

AVR Digital IO Library  
1.0

Generated by Doxygen 1.7.5.1

Sun Jan 27 2013 11:49:05



# Contents

|          |                                       |          |
|----------|---------------------------------------|----------|
| <b>1</b> | <b>File Index</b>                     | <b>1</b> |
| 1.1      | File List . . . . .                   | 1        |
| <b>2</b> | <b>File Documentation</b>             | <b>3</b> |
| 2.1      | digital_io.c File Reference . . . . . | 3        |
| 2.1.1    | Function Documentation . . . . .      | 3        |
| 2.1.1.1  | __port_setup . . . . .                | 3        |
| 2.1.1.2  | argsToNumber . . . . .                | 4        |
| 2.2      | digital_io.c . . . . .                | 4        |
| 2.3      | digital_io.h File Reference . . . . . | 5        |
| 2.3.1    | Define Documentation . . . . .        | 6        |
| 2.3.1.1  | pin_input_hi . . . . .                | 6        |
| 2.3.1.2  | pin_input_lo . . . . .                | 6        |
| 2.3.1.3  | pin_output_hi . . . . .               | 6        |
| 2.3.1.4  | pin_output_lo . . . . .               | 6        |
| 2.3.1.5  | pin_read . . . . .                    | 7        |
| 2.3.1.6  | pin_write_hi . . . . .                | 7        |
| 2.3.1.7  | pin_write_lo . . . . .                | 7        |
| 2.3.1.8  | port_input_hi . . . . .               | 7        |
| 2.3.1.9  | port_input_lo . . . . .               | 7        |
| 2.3.1.10 | port_output_hi . . . . .              | 8        |
| 2.3.1.11 | port_output_lo . . . . .              | 8        |
| 2.3.1.12 | port_read . . . . .                   | 8        |
| 2.3.1.13 | port_write_hi . . . . .               | 8        |
| 2.3.1.14 | port_write_lo . . . . .               | 8        |
| 2.3.2    | Function Documentation . . . . .      | 9        |

|         |  |    |
|---------|--|----|
| 2.3.2.1 | <a href="#">__port_setup</a>           | 9  |
| 2.3.2.2 | <a href="#">argsToNumber</a>           | 9  |
| 2.4     | <a href="#">digital_io.h</a>           | 9  |
| 2.5     | <a href="#">main.c File Reference</a>  | 10 |
| 2.5.1   | <a href="#">Function Documentation</a> | 10 |
| 2.5.1.1 | <a href="#">main</a>                   | 10 |
| 2.6     | <a href="#">main.c</a>                 | 11 |

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

|                              |    |
|------------------------------|----|
| <a href="#">digital_io.c</a> | 3  |
| <a href="#">digital_io.h</a> | 5  |
| <a href="#">main.c</a>       | 10 |



## Chapter 2

# File Documentation

### 2.1 digital\_io.c File Reference

```
#include <avr/io.h> #include <stdarg.h> #include "digital-  
_io.h"
```

#### Functions

- `const uint8_t argsToNumber (const uint8_t n,...)`  
*ArgsToNumber berfungsi untuk merubah parameter list ... dengan jumlah uint8\_t n menjadi bentuk \_BV(item parameter list 1)|\_BV(\_BV(item parameter list 1))| dst.*
- `void __port_setup (volatile uint8_t *PORT, const uint8_t flag01, const uint8_t __PORT)`  
*berfungsi untuk mengatur PORT dan DDR fungsi ini tidak dirancang untuk digunakan secara langsung namun dibungkus melalui wrapper untuk menyederhanakan pemakaian*

#### 2.1.1 Function Documentation

2.1.1.1 `void __port_setup ( volatile uint8_t * PORT, const uint8_t flag0, const uint8_t __PORT )`

berfungsi untuk mengatur PORT dan DDR fungsi ini tidak dirancang untuk digunakan secara langsung namun dibungkus melalui wrapper untuk menyederhanakan pemakaian

##### Parameters

|               |                   |
|---------------|-------------------|
| <i>PORT</i>   | volatile uint8_t* |
| <i>flag0</i>  | const uint8_t     |
| <i>__PORT</i> | const uint8_t     |

**Returns**

void

Definition at line 25 of file [digital\\_io.c](#).**2.1.1.2 const uint8\_t argsToNumber ( const uint8\_t n, ... )**

ArgsToNumber berfungsi untuk merubah parameter list ... dengan jumlah uint8\_t n menjadi bentuk `_BV(item parameter list 1)|_BV(_BV(item parameter list 1))| dst.`

**Parameters**

|          |               |
|----------|---------------|
| <i>n</i> | const uint8_t |
| ...      |               |

**Returns**

const uint8\_t

Definition at line 10 of file [digital\\_io.c](#).**2.2 digital\_io.c**

```

00001 /*
00002  * kenzanin AVR Pin Input Output Driver
00003  * license LGPLv3
00004  * Copy-left kenzanin[at]gmail[dot]com
00005  */
00006 #include <avr/io.h>
00007 #include <stdarg.h>
00008 #include "digital_io.h"
00009
00010 const uint8_t argsToNumber(const uint8_t n,...)
00011 {
00012     uint8_t i=0;
00013     uint8_t ii=0;
00014
00015     va_list lst;
00016     va_start(lst,n);
00017     for(; i<n ; ++i)
00018     {
00019         ii|=_BV(va_arg ( lst, int));
00020     }
00021     va_end(lst);
00022     return ii;
00023 }
00024
00025 void __port_setup(volatile uint8_t* PORT,const uint8_t flag01,const uint8_t
__PORT)
00026 {
00027     /*
00028     flag01
00029     1  = DDR  output
00030     2  = DDR  input
00031     4  = PORT 1
00032     8  = PORT 0
00033     */
00034     uint8_t _port=*PORT;
00035     uint8_t _ddr =*(PORT-1);

```



```

00036
00037     if(flag01&0x01)
00038     {
00039         _ddr|=__PORT;
00040     }
00041     else if(flag01&0x02)
00042     {
00043         _ddr&=~(__PORT);
00044     }
00045
00046     if(flag01&0x04)
00047     {
00048         _port|=__PORT;
00049     }
00050     else if(flag01&0x08)
00051     {
00052         _port&=~(__PORT);
00053     }
00054     * (PORT)          =_port;
00055     * (PORT-1)       =_ddr;
00056 }

```

## 2.3 digital\_io.h File Reference

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

### Defines

- #define [port\\_output\\_hi](#)(x, y,...) \_\_port\_setup(&(x),5,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [port\\_output\\_lo](#)(x, y,...) \_\_port\_setup(&(x),9,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [port\\_input\\_hi](#)(x, y,...) \_\_port\_setup(&(x),6,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [port\\_input\\_lo](#)(x, y,...) \_\_port\_setup(&(x),10,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [pin\\_output\\_hi](#)(x, y) \_\_port\_setup(&(x),5,\_BV(y))
- #define [pin\\_output\\_lo](#)(x, y) \_\_port\_setup(&(x),9,\_BV(y))
- #define [pin\\_input\\_hi](#)(x, y) \_\_port\_setup(&(x),6,\_BV(y))
- #define [pin\\_input\\_lo](#)(x, y) \_\_port\_setup(&(x),10,\_BV(y))
- #define [port\\_write\\_hi](#)(x, y,...) \_\_port\_setup(&(x),4,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [port\\_write\\_lo](#)(x, y,...) \_\_port\_setup(&(x),8,argsToNumber((y),\_\_VA\_ARGS\_\_))
- #define [pin\\_write\\_hi](#)(x, y) sbi(x,y)
- #define [pin\\_write\\_lo](#)(x, y) cbi(x,y)
- #define [port\\_read](#)(x) \*(&(x)-2)
- #define [pin\\_read](#)(x, y) (\_Bool)\*(&(x)-2)&\_BV(y)

### Functions

- const uint8\_t [argsToNumber](#) (const uint8\_t n,...) \_\_attribute\_\_((pure))

*ArgsToNumber berfungsi untuk merubah parameter list ... dengan jumlah uint8\_t n menjadi bentuk \_BV(item parameter list 1)|\_BV(\_BV(item parameter list 1))| dst.*

- void `__port_setup` (volatile uint8\_t \*PORT, const uint8\_t flag0, const uint8\_t \_\_-PORT)

*berfungsi untuk mengatur PORT dan DDR fungsi ini tidak dirancang untuk digunakan secara langsung namun dibungkus melalui wrapper untuk menyederhanakan pemakaian*

### 2.3.1 Define Documentation

#### 2.3.1.1 #define pin\_input\_hi( x, y ) \_\_port\_setup(&(x),6,\_BV(y))

pin\_input\_hi adalah wrapper \_\_port\_setup, untuk inialisasi pin pada PORT,sebagai inout dengan kondisi awal 1 contoh pemakaian

```
pin_output_hi (PORTC, 2) ;
```

Definition at line 92 of file [digital\\_io.h](#).

#### 2.3.1.2 #define pin\_input\_lo( x, y ) \_\_port\_setup(&(x),10,\_BV(y))

pin\_input\_lo adalah wrapper \_\_port\_setup, untuk inialisasi pin pada PORT,sebagai inout dengan kondisi awal 0 contoh pemakaian

```
pin_output_lo (PORTC, 2) ;
```

Definition at line 100 of file [digital\\_io.h](#).

#### 2.3.1.3 #define pin\_output\_hi( x, y ) \_\_port\_setup(&(x),5,\_BV(y))

pin\_input\_hi adalah wrapper \_\_port\_setup, untuk inialisasi pin pada PORT,sebagai output dengan kondisi awal 1 contoh pemakaian

```
pin_output_hi (PORTC, 2) ;
```

Definition at line 76 of file [digital\\_io.h](#).

#### 2.3.1.4 #define pin\_output\_lo( x, y ) \_\_port\_setup(&(x),9,\_BV(y))

pin\_input\_lo adalah wrapper \_\_port\_setup, untuk inialisasi pin pada PORT,sebagai output dengan kondisi awal 0 contoh pemakaian

```
pin_output_lo (PORTC, 2) ;
```

Definition at line 84 of file [digital\\_io.h](#).

### 2.3.1.5 #define pin\_read( x, y ) ( \_Bool )\*( &(x)-2 )&\_BV(y)

Definition at line 136 of file [digital\\_io.h](#).

### 2.3.1.6 #define pin\_write\_hi( x, y ) sbi(x,y)

pin\_write\_hi adalah wrapper `__port_setup`, untuk mengset pin pada PORT, sebagai output dengan kondisi 1 contoh pemakaian

```
pin_write_hi(PORTC, 2);
```

Definition at line 124 of file [digital\\_io.h](#).

### 2.3.1.7 #define pin\_write\_lo( x, y ) cbi(x,y)

pin\_write\_lo adalah wrapper `__port_setup`, untuk mereset pin pada PORT, sebagai output dengan kondisi 0 contoh pemakaian

```
pin_write_lo(PORTC, 2);
```

Definition at line 132 of file [digital\\_io.h](#).

### 2.3.1.8 #define port\_input\_hi( x, y, ... ) \_\_port\_setup(&(x),6,argsToNumber((y),\_\_VA\_ARGS\_\_))

port\_input\_hi adalah wrapper `__port_setup`, untuk inialisasi pin pada PORT, dengan jumlah y dan list pin yang di pisahkan oleh koma pada argument ... sebagai input dengan kondisi awal 1 contoh pemakaian

```
port_input_hi(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 59 of file [digital\\_io.h](#).

### 2.3.1.9 #define port\_input\_lo( x, y, ... ) \_\_port\_setup(&(x),10,argsToNumber((y),\_\_VA\_ARGS\_\_))

port\_input\_lo adalah wrapper `__port_setup`, untuk inialisasi pin pada PORT, dengan jumlah y dan list pin yang di pisahkan oleh koma sebagai input dengan kondisi awal 0 contoh pemakaian

```
port_input_lo(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 68 of file [digital\\_io.h](#).

2.3.1.10 `#define port_output_hi( x, y, ... ) ___port_setup(&(x),5,argsToNumber((y),__VA_ARGS__--))`

`port_output_hi` adalah wrapper `___port_setup`, untuk inialisasi pin pada PORT, dengan jumlah `y` dan list pin yang di pisahkan oleh koma pada argument `...` sebagai output dengan kondisi awal 1 contoh pemakaian

```
port_output_hi(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 41 of file [digital\\_io.h](#).

2.3.1.11 `#define port_output_lo( x, y, ... ) ___port_setup(&(x),9,argsToNumber((y),__VA_ARGS__--))`

`port_output_lo` adalah wrapper `___port_setup`, untuk inialisasi pin pada PORT, dengan jumlah `y` dan list pin yang di pisahkan oleh koma pada argument `...` sebagai output dengan kondisi awal 0 contoh pemakaian

```
port_output_lo(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 50 of file [digital\\_io.h](#).

2.3.1.12 `#define port_read( x ) *(&(x)-2)`

Definition at line 134 of file [digital\\_io.h](#).

2.3.1.13 `#define port_write_hi( x, y, ... ) ___port_setup(&(x),4,argsToNumber((y),__VA_ARGS__--))`

`port_write_hi` adalah wrapper `___port_setup`, untuk menset pin pada PORT, dengan jumlah `y` dan list pin yang dipisahkan oleh koma pada argument `...` menjadi bernilai 1

```
port_write_hi(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 108 of file [digital\\_io.h](#).

2.3.1.14 `#define port_write_lo( x, y, ... ) ___port_setup(&(x),8,argsToNumber((y),__VA_ARGS__--))`

`port_write_lo` adalah wrapper `___port_setup`, untuk mereset pin pada PORT, dengan jumlah `y` dan list pin yang dipisahkan oleh koma pada argument `...` menjadi bernilai 0

```
port_write_lo(PORTC, 4, 0, 1, 2, 3);
```

Definition at line 116 of file [digital\\_io.h](#).

### 2.3.2 Function Documentation

2.3.2.1 void `__port_setup` ( volatile uint8\_t \* *PORT*, const uint8\_t *flag0*, const uint8\_t *\_\_PORT* )

berfungsi untuk mengatur PORT dan DDR fungsi ini tidak dirancang untuk digunakan secara langsung namun dibungkus melalui wrapper untuk menyederhanakan pemakaian

#### Parameters

|               |                   |
|---------------|-------------------|
| <i>PORT</i>   | volatile uint8_t* |
| <i>flag0</i>  | const uint8_t     |
| <i>__PORT</i> | const uint8_t     |

#### Returns

void

Definition at line 25 of file [digital\\_io.c](#).

2.3.2.2 const uint8\_t `argsToNumber` ( const uint8\_t *n*, ... )

`ArgsToNumber` berfungsi untuk merubah parameter list ... dengan jumlah uint8\_t *n* menjadi bentuk `_BV(item parameter list 1)|_BV(_BV(item parameter list 1))| dst.`

#### Parameters

|          |               |
|----------|---------------|
| <i>n</i> | const uint8_t |
| ...      |               |

#### Returns

const uint8\_t

Definition at line 10 of file [digital\\_io.c](#).

## 2.4 digital\_io.h

```
00001 #ifndef DIGITAL_IO_H_INCLUDED
00002 #define DIGITAL_IO_H_INCLUDED
00003
00004 /*
00005  * Library Digital IO
00006  * License LGPLv3
00007  * Copy-left kenzanin[AT]gmail[dot]com
00008  */
00009
00010 #include <avr/io.h>
00011 #include <compat/deprecated.h>
00012
00020 const uint8_t argsToNumber(const uint8_t n,...)__attribute__((pure));
00021
```

```

00022
00032 void __port_setup(volatile uint8_t* PORT, const uint8_t flag0, const uint8_t
__PORT);
00033
00041 #define port_output_hi(x,y,...)
__port_setup(&(x),5,argsToNumber((y),__VA_ARGS__))
00042
00050 #define port_output_lo(x,y,...)
__port_setup(&(x),9,argsToNumber((y),__VA_ARGS__))
00051
00059 #define port_input_hi(x,y,...)
__port_setup(&(x),6,argsToNumber((y),__VA_ARGS__))
00060
00068 #define port_input_lo(x,y,...)
__port_setup(&(x),10,argsToNumber((y),__VA_ARGS__))
00069
00076 #define pin_output_hi(x,y) __port_setup(&(x),5,_BV(y))
00077
00084 #define pin_output_lo(x,y) __port_setup(&(x),9,_BV(y))
00085
00092 #define pin_input_hi(x,y) __port_setup(&(x),6,_BV(y))
00093
00100 #define pin_input_lo(x,y) __port_setup(&(x),10,_BV(y))
00101
00108 #define port_write_hi(x,y,...)
__port_setup(&(x),4,argsToNumber((y),__VA_ARGS__))
00109
00116 #define port_write_lo(x,y,...)
__port_setup(&(x),8,argsToNumber((y),__VA_ARGS__))
00117
00124 #define pin_write_hi(x,y) sbi(x,y)
00125
00132 #define pin_write_lo(x,y) cbi(x,y)
00133
00134 #define port_read(x) *(&(x)-2)
00135
00136 #define pin_read(x,y) (_Bool)*(&(x)-2)&_BV(y)
00137 // #define pin_read_wait(x,y,z) <todo>
00138
00139 #endif // DIGITAL_IO_H_INCLUDED

```

## 2.5 main.c File Reference

```
#include <avr/io.h> #include <stdarg.h> #include "digital-
_io/digital_io.h"
```

### Functions

- int [main](#) (void)

#### 2.5.1 Function Documentation

##### 2.5.1.1 int main ( void )

kenzanin AVR Pin Input Output Driver license LGPLv3 Copy-left kenzanin[at]gmail[dot]com

Definition at line 11 of file [main.c](#).

## 2.6 main.c

```
00001
00007 #include <avr/io.h>
00008 #include <stdarg.h>
00009 #include "digital_io/digital_io.h"
00010
00011 int main(void)
00012 {
00013     port_output_hi(PORTB, 4, 0, 1, 2, 3);
00014     port_output_lo(PORTB, 4, 4, 5, 6, 7);
00015     port_input_hi(PORTC, 4, 0, 1, 2, 3);
00016     port_input_lo(PORTC, 4, 0, 1, 2, 3);
00017
00018     port_write_hi(PORTB, 4, 4, 5, 6, 7);
00019     port_write_lo(PORTB, 4, 0, 1, 2, 3);
00020
00021     while(1)
00022     {
00023     }
00024     return 0;
00025
00026 }
```