

AVR Digital IO Library
1.0

Generated by Doxygen 1.7.5.1

Sun Jan 27 2013 11:34:27

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
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.h File Reference	4
2.2.1	Define Documentation	5
2.2.1.1	pin_input_hi	5
2.2.1.2	pin_input_lo	5
2.2.1.3	pin_output_hi	5
2.2.1.4	pin_output_lo	5
2.2.1.5	pin_read	6
2.2.1.6	pin_write_hi	6
2.2.1.7	pin_write_lo	6
2.2.1.8	port_input_hi	6
2.2.1.9	port_input_lo	6
2.2.1.10	port_output_hi	6
2.2.1.11	port_output_lo	7
2.2.1.12	port_read	7
2.2.1.13	port_write_hi	7
2.2.1.14	port_write_lo	7
2.2.2	Function Documentation	7
2.2.2.1	__port_setup	7

2.2.2.2	argsToNumber	8
2.3	main.c File Reference	8
2.3.1	Function Documentation	8
2.3.1.1	main	8

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

digital_io.c	3
digital_io.h	4
main.c	8

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.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

2.2.1 Define Documentation

2.2.1.1 #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`.

2.2.1.2 #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`.

2.2.1.3 #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`.

2.2.1.4 #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`.

2.2.1.5 `#define pin_read(x, y) (_Bool)*(&(x)-2)&_BV(y)`

Definition at line 136 of file `digital_io.h`.

2.2.1.6 `#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`.

2.2.1.7 `#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`.

2.2.1.8 `#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`.

2.2.1.9 `#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`.

2.2.1.10 `#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.

2.2.1.11 `#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.

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

Definition at line 134 of file digital_io.h.

2.2.1.13 `#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.

2.2.1.14 `#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.

2.2.2 Function Documentation

2.2.2.1 `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.

2.2.2.2 `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.

2.3 main.c File Reference

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

Functions

- int `main` (void)

2.3.1 Function Documentation

2.3.1.1 `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.

Index

- __port_setup
 - digital_io.c, [3](#)
 - digital_io.h, [6](#)
- argsToNumber
 - digital_io.c, [3](#)
 - digital_io.h, [6](#)
- digital_io.c, [3](#)
 - __port_setup, [3](#)
 - argsToNumber, [3](#)
- digital_io.h, [4](#)
 - __port_setup, [6](#)
 - argsToNumber, [6](#)
 - pin_input_hi, [4](#)
 - pin_input_lo, [4](#)
 - pin_output_hi, [4](#)
 - pin_output_lo, [5](#)
 - pin_read, [5](#)
 - pin_read_wait, [5](#)
 - pin_write_hi, [5](#)
 - pin_write_lo, [5](#)
 - port_input_hi, [5](#)
 - port_input_lo, [5](#)
 - port_output_hi, [5](#)
 - port_output_lo, [5](#)
 - port_read, [5](#)
 - port_write_hi, [6](#)
 - port_write_lo, [6](#)
- main
 - main.c, [7](#)
- main.c, [6](#)
 - main, [7](#)
- pin_input_hi
 - digital_io.h, [4](#)
- pin_input_lo
 - digital_io.h, [4](#)
- pin_output_hi
 - digital_io.h, [4](#)
- pin_output_lo
 - digital_io.h, [5](#)
- pin_read
 - digital_io.h, [5](#)
- pin_read_wait
 - digital_io.h, [5](#)
- pin_write_hi
 - digital_io.h, [5](#)
- pin_write_lo
 - digital_io.h, [5](#)
- port_input_hi
 - digital_io.h, [5](#)
- port_input_lo
 - digital_io.h, [5](#)
- port_output_hi
 - digital_io.h, [5](#)
- port_output_lo
 - digital_io.h, [5](#)
- port_read
 - digital_io.h, [5](#)
- port_write_hi
 - digital_io.h, [6](#)
- port_write_lo
 - digital_io.h, [6](#)