

Interfejs czujników

Wygenerowano przez Doxygen 1.8.5

Wt, 20 sty 2015 13:33:29

Spis treści

| | |
|--|----------|
| 1 Indeks plików | 1 |
| 1.1 Lista plików | 1 |
| 2 Dokumentacja plików | 1 |
| 2.1 Dokumentacja pliku config.c | 1 |
| 2.1.1 Opis szczegółowy | 1 |
| 2.1.2 Dokumentacja funkcji | 2 |
| 2.2 Dokumentacja pliku config.h | 2 |
| 2.2.1 Opis szczegółowy | 4 |
| 2.2.2 Dokumentacja definicji | 4 |
| 2.2.3 Dokumentacja funkcji | 6 |
| 2.3 Dokumentacja pliku sensors_interface.c | 7 |
| 2.3.1 Opis szczegółowy | 8 |
| 2.3.2 Dokumentacja funkcji | 8 |
| 2.3.3 Dokumentacja zmiennych | 9 |
| 2.4 Dokumentacja pliku sensors_interface.h | 10 |
| 2.4.1 Opis szczegółowy | 11 |
| 2.4.2 Dokumentacja funkcji | 11 |
| 2.5 Dokumentacja pliku usart.c | 12 |
| 2.5.1 Opis szczegółowy | 13 |
| 2.5.2 Dokumentacja funkcji | 13 |
| 2.5.3 Dokumentacja zmiennych | 14 |
| 2.6 Dokumentacja pliku usart.h | 14 |
| 2.6.1 Opis szczegółowy | 14 |
| 2.6.2 Dokumentacja definicji | 15 |
| 2.6.3 Dokumentacja funkcji | 15 |

1 Indeks plików

1.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

| | |
|--|----------|
| config.c | |
| Microcontroller configuration | 1 |
| config.h | |
| Microcontroller configuration | 2 |
| sensors_interface.c | |
| Main file of sensors interface program | 7 |

| | |
|--|----|
| sensors_interface.h | |
| Main file of sensors interface program | 10 |
| usart.c | |
| USART interface | 12 |
| usart.h | |
| USART interface and communication protocol | 14 |

2 Dokumentacja plików

2.1 Dokumentacja pliku config.c

Microcontroller configuration.

```
#include "config.h"
```

Funkcje

- void [InitPorts](#) ()
Microcontroller ports initializatin.
- void [wait_ms](#) (int ms)
Implements delay in ms.
- void [wait_us](#) (int us)
Implements delay in us.
- uint8_t [Checksum8](#) (uint8_t data)
Calculate checksum for 1-byte variable.
- uint8_t [Checksum16](#) (uint16_t data)
Calculate checksum for 2-byte variable.

2.1.1 Opis szczegółowy

Microcontroller configuration.

Autor

Michał Drwięga

Data

2014.11.28

Wersja

1.1

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

2.1.2 Dokumentacja funkcji

2.1.2.1 uint8_t CheckSum16 (uint16_t data)

Calculate checksum for 2-byte variable.

Definicja w linii 54 pliku config.c.

2.1.2.2 uint8_t CheckSum8 (uint8_t data)

Calculate checksum for 1-byte variable.

Definicja w linii 41 pliku config.c.

2.1.2.3 void InitPorts ()

Microcontroller ports initialization.

Definicja w linii 12 pliku config.c.

2.1.2.4 void wait_ms (int ms)

Implements delay in ms.

Definicja w linii 23 pliku config.c.

2.1.2.5 void wait_us (int us)

Implements delay in us.

Definicja w linii 32 pliku config.c.

2.2 Dokumentacja pliku config.h

Microcontroller configuration.

```
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/io.h>
```

Definicje

- #define F_CPU 8000000L
Microcontroller frequency in Hz.
- #define ADC_REF_CHANNEL ((0 << REFS1) | (1 << REFS0))
Selection of source of reference voltage for ADC – AVCC.
- #define ADMUX_CONF(mux) (mux | (0 << ADLAR) | ADC_REF_CHANNEL)
Macro for ADMUX settings.
- #define AREF_VALUE 5
Reference voltage value.
- #define VIN(ADC) (ADC * AREF_VALUE * 1000UL / 1024)
*Macro for scaling ADC measurement value in mV (VIN = ADC * VREF / 1024)*
- #define SONAR_PRESCALER 58
Sonar prescaler value for calculating distance from time of flight.
- #define MAX_PPM_WIDTH 3000
Pulse width which indicate that the new PPM frame arrived.
- #define SET_LED1_OUT DDRC |= _BV(2)
- #define SET_LED1 PORTC |= _BV(2)
- #define CLR_LED1 PORTC &= ~_BV(2)
- #define SET_LED2_OUT DDRC |= _BV(3)
- #define SET_LED2 PORTC |= _BV(3)
- #define CLR_LED2 PORTC &= ~_BV(3)
- #define SET_LED3_OUT DDRC |= _BV(4)
- #define SET_LED3 PORTC |= _BV(4)
- #define CLR_LED3 PORTC &= ~_BV(4)

- `#define SET_LED4_OUT DDRC |= _BV(5)`
- `#define SET_LED4 PORTC |= _BV(5)`
- `#define CLR_LED4 PORTC &= ~_BV(5)`
- `#define SET_LED5_OUT DDRC |= _BV(6)`
- `#define SET_LED5 PORTC |= _BV(6)`
- `#define CLR_LED5 PORTC &= ~_BV(6)`
- `#define SET_LED6_OUT DDRC |= _BV(7)`
- `#define SET_LED6 PORTC |= _BV(7)`
- `#define CLR_LED6 PORTC &= ~_BV(7)`
- `#define SET_ALL_LEDS PORTC |= 0xFE`
- `#define CLR_ALL_LEDS PORTC &= ~0xFE`
- `#define SET_SONAR1_EN_OUT DDRD |= _BV(4)`
- `#define SET_SONAR1_EN PORTD &= ~_BV(4)`
- `#define CLR_SONAR1_EN PORTD |= _BV(4)`
- `#define SET_SONAR2_EN_OUT DDRD |= _BV(5)`
- `#define SET_SONAR2_EN PORTD &= ~_BV(5)`
- `#define CLR_SONAR2_EN PORTD |= _BV(5)`
- `#define SET_SONAR1_IN DDRD &= ~_BV(2)`
- `#define SET_SONAR2_IN DDRD &= ~_BV(3)`
- `#define SET_PPM_IN DDRB &= ~_BV(2)`

Funkcje

- void `InitPorts` ()
Microcontroller ports initialization.
- void `wait_ms` (int ms)
Implements delay in ms.
- void `wait_us` (int us)
Implements delay in us.
- uint8_t `Checksum8` (uint8_t data)
Calculate checksum for 1-byte variable.
- uint8_t `Checksum16` (uint16_t data)
Calculate checksum for 2-byte variable.

2.2.1 Opis szczegółowy

Microcontroller configuration.

Autor

Michał Drwięga

Data

2014.10.12

Wersja

1.0

Definicja w pliku `config.h`.

2.2.2 Dokumentacja definicji

2.2.2.1 `#define ADC_REF_CHANNEL ((0 << REFS1) | (1 << REFS0))`

Selection of source of reference voltage for ADC – AVCC.

Definicja w linii 28 pliku config.h.

2.2.2.2 `#define ADMUX_CONF(mux)(mux | (0 << ADLAR) | ADC_REF_CHANNEL)`

Macro for ADMUX settings.

Definicja w linii 31 pliku config.h.

2.2.2.3 `#define AREF_VALUE 5`

Reference voltage value.

Definicja w linii 34 pliku config.h.

2.2.2.4 `#define CLR_ALL_LEDS PORTC &= ~0xFE`

Definicja w linii 74 pliku config.h.

2.2.2.5 `#define CLR_LED1 PORTC &= ~_BV(2)`

Definicja w linii 51 pliku config.h.

2.2.2.6 `#define CLR_LED2 PORTC &= ~_BV(3)`

Definicja w linii 55 pliku config.h.

2.2.2.7 `#define CLR_LED3 PORTC &= ~_BV(4)`

Definicja w linii 59 pliku config.h.

2.2.2.8 `#define CLR_LED4 PORTC &= ~_BV(5)`

Definicja w linii 63 pliku config.h.

2.2.2.9 `#define CLR_LED5 PORTC &= ~_BV(6)`

Definicja w linii 67 pliku config.h.

2.2.2.10 `#define CLR_LED6 PORTC &= ~_BV(7)`

Definicja w linii 71 pliku config.h.

2.2.2.11 `#define CLR_SONAR1_EN PORTD |= _BV(4)`

Definicja w linii 79 pliku config.h.

2.2.2.12 `#define CLR_SONAR2_EN PORTD |= _BV(5)`

Definicja w linii 83 pliku config.h.

2.2.2.13 `#define F_CPU 8000000L`

Microcontroller frequency in Hz.

Definicja w linii 13 pliku config.h.

2.2.2.14 #define MAX_PPM_WIDTH 3000

Pulse width which indicate that the new PPM frame arrived.

Definicja w linii 43 pliku config.h.

2.2.2.15 #define SET_ALL_LEDS PORTC |= 0xFE

Definicja w linii 73 pliku config.h.

2.2.2.16 #define SET_LED1 PORTC |= _BV(2)

Definicja w linii 50 pliku config.h.

2.2.2.17 #define SET_LED1_OUT DDRC |= _BV(2)

Definicja w linii 49 pliku config.h.

2.2.2.18 #define SET_LED2 PORTC |= _BV(3)

Definicja w linii 54 pliku config.h.

2.2.2.19 #define SET_LED2_OUT DDRC |= _BV(3)

Definicja w linii 53 pliku config.h.

2.2.2.20 #define SET_LED3 PORTC |= _BV(4)

Definicja w linii 58 pliku config.h.

2.2.2.21 #define SET_LED3_OUT DDRC |= _BV(4)

Definicja w linii 57 pliku config.h.

2.2.2.22 #define SET_LED4 PORTC |= _BV(5)

Definicja w linii 62 pliku config.h.

2.2.2.23 #define SET_LED4_OUT DDRC |= _BV(5)

Definicja w linii 61 pliku config.h.

2.2.2.24 #define SET_LED5 PORTC |= _BV(6)

Definicja w linii 66 pliku config.h.

2.2.2.25 #define SET_LED5_OUT DDRC |= _BV(6)

Definicja w linii 65 pliku config.h.

2.2.2.26 #define SET_LED6 PORTC |= _BV(7)

Definicja w linii 70 pliku config.h.

2.2.2.27 #define SET_LED6_OUT DDRC |= _BV(7)

Definicja w linii 69 pliku config.h.

2.2.2.28 #define SET_PPM_IN DDRB &= ~_BV(2)

Definicja w linii 89 pliku config.h.

2.2.2.29 `#define SET_SONAR1_EN PORTD &= ~_BV(4)`

Definicja w linii 78 pliku config.h.

2.2.2.30 `#define SET_SONAR1_EN_OUT DDRD |= _BV(4)`

Definicja w linii 77 pliku config.h.

2.2.2.31 `#define SET_SONAR1_IN DDRD &= ~_BV(2)`

Definicja w linii 85 pliku config.h.

2.2.2.32 `#define SET_SONAR2_EN PORTD &= ~_BV(5)`

Definicja w linii 82 pliku config.h.

2.2.2.33 `#define SET_SONAR2_EN_OUT DDRD |= _BV(5)`

Definicja w linii 81 pliku config.h.

2.2.2.34 `#define SET_SONAR2_IN DDRD &= ~_BV(3)`

Definicja w linii 86 pliku config.h.

2.2.2.35 `#define SONAR_PRESCALER 58`

Sonar prescaler value for calculating distance from time of flight.

Definicja w linii 40 pliku config.h.

2.2.2.36 `#define VIN(ADC) (ADC * AREF_VALUE * 1000UL / 1024)`

Macro for scaling ADC measurement value in mV ($VIN = ADC * VREF / 1024$)

Definicja w linii 37 pliku config.h.

2.2.3 Dokumentacja funkcji

2.2.3.1 `uint8_t CheckSum16 (uint16_t data)`

Calculate checksum for 2-byte variable.

Definicja w linii 54 pliku config.c.

2.2.3.2 `uint8_t CheckSum8 (uint8_t data)`

Calculate checksum for 1-byte variable.

Definicja w linii 41 pliku config.c.

2.2.3.3 `void InitPorts ()`

Microcontroller ports initialization.

Definicja w linii 12 pliku config.c.

2.2.3.4 `void wait_ms (int ms)`

Implements delay in ms.

Definicja w linii 23 pliku config.c.

2.2.3.5 void wait_us (int us)

Implements delay in us.

Definicja w linii 32 pliku config.c.

2.3 Dokumentacja pliku sensors_interface.c

Main file of sensors interface program.

```
#include "sensors_interface.h"
```

Funkcje

- [ISR](#) (TIMER0_COMP_vect)
Measurements interrupt.
- [ISR](#) (TIMER1_OVF_vect)
- [ISR](#) (TIMER2_OVF_vect)
- [ISR](#) (INT0_vect)
- [ISR](#) (INT1_vect)
- [ISR](#) (INT2_vect)
- void [StartMeasureSonarPW](#) (int nr)
External interrupts initialization for sonars measurements.
- void [InitTimer0](#) ()
Timer0 initialization for generate interrupts for synchronous measurements.
- void [InitTimer1](#) ()
Timer1 initialization to time measurement.
- void [InitTimer2](#) ()
Timer2 initialization to time measurement for PPM decoder.
- void [InitADC](#) ()
ADC initialization.
- uint16_t [MeasureADC](#) (uint8_t admuxSetting)
Make single measurement of voltage on selected ADC channel.
- void [UsartTransmitData](#) ()
- int [main](#) (void)

Zmienne

- volatile uint16_t [sharpVoltage](#) [7]
Voltages on SHARP sensors.
- volatile uint16_t [sonarValue](#) [2]
Distances measured by sonars.
- volatile uint8_t [txFlag](#) = 0
Flag indicate if send measurements is allowed.
- volatile uint8_t [cnt](#) = 0
Counter for calculating measure intervals.
- volatile uint16_t [timeCntPPM](#) = 0
Counter for calculating PPM pulses width.
- volatile uint8_t [channelPPMcnt](#) = 0
Indicates which channel of PPM signal is measured.
- volatile uint16_t [channelPPM](#) [8]
Store measured values for PPM channels.

2.3.1 Opis szczegółowy

Main file of sensors interface program.

Autor

Michał Drwiega

Data

2014.11.10

Wersja

1.2

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

2.3.2 Dokumentacja funkcji

2.3.2.1 void InitADC ()

ADC initialization.

Definicja w linii 186 pliku `sensors_interface.c`.

2.3.2.2 void InitTimer0 ()

Timer0 initialization for generate interrupts for synchronous measurements.

Definicja w linii 156 pliku `sensors_interface.c`.

2.3.2.3 void InitTimer1 ()

Timer1 initialization to time measurement.

Definicja w linii 167 pliku `sensors_interface.c`.

2.3.2.4 void InitTimer2 ()

Timer2 initialization to time measurement for PPM decoder.

Definicja w linii 177 pliku `sensors_interface.c`.

2.3.2.5 ISR (TIMER0_COMP_vect)

Measurements interrupt.

Definicja w linii 33 pliku `sensors_interface.c`.

2.3.2.6 ISR (TIMER1_OVF_vect)

Definicja w linii 51 pliku `sensors_interface.c`.

2.3.2.7 ISR (TIMER2_OVF_vect)

Definicja w linii 62 pliku `sensors_interface.c`.

2.3.2.8 ISR (INT0_vect)

Definicja w linii 68 pliku `sensors_interface.c`.

2.3.2.9 ISR (INT1_vect)

Definicja w linii 89 pliku `sensors_interface.c`.

2.3.2.10 ISR (INT2_vect)

Definicja w linii 108 pliku `sensors_interface.c`.

2.3.2.11 int main (void)

Definicja w linii 239 pliku `sensors_interface.c`.

2.3.2.12 uint16_t MeasureADC (uint8_t *admuxSetting*)

Make single measurement of voltage on selected ADC channel.

Parametry

| | |
|---------------------|--|
| <i>admuxSetting</i> | - ADMUX register setting for current measurement |
|---------------------|--|

Zwraca

Return measured value of voltage

Definicja w linii 198 pliku `sensors_interface.c`.

2.3.2.13 void StartMeasureSonarPW (int *nr*)

External interrupts initialization for sonars measurements.

Definicja w linii 137 pliku `sensors_interface.c`.

2.3.2.14 void UsartTransmitData ()

Definicja w linii 207 pliku `sensors_interface.c`.

2.3.3 Dokumentacja zmiennych

2.3.3.1 volatile uint16_t channelPPM[8]

Store measured values for PPM channels.

Definicja w linii 30 pliku `sensors_interface.c`.

2.3.3.2 volatile uint8_t channelPPMcnt = 0

Indicates which channel of PPM signal is measured.

Definicja w linii 27 pliku `sensors_interface.c`.

2.3.3.3 volatile uint8_t cnt = 0

Counter for calculating measure intervals.

Definicja w linii 21 pliku `sensors_interface.c`.

2.3.3.4 volatile uint16_t sharpVoltage[7]

Voltages on SHARP sensors.

Definicja w linii 12 pliku `sensors_interface.c`.

2.3.3.5 volatile uint16_t sonarValue[2]

Distances measured by sonars.

Definicja w linii 15 pliku sensors_interface.c.

2.3.3.6 volatile uint16_t timeCntPPM = 0

Counter for calculating PPM pulses width.

Definicja w linii 24 pliku sensors_interface.c.

2.3.3.7 volatile uint8_t txFlag = 0

Flag indicate if send measurements is allowed.

Definicja w linii 18 pliku sensors_interface.c.

2.4 Dokumentacja pliku sensors_interface.h

Main file of sensors interface program.

```
#include "config.h"
#include "usart.h"
```

Funkcje

- [ISR](#) (TIMER0_COMP_vect)
Measurements interrupt.
- [ISR](#) (TIMER1_OVF_vect)
- [ISR](#) (TIMER2_OVF_vect)
- [ISR](#) (INT0_vect)
- [ISR](#) (INT1_vect)
- [ISR](#) (INT2_vect)
- void [InitTimer0](#) ()
Timer0 initialization for generate interrupts for synchronous measurements.
- void [InitTimer1](#) ()
Timer1 initialization to time measurement.
- void [InitTimer2](#) ()
Timer2 initialization to time measurement for PPM decoder.
- void [StartMeasureSonarPW](#) (int nr)
External interrupts initialization for sonars measurements.
- void [InitADC](#) ()
ADC initialization.
- uint16_t [MeasureADC](#) (uint8_t admuxSetting)
Make single measurement of voltage on selected ADC channel.
- void [UsartTransmitData](#) ()

2.4.1 Opis szczegółowy

Main file of sensors interface program.

Autor

Michał Drwiega

Data

2014.11.10

Wersja

1.2

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

2.4.2 Dokumentacja funkcji**2.4.2.1 void InitADC ()**

ADC initialization.

Definicja w linii 186 pliku `sensors_interface.c`.

2.4.2.2 void InitTimer0 ()

Timer0 initialization for generate interrupts for synchronous measurements.

Definicja w linii 156 pliku `sensors_interface.c`.

2.4.2.3 void InitTimer1 ()

Timer1 initialization to time measurement.

Definicja w linii 167 pliku `sensors_interface.c`.

2.4.2.4 void InitTimer2 ()

Timer2 initialization to time measurement for PPM decoder.

Definicja w linii 177 pliku `sensors_interface.c`.

2.4.2.5 ISR (TIMER0_COMP_vect)

Measurements interrupt.

Definicja w linii 33 pliku `sensors_interface.c`.

2.4.2.6 ISR (TIMER1_OVF_vect)

Definicja w linii 51 pliku `sensors_interface.c`.

2.4.2.7 ISR (TIMER2_OVF_vect)

Definicja w linii 62 pliku `sensors_interface.c`.

2.4.2.8 ISR (INT0_vect)

Definicja w linii 68 pliku `sensors_interface.c`.

2.4.2.9 ISR (INT1_vect)

Definicja w linii 89 pliku `sensors_interface.c`.

2.4.2.10 ISR (INT2_vect)

Definicja w linii 108 pliku `sensors_interface.c`.

2.4.2.11 `uint16_t MeasureADC (uint8_t admuxSetting)`

Make single measurement of voltage on selected ADC channel.

Parametry

| | |
|---------------------|--|
| <i>admuxSetting</i> | - ADMUX register setting for current measurement |
|---------------------|--|

Zwraca

Return measured value of voltage

Definicja w linii 198 pliku `sensors_interface.c`.

2.4.2.12 void StartMeasureSonarPW (int nr)

External interrupts initialization for sonars measurements.

Definicja w linii 137 pliku `sensors_interface.c`.

2.4.2.13 void UsartTransmitData ()

Definicja w linii 207 pliku `sensors_interface.c`.

2.5 Dokumentacja pliku `usart.c`

USART interface.

```
#include "usart.h"
```

Funkcje

- void `InitUsart` ()
- void `UsartTransmit1byte` (uint8_t data)
Send one byte of data by USART interface.
- void `UsartTransmit1byteAndChecksum` (uint8_t data)
- void `UsartTransmit2byte` (uint16_t data)
Send 2-byte variable by usart interface.
- void `UsartTransmitData16` (uint8_t order, uint16_t data)
- uint8_t `USART_Receive` (void)
USART data receive.
- `ISR` (USART_RXC_vect)
Interrupt from received data by USART.

Zmienne

- uint8_t `txFlag`
Flag indicate if send measurements is allowed.

2.5.1 Opis szczegółowy

USART interface.

Autor

Michał Drwięga

Data

2014.10.12

Wersja

1.0

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

2.5.2 Dokumentacja funkcji**2.5.2.1 void InitUsart ()**

Definicja w linii 18 pliku usart.c.

2.5.2.2 ISR (USART_RXC_vect)

Interrupt from received data by USART.

Definicja w linii 94 pliku usart.c.

2.5.2.3 uint8_t USART_Receive (void)

USART data receive.

Definicja w linii 82 pliku usart.c.

2.5.2.4 void UsartTransmit1byte (uint8_t data)

Send one byte of data by USART interface.

Definicja w linii 32 pliku usart.c.

2.5.2.5 void UsartTransmit1byteAndChecksum (uint8_t data)

Definicja w linii 39 pliku usart.c.

2.5.2.6 void UsartTransmit2byte (uint16_t data)

Send 2-byte variable by usart interface.

Use Big Endian convention, czyli bardziej znaczący bajt jako pierwszy.

Definicja w linii 51 pliku usart.c.

2.5.2.7 void UsartTransmitData16 (uint8_t order, uint16_t data)

Definicja w linii 60 pliku usart.c.

2.5.3 Dokumentacja zmiennych**2.5.3.1 uint8_t txFlag**

Flag indicate if send measurements is allowed.

Definicja w linii 18 pliku sensors_interface.c.

2.6 Dokumentacja pliku usart.h

USART interface and communication protocol.


```
#include "config.h"
#include "sensors_interface.h"
```

Definicje

- `#define USART_BAUDRATE 9600`
Baudrate speed value.
- `#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)`
Calculation of baudrate speed register value.

Funkcje

- `void InitUsart ()`
- `void UsartTransmit1byte (uint8_t data)`
Send one byte of data by USART interface.
- `void UsartTransmit2byte (uint16_t data)`
Send 2-byte variable by usart interface.
- `uint8_t USART_Receive (void)`
USART data receive.
- `ISR (USART_RXC_vect)`
Interrupt from received data by USART.

2.6.1 Opis szczegółowy

USART interface and communication protocol.

Autor

Michał Drwiega

Data

2014.10.12

Wersja

1.0

W pierwszym odebranym bajcie danych dwa najstarsze bity są bitami kontrolnymi o wartościach wysokiego stanu logicznego (11). Kolejne 6 bitów określa rozkaz, który należy wykonać. W ten sposób można zakodować do 64 różnych rozkazów, co wydaje się być wartością zupełnie wystarczającą. Zaimplementowany protokół zakłada przysyłanie bardziej znaczących bajtów przed mniej znaczącymi (Big endian). Rozkazy podzielono na cztery grupy w zależności od wartości pierwszych czterech bitów:

- 0xC (1100) - żądania przesłania informacji zwrotnej.
- 0xD (1101) - rozkazy konfiguracyjne.

0xE (1110) - pozostałe rozkazy.

- 0xF (1111) - pozostałe rozkazy.

Poniżej znajduje się lista rozkazów z opisem.

1100 0000 - 0xC0 Wstępna weryfikacja poprawności działania transmisji. Oczekuje się odpowiedzi w postaci 0101 0101 - 0x55.

1100 1001 - 0xC1 Polecenie wysłania odczytanych wartości z sensorów odległości. Kolejność wysyłanych pomiarów jest następująca:

- 2 bajty - napięcie w mV odczytane z sensora Sharp 1,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 2,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 3,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 4,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 5,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 6,
- 2 bajty - napięcie w mV odczytane z sensora Sharp 7,
- 2 bajty - zmierzona odległość za pomocą sonaru nr 1,
- 2 bajty - zmierzona odległość za pomocą sonaru nr 2,
- 16 bajtów - wartości zmierzone dla poszczególnych kanałów PPM,

1110 0000 - 0xD0 Polecenie uruchomienia oświetlenia quadcoptera. 1110 0000 - 0xD1 Polecenie wyłączenia oświetlenia quadcoptera.

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

2.6.2 Dokumentacja definicji

2.6.2.1 #define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

Calculation of baudrate speed register value.

Definicja w linii 63 pliku usart.h.

2.6.2.2 #define USART_BAUDRATE 9600

Baudrate speed value.

Definicja w linii 60 pliku usart.h.

2.6.3 Dokumentacja funkcji

2.6.3.1 void InitUsart ()

Definicja w linii 18 pliku usart.c.

2.6.3.2 ISR (USART_RXC_vect)

Interrupt from received data by USART.

Definicja w linii 94 pliku usart.c.

2.6.3.3 uint8_t USART_Receive (void)

USART data receive.

Definicja w linii 82 pliku usart.c.

2.6.3.4 void UsartTransmit1byte (uint8_t data)

Send one byte of data by USART interface.

Definicja w linii 32 pliku usart.c.

2.6.3.5 void UsartTransmit2byte (uint16_t data)

Send 2-byte variable by usart interface.

Use Big Endian convention, czyli bardziej znaczący bajt jako pierwszy.

Definicja w linii 51 pliku usart.c.