KommunikasjonsProtokoll

Generated by Doxygen 1.8.14

Contents

Chapter 1

My Personal Index Page

1.1 Introduction

This is the introduction.

1.2 Installation

1.2.1 Step 1: Opening the box

etc...

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

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

Chapter 4

Class Documentation

4.1 circular_buf_t Struct Reference

Public Attributes

- uint8_t * **buffer**
- uint8_t volatile **head**
- uint8_t volatile **tail**
- uint8_t **size**

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

- main.c

4.2 packet_data Struct Reference

Public Attributes

- uint8_t volatile **mainComm_Counter**
- uint8_t volatile **subComm_Counter**
- uint8_t volatile **maxMainComms**
- uint16_t **crc16**

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

- main.c

Chapter 5

File Documentation

5.1 comm.c File Reference

All the configurations for SPI and I2C Author: Christoffer Boothby. Version: 0.0.1.2. Comments:.

```
#include <stdint.h>
#include "comm.h"
#include <avr/io.h>
#include <util/twi.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <util/atomic.h>
```

Macros

- #define **PORT_SPI** PORTB
- #define **DDR_SPI** DDRB
- #define **DD_MISO** DDRB4
- #define **DD_MOSI** DDRB3
- #define **DD_SCK** DDRB5
- #define **TWI_FREQ** 2000
- #define **Prescaler** 64

Functions

- void **spi_init_adc** ()
- void **spiTransmitADC_1** (uint8_t ∗dataout, uint8_t datain)
- void **spiTransmitADC_2** (uint8_t ∗dataout, uint8_t datain)
- void **spi_init_dac** ()
- void **spiTransmitDAC_1** (uint8_t dacAdress, uint8_t dacData)
- void **spiTransmitDAC_2** (uint8_t dacAdress, uint8_t dacData)
- void **i2c_init** (void)
- uint8_t **i2c_start** (uint8_t address)
- uint8_t **i2c_write** (uint8_t data)
- uint8_t **i2c_read_ack** (void)
- uint8_t **i2c_read_nack** (void)
- uint8_t **i2c_transmit** (uint8_t address, uint8_t ∗data, uint16_t length)
- uint8_t **i2c_receive** (uint8_t address, uint8_t ∗data, uint16_t length)
- uint8_t **i2c_writeReg** (uint8_t devaddr, uint8_t regaddr, uint8_t ∗data, uint16_t length)
- uint8_t **i2c_readReg** (uint8_t devaddr, uint8_t regaddr, uint8_t ∗data, uint16_t length)
- void **i2c_stop** (void)

5.1.1 Detailed Description

All the configurations for SPI and I2C Author: Christoffer Boothby. Version: 0.0.1.2. Comments:.

5.2 comm.h File Reference

All the defines for the COMM.c usage. Author: Christoffer Boothby. Version: 0.0.1.2. Comments:.

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

Macros

- #define **COMM_H_**
- #define **F_CPU** (14745600UL)
- #define **ADV_CONVERSION_START_1** DDRE2
- #define **ADC_READ_1** DDRE3
- #define **ADC_1_BUSY** DDRC0
- #define **ADV_CONVERSION_START_2** DDRB6
- #define **ADC_READ_2** DDD7
- #define **ADC_2_BUSY** DDRB0
- #define **CS_DAC_1** DDRC1
- #define **LD_DAC_1** DDRC2
- #define **CS_DAC_2** DDRB1
- #define **LD_DAC_2** DDRB2
- #define **LTC1859_CH0** 0b10000100
- #define **LTC1859_CH1** 0b11010100
- #define **LTC1859_CH2** 0b10010100
- #define **LTC1859_CH3** 0b11010100
- #define **LTC1859_CH4** 0b10100100
- #define **LTC1859_CH5** 0b11100100
- #define **LTC1859_CH6** 0b10110100
- #define **LTC1859_CH7** 0b11110100
- #define **G1_BIAS_1** (uint16_t)0xFFF
- #define **G2_BIAS_1** (uint16_t)0xFFF
- #define **G3_BIAS_1** (uint16_t)0xFFF
- #define **G1_BIAS_2** (uint16_t)0xFFF
- #define **G2_BIAS_2** (uint16_t)0xFFF
- #define **G3_BIAS_2** (uint16_t)0xFFF
- #define **DAC_B** 0x4
- #define **DAC_C** 0x8
- #define **DAC_D** 0xC
- #define **U7_ADDR** 0xD2
- #define **U8_ADDR** 0xCE
- #define **U9_ADDR** 0xDE
- #define **I2C_READ** 0x01
- #define **I2C_WRITE** 0x00

Functions

- void **spi_init_dac** ()
- void **spi_init_adc** ()
- void **spiTransmitADC_1** (uint8_t *dataout, uint8_t datain)
- void **spiTransmitADC_2** (uint8_t *dataout, uint8_t datain)
- void **spiTransmitDAC_1** (uint8_t dacAdress, uint8_t dacData)
- void **spiTransmitDAC_2** (uint8_t dacAdress, uint8_t dacData)
- void **i2c_init** (void)
- uint8_t **i2c_start** (uint8_t address)
- uint8_t **i2c_write** (uint8_t data)
- uint8_t **i2c_read_ack** (void)
- uint8_t **i2c_read_nack** (void)
- uint8_t **i2c_transmit** (uint8_t address, uint8_t *data, uint16_t length)
- uint8_t **i2c_receive** (uint8_t address, uint8_t *data, uint16_t length)
- uint8_t **i2c_writeReg** (uint8_t devaddr, uint8_t regaddr, uint8_t *data, uint16_t length)
- uint8_t **i2c_readReg** (uint8_t devaddr, uint8_t regaddr, uint8_t *data, uint16_t length)
- void **i2c_stop** (void)

5.2.1 Detailed Description

All the defines for the COMM.c usage. Author: Christoffer Boothby. Version: 0.0.1.2. Comments:.

5.3 main.c File Reference

Main program for the G-Chaser Project using Atmega328PB. Author: Christoffer Boothby Version: 0.0.1.↩ 2 Comments:

```
#include "comm.h"
#include <stdint.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdbool.h>
#include <stdlib.h>
#include <util/crc16.h>
#include <util/delay.h>
#include <util/twi.h>
#include <util/atomic.h>
#include <util/setbaud.h>
```

Classes

- struct circular_buf_t
- struct packet_data

Macros

- #define **BAUD** 1200
- #define **UART_BUFFER_SIZE** 128
- #define **UART_TX0_MAXBUFFER** (UART_BUFFER_SIZE-1)

Functions

- void circular_buf_put (circular_buf_t ∗cbuf, packet_data ∗pData, uint8_t data)

  Putting 1 byte into the buffer.
- void Port_Init ()

  Port initalization function.
- void **USART_Init** ()
- **ISR** (USART0_UDRE_vect)
- void **subCommFormat** (circular_buf_t ∗cbuf, packet_data ∗pData)
- void **packetFormat** (circular_buf_t ∗cbuf, packet_data ∗pData)
- int **main** (void)

Variables

- circular_buf_t **cbuf**
- packet_data **pData**
- uint8_t **array** [UART_BUFFER_SIZE]
- uint8_t **testvalue**

5.3.1 Detailed Description

Main program for the G-Chaser Project using Atmega328PB. Author: Christoffer Boothby Version: 0.0.1.↩ 2 Comments:

5.3.2 Function Documentation

5.3.2.1 circular_buf_put()

```
void circular_buf_put (
            circular_buf_t * cbuf,
            packet_data * pData,
            uint8_t data )
```

Putting 1 byte into the buffer.

Parameters

in	*data*	The data that goes into the buffer.
in	*cbuf*	refrence to the circular buffer.
in	*pdata*	refrence for pdata.

Returns

None.

5.3.2.2 Port_Init()

```
void Port_Init ( )
```

Port initalization function.

Parameters

None.	

Returns

None.

Index