

# Arduino Handmade Peripherals functions

0.1

Generated by Doxygen 1.8.6

Wed Mar 9 2016 13:43:04



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	port Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Constructor & Destructor Documentation . . . . .	6
3.1.2.1	port . . . . .	6
3.1.3	Member Function Documentation . . . . .	7
3.1.3.1	get_state . . . . .	7
3.1.3.2	invert . . . . .	7
3.1.3.3	set_input . . . . .	7
3.1.3.4	set_input_PUE . . . . .	7
3.1.3.5	set_output . . . . .	8
3.1.4	Member Data Documentation . . . . .	8
3.1.4.1	broche . . . . .	8
3.1.4.2	port_name . . . . .	8
3.2	serial Class Reference . . . . .	8
3.2.1	Detailed Description . . . . .	9
3.2.2	Constructor & Destructor Documentation . . . . .	9
3.2.2.1	serial . . . . .	9
3.2.3	Member Function Documentation . . . . .	9
3.2.3.1	data_received . . . . .	9
3.2.3.2	init . . . . .	9
3.2.3.3	read . . . . .	10
3.2.3.4	write . . . . .	10
<b>4</b>	<b>File Documentation</b>	<b>11</b>
4.1	headers/ports.h File Reference . . . . .	11

---

4.1.1	Detailed Description	11
4.2	headers/serial.h File Reference	12
4.3	main.cpp File Reference	12
4.3.1	Macro Definition Documentation	13
4.3.1.1	HIGH	13
4.3.1.2	high	13
4.3.1.3	LOW	13
4.3.1.4	low	13
4.3.2	Function Documentation	13
4.3.2.1	main	13
4.4	sources/ports.cpp File Reference	13
4.5	sources/serial.cpp File Reference	13
4.5.1	Macro Definition Documentation	14
4.5.1.1	BAUD	14
4.5.1.2	F_CPU	14
<b>Index</b>		<b>15</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">port</a>	This contains the attributs and methods for the port driver . . . . .	??
<a href="#">serial</a>	Class prototype for the serial communication driver . . . . .	??



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	.....	??
headers/ <a href="#">ports.h</a>		
Class prototype for the ports driver	.....	??
headers/ <a href="#">serial.h</a>	.....	??
sources/ <a href="#">ports.cpp</a>	.....	??
sources/ <a href="#">serial.cpp</a>	.....	??





## Chapter 3

# Class Documentation

### 3.1 port Class Reference

this contains the attributs and methods for the port driver

```
#include <ports.h>
```

#### Public Member Functions

- `port` (volatile uint8\_t &nom)  
*Constructor creates a new port instance.*
- void `set_output` (int pin, bool state)  
*sets the pin in DIGITAL OUTPUT MODE*
- void `set_input` (int pin)  
*sets the pin as INPUT MODE with pull-up disable*
- void `set_input_PUE` (int pin)  
*sets the pin as INPUT MODE PULL UP ENABLE*
- bool `get_state` (int pin)  
*gives the digital state(HIGH or LOW) of the pin*
- void `invert` (const int &time)  
*inverts the state of the specified PORT (DEFAULT STATE: 10101010)*

#### Protected Attributes

- volatile uint8\_t \* `port_name`  
*pointer to the io register PORTx (x{B,C,D})*
- bool `broche`  
*ensure the creation of a port instance*

#### 3.1.1 Detailed Description

this contains the attributs and methods for the port driver

Definition at line 20 of file ports.h.

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 `port::port ( volatile uint8_t & nom )`

Constructor creates a new port instance.

## Parameters

<i>&amp;nom</i>	the name of the port (PORTB,PORTC,PORTD)
-----------------	--

## Returns

nothing

Definition at line 19 of file ports.cpp.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 bool port::get\_state ( int *pin* )

gives the digital state(HIGH or LOW) of the pin

## Parameters

<i>pin</i>	the pin number of the port
------------	----------------------------

## Returns

Digital state of the pin (bool)

Definition at line 147 of file ports.cpp.

#### 3.1.3.2 void port::invert ( const int & *time* )

inverts the state of the specified PORT (DEFAULT STATE: 10101010)

## Parameters

<i>time</i>	gives the time between each state (in milliseconds)
-------------	---

## Returns

nothing

Definition at line 187 of file ports.cpp.

#### 3.1.3.3 void port::set\_input ( int *pin* )

sets the pin as INPUT MODE with pull-up disable

## Parameters

<i>pin</i>	the pin number of the port (PIN0..PIN7)
------------	---

## Returns

nothing

Definition at line 74 of file ports.cpp.

#### 3.1.3.4 void port::set\_input\_PUE ( int *pin* )

sets the pin as INPUT MODE PULL UP ENABLE

**Parameters**

<i>pin</i>	the pin number of the port
------------	----------------------------

**Returns**

nothing

Definition at line 111 of file ports.cpp.

**3.1.3.5 void port::set\_output ( int *pin*, bool *state* )**

sets the pin in DIGITAL OUTPUT MODE

**Parameters**

<i>pin</i>	the pin number of the port (PIN0..PIN7)
<i>state</i>	the pin state (HIGH or LOW)

**Returns**

nothing

Definition at line 23 of file ports.cpp.

**3.1.4 Member Data Documentation****3.1.4.1 bool port::broche [protected]**

ensure the creation of a port instance

Definition at line 83 of file ports.h.

**3.1.4.2 volatile uint8\_t\* port::port\_name [protected]**

pointer to the io register PORTx (x{B,C,D})

Definition at line 82 of file ports.h.

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

- headers/[ports.h](#)
- sources/[ports.cpp](#)

**3.2 serial Class Reference**

Class prototype for the serial communication driver.

```
#include <serial.h>
```

**Public Member Functions**

- [serial](#) ()  
    *Constructor.*
- void [init](#) ()

- USART interface initialize the interface.*
- void [write](#) (char &data)  
*Write data into the buffer.*
- char [read](#) ()  
*read data from the buffer*
- char [data\\_received](#) ()  
*Verify if the buffer has data to be read.*

### 3.2.1 Detailed Description

Class prototype for the serial communication driver.

file [serial.h](#)

#### Author

Alex Velásquez Meling No Known bugs

this contains the attributs and methods for the serial communication driver

Definition at line 17 of file serial.h.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 [serial::serial](#) ( )

Constructor.

#### Returns

nothing

Definition at line 26 of file serial.cpp.

### 3.2.3 Member Function Documentation

#### 3.2.3.1 [char serial::data\\_received](#) ( )

Verify if the buffer has data to be read.

#### Returns

true if data is received or false if it is not.

Definition at line 58 of file serial.cpp.

#### 3.2.3.2 [void serial::init](#) ( )

USART interface initialize the interface.

#### Returns

nothing

Definition at line 31 of file serial.cpp.

### 3.2.3.3 char serial::read ( )

read data from the buffer

#### Returns

data read

Definition at line 50 of file serial.cpp.

### 3.2.3.4 void serial::write ( char & *data* )

Write data into the buffer.

#### Parameters

<i>data</i>	data to be sent
-------------	-----------------

#### Returns

nothing

Definition at line 43 of file serial.cpp.

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

- headers/[serial.h](#)
- sources/[serial.cpp](#)

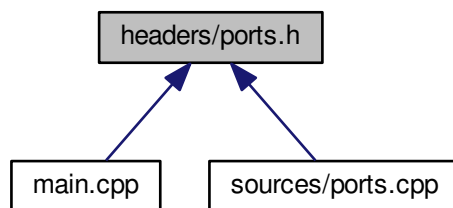
## Chapter 4

# File Documentation

### 4.1 headers/ports.h File Reference

Class prototype for the ports driver.

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



### Classes

- class `port`

*this contains the attributs and methods for the port driver*

#### 4.1.1 Detailed Description

Class prototype for the ports driver. This contains the class for the ports driver and also the attributs and methods you will need

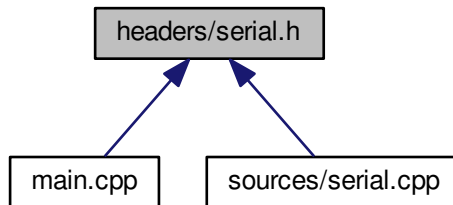
#### Author

Alex Velásquez Meling No known bugs

Definition in file `ports.h`.

## 4.2 headers/serial.h File Reference

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



### Classes

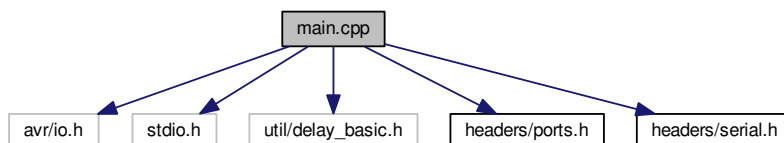
- class [serial](#)

*Class prototype for the serial communication driver.*

## 4.3 main.cpp File Reference

```
#include <avr/io.h>
#include <stdio.h>
#include <util/delay_basic.h>
#include "headers/ports.h"
#include "headers/serial.h"
```

Include dependency graph for main.cpp:



### Macros

- #define [HIGH](#) true
- #define [LOW](#) false
- #define [low](#) LOW
- #define [high](#) HIGH

### Functions

- int [main](#) (void)



### 4.3.1 Macro Definition Documentation

#### 4.3.1.1 `#define HIGH true`

Definition at line 7 of file main.cpp.

#### 4.3.1.2 `#define high HIGH`

Definition at line 10 of file main.cpp.

#### 4.3.1.3 `#define LOW false`

Definition at line 8 of file main.cpp.

#### 4.3.1.4 `#define low LOW`

Definition at line 9 of file main.cpp.

### 4.3.2 Function Documentation

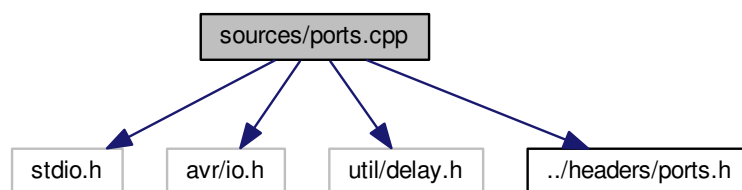
#### 4.3.2.1 `int main ( void )`

Definition at line 13 of file main.cpp.

## 4.4 sources/ports.cpp File Reference

```
#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>
#include "../headers/ports.h"
```

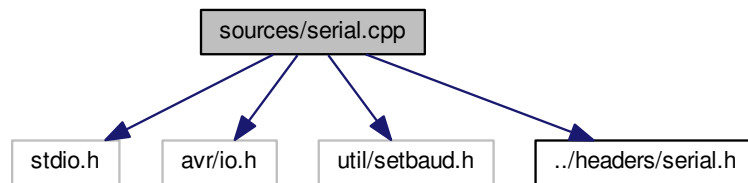
Include dependency graph for ports.cpp:



## 4.5 sources/serial.cpp File Reference

```
#include <stdio.h>
#include <avr/io.h>
#include <util/setbaud.h>
#include "../headers/serial.h"
```

Include dependency graph for serial.cpp:



## Macros

- `#define F_CPU 20000000L`
- `#define BAUD 9600`

### 4.5.1 Macro Definition Documentation

#### 4.5.1.1 `#define BAUD 9600`

Definition at line 14 of file `serial.cpp`.

#### 4.5.1.2 `#define F_CPU 20000000L`

Definition at line 13 of file `serial.cpp`.

# Index

BAUD  
    serial.cpp, [14](#)  
broche  
    port, [8](#)  
data\_received  
    serial, [9](#)  
F\_CPU  
    serial.cpp, [14](#)  
get\_state  
    port, [7](#)  
HIGH  
    main.cpp, [13](#)  
headers/ports.h, [11](#)  
headers/serial.h, [12](#)  
high  
    main.cpp, [13](#)  
init  
    serial, [9](#)  
invert  
    port, [7](#)  
LOW  
    main.cpp, [13](#)  
low  
    main.cpp, [13](#)  
main  
    main.cpp, [13](#)  
main.cpp, [12](#)  
    HIGH, [13](#)  
    high, [13](#)  
    LOW, [13](#)  
    low, [13](#)  
    main, [13](#)  
port, [5](#)  
    broche, [8](#)  
    get\_state, [7](#)  
    invert, [7](#)  
    port, [6](#)  
    port\_name, [8](#)  
    set\_input, [7](#)  
    set\_input\_PUE, [7](#)  
    set\_output, [8](#)  
port\_name  
    port, [8](#)

read  
    serial, [9](#)  
serial, [8](#)  
    data\_received, [9](#)  
    init, [9](#)  
    read, [9](#)  
    serial, [9](#)  
    write, [10](#)  
serial.cpp  
    BAUD, [14](#)  
    F\_CPU, [14](#)  
set\_input  
    port, [7](#)  
set\_input\_PUE  
    port, [7](#)  
set\_output  
    port, [8](#)  
sources/ports.cpp, [13](#)  
sources/serial.cpp, [13](#)  
write  
    serial, [10](#)