Arduino

1.0

Generated by Doxygen 1.8.14

Contents

1 Class Index **1**

1.1 Class List 1

2 File Index **3**

2.1 File List 3

3 Class Documentation **5**

3.1 dht22 Class Reference 5

3.1.1 Detailed Description 5

3.1.2 Constructor & Destructor Documentation 5

3.1.2.1 dht22() 5

3.1.3 Member Function Documentation 6

3.1.3.1 read() 6

3.2 dio Class Reference 7

3.2.1 Detailed Description 7

3.2.2 Constructor & Destructor Documentation 7

3.2.2.1 dio() 7

3.2.3 Member Function Documentation 7

3.2.3.1 dio_changePortBPinCnf() 7

3.2.3.2 dio_getPortB() 8

3.2.3.3 dio_invertPortB() 9

3.2.3.4 dio_setPortB() 9

3.3 keepAliveLed Class Reference 10

3.3.1 Detailed Description 10

3.3.2 Constructor & Destructor Documentation . . . 10
3.3.2.1 keepAliveLed() . . . 11
3.3.3 Member Function Documentation . . . 11
3.3.3.1 blinkLed_task() . . . 11
3.4 scheduler Class Reference . . . 12
3.4.1 Detailed Description . . . 12
3.4.2 Constructor & Destructor Documentation . . . 12
3.4.2.1 scheduler() . . . 13
3.4.3 Member Function Documentation . . . 13
3.4.3.1 addPeriodicTask() . . . 13
3.4.3.2 getPitNumber() . . . 14
3.4.3.3 launchPeriodicTasks() . . . 14
3.4.3.4 startScheduling() . . . 15
3.5 T_ASW_cnf_struct Struct Reference . . . 16
3.5.1 Detailed Description . . . 16
3.5.2 Member Data Documentation . . . 16
3.5.2.1 p_keepAliveLed . . . 16
3.5.2.2 p_TempSensor . . . 17
3.5.2.3 p_usartDebug . . . 17
3.6 T_BSW_cnf_struct Struct Reference . . . 17
3.6.1 Detailed Description . . . 18
3.6.2 Member Data Documentation . . . 18
3.6.2.1 p_dht22 . . . 18
3.6.2.2 p_dio . . . 18
3.6.2.3 p_timer . . . 18
3.6.2.4 p_usart . . . 18
3.7 TempSensor Class Reference . . . 19
3.7.1 Detailed Description . . . 19
3.7.2 Constructor & Destructor Documentation . . . 19
3.7.2.1 TempSensor() . . . 19

3.7.3 Member Function Documentation . . . 20
3.7.3.1 readTempSensor_task() . . . 20
3.8 timer Class Reference . . . 20
3.8.1 Detailed Description . . . 21
3.8.2 Constructor & Destructor Documentation . . . 21
3.8.2.1 timer() . . . 21
3.8.3 Member Function Documentation . . . 21
3.8.3.1 configureTimer1() . . . 21
3.8.3.2 startTimer1() . . . 22
3.8.3.3 stopTimer1() . . . 23
3.9 usart Class Reference . . . 23
3.9.1 Detailed Description . . . 23
3.9.2 Constructor & Destructor Documentation . . . 23
3.9.2.1 usart() . . . 23
3.9.3 Member Function Documentation . . . 24
3.9.3.1 setBaudRate() . . . 24
3.9.3.2 usart_init() . . . 25
3.9.3.3 usart_sendString() . . . 25
3.10 UsartDebug Class Reference . . . 26
3.10.1 Detailed Description . . . 26
3.10.2 Constructor & Destructor Documentation . . . 27
3.10.2.1 UsartDebug() . . . 27
3.10.3 Member Function Documentation . . . 27
3.10.3.1 sendBool() . . . 27
3.10.3.2 sendData() . . . 28
3.10.3.3 sendInteger() . . . 29

4 File Documentation **31**

4.1 work/asw/asw.cpp File Reference . . . 31

4.1.1 Detailed Description . . . 32

4.1.2 Function Documentation . . . 32

4.1.2.1 asw_init() . . . 32

4.1.3 Variable Documentation . . . 32

4.1.3.1 ASW_cnf_struct . . . 33

4.2 work/asw/asw.h File Reference . . . 33

4.2.1 Detailed Description . . . 34

4.2.2 Function Documentation . . . 34

4.2.2.1 asw_init() . . . 34

4.2.3 Variable Documentation . . . 35

4.2.3.1 ASW_cnf_struct . . . 35

4.3 work/asw/keepAliveLed/keepAliveLed.cpp File Reference . . . 35

4.3.1 Detailed Description . . . 35

4.4 work/asw/keepAliveLed/keepAliveLed.h File Reference . . . 36

4.4.1 Detailed Description . . . 37

4.4.2 Macro Definition Documentation . . . 37

4.4.2.1 PERIOD_MS_TASK_LED . . . 37

4.5 work/asw/log/log.cpp File Reference . . . 38

4.5.1 Detailed Description . . . 38

4.6 work/asw/log/log.h File Reference . . . 38

4.6.1 Detailed Description . . . 40

4.7 work/asw/TempSensor/TempSensor.cpp File Reference . . . 40

4.7.1 Detailed Description . . . 41

4.8 work/asw/TempSensor/TempSensor.h File Reference . . . 41

4.8.1 Detailed Description . . . 42

4.8.2 Macro Definition Documentation . . . 42

4.8.2.1 PERIOD_MS_TASK_TEMP_SENSOR . . . 43

4.9 work/bsw/bsw.cpp File Reference . . . 43

4.9.1 Detailed Description . . . 43
4.9.2 Function Documentation . . . 44
4.9.2.1 bsw_init() . . . 44
4.9.3 Variable Documentation . . . 44
4.9.3.1 BSW_cnf_struct . . . 44
4.10 work/bsw/bsw.h File Reference . . . 45
4.10.1 Detailed Description . . . 46
4.10.2 Macro Definition Documentation . . . 46
4.10.2.1 USART_BAUDRATE . . . 46
4.10.3 Function Documentation . . . 46
4.10.3.1 bsw_init() . . . 47
4.10.4 Variable Documentation . . . 47
4.10.4.1 BSW_cnf_struct . . . 47
4.11 work/bsw/dht22/dht22.cpp File Reference . . . 48
4.11.1 Detailed Description . . . 48
4.11.2 Macro Definition Documentation . . . 48
4.11.2.1 MAX_WAIT_TIME_US . . . 49
4.12 work/bsw/dht22/dht22.h File Reference . . . 49
4.12.1 Detailed Description . . . 50
4.12.2 Macro Definition Documentation . . . 50
4.12.2.1 DHT22_PORT . . . 51
4.13 work/bsw/dio/dio.cpp File Reference . . . 51
4.13.1 Detailed Description . . . 51
4.13.2 Macro Definition Documentation . . . 52
4.13.2.1 PORTB_CNF_DDRB . . . 52
4.13.2.2 PORTB_CNF_PORTB . . . 52
4.14 work/bsw/dio/dio.h File Reference . . . 53
4.14.1 Detailed Description . . . 53
4.14.2 Macro Definition Documentation . . . 53
4.14.2.1 PORT_CNF_IN . . . 54

4.14.2.2 PORT_CNF_OUT . . . 54
4.15 work/bsw/timer/timer.cpp File Reference . . . 54
4.15.1 Detailed Description . . . 54
4.16 work/bsw/timer/timer.h File Reference . . . 55
4.16.1 Detailed Description . . . 55
4.17 work/bsw/usart/usart.cpp File Reference . . . 56
4.17.1 Detailed Description . . . 56
4.18 work/bsw/usart/usart.h File Reference . . . 56
4.18.1 Detailed Description . . . 57
4.19 work/lib/operators.cpp File Reference . . . 57
4.19.1 Detailed Description . . . 58
4.19.2 Function Documentation . . . 58
4.19.2.1 operator delete() . . . 58
4.19.2.2 operator new() . . . 59
4.20 work/lib/operators.h File Reference . . . 59
4.20.1 Detailed Description . . . 60
4.20.2 Function Documentation . . . 60
4.20.2.1 operator delete() . . . 61
4.20.2.2 operator new() . . . 62
4.21 work/main.cpp File Reference . . . 62
4.21.1 Detailed Description . . . 63
4.21.2 Function Documentation . . . 63
4.21.2.1 ISR() . . . 64
4.21.2.2 main() . . . 64
4.22 work/main.h File Reference . . . 65
4.22.1 Detailed Description . . . 66
4.23 work/scheduler/scheduler.cpp File Reference . . . 66
4.23.1 Detailed Description . . . 66
4.23.2 Variable Documentation . . . 67
4.23.2.1 p_scheduler . . . 67
4.24 work/scheduler/scheduler.h File Reference . . . 67
4.24.1 Detailed Description . . . 68
4.24.2 Macro Definition Documentation . . . 68
4.24.2.1 PRESCALER_PERIODIC_TIMER . . . 68
4.24.2.2 SW_PERIOD_MS . . . 69
4.24.2.3 TIMER_CTC_VALUE . . . 69
4.24.3 Typedef Documentation . . . 69
4.24.3.1 TaskPtr_t . . . 69
4.24.4 Variable Documentation . . . 69
4.24.4.1 p_scheduler . . . 69

Index **71**

Chapter 1

Class Index

1.1 Class List

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

dht22
: DHT 22 driver class . . . 5

dio
: DIO class . . . 7

keepAliveLed
: Class for keep-alive LED blinking . . . 10

scheduler
: Scheduler class . . . 12

T_ASW_cnf_struct
: ASW configuration structure . . . 16

T_BSW_cnf_struct
: BSW configuration structure . . . 17

TempSensor . . . 19

timer
: Class defining a timer . . . 20

usart
: USART serial bus class . . . 23

UsartDebug
: Class used for debugging on usart link . . . 26

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

work/main.cpp
Background task file . . . 62
work/main.h
Background task header file . . . 65
work/asw/asw.cpp
ASW main file . . . 31
work/asw/asw.h
ASW main header file . . . 33
work/asw/keepAliveLed/keepAliveLed.cpp
Definition of function for class keepAliveLed . . . 35
work/asw/keepAliveLed/keepAliveLed.h
Class keepAliveLed header file . . . 36
work/asw/log/log.cpp
This file defines classes for log and debug data transmission on USART link . . . 38
work/asw/log/log.h
Header file for debug and logging functions . . . 38
work/asw/TempSensor/TempSensor.cpp
Defines function of class TempSensor . . . 40
work/asw/TempSensor/TempSensor.h
Class TempSensor header file . . . 41
work/bsw/bsw.cpp
BSW main file . . . 43
work/bsw/bsw.h
BSW main header file . . . 45
work/bsw/dht22/dht22.cpp
This file defines classes for DHT22 driver . . . 48
work/bsw/dht22/dht22.h
DHT22 driver header file . . . 49
work/bsw/dio/dio.cpp
DIO library . . . 51
work/bsw/dio/dio.h
DIO library header file . . . 53
work/bsw/timer/timer.cpp
Defines function for class timer . . . 54
work/bsw/timer/timer.h
Timer class header file . . . 55

work/bsw/usart/usart.cpp
BSW library for USART 56
work/bsw/usart/usart.h
Header file for USART library 56
work/lib/operators.cpp
C++ operators definitions 57
work/lib/operators.h
C++ operators definitions header file 59
work/scheduler/scheduler.cpp
Defines scheduler class 66
work/scheduler/scheduler.h
Scheduler class header file 67

Chapter 3

Class Documentation

3.1 dht22 Class Reference

DHT 22 driver class.

`#include <dht22.h>`

Public Member Functions

- dht22 ()
  dht22 class constructor
- bool read ()
  Reads the data from DHT22.

3.1.1 Detailed Description

DHT 22 driver class.

This class defines all useful functions for DHT22 temperature and humidity sensor

Definition at line 22 of file dht22.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 dht22()

`dht22::dht22 ( )`

dht22 class constructor

Initializes the class dht22

Returns

Nothing

Definition at line 22 of file dht22.cpp.

3.1.3 Member Function Documentation

3.1.3.1 read()

```
bool dht22::read ( )
```

Reads the data from DHT22.

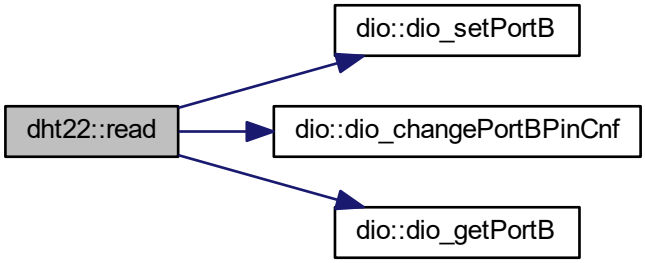
This function communicates with DHT22 using 1-wire protocol to read raw values of temperature and humidity. A checksum check is done when communication is finished to validate the received data

Returns

Validity of the read value

Definition at line 30 of file dht22.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- work/bsw/dht22/dht22.h
- work/bsw/dht22/dht22.cpp

3.2 dio Class Reference

DIO class.

`#include <dio.h>`

Public Member Functions

- dio ()
  dio class constructor
- void dio_setPortB (uint8_t pin, bool state)
  Port B setting function.
- void dio_invertPortB (uint8_t pin)
  Inverts the state of output port.
- bool dio_getPortB (uint8_t pin)
  Gets the logical state of selected pin.
- void dio_changePortBPinCnf (uint8_t pin, uint8_t cnf)
  Changes the IO configuration of the selected pin of port B.

3.2.1 Detailed Description

DIO class.

This class defines all useful functions for digital input/output ports

Definition at line 22 of file dio.h.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 dio()

```
dio::dio ( )
```

dio class constructor

Initializes class dio and calls DIO hardware intialization function

Returns

Nothing

Definition at line 52 of file dio.cpp.

3.2.3 Member Function Documentation

3.2.3.1 dio_changePortBPinCnf()

```
void dio::dio_changePortBPinCnf (
            uint8_t pin,
            uint8_t cnf )
```

Changes the IO configuration of the selected pin of port B.

This function configures the selected pin of port B as input or output according to parameter cnf

Parameters

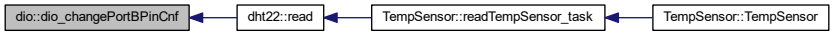
in	*pin*	pin selected for configuration change
in	*cnf*	Requested configuration for the selected pin PORT_CNF_OUT (1) : pin configured as output PORT_CNF_IN (0) : pin configured as input

Returns

Nothing

Definition at line 84 of file dio.cpp.

Here is the caller graph for this function:



3.2.3.2 dio_getPortB()

```
bool dio::dio_getPortB (
            uint8_t pin )
```

Gets the logical state of selected pin.

This function gets the logical value of the selected pin of port B

Parameters

in	*pin*	pin of port B to get value

Returns

Logical state of selected pin

Definition at line 75 of file dio.cpp.

Here is the caller graph for this function:

3.2.3.3 dio_invertPortB()

```
void dio::dio_invertPortB (
            uint8_t pin )
```

Inverts the state of output port.

This function inverts the state of the chosen pin of port B

Parameters

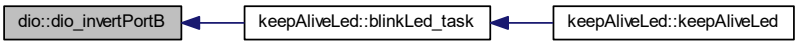
in	*pin*	Pin to invert

Returns

Nothing

Definition at line 69 of file dio.cpp.

Here is the caller graph for this function:



3.2.3.4 dio_setPortB()

```
void dio::dio_setPortB (
            uint8_t pin,
            bool state )
```

Port B setting function.

This function sets the requested digital output on port B to the requested state

Parameters

in	*pin*	pin of PORT B to set
in	*state*	requested state to set pin

Returns

Nothing

Definition at line 58 of file dio.cpp.

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- work/bsw/dio/dio.h
- work/bsw/dio/dio.cpp

3.3 keepAliveLed Class Reference

Class for keep-alive LED blinking.

`#include <keepAliveLed.h>`

Public Member Functions

- keepAliveLed ()
  Class constructor.

Static Public Member Functions

- static void blinkLed_task ()
  Task for LED blinking.

3.3.1 Detailed Description

Class for keep-alive LED blinking.

This class defines all functions to make keep-alive LED blink

Definition at line 21 of file keepAliveLed.h.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 keepAliveLed()

```
keepAliveLed::keepAliveLed ( )
```

Class constructor.

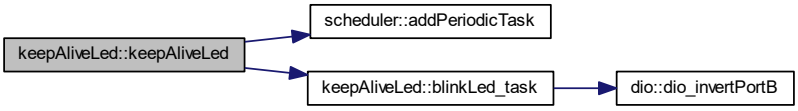
This function initializes the class keppAliveLed

Returns

Nothing

Definition at line 15 of file keepAliveLed.cpp.

Here is the call graph for this function:



3.3.3 Member Function Documentation

3.3.3.1 blinkLed_task()

```
void keepAliveLed::blinkLed_task ( )  [static]
```

Task for LED blinking.

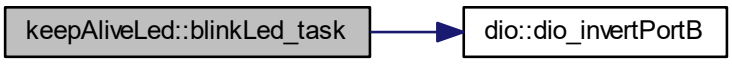
This function is inserted into the scheduler. It changes the state of the LED output to make it blink

Returns

Nothing

Definition at line 21 of file keepAliveLed.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- work/asw/keepAliveLed/keepAliveLed.h
- work/asw/keepAliveLed/keepAliveLed.cpp

3.4 scheduler Class Reference

Scheduler class.

`#include <scheduler.h>`

Public Member Functions

- scheduler ()
  scheduler class constructor
- void launchPeriodicTasks ()
  Main scheduler function.
- void startScheduling ()
  Starts the tasks scheduling.
- void addPeriodicTask (TaskPtr_t task_ptr, uint16_t a_period)
  Add a task into the scheduler.
- uint32_t getPitNumber ()
  Get function for PIT number.

3.4.1 Detailed Description

Scheduler class.

This class defines the scheduler of the system. It is called by the main interrupt and calls successively all applicative functions according to their recurrence time.

Definition at line 32 of file scheduler.h.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 scheduler()

```
scheduler::scheduler ( )
```

scheduler class constructor

This function initializes the class scheduler

Returns

Nothing

Definition at line 19 of file scheduler.cpp.

Here is the call graph for this function:

3.4.3 Member Function Documentation

3.4.3.1 addPeriodicTask()

```
void scheduler::addPeriodicTask (
            TaskPtr_t task_ptr,
            uint16_t a_period )
```

Add a task into the scheduler.

This function create a new task in the scheduler linked to the function task_ptr with a period a_period

Parameters

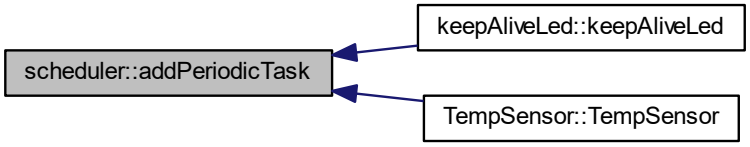
in	*task_ptr*	Pointer to the task which will be added
in	*a_period*	Period of the new task

Returns

Nothing

Definition at line 73 of file scheduler.cpp.

Here is the caller graph for this function:



3.4.3.2 getPitNumber()

```
uint32_t scheduler::getPitNumber ( )  [inline]
```

Get function for PIT number.

This function returns the PIT number

Returns

PIT number

Definition at line 103 of file scheduler.cpp.

3.4.3.3 launchPeriodicTasks()

```
void scheduler::launchPeriodicTasks ( )
```

Main scheduler function.

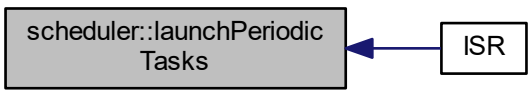
This function launches the scheduled tasks according to current software time and task configuration

Returns

Nothing

Definition at line 32 of file scheduler.cpp.

Here is the caller graph for this function:



3.4.3.4 startScheduling()

```
void scheduler::startScheduling ( )
```

Starts the tasks scheduling.

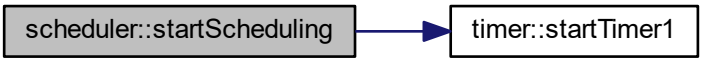
This function starts the timer which will trigger an interrupt every software period. When the interrupt is raised the scheduler will launch applications

Returns

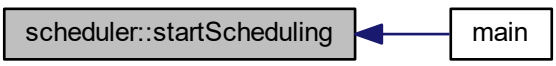
Nothing

Definition at line 67 of file scheduler.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

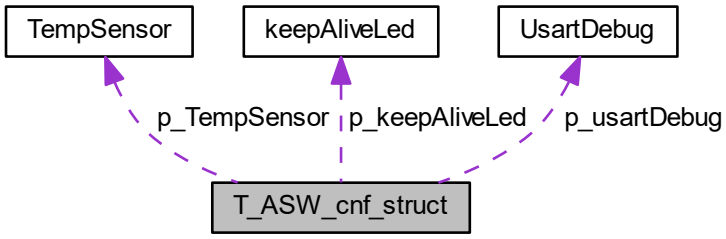
- work/scheduler/scheduler.h
- work/scheduler/scheduler.cpp

3.5 T_ASW_cnf_struct Struct Reference

ASW configuration structure.

`#include <asw.h>`

Collaboration diagram for T_ASW_cnf_struct:



Public Attributes

- UsartDebug * p_usartDebug
- keepAliveLed * p_keepAliveLed
- TempSensor * p_TempSensor

3.5.1 Detailed Description

ASW configuration structure.

This structure contains all pointers to instanced applicative objects

Definition at line 23 of file asw.h.

3.5.2 Member Data Documentation

3.5.2.1 p_keepAliveLed

```
keepAliveLed* T_ASW_cnf_struct::p_keepAliveLed
```

Pointer to keepAliveLed object

Definition at line 26 of file asw.h.

3.5.2.2 p_TempSensor

```
TempSensor* T_ASW_cnf_struct::p_TempSensor
```

Pointer to TempSensor object

Definition at line 27 of file asw.h.

3.5.2.3 p_usartDebug

```
UsartDebug* T_ASW_cnf_struct::p_usartDebug
```

Pointer to usart debug object

Definition at line 25 of file asw.h.

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

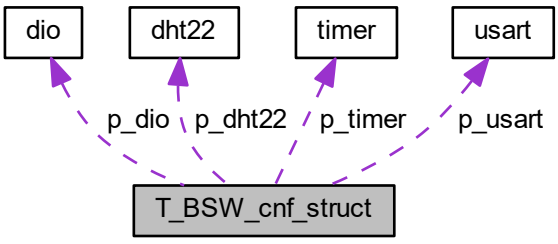
- work/asw/asw.h

3.6 T_BSW_cnf_struct Struct Reference

BSW configuration structure.

```
#include <bsw.h>
```

Collaboration diagram for T_BSW_cnf_struct:



Public Attributes

- usart * p_usart
- dio * p_dio
- timer * p_timer
- dht22 * p_dht22

3.6.1 Detailed Description

BSW configuration structure.

This structure contains all pointers to instanced drivers objects

Definition at line 29 of file bsw.h.

3.6.2 Member Data Documentation

3.6.2.1 p_dht22

```
dht22* T_BSW_cnf_struct::p_dht22
```

Pointer to dht22 driver object

Definition at line 34 of file bsw.h.

3.6.2.2 p_dio

```
dio* T_BSW_cnf_struct::p_dio
```

Pointer to dio driver object

Definition at line 32 of file bsw.h.

3.6.2.3 p_timer

```
timer* T_BSW_cnf_struct::p_timer
```

Pointer to timer driver object

Definition at line 33 of file bsw.h.

3.6.2.4 p_usart

```
usart* T_BSW_cnf_struct::p_usart
```

Pointer to usart driver object

Definition at line 31 of file bsw.h.

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

- work/bsw/bsw.h

3.7 TempSensor Class Reference

`#include <TempSensor.h>`

Public Member Functions

- TempSensor ()
  Class constructor.

Static Public Member Functions

- static void readTempSensor_task ()
  Task for reading temperature and humidity values.

3.7.1 Detailed Description

Definition at line 15 of file TempSensor.h.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 TempSensor()

`TempSensor::TempSensor ( )`

Class constructor.

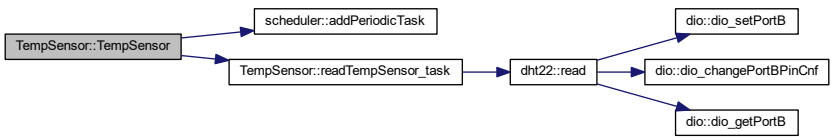
This function initializes all data of the class TempSensor

Returns

Nothing

Definition at line 16 of file TempSensor.cpp.

Here is the call graph for this function:



3.7.3 Member Function Documentation

3.7.3.1 readTempSensor_task()

```
void TempSensor::readTempSensor_task ( )  [static]
```

Task for reading temperature and humidity values.

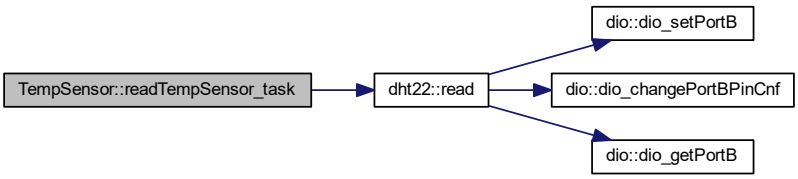
This task reads temperature and humidity data using DHT22 driver. It is called every 5 seconds.

Returns

Nothing

Definition at line 22 of file TempSensor.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- work/asw/TempSensor/TempSensor.h
- work/asw/TempSensor/TempSensor.cpp

3.8 timer Class Reference

Class defining a timer.

```
#include <timer.h>
```

Public Member Functions

- timer ()
  Class constructor.
- void configureTimer1 (uint16_t a_prescaler, uint16_t a_ctcValue)
  Configures Timer #1.
- void startTimer1 ()
  Start Timer #1.
- void stopTimer1 ()
  Stops Timer #1.

3.8.1 Detailed Description

Class defining a timer.

This class defines a timer/counter. The selected timer is configured in CTC mode and interrupts are enabled. The prescaler value and CTC value can both be configured by user.

Definition at line 22 of file timer.h.

3.8.2 Constructor & Destructor Documentation

3.8.2.1 timer()

```
timer::timer ( )
```

Class constructor.

This function initializes class attributes

Returns

Nothing

Definition at line 13 of file timer.cpp.

3.8.3 Member Function Documentation

3.8.3.1 configureTimer1()

```
void timer::configureTimer1 (
            uint16_t a_prescaler,
            uint16_t a_ctcValue )
```

Configures Timer #1.

This function configures hardware timer #1 in CTC mode, enables its interrupts, sets prescaler to a_prescaler and CTC value to a_ctcValue

Parameters

in	*a_prescaler*	prescaler value
in	*a_ctcValue*	Value to which the counter will compare before raising an interrupt

Returns

Nothing

Definition at line 18 of file timer.cpp.

Here is the caller graph for this function:

3.8.3.2 startTimer1()

```
void timer::startTimer1 ( )
```

Start Timer #1.

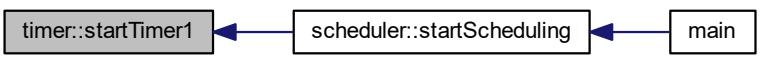
This functions starts Timer #1. Timer shall be initialized before this function is called.

Returns

Nothing

Definition at line 56 of file timer.cpp.

Here is the caller graph for this function:



3.8.3.3 stopTimer1()

```
void timer::stopTimer1 ( )
```

Stops Timer #1.

This functions stops timer #1 by resetting bits 0-2 of TCCR1B

Returns

Nothing

Definition at line 67 of file timer.cpp.

The documentation for this class was generated from the following files:

- work/bsw/timer/timer.h
- work/bsw/timer/timer.cpp

3.9 usart Class Reference

USART serial bus class.

```
#include <usart.h>
```

Public Member Functions

- usart (uint16_t a_BaudRate)
  Class usart constructor.
- void usart_sendString (uint8_t *str)
  Sending a string on USART link.
- void setBaudRate (uint16_t a_BaudRate)
  Setting baud rate.
- void usart_init ()
  USART hardware initialization.

3.9.1 Detailed Description

USART serial bus class.

This class defines all useful functions for USART serial bus

Definition at line 16 of file usart.h.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 usart()

```
usart::usart (
            uint16_t a_BaudRate )
```

Class usart constructor.

Initializes the class and call hardware initialization function

Parameters

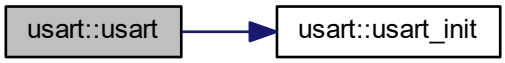
in	*a_BaudRate*	Desired Baud Rate (16 bit) - up to 57600

Returns

Nothing.

Definition at line 14 of file usart.cpp.

Here is the call graph for this function:



3.9.3 Member Function Documentation

3.9.3.1 setBaudRate()

```
void usart::setBaudRate (
            uint16_t a_BaudRate )  [inline]
```

Setting baud rate.

This function sets the attribute BaudRate of the class usart

Parameters

in	*a_BaudRate*	Desired Baud Rate (16 bit) - up to 57600

Returns

Nothing

Definition at line 62 of file usart.cpp.

3.9.3.2 usart_init()

```
void usart::usart_init ( )
```

USART hardware initialization.

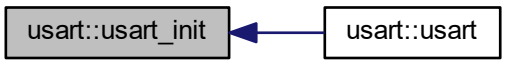
This function will initialize the USART using selected baudrate. User must pay attention to select one of the usually used Baud Rate (9600, 19200, 38400, 57600). Note that since an uint16 is used as argument, Baud rate cannot be more than 57600.

Returns

Nothing.

Definition at line 21 of file usart.cpp.

Here is the caller graph for this function:



3.9.3.3 usart_sendString()

```
void usart::usart_sendString (
            uint8_t * str )
```

Sending a string on USART link.

Just write data to the Serial link using usart_trabsmit function

Parameters

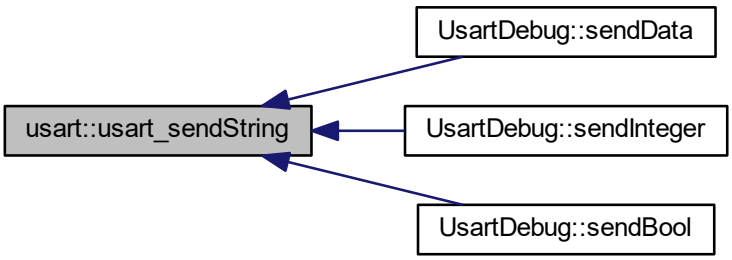
in	*str*	Pointer to the string being sent

Returns

Nothing.

Definition at line 43 of file usart.cpp.

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- work/bsw/usart/usart.h
- work/bsw/usart/usart.cpp

3.10 UsartDebug Class Reference

Class used for debugging on usart link.

`#include <log.h>`

Public Member Functions

- UsartDebug ()

  Class UsartDebug constructor.
- void sendData (char *str)

  Send a string on USART link.
- void sendInteger (uint16_t data, uint8_t base)

  Send a integer data on USART link.
- void sendBool (bool data)

  Send a boolean data on USART link.

3.10.1 Detailed Description

Class used for debugging on usart link.

This class defines functions used for sending debug data on USART link.

Definition at line 19 of file log.h.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 UsartDebug()

```
UsartDebug::UsartDebug ( )
```

Class UsartDebug constructor.

Initializes the class UsartDebug

Returns

Nothing

Definition at line 17 of file log.cpp.

3.10.3 Member Function Documentation

3.10.3.1 sendBool()

```
void UsartDebug::sendBool (
            bool data )
```

Send a boolean data on USART link.

This functions sends the requested boolean on USART link by calling driver's transmission function. The boolean data is first converted into a string and then sent

Parameters

in	*data*	boolean data to be sent

Returns

Nothing

Definition at line 42 of file log.cpp.

Here is the call graph for this function:

3.10.3.2 sendData()

```
void UsartDebug::sendData (
            char * str )
```

Send a string on USART link.

This functions sends the requested string on USART link by calling driver's transmission function

Parameters

in	*str*	Pointer to the string being sent

Returns

Nothing

Definition at line 22 of file log.cpp.

Here is the call graph for this function:

3.10.3.3 sendInteger()

```
void UsartDebug::sendInteger (
            uint16_t data,
            uint8_t base )
```

Send a integer data on USART link.

This functions sends the requested integer on USART link by calling driver's transmission function. The integer is first converted into a string and then sent

Parameters

in	*data*	integer data to be sent
in	*base*	numerical base used to convert integer into string (between 2 and 36)

Returns

Nothing

Definition at line 28 of file log.cpp.

Here is the call graph for this function:

The documentation for this class was generated from the following files:

- work/asw/log/log.h
- work/asw/log/log.cpp

Chapter 4

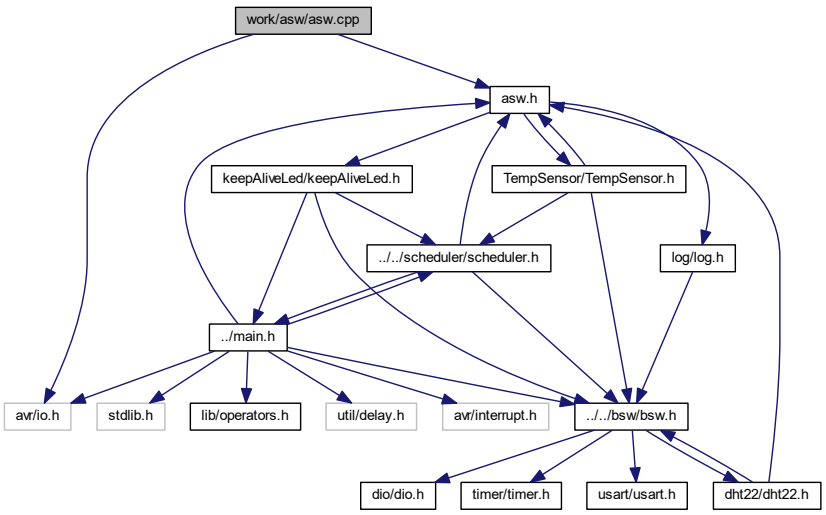
File Documentation

4.1 work/asw/asw.cpp File Reference

ASW main file.

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

Include dependency graph for asw.cpp:



Functions

- void asw_init ()

  Initialization of ASW.

Variables

- T_ASW_cnf_struct ASW_cnf_struct

4.1.1 Detailed Description

ASW main file.

Date

15 mars 2018

Author

nicls67

4.1.2 Function Documentation

4.1.2.1 asw_init()

```
void asw_init ( )
```

Initialization of ASW.

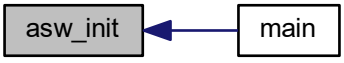
This function instantiates all applicative objects. The addresses of objects are then stored in ASW_cnf_struct structure. This function shall be called after BSW initialization function.

Returns

Nothing

Definition at line 20 of file asw.cpp.

Here is the caller graph for this function:



4.1.3 Variable Documentation

4.1.3.1 ASW_cnf_struct

```
T_ASW_cnf_struct ASW_cnf_struct
```

ASW configuration structure

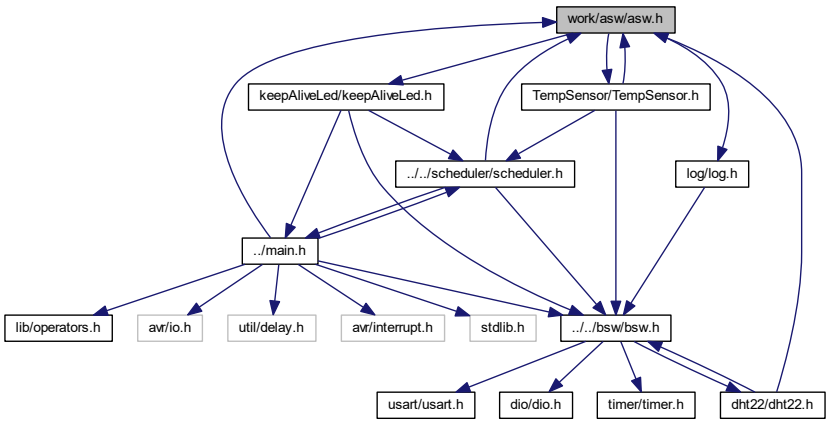
Definition at line 17 of file asw.cpp.

4.2 work/asw/asw.h File Reference

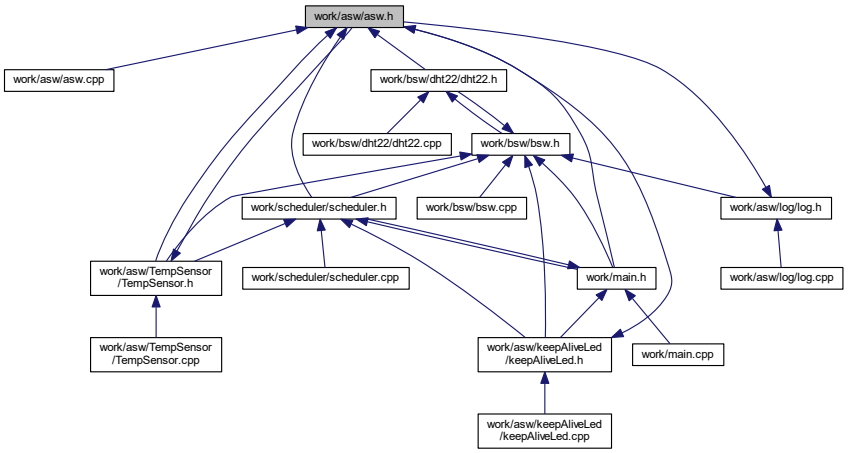
ASW main header file.

```
#include "log/log.h"
#include "keepAliveLed/keepAliveLed.h"
#include "TempSensor/TempSensor.h"
```

Include dependency graph for asw.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct T_ASW_cnf_struct

  ASW configuration structure.

Functions

- void asw_init ()

  Initialization of ASW.

Variables

- T_ASW_cnf_struct ASW_cnf_struct

4.2.1 Detailed Description

ASW main header file.

Date

15 mars 2018

Author

nicls67

4.2.2 Function Documentation

4.2.2.1 asw_init()

```
void asw_init ( )
```

Initialization of ASW.

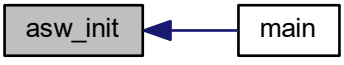
This function instantiates all applicative objects. The addresses of objects are then stored in ASW_cnf_struct structure. This function shall be called after BSW initialization function.

Returns

Nothing

Definition at line 20 of file asw.cpp.

Here is the caller graph for this function:



4.2.3 Variable Documentation

4.2.3.1 ASW_cnf_struct

```
T_ASW_cnf_struct ASW_cnf_struct
```

ASW configuration structure

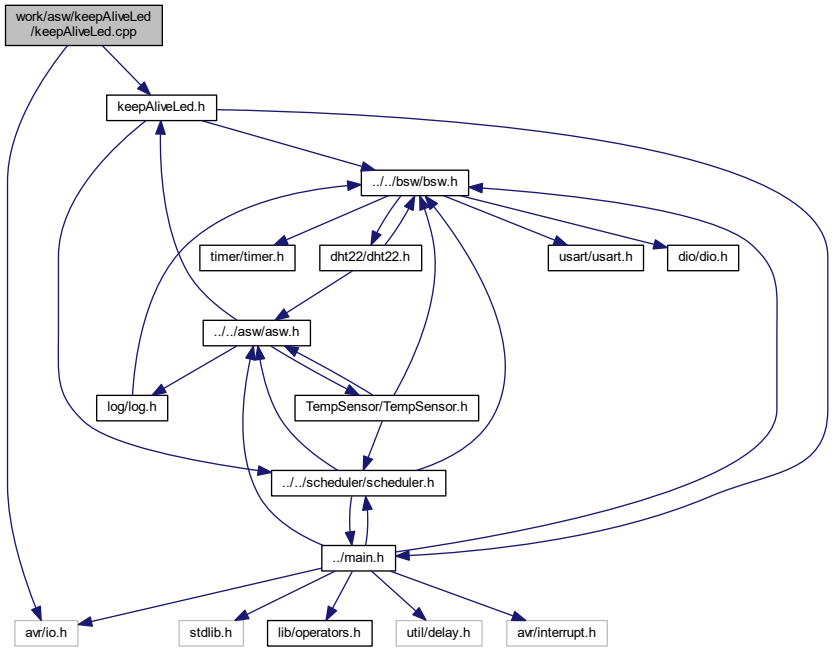
Definition at line 17 of file asw.cpp.

4.3 work/asw/keepAliveLed/keepAliveLed.cpp File Reference

Definition of function for class keepAliveLed.

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

Include dependency graph for keepAliveLed.cpp:



4.3.1 Detailed Description

Definition of function for class keepAliveLed.

Date

17 mars 2018

Author

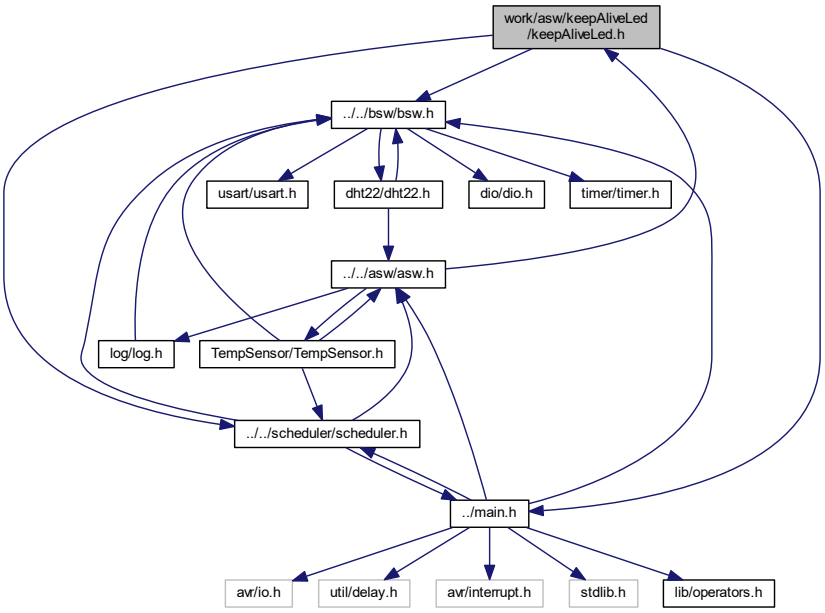
nicls67

4.4 work/asw/keepAliveLed/keepAliveLed.h File Reference

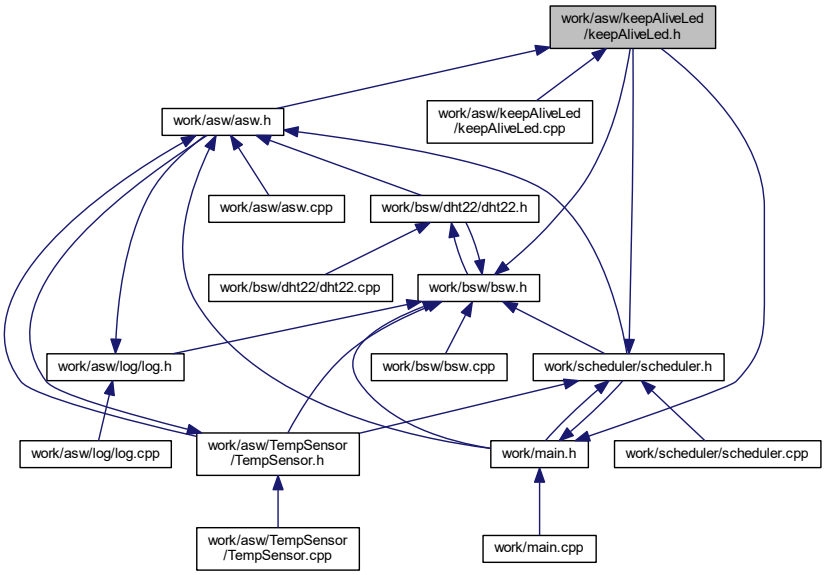
Class keepAliveLed header file.

```
#include "../../bsw/bsw.h"
#include "../../scheduler/scheduler.h"
#include "../../main.h"
```

Include dependency graph for keepAliveLed.h:



This graph shows which files directly or indirectly include this file:



Classes

- class keepAliveLed

  Class for keep-alive LED blinking.

Macros

- #define PERIOD_MS_TASK_LED SW_PERIOD_MS

4.4.1 Detailed Description

Class keepAliveLed header file.

Date

17 mars 2018

Author

nicls67

4.4.2 Macro Definition Documentation

4.4.2.1 PERIOD_MS_TASK_LED

```
#define PERIOD_MS_TASK_LED SW_PERIOD_MS
```

Period for led blinking

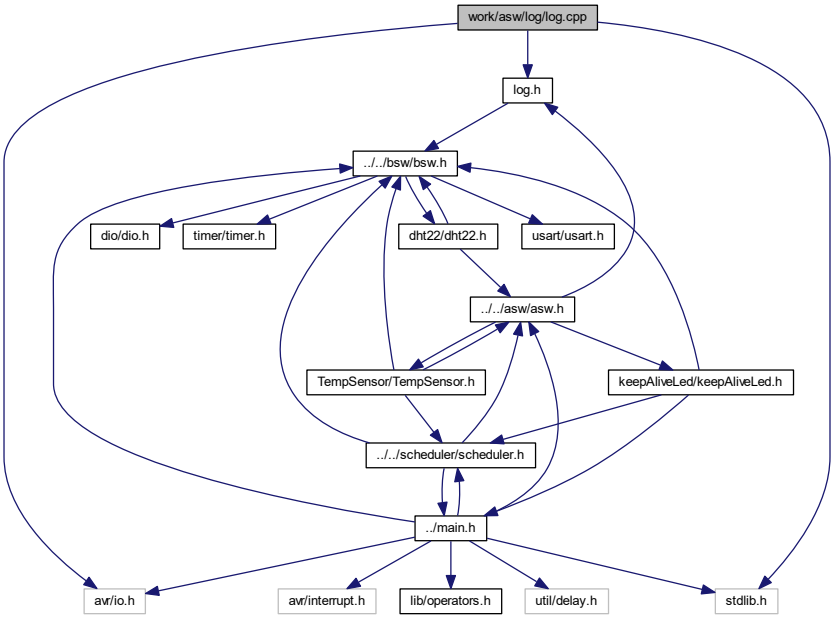
Definition at line 15 of file keepAliveLed.h.

4.5 work/asw/log/log.cpp File Reference

This file defines classes for log and debug data transmission on USART link.

```
#include <avr/io.h>
#include <stdlib.h>
#include "log.h"
```

Include dependency graph for log.cpp:



4.5.1 Detailed Description

This file defines classes for log and debug data transmission on USART link.

Date

15 mars 2018

Author

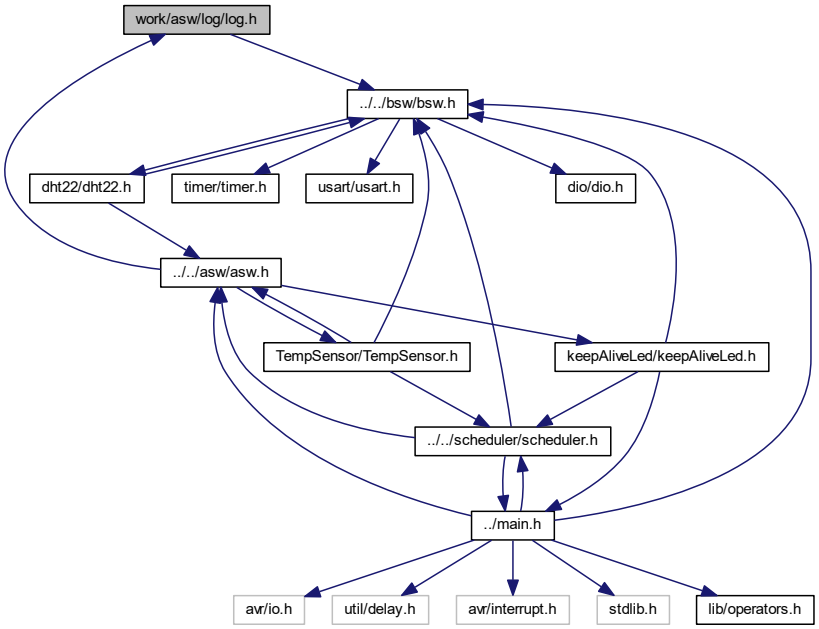
nicls67

4.6 work/asw/log/log.h File Reference

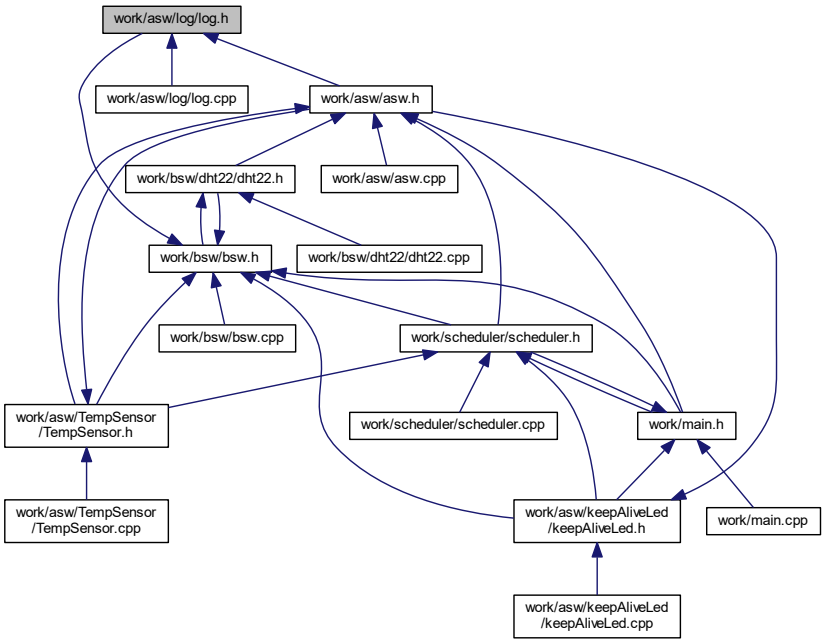
Header file for debug and logging functions.

```
#include "../../bsw/bsw.h"
```

Include dependency graph for log.h:



This graph shows which files directly or indirectly include this file:



Classes

- class UsartDebug

  Class used for debugging on usart link.

4.6.1 Detailed Description

Header file for debug and logging functions.

Date

15 mars 2018

Author

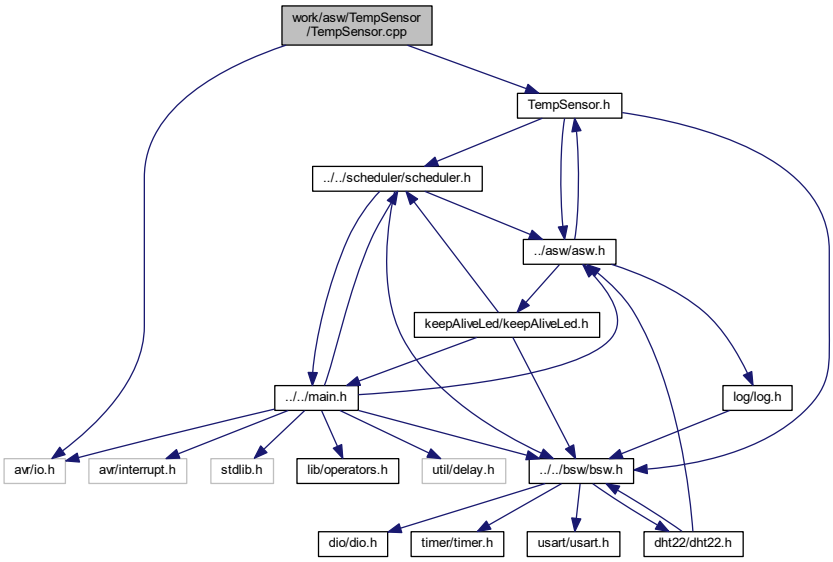
nicls67

4.7 work/asw/TempSensor/TempSensor.cpp File Reference

Defines function of class TempSensor.

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

Include dependency graph for TempSensor.cpp:



4.7.1 Detailed Description

Defines function of class TempSensor.

Date

23 mars 2018

Author

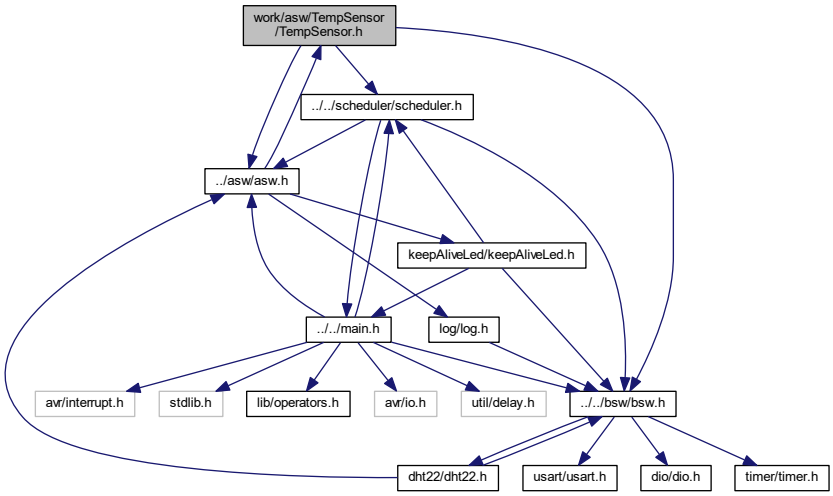
nicls67

4.8 work/asw/TempSensor/TempSensor.h File Reference

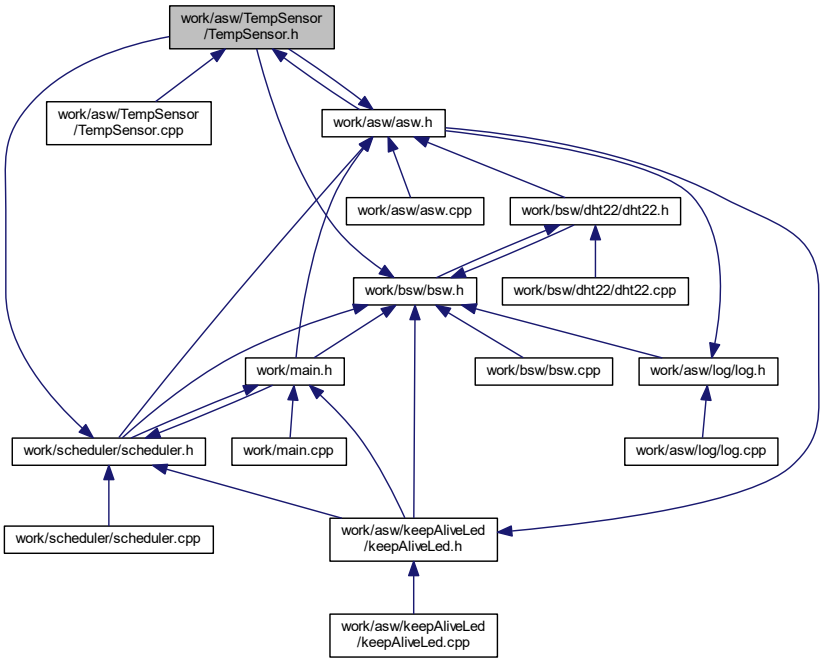
Class TempSensor header file.

```
#include "../../scheduler/scheduler.h"
#include "../../bsw/bsw.h"
#include "../asw.h"
```

Include dependency graph for TempSensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TempSensor

Macros

- #define PERIOD_MS_TASK_TEMP_SENSOR 5000

4.8.1 Detailed Description

Class TempSensor header file.

Date

23 mars 2018

Author

nicls67

4.8.2 Macro Definition Documentation

4.8.2.1 PERIOD_MS_TASK_TEMP_SENSOR

```
#define PERIOD_MS_TASK_TEMP_SENSOR 5000
```

Period for reading temperature data

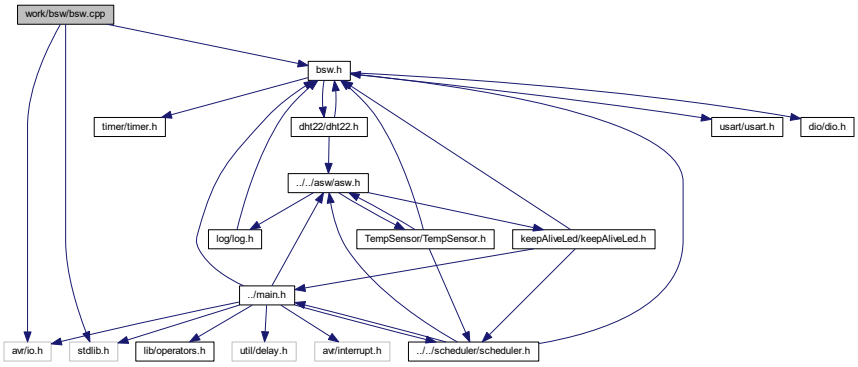
Definition at line 13 of file TempSensor.h.

4.9 work/bsw/bsw.cpp File Reference

BSW main file.

```
#include <avr/io.h>
#include <stdlib.h>
#include "bsw.h"
```

Include dependency graph for bsw.cpp:



Functions

- void bsw_init ()

  Initialization of BSW.

Variables

- T_BSW_cnf_struct BSW_cnf_struct

4.9.1 Detailed Description

BSW main file.

Date

13 mars 2018

Author

nicls67

4.9.2 Function Documentation

4.9.2.1 bsw_init()

```
void bsw_init ( )
```

Initialization of BSW.

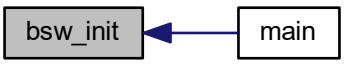
This function instantiates all driver objects, leading hardware initialization. The addresses of driver objects are then stored in BSW_cnf_struct structure.

Returns

Nothing

Definition at line 18 of file bsw.cpp.

Here is the caller graph for this function:



4.9.3 Variable Documentation

4.9.3.1 BSW_cnf_struct

```
T_BSW_cnf_struct BSW_cnf_struct
```

BSW configuration structure

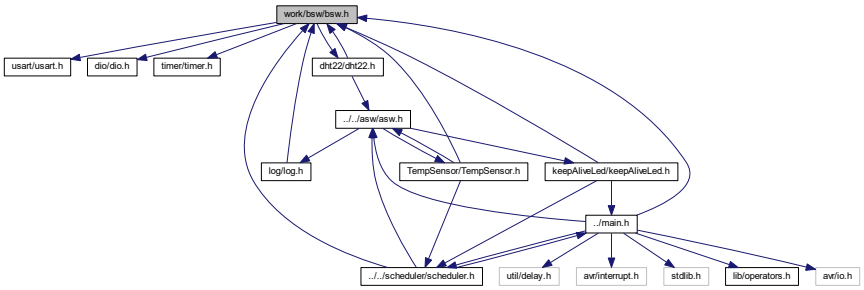
Definition at line 16 of file bsw.cpp.

4.10 work/bsw/bsw.h File Reference

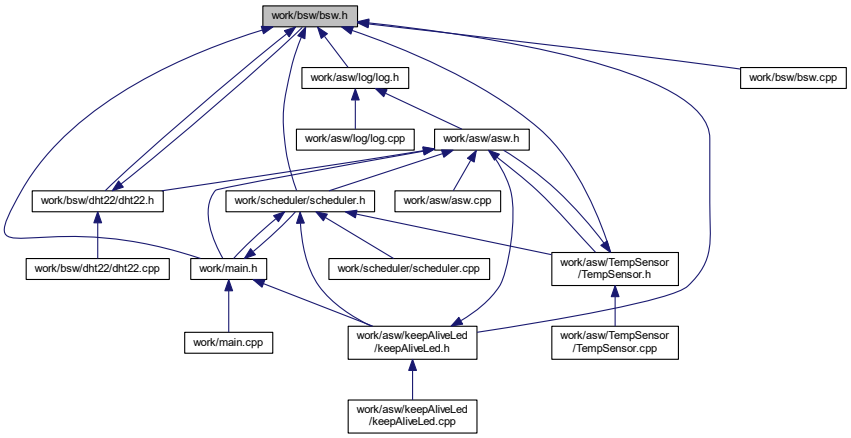
BSW main header file.

```
#include "usart/usart.h"
#include "dio/dio.h"
#include "timer/timer.h"
#include "dht22/dht22.h"
```

Include dependency graph for bsw.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct T_BSW_cnf_struct

  BSW configuration structure.

Macros

- #define USART_BAUDRATE (uint16_t)9600

Functions

- void bsw_init ()

  Initialization of BSW.

Variables

- T_BSW_cnf_struct BSW_cnf_struct

4.10.1 Detailed Description

BSW main header file.

Date

13 mars 2018

Author

nicls67

4.10.2 Macro Definition Documentation

4.10.2.1 USART_BAUDRATE

```
#define USART_BAUDRATE (uint16_t)9600
```

usart connection to PC uses a baud rate of 9600

Definition at line 23 of file bsw.h.

4.10.3 Function Documentation

4.10.3.1 bsw_init()

```
void bsw_init ( )
```

Initialization of BSW.

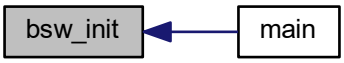
This function instantiates all driver objects, leading hardware initialization. The addresses of driver objects are then stored in BSW_cnf_struct structure.

Returns

Nothing

Definition at line 18 of file bsw.cpp.

Here is the caller graph for this function:



4.10.4 Variable Documentation

4.10.4.1 BSW_cnf_struct

```
T_BSW_cnf_struct BSW_cnf_struct
```

BSW configuration structure

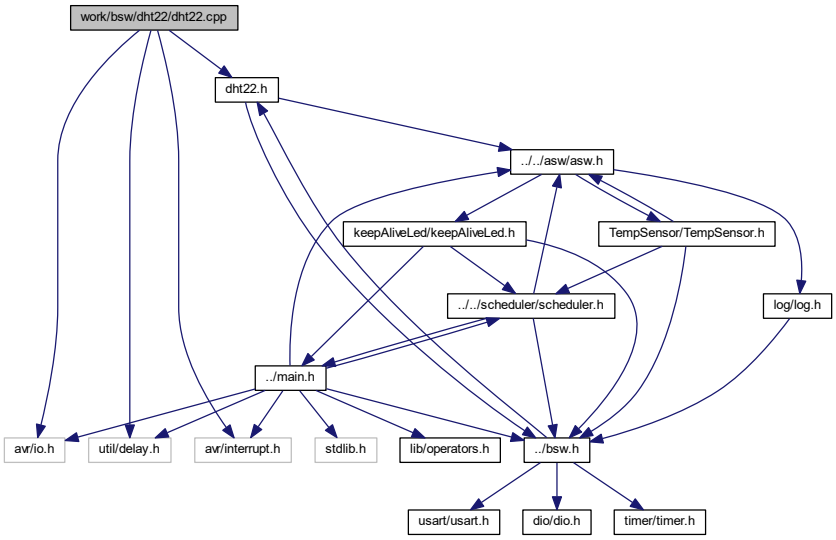
Definition at line 16 of file bsw.cpp.

4.11 work/bsw/dht22/dht22.cpp File Reference

This file defines classes for DHT22 driver.

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "dht22.h"
```

Include dependency graph for dht22.cpp:



Macros

- #define MAX_WAIT_TIME_US 100

4.11.1 Detailed Description

This file defines classes for DHT22 driver.

Date

23 mars 2018

Author

nicls67

4.11.2 Macro Definition Documentation

4.11.2.1 MAX_WAIT_TIME_US

```
#define MAX_WAIT_TIME_US 100
```

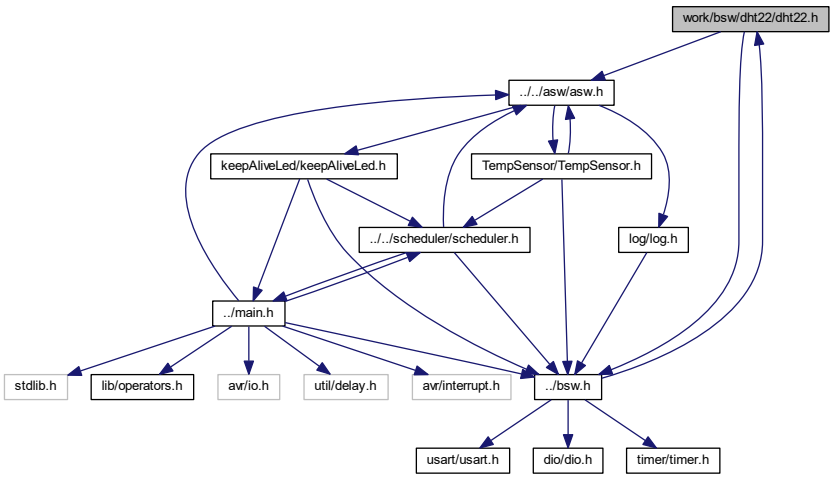
Definition at line 20 of file dht22.cpp.

4.12 work/bsw/dht22/dht22.h File Reference

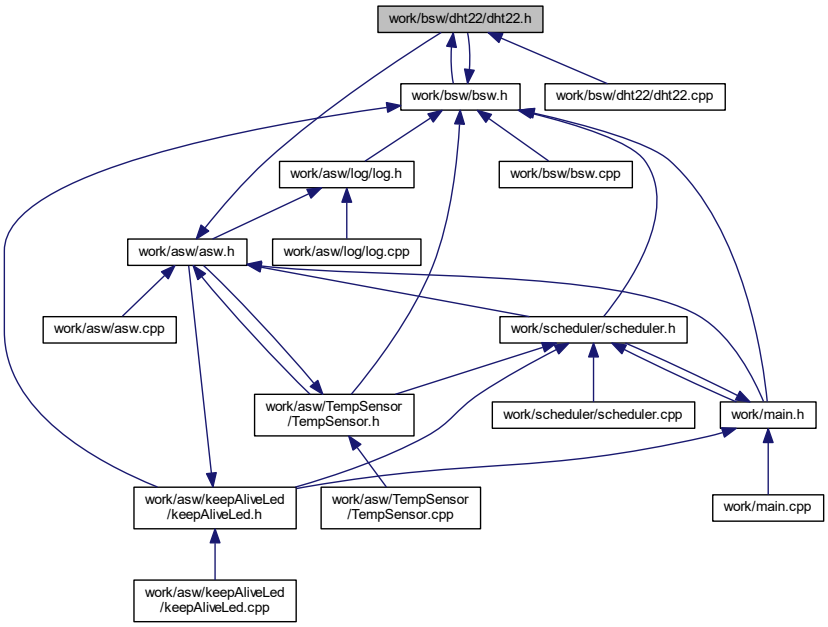
DHT22 driver header file.

```
#include "../bsw.h"
#include "../../asw/asw.h"
```

Include dependency graph for dht22.h:



This graph shows which files directly or indirectly include this file:



Classes

- class dht22

  DHT 22 driver class.

Macros

- #define DHT22_PORT PB6

4.12.1 Detailed Description

DHT22 driver header file.

Date

23 mars 2018

Author

nicls67

4.12.2 Macro Definition Documentation

4.12.2.1 DHT22_PORT

```
#define DHT22_PORT PB6
```

DHT22 is connected to port PB6

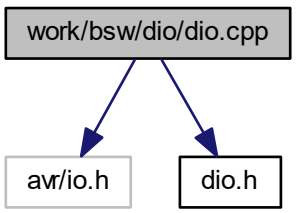
Definition at line 16 of file dht22.h.

4.13 work/bsw/dio/dio.cpp File Reference

DIO library.

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

Include dependency graph for dio.cpp:



Macros

- #define PORTB_CNF_DDRB (uint8_t)0b11000000
  Defines the configuration of DDRB register.
- #define PORTB_CNF_PORTB (uint8_t)0b11000000
  Defines the configuration of PORTB register.

4.13.1 Detailed Description

DIO library.

Date

13 mars 2018

Author

nicls67

4.13.2 Macro Definition Documentation

4.13.2.1 PORTB_CNF_DDRB

```
#define PORTB_CNF_DDRB (uint8_t)0b11000000
```

Defines the configuration of DDRB register.

This constant defines the direction of IO pins of PORT B. It will configure register DDRB.
PB0 : N/A
PB1 : N/A
PB2 : N/A
PB3 : N/A
PB4 : N/A
PB5 : N/A
PB6 : OUT
PB7 : OUT

Definition at line 26 of file dio.cpp.

4.13.2.2 PORTB_CNF_PORTB

```
#define PORTB_CNF_PORTB (uint8_t)0b11000000
```

Defines the configuration of PORTB register.

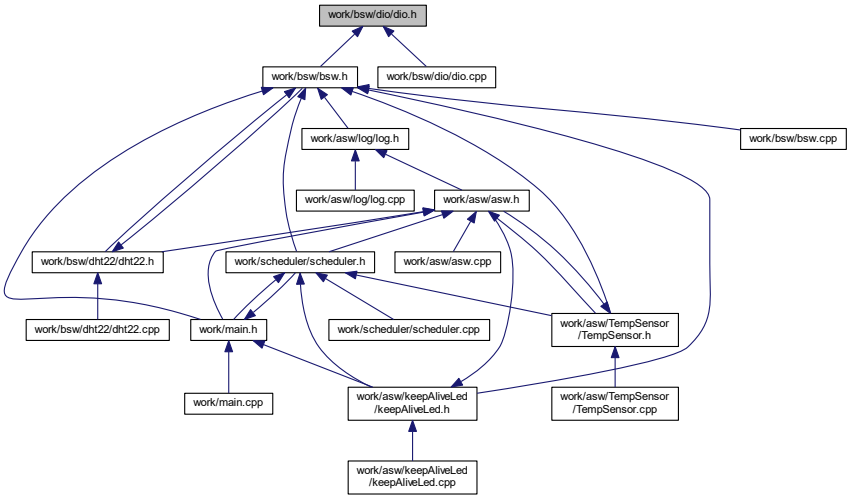
This constant defines the initial value of IO pins for PORT B. It will configure register PORTB. Pins configured as input shall not be configured here.
PB0 : N/A
PB1 : N/A
PB2 : N/A
PB3 : N/A
PB4 : N/A
PB5 : N/A
PB6 : HIGH
PB7 : HIGH

Definition at line 41 of file dio.cpp.

4.14 work/bsw/dio/dio.h File Reference

DIO library header file.

This graph shows which files directly or indirectly include this file:



Classes

- class dio

  DIO class.

Macros

- #define PORT_CNF_OUT 1
- #define PORT_CNF_IN 0

4.14.1 Detailed Description

DIO library header file.

Date

13 mars 2018

Author

nicls67

4.14.2 Macro Definition Documentation

4.14.2.1 PORT_CNF_IN

```
#define PORT_CNF_IN 0
```

Definition at line 15 of file dio.h.

4.14.2.2 PORT_CNF_OUT

```
#define PORT_CNF_OUT 1
```

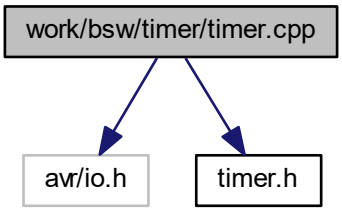
Definition at line 14 of file dio.h.

4.15 work/bsw/timer/timer.cpp File Reference

Defines function for class timer.

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

Include dependency graph for timer.cpp:



4.15.1 Detailed Description

Defines function for class timer.

Date

15 mars 2018

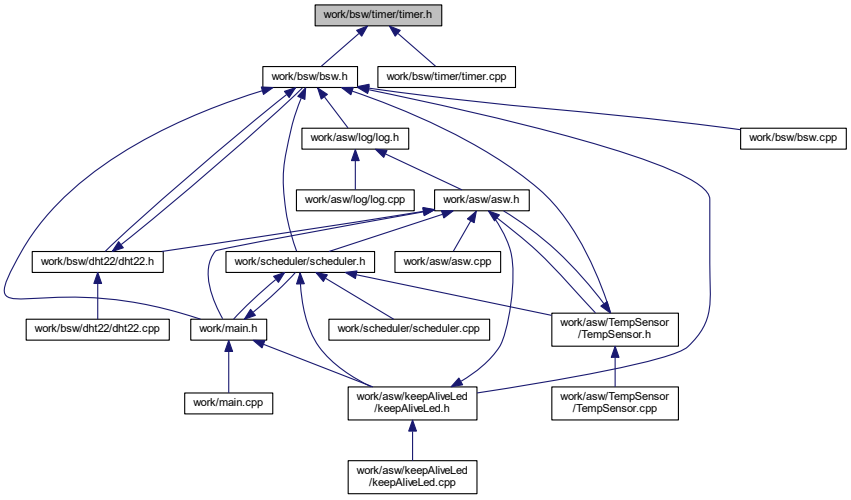
Author

nicls67

4.16 work/bsw/timer/timer.h File Reference

Timer class header file.

This graph shows which files directly or indirectly include this file:



Classes

- class timer

  Class defining a timer.

4.16.1 Detailed Description

Timer class header file.

Date

15 mars 2018

Author

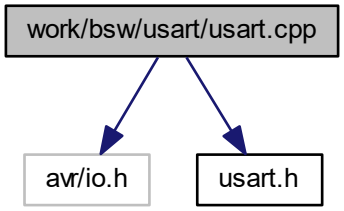
nicls67

4.17 work/bsw/usart/usart.cpp File Reference

BSW library for USART.

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

Include dependency graph for usart.cpp:



4.17.1 Detailed Description

BSW library for USART.

Date

13 mars 2018

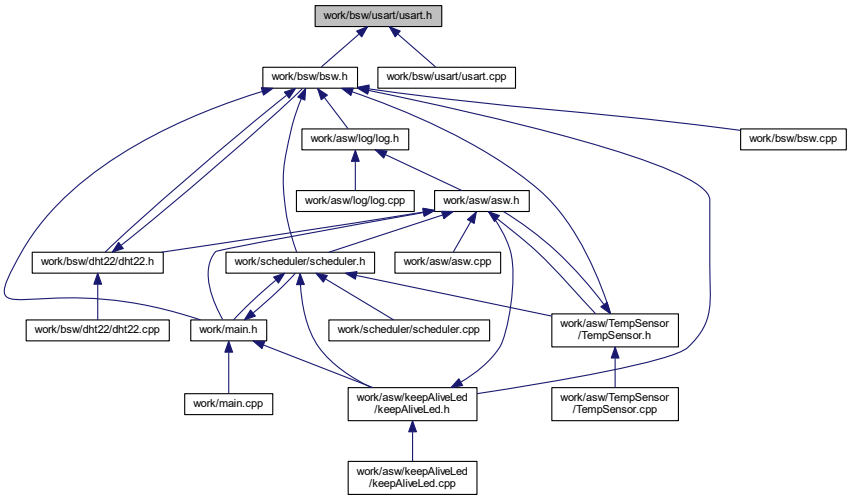
Author

nicls67

4.18 work/bsw/usart/usart.h File Reference

Header file for USART library.

This graph shows which files directly or indirectly include this file:



Classes

- class usart

  USART serial bus class.

4.18.1 Detailed Description

Header file for USART library.

Date

13 mars 2018

Author

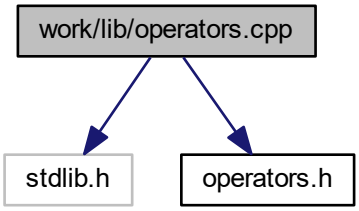
nicls67

4.19 work/lib/operators.cpp File Reference

c++ operators definitions

```
#include <stdlib.h>
#include "operators.h"
```

Include dependency graph for operators.cpp:



Functions

- void * operator new (size_t a_size)

  Operator new.
- void operator delete (void *ptr)

  Operator delete.

4.19.1 Detailed Description

c++ operators definitions

Date

14 mars 2018

Author

nicls67

4.19.2 Function Documentation

4.19.2.1 operator delete()

```
void operator delete (
            void * ptr )
```

Operator delete.

Equivalent to free function in C Free the memory zone at address ptr

Parameters

in	*ptr*	Pointer to the start of memory zone to free

Returns

Nothing

Definition at line 18 of file operators.cpp.

4.19.2.2 operator new()

```
void* operator new (
            size_t a_size )
```

Operator new.

Equivalent to malloc function in C Allocates a memory zone of size a_size

Parameters

in	*a_size*	memory size to allocate

Returns

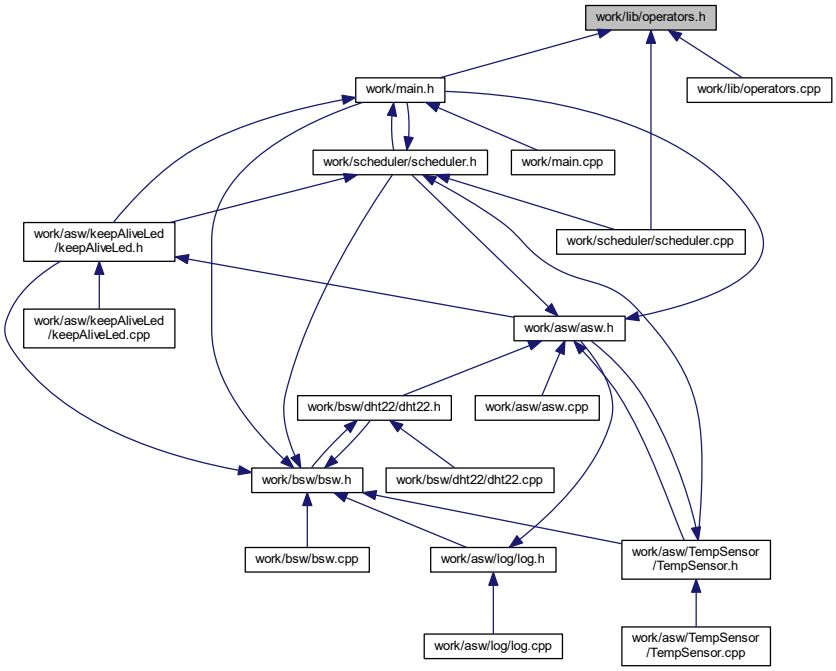
Pointer to the start of allocated memory zone

Definition at line 13 of file operators.cpp.

4.20 work/lib/operators.h File Reference

c++ operators definitions header file

This graph shows which files directly or indirectly include this file:



Functions

- void ∗ operator new (size_t a_size)

  Operator new.
- void operator delete (void ∗ptr)

  Operator delete.

4.20.1 Detailed Description

c++ operators definitions header file

Date

14 mars 2018

Author

nicls67

4.20.2 Function Documentation

4.20.2.1 operator delete()

```
void operator delete (
             void * ptr )
```

Operator delete.

Equivalent to free function in C Free the memory zone at address ptr

Parameters

in	*ptr*	Pointer to the start of memory zone to free

Returns

Nothing

Definition at line 18 of file operators.cpp.

4.20.2.2 operator new()

```
void* operator new (
            size_t a_size )
```

Operator new.

Equivalent to malloc function in C Allocates a memory zone of size a_size

Parameters

in	*a_size*	memory size to allocate

Returns

Pointer to the start of allocated memory zone

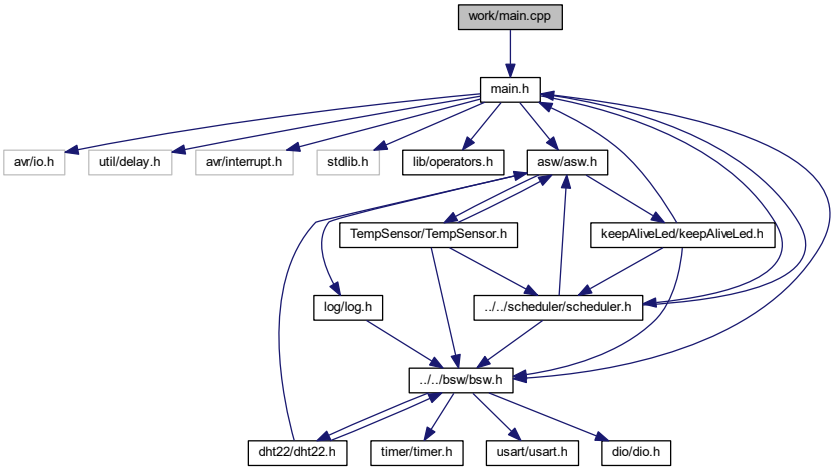
Definition at line 13 of file operators.cpp.

4.21 work/main.cpp File Reference

Background task file.

```
#include "main.h"
```

Include dependency graph for main.cpp:



Functions

- ISR (TIMER1_COMPA_vect)

  Main software interrupt.
- int main (void)

  Background task of program.

4.21.1 Detailed Description

Background task file.

Date

12 mars 2018

Author

nicls67

4.21.2 Function Documentation

4.21.2.1 ISR()

```
ISR (
            TIMER1_COMPA_vect  )
```

Main software interrupt.

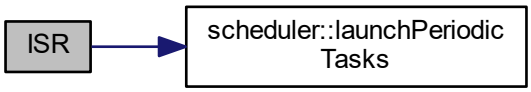
This function handles the interrupt raised by Timer #1. It wakes up the software every 500 ms to perform applications.

Returns

Nothing

Definition at line 19 of file main.cpp.

Here is the call graph for this function:



4.21.2.2 main()

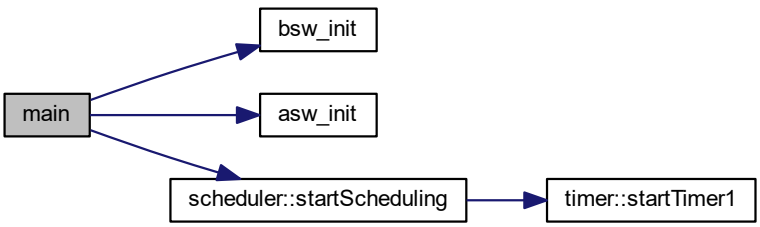
```
int main (
            void  )
```

Background task of program.

This function initializes all the software and then goes into an infinite loop. Periodic interrupt will wake up the software to perform application

Definition at line 29 of file main.cpp.

Here is the call graph for this function:

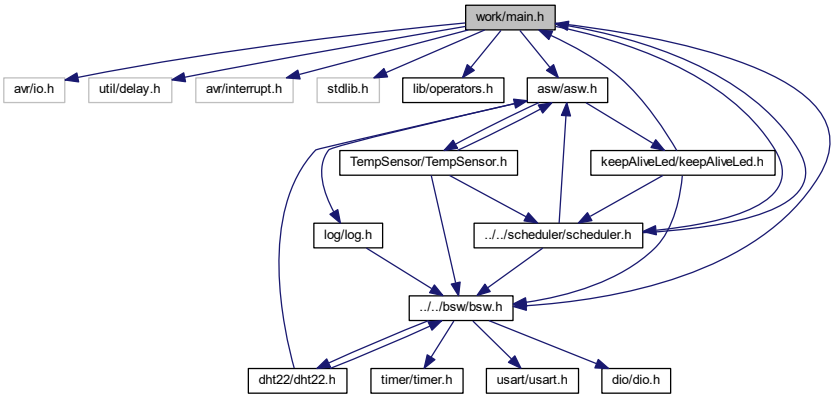


4.22 work/main.h File Reference

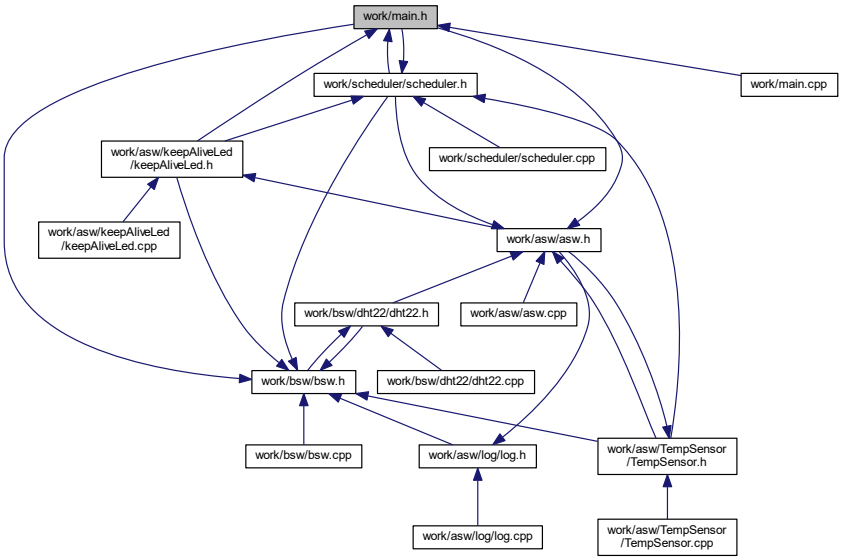
Background task header file.

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdlib.h>
#include "lib/operators.h"
#include "asw/asw.h"
#include "bsw/bsw.h"
#include "scheduler/scheduler.h"
```

Include dependency graph for main.h:



This graph shows which files directly or indirectly include this file:



4.22.1 Detailed Description

Background task header file.

Date

17 mars 2018

Author

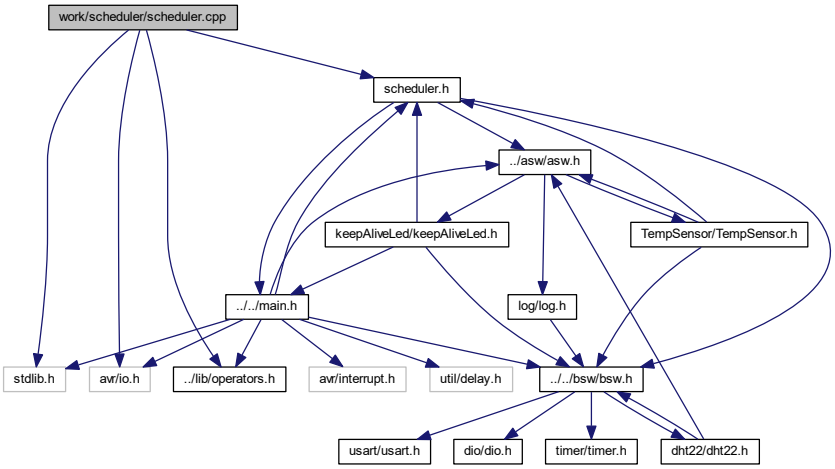
nicls67

4.23 work/scheduler/scheduler.cpp File Reference

Defines scheduler class.

```
#include <stdlib.h>
#include <avr/io.h>
#include "../lib/operators.h"
#include "scheduler.h"
```

Include dependency graph for scheduler.cpp:



Variables

- scheduler * p_scheduler

4.23.1 Detailed Description

Defines scheduler class.

Date

16 mars 2018

Author

nicls67

4.23.2 Variable Documentation

4.23.2.1 p_scheduler

```
scheduler* p_scheduler
```

Pointer to scheduler object

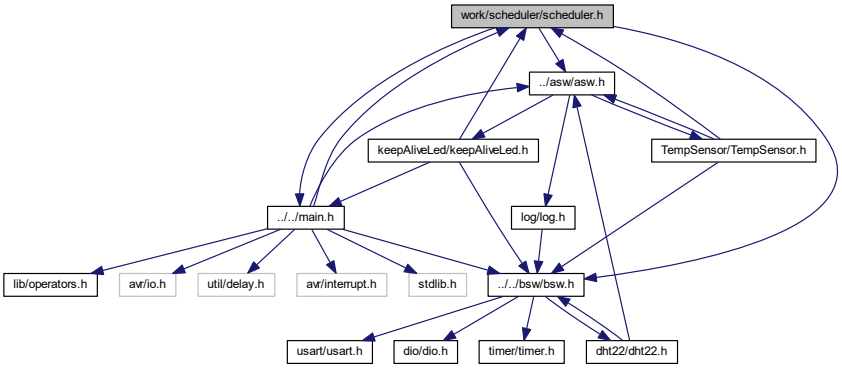
Definition at line 17 of file scheduler.cpp.

4.24 work/scheduler/scheduler.h File Reference

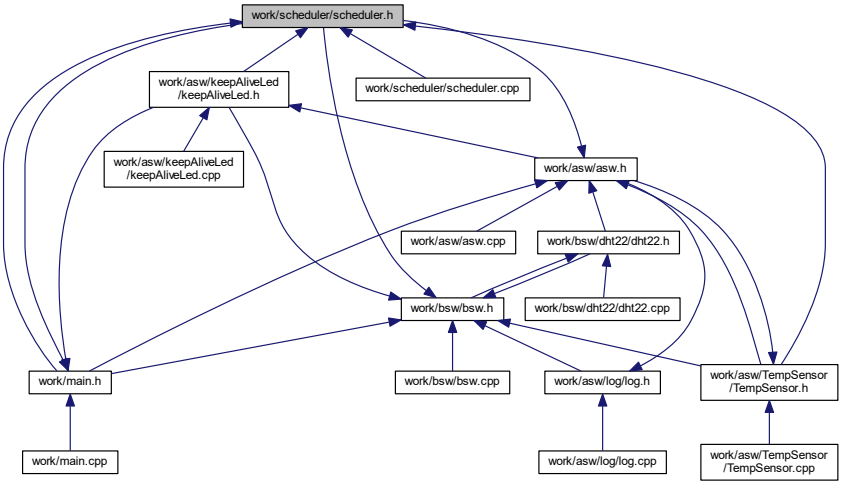
Scheduler class header file.

```
#include "../asw/asw.h"
#include "../bsw/bsw.h"
#include "../main.h"
```

Include dependency graph for scheduler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class scheduler

  Scheduler class.

Macros

- #define SW_PERIOD_MS 500
- #define PRESCALER_PERIODIC_TIMER 256
- #define TIMER_CTC_VALUE ((F_CPU/PRESCALER_PERIODIC_TIMER)/(1000/SW_PERIOD_MS))

Typedefs

- typedef void(∗ TaskPtr_t) (void)

  Type defining a pointer to function.

Variables

- scheduler ∗ p_scheduler

4.24.1 Detailed Description

Scheduler class header file.

Date

16 mars 2018

Author

nicls67

4.24.2 Macro Definition Documentation

4.24.2.1 PRESCALER_PERIODIC_TIMER

```
#define PRESCALER_PERIODIC_TIMER 256
```

Value of prescaler to use for periodic timer

Definition at line 19 of file scheduler.h.

4.24.2.2 SW_PERIOD_MS

```
#define SW_PERIOD_MS 500
```

Software period, used to define periodic timer interrupt

Definition at line 18 of file scheduler.h.

4.24.2.3 TIMER_CTC_VALUE

```
#define TIMER_CTC_VALUE ((F_CPU/PRESCALER_PERIODIC_TIMER)/(1000/SW_PERIOD_MS))
```

Compare value for periodic timer

Definition at line 20 of file scheduler.h.

4.24.3 Typedef Documentation

4.24.3.1 TaskPtr_t

```
typedef void(* TaskPtr_t) (void)
```

Type defining a pointer to function.

Definition at line 25 of file scheduler.h.

4.24.4 Variable Documentation

4.24.4.1 p_scheduler

```
scheduler* p_scheduler
```

Pointer to scheduler object

Definition at line 17 of file scheduler.cpp.

Index

ASW_cnf_struct
    asw.cpp, 32
    asw.h, 35
addPeriodicTask
    scheduler, 13
asw.cpp
    ASW_cnf_struct, 32
    asw_init, 32
asw.h
    ASW_cnf_struct, 35
    asw_init, 34
asw_init
    asw.cpp, 32
    asw.h, 34

BSW_cnf_struct
    bsw.cpp, 44
    bsw.h, 47
blinkLed_task
    keepAliveLed, 11
bsw.cpp
    BSW_cnf_struct, 44
    bsw_init, 44
bsw.h
    BSW_cnf_struct, 47
    bsw_init, 46
    USART_BAUDRATE, 46
bsw_init
    bsw.cpp, 44
    bsw.h, 46

configureTimer1
    timer, 21

DHT22_PORT
    dht22.h, 50
dht22, 5
    dht22, 5
    read, 6
dht22.cpp
    MAX_WAIT_TIME_US, 48
dht22.h
    DHT22_PORT, 50
dio, 7
    dio, 7
    dio_changePortBPinCnf, 7
    dio_getPortB, 8
    dio_invertPortB, 8
    dio_setPortB, 9
dio.cpp
    PORTB_CNF_DDRB, 52
    PORTB_CNF_PORTB, 52
dio.h
    PORT_CNF_IN, 53
    PORT_CNF_OUT, 54
dio_changePortBPinCnf
    dio, 7
dio_getPortB
    dio, 8
dio_invertPortB
    dio, 8
dio_setPortB
    dio, 9

getPitNumber
    scheduler, 14

ISR
    main.cpp, 63

keepAliveLed, 10
    blinkLed_task, 11
    keepAliveLed, 10
keepAliveLed.h
    PERIOD_MS_TASK_LED, 37

launchPeriodicTasks
    scheduler, 14

MAX_WAIT_TIME_US
    dht22.cpp, 48
main
    main.cpp, 64
main.cpp
    ISR, 63
    main, 64

operator delete
    operators.cpp, 58
    operators.h, 60
operator new
    operators.cpp, 59
    operators.h, 62
operators.cpp
    operator delete, 58
    operator new, 59
operators.h
    operator delete, 60
    operator new, 62

p_TempSensor

T_ASW_cnf_struct, 16
p_dht22
T_BSW_cnf_struct, 18
p_dio
T_BSW_cnf_struct, 18
p_keepAliveLed
T_ASW_cnf_struct, 16
p_scheduler
scheduler.cpp, 67
scheduler.h, 69
p_timer
T_BSW_cnf_struct, 18
p_usart
T_BSW_cnf_struct, 18
p_usartDebug
T_ASW_cnf_struct, 17
PERIOD_MS_TASK_LED
keepAliveLed.h, 37
PERIOD_MS_TASK_TEMP_SENSOR
TempSensor.h, 42
PORT_CNF_IN
dio.h, 53
PORT_CNF_OUT
dio.h, 54
PORTB_CNF_DDRB
dio.cpp, 52
PORTB_CNF_PORTB
dio.cpp, 52
PRESCALER_PERIODIC_TIMER
scheduler.h, 68

read
dht22, 6
readTempSensor_task
TempSensor, 20

SW_PERIOD_MS
scheduler.h, 68
scheduler, 12
addPeriodicTask, 13
getPitNumber, 14
launchPeriodicTasks, 14
scheduler, 12
startScheduling, 14
scheduler.cpp
p_scheduler, 67
scheduler.h
p_scheduler, 69
PRESCALER_PERIODIC_TIMER, 68
SW_PERIOD_MS, 68
TIMER_CTC_VALUE, 69
TaskPtr_t, 69
sendBool
UsartDebug, 27
sendData
UsartDebug, 28
sendInteger
UsartDebug, 28
setBaudRate
usart, 24
startScheduling
scheduler, 14
startTimer1
timer, 22
stopTimer1
timer, 22

T_ASW_cnf_struct, 16
p_TempSensor, 16
p_keepAliveLed, 16
p_usartDebug, 17
T_BSW_cnf_struct, 17
p_dht22, 18
p_dio, 18
p_timer, 18
p_usart, 18
TIMER_CTC_VALUE
scheduler.h, 69
TaskPtr_t
scheduler.h, 69
TempSensor, 19
readTempSensor_task, 20
TempSensor, 19
TempSensor.h
PERIOD_MS_TASK_TEMP_SENSOR, 42
timer, 20
configureTimer1, 21
startTimer1, 22
stopTimer1, 22
timer, 21

USART_BAUDRATE
bsw.h, 46
usart, 23
setBaudRate, 24
usart, 23
usart_init, 24
usart_sendString, 25
usart_init
usart, 24
usart_sendString
usart, 25
UsartDebug, 26
sendBool, 27
sendData, 28
sendInteger, 28
UsartDebug, 27

work/asw/TempSensor/TempSensor.cpp, 40
work/asw/TempSensor/TempSensor.h, 41
work/asw/asw.cpp, 31
work/asw/asw.h, 33
work/asw/keepAliveLed/keepAliveLed.cpp, 35
work/asw/keepAliveLed/keepAliveLed.h, 36
work/asw/log/log.cpp, 38
work/asw/log/log.h, 38
work/bsw/bsw.cpp, 43
work/bsw/bsw.h, 45

work/bsw/dht22/dht22.cpp, 48
work/bsw/dht22/dht22.h, 49
work/bsw/dio/dio.cpp, 51
work/bsw/dio/dio.h, 53
work/bsw/timer/timer.cpp, 54
work/bsw/timer/timer.h, 55
work/bsw/usart/usart.cpp, 56
work/bsw/usart/usart.h, 56
work/lib/operators.cpp, 57
work/lib/operators.h, 59
work/main.cpp, 62
work/main.h, 65
work/scheduler/scheduler.cpp, 66
work/scheduler/scheduler.h, 67