TTK4155

Generated by Doxygen 1.9.2

Chapter 1

User Application template doxygen documentation

Empty user application template

Bare minimum empty user application template

Content

1. Include the ASF header files (through asf.h)
2. "Insert system clock initialization code here" comment
3. Minimal main function that starts with a call to board_init()
4. "Insert application code here" comment

Chapter 2

TTK4155 - Industrial and Embedded Computer Systems Design

2.0.1 Term project

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Data Structure Documentation

5.1 CAN_message Struct Reference

`#include <CAN.h>`

Data Fields

- uint16_t length
- uint16_t id
- int8_t data [8]

5.1.1 Field Documentation

5.1.1.1 data

`int8_t data[8]`

5.1.1.2 id

`uint16_t id`

5.1.1.3 length

`uint16_t length`

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

- D:/Progetti/TTK4155/Node1/TTK4155/CAN.h

5.2 can_message_t Struct Reference

Structure of the CAN message.

`#include <can_controller.h>`

Data Fields

- uint16_t id
- uint16_t data_length
- int8_t data [8]

5.2.1 Detailed Description

Structure of the CAN message.

5.2.2 Field Documentation

5.2.2.1 data

`int8_t data[8]`

5.2.2.2 data_length

`uint16_t data_length`

5.2.2.3 id

`uint16_t id`

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

- D:/Progetti/TTK4155/Node2/Node2/can_controller.h

5.3 direction Struct Reference

`#include <Joystick.h>`

Data Fields

- char x_dir
- char y_dir

5.3.1 Field Documentation

5.3.1.1 x_dir

```
char x_dir
```

5.3.1.2 y_dir

```
char y_dir
```

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

- D:/Progetti/TTK4155/Node1/TTK4155/Joystick.h

5.4 menu_entry Struct Reference

```
#include <OLED.h>
```

Data Fields

- char * name
- uint8_t length

5.4.1 Field Documentation

5.4.1.1 length

```
uint8_t length
```

5.4.1.2 name

```
char* name
```

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

- D:/Progetti/TTK4155/Node1/TTK4155/OLED.h

5.5 PID_DATA Struct Reference

```
#include <PI.h>
```

Data Fields

- int16_t error
- int32_t integral
- int16_t P_Factor
- int32_t I_Factor

5.5.1 Field Documentation

5.5.1.1 error

```
int16_t error
```

5.5.1.2 I_Factor

```
int32_t I_Factor
```

5.5.1.3 integral

```
int32_t integral
```

5.5.1.4 P_Factor

```
int16_t P_Factor
```

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

- D:/Progetti/TTK4155/Node2/Node2/PI.h

5.6 pos_t Struct Reference

```
#include <Joystick.h>
```

Data Fields

- uint8_t x
- uint8_t y

5.6.1 Field Documentation

5.6.1.1 x

```
uint8_t x
```

5.6.1.2 y

```
uint8_t y
```

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

- D:/Progetti/TTK4155/Node1/TTK4155/Joystick.h

5.7 sliderpos_t Struct Reference

Data type for sliders position L (left), R (right)

```
#include <Slider.h>
```

Data Fields

- uint8_t L
- uint8_t R

5.7.1 Detailed Description

Data type for sliders position L (left), R (right)

5.7.2 Field Documentation

5.7.2.1 L

```
uint8_t L
```

5.7.2.2 R

```
uint8_t R
```

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

- D:/Progetti/TTK4155/Node1/TTK4155/Slider.h

5.8 uart_ringbuffer_t Struct Reference

```
#include <uart.h>
```

Data Fields

- uint8_t head
- uint8_t tail
- char data [UART_RINGBUFFER_SIZE]

5.8.1 Field Documentation

5.8.1.1 data

```
char data[UART_RINGBUFFER_SIZE]
```

5.8.1.2 head

```
uint8_t head
```

5.8.1.3 tail

```
uint8_t tail
```

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

- D:/Progetti/TTK4155/Node2/Node2/uart.h

Chapter 6

File Documentation

6.1 D:/Progetti/TTK4155/Node1/TTK4155/ADC.c File Reference

```
#include "ADC.h"
```

Functions

- void adc_init ()
  Initialization function of the ADC.
- uint8_t adc_read (uint8_t channel)
  Reading data from the ADC.
- void adc_read_all_channels ()
  Debug function to read all the channels from the ADC.

Variables

- ext_adc = ADC_ADDRESS

6.1.1 Function Documentation

6.1.1.1 adc_init()

```
void adc_init ( )
```

Initialization function of the ADC.

Initializes the ADC and its clock signal.

6.1.1.2 adc_read()

```
uint8_t adc_read (
            uint8_t channel )
```

Reading data from the ADC.

Reads a certain analog channel of the ADC.

Parameters

channel	

Returns

uint8_t

6.1.1.3 adc_read_all_channels()

```
void adc_read_all_channels ( )
```

Debug function to read all the channels from the ADC.

6.1.2 Variable Documentation

6.1.2.1 ext_adc

```
ext_adc = ADC_ADDRESS
```

6.2 D:/Progetti/TTK4155/Node2/Node2/ADC.c File Reference

```
#include "ADC.h"
```

Functions

- void ADC_init ()
  Initialization function for the ADC.
- int ADC_read ()
  Read data for the ADC.
- int ADC_check_goal ()
  Check if the user has scored a goal.
- void ADC_print ()
  Debug function that prints ADC data.

6.2.1 Function Documentation

6.2.1.1 ADC_check_goal()

```
int ADC_check_goal ( )
```

Check if the user has scored a goal.

Returns

int Return 1 if the goal has been scored

6.2.1.2 ADC_init()

```
void ADC_init ( )
```

Initialization function for the ADC.

6.2.1.3 ADC_print()

```
void ADC_print ( )
```

Debug function that prints ADC data.

6.2.1.4 ADC_read()

```
int ADC_read ( )
```

Read data for the ADC.

Returns

int

6.3 D:/Progetti/TTK4155/Node1/TTK4155/ADC.h File Reference

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

Macros

- #define ADC_ADDRESS 0x1400
- #define SINGLE_CHANNEL_SAMPLE 7

Functions

- void adc_init (void)

  Initializes the ADC and its clock signal.
- uint8_t adc_read (uint8_t channel)

  Reads a certain analog channel of the ADC.
- void adc_read_all_channels ()

  Debug function to read all the channels from the ADC.

Variables

- volatile char * ext_adc

  Pointer to the array of all addresses we will write to. ADC starting at 0x1400.
- volatile char ADC_data

  Data retrieved from the ADC.

6.3.1 Macro Definition Documentation

6.3.1.1 ADC_ADDRESS

```
#define ADC_ADDRESS 0x1400
```

6.3.1.2 SINGLE_CHANNEL_SAMPLE

```
#define SINGLE_CHANNEL_SAMPLE 7
```

6.3.2 Function Documentation

6.3.2.1 adc_init()

```
void adc_init (
            void  )
```

Initializes the ADC and its clock signal.

Initialization function of the ADC.

Initializes the ADC and its clock signal.

6.3.2.2 adc_read()

```
uint8_t adc_read (
            uint8_t channel )
```

Reads a certain analog channel of the ADC.

Reading data from the ADC.

Parameters

channel	

Returns

uint8_t

Reads a certain analog channel of the ADC.

Parameters

channel	

Returns

uint8_t

6.3.2.3 adc_read_all_channels()

```
void adc_read_all_channels ( )
```

Debug function to read all the channels from the ADC.

6.3.3 Variable Documentation

6.3.3.1 ADC_data

```
volatile char ADC_data
```

Data retrieved from the ADC.

6.3.3.2 ext_adc

```
volatile char* ext_adc
```

Pointer to the array of all addresses we will write to. ADC starting at 0x1400.

6.4 ADC.h

Go to the documentation of this file.

```
1 #ifndef ADC_H
2 #define ADC_H
3
4 #include <avr/io.h>
5 #include <avr/interrupt.h>
6 #include <avr/delay.h>
7
8 #define ADC_ADDRESS 0x1400
9
10 #define SINGLE_CHANNEL_SAMPLE 7
11
16 volatile char* ext_adc;
17
22 volatile char ADC_data;
23
28 void adc_init(void);
29
36 uint8_t adc_read(uint8_t channel);
37
42 void adc_init();
43
50 uint8_t adc_read(uint8_t channel);
51
56 void adc_read_all_channels();
57
58 #endif
```

6.5 D:/Progetti/TTK4155/Node2/Node2/ADC.h File Reference

```
#include <stdio.h>
#include "sam.h"
```

Macros

- #define ADC_LIMIT 2000

Functions

- void ADC_init ()

  Initialization function for the ADC.
- int ADC_read ()

  Read data for the ADC.
- int ADC_check_goal ()

  Check if the user has scored a goal.
- void ADC_print ()

  Debug function that prints ADC data.

6.5.1 Macro Definition Documentation

6.5.1.1 ADC_LIMIT

```
#define ADC_LIMIT 2000
```

6.5.2 Function Documentation

6.5.2.1 ADC_check_goal()

```
int ADC_check_goal ( )
```

Check if the user has scored a goal.

Returns

int Return 1 if the goal has been scored

6.5.2.2 ADC_init()

```
void ADC_init ( )
```

Initialization function for the ADC.

6.5.2.3 ADC_print()

```
void ADC_print ( )
```

Debug function that prints ADC data.

6.5.2.4 ADC_read()

```
int ADC_read ( )
```

Read data for the ADC.

Returns

int

6.6 ADC.h

Go to the documentation of this file.

```
1 #ifndef ADC_H
2 #define ADC_H
3
4 #include <stdio.h>
5 #include "sam.h"
6
7 #define ADC_LIMIT 2000
8
13 void ADC_init();
14
20 int ADC_read();
21
27 int ADC_check_goal();
28
33 void ADC_print();
34
35 #endif
```

6.7 D:/Progetti/TTK4155/Node1/TTK4155/CAN.c File Reference

```
#include "CAN.h"
```

Functions

- void CAN_init (uint8_t mode)

  Initialization function for the CAN bus.
- void CAN_send (CAN_message message)

  Send data though the CAN bus.
- void CAN_receive (void)

  Checks registers for received messages.
- void CAN_debug_print (uint8_t id, char data, uint8_t position)

  Debug function for printing incoming messages.

6.7.1 Function Documentation

6.7.1.1 CAN_debug_print()

```
void CAN_debug_print (
            uint8_t id,
            char data,
            uint8_t position )
```

Debug function for printing incoming messages.

Parameters

id	
data	
position	

6.7.1.2 CAN_init()

```
void CAN_init (
            uint8_t mode )
```

Initialization function for the CAN bus.

Parameters

mode	

6.7.1.3 CAN_receive()

```
void CAN_receive (
            void  )
```

Checks registers for received messages.

6.7.1.4 CAN_send()

```
void CAN_send (
            CAN_message message )
```

Send data though the CAN bus.

Parameters

message	

6.8 D:/Progetti/TTK4155/Node1/TTK4155/CAN.h File Reference

```
#include "CAN_Controller.h"
#include "avr/interrupt.h"
```

Data Structures

- struct CAN_message

Functions

- void CAN_init (uint8_t mode)

  Initialization function for the CAN bus.
- void CAN_send (CAN_message message)

  Send data though the CAN bus.
- void CAN_receive ()

  Checks registers for received messages.
- void CAN_debug_print (uint8_t id, char data, uint8_t position)

  Debug function for printing incoming messages.
- ISR (INT0_vect)

6.8.1 Function Documentation

6.8.1.1 CAN_debug_print()

```
void CAN_debug_print (
            uint8_t id,
            char data,
            uint8_t position )
```

Debug function for printing incoming messages.

Parameters

id	
data	
position	

6.8.1.2 CAN_init()

```
void CAN_init (
            uint8_t mode )
```

Initialization function for the CAN bus.

Parameters

mode	

6.8.1.3 CAN_receive()

```
void CAN_receive ( )
```

Checks registers for received messages.

6.8.1.4 CAN_send()

```
void CAN_send (
              CAN_message message )
```

Send data though the CAN bus.

Parameters

message	

6.8.1.5 ISR()

```
ISR (
              INT0_vect   )
```

6.9 CAN.h

Go to the documentation of this file.

```
1 #ifndef CAN_H
2 #define CAN_H
3
4 #include "CAN_Controller.h"
5 #include "avr/interrupt.h"
6
7 /*
8     "Buttons" variable:
9     Joystick button « 0
10     Right button « 1
11 */
12
13 typedef struct{
14     uint16_t length;
15     uint16_t id;
16     int8_t data[8];
17 } CAN_message;
18
24 void CAN_init(uint8_t mode);
25
31 void CAN_send(CAN_message message);
32
37 void CAN_receive();
38
46 void CAN_debug_print(uint8_t id, char data, uint8_t position);
47
48
49 // Interrupt 0 trigger function
50 ISR(INT0_vect){
51     printf("message received \r\n");
52     CAN_receive();
53 }
54
55 #endif
```

6.10 D:/Progetti/TTK4155/Node1/TTK4155/CAN_Controller.c File Reference

```
#include "CAN_Controller.h"
```

Functions

- uint8_t MCP2515_read (uint8_t address)

  Reading instruction for the MCP2515.
- void MCP2515_write (uint8_t address, uint8_t data)

  Writing data to a register of the MCP2515.
- void MCP2515_RTS (uint8_t buffer)

  Ready To Send instruction for the MCP2515.
- uint8_t MCP2515_read_status ()

  Read status instruction for the MCP2515.
- void MCP2515_bit_modify (uint8_t address, uint8_t byte_mask, uint8_t data)

  Modify bit/bits of a register providing a mask and the new data.
- void MCP2515_reset ()

  Reset the MCP2515.
- void MCP2515_set_mode (uint8_t mode)

  Set mode for the MCP2515.

6.10.1 Function Documentation

6.10.1.1 MCP2515_bit_modify()

```
void MCP2515_bit_modify (
            uint8_t address,
            uint8_t byte_mask,
            uint8_t data )
```

Modify bit/bits of a register providing a mask and the new data.

Parameters

address	
byte_mask	
data	

6.10.1.2 MCP2515_read()

```
uint8_t MCP2515_read (
            uint8_t address )
```

Reading instruction for the MCP2515.

Parameters

address	Address of the register to read

Returns

uint8_t

6.10.1.3 MCP2515_read_status()

```
uint8_t MCP2515_read_status ( )
```

Read status instruction for the MCP2515.

Returns

uint8_t

6.10.1.4 MCP2515_reset()

```
void MCP2515_reset ( )
```

Reset the MCP2515.

6.10.1.5 MCP2515_RTS()

```
void MCP2515_RTS (
            uint8_t buffer )
```

Ready To Send instruction for the MCP2515.

Parameters

buffer	

6.10.1.6 MCP2515_set_mode()

```
void MCP2515_set_mode (
```

```
            uint8_t mode )
```

Set mode for the MCP2515.

Parameters

mode	

6.10.1.7 MCP2515_write()

```
void MCP2515_write (
            uint8_t address,
            uint8_t data )
```

Writing data to a register of the MCP2515.

Parameters

address	
data	

6.11 D:/Progetti/TTK4155/Node2/Node2/can_controller.c File Reference

```
#include "can_controller.h"
#include "sam.h"
#include "printf-stdarg.h"
```

Functions

- uint8_t can_init_def_tx_rx_mb (uint32_t can_br)

  Initialize can bus with predefined number of rx and tx mailboxes, CAN0->CAN_MB[0] is used for transmitting CAN0->CAN_MB[1,2] is used for receiving.
- uint8_t can_init (uint32_t can_br, uint8_t num_tx_mb, uint8_t num_rx_mb)

  Initialize can bus.
- uint8_t can_send (CAN_MESSAGE *can_msg, uint8_t tx_mb_id)

  Send can message from mailbox.
- uint8_t can_receive (CAN_MESSAGE *can_msg, uint8_t rx_mb_id)

  Read can message from mailbox.

6.11.1 Function Documentation

6.11.1.1 can_init()

```
uint8_t can_init (
            uint32_t can_br,
            uint8_t num_tx_mb,
            uint8_t num_rx_mb )
```

Initialize can bus.

Parameters

can_br	Value to be set in CAN0->CAN_BR register to match can bus bit timing
num_tx_mb	Number of transmit mailboxes, tx mb indexes: [0 , num_tx_mb-1]
num_rx_mb	Number of receive mailboxes, rx mb indexes: [num_tx_mb, num_rx_mb-1]

Return values

Success(0)	or failure(1)

6.11.1.2 can_init_def_tx_rx_mb()

```
uint8_t can_init_def_tx_rx_mb (
            uint32_t can_br )
```

Initialize can bus with predefined number of rx and tx mailboxes, CAN0->CAN_MB[0] is used for transmitting CAN0->CAN_MB[1,2] is used for receiving.

Parameters

can_br	Value to be set in CAN0->CAN_BR register to match can bus bit timing

Return values

Success(0)	or failure(1)

6.11.1.3 can_receive()

```
uint8_t can_receive (
            CAN_MESSAGE * can_msg,
            uint8_t rx_mb_id )
```

Read can message from mailbox.

Parameters

can_msg	struct instance to save received data
rx_mb↩ _id	ID of receive mailbox to be used

Return values

Success(0)	or failure(1)

6.11.1.4 can_send()

```
uint8_t can_send (
            CAN_MESSAGE * can_msg,
            uint8_t tx_mb_id )
```

Send can message from mailbox.

Parameters

can_msg	message to be sent
tx_mb↩ _id	ID of transmit mailbox to be used

Return values

Success(0)	or failure(1)

6.12 D:/Progetti/TTK4155/Node1/TTK4155/CAN_Controller.h File Reference

```
#include "SPI.h"
#include "mcp2515.h"
```

Macros

- #define MCP_TXB0SIDH 0x31
- #define MCP_TXB0SIDL 0x32
- #define MCP_TXB0DLC 0x35
- #define MCP_TXB0D0 0x36
- #define MCP_RXB0SIDL 0X62
- #define MCP_RXB0DLC 0x65
- #define MCP_RXB0D0 0x66
- #define MCP_RXB1SIDL 0X72
- #define MCP_RXB1DLC 0X75
- #define MCP_RXB1D0 0x76

Functions

- uint8_t MCP2515_read (uint8_t address)

  Reading instruction for the MCP2515.
- void MCP2515_write (uint8_t address, uint8_t data)

  Writing data to a register of the MCP2515.
- void MCP2515_RTS (uint8_t buffer)

  Ready To Send instruction for the MCP2515.
- uint8_t MCP2515_read_status ()

  Read status instruction for the MCP2515.
- void MCP2515_bit_modify (uint8_t address, uint8_t byte_mask, uint8_t data)

  Modify bit/bits of a register providing a mask and the new data.
- void MCP2515_reset ()

  Reset the MCP2515.
- void MCP2515_set_mode (uint8_t mode)

  Set mode for the MCP2515.

6.12.1 Macro Definition Documentation

6.12.1.1 MCP_RXB0D0

```
#define MCP_RXB0D0 0x66
```

6.12.1.2 MCP_RXB0DLC

```
#define MCP_RXB0DLC 0x65
```

6.12.1.3 MCP_RXB0SIDL

```
#define MCP_RXB0SIDL 0X62
```

6.12.1.4 MCP_RXB1D0

```
#define MCP_RXB1D0 0x76
```

6.12.1.5 MCP_RXB1DLC

```
#define MCP_RXB1DLC 0X75
```

6.12.1.6 MCP_RXB1SIDL

```
#define MCP_RXB1SIDL 0X72
```

6.12.1.7 MCP_TXB0D0

```
#define MCP_TXB0D0 0x36
```

6.12.1.8 MCP_TXB0DLC

```
#define MCP_TXB0DLC 0x35
```

6.12.1.9 MCP_TXB0SIDH

```
#define MCP_TXB0SIDH 0x31
```

6.12.1.10 MCP_TXB0SIDL

```
#define MCP_TXB0SIDL 0x32
```

6.12.2 Function Documentation

6.12.2.1 MCP2515_bit_modify()

```
void MCP2515_bit_modify (
            uint8_t address,
            uint8_t byte_mask,
            uint8_t data )
```

Modify bit/bits of a register providing a mask and the new data.

Parameters

address	
byte_mask	
data	

6.12.2.2 MCP2515_read()

```
uint8_t MCP2515_read (
            uint8_t address )
```

Reading instruction for the MCP2515.

Parameters

address	Address of the register to read

Returns

uint8_t

6.12.2.3 MCP2515_read_status()

```
uint8_t MCP2515_read_status ( )
```

Read status instruction for the MCP2515.

Returns

uint8_t

6.12.2.4 MCP2515_reset()

```
void MCP2515_reset ( )
```

Reset the MCP2515.

6.12.2.5 MCP2515_RTS()

```
void MCP2515_RTS (
            uint8_t buffer )
```

Ready To Send instruction for the MCP2515.

Parameters

buffer	

6.12.2.6 MCP2515_set_mode()

```
void MCP2515_set_mode (
            uint8_t mode )
```

Set mode for the MCP2515.

Parameters

mode	

6.12.2.7 MCP2515_write()

```
void MCP2515_write (
            uint8_t address,
            uint8_t data )
```

Writing data to a register of the MCP2515.

Parameters

address	
data	

6.13 CAN_Controller.h

Go to the documentation of this file.

```
1 #ifndef CAN_CONTROLLER_H
2 #define CAN_CONTROLLER_H
3
4 #include "SPI.h"
5 #include "mcp2515.h"
6
7 // From the Register Map table 11.1 page 63 of MCP2515 datasheet
8 #define MCP_TXB0SIDH    0x31
9 #define MCP_TXB0SIDL    0x32
10 #define MCP_TXB0DLC     0x35
11 #define MCP_TXB0D0      0x36
12 #define MCP_RXB0SIDL    0X62
13 #define MCP_RXB0DLC     0x65
14 #define MCP_RXB0D0      0x66
15 #define MCP_RXB1SIDL    0X72
16 #define MCP_RXB1DLC     0X75
17 #define MCP_RXB1D0      0x76
18
25 uint8_t MCP2515_read(uint8_t address);
```

```
26
33 void MCP2515_write(uint8_t address, uint8_t data);
34
40 void MCP2515_RTS(uint8_t buffer);
41
47 uint8_t MCP2515_read_status();
48
56 void MCP2515_bit_modify(uint8_t address, uint8_t byte_mask, uint8_t data);
57
62 void MCP2515_reset();
63
69 void MCP2515_set_mode(uint8_t mode);
70
71 #endif
```

6.14 D:/Progetti/TTK4155/Node2/Node2/can_controller.h File Reference

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

Data Structures

- struct can_message_t

  Structure of the CAN message.

Typedefs

- typedef struct can_message_t CAN_MESSAGE

  Structure of the CAN message.

Functions

- uint8_t can_init_def_tx_rx_mb (uint32_t can_br)

  Initialize can bus with predefined number of rx and tx mailboxes, CAN0->CAN_MB[0] is used for transmitting CAN0->CAN_MB[1,2] is used for receiving.
- uint8_t can_init (uint32_t can_br, uint8_t num_tx_mb, uint8_t num_rx_mb)

  Initialize can bus.
- uint8_t can_send (CAN_MESSAGE *can_msg, uint8_t mailbox_id)

  Send can message from mailbox.
- uint8_t can_receive (CAN_MESSAGE *can_msg, uint8_t mailbox_id)

  Read can message from mailbox.

6.14.1 Typedef Documentation

6.14.1.1 CAN_MESSAGE

```
typedef struct can_message_t CAN_MESSAGE
```

Structure of the CAN message.

6.14.2 Function Documentation

6.14.2.1 can_init()

```
uint8_t can_init (
            uint32_t can_br,
            uint8_t num_tx_mb,
            uint8_t num_rx_mb )
```

Initialize can bus.

Parameters

can_br	Value to be set in CAN0->CAN_BR register to match can bus bit timing
num_tx_mb	Number of transmit mailboxes, tx mb indexes: [0 , num_tx_mb-1]
num_rx_mb	Number of receive mailboxes, rx mb indexes: [num_tx_mb, num_rx_mb-1]

Return values

Success(0)	or failure(1)

6.14.2.2 can_init_def_tx_rx_mb()

```
uint8_t can_init_def_tx_rx_mb (
            uint32_t can_br )
```

Initialize can bus with predefined number of rx and tx mailboxes, CAN0->CAN_MB[0] is used for transmitting CAN0->CAN_MB[1,2] is used for receiving.

Parameters

can_br	Value to be set in CAN0->CAN_BR register to match can bus bit timing

Return values

Success(0)	or failure(1)

6.14.2.3 can_receive()

```
uint8_t can_receive (
```

```
            CAN_MESSAGE * can_msg,
            uint8_t rx_mb_id )
```

Read can message from mailbox.

Parameters

can_msg	struct instance to save received data
rx_mb↩ _id	ID of receive mailbox to be used

Return values

Success(0)	or failure(1)

6.14.2.4 can_send()

```
uint8_t can_send (
            CAN_MESSAGE * can_msg,
            uint8_t tx_mb_id )
```

Send can message from mailbox.

Parameters

can_msg	message to be sent
tx_mb↩ _id	ID of transmit mailbox to be used

Return values

Success(0)	or failure(1)

6.15 can_controller.h

Go to the documentation of this file.

```
1 /*
2  * can_controller.h
3  *
4  * Author: Gustav O. Often and Eivind H. J⌦lsgard
5  *
6  * For use in TTK4155 Embedded and Industrial Computer Systems Design
7  * NTNU - Norwegian University of Science and Technology
8  *
9  */
10
11
12 #ifndef CAN_CONTROLLER_H_
13 #define CAN_CONTROLLER_H_
14
15 #include <stdint.h>
```

```
16
21 typedef struct can_message_t
22 {
23     uint16_t  id;
24     uint16_t data_length;
25     int8_t data[8];
26 } CAN_MESSAGE;
27
28 uint8_t can_init_def_tx_rx_mb(uint32_t can_br);
29 uint8_t can_init(uint32_t can_br, uint8_t num_tx_mb, uint8_t num_rx_mb);
30
31 uint8_t can_send(CAN_MESSAGE* can_msg, uint8_t mailbox_id);
32 uint8_t can_receive(CAN_MESSAGE* can_msg, uint8_t mailbox_id);
33
34 #endif /* CAN_CONTROLLER_H_ */
```

6.16 D:/Progetti/TTK4155/Node1/TTK4155/Debug/main.d File Reference

6.17 D:/Progetti/TTK4155/Node2/Node2/Debug/main.d File Reference

6.18 D:/Progetti/TTK4155/Node1/TTK4155/fonts.h File Reference

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

Variables

- const unsigned char PROGMEM font8 [95][8]
- const unsigned char PROGMEM font5 [95][5]
- const unsigned char PROGMEM font4 [95][4]

6.18.1 Variable Documentation

6.18.1.1 font4

```
const unsigned char PROGMEM font4[95][4]
```

6.18.1.2 font5

```
const unsigned char PROGMEM font5[95][5]
```

6.18.1.3 font8

```
const unsigned char PROGMEM font8[95][8]
```

6.19 fonts.h

Go to the documentation of this file.

```
1 /*
2  * fonts.h
3  *
4  * Library of fonts
5  * Large: 8x8, normal: 5x7 and small: 4x7
6  */
7 #ifndef FONTS_H_
8 #define FONTS_H_
9
10
11 #include <avr/pgmspace.h>
12
13 // Font 8x8 - Large
14 const unsigned char PROGMEM font8[95][8] = {
15     {0b00000000,0b00000000,0b00000000,0b00000000,0b00000000,0b00000000,0b00000000,0b00000000}, //
16     {0b00000000,0b00000110,0b01011111,0b01011111,0b00000110,0b00000000,0b00000000,0b00000000}, // !
17     {0b00000000,0b00000111,0b00000111,0b00000000,0b00000111,0b00000111,0b00000000,0b00000000}, // "
18     {0b00010100,0b01111111,0b01111111,0b00010100,0b01111111,0b01111111,0b00010100,0b00000000}, // #
19     {0b00100100,0b00101110,0b01101011,0b01101011,0b00111010,0b00010010,0b00000000,0b00000000}, // $
20     {0b01000110,0b01100110,0b00110000,0b00011000,0b00001100,0b01100110,0b01100010,0b00000000}, // %
21     {0b00110000,0b01111010,0b01001111,0b01011101,0b00110111,0b01111010,0b01001000,0b00000000}, // &
22     {0b00000100,0b00000111,0b00000011,0b00000000,0b00000000,0b00000000,0b00000000,0b00000000}, // '
23     {0b00000000,0b00011100,0b00111110,0b01100011,0b01000001,0b00000000,0b00000000,0b00000000}, // (
24     {0b00000000,0b01000001,0b01100011,0b00111110,0b00011100,0b00000000,0b00000000,0b00000000}, // )
25     {0b00001000,0b00101010,0b00111110,0b00011100,0b00011100,0b00111110,0b00101010,0b00001000}, // *
26     {0b00001000,0b00001000,0b00111110,0b00111110,0b00001000,0b00001000,0b00000000,0b00000000}, // +
27     {0b00000000,0b10100000,0b11100000,0b01100000,0b00000000,0b00000000,0b00000000,0b00000000}, // ,
28     {0b00001000,0b00001000,0b00001000,0b00001000,0b00001000,0b00001000,0b00000000,0b00000000}, // -
29     {0b00000000,0b00000000,0b01100000,0b01100000,0b00000000,0b00000000,0b00000000,0b00000000}, // .
30     {0b01100000,0b00110000,0b00011000,0b00001100,0b00000110,0b00000011,0b00000001,0b00000000}, // /
31     {0b00111110,0b01111111,0b01011001,0b01001101,0b01111111,0b00111110,0b00000000,0b00000000}, // 0
32     {0b01000010,0b01000010,0b01111111,0b01111111,0b01000000,0b01000000,0b00000000,0b00000000}, // 1
33     {0b01100010,0b01110011,0b01011001,0b01001001,0b01101111,0b01100110,0b00000000,0b00000000}, // 2
34     {0b00100010,0b01100011,0b01001001,0b01001001,0b01111111,0b00110110,0b00000000,0b00000000}, // 3
35     {0b00011000,0b00011100,0b00010110,0b00010011,0b01111111,0b01111111,0b00010000,0b00000000}, // 4
36     {0b00100111,0b01100111,0b01000101,0b01000101,0b01111101,0b00111001,0b00000000,0b00000000}, // 5
37     {0b00111100,0b01111110,0b01001011,0b01001001,0b01111001,0b00110000,0b00000000,0b00000000}, // 6
38     {0b00000011,0b01100011,0b01110001,0b00011001,0b00001111,0b00000111,0b00000000,0b00000000}, // 7
39     {0b00110110,0b01111111,0b01001001,0b01001001,0b01111111,0b00110110,0b00000000,0b00000000}, // 8
40     {0b00000110,0b01001111,0b01001001,0b01101001,0b00111111,0b00011110,0b00000000,0b00000000}, // 9
41     {0b00000000,0b00000000,0b01101100,0b01101100,0b00000000,0b00000000,0b00000000,0b00000000}, // :
42     {0b00000000,0b10100000,0b11101100,0b01101100,0b00000000,0b00000000,0b00000000,0b00000000}, // ;
43     {0b00001000,0b00011100,0b00110110,0b01100011,0b01000001,0b00000000,0b00000000,0b00000000}, // <
44     {0b00010100,0b00010100,0b00010100,0b00010100,0b00010100,0b00010100,0b00000000,0b00000000}, // =
45     {0b00000000,0b01000001,0b01100011,0b00110110,0b00011100,0b00001000,0b00000000,0b00000000}, // >
46     {0b00000010,0b00000011,0b01010001,0b01011001,0b00001111,0b00000110,0b00000000,0b00000000}, // ?
47     {0b00111110,0b01111111,0b01000001,0b01011101,0b01011101,0b00011111,0b00011110,0b00000000}, // @
48     {0b01111100,0b01111110,0b00010011,0b00010011,0b01111110,0b01111100,0b00000000,0b00000000}, // A
49     {0b01000001,0b01111111,0b01111111,0b01001001,0b01001001,0b01111111,0b00110110,0b00000000}, // B
50     {0b00011100,0b00111110,0b01100011,0b01000001,0b01000001,0b01100011,0b00100010,0b00000000}, // C
51     {0b01000001,0b01111111,0b01111111,0b01000001,0b01100011,0b01111111,0b00011100,0b00000000}, // D
52     {0b01000001,0b01111111,0b01111111,0b01001001,0b01011101,0b01000001,0b01100011,0b00000000}, // E
53     {0b01000001,0b01111111,0b01111111,0b01001001,0b00011101,0b00000001,0b00000011,0b00000000}, // F
54     {0b00011100,0b00111110,0b01100011,0b01000001,0b01010001,0b01110011,0b01110010,0b00000000}, // G
55     {0b01111111,0b01111111,0b00001000,0b00001000,0b01111111,0b01111111,0b00000000,0b00000000}, // H
56     {0b00000000,0b01000001,0b01111111,0b01111111,0b01000001,0b00000000,0b00000000,0b00000000}, // I
57     {0b00110000,0b01110000,0b01000000,0b01000001,0b01111111,0b00111111,0b00000001,0b00000000}, // J
58     {0b01000001,0b01111111,0b01111111,0b00001000,0b00011100,0b01110111,0b01100011,0b00000000}, // K
59     {0b01000001,0b01111111,0b01111111,0b01000001,0b01000000,0b01100000,0b01110000,0b00000000}, // L
60     {0b01111111,0b01111111,0b00000110,0b00001100,0b00000110,0b01111111,0b01111111,0b00000000}, // M
61     {0b01111111,0b01111111,0b00000110,0b00001100,0b00011000,0b01111111,0b01111111,0b00000000}, // N
62     {0b00011100,0b00111110,0b01100011,0b01000001,0b01100011,0b00111110,0b00011100,0b00000000}, // O
63     {0b01000001,0b01111111,0b01111111,0b01001001,0b00001001,0b00001111,0b00000110,0b00000000}, // P
64     {0b00011110,0b00111111,0b00100001,0b01110001,0b01111111,0b01011110,0b00000000,0b00000000}, // Q
65     {0b01000001,0b01111111,0b01111111,0b00011001,0b00111001,0b01101111,0b01000110,0b00000000}, // R
66     {0b00100110,0b01100111,0b01001101,0b01011001,0b01111011,0b00110010,0b00000000,0b00000000}, // S
67     {0b00000011,0b01000001,0b01111111,0b01111111,0b01000001,0b00000011,0b00000000,0b00000000}, // T
68     {0b01111111,0b01111111,0b01000000,0b01000000,0b01111111,0b01111111,0b00000000,0b00000000}, // U
69     {0b00011111,0b00111111,0b01100000,0b01100000,0b00111111,0b00011111,0b00000000,0b00000000}, // V
70     {0b01111111,0b01111111,0b00110000,0b00011000,0b00110000,0b01111111,0b01111111,0b00000000}, // W
71     {0b01100011,0b01110111,0b00011100,0b00001000,0b00011100,0b01110111,0b01100011,0b00000000}, // X
72     {0b00000111,0b01001111,0b01111000,0b01111000,0b01001111,0b00000111,0b00000000,0b00000000}, // Y
73     {0b01100111,0b01110011,0b01011001,0b01001101,0b01000111,0b01100011,0b01110001,0b00000000}, // Z
74     {0b00000000,0b01111111,0b01111111,0b01000001,0b01000001,0b00000000,0b00000000,0b00000000}, // [
75     {0b00000001,0b00000011,0b00000110,0b00001100,0b00011000,0b00110000,0b01100000,0b00000000}, // "\"
76     {0b00000000,0b01000001,0b01000001,0b01111111,0b01111111,0b00000000,0b00000000,0b00000000}, // ]
77     {0b00001000,0b00001100,0b00000110,0b00000011,0b00000110,0b00001100,0b00001000,0b00000000}, // ^
78     {0b10000000,0b10000000,0b10000000,0b10000000,0b10000000,0b10000000,0b10000000,0b10000000}, // _
79     {0b00000000,0b00000000,0b00000011,0b00000111,0b00000100,0b00000000,0b00000000,0b00000000}, // `
80     {0b00100000,0b01110100,0b01010100,0b01010100,0b00111100,0b01111000,0b01000000,0b00000000}, // a
81     {0b01000001,0b00111111,0b01111111,0b01000100,0b01000100,0b01111100,0b00111000,0b00000000}, // b
82     {0b00111000,0b01111100,0b01000100,0b01000100,0b01101100,0b00101000,0b00000000,0b00000000}, // c
```

```
83      {0b00110000,0b01111000,0b01001000,0b01001001,0b00111111,0b01111111,0b01000000,0b00000000}, // d
84      {0b00111000,0b01111100,0b01010100,0b01010100,0b01011100,0b00011000,0b00000000,0b00000000}, // e
85      {0b01001000,0b01111110,0b01111111,0b01001001,0b00000011,0b00000010,0b00000000,0b00000000}, // f
86      {0b10011000,0b10111100,0b10100100,0b10100100,0b11111000,0b01111100,0b00000100,0b00000000}, // g
87      {0b01000001,0b01111111,0b01111111,0b00001000,0b00000100,0b01111100,0b01111000,0b00000000}, // h
88      {0b00000000,0b01000100,0b01111101,0b01111101,0b01000000,0b00000000,0b00000000,0b00000000}, // i
89      {0b01000000,0b11000100,0b10000100,0b11111101,0b01111101,0b00000000,0b00000000,0b00000000}, // j
90      {0b01000001,0b01111111,0b01111111,0b00010000,0b00111000,0b01101100,0b01000100,0b00000000}, // k
91      {0b00000000,0b01000001,0b01111111,0b01111111,0b01000000,0b00000000,0b00000000,0b00000000}, // l
92      {0b01111100,0b01111100,0b00001100,0b00011000,0b00001100,0b01111100,0b01111000,0b00000000}, // m
93      {0b01111100,0b01111100,0b00000100,0b00000100,0b01111100,0b01111000,0b00000000,0b00000000}, // n
94      {0b00111000,0b01111100,0b01000100,0b01000100,0b01111100,0b00111000,0b00000000,0b00000000}, // o
95      {0b10000100,0b11111100,0b11111000,0b10100100,0b00100100,0b00111100,0b00011000,0b00000000}, // p
96      {0b00011000,0b00111100,0b00100100,0b10100100,0b11111000,0b11111100,0b10000100,0b00000000}, // q
97      {0b01000100,0b01111100,0b01111000,0b01000100,0b00011100,0b00011000,0b00000000,0b00000000}, // r
98      {0b01001000,0b01011100,0b01010100,0b01010100,0b01110100,0b00100100,0b00000000,0b00000000}, // s
99      {0b00000000,0b00000100,0b00111110,0b01111111,0b01000100,0b00100100,0b00000000,0b00000000}, // t
100      {0b00111100,0b01111100,0b01000000,0b01000000,0b00111100,0b01111100,0b01000000,0b00000000}, // u
101      {0b00011100,0b00111100,0b01100000,0b01100000,0b00111100,0b00011100,0b00000000,0b00000000}, // v
102      {0b00111100,0b01111100,0b01100000,0b00110000,0b01100000,0b01111100,0b00111100,0b00000000}, // w
103      {0b01000100,0b01101100,0b00111000,0b00010000,0b00111000,0b01101100,0b01000100,0b00000000}, // x
104      {0b10011100,0b10111100,0b10100000,0b10100000,0b11111100,0b01111100,0b00000000,0b00000000}, // y
105      {0b01001100,0b01100100,0b01110100,0b01011100,0b01001100,0b01100100,0b00000000,0b00000000}, // z
106      {0b00001000,0b00001000,0b00111110,0b01110111,0b01000001,0b01000001,0b00000000,0b00000000}, // {
107      {0b00000000,0b00000000,0b00000000,0b01110111,0b01110111,0b00000000,0b00000000,0b00000000}, // |
108      {0b01000001,0b01000001,0b01110111,0b00111110,0b00001000,0b00001000,0b00000000,0b00000000}, // }
109      {0b00000010,0b00000011,0b00000001,0b00000011,0b00000010,0b00000011,0b00000001,0b00000000}, // ~
110 };
111
112 // Font 5x7 - normal
113 const unsigned char PROGMEM font5[95][5] = {
114      {0b00000000,0b00000000,0b00000000,0b00000000,0b00000000}, //
115      {0b00000000,0b00000000,0b01011111,0b00000000,0b00000000}, // !
116      {0b00000000,0b00000111,0b00000000,0b00000111,0b00000000}, // "
117      {0b00010100,0b01111111,0b00010100,0b01111111,0b00010100}, // #
118      {0b00100100,0b00101010,0b01111111,0b00101010,0b00010010}, // $
119      {0b00100011,0b00010011,0b00001000,0b01100100,0b01100010}, // %
120      {0b00110110,0b01001001,0b01010101,0b00100010,0b01010000}, // &
121      {0b00000000,0b00000101,0b00000011,0b00000000,0b00000000}, // '
122      {0b00000000,0b00011100,0b00100010,0b01000001,0b00000000}, // (
123      {0b00000000,0b01000001,0b00100010,0b00011100,0b00000000}, // )
124      {0b00001000,0b00101010,0b00011100,0b00101010,0b00001000}, // *
125      {0b00001000,0b00001000,0b00111110,0b00001000,0b00001000}, // +
126      {0b00000000,0b01010000,0b00110000,0b00000000,0b00000000}, // ,
127      {0b00001000,0b00001000,0b00001000,0b00001000,0b00001000}, // -
128      {0b00000000,0b01100000,0b01100000,0b00000000,0b00000000}, // .
129      {0b00100000,0b00010000,0b00001000,0b00000100,0b00000010}, // /
130      {0b00111110,0b01010001,0b01001001,0b01000101,0b00111110}, // 0
131      {0b00000000,0b01000010,0b01111111,0b01000000,0b00000000}, // 1
132      {0b01000010,0b01100001,0b01010001,0b01001001,0b01000110}, // 2
133      {0b00100001,0b01000001,0b01000101,0b01001011,0b00110001}, // 3
134      {0b00011000,0b00010100,0b00010010,0b01111111,0b00010000}, // 4
135      {0b00100111,0b01000101,0b01000101,0b01000101,0b00111001}, // 5
136      {0b00111100,0b01001010,0b01001001,0b01001001,0b00110000}, // 6
137      {0b00000001,0b01110001,0b00001001,0b00000101,0b00000011}, // 7
138      {0b00110110,0b01001001,0b01001001,0b01001001,0b00110110}, // 8
139      {0b00000110,0b01001001,0b01001001,0b00101001,0b00011110}, // 9
140      {0b00000000,0b00110110,0b00110110,0b00000000,0b00000000}, // :
141      {0b00000000,0b01010110,0b00110110,0b00000000,0b00000000}, // ;
142      {0b00000000,0b00001000,0b00010100,0b00100010,0b01000001}, // <
143      {0b00010100,0b00010100,0b00010100,0b00010100,0b00010100}, // =
144      {0b01000001,0b00100010,0b00010100,0b00001000,0b00000000}, // >
145      {0b00000010,0b00000001,0b01010001,0b00001001,0b00000110}, // ?
146      {0b00110010,0b01001001,0b01111001,0b01000001,0b00111110}, // @
147      {0b01111110,0b00010001,0b00010001,0b00010001,0b01111110}, // A
148      {0b01111111,0b01001001,0b01001001,0b01001001,0b00110110}, // B
149      {0b00111110,0b01000001,0b01000001,0b01000001,0b00100010}, // C
150      {0b01111111,0b01000001,0b01000001,0b00100010,0b00011100}, // D
151      {0b01111111,0b01001001,0b01001001,0b01001001,0b01000001}, // E
152      {0b01111111,0b00001001,0b00001001,0b00000001,0b00000001}, // F
153      {0b00111110,0b01000001,0b01000001,0b01010001,0b00110010}, // G
154      {0b01111111,0b00001000,0b00001000,0b00001000,0b01111111}, // H
155      {0b00000000,0b01000001,0b01111111,0b01000001,0b00000000}, // I
156      {0b00100000,0b01000000,0b01000001,0b00111111,0b00000001}, // J
157      {0b01111111,0b00001000,0b00010100,0b00100010,0b01000001}, // K
158      {0b01111111,0b01000000,0b01000000,0b01000000,0b01000000}, // L
159      {0b01111111,0b00000010,0b00000100,0b00000010,0b01111111}, // M
160      {0b01111111,0b00000100,0b00001000,0b00010000,0b01111111}, // N
161      {0b00111110,0b01000001,0b01000001,0b01000001,0b00111110}, // O
162      {0b01111111,0b00001001,0b00001001,0b00001001,0b00000110}, // P
163      {0b00111110,0b01000001,0b01010001,0b00100001,0b01011110}, // Q
164      {0b01111111,0b00001001,0b00011001,0b00101001,0b01000110}, // R
165      {0b01000110,0b01001001,0b01001001,0b01001001,0b00110001}, // S
166      {0b00000001,0b00000001,0b01111111,0b00000001,0b00000001}, // T
167      {0b00111111,0b01000000,0b01000000,0b01000000,0b00111111}, // U
168      {0b00011111,0b00100000,0b01000000,0b00100000,0b00011111}, // V
169      {0b01111111,0b00100000,0b00011000,0b00100000,0b01111111}, // W
```

```
170     {0b01100011,0b00010100,0b00001000,0b00010100,0b01100011}, // X
171     {0b00000011,0b00000100,0b01111000,0b00000100,0b00000011}, // Y
172     {0b01100001,0b01010001,0b01001001,0b01000101,0b01000011}, // Z
173     {0b00000000,0b00000000,0b01111111,0b01000001,0b01000001}, // [
174     {0b00000010,0b00000100,0b00001000,0b00010000,0b00100000}, // "\"
175     {0b01000001,0b01000001,0b01111111,0b00000000,0b00000000}, // ]
176     {0b00000100,0b00000010,0b00000001,0b00000010,0b00000100}, // ^
177     {0b01000000,0b01000000,0b01000000,0b01000000,0b01000000}, // _
178     {0b00000000,0b00000001,0b00000010,0b00000100,0b00000000}, // `
179     {0b00100000,0b01010100,0b01010100,0b01010100,0b01111000}, // a
180     {0b01111111,0b01001000,0b01000100,0b01000100,0b00111000}, // b
181     {0b00111000,0b01000100,0b01000100,0b01000100,0b00100000}, // c
182     {0b00111000,0b01000100,0b01000100,0b01001000,0b01111111}, // d
183     {0b00111000,0b01010100,0b01010100,0b01010100,0b00011000}, // e
184     {0b00001000,0b01111110,0b00001001,0b00000001,0b00000010}, // f
185     {0b00001000,0b00010100,0b01010100,0b01010100,0b00111100}, // g
186     {0b01111111,0b00001000,0b00000100,0b00000100,0b01111000}, // h
187     {0b00000000,0b01000100,0b01111101,0b01000000,0b00000000}, // i
188     {0b00100000,0b01000000,0b01000100,0b00111101,0b00000000}, // j
189     {0b00000000,0b01111111,0b00010000,0b00101000,0b01000100}, // k
190     {0b00000000,0b01000001,0b01111111,0b01000000,0b00000000}, // l
191     {0b01111100,0b00000100,0b00011000,0b00000100,0b01111000}, // m
192     {0b01111100,0b00001000,0b00000100,0b00000100,0b01111000}, // n
193     {0b00111000,0b01000100,0b01000100,0b01000100,0b00111000}, // o
194     {0b01111100,0b00010100,0b00010100,0b00010100,0b00001000}, // p
195     {0b00001000,0b00010100,0b00010100,0b00011000,0b01111100}, // q
196     {0b01111100,0b00001000,0b00000100,0b00000100,0b00001000}, // r
197     {0b01001000,0b01010100,0b01010100,0b01010100,0b00100000}, // s
198     {0b00000100,0b00111111,0b01000100,0b01000000,0b00100000}, // t
199     {0b00111100,0b01000000,0b01000000,0b00100000,0b01111100}, // u
200     {0b00011100,0b00100000,0b01000000,0b00100000,0b00011100}, // v
201     {0b00111100,0b01000000,0b00110000,0b01000000,0b00111100}, // w
202     {0b01000100,0b00101000,0b00010000,0b00101000,0b01000100}, // x
203     {0b00001100,0b01010000,0b01010000,0b01010000,0b00111100}, // y
204     {0b01000100,0b01100100,0b01010100,0b01001100,0b01000100}, // z
205     {0b00000000,0b00001000,0b00110110,0b01000001,0b00000000}, // {
206     {0b00000000,0b00000000,0b01111111,0b00000000,0b00000000}, // |
207     {0b00000000,0b01000001,0b00110110,0b00001000,0b00000000}, // }
208     {0b00000010,0b00000001,0b00000011,0b00000010,0b00000001}, // ~
209     };
210
211 // Font 4x6 - Small
212 const unsigned char PROGMEM font4[95][4] = {
213         {0b00000000,0b00000000,0b00000000,0b00000000}, //
214         {0b00000000,0b01011100,0b00000000,0b00000000}, // !
215         {0b00001100,0b00000000,0b00001100,0b00000000}, // "
216         {0b01111100,0b00101000,0b01111100,0b00101000}, // #
217         {0b01011000,0b11011100,0b01101000,0b00000000}, // $
218         {0b00100100,0b00010000,0b01001000,0b00000000}, // %
219         {0b00101000,0b01010100,0b00101000,0b01000000}, // &
220         {0b00000000,0b00001100,0b00000000,0b00000000}, // ’
221         {0b00000000,0b01111000,0b10000100,0b00000000}, // (
222         {0b10000100,0b01111000,0b00000000,0b00000000}, // )
223         {0b01010100,0b00111000,0b01010100,0b00000000}, // *
224         {0b00010000,0b01111100,0b00010000,0b00000000}, // +
225         {0b10000000,0b01000000,0b00000000,0b00000000}, // ,
226         {0b00010000,0b00010000,0b00010000,0b00000000}, // -
227         {0b00000000,0b01000000,0b00000000,0b00000000}, // .
228         {0b01100000,0b00010000,0b00001100,0b00000000}, // /
229         {0b00111000,0b01010100,0b00111000,0b00000000}, // 0
230         {0b01001000,0b01111100,0b01000000,0b00000000}, // 1
231         {0b01001000,0b01100100,0b01011000,0b00000000}, // 2
232         {0b01000100,0b01010100,0b00101100,0b00000000}, // 3
233         {0b00011100,0b00010000,0b01111100,0b00000000}, // 4
234         {0b01011100,0b01010100,0b00100100,0b00000000}, // 5
235         {0b00111000,0b01010100,0b00100100,0b00000000}, // 6
236         {0b01100100,0b00010100,0b00001100,0b00000000}, // 7
237         {0b01101000,0b01010100,0b00101100,0b00000000}, // 8
238         {0b01001000,0b01010100,0b00111000,0b00000000}, // 9
239         {0b00000000,0b01001000,0b00000000,0b00000000}, // :
240         {0b10000000,0b01001000,0b00000000,0b00000000}, // ;
241         {0b00010000,0b00101000,0b01000100,0b00000000}, // <
242         {0b00101000,0b00101000,0b00101000,0b00000000}, // =
243         {0b01000100,0b00101000,0b00010000,0b00000000}, // >
244         {0b00000100,0b01010100,0b00001000,0b00000000}, // ?
245         {0b00111000,0b01000100,0b01011100,0b00000000}, // @
246         {0b01111000,0b00010100,0b01111000,0b00000000}, // A
247         {0b01111100,0b01010100,0b00101000,0b00000000}, // B
248         {0b00111000,0b01000100,0b00101000,0b00000000}, // C
249         {0b01111100,0b01000100,0b00111000,0b00000000}, // D
250         {0b01111100,0b01010100,0b01000100,0b00000000}, // E
251         {0b01111100,0b00010100,0b00000100,0b00000000}, // F
252         {0b00111000,0b01000100,0b01110100,0b00000000}, // G
253         {0b01111100,0b00010000,0b01111100,0b00000000}, // H
254         {0b01000100,0b01111100,0b01000100,0b00000000}, // I
255         {0b00100000,0b01000000,0b00111100,0b00000000}, // J
256         {0b01111100,0b00010000,0b01101100,0b00000000}, // K
```

```
257         {0b01111100,0b01000000,0b01000000,0b00000000}, // L
258         {0b01111100,0b00011000,0b01111100,0b00000000}, // M
259         {0b01111000,0b00010000,0b00111100,0b00000000}, // N
260         {0b00111000,0b01000100,0b00111000,0b00000000}, // O
261         {0b01111100,0b00010100,0b00001000,0b00000000}, // P
262         {0b00111000,0b01000100,0b10111000,0b00000000}, // Q
263         {0b01111100,0b00010100,0b01101000,0b00000000}, // R
264         {0b01001000,0b01010100,0b00100100,0b00000000}, // S
265         {0b00000100,0b01111100,0b00000100,0b00000000}, // T
266         {0b01111100,0b01000000,0b01111100,0b00000000}, // U
267         {0b00111100,0b01100000,0b00111100,0b00000000}, // V
268         {0b01111100,0b00110000,0b01111100,0b00000000}, // W
269         {0b01101100,0b00010000,0b01101100,0b00000000}, // X
270         {0b00001100,0b01110000,0b00001100,0b00000000}, // Y
271         {0b01100100,0b01010100,0b01001100,0b00000000}, // Z
272         {0b00000000,0b01111100,0b01000100,0b00000000}, // [
273         {0b00001100,0b00010000,0b01100000,0b00000000}, // "\"
274         {0b01000100,0b01111100,0b00000000,0b00000000}, // ]
275         {0b00001000,0b00000100,0b00001000,0b00000000}, // ^
276         {0b10000000,0b10000000,0b10000000,0b00000000}, // _
277         {0b00000000,0b00000100,0b00001000,0b00000000}, // `
278         {0b00110000,0b01001000,0b01111000,0b00000000}, // a
279         {0b01111100,0b01001000,0b00110000,0b00000000}, // b
280         {0b00110000,0b01001000,0b01001000,0b00000000}, // c
281         {0b00110000,0b01001000,0b01111100,0b00000000}, // d
282         {0b00110000,0b01101000,0b01010000,0b00000000}, // e
283         {0b00010000,0b01111000,0b00010100,0b00000000}, // f
284         {0b10010000,0b10101000,0b01111000,0b00000000}, // g
285         {0b01111100,0b00001000,0b01110000,0b00000000}, // h
286         {0b01010000,0b01110100,0b01000000,0b00000000}, // i
287         {0b10000000,0b10000000,0b01110100,0b00000000}, // j
288         {0b01111100,0b00010000,0b01101000,0b00000000}, // k
289         {0b01000100,0b01111100,0b01000000,0b00000000}, // l
290         {0b01111000,0b00010000,0b01111000,0b00000000}, // m
291         {0b01111000,0b00001000,0b01110000,0b00000000}, // n
292         {0b00110000,0b01001000,0b00110000,0b00000000}, // o
293         {0b11111000,0b00101000,0b00010000,0b00000000}, // p
294         {0b00110000,0b01001000,0b11111000,0b00000000}, // q
295         {0b01111000,0b00010000,0b00001000,0b00000000}, // r
296         {0b01010000,0b01011000,0b00101000,0b00000000}, // s
297         {0b00001000,0b00111100,0b01001000,0b00000000}, // t
298         {0b00111000,0b01000000,0b01111000,0b00000000}, // u
299         {0b00111000,0b01000000,0b00111000,0b00000000}, // v
300         {0b01111000,0b00100000,0b01111000,0b00000000}, // w
301         {0b01001000,0b00110000,0b01001000,0b00000000}, // x
302         {0b10011000,0b10100000,0b01111000,0b00000000}, // y
303         {0b01001000,0b01101000,0b01011000,0b00000000}, // z
304         {0b00010000,0b01111000,0b10000100,0b00000000}, // {
305         {0b00000000,0b01111100,0b00000000,0b00000000}, // |
306         {0b10000100,0b01111000,0b00010000,0b00000000}, // }
307         {0b00001000,0b00000100,0b00001000,0b00000100}, // ~
308         };
309
310
311 #endif /* FONTS_H_ */
```

6.20 D:/Progetti/TTK4155/Node1/TTK4155/Interrupt.c File Reference

```
#include "Interrupt.h"
```

Functions

- void interrupt_init ()

  initialization function for the interrupt
- void interrupt_polling ()

  Resets variables and prints in the console if a button is pressed.

Variables

- RIGHT_BUTTON_PRESSED = 0
- LEFT_BUTTON_PRESSED = 0
- JOYSTICK_BUTTON_PRESSED = 0

6.20.1 Function Documentation

6.20.1.1 interrupt_init()

```
void interrupt_init ( )
```

initialization function for the interrupt

6.20.1.2 interrupt_polling()

```
void interrupt_polling ( )
```

Resets variables and prints in the console if a button is pressed.

6.20.2 Variable Documentation

6.20.2.1 JOYSTICK_BUTTON_PRESSED

```
JOYSTICK_BUTTON_PRESSED = 0
```

6.20.2.2 LEFT_BUTTON_PRESSED

```
LEFT_BUTTON_PRESSED = 0
```

6.20.2.3 RIGHT_BUTTON_PRESSED

```
RIGHT_BUTTON_PRESSED = 0
```

6.21 D:/Progetti/TTK4155/Node1/TTK4155/Interrupt.h File Reference

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

Functions

- void interrupt_init ()

  initialization function for the interrupt
- void interrupt_polling ()

  Resets variables and prints in the console if a button is pressed.
- ISR (INT2_vect)
- ISR (INT1_vect)

Variables

- volatile uint8_t RIGHT_BUTTON_PRESSED
- volatile uint8_t LEFT_BUTTON_PRESSED
- volatile uint8_t JOYSTICK_BUTTON_PRESSED

6.21.1 Function Documentation

6.21.1.1 interrupt_init()

```
void interrupt_init ( )
```

initialization function for the interrupt

6.21.1.2 interrupt_polling()

```
void interrupt_polling ( )
```

Resets variables and prints in the console if a button is pressed.

6.21.1.3 ISR() [1/2]

```
ISR (
             INT1_vect  )
```

6.21.1.4 ISR() [2/2]

```
ISR (
             INT2_vect  )
```

6.21.2 Variable Documentation

6.21.2.1 JOYSTICK_BUTTON_PRESSED

```
volatile uint8_t JOYSTICK_BUTTON_PRESSED
```

6.21.2.2 LEFT_BUTTON_PRESSED

```
volatile uint8_t LEFT_BUTTON_PRESSED
```

6.21.2.3 RIGHT_BUTTON_PRESSED

```
volatile uint8_t RIGHT_BUTTON_PRESSED
```

6.22 Interrupt.h

Go to the documentation of this file.

```
1 #ifndef INTERRUPT_H
2 #define INTERRUPT_H
3
4 #include <avr/io.h>
5 #include <avr/interrupt.h>
6
7 /*
8     INT2 -> RIGHT BUTTON
9     INT1 -> LEFT BUTTON
10     INT0 -> CAN INTERRUPT
11 */
12
13 volatile uint8_t RIGHT_BUTTON_PRESSED;
14 volatile uint8_t LEFT_BUTTON_PRESSED;
15 volatile uint8_t JOYSTICK_BUTTON_PRESSED;
16
21 void interrupt_init();
22
27 void interrupt_polling();
28
29 ISR(INT2_vect)
30 {
31     // Wakes the MCU up when right button is pressed
32     RIGHT_BUTTON_PRESSED = 1;
33 }
34
35 ISR(INT1_vect)
36 {
37     // Wakes the MCU up when left button is pressed
38     LEFT_BUTTON_PRESSED = 1;
39 }
40
41 #endif
```

6.23 D:/Progetti/TTK4155/Node1/TTK4155/Joystick.c File Reference

```
#include "Joystick.h"
```

Functions

- void joystick_init ()
  Initialization function for the joystick.
- pos_t joystick_pos_read ()
  Read the position of the joystick axes X,Y.
- void print_joystick_position ()
  Debug function for checking the position of the axes X,Y.
- direction joystick_dir_read ()
  Read the position of the joystick and converts to LEFT,RIGHT,UP,DOWN or IDLE if the joystick is untouched.
- void joystick_menu_navigation ()
  Updates the menu on the OLED when something happens with the joystick.
- void print_joystick_direction ()
  Debug function for checking the directoin of the joystick.

Variables

- stop = 0
- button = 0

6.23.1 Function Documentation

6.23.1.1 joystick_dir_read()

```
direction joystick_dir_read ( )
```

Read the position of the joystick and converts to LEFT,RIGHT,UP,DOWN or IDLE if the joystick is untouched.

Returns

direction

6.23.1.2 joystick_init()

```
void joystick_init ( )
```

Initialization function for the joystick.

6.23.1.3 joystick_menu_navigation()

```
void joystick_menu_navigation ( )
```

Updates the menu on the OLED when something happens with the joystick.

6.23.1.4 joystick_pos_read()

```
pos_t joystick_pos_read ( )
```

Read the position of the joystick axes X,Y.

Returns

pos_t

6.23.1.5 print_joystick_direction()

```
void print_joystick_direction ( )
```

Debug function for checking the directoin of the joystick.

6.23.1.6 print_joystick_position()

```
void print_joystick_position ( )
```

Debug function for checking the position of the axes X,Y.

6.23.2 Variable Documentation

6.23.2.1 button

```
button = 0
```

6.23.2.2 stop

```
stop = 0
```

6.24 D:/Progetti/TTK4155/Node1/TTK4155/Joystick.h File Reference

```
#include "ADC.h"
#include "OLED.h"
```

Data Structures

- struct pos_t
- struct direction

Macros

- #define J_X_ADDRESS 0x01
- #define J_Y_ADDRESS 0x00
- #define IDLE_X_MIN 20
- #define IDLE_X_MAX 80
- #define IDLE_Y_MIN 20
- #define IDLE_Y_MAX 80

Enumerations

- enum x_direction { X_IDLE , LEFT , RIGHT }
- enum y_direction { Y_IDLE , UP , DOWN }

Functions

- pos_t pos_read (void)

  Reads position of the joystick.
- void joystick_init ()

  Initialization function for the joystick.
- pos_t joystick_pos_read ()

  Read the position of the joystick axes X,Y.
- void print_joystick_position ()

  Debug function for checking the position of the axes X,Y.
- direction joystick_dir_read ()

  Read the position of the joystick and converts to LEFT,RIGHT,UP,DOWN or IDLE if the joystick is untouched.
- void joystick_menu_navigation ()

  Updates the menu on the OLED when something happens with the joystick.
- void print_joystick_direction ()

  Debug function for checking the directoin of the joystick.
- long map (long x, long in_min, long in_max, long out_min, long out_max)

  Function to remap values in a certain range.

Variables

- uint8_t stop
- uint8_t button

6.24.1 Macro Definition Documentation

6.24.1.1 IDLE_X_MAX

```
#define IDLE_X_MAX 80
```

6.24.1.2 IDLE_X_MIN

```
#define IDLE_X_MIN 20
```

6.24.1.3 IDLE_Y_MAX

```
#define IDLE_Y_MAX 80
```

6.24.1.4 IDLE_Y_MIN

```
#define IDLE_Y_MIN 20
```

6.24.1.5 J_X_ADDRESS

```
#define J_X_ADDRESS 0x01
```

6.24.1.6 J_Y_ADDRESS

```
#define J_Y_ADDRESS 0x00
```

6.24.2 Enumeration Type Documentation

6.24.2.1 x_direction

```
enum x_direction
```

Enumerator

X_IDLE	
LEFT	
RIGHT	

6.24.2.2 y_direction

```
enum y_direction
```

Enumerator

Y_IDLE	
UP	
DOWN	

6.24.3 Function Documentation

6.24.3.1 joystick_dir_read()

```
direction joystick_dir_read ( )
```

Read the position of the joystick and converts to LEFT,RIGHT,UP,DOWN or IDLE if the joystick is untouched.

Returns

direction

6.24.3.2 joystick_init()

```
void joystick_init ( )
```

Initialization function for the joystick.

6.24.3.3 joystick_menu_navigation()

```
void joystick_menu_navigation ( )
```

Updates the menu on the OLED when something happens with the joystick.

6.24.3.4 joystick_pos_read()

```
pos_t joystick_pos_read ( )
```

Read the position of the joystick axes X,Y.

Returns

pos_t

6.24.3.5 map()

```
long map (
            long x,
            long in_min,
            long in_max,
            long out_min,
            long out_max )
```

Function to remap values in a certain range.

Function to map a value in a certain range.

Parameters

x	The value to remap
in_min	Input range minumum value
in_max	Input range maximum value
out_min	Output range minumum value
out_max	Output range maximum value

Returns

long

6.24.3.6 pos_read()

```
pos_t pos_read (
            void  )
```

Reads position of the joystick.

Returns

pos_t

6.24.3.7 print_joystick_direction()

```
void print_joystick_direction ( )
```

Debug function for checking the directoin of the joystick.

6.24.3.8 print_joystick_position()

```
void print_joystick_position ( )
```

Debug function for checking the position of the axes X,Y.

6.24.4 Variable Documentation

6.24.4.1 button

```
uint8_t button
```

6.24.4.2 stop

```
uint8_t stop
```

6.25 Joystick.h

Go to the documentation of this file.

```
1 #ifndef JOYSTICK_H
2 #define JOYSTICK_H
3
4 #include "ADC.h"
5 #include "OLED.h"
6
7 // Joystick channel addresses
8 #define J_X_ADDRESS 0x01
9 #define J_Y_ADDRESS 0x00
10
11 // Idle values limit interval
12 #define IDLE_X_MIN 20
13 #define IDLE_X_MAX 80
14 #define IDLE_Y_MIN 20
15 #define IDLE_Y_MAX 80
16
17 // Position structure for x,y values
18 // 8-bit each value
19 typedef struct
20 {
21     uint8_t x;
22     uint8_t y;
23 } pos_t;
24
25 typedef enum
26 {
```

```
27     X_IDLE,
28     LEFT,
29     RIGHT
30 } x_direction;
31
32 typedef enum
33 {
34     Y_IDLE,
35     UP,
36     DOWN
37 } y_direction;
38
39 typedef struct
40 {
41     char x_dir;
42     char y_dir;
43 } direction;
44
45 // Prevents the joystick to get more than one input
46 uint8_t stop;
47 uint8_t button;
48
54 pos_t pos_read(void);
55
60 void joystick_init();
61
67 pos_t joystick_pos_read();
68
73 void print_joystick_position();
74
80 direction joystick_dir_read();
81
86 void joystick_menu_navigation();
87
92 void print_joystick_direction();
93
104 long map(long x, long in_min, long in_max, long out_min, long out_max)
105 {
106     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
107 }
108
109 #endif
```

6.26 D:/Progetti/TTK4155/Node1/TTK4155/main.c File Reference

```
#include <avr/io.h>
#include "USART.h"
#include "SRAM.h"
#include "ADC.h"
#include "Joystick.h"
#include "Interrupt.h"
#include "Slider.h"
#include "OLED.h"
#include "CAN.h"
#include "mcp2515.h"
#include <stdlib.h>
```

Macros

- #define FOSC 4915200
- #define BAUD 9600
- #define MYUBRR FOSC / 16 / BAUD - 1

Functions

- int main (void)

6.26.1 Macro Definition Documentation

6.26.1.1 BAUD

```
#define BAUD 9600
```

6.26.1.2 FOSC

```
#define FOSC 4915200
```

6.26.1.3 MYUBRR

```
#define MYUBRR FOSC / 16 / BAUD - 1
```

6.26.2 Function Documentation

6.26.2.1 main()

```
int main (
            void  )
```

6.27 D:/Progetti/TTK4155/Node2/Node2/main.c File Reference

Empty user application template.

```
#include "sam.h"
#include "uart.h"
#include "can_controller.h"
#include "can_interrupt.h"
#include "PWM.h"
#include "ADC.h"
#include "Motor.h"
```

Functions

- int main (void)

Variables

- uint8_t score = 0
- uint8_t game_pause = 0

6.27.1 Detailed Description

Empty user application template.

6.27.2 Function Documentation

6.27.2.1 main()

```
int main (
            void  )
```

6.27.3 Variable Documentation

6.27.3.1 game_pause

```
uint8_t game_pause = 0
```

6.27.3.2 score

```
uint8_t score = 0
```

6.28 D:/Progetti/TTK4155/Node1/TTK4155/mcp2515.h File Reference

Macros

- #define MCP_RXF0SIDH 0x00
- #define MCP_RXF0SIDL 0x01
- #define MCP_RXF0EID8 0x02
- #define MCP_RXF0EID0 0x03
- #define MCP_RXF1SIDH 0x04
- #define MCP_RXF1SIDL 0x05
- #define MCP_RXF1EID8 0x06
- #define MCP_RXF1EID0 0x07
- #define MCP_RXF2SIDH 0x08
- #define MCP_RXF2SIDL 0x09
- #define MCP_RXF2EID8 0x0A
- #define MCP_RXF2EID0 0x0B
- #define MCP_CANSTAT 0x0E
- #define MCP_CANCTRL 0x0F
- #define MCP_RXF3SIDH 0x10
- #define MCP_RXF3SIDL 0x11
- #define MCP_RXF3EID8 0x12
- #define MCP_RXF3EID0 0x13
- #define MCP_RXF4SIDH 0x14
- #define MCP_RXF4SIDL 0x15
- #define MCP_RXF4EID8 0x16
- #define MCP_RXF4EID0 0x17
- #define MCP_RXF5SIDH 0x18
- #define MCP_RXF5SIDL 0x19
- #define MCP_RXF5EID8 0x1A
- #define MCP_RXF5EID0 0x1B
- #define MCP_TEC 0x1C
- #define MCP_REC 0x1D
- #define MCP_RXM0SIDH 0x20
- #define MCP_RXM0SIDL 0x21
- #define MCP_RXM0EID8 0x22
- #define MCP_RXM0EID0 0x23
- #define MCP_RXM1SIDH 0x24
- #define MCP_RXM1SIDL 0x25
- #define MCP_RXM1EID8 0x26
- #define MCP_RXM1EID0 0x27
- #define MCP_CNF3 0x28
- #define MCP_CNF2 0x29
- #define MCP_CNF1 0x2A
- #define MCP_CANINTE 0x2B
- #define MCP_CANINTF 0x2C
- #define MCP_EFLG 0x2D
- #define MCP_TXB0CTRL 0x30
- #define MCP_TXB1CTRL 0x40
- #define MCP_TXB2CTRL 0x50
- #define MCP_RXB0CTRL 0x60
- #define MCP_RXB0SIDH 0x61
- #define MCP_RXB1CTRL 0x70
- #define MCP_RXB1SIDH 0x71
- #define MCP_TX_INT 0x1C

- #define MCP_TX01_INT 0x0C
- #define MCP_RX_INT 0x03
- #define MCP_NO_INT 0x00
- #define MCP_TX01_MASK 0x14
- #define MCP_TX_MASK 0x54
- #define MCP_WRITE 0x02
- #define MCP_READ 0x03
- #define MCP_BITMOD 0x05
- #define MCP_LOAD_TX0 0x40
- #define MCP_LOAD_TX1 0x42
- #define MCP_LOAD_TX2 0x44
- #define MCP_RTS_TX0 0x81
- #define MCP_RTS_TX1 0x82
- #define MCP_RTS_TX2 0x84
- #define MCP_RTS_ALL 0x87
- #define MCP_READ_RX0 0x90
- #define MCP_READ_RX1 0x94
- #define MCP_READ_STATUS 0xA0
- #define MCP_RX_STATUS 0xB0
- #define MCP_RESET 0xC0
- #define MODE_NORMAL 0x00
- #define MODE_SLEEP 0x20
- #define MODE_LOOPBACK 0x40
- #define MODE_LISTENONLY 0x60
- #define MODE_CONFIG 0x80
- #define MODE_POWERUP 0xE0
- #define MODE_MASK 0xE0
- #define ABORT_TX 0x10
- #define MODE_ONESHOT 0x08
- #define CLKOUT_ENABLE 0x04
- #define CLKOUT_DISABLE 0x00
- #define CLKOUT_PS1 0x00
- #define CLKOUT_PS2 0x01
- #define CLKOUT_PS4 0x02
- #define CLKOUT_PS8 0x03
- #define SJW1 0x00
- #define SJW2 0x40
- #define SJW3 0x80
- #define SJW4 0xC0
- #define BTLMODE 0x80
- #define SAMPLE_1X 0x00
- #define SAMPLE_3X 0x40
- #define SOF_ENABLE 0x80
- #define SOF_DISABLE 0x00
- #define WAKFIL_ENABLE 0x40
- #define WAKFIL_DISABLE 0x00
- #define MCP_RX0IF 0x01
- #define MCP_RX1IF 0x02
- #define MCP_TX0IF 0x04
- #define MCP_TX1IF 0x08
- #define MCP_TX2IF 0x10
- #define MCP_ERRIF 0x20
- #define MCP_WAKIF 0x40
- #define MCP_MERRF 0x80

6.28.1 Macro Definition Documentation

6.28.1.1 ABORT_TX

```
#define ABORT_TX 0x10
```

6.28.1.2 BTLMODE

```
#define BTLMODE 0x80
```

6.28.1.3 CLKOUT_DISABLE

```
#define CLKOUT_DISABLE 0x00
```

6.28.1.4 CLKOUT_ENABLE

```
#define CLKOUT_ENABLE 0x04
```

6.28.1.5 CLKOUT_PS1

```
#define CLKOUT_PS1 0x00
```

6.28.1.6 CLKOUT_PS2

```
#define CLKOUT_PS2 0x01
```

6.28.1.7 CLKOUT_PS4

```
#define CLKOUT_PS4 0x02
```

6.28.1.8 CLKOUT_PS8

```
#define CLKOUT_PS8 0x03
```

6.28.1.9 MCP_BITMOD

```
#define MCP_BITMOD 0x05
```

6.28.1.10 MCP_CANCTRL

```
#define MCP_CANCTRL 0x0F
```

6.28.1.11 MCP_CANINTE

```
#define MCP_CANINTE 0x2B
```

6.28.1.12 MCP_CANINTF

```
#define MCP_CANINTF 0x2C
```

6.28.1.13 MCP_CANSTAT

```
#define MCP_CANSTAT 0x0E
```

6.28.1.14 MCP_CNF1

```
#define MCP_CNF1 0x2A
```

6.28.1.15 MCP_CNF2

```
#define MCP_CNF2 0x29
```

6.28.1.16 MCP_CNF3

```
#define MCP_CNF3 0x28
```

6.28.1.17 MCP_EFLG

```
#define MCP_EFLG 0x2D
```

6.28.1.18 MCP_ERRIF

```
#define MCP_ERRIF 0x20
```

6.28.1.19 MCP_LOAD_TX0

```
#define MCP_LOAD_TX0 0x40
```

6.28.1.20 MCP_LOAD_TX1

```
#define MCP_LOAD_TX1 0x42
```

6.28.1.21 MCP_LOAD_TX2

```
#define MCP_LOAD_TX2 0x44
```

6.28.1.22 MCP_MERRF

```
#define MCP_MERRF 0x80
```

6.28.1.23 MCP_NO_INT

```
#define MCP_NO_INT 0x00
```

6.28.1.24 MCP_READ

```
#define MCP_READ 0x03
```

6.28.1.25 MCP_READ_RX0

```
#define MCP_READ_RX0 0x90
```

6.28.1.26 MCP_READ_RX1

```
#define MCP_READ_RX1 0x94
```

6.28.1.27 MCP_READ_STATUS

```
#define MCP_READ_STATUS 0xA0
```

6.28.1.28 MCP_REC

```
#define MCP_REC 0x1D
```

6.28.1.29 MCP_RESET

```
#define MCP_RESET 0xC0
```

6.28.1.30 MCP_RTS_ALL

```
#define MCP_RTS_ALL 0x87
```

6.28.1.31 MCP_RTS_TX0

```
#define MCP_RTS_TX0 0x81
```

6.28.1.32 MCP_RTS_TX1

```
#define MCP_RTS_TX1 0x82
```

6.28.1.33 MCP_RTS_TX2

```
#define MCP_RTS_TX2 0x84
```

6.28.1.34 MCP_RX0IF

```
#define MCP_RX0IF 0x01
```

6.28.1.35 MCP_RX1IF

```
#define MCP_RX1IF 0x02
```

6.28.1.36 MCP_RX_INT

```
#define MCP_RX_INT 0x03
```

6.28.1.37 MCP_RX_STATUS

```
#define MCP_RX_STATUS 0xB0
```

6.28.1.38 MCP_RXB0CTRL

```
#define MCP_RXB0CTRL 0x60
```

6.28.1.39 MCP_RXB0SIDH

```
#define MCP_RXB0SIDH 0x61
```

6.28.1.40 MCP_RXB1CTRL

```
#define MCP_RXB1CTRL 0x70
```

6.28.1.41 MCP_RXB1SIDH

```
#define MCP_RXB1SIDH 0x71
```

6.28.1.42 MCP_RXF0EID0

```
#define MCP_RXF0EID0 0x03
```

6.28.1.43 MCP_RXF0EID8

```
#define MCP_RXF0EID8 0x02
```

6.28.1.44 MCP_RXF0SIDH

```
#define MCP_RXF0SIDH 0x00
```

6.28.1.45 MCP_RXF0SIDL

```
#define MCP_RXF0SIDL 0x01
```

6.28.1.46 MCP_RXF1EID0

```
#define MCP_RXF1EID0 0x07
```

6.28.1.47 MCP_RXF1EID8

```
#define MCP_RXF1EID8 0x06
```

6.28.1.48 MCP_RXF1SIDH

```
#define MCP_RXF1SIDH 0x04
```

6.28.1.49 MCP_RXF1SIDL

```
#define MCP_RXF1SIDL 0x05
```

6.28.1.50 MCP_RXF2EID0

```
#define MCP_RXF2EID0 0x0B
```

6.28.1.51 MCP_RXF2EID8

```
#define MCP_RXF2EID8 0x0A
```

6.28.1.52 MCP_RXF2SIDH

```
#define MCP_RXF2SIDH 0x08
```

6.28.1.53 MCP_RXF2SIDL

```
#define MCP_RXF2SIDL 0x09
```

6.28.1.54 MCP_RXF3EID0

```
#define MCP_RXF3EID0 0x13
```

6.28.1.55 MCP_RXF3EID8

```
#define MCP_RXF3EID8 0x12
```

6.28.1.56 MCP_RXF3SIDH

```
#define MCP_RXF3SIDH 0x10
```

6.28.1.57 MCP_RXF3SIDL

```
#define MCP_RXF3SIDL 0x11
```

6.28.1.58 MCP_RXF4EID0

```
#define MCP_RXF4EID0 0x17
```

6.28.1.59 MCP_RXF4EID8

```
#define MCP_RXF4EID8 0x16
```

6.28.1.60 MCP_RXF4SIDH

```
#define MCP_RXF4SIDH 0x14
```

6.28.1.61 MCP_RXF4SIDL

```
#define MCP_RXF4SIDL 0x15
```

6.28.1.62 MCP_RXF5EID0

```
#define MCP_RXF5EID0 0x1B
```

6.28.1.63 MCP_RXF5EID8

```
#define MCP_RXF5EID8 0x1A
```

6.28.1.64 MCP_RXF5SIDH

```
#define MCP_RXF5SIDH 0x18
```

6.28.1.65 MCP_RXF5SIDL

```
#define MCP_RXF5SIDL 0x19
```

6.28.1.66 MCP_RXM0EID0

```
#define MCP_RXM0EID0 0x23
```

6.28.1.67 MCP_RXM0EID8

```
#define MCP_RXM0EID8 0x22
```

6.28.1.68 MCP_RXM0SIDH

```
#define MCP_RXM0SIDH 0x20
```

6.28.1.69 MCP_RXM0SIDL

```
#define MCP_RXM0SIDL 0x21
```

6.28.1.70 MCP_RXM1EID0

```
#define MCP_RXM1EID0 0x27
```

6.28.1.71 MCP_RXM1EID8

```
#define MCP_RXM1EID8 0x26
```

6.28.1.72 MCP_RXM1SIDH

```
#define MCP_RXM1SIDH 0x24
```

6.28.1.73 MCP_RXM1SIDL

```
#define MCP_RXM1SIDL 0x25
```

6.28.1.74 MCP_TEC

```
#define MCP_TEC 0x1C
```

6.28.1.75 MCP_TX01_INT

```
#define MCP_TX01_INT 0x0C
```

6.28.1.76 MCP_TX01_MASK

```
#define MCP_TX01_MASK 0x14
```

6.28.1.77 MCP_TX0IF

```
#define MCP_TX0IF 0x04
```

6.28.1.78 MCP_TX1IF

```
#define MCP_TX1IF 0x08
```

6.28.1.79 MCP_TX2IF

```
#define MCP_TX2IF 0x10
```

6.28.1.80 MCP_TX_INT

```
#define MCP_TX_INT 0x1C
```

6.28.1.81 MCP_TX_MASK

```
#define MCP_TX_MASK 0x54
```

6.28.1.82 MCP_TXB0CTRL

```
#define MCP_TXB0CTRL 0x30
```

6.28.1.83 MCP_TXB1CTRL

```
#define MCP_TXB1CTRL 0x40
```

6.28.1.84 MCP_TXB2CTRL

```
#define MCP_TXB2CTRL 0x50
```

6.28.1.85 MCP_WAKIF

```
#define MCP_WAKIF 0x40
```

6.28.1.86 MCP_WRITE

```
#define MCP_WRITE 0x02
```

6.28.1.87 MODE_CONFIG

```
#define MODE_CONFIG 0x80
```

6.28.1.88 MODE_LISTENONLY

```
#define MODE_LISTENONLY 0x60
```

6.28.1.89 MODE_LOOPBACK

```
#define MODE_LOOPBACK 0x40
```

6.28.1.90 MODE_MASK

```
#define MODE_MASK 0xE0
```

6.28.1.91 MODE_NORMAL

```
#define MODE_NORMAL 0x00
```

6.28.1.92 MODE_ONESHOT

```
#define MODE_ONESHOT 0x08
```

6.28.1.93 MODE_POWERUP

```
#define MODE_POWERUP 0xE0
```

6.28.1.94 MODE_SLEEP

```
#define MODE_SLEEP 0x20
```

6.28.1.95 SAMPLE_1X

```
#define SAMPLE_1X 0x00
```

6.28.1.96 SAMPLE_3X

```
#define SAMPLE_3X 0x40
```

6.28.1.97 SJW1

```
#define SJW1 0x00
```

6.28.1.98 SJW2

```
#define SJW2 0x40
```

6.28.1.99 SJW3

```
#define SJW3 0x80
```

6.28.1.100 SJW4

```
#define SJW4 0xC0
```

6.28.1.101 SOF_DISABLE

```
#define SOF_DISABLE 0x00
```

6.28.1.102 SOF_ENABLE

```
#define SOF_ENABLE 0x80
```

6.28.1.103 WAKFIL_DISABLE

```
#define WAKFIL_DISABLE 0x00
```

6.28.1.104 WAKFIL_ENABLE

```
#define WAKFIL_ENABLE 0x40
```

6.29 mcp2515.h

Go to the documentation of this file.

```
1 #ifndef __MCP2515_H
2 #define __MCP2515_H
3
4 /*
5 mcp2515.h
6
7 This file contains constants that are specific to the MCP2515.
8
9 Version     Date        Description
10 ----------------------------------------------------------------------
11 v1.00       2003/12/11  Initial release
12
13 Copyright 2003 Kimberly Otten Software Consulting
14 */
15
16 // Define MCP2515 register addresses
17
18 #define MCP_RXF0SIDH    0x00
19 #define MCP_RXF0SIDL    0x01
20 #define MCP_RXF0EID8    0x02
21 #define MCP_RXF0EID0    0x03
22 #define MCP_RXF1SIDH    0x04
23 #define MCP_RXF1SIDL    0x05
24 #define MCP_RXF1EID8    0x06
25 #define MCP_RXF1EID0    0x07
26 #define MCP_RXF2SIDH    0x08
27 #define MCP_RXF2SIDL    0x09
28 #define MCP_RXF2EID8    0x0A
29 #define MCP_RXF2EID0    0x0B
30 #define MCP_CANSTAT     0x0E
31 #define MCP_CANCTRL     0x0F
32 #define MCP_RXF3SIDH    0x10
33 #define MCP_RXF3SIDL    0x11
34 #define MCP_RXF3EID8    0x12
35 #define MCP_RXF3EID0    0x13
36 #define MCP_RXF4SIDH    0x14
37 #define MCP_RXF4SIDL    0x15
38 #define MCP_RXF4EID8    0x16
39 #define MCP_RXF4EID0    0x17
40 #define MCP_RXF5SIDH    0x18
41 #define MCP_RXF5SIDL    0x19
42 #define MCP_RXF5EID8    0x1A
43 #define MCP_RXF5EID0    0x1B
44 #define MCP_TEC         0x1C
45 #define MCP_REC         0x1D
46 #define MCP_RXM0SIDH    0x20
47 #define MCP_RXM0SIDL    0x21
48 #define MCP_RXM0EID8    0x22
49 #define MCP_RXM0EID0    0x23
50 #define MCP_RXM1SIDH    0x24
51 #define MCP_RXM1SIDL    0x25
52 #define MCP_RXM1EID8    0x26
53 #define MCP_RXM1EID0    0x27
54 #define MCP_CNF3        0x28
55 #define MCP_CNF2        0x29
56 #define MCP_CNF1        0x2A
57 #define MCP_CANINTE     0x2B
58 #define MCP_CANINTF     0x2C
59 #define MCP_EFLG        0x2D
60 #define MCP_TXB0CTRL    0x30
61 #define MCP_TXB1CTRL    0x40
62 #define MCP_TXB2CTRL    0x50
63 #define MCP_RXB0CTRL    0x60
64 #define MCP_RXB0SIDH    0x61
65 #define MCP_RXB1CTRL    0x70
66 #define MCP_RXB1SIDH    0x71
67
68
69 #define MCP_TX_INT      0x1C        // Enable all transmit interrupts
70 #define MCP_TX01_INT    0x0C        // Enable TXB0 and TXB1 interrupts
71 #define MCP_RX_INT      0x03        // Enable receive interrupts
72 #define MCP_NO_INT      0x00        // Disable all interrupts
```

```
73
74 #define MCP_TX01_MASK   0x14
75 #define MCP_TX_MASK     0x54
76
77 // Define SPI Instruction Set
78
79 #define MCP_WRITE       0x02
80
81 #define MCP_READ        0x03
82
83 #define MCP_BITMOD      0x05
84
85 #define MCP_LOAD_TX0    0x40
86 #define MCP_LOAD_TX1    0x42
87 #define MCP_LOAD_TX2    0x44
88
89 #define MCP_RTS_TX0     0x81
90 #define MCP_RTS_TX1     0x82
91 #define MCP_RTS_TX2     0x84
92 #define MCP_RTS_ALL     0x87
93
94 #define MCP_READ_RX0    0x90
95 #define MCP_READ_RX1    0x94
96
97 #define MCP_READ_STATUS 0xA0
98
99 #define MCP_RX_STATUS   0xB0
100
101 #define MCP_RESET       0xC0
102
103
104 // CANCTRL Register Values
105
106 #define MODE_NORMAL     0x00
107 #define MODE_SLEEP      0x20
108 #define MODE_LOOPBACK   0x40
109 #define MODE_LISTENONLY 0x60
110 #define MODE_CONFIG     0x80
111 #define MODE_POWERUP    0xE0
112 #define MODE_MASK       0xE0
113 #define ABORT_TX        0x10
114 #define MODE_ONESHOT    0x08
115 #define CLKOUT_ENABLE   0x04
116 #define CLKOUT_DISABLE  0x00
117 #define CLKOUT_PS1      0x00
118 #define CLKOUT_PS2      0x01
119 #define CLKOUT_PS4      0x02
120 #define CLKOUT_PS8      0x03
121
122
123 // CNF1 Register Values
124
125 #define SJW1            0x00
126 #define SJW2            0x40
127 #define SJW3            0x80
128 #define SJW4            0xC0
129
130
131 // CNF2 Register Values
132
133 #define BTLMODE         0x80
134 #define SAMPLE_1X       0x00
135 #define SAMPLE_3X       0x40
136
137
138 // CNF3 Register Values
139
140 #define SOF_ENABLE      0x80
141 #define SOF_DISABLE     0x00
142 #define WAKFIL_ENABLE   0x40
143 #define WAKFIL_DISABLE  0x00
144
145
146 // CANINTF Register Bits
147
148 #define MCP_RX0IF       0x01
149 #define MCP_RX1IF       0x02
150 #define MCP_TX0IF       0x04
151 #define MCP_TX1IF       0x08
152 #define MCP_TX2IF       0x10
153 #define MCP_ERRIF       0x20
154 #define MCP_WAKIF       0x40
155 #define MCP_MERRF       0x80
156
157
158
159 #endif
```

6.30 D:/Progetti/TTK4155/Node1/TTK4155/OLED.c File Reference

```
#include "OLED.h"
```

Functions

- void OLED_write_command (uint8_t c)
  Send command to the OLED controller.
- void OLED_write_data (uint8_t d)
  Write data to the OLED.
- void OLED_init ()
  Initialization function for the OLED.
- void OLED_goto_line (uint8_t line)
  Set cursor to a certain line of the OLEd.
- void OLED_goto_column (uint8_t column)
  Set the cursor to a certain column.
- void OLED_set_pos (uint8_t line, uint8_t column)
  Set the cursor to a certain line and column.
- void OLED_fill_line (uint8_t line)
  Fill line with white pixels.
- void OLED_clear_all ()
  Clear the screen.
- void OLED_print_char (char c, uint8_t line, uint8_t col)
  Print character on the OLED.
- void OLED_print_string (char *s, uint8_t length, uint8_t line)
  Print string to the OLED.
- void OLED_update_menu (uint8_t pos)
  Update the cursor on the menu.
- void OLED_print_menu (uint8_t menutype)
  Print menu on the OLED.

Variables

- main_menu [4] = {{"Play", 4}, {"Settings", 8}, {"Credits", 7}, {"About", 5}}
- menu_pos = 1

6.30.1 Function Documentation

6.30.1.1 OLED_clear_all()

```
void OLED_clear_all ( )
```

Clear the screen.

6.30.1.2 OLED_fill_line()

```
void OLED_fill_line (
            uint8_t line )
```

Fill line with white pixels.

Parameters

line	Line number

6.30.1.3 OLED_goto_column()

```
void OLED_goto_column (
            uint8_t column )
```

Set the cursor to a certain column.

Parameters

column	Column number

6.30.1.4 OLED_goto_line()

```
void OLED_goto_line (
            uint8_t line )
```

Set cursor to a certain line of the OLEd.

Parameters

line	Line number

6.30.1.5 OLED_init()

```
void OLED_init ( )
```

Initialization function for the OLED.

6.30.1.6 OLED_print_char()

```
void OLED_print_char (
            char c,
            uint8_t line,
            uint8_t col )
```

Print character on the OLED.

Parameters

c	Character
line	line number
col	Column number

6.30.1.7 OLED_print_menu()

```
void OLED_print_menu (
            uint8_t menutype )
```

Print menu on the OLED.

Parameters

menutype	

6.30.1.8 OLED_print_string()

```
void OLED_print_string (
            char * s,
            uint8_t length,
            uint8_t line )
```

Print string to the OLED.

Parameters

s	String
length	Length of the string
line	Line number

6.30.1.9 OLED_set_pos()

```
void OLED_set_pos (
            uint8_t line,
            uint8_t column )
```

Set the cursor to a certain line and column.

Parameters

line	Line number
column	Column number

6.30.1.10 OLED_update_menu()

```
void OLED_update_menu (
            uint8_t pos )
```

Update the cursor on the menu.

Parameters

pos	

6.30.1.11 OLED_write_command()

```
void OLED_write_command (
            uint8_t c )
```

Send command to the OLED controller.

Parameters

c	Command to be sent

6.30.1.12 OLED_write_data()

```
void OLED_write_data (
            uint8_t d )
```

Write data to the OLED.

Parameters

d	Data to write

6.30.2 Variable Documentation

6.30.2.1 main_menu

```
main_menu[4] = {{"Play", 4}, {"Settings", 8}, {"Credits", 7}, {"About", 5}}
```

6.30.2.2 menu_pos

```
menu_pos = 1
```

6.31 D:/Progetti/TTK4155/Node1/TTK4155/OLED.h File Reference

```
#include <avr/delay.h>
#include "fonts.h"
```

Data Structures

- struct menu_entry

Macros

- #define OLED_COMMAND_PIN 3
- #define HEIGHT 64
- #define WIDTH 128

Enumerations

- enum menu_name {
  MAIN , PLAYMENU , SETTINGS , CREDITS ,
  ABOUT }

Functions

- void OLED_write_command (uint8_t c)

  Send command to the OLED controller.
- void OLED_write_data (uint8_t d)

  Write data to the OLED.
- void OLED_init ()

  Initialization function for the OLED.
- void OLED_goto_line (uint8_t line)

  Set cursor to a certain line of the OLEd.
- void OLED_goto_column (uint8_t column)

  Set the cursor to a certain column.
- void OLED_set_pos (uint8_t line, uint8_t column)

  Set the cursor to a certain line and column.
- void OLED_fill_line (uint8_t line)

  Fill line with white pixels.
- void OLED_clear_all ()

  Clear the screen.
- void OLED_print_char (char c, uint8_t line, uint8_t col)

  Print character on the OLED.
- void OLED_print_string (char ∗s, uint8_t length, uint8_t line)

  Print string to the OLED.
- void OLED_update_menu (uint8_t pos)

  Update the cursor on the menu.
- void OLED_print_menu (uint8_t menutype)

  Print menu on the OLED.

Variables

- volatile char ∗ OLED_COMMAND = (char ∗)0x1000
- volatile char ∗ OLED_DATA = (char ∗)0x1200
- menu_entry main_menu [4]
- volatile uint8_t menu_pos

6.31.1 Macro Definition Documentation

6.31.1.1 HEIGHT

```
#define HEIGHT 64
```

6.31.1.2 OLED_COMMAND_PIN

```
#define OLED_COMMAND_PIN 3
```

6.31.1.3 WIDTH

```
#define WIDTH 128
```

6.31.2 Enumeration Type Documentation

6.31.2.1 menu_name

```
enum menu_name
```

Enumerator

MAIN	
PLAYMENU	
SETTINGS	
CREDITS	
ABOUT	

6.31.3 Function Documentation

6.31.3.1 OLED_clear_all()

```
void OLED_clear_all ( )
```

Clear the screen.

6.31.3.2 OLED_fill_line()

```
void OLED_fill_line (
            uint8_t line )
```

Fill line with white pixels.

Parameters

line	Line number

6.31.3.3 OLED_goto_column()

```
void OLED_goto_column (
            uint8_t column )
```

Set the cursor to a certain column.

Parameters

column	Column number

6.31.3.4 OLED_goto_line()

```
void OLED_goto_line (
            uint8_t line )
```

Set cursor to a certain line of the OLEd.

Parameters

line	Line number

6.31.3.5 OLED_init()

```
void OLED_init ( )
```

Initialization function for the OLED.

6.31.3.6 OLED_print_char()

```
void OLED_print_char (
            char c,
            uint8_t line,
            uint8_t col )
```

Print character on the OLED.

Parameters

c	Character
line	line number
col	Column number

6.31.3.7 OLED_print_menu()

```
void OLED_print_menu (
            uint8_t menutype )
```

Print menu on the OLED.

Parameters

menutype	

6.31.3.8 OLED_print_string()

```
void OLED_print_string (
            char * s,
            uint8_t length,
            uint8_t line )
```

Print string to the OLED.

Parameters

s	String
length	Length of the string
line	Line number

6.31.3.9 OLED_set_pos()

```
void OLED_set_pos (
            uint8_t line,
            uint8_t column )
```

Set the cursor to a certain line and column.

Parameters

line	Line number
column	Column number

6.31.3.10 OLED_update_menu()

```
void OLED_update_menu (
            uint8_t pos )
```

Update the cursor on the menu.

Parameters

pos	

6.31.3.11 OLED_write_command()

```
void OLED_write_command (
            uint8_t c )
```

Send command to the OLED controller.

Parameters

c	Command to be sent

6.31.3.12 OLED_write_data()

```
void OLED_write_data (
            uint8_t d )
```

Write data to the OLED.

Parameters

d	Data to write

6.31.4 Variable Documentation

6.31.4.1 main_menu

```
menu_entry main_menu[4]
```

6.31.4.2 menu_pos

```
volatile uint8_t menu_pos
```

6.31.4.3 OLED_COMMAND

```
volatile char* OLED_COMMAND = (char *)0x1000
```

6.31.4.4 OLED_DATA

```
volatile char* OLED_DATA = (char *)0x1200
```

6.32 OLED.h

Go to the documentation of this file.

```
1 #ifndef OLED_H
2 #define OLED_H
3
4 #include <avr/delay.h>
5
6 #include "fonts.h"
7
8 #define OLED_COMMAND_PIN 3
9 volatile char *OLED_COMMAND = (char *)0x1000;
10 volatile char *OLED_DATA = (char *)0x1200;
11
12 #define HEIGHT 64
13 #define WIDTH 128
14
15 // Types of available menu
16 typedef enum
17 {
18     MAIN,
19     PLAYMENU,
20     SETTINGS,
21     CREDITS,
22     ABOUT
23 } menu_name;
24
25 // Struct to have both text and length of string on the same place
26 typedef struct
27 {
28     char *name;
29     uint8_t length;
30 } menu_entry;
31
32 // Creating menus HERE:
33 menu_entry main_menu[4];
34
35 // Current cursor position on the screen (line number)
36 volatile uint8_t menu_pos;
37
43 void OLED_write_command(uint8_t c);
44
50 void OLED_write_data(uint8_t d);
51
56 void OLED_init();
57
63 void OLED_goto_line(uint8_t line);
64
70 void OLED_goto_column(uint8_t column);
71
78 void OLED_set_pos(uint8_t line, uint8_t column);
79
```

```
85 void OLED_fill_line(uint8_t line);
86
91 void OLED_clear_all();
92
100 void OLED_print_char(char c, uint8_t line, uint8_t col);
101
109 void OLED_print_string(char *s, uint8_t length, uint8_t line);
110
116 void OLED_update_menu(uint8_t pos);
117
123 void OLED_print_menu(uint8_t menutype);
124
125 #endif
```

6.33 D:/Progetti/TTK4155/Node1/TTK4155/Slider.c File Reference

```
#include "Slider.h"
```

Functions

- sliderpos_t slider_pos_read ()
  Read the position of both sliders.
- void print_slider_position ()
  Print position of both sliders L.R.

6.33.1 Function Documentation

6.33.1.1 print_slider_position()

```
void print_slider_position ( )
```

Print position of both sliders L.R.

6.33.1.2 slider_pos_read()

```
sliderpos_t slider_pos_read ( )
```

Read the position of both sliders.

Returns

sliderpos_t

6.34 D:/Progetti/TTK4155/Node1/TTK4155/Slider.h File Reference

```
#include "ADC.h"
```

Data Structures

- struct sliderpos_t

  Data type for sliders position L (left), R (right)

Macros

- #define S_L_ADDRESS 0x03
- #define S_R_ADDRESS 0x02

Functions

- sliderpos_t slider_pos_read ()

  Read the position of both sliders.
- void print_slider_position ()

  Print position of both sliders L.R.

6.34.1 Macro Definition Documentation

6.34.1.1 S_L_ADDRESS

```
#define S_L_ADDRESS 0x03
```

6.34.1.2 S_R_ADDRESS

```
#define S_R_ADDRESS 0x02
```

6.34.2 Function Documentation

6.34.2.1 print_slider_position()

```
void print_slider_position ( )
```

Print position of both sliders L.R.

6.34.2.2 slider_pos_read()

```
sliderpos_t slider_pos_read ( )
```

Read the position of both sliders.

Returns

sliderpos_t

6.35 Slider.h

Go to the documentation of this file.

```
1 #ifndef SLIDER_H
2 #define SLIDER_H
3
4 #include "ADC.h"
5
6 // Joystick channel addresses
7 #define S_L_ADDRESS 0x03
8 #define S_R_ADDRESS 0x02
9
14 typedef struct
15 {
16     uint8_t L;
17     uint8_t R;
18 } sliderpos_t;
19
25 sliderpos_t slider_pos_read();
26
31 void print_slider_position();
32
33 #endif
```

6.36 D:/Progetti/TTK4155/Node1/TTK4155/SPI.c File Reference

```
#include "SPI.h"
```

Functions

- void SPI_slave_enable ()
  Enable the SPI slave.
- void SPI_slave_disable ()
  Disable the SPI slave.
- void SPI_master_init ()
  Initialize the SPI master.
- void SPI_master_transmit (char cData)
  Transmit data with SPI.
- char SPI_master_read ()
  Read data with SPI.

6.36.1 Function Documentation

6.36.1.1 SPI_master_init()

```
void SPI_master_init ( )
```

Initialize the SPI master.

6.36.1.2 SPI_master_read()

```
char SPI_master_read ( )
```

Read data with SPI.

Returns

char

6.36.1.3 SPI_master_transmit()

```
void SPI_master_transmit (
            char cData )
```

Transmit data with SPI.

Parameters

cData	

6.36.1.4 SPI_slave_disable()

```
void SPI_slave_disable ( )
```

Disable the SPI slave.

6.36.1.5 SPI_slave_enable()

```
void SPI_slave_enable ( )
```

Enable the SPI slave.

6.37 D:/Progetti/TTK4155/Node1/TTK4155/SPI.h File Reference

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

Functions

- void SPI_slave_enable ()
  Enable the SPI slave.
- void SPI_slave_disable ()
  Disable the SPI slave.
- void SPI_master_init ()
  Initialize the SPI master.
- void SPI_master_transmit (char cData)
  Transmit data with SPI.
- char SPI_master_read ()
  Read data with SPI.

6.37.1 Function Documentation

6.37.1.1 SPI_master_init()

```
void SPI_master_init ( )
```

Initialize the SPI master.

6.37.1.2 SPI_master_read()

```
char SPI_master_read ( )
```

Read data with SPI.

Returns

char

6.37.1.3 SPI_master_transmit()

```
void SPI_master_transmit (
            char cData )
```

Transmit data with SPI.

Parameters

cData	

6.37.1.4 SPI_slave_disable()

```
void SPI_slave_disable ( )
```

Disable the SPI slave.

6.37.1.5 SPI_slave_enable()

```
void SPI_slave_enable ( )
```

Enable the SPI slave.

6.38 SPI.h

Go to the documentation of this file.

```
1 #ifndef SPI_H
2 #define SPI_H
3
4 #include <avr/io.h>
5
10 void SPI_slave_enable();
11
16 void SPI_slave_disable();
17
22 void SPI_master_init();
23
29 void SPI_master_transmit(char cData);
30
36 char SPI_master_read();
37
38 #endif
```

6.39 D:/Progetti/TTK4155/Node1/TTK4155/SRAM.h File Reference

Functions

- void SRAM_init (void)

  Initialization function for the SRAM.
- void SRAM_test (void)

  Test the SRAM.

6.39.1 Function Documentation

6.39.1.1 SRAM_init()

```
void SRAM_init (
            void  )
```

Initialization function for the SRAM.

6.39.1.2 SRAM_test()

```
void SRAM_test (
            void  )
```

Test the SRAM.

6.40 SRAM.h

Go to the documentation of this file.

```
1 #ifndef SRAM_H
2 #define SRAM_H
3
8 void SRAM_init(void)
9 {
10     /*Enabling memory reading*/
11     MCUCR |= (1 << SRE);
12     SFIOR |= (1 << XMM2);
13 }
14
19 void SRAM_test(void)
20 {
21     volatile char *ext_ram = (char *)0x1800; // Start address for the SRAM
22     uint16_t ext_ram_size = 0x800;
23     uint16_t write_errors = 0;
24     uint16_t retrieval_errors = 0;
25     printf("Starting SRAM test...\r\n");
26     // rand() stores some internal state, so calling this function in a loop will
27     // yield different seeds each time (unless srand() is called before this function)
28     uint16_t seed = rand();
29     // Write phase: Immediately check that the correct value was stored
30     srand(seed);
31     for (uint16_t i = 0; i < ext_ram_size; i++)
32     {
33         uint8_t some_value = rand();
34         ext_ram[i] = some_value;
35         uint8_t retreived_value = ext_ram[i];
36         if (retreived_value != some_value)
37         {
38             printf("Write phase error: ext_ram[%4d] = %02X (should be %02X)\r\n", i, retreived_value,
    some_value);
39             write_errors++;
40         }
41     }
42     // Retrieval phase: Check that no values were changed during or after the write phase
43     srand(seed);
44     // Reset the PRNG to the state it had before the write phase
45     for (uint16_t i = 0; i < ext_ram_size; i++)
46     {
47         uint8_t some_value = rand();
48         uint8_t retreived_value = ext_ram[i];
49         if (retreived_value != some_value)
50         {
51             printf("Retrieval phase error: ext_ram[%4d] = %02X (should be %02X)\r\n", i, retreived_value,
    some_value);
52             retrieval_errors++;
53         }
54     }
55     printf("SRAM test completed with \r\n%4d errors in write phase and \r\n%4d errors in retrieval
    phase\r\n", write_errors, retrieval_errors);
56 }
57
58 #endif
```

6.41 D:/Progetti/TTK4155/Node1/TTK4155/USART.c File Reference

```
#include "USART.h"
```

Functions

- void USART_Init (unsigned int ubrr)

  Initialization function for the USART.
- int USART_Transmit (unsigned char data)

  Transmit data over the USART.
- int USART_Receive (void)

  Receives data over the USART.

6.41.1 Function Documentation

6.41.1.1 USART_Init()

```
void USART_Init (
            unsigned int ubrr )
```

Initialization function for the USART.

Parameters

ubrr	USART bitrate

6.41.1.2 USART_Receive()

```
int USART_Receive ( )
```

Receives data over the USART.

Returns

int Received data

6.41.1.3 USART_Transmit()

```
int USART_Transmit (
            unsigned char data )
```

Transmit data over the USART.

Parameters

data	

Returns

int

6.42 D:/Progetti/TTK4155/Node1/TTK4155/USART.h File Reference

```
#include <stdio.h>
```

Functions

- void USART_Init (unsigned int ubrr)

  Initialization function for the USART.
- int USART_Transmit (unsigned char data)

  Transmit data over the USART.
- int USART_Receive ()

  Receives data over the USART.

6.42.1 Function Documentation

6.42.1.1 USART_Init()

```
void USART_Init (
            unsigned int ubrr )
```

Initialization function for the USART.

Parameters

ubrr	USART bitrate

6.42.1.2 USART_Receive()

```
int USART_Receive ( )
```

Receives data over the USART.

Returns

int Received data

6.42.1.3 USART_Transmit()

```
int USART_Transmit (
            unsigned char data )
```

Transmit data over the USART.

Parameters

data	

Returns

int

6.43 USART.h

Go to the documentation of this file.

```
1 #ifndef USART_H
2 #define USART_H
3
4 #include <stdio.h>
5
11 void USART_Init(unsigned int ubrr);
12
19 int USART_Transmit(unsigned char data);
20
26 int USART_Receive();
27
28 #endif
```

6.44 D:/Progetti/TTK4155/Node2/Node2/can_interrupt.c File Reference

```
#include "can_interrupt.h"
#include <stdio.h>
#include "sam.h"
#include "printf-stdarg.h"
#include "can_controller.h"
#include "Motor.h"
```

Macros

- #define CAN_INTERRUPT_H
- #define DEBUG_INTERRUPT 0

Functions

- void CAN0_Handler (void)

  CAN0 Interrupt handler for RX, TX and bus error interrupts.

6.44.1 Macro Definition Documentation

6.44.1.1 CAN_INTERRUPT_H

```
#define CAN_INTERRUPT_H
```

6.44.1.2 DEBUG_INTERRUPT

```
#define DEBUG_INTERRUPT 0
```

6.44.2 Function Documentation

6.44.2.1 CAN0_Handler()

```
void CAN0_Handler (
            void  )
```

CAN0 Interrupt handler for RX, TX and bus error interrupts.

Parameters

void	

Return values

6.45 D:/Progetti/TTK4155/Node2/Node2/can_interrupt.h File Reference

```
#include "sam.h"
```

Functions

- void CAN0_Handler (void)

  CAN0 Interrupt handler for RX, TX and bus error interrupts.

Variables

- uint8_t buttons

6.45.1 Function Documentation

6.45.1.1 CAN0_Handler()

```
void CAN0_Handler (
            void  )
```

CAN0 Interrupt handler for RX, TX and bus error interrupts.

Parameters

void	

Return values

6.45.2 Variable Documentation

6.45.2.1 buttons

```
uint8_t buttons
```

6.46 can_interrupt.h

Go to the documentation of this file.

```
1 /*
2  * can_interrupt.h
3  *
4  * Author: Gustav O. Often and Eivind H. Jlsgard
5  *
6  * For use in TTK4155 Embedded and Industrial Computer Systems Design
7  * NTNU - Norwegian University of Science and Technology
```

```
8  *
9  */
10
11
12 #ifndef CAN_INTERRUPT_H_
13 #define CAN_INTERRUPT_H_
14
15 #include "sam.h"
16
17 /*
18     "Buttons" variable:
19     Joystick button « 0
20     Right button « 1
21 */
22
23 uint8_t buttons;
24
25
26 void CAN0_Handler       ( void );
27
28 #endif /* CAN_INTERRUPT_H_ */
```

6.47 D:/Progetti/TTK4155/Node2/Node2/Debug/can_controller.d File Reference

6.48 D:/Progetti/TTK4155/Node2/Node2/Debug/can_interrupt.d File Reference

6.49 D:/Progetti/TTK4155/Node2/Node2/Debug/delay.d File Reference

6.50 D:/Progetti/TTK4155/Node2/Node2/Debug/Device_Startup/startup_sam3xa.d File Reference

6.51 D:/Progetti/TTK4155/Node2/Node2/Debug/Device_Startup/system_sam3xa.d File Reference

6.52 D:/Progetti/TTK4155/Node2/Node2/Debug/motor.d File Reference

6.53 D:/Progetti/TTK4155/Node2/Node2/Debug/PI.d File Reference

6.54 D:/Progetti/TTK4155/Node2/Node2/Debug/printf-stdarg.d File Reference

6.55 D:/Progetti/TTK4155/Node2/Node2/Debug/PWM.d File Reference

6.56 D:/Progetti/TTK4155/Node2/Node2/Debug/solenoid.d File Reference

6.57 D:/Progetti/TTK4155/Node2/Node2/Debug/uart.d File Reference

6.58 D:/Progetti/TTK4155/Node2/Node2/delay.c File Reference

```
#include "delay.h"
```

Functions

- void test_delay_us (int16_t value)
- void delay_us (uint16_t value)

  Delay function.

6.58.1 Function Documentation

6.58.1.1 delay_us()

```
void delay_us (
            uint16_t value )
```

Delay function.

Parameters

value	

6.58.1.2 test_delay_us()

```
void test_delay_us (
            int16_t value )
```

6.59 D:/Progetti/TTK4155/Node2/Node2/delay.h File Reference

```
#include "sam.h"
```

Functions

- void test_delay_us (int16_t value)
- void delay_us (uint16_t value)

  Delay function.

6.59.1 Function Documentation

6.59.1.1 delay_us()

```
void delay_us (
            uint16_t value )
```

Delay function.

Parameters

value	

6.59.1.2 test_delay_us()

```
void test_delay_us (
            int16_t value )
```

6.60 delay.h

Go to the documentation of this file.

```
1 #ifndef DELAY_H
2 #define DELAY_H
3
4 #include "sam.h"
5
6 void test_delay_us(int16_t value);
7
13 void delay_us(uint16_t value);
14
15 #endif
```

6.61 D:/Progetti/TTK4155/Node2/Node2/Device_Startup/startup_sam3xa.c File Reference

```
#include "sam3xa.h"
```

Functions

- void __libc_init_array (void)
- void Dummy_Handler (void)

  Default interrupt handler for unused IRQs.
- void NMI_Handler (void HardFault_Handler void)
- void Reset_Handler (void)

  This is the code that gets called on processor reset. To initialize the device, and call the main() routine.

Variables

- uint32_t _sfixed
- uint32_t _efixed
- uint32_t _etext
- uint32_t _srelocate
- uint32_t _erelocate
- uint32_t _szero
- uint32_t _ezero
- uint32_t _sstack
- uint32_t _estack

6.61.1 Function Documentation

6.61.1.1 __libc_init_array()

```
void __libc_init_array (
            void  )
```

6.61.1.2 Dummy_Handler()

```
void Dummy_Handler (
            void  )
```

Default interrupt handler for unused IRQs.

6.61.1.3 NMI_Handler()

```
void NMI_Handler (
            void HardFault_Handler void )
```

6.61.1.4 Reset_Handler()

```
void Reset_Handler (
            void  )
```

This is the code that gets called on processor reset. To initialize the device, and call the main() routine.

6.61.2 Variable Documentation

6.61.2.1 _efixed

```
uint32_t _efixed  [extern]
```

6.61.2.2 _erelocate

```
uint32_t _erelocate  [extern]
```

6.61.2.3 _estack

```
uint32_t _estack  [extern]
```

6.61.2.4 _etext

```
uint32_t _etext  [extern]
```

6.61.2.5 _ezero

```
uint32_t _ezero  [extern]
```

6.61.2.6 _sfixed

```
uint32_t _sfixed  [extern]
```

6.61.2.7 _srelocate

```
uint32_t _srelocate  [extern]
```

6.61.2.8 _sstack

```
uint32_t _sstack  [extern]
```

6.61.2.9 _szero

```
uint32_t _szero  [extern]
```

6.62 D:/Progetti/TTK4155/Node2/Node2/Device_Startup/system_↩ sam3xa.c File Reference

```
#include "sam3xa.h"
```

Macros

- #define SYS_BOARD_OSCOUNT (CKGR_MOR_MOSCXTST(0x8))
- #define SYS_BOARD_PLLAR (CKGR_PLLAR_ONE | CKGR_PLLAR_MULA(0xdUL) | CKGR_PLLAR_↩ PLLACOUNT(0x3fUL) | CKGR_PLLAR_DIVA(0x1UL))
- #define SYS_BOARD_MCKR (PMC_MCKR_PRES_CLK_2 | PMC_MCKR_CSS_PLLA_CLK)

Functions

- void SystemInit (void)

  Setup the microcontroller system. Initialize the System and update the SystemFrequency variable.
- void SystemCoreClockUpdate (void)
- void system_init_flash (uint32_t dw_clk)

Variables

- uint32_t SystemCoreClock = CHIP_FREQ_MAINCK_RC_4MHZ

6.62.1 Macro Definition Documentation

6.62.1.1 SYS_BOARD_MCKR

```
#define SYS_BOARD_MCKR (PMC_MCKR_PRES_CLK_2 | PMC_MCKR_CSS_PLLA_CLK)
```

6.62.1.2 SYS_BOARD_OSCOUNT

```
#define SYS_BOARD_OSCOUNT (CKGR_MOR_MOSCXTST(0x8))
```

6.62.1.3 SYS_BOARD_PLLAR

```
#define SYS_BOARD_PLLAR (CKGR_PLLAR_ONE | CKGR_PLLAR_MULA(0xdUL) | CKGR_PLLAR_PLLACOUNT(0x3fUL)
| CKGR_PLLAR_DIVA(0x1UL))
```

6.62.2 Function Documentation

6.62.2.1 system_init_flash()

```
void system_init_flash (
            uint32_t dw_clk )
```

Initialize flash.

6.62.2.2 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (
            void  )
```

6.62.2.3 SystemInit()

```
void SystemInit (
            void  )
```

Setup the microcontroller system. Initialize the System and update the SystemFrequency variable.

6.62.3 Variable Documentation

6.62.3.1 SystemCoreClock

```
uint32_t SystemCoreClock = CHIP_FREQ_MAINCK_RC_4MHZ
```

6.63 D:/Progetti/TTK4155/Node2/Node2/Motor.c File Reference

```
#include "motor.h"
```

Functions

- void motor_init ()

  Initialization function for the motor.
- void motor_set_direction (uint8_t direction)

  Set direction of the motor LEFT/RIGHT.
- void motor_set_speed (uint8_t speed)

  Set speed of the motor, normalized to 0-100.
- void motor_set_direction_speed (uint8_t direction, uint16_t speed)

  Set both speed and direction for the motor.
- int16_t motor_encoder_read ()

  Read data from the encoder.
- void motor_set_with_PI (uint16_t desired_value)

  Set the position of the motor with PI.

6.63.1 Function Documentation

6.63.1.1 motor_encoder_read()

```
int16_t motor_encoder_read ( )
```

Read data from the encoder.

Returns

int16_t

6.63.1.2 motor_init()

```
void motor_init ( )
```

Initialization function for the motor.

6.63.1.3 motor_set_direction()

```
void motor_set_direction (
            uint8_t direction )
```

Set direction of the motor LEFT/RIGHT.

Parameters

direction	

6.63.1.4 motor_set_direction_speed()

```
void motor_set_direction_speed (
            uint8_t direction,
            uint16_t speed )
```

Set both speed and direction for the motor.

Parameters

direction	
speed	

6.63.1.5 motor_set_speed()

```
void motor_set_speed (
            uint8_t speed )
```

Set speed of the motor, normalized to 0-100.

Parameters

speed	

6.63.1.6 motor_set_with_PI()

```
void motor_set_with_PI (
            uint16_t desired_value )
```

Set the position of the motor with PI.

Parameters

desired_value	

6.64 D:/Progetti/TTK4155/Node2/Node2/Motor.h File Reference

```
#include "sam.h"
#include "delay.h"
#include "PI.h"
```

Macros

- #define LEFT 0
- #define RIGHT 1
- #define NONE 2

Functions

- void motor_init ()

  Initialization function for the motor.
- void motor_set_direction (uint8_t direction)

  Set direction of the motor LEFT/RIGHT.
- void motor_set_speed (uint8_t speed)

  Set speed of the motor, normalized to 0-100.
- void motor_set_direction_speed (uint8_t direction, uint16_t speed)

  Set both speed and direction for the motor.
- int16_t motor_encoder_read ()

  Read data from the encoder.
- void motor_set_with_PI (uint16_t desired_value)

  Set the position of the motor with PI.
- void motor_center ()

  Center motor.

Variables

- pidData_t pid

6.64.1 Macro Definition Documentation

6.64.1.1 LEFT

```
#define LEFT 0
```

6.64.1.2 NONE

```
#define NONE 2
```

6.64.1.3 RIGHT

```
#define RIGHT 1
```

6.64.2 Function Documentation

6.64.2.1 motor_center()

```
void motor_center ( )
```

Center motor.

6.64.2.2 motor_encoder_read()

```
int16_t motor_encoder_read ( )
```

Read data from the encoder.

Returns

int16_t

6.64.2.3 motor_init()

```
void motor_init ( )
```

Initialization function for the motor.

6.64.2.4 motor_set_direction()

```
void motor_set_direction (
            uint8_t direction )
```

Set direction of the motor LEFT/RIGHT.

Parameters

direction	

6.64.2.5 motor_set_direction_speed()

```
void motor_set_direction_speed (
            uint8_t direction,
            uint16_t speed )
```

Set both speed and direction for the motor.

Parameters

direction	
speed	

6.64.2.6 motor_set_speed()

```
void motor_set_speed (
            uint8_t speed )
```

Set speed of the motor, normalized to 0-100.

Parameters

speed	

6.64.2.7 motor_set_with_PI()

```
void motor_set_with_PI (
            uint16_t desired_value )
```

Set the position of the motor with PI.

Parameters

desired_value	

6.64.3 Variable Documentation

6.64.3.1 pid

```
pidData_t pid
```

6.65 Motor.h

Go to the documentation of this file.

```
1 #ifndef MOTOR_H
2 #define MOTOR_H
3
4 #define LEFT 0
5 #define RIGHT 1
6 #define NONE 2
7
8 #include "sam.h"
9 #include "delay.h"
10 #include "PI.h"
11
12 pidData_t pid;
13
18 void motor_init();
19
25 void motor_set_direction(uint8_t direction);
26
32 void motor_set_speed(uint8_t speed);
33
40 void motor_set_direction_speed(uint8_t direction, uint16_t speed);
41
47 int16_t motor_encoder_read();
48
54 void motor_set_with_PI(uint16_t desired_value);
55
60 void motor_center();
61
62 #endif
```

6.66 D:/Progetti/TTK4155/Node2/Node2/PI.c File Reference

```
#include "PI.h"
```

Functions

- void PI_init (int16_t P_factor, int16_t I_factor, pidData_t *pid_st)

  Initialization function for the PI controller.
- int16_t PI_controller (int16_t set_point, int16_t process_value, pidData_t *pid_st)

  Updates the value with the PI controller.

6.66.1 Function Documentation

6.66.1.1 PI_controller()

```
int16_t PI_controller (
            int16_t set_point,
            int16_t process_value,
            pidData_t * pid_st )
```

Updates the value with the PI controller.

Parameters

set_point	
process_value	
pid_st	

Returns

int16_t

6.66.1.2 PI_init()

```
void PI_init (
            int16_t P_factor,
            int16_t I_factor,
            pidData_t * pid_st )
```

Initialization function for the PI controller.

Parameters

P_factor	
I_factor	
pid_st	

6.67 D:/Progetti/TTK4155/Node2/Node2/PI.h File Reference

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

Data Structures

- struct PID_DATA

Typedefs

- typedef struct PID_DATA pidData_t

Functions

- void PI_init (int16_t P_factor, int16_t I_factor, pidData_t *pid_st)

  Initialization function for the PI controller.
- int16_t PI_controller (int16_t set_point, int16_t process_value, pidData_t *pid_st)

  Updates the value with the PI controller.

6.67.1 Typedef Documentation

6.67.1.1 pidData_t

```
typedef struct PID_DATA pidData_t
```

6.67.2 Function Documentation

6.67.2.1 PI_controller()

```
int16_t PI_controller (
            int16_t set_point,
            int16_t process_value,
            pidData_t * pid_st )
```

Updates the value with the PI controller.

Parameters

set_point	
process_value	
pid_st	

Returns

int16_t

6.67.2.2 PI_init()

```
void PI_init (
            int16_t P_factor,
            int16_t I_factor,
            pidData_t * pid_st )
```

Initialization function for the PI controller.

Parameters

P_factor	
I_factor	
pid_st	

6.68 PI.h

Go to the documentation of this file.

```
1 #ifndef PI_H
2 #define PI_H
3
4 #include <stdint.h>
5
6 typedef struct PID_DATA{
7     // Structure adapted from example code
8     // To calculate Error
9     int16_t error;
10     // Summation of errors, used for integrate calculations
11     int32_t integral;
12     // The Proportional tuning constant, multiplied with SCALING_FACTOR
13     int16_t P_Factor;
14     // The Integral tuning constant, multiplied with SCALING_FACTOR
15     int32_t I_Factor;
16 } pidData_t;
17
25 void PI_init(int16_t P_factor, int16_t I_factor, pidData_t* pid_st);
26
35 int16_t PI_controller(int16_t set_point, int16_t process_value, pidData_t  *pid_st);
36
37 #endif
```

6.69 D:/Progetti/TTK4155/Node2/Node2/printf-stdarg.c File Reference

```
#include <stdarg.h>
#include "uart.h"
```

Macros

- #define PAD_RIGHT 1
- #define PAD_ZERO 2
- #define PRINT_BUF_LEN 12

Functions

- int printf (const char *format,...)
- int sprintf (char *out, const char *format,...)
- int snprintf (char *buf, unsigned int count, const char *format,...)

6.69.1 Macro Definition Documentation

6.69.1.1 PAD_RIGHT

```
#define PAD_RIGHT 1
```

6.69.1.2 PAD_ZERO

```
#define PAD_ZERO 2
```

6.69.1.3 PRINT_BUF_LEN

```
#define PRINT_BUF_LEN 12
```

6.69.2 Function Documentation

6.69.2.1 printf()

```
int printf (
            const char * format,
             ...  )
```

6.69.2.2 snprintf()

```
int snprintf (
            char * buf,
            unsigned int count,
            const char * format,
             ...  )
```

6.69.2.3 sprintf()

```
int sprintf (
            char * out,
            const char * format,
             ...  )
```

6.70 D:/Progetti/TTK4155/Node2/Node2/printf-stdarg.h File Reference

Macros

- #define PRINTF -STDARG_H_

Functions

- int printf (const char *format,...)

6.70.1 Macro Definition Documentation

6.70.1.1 PRINTF

```
#define PRINTF -STDARG_H_
```

6.70.2 Function Documentation

6.70.2.1 printf()

```
int printf (
            const char * format,
             ...  )
```

6.71 printf-stdarg.h

Go to the documentation of this file.

```
1 /*
2     Copyright 2001, 2002 Georges Menie (www.menie.org)
3     stdarg version contributed by Christian Ettinger
4     This program is free software; you can redistribute it and/or modify
5     it under the terms of the GNU Lesser General Public License as published by
6     the Free Software Foundation; either version 2 of the License, or
7     (at your option) any later version.
8     This program is distributed in the hope that it will be useful,
9     but WITHOUT ANY WARRANTY; without even the implied warranty of
10     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
11     GNU Lesser General Public License for more details.
12     You should have received a copy of the GNU Lesser General Public License
13     along with this program; if not, write to the Free Software
14     Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
15 */
16
17
18 #ifndef PRINTF-STDARG_H_
19 #define PRINTF-STDARG_H_
20
21
22 int printf(const char *format, ...);
23
24
25 #endif /* PRINTF-STDARG_H_ */
```

6.72 D:/Progetti/TTK4155/Node2/Node2/PWM.c File Reference

```
#include "PWM.h"
```

Functions

- void PWM_init ()

  Initialization function for the PWM.
- long map (long x, long in_min, long in_max, long out_min, long out_max)

  Function to map a value in a certain range.
- int PWM_set_value (int8_t value)

  Set value to PWM.

6.72.1 Function Documentation

6.72.1.1 map()

```
long map (
            long x,
            long in_min,
            long in_max,
            long out_min,
            long out_max )
```

Function to map a value in a certain range.

Parameters

x	Value to be mapped
in_min	
in_max	
out_min	
out_max	

Returns

long

Function to map a value in a certain range.

Parameters

x	The value to remap
in_min	Input range minumum value
in_max	Input range maximum value
out_min	Output range minumum value
out_max	Output range maximum value

Returns

long

6.72.1.2 PWM_init()

```
void PWM_init ( )
```

Initialization function for the PWM.

6.72.1.3 PWM_set_value()

```
int PWM_set_value (
            int8_t value )
```

Set value to PWM.

Parameters

value	

Returns

int

6.73 D:/Progetti/TTK4155/Node2/Node2/PWM.h File Reference

```
#include "sam.h"
```

Macros

- #define STEP_CENTER 1320
- #define STEP_RIGHT_MAX 1740
- #define STEP_LEFT_MAX 900

Functions

- void PWM_init ()

  Initialization function for the PWM.
- long map (long x, long in_min, long in_max, long out_min, long out_max)

  Function to map a value in a certain range.
- int PWM_set_value (int8_t value)

  Set value to PWM.

6.73.1 Macro Definition Documentation

6.73.1.1 STEP_CENTER

```
#define STEP_CENTER 1320
```

6.73.1.2 STEP_LEFT_MAX

```
#define STEP_LEFT_MAX 900
```

6.73.1.3 STEP_RIGHT_MAX

```
#define STEP_RIGHT_MAX 1740
```

6.73.2 Function Documentation

6.73.2.1 map()

```
long map (
             long x,
             long in_min,
             long in_max,
             long out_min,
             long out_max )
```

Function to map a value in a certain range.

Parameters

x	Value to be mapped
in_min	
in_max	
out_min	
out_max	

Returns

long

Function to map a value in a certain range.

Parameters

x	The value to remap
in_min	Input range minumum value
in_max	Input range maximum value
out_min	Output range minumum value
out_max	Output range maximum value

Returns

long

6.73.2.2 PWM_init()

```
void PWM_init ( )
```

Initialization function for the PWM.

6.73.2.3 PWM_set_value()

```
int PWM_set_value (
            int8_t value )
```

Set value to PWM.

Parameters

value	

Returns

int

6.74 PWM.h

Go to the documentation of this file.

```
1 #ifndef PWM_H
2 #define PWM_H
3
4 #include "sam.h"
5
6 #define STEP_CENTER 1320
7 #define STEP_RIGHT_MAX 1740
8 #define STEP_LEFT_MAX 900
```

```
9
14 void PWM_init();
15
26 long map(long x, long in_min, long in_max, long out_min, long out_max);
27
34 int PWM_set_value(int8_t value);
35
36 #endif PWM_H
```

6.75 D:/Progetti/TTK4155/Node2/Node2/solenoid.c File Reference

```
#include "solenoid.h"
```

Functions

- void solenoid_init ()

  Initalization function for the solenoid.
- void solenoid_routine (uint8_t pressed)

  Send impulse to the solenoid.

6.75.1 Function Documentation

6.75.1.1 solenoid_init()

```
void solenoid_init ( )
```

Initalization function for the solenoid.

6.75.1.2 solenoid_routine()

```
void solenoid_routine (
            uint8_t in_state )
```

Send impulse to the solenoid.

Parameters

in_state	

6.76 D:/Progetti/TTK4155/Node2/Node2/solenoid.h File Reference

```
#include "sam.h"
```

Functions

- void solenoid_init ()

  Initalization function for the solenoid.
- void solenoid_routine (uint8_t in_state)

  Send impulse to the solenoid.

6.76.1 Function Documentation

6.76.1.1 solenoid_init()

```
void solenoid_init ( )
```

Initalization function for the solenoid.

6.76.1.2 solenoid_routine()

```
void solenoid_routine (
            uint8_t in_state )
```

Send impulse to the solenoid.

Parameters

in_state	

6.77 solenoid.h

Go to the documentation of this file.

```
1 #ifndef SOLENOID_H_
2 #define SOLENOID_H_
3
4 #include "sam.h"
5
10 void solenoid_init();
11
17 void solenoid_routine(uint8_t in_state);
18
19 #endif
```

6.78 D:/Progetti/TTK4155/Node2/Node2/uart.c File Reference

```
#include <stdint.h>
#include "sam.h"
#include "uart.h"
```

Functions

- void configure_uart (void)

  Configure UART.
- int uart_getchar (uint8_t *c)

  Get character from UART.
- int uart_putchar (const uint8_t c)
- void UART_Handler (void)

Variables

- uart_ringbuffer rx_buffer

6.78.1 Function Documentation

6.78.1.1 configure_uart()

```
void configure_uart (
            void  )
```

Configure UART.

Parameters

void	

Return values

void.	

6.78.1.2 uart_getchar()

```
int uart_getchar (
            uint8_t * c )
```

Get character from UART.

Parameters

∗c	location of character

Return values

Success(0)	or failure(1)

6.78.1.3 UART_Handler()

```
void UART_Handler (
             void  )
```

6.78.1.4 uart_putchar()

```
int uart_putchar (
             const uint8_t c )
```

6.78.2 Variable Documentation

6.78.2.1 rx_buffer

```
uart_ringbuffer rx_buffer
```

6.79 D:/Progetti/TTK4155/Node2/Node2/uart.h File Reference

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

Data Structures

- struct uart_ringbuffer_t

Macros

- #define UART_RINGBUFFER_SIZE 64

Typedefs

- typedef struct uart_ringbuffer_t uart_ringbuffer

Functions

- void configure_uart (void)

  Configure UART.
- int uart_getchar (uint8_t *c)

  Get character from UART.
- int uart_putchar (const uint8_t c)
- void UART_Handler (void)

6.79.1 Macro Definition Documentation

6.79.1.1 UART_RINGBUFFER_SIZE

```
#define UART_RINGBUFFER_SIZE 64
```

6.79.2 Typedef Documentation

6.79.2.1 uart_ringbuffer

```
typedef struct uart_ringbuffer_t uart_ringbuffer
```

6.79.3 Function Documentation

6.79.3.1 configure_uart()

```
void configure_uart (
            void  )
```

Configure UART.

Parameters

void	

Return values

void.	

6.79.3.2 uart_getchar()

```
int uart_getchar (
            uint8_t * c )
```

Get character from UART.

Parameters

*c	location of character

Return values

Success(0)	or failure(1)

6.79.3.3 UART_Handler()

```
void UART_Handler (
            void  )
```

6.79.3.4 uart_putchar()

```
int uart_putchar (
            const uint8_t c )
```

6.80 uart.h

Go to the documentation of this file.

```
1 /*
2  * uart.h
3  *
4  * Author: Gustav O. Often and Eivind H. Jlsgard
5  *
6  * For use in TTK4155 Embedded and Industrial Computer Systems Design
7  * NTNU - Norwegian University of Science and Technology
8  *
9  * A simple interface for receiving and transmitting characters to a computer using UART via the on board
       USB-connector
10  */
11
12
```

```
13 #ifndef UART_H_
14 #define UART_H_
15
16 #include <stdint.h>
17 #define UART_RINGBUFFER_SIZE 64
18 /*
19  * Ringbuffer for receiving characters from
20  */
21 typedef struct uart_ringbuffer_t
22 {
23     uint8_t head, tail;
24     char data[UART_RINGBUFFER_SIZE];
25     } uart_ringbuffer;
26
27
28 void configure_uart(void);
29
30 int uart_getchar(uint8_t *c);
31 int uart_putchar(const uint8_t c);
32
33 void UART_Handler       ( void );
34
35
36
37 #endif /* UART_H_ */
```

6.81 D:/Progetti/TTK4155/Node3/mPWM.h File Reference

Macros

- #define mPWM_H

Functions

- void mPWM_init ()

6.81.1 Macro Definition Documentation

6.81.1.1 mPWM_H

```
#define mPWM_H
```

6.81.2 Function Documentation

6.81.2.1 mPWM_init()

```
void mPWM_init ( )
```

6.82 mPWM.h

Go to the documentation of this file.

```
1 #ifndef mPWM
2 #define mPWM_H
3
4
5
6 void mPWM_init()
7 {
8     REG_PMC_PCER1 |= PMC_PCER1_PID36;                    // Enable PWM
9     REG_PIOC_ABSR |= PIO_ABSR_P20;                       // Set PWM pin 20 peripheral C
10     REG_PIOC_PDR |= PIO_PDR_P20;                         // Set PWM pin 20 to an output
11     REG_PWM_CLK = PWM_CLK_PREA(0) | PWM_CLK_DIVA(42);    // Set the PWM clock rate to 2MHz (84MHz/42)
12     REG_PWM_CMR5 = PWM_CMR_CALG | PWM_CMR_CPRE_CLKA;     // Enable dual slope PWM and set the clock
       source as CLKA
13     REG_PWM_CPRD5 = 20000;                               // Set the PWM frequency 2MHz/(2 * 20000) =
       50Hz = 20ms
14     REG_PWM_CDTY5 = 20000-STEP_CENTER;                   // Set the PWM duty cycle to 1500 - centre the
       servo
15     REG_PWM_ENA = PWM_ENA_CHID5;                         // Enable the PWM channel
16 }
17
18
```

6.83 D:/Progetti/TTK4155/Node3/music.h File Reference

Variables

- const int star_wars_theme[ ] PROGMEM

6.83.1 Variable Documentation

6.83.1.1 PROGMEM

```
const int cannon_in_d_pachelbel [ ] PROGMEM
```

Initial value:

```
= {
    NOTE_AS4,8,     NOTE_AS4,       NOTE_AS4,
    NOTE_F5,2,       NOTE_C6,
    NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
    NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
    NOTE_AS5,8,     NOTE_A5,     NOTE_AS5,       NOTE_G5,2,       NOTE_C5,8,       NOTE_C5, NOTE_C5,
    NOTE_F5,2,       NOTE_C6,
    NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
    NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
    NOTE_AS5,8,     NOTE_A5,     NOTE_AS5,       NOTE_G5,2,       NOTE_C5,-8,      NOTE_C5,16,
    NOTE_D5,-4,      NOTE_D5,8,       NOTE_AS5,       NOTE_A5,     NOTE_G5,     NOTE_F5,
    NOTE_F5,     NOTE_G5,     NOTE_A5,     NOTE_G5,4,       NOTE_D5,8,       NOTE_E5,4,   NOTE_C5,-8,
       NOTE_C5,16,
    NOTE_D5,-4,      NOTE_D5,8,       NOTE_AS5,       NOTE_A5,     NOTE_G5,     NOTE_F5,
    NOTE_C6,-8,      NOTE_G5,16,      NOTE_G5,2,       REST,8,     NOTE_C5,
    NOTE_D5,-4,      NOTE_D5,8,       NOTE_AS5,       NOTE_A5, NOTE_G5,         NOTE_F5,
    NOTE_F5,     NOTE_G5,     NOTE_A5,     NOTE_G5,4,   NOTE_D5,8,           NOTE_E5,4,   NOTE_C6,-8,
       NOTE_C6,16,
    NOTE_F6,4,       NOTE_DS6,8,     NOTE_CS6,4,     NOTE_C6,8,   NOTE_AS5,4,         NOTE_GS5,8, NOTE_G5,4,
             NOTE_F5,8,
    NOTE_C6,1,
    MUSIC_END
}
```

6.84 music.h

Go to the documentation of this file.

```
1
2 #ifndef NOTES
3 #define NOTES
4
5 /*************************************************************
6     CHIP TUNES
7 *************************************************************/
8
9 // Notes of the melody followed by the duration.
10 // A 4 means a quarter note, 8 an eighteenth, 16 sixteenth, so on.
11 // Negative numbers are used to represent dotted notes,
12 // so -4 means a dotted quarter note, that is, a quarter plus an eighteenth
13
14 /******************** CONTENTS TABLE ************************
15     1. MOVIES SECTION
16
17     1.1     Dart Vader theme (Imperial March) - Star wars
18
19
20     2. GAMES SECTION
21
22     2.1     Tetris theme - (Korobeiniki)
23     2.2     Mario Main Theme
24     2.3     Mario Underworld Melody
25
26
27     3. CLASSIC SECTION
28
29     3.1     Fur Elise - Ludwig van Beethovem
30     3.2     Cannon in D - Pachelbel
31
32 ******************** END OF CONTENTS TABLE ********************/
33 /*
34 #define CHIPTUNES_INCLUDE_1_1       1
35
36 #define CHIPTUNES_INCLUDE_2_1       1
37 #define CHIPTUNES_INCLUDE_2_2       1
38 #define CHIPTUNES_INCLUDE_2_3       1
39
40 #define CHIPTUNES_INCLUDE_3_1       1
41 #define CHIPTUNES_INCLUDE_3_2       1
42 #define CHIPTUNES_INCLUDE_3_3       1
43 #define CHIPTUNES_INCLUDE_3_4       1
44
45 #define CHIPTUNES_INCLUDE_4_1       1
46 */
47
48
       /*************************************************************************************************************************
49     MOVIES
50
       *************************************************************************************************************************
51
52
       /*-------------------------------------------------------------------------------------------------------------
53 *   Title:      Dart Vader theme (Imperial March) - Star wars
54 *
55 *   Tempo:      108
56 *
57 *   Source:     https://github.com/robsoncouto/arduino-songs/blob/master/starwars/starwars.ino
58 *
59 *   Score available at https://musescore.com/user/202909/scores/1141521
60 *   The tenor saxophone part was used
61 *
62
       *-------------------------------------------------------------------------------------------------------------*/
63
64 //#if CHIPTUNES_INCLUDE_1_1 == 1
65
66 const int star_wars_theme[] PROGMEM = {
67     NOTE_AS4,8,     NOTE_AS4,       NOTE_AS4,
       //1
68     NOTE_F5,2,       NOTE_C6,
69     NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
70     NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
71     NOTE_AS5,8,     NOTE_A5,     NOTE_AS5,       NOTE_G5,2,       NOTE_C5,8,       NOTE_C5, NOTE_C5,
72     NOTE_F5,2,       NOTE_C6,
73     NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,
74
75     NOTE_AS5,8,     NOTE_A5,     NOTE_G5,     NOTE_F6,2,       NOTE_C6,4,                          //8
76     NOTE_AS5,8,     NOTE_A5,     NOTE_AS5,       NOTE_G5,2,       NOTE_C5,-8,      NOTE_C5,16,
77     NOTE_D5,-4,      NOTE_D5,8,       NOTE_AS5,       NOTE_A5,     NOTE_G5,     NOTE_F5,
```

```
78      NOTE_F5,      NOTE_G5,      NOTE_A5,      NOTE_G5,4,        NOTE_D5,8,        NOTE_E5,4,    NOTE_C5,-8,
         NOTE_C5,16,
79      NOTE_D5,-4,       NOTE_D5,8,        NOTE_AS5,        NOTE_A5,      NOTE_G5,      NOTE_F5,
80
81      NOTE_C6,-8,       NOTE_G5,16,       NOTE_G5,2,        REST,8,      NOTE_C5,
        //13
82      NOTE_D5,-4,       NOTE_D5,8,        NOTE_AS5,        NOTE_A5, NOTE_G5,          NOTE_F5,
83      NOTE_F5,      NOTE_G5,      NOTE_A5,      NOTE_G5,4,    NOTE_D5,8,            NOTE_E5,4,    NOTE_C6,-8,
         NOTE_C6,16,
84      NOTE_F6,4,        NOTE_DS6,8,      NOTE_CS6,4,      NOTE_C6,8,    NOTE_AS5,4,          NOTE_GS5,8,
        NOTE_G5,4,        NOTE_F5,8,
85      NOTE_C6,1,
86      MUSIC_END
87  };
88
89  //#endif
90
91
92
        /*****************************************************************************************************************************
93      GAMES
94
        *****************************************************************************************************************************
95
96
97
        /*-------------------------------------------------------------------------------------------------------------
98  *   Title:      Tetris theme - (Korobeiniki)
99  *
100 *    Tempo:      144
101 *
102 *    Source:     https://dragaosemchama.com/en/2019/02/songs-for-arduino/
103 *
104 *    Based on the arrangement at https://www.flutetunes.com/tunes.php?id=192
105 *
106
        *-------------------------------------------------------------------------------------------------------------*/
107
108 //#if CHIPTUNES_INCLUDE_2_1 == 1
109
110 const int tetris_theme[] PROGMEM = {
111     NOTE_E5, 4,       NOTE_B4, 8,       NOTE_C5,          NOTE_D5, 4,       NOTE_C5, 8,       NOTE_B4,
112     NOTE_A4, 4,       NOTE_A4, 8,       NOTE_C5,          NOTE_E5, 4,       NOTE_D5, 8,       NOTE_C5,
113     NOTE_B4, -4,      NOTE_C5, 8,       NOTE_D5, 4,       NOTE_E5,
114     NOTE_C5,          NOTE_A4,      NOTE_A4, 8,       NOTE_A4, 4,       NOTE_B4, 8,       NOTE_C5,
115
116     NOTE_D5, -4, NOTE_F5, 8,       NOTE_A5, 4,       NOTE_G5, 8,       NOTE_F5,
117     NOTE_E5, -4, NOTE_C5, 8,       NOTE_E5, 4,       NOTE_D5, 8,       NOTE_C5,
118     NOTE_B4, 4,       NOTE_B4, 8,       NOTE_C5,      NOTE_D5, 4,       NOTE_E5,
119     NOTE_C5,      NOTE_A4,      NOTE_A4,      REST,
120
121     NOTE_E5,      NOTE_B4, 8,       NOTE_C5,      NOTE_D5, 4,       NOTE_C5, 8,       NOTE_B4,
122     NOTE_A4, 4,       NOTE_A4, 8,       NOTE_C5,      NOTE_E5, 4,       NOTE_D5, 8,       NOTE_C5,
123     NOTE_B4, -4, NOTE_C5, 8,       NOTE_D5, 4,       NOTE_E5,
124     NOTE_C5,      NOTE_A4,      NOTE_A4, 8,       NOTE_A4, 4,       NOTE_B4, 8,       NOTE_C5,
125
126     NOTE_D5, -4, NOTE_F5, 8,       NOTE_A5, 4,       NOTE_G5, 8,       NOTE_F5,
127     NOTE_E5, -4, NOTE_C5, 8,       NOTE_E5, 4,       NOTE_D5, 8,       NOTE_C5,
128     NOTE_B4, 4,       NOTE_B4, 8,       NOTE_C5,      NOTE_D5, 4,       NOTE_E5,
129     NOTE_C5,      NOTE_A4,      NOTE_A4,      REST,
130
131     NOTE_E5, 2,       NOTE_C5,
132     NOTE_D5,      NOTE_B4,
133     NOTE_C5,      NOTE_A4,
134     NOTE_GS4,         NOTE_B4, 4,       REST, 8,
135     NOTE_E5, 2,       NOTE_C5,
136     NOTE_D5,      NOTE_B4,
137     NOTE_C5, 4,       NOTE_E5,      NOTE_A5, 2,
138     NOTE_GS5,
139     MUSIC_END
140 };
141
142 //#endif
143
144
145
        /*-------------------------------------------------------------------------------------------------------------
146 *    Title:      Mario Main Theme
147 *
148 *    Tempo:      120
149 *
150 *    Source:     http://aquaticus.info/pwm-music
151 *
152
        *-------------------------------------------------------------------------------------------------------------*/
153
154 //#if CHIPTUNES_INCLUDE_2_2 == 1
```

```
155
156 const int mario_main_theme[] PROGMEM = {
157     NOTE_E7,12,      NOTE_E7, REST,       NOTE_E7, REST,       NOTE_C7, NOTE_E7, REST,
158     NOTE_G7,     REST,       REST,       REST,       NOTE_G6, REST,       REST,       REST,
159     NOTE_C7,     REST,       REST,       NOTE_G6, REST,       REST,       NOTE_E6, REST,
160     REST,           NOTE_A6, REST,       NOTE_B6, REST,       NOTE_AS6,   NOTE_A6, REST,
161     NOTE_G6,9,       NOTE_E7, NOTE_G7, NOTE_A7,12,  REST,       NOTE_F7, NOTE_G7, REST,
162     NOTE_E7,     REST,       NOTE_C7, NOTE_D7, NOTE_B6, REST,       REST,       NOTE_C7,
163     REST,           REST,       NOTE_G6, REST,       REST,       NOTE_E6, REST,       REST,
164     NOTE_A6,     REST,       NOTE_B6, REST,       NOTE_AS6,   NOTE_A6, REST,       NOTE_G6,9,
165     NOTE_E7,     NOTE_G7, NOTE_A7,12,  REST,       NOTE_F7, NOTE_G7, REST,       NOTE_E7,
166     REST,           NOTE_C7, NOTE_D7, NOTE_B6, REST,       REST,       NOTE_C7,
167     MUSIC_END
168 };
169
170 //#endif
171
172
173
174
    /*-----------------------------------------------------------------------------------------------------------
175 *   Title:      Mario Underworld Melody
176 *
177 *   Tempo:      120
178 *
179 *   Source:     http://aquaticus.info/pwm-music
180 *
181
    *-----------------------------------------------------------------------------------------------------------*/
182
183 //#if CHIPTUNES_INCLUDE_2_3 == 1
184
185 const int mario_underworld_melody[] PROGMEM = {
186     NOTE_C4,12,      NOTE_C5,     NOTE_A3,     NOTE_A4,     NOTE_AS3,       NOTE_AS4,       REST,6,
    REST,3,
187     NOTE_C4,12,      NOTE_C5,     NOTE_A3,     NOTE_A4,     NOTE_AS3,       NOTE_AS4,       REST,6,
    REST,3,
188     NOTE_F3,12,      NOTE_F4,     NOTE_D3,     NOTE_D4,     NOTE_D3,     NOTE_DS4,       REST,6,
    REST,3,
189     NOTE_F3,12,      NOTE_F4,     NOTE_D3,     NOTE_D4,     NOTE_DS3,       NOTE_DS4,       REST,6,
    REST,
190     NOTE_DS4,18,     NOTE_CS4,       NOTE_D4,     NOTE_CS4,6,     NOTE_DS4,       NOTE_DS4,
    NOTE_GS3,   NOTE_G3,
191     NOTE_CS4,       NOTE_C4,18,      NOTE_FS4,       NOTE_F4,     NOTE_E3,     NOTE_AS4,       NOTE_A4,
    NOTE_GS4,10,
192     NOTE_DS4,       NOTE_B3,     NOTE_AS3,       NOTE_A3,     NOTE_GS3,       REST,3,         REST,
    REST,
193     MUSIC_END
194 };
195
196 //#endif
197
198
199
200
    /*************************************************************************************************************
201     CLASSIC
202
    *************************************************************************************************************
203
204
    /*-----------------------------------------------------------------------------------------------------------
205 *   Title:      Fur Elise - Ludwig van Beethovem
206 *
207 *   Tempo:      80
208 *
209 *   Source:     https://github.com/robsoncouto/arduino-songs/blob/master/furelise/furelise.ino
210 *
211 *   Score available at https://musescore.com/user/28149610/scores/5281944
212 *
213
    *-----------------------------------------------------------------------------------------------------------*/
214
215 //#if CHIPTUNES_INCLUDE_3_1 == 1
216
217 const int fur_elise[] PROGMEM = {
218     // starts from 1 ending on 9
219     __CHIP_TUNES_START_MARKER__,
220     NOTE_E5, 16, NOTE_DS5,                                                  //1
221     NOTE_E5,     NOTE_DS5,       NOTE_E5, NOTE_B4, NOTE_D5, NOTE_C5,
222     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4, NOTE_A4,
223     NOTE_B4, -8, NOTE_E4, 16, NOTE_GS4,   NOTE_B4,
224     NOTE_C5, 8,      REST, 16,       NOTE_E4, NOTE_E5, NOTE_DS5,
225
226     NOTE_E5,     NOTE_DS5,       NOTE_E5, NOTE_B4, NOTE_D5, NOTE_C5, //6
227     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4, NOTE_A4,
228     NOTE_B4, -8, NOTE_E4, 16, NOTE_C5, NOTE_B4,
```

```
229     __CHIP_TUNES_END_MARKER__,
230     NOTE_A4 , 4, REST, 8,                                                   //9 - 1st ending
231
232     // repeats from 1 ending on 10
233     __CHIP_TUNES_GOTO_MARKER__,
234     NOTE_A4, 8,       REST, 16,       NOTE_B4, NOTE_C5, NOTE_D5,            //10 - 2nd ending
235
236     // continues from 11
237     __CHIP_TUNES_START_MARKER__,
238     NOTE_E5, -8, NOTE_G4, 16, NOTE_F5, NOTE_E5,
239     NOTE_D5, -8, NOTE_F4, 16, NOTE_E5, NOTE_D5,                           //12
240
241     NOTE_C5, -8, NOTE_E4, 16, NOTE_D5, NOTE_C5,                           //13
242     NOTE_B4, 8,       REST, 16,       NOTE_E4, NOTE_E5, REST,
243     REST,           NOTE_E5,     NOTE_E6, REST,       REST,       NOTE_DS5,
244     NOTE_E5,      REST,           REST,       NOTE_DS5,   NOTE_E5, NOTE_DS5,
245     NOTE_E5,      NOTE_DS5,       NOTE_E5, NOTE_B4, NOTE_D5, NOTE_C5,
246     NOTE_A4, 8,       REST, 16,       NOTE_C4, NOTE_E4, NOTE_A4,
247
248     NOTE_B4, 8,       REST, 16,       NOTE_E4, NOTE_GS4,   NOTE_B4,         //19
249     NOTE_C5, 8,       REST, 16,       NOTE_E4, NOTE_E5, NOTE_DS5,
250     NOTE_E5,      NOTE_DS5,       NOTE_E5, NOTE_B4, NOTE_D5, NOTE_C5,
251     NOTE_A4, 8,       REST, 16,       NOTE_C4, NOTE_E4, NOTE_A4,
252     NOTE_B4, 8,       REST, 16,       NOTE_E4, NOTE_C5, NOTE_B4,
253     __CHIP_TUNES_END_MARKER__,
254     NOTE_A4, 8,       REST, 16,       NOTE_B4, NOTE_C5, NOTE_D5,            //24 (1st ending)
255
256     // repeats from 11
257     __CHIP_TUNES_GOTO_MARKER__,
258     NOTE_A4, 8,       REST, 16,       NOTE_C5, NOTE_C5, NOTE_C5,            //25 - 2nd ending
259
260     // continues from 26
261     NOTE_C5, 4,       NOTE_F5, -16,    NOTE_E5, 32,                         //26
262     NOTE_E5, 8,       NOTE_D5,     NOTE_AS5, -16,  NOTE_A5, 32,
263     NOTE_A5, 16, NOTE_G5,     NOTE_F5,     NOTE_E5,     NOTE_D5,     NOTE_C5,
264     NOTE_AS4, 8,    NOTE_A4,     NOTE_A4, 32, NOTE_G4,     NOTE_A4,     NOTE_B4,
265     NOTE_C5, 4,       NOTE_D5, 16, NOTE_DS5,
266     NOTE_E5, -8, NOTE_E5, 16, NOTE_F5,     NOTE_A4,
267     NOTE_C5, 4,       NOTE_D5, -16,    NOTE_B4, 32,
268
269
270     NOTE_C5,     NOTE_G5,     NOTE_G4,     NOTE_G5, NOTE_A4, NOTE_G5, NOTE_B4, NOTE_G5, NOTE_C5,
        NOTE_G5, NOTE_D5, NOTE_G5,     //33
271     NOTE_E5,     NOTE_G5,     NOTE_C6,     NOTE_B5, NOTE_A5, NOTE_G5, NOTE_F5, NOTE_E5, NOTE_D5,
        NOTE_G5, NOTE_F5, NOTE_D5,
272     NOTE_C5,     NOTE_G5,     NOTE_G4,     NOTE_G5, NOTE_A4, NOTE_G5, NOTE_B4, NOTE_G5, NOTE_C5,
        NOTE_G5, NOTE_D5, NOTE_G5,
273
274     NOTE_E5,     NOTE_G5,     NOTE_C6,     NOTE_B5, NOTE_A5, NOTE_G5, NOTE_F5, NOTE_E5, NOTE_D5,
        NOTE_G5, NOTE_F5, NOTE_D5,     //36
275     NOTE_E5,     NOTE_F5,     NOTE_E5,     NOTE_DS5,   NOTE_E5, NOTE_B4, NOTE_E5, NOTE_DS5,   NOTE_E5,
        NOTE_B4, NOTE_E5, NOTE_DS5,
276     NOTE_E5, -8, NOTE_B4, 16, NOTE_E5,     NOTE_DS5,
277     NOTE_E5, -8, NOTE_B4, 16, NOTE_E5,     REST,
278
279     REST,           NOTE_DS5,       NOTE_E5,     REST,       REST,       NOTE_DS5,       //40
280     NOTE_E5,      NOTE_DS5,       NOTE_E5,     NOTE_B4, NOTE_D5, NOTE_C5,
281     NOTE_A4, 8,       REST, 16,       NOTE_C4,     NOTE_E4, NOTE_A4,
282     NOTE_B4, 8,       REST, 16,       NOTE_E4,     NOTE_GS4,   NOTE_B4,
283     NOTE_C5, 8,       REST, 16,       NOTE_E4,     NOTE_E5, NOTE_DS5,
284     NOTE_E5,      NOTE_DS5,       NOTE_E5,     NOTE_B4, NOTE_D5, NOTE_C5,
285
286     NOTE_A4, 8,       REST, 16,       NOTE_C4,     NOTE_E4, NOTE_A4,           //46
287     NOTE_B4, 8,       REST, 16,       NOTE_E4,     NOTE_C5, NOTE_B4,
288     NOTE_A4, 8,       REST, 16,       NOTE_B4,     NOTE_C5, NOTE_D5,
289     NOTE_E5, -8, NOTE_G4, 16, NOTE_F5,     NOTE_E5,
290     NOTE_D5, -8, NOTE_F4, 16, NOTE_E5,     NOTE_D5,
291     NOTE_C5, -8, NOTE_E4, 16, NOTE_D5,     NOTE_C5,
292     NOTE_B4, 8,       REST, 16,       NOTE_E4,     NOTE_E5, REST,
293     REST,           NOTE_E5,     NOTE_E6,     REST,       REST,       NOTE_DS5,
294
295     NOTE_E5,      REST,           REST,           NOTE_DS5,   NOTE_E5, NOTE_D5,     //54
296     NOTE_E5,      NOTE_DS5,       NOTE_E5,     NOTE_B4, NOTE_D5, NOTE_C5,
297     NOTE_A4, 8,       REST, 16,       NOTE_C4,     NOTE_E4, NOTE_A4,
298     NOTE_B4, 8,       REST, 16,       NOTE_E4,     NOTE_GS4,   NOTE_B4,
299     NOTE_C5, 8,       REST, 16,       NOTE_E4,     NOTE_E5, NOTE_DS5,
300     NOTE_E5,      NOTE_DS5,       NOTE_E5,     NOTE_B4, NOTE_D5, NOTE_C5,
301
302     NOTE_A4, 8,       REST, 16,       NOTE_C4,     NOTE_E4, NOTE_A4,           //60
303     NOTE_B4, 8,       REST, 16,       NOTE_E4,     NOTE_C5, NOTE_B4,
304     NOTE_A4, 8,       REST, 16,       REST,           REST, 8,    NOTE_CS5, -4,
305     NOTE_D5, 4,       NOTE_E5, 16, NOTE_F5,
306     NOTE_F5, 4,       NOTE_F5, 8,  NOTE_E5, -4,
307     NOTE_D5, 4,       NOTE_C5, 16, NOTE_B4,
308     NOTE_A4, 4,       NOTE_A4, 8,       NOTE_A4,     NOTE_C5,     NOTE_B4,
309     NOTE_A4, -4, NOTE_CS5, -4,
310
```

```
311     NOTE_D5, 4,       NOTE_E5, 16, NOTE_F5,                                          //72
312     NOTE_F5, 4,       NOTE_F5, 8,       NOTE_F5, -4,
313     NOTE_DS5, 4,     NOTE_D5, 16, NOTE_C5,
314     NOTE_AS4, 4,     NOTE_A4, 8,
315     NOTE_GS4, 4,     NOTE_G4, 8,       NOTE_A4, -4,
316     NOTE_B4, 4,       REST, 8,
317     NOTE_A3, -32,     NOTE_C4,      NOTE_E4,      NOTE_A4, NOTE_C5, NOTE_E5, NOTE_D5, NOTE_C5, NOTE_B4,
318
319     NOTE_A4,      NOTE_C5,      NOTE_E5,      NOTE_A5, NOTE_C6, NOTE_E6, NOTE_D6,      NOTE_C6, NOTE_B5,
    //80
320     NOTE_A4,      NOTE_C5,      NOTE_E5,      NOTE_A5, NOTE_C6, NOTE_E6, NOTE_D6, NOTE_C6, NOTE_B5,
321     NOTE_AS5,        NOTE_A5,      NOTE_GS5,        NOTE_G5, NOTE_FS5,    NOTE_F5, NOTE_E5, NOTE_DS5,
    NOTE_D5,
322
323     NOTE_CS5,        NOTE_C5,      NOTE_B4,      NOTE_AS4,    NOTE_A4, NOTE_GS4,    NOTE_G4, NOTE_FS4,
    NOTE_F4,      //84
324     NOTE_E4, 16, NOTE_DS5,        NOTE_E5,      NOTE_B4, NOTE_D5, NOTE_C5,
325     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4,      NOTE_A4,
326     NOTE_B4, -8, NOTE_E4, 16, NOTE_GS4,        NOTE_B4,
327
328     NOTE_C5, 8,       REST, 16,        NOTE_E4,      NOTE_E5, NOTE_DS5,
                        //88
329     NOTE_E5,      NOTE_DS5,        NOTE_E5,      NOTE_B4, NOTE_D5, NOTE_C5,
330     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4,      NOTE_A4,
331     NOTE_B4, -8, NOTE_E4, 16, NOTE_C5,      NOTE_B4,
332     NOTE_A4, -8, REST,            REST,            NOTE_G4, 16, NOTE_F5,      NOTE_E5,
333     NOTE_D5, 4,       REST, 8,         REST, -8,        NOTE_E4, 16, NOTE_D5,      NOTE_C5,
334
335     NOTE_B4, -8, NOTE_E4, 16, NOTE_E5, 8,
                        //96
336     NOTE_E5,      NOTE_E6, -8, NOTE_DS5, 16,    NOTE_E5, REST,        REST,    NOTE_DS5,    NOTE_E5,
    NOTE_DS5,
337     NOTE_E5,      NOTE_DS5,        NOTE_E5,      NOTE_B4, NOTE_D5, NOTE_C5,
338     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4,      NOTE_A4,
339     NOTE_B4, -8, NOTE_E4, 16, NOTE_GS4,        NOTE_B4,
340
341     NOTE_C5, 8,       REST, 16,        NOTE_E4,      NOTE_E5, NOTE_DS5,
                        //102
342     NOTE_E5,      NOTE_DS5,        NOTE_E5,      NOTE_B4, NOTE_D5, NOTE_C5,
343     NOTE_A4, -8, NOTE_C4, 16, NOTE_E4,      NOTE_A4,
344     NOTE_B4, -8, NOTE_E4, 16, NOTE_C5,      NOTE_B4, NOTE_A4, -4,
345     MUSIC_END
346 };
347
348 //#endif
349
350
351
    /*-----------------------------------------------------------------------------------------------------------------
352 *   Title:      Cannon in D - Pachelbel
353 *
354 *   Tempo:      100
355 *
356 *   Source:     https://github.com/robsoncouto/arduino-songs/blob/master/cannonind/cannonind.ino
357 *
358 *   Score available at https://musescore.com/user/4710311/scores/1975521
359 *
360
    *-----------------------------------------------------------------------------------------------------------------*/
361
362 //#if CHIPTUNES_INCLUDE_3_2 == 1
363
364 const int cannon_in_d_pachelbel[] PROGMEM = {
365     // C F
366     NOTE_FS4,2,      NOTE_E4,      NOTE_D4,      NOTE_CS4,
367     NOTE_B3,      NOTE_A3,      NOTE_B3,      NOTE_CS4,
368     NOTE_FS4,        NOTE_E4,      NOTE_D4,      NOTE_CS4,
369     NOTE_B3,      NOTE_A3,      NOTE_B3,      NOTE_CS4,
370     NOTE_D4,      NOTE_CS4,        NOTE_B3,      NOTE_A3,
371     NOTE_G3,      NOTE_FS3,        NOTE_G3,      NOTE_A3,
372
373     NOTE_D4,4,        NOTE_FS4,8,      NOTE_G4,      NOTE_A4,4,        NOTE_FS4,8,      NOTE_G4,
374     NOTE_A4,4,        NOTE_B3,8,        NOTE_CS4,        NOTE_D4,      NOTE_E4,      NOTE_FS4,        NOTE_G4,
375     NOTE_FS4,4,      NOTE_D4,8,        NOTE_E4,      NOTE_FS4,4,      NOTE_FS3,8,      NOTE_G3,
376     NOTE_A3,      NOTE_G3,      NOTE_FS3,        NOTE_G3,      NOTE_A3,2,
377     NOTE_G3,4,        NOTE_B3,8,        NOTE_A3,      NOTE_G3,4,        NOTE_FS3,8,      NOTE_E3,
378     NOTE_FS3,4,      NOTE_D3,8,        NOTE_E3,      NOTE_FS3,        NOTE_G3,      NOTE_A3,      NOTE_B3,
379
380     NOTE_G3,4,        NOTE_B3,8,        NOTE_A3,      NOTE_B3,4,        NOTE_CS4,8,      NOTE_D4,
381     NOTE_A3,      NOTE_B3,      NOTE_CS4,        NOTE_D4,      NOTE_E4,      NOTE_FS4,        NOTE_G4,
    NOTE_A4,2,
382     NOTE_A4,4,        NOTE_FS4,8,      NOTE_G4,      NOTE_A4,4,
383     NOTE_FS4,8,      NOTE_G4,      NOTE_A4,      NOTE_A3,      NOTE_B3,      NOTE_CS4,
384     NOTE_D4,      NOTE_E4,      NOTE_FS4,        NOTE_G4,      NOTE_FS4,4,      NOTE_D4,8,        NOTE_E4,
385     NOTE_FS4,        NOTE_CS4,        NOTE_A3,      NOTE_A3,
386
387     NOTE_CS4,4,      NOTE_B3,      NOTE_D4,8,        NOTE_CS4,        NOTE_B3,4,
```

```
388     NOTE_A3,8,       NOTE_G3,     NOTE_A3,4,       NOTE_D3,8,       NOTE_E3,     NOTE_FS3,
        NOTE_G3,
389     NOTE_A3,     NOTE_B3,4,       NOTE_G3,     NOTE_B3,8,       NOTE_A3,     NOTE_B3,4,
390     NOTE_CS4,8,      NOTE_D4,     NOTE_A3,     NOTE_B3,     NOTE_CS4,       NOTE_D4,     NOTE_E4,
391     NOTE_FS4,        NOTE_G4,     NOTE_A4,2,
392     MUSIC_END
393 };
394
395 //#endif
396
397 #endif
```

6.85 D:/Progetti/TTK4155/Node3/notes.h File Reference

Macros

- #define NOTE_B0 31
- #define NOTE_C1 33
- #define NOTE_CS1 35
- #define NOTE_D1 37
- #define NOTE_DS1 39
- #define NOTE_E1 41
- #define NOTE_F1 44
- #define NOTE_FS1 46
- #define NOTE_G1 49
- #define NOTE_GS1 52
- #define NOTE_A1 55
- #define NOTE_AS1 58
- #define NOTE_B1 62
- #define NOTE_C2 65
- #define NOTE_CS2 69
- #define NOTE_D2 73
- #define NOTE_DS2 78
- #define NOTE_E2 82
- #define NOTE_F2 87
- #define NOTE_FS2 93
- #define NOTE_G2 98
- #define NOTE_GS2 104
- #define NOTE_A2 110
- #define NOTE_AS2 117
- #define NOTE_B2 123
- #define NOTE_C3 131
- #define NOTE_CS3 139
- #define NOTE_D3 147
- #define NOTE_DS3 156
- #define NOTE_E3 165
- #define NOTE_F3 175
- #define NOTE_FS3 185
- #define NOTE_G3 196
- #define NOTE_GS3 208
- #define NOTE_A3 220
- #define NOTE_AS3 233
- #define NOTE_B3 247
- #define NOTE_C4 262
- #define NOTE_CS4 277
- #define NOTE_D4 294

- #define NOTE_DS4 311
- #define NOTE_E4 330
- #define NOTE_F4 349
- #define NOTE_FS4 370
- #define NOTE_G4 392
- #define NOTE_GS4 415
- #define NOTE_A4 440
- #define NOTE_AS4 466
- #define NOTE_B4 494
- #define NOTE_C5 523
- #define NOTE_CS5 554
- #define NOTE_D5 587
- #define NOTE_DS5 622
- #define NOTE_E5 659
- #define NOTE_F5 698
- #define NOTE_FS5 740
- #define NOTE_G5 784
- #define NOTE_GS5 831
- #define NOTE_A5 880
- #define NOTE_AS5 932
- #define NOTE_B5 988
- #define NOTE_C6 1047
- #define NOTE_CS6 1109
- #define NOTE_D6 1175
- #define NOTE_DS6 1245
- #define NOTE_E6 1319
- #define NOTE_F6 1397
- #define NOTE_FS6 1480
- #define NOTE_G6 1568
- #define NOTE_GS6 1661
- #define NOTE_A6 1760
- #define NOTE_AS6 1865
- #define NOTE_B6 1976
- #define NOTE_C7 2093
- #define NOTE_CS7 2217
- #define NOTE_D7 2349
- #define NOTE_DS7 2489
- #define NOTE_E7 2637
- #define NOTE_F7 2794
- #define NOTE_FS7 2960
- #define NOTE_G7 3136
- #define NOTE_GS7 3322
- #define NOTE_A7 3520
- #define NOTE_AS7 3729
- #define NOTE_B7 3951
- #define NOTE_C8 4186
- #define NOTE_CS8 4435
- #define NOTE_D8 4699
- #define NOTE_DS8 4978
- #define REST 1
- #define MUSIC_END 0

6.85.1 Macro Definition Documentation

6.85.1.1 MUSIC_END

```
#define MUSIC_END 0
```

6.85.1.2 NOTE_A1

```
#define NOTE_A1 55
```

6.85.1.3 NOTE_A2

```
#define NOTE_A2 110
```

6.85.1.4 NOTE_A3

```
#define NOTE_A3 220
```

6.85.1.5 NOTE_A4

```
#define NOTE_A4 440
```

6.85.1.6 NOTE_A5

```
#define NOTE_A5 880
```

6.85.1.7 NOTE_A6

```
#define NOTE_A6 1760
```

6.85.1.8 NOTE_A7

```
#define NOTE_A7 3520
```

6.85.1.9 NOTE_AS1

```
#define NOTE_AS1 58
```

6.85.1.10 NOTE_AS2

```
#define NOTE_AS2 117
```

6.85.1.11 NOTE_AS3

```
#define NOTE_AS3 233
```

6.85.1.12 NOTE_AS4

```
#define NOTE_AS4 466
```

6.85.1.13 NOTE_AS5

```
#define NOTE_AS5 932
```

6.85.1.14 NOTE_AS6

```
#define NOTE_AS6 1865
```

6.85.1.15 NOTE_AS7

```
#define NOTE_AS7 3729
```

6.85.1.16 NOTE_B0

```
#define NOTE_B0 31
```

6.85.1.17 NOTE_B1

```
#define NOTE_B1 62
```

6.85.1.18 NOTE_B2

```
#define NOTE_B2 123
```

6.85.1.19 NOTE_B3

```
#define NOTE_B3 247
```

6.85.1.20 NOTE_B4

```
#define NOTE_B4 494
```

6.85.1.21 NOTE_B5

```
#define NOTE_B5 988
```

6.85.1.22 NOTE_B6

```
#define NOTE_B6 1976
```

6.85.1.23 NOTE_B7

```
#define NOTE_B7 3951
```

6.85.1.24 NOTE_C1

```
#define NOTE_C1 33
```

6.85.1.25 NOTE_C2

```
#define NOTE_C2 65
```

6.85.1.26 NOTE_C3

```
#define NOTE_C3 131
```

6.85.1.27 NOTE_C4

```
#define NOTE_C4 262
```

6.85.1.28 NOTE_C5

```
#define NOTE_C5 523
```

6.85.1.29 NOTE_C6

```
#define NOTE_C6 1047
```

6.85.1.30 NOTE_C7

```
#define NOTE_C7 2093
```

6.85.1.31 NOTE_C8

```
#define NOTE_C8 4186
```

6.85.1.32 NOTE_CS1

```
#define NOTE_CS1 35
```

6.85.1.33 NOTE_CS2

```
#define NOTE_CS2 69
```

6.85.1.34 NOTE_CS3

```
#define NOTE_CS3 139
```

6.85.1.35 NOTE_CS4

```
#define NOTE_CS4 277
```

6.85.1.36 NOTE_CS5

```
#define NOTE_CS5 554
```

6.85.1.37 NOTE_CS6

```
#define NOTE_CS6 1109
```

6.85.1.38 NOTE_CS7

```
#define NOTE_CS7 2217
```

6.85.1.39 NOTE_CS8

```
#define NOTE_CS8 4435
```

6.85.1.40 NOTE_D1

```
#define NOTE_D1 37
```

6.85.1.41 NOTE_D2

```
#define NOTE_D2 73
```

6.85.1.42 NOTE_D3

```
#define NOTE_D3 147
```

6.85.1.43 NOTE_D4

```
#define NOTE_D4 294
```

6.85.1.44 NOTE_D5

```
#define NOTE_D5 587
```

6.85.1.45 NOTE_D6

```
#define NOTE_D6 1175
```

6.85.1.46 NOTE_D7

```
#define NOTE_D7 2349
```

6.85.1.47 NOTE_D8

```
#define NOTE_D8 4699
```

6.85.1.48 NOTE_DS1

```
#define NOTE_DS1 39
```

6.85.1.49 NOTE_DS2

```
#define NOTE_DS2 78
```

6.85.1.50 NOTE_DS3

```
#define NOTE_DS3 156
```

6.85.1.51 NOTE_DS4

```
#define NOTE_DS4 311
```

6.85.1.52 NOTE_DS5

```
#define NOTE_DS5 622
```

6.85.1.53 NOTE_DS6

```
#define NOTE_DS6 1245
```

6.85.1.54 NOTE_DS7

```
#define NOTE_DS7 2489
```

6.85.1.55 NOTE_DS8

```
#define NOTE_DS8 4978
```

6.85.1.56 NOTE_E1

```
#define NOTE_E1 41
```

6.85.1.57 NOTE_E2

```
#define NOTE_E2 82
```

6.85.1.58 NOTE_E3

```
#define NOTE_E3 165
```

6.85.1.59 NOTE_E4

```
#define NOTE_E4 330
```

6.85.1.60 NOTE_E5

```
#define NOTE_E5 659
```

6.85.1.61 NOTE_E6

```
#define NOTE_E6 1319
```

6.85.1.62 NOTE_E7

```
#define NOTE_E7 2637
```

6.85.1.63 NOTE_F1

```
#define NOTE_F1 44
```

6.85.1.64 NOTE_F2

```
#define NOTE_F2 87
```

6.85.1.65 NOTE_F3

```
#define NOTE_F3 175
```

6.85.1.66 NOTE_F4

```
#define NOTE_F4 349
```

6.85.1.67 NOTE_F5

```
#define NOTE_F5 698
```

6.85.1.68 NOTE_F6

```
#define NOTE_F6 1397
```

6.85.1.69 NOTE_F7

```
#define NOTE_F7 2794
```

6.85.1.70 NOTE_FS1

```
#define NOTE_FS1 46
```

6.85.1.71 NOTE_FS2

```
#define NOTE_FS2 93
```

6.85.1.72 NOTE_FS3

```
#define NOTE_FS3 185
```

6.85.1.73 NOTE_FS4

```
#define NOTE_FS4 370
```

6.85.1.74 NOTE_FS5

```
#define NOTE_FS5 740
```

6.85.1.75 NOTE_FS6

```
#define NOTE_FS6 1480
```

6.85.1.76 NOTE_FS7

```
#define NOTE_FS7 2960
```

6.85.1.77 NOTE_G1

```
#define NOTE_G1 49
```

6.85.1.78 NOTE_G2

```
#define NOTE_G2 98
```

6.85.1.79 NOTE_G3

```
#define NOTE_G3 196
```

6.85.1.80 NOTE_G4

```
#define NOTE_G4 392
```

6.85.1.81 NOTE_G5

```
#define NOTE_G5 784
```

6.85.1.82 NOTE_G6

```
#define NOTE_G6 1568
```

6.85.1.83 NOTE_G7

```
#define NOTE_G7 3136
```

6.85.1.84 NOTE_GS1

```
#define NOTE_GS1 52
```

6.85.1.85 NOTE_GS2

```
#define NOTE_GS2 104
```

6.85.1.86 NOTE_GS3

```
#define NOTE_GS3 208
```

6.85.1.87 NOTE_GS4

```
#define NOTE_GS4 415
```

6.85.1.88 NOTE_GS5

```
#define NOTE_GS5 831
```

6.85.1.89 NOTE_GS6

```
#define NOTE_GS6 1661
```

6.85.1.90 NOTE_GS7

```
#define NOTE_GS7 3322
```

6.85.1.91 REST

```
#define REST 1
```

6.86 notes.h

Go to the documentation of this file.

```
1 #ifndef NOTES
2 #define NOTES
3
4 #define NOTE_B0  31
5 #define NOTE_C1  33
6 #define NOTE_CS1 35
7 #define NOTE_D1  37
8 #define NOTE_DS1 39
9 #define NOTE_E1  41
10 #define NOTE_F1  44
11 #define NOTE_FS1 46
12 #define NOTE_G1  49
13 #define NOTE_GS1 52
14 #define NOTE_A1  55
15 #define NOTE_AS1 58
16 #define NOTE_B1  62
17
18 #define NOTE_C2  65
19 #define NOTE_CS2 69
20 #define NOTE_D2  73
21 #define NOTE_DS2 78
22 #define NOTE_E2  82
23 #define NOTE_F2  87
24 #define NOTE_FS2 93
25 #define NOTE_G2  98
26 #define NOTE_GS2 104
27 #define NOTE_A2  110
28 #define NOTE_AS2 117
29 #define NOTE_B2  123
30
31 #define NOTE_C3  131
32 #define NOTE_CS3 139
33 #define NOTE_D3  147
34 #define NOTE_DS3 156
35 #define NOTE_E3  165
36 #define NOTE_F3  175
37 #define NOTE_FS3 185
38 #define NOTE_G3  196
```

```
39 #define NOTE_GS3 208
40 #define NOTE_A3  220
41 #define NOTE_AS3 233
42 #define NOTE_B3  247
43
44 #define NOTE_C4  262
45 #define NOTE_CS4 277
46 #define NOTE_D4  294
47 #define NOTE_DS4 311
48 #define NOTE_E4  330
49 #define NOTE_F4  349
50 #define NOTE_FS4 370
51 #define NOTE_G4  392
52 #define NOTE_GS4 415
53 #define NOTE_A4  440
54 #define NOTE_AS4 466
55 #define NOTE_B4  494
56
57 #define NOTE_C5  523
58 #define NOTE_CS5 554
59 #define NOTE_D5  587
60 #define NOTE_DS5 622
61 #define NOTE_E5  659
62 #define NOTE_F5  698
63 #define NOTE_FS5 740
64 #define NOTE_G5  784
65 #define NOTE_GS5 831
66 #define NOTE_A5  880
67 #define NOTE_AS5 932
68 #define NOTE_B5  988
69
70 #define NOTE_C6  1047
71 #define NOTE_CS6 1109
72 #define NOTE_D6  1175
73 #define NOTE_DS6 1245
74 #define NOTE_E6  1319
75 #define NOTE_F6  1397
76 #define NOTE_FS6 1480
77 #define NOTE_G6  1568
78 #define NOTE_GS6 1661
79 #define NOTE_A6  1760
80 #define NOTE_AS6 1865
81 #define NOTE_B6  1976
82
83 #define NOTE_C7  2093
84 #define NOTE_CS7 2217
85 #define NOTE_D7  2349
86 #define NOTE_DS7 2489
87 #define NOTE_E7  2637
88 #define NOTE_F7  2794
89 #define NOTE_FS7 2960
90 #define NOTE_G7  3136
91 #define NOTE_GS7 3322
92 #define NOTE_A7  3520
93 #define NOTE_AS7 3729
94 #define NOTE_B7  3951
95
96 #define NOTE_C8  4186
97 #define NOTE_CS8 4435
98 #define NOTE_D8  4699
99 #define NOTE_DS8 4978
100
101 #define REST 1
102 #define MUSIC_END 0
103
104 #endif
```

6.87 D:/Progetti/TTK4155/README.md File Reference

Index