować licznik Timer1 tak by pozwalał na generację przerwania o określony czas.

Parametry

<code>systick</code>	wartość czasu podana w [ms]
----------------------	-----------------------------

Zwraca

zwraca kod określający, czy licznik udało się poprawnie skonfigurować i z jaką dokładnością będzie odmierzany czas

Przykład wykorzystania

```
#include "systimer.h"

void main(void)
{
    systimer_init(100); // odmierzany interwał czasu 100ms

    while(1)
    {
    }
}

void systimer_callback()
{
    // tutaj wpisać instrukcje, które będą wykonywane co 100ms
}
```

Definicja w linii 23 pliku [systimer.c](#).

3.6 systimer.h

[Idź do dokumentacji tego pliku.](#)

```
00001
00012 #ifndef SYSTIMER_H_
00013 #define SYSTIMER_H_
00014
00015 #include <stdint.h>
00016 #include <avr/interrupt.h>
00017 #include <avr/pgmspace.h>
00018
00019 #ifndef F_CPU
00023 #define F_CPU 14745600UL
00024 #warning F_CPU is defined as 14745600UL
00025 #endif
00026
00027 #define SYSTIMER_NO_ERROR 0
00028 #define SYSTIMER_ERROR_1 1
00029 #define SYSTIMER_ERROR_2 2
00030
00058 uint8_t systimer_init(uint16_t systick);
00059
00069 void systimer_callback(void) __attribute__((weak));
00070
00079 #endif /* SYSTIMER_H_ */
00080
```


Skorowidz

F_CPU

main.c, [6](#)
systimer.h, [11](#)

ISR

systimer.c, [8](#)

main

main.c, [6](#)

main.c

F_CPU, [6](#)
main, [6](#)
systimer_callback, [6](#)

Opis projektu, [1](#)

PROGMEM

systimer.c, [9](#)

systimer.c

ISR, [8](#)
PROGMEM, [9](#)
systimer_init, [8](#)

systimer.h

F_CPU, [11](#)
systimer_callback, [12](#)
SYSTIMER_ERROR_1, [11](#)
SYSTIMER_ERROR_2, [11](#)
systimer_init, [12](#)
SYSTIMER_NO_ERROR, [11](#)

SysTimer/main.c, [5](#), [7](#)

SysTimer/systimer.c, [7](#), [9](#)

SysTimer/systimer.h, [9](#), [13](#)

systimer_callback

main.c, [6](#)
systimer.h, [12](#)

SYSTIMER_ERROR_1

systimer.h, [11](#)

SYSTIMER_ERROR_2

systimer.h, [11](#)

systimer_init

systimer.c, [8](#)
systimer.h, [12](#)

SYSTIMER_NO_ERROR

systimer.h, [11](#)