Documentación de la placa de desarrollo

Generated by Doxygen 1.7.6.1

Sat Mar 30 2013 00:41:44

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 IOPIN_t Struct Reference

`#include <pin.h>`

3.1.1 Detailed Description

IOPIN_t Estructura que define todo lo necesario para controlar un pin.
Ejemplo de uso:

```
#include <avr/io.h>
IOPIN_t led1 = {&PORTC, &DDRC, &PINC, PIN3);
```

See also

configPin()

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

- pin.h

3.2 PWM2_t Struct Reference

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

- pwm.h

Chapter 4

File Documentation

4.1 bitmanip.h File Reference

Defines

- #define true 1
- #define false 0
- #define nop() asm volatile("nop")
- #define setBit(Byte, Bit) ((Byte) |= (1<<(Bit)))
- #define clearBit(Byte, Bit) ((Byte) &= (~(1<<(Bit))))
- #define isBitSet(Byte, Bit) (((Byte) & (1<<(Bit))) ? true : false)
- #define isBitClear(Byte, Bit) (((Byte) & (1<<(Bit))) ? false : true)

4.1.1 Detailed Description

Macros para manipulación de bits.

4.1.2 Define Documentation

4.1.2.1 #define clearBit(*Byte, Bit*) ((Byte) &= (~(1<<(Bit))))

Pone un bit en cero dentro de un byte.

Parameters

Byte	palabra de 8 bits
Bit	posición del bit a apagar. Rango de 0 a 7.

4.1.2.2 #define false 0

Constante false

4.1.2.3 #define isBitClear(*Byte, Bit*) (((Byte) & (1<<(Bit))) ? false : true)

Indica si un bit está apagado.

Parameters

Byte	palabra de 8 bits
Bit	posición del bit. Rango de 0 a 7.

Returns

- true: si está apagado
- false: si está encendido

4.1.2.4 #define isBitSet(*Byte, Bit*) (((Byte) & (1<<(Bit))) ? true : false)

Indica si un bit está encendido.

Parameters

Byte	palabra de 8 bits
Bit	posición del bit. Rango de 0 a 7.

Returns

- true: si está encendido
- false: si está apagado

4.1.2.5 #define nop( ) asm volatile("nop")

Instrucción vacía. Sirve para insertar una demora de exactamente un ciclo de reloj.

4.1.2.6 #define setBit(*Byte, Bit*) ((Byte) |= (1<<(Bit)))

Pone un bit en uno dentro de un byte.

Parameters

Byte	palabra de 8 bits
Bit	posición del bit a encender. Rango de 0 a 7.

4.1.2.7 #define true 1

Constante true

4.2 board.h File Reference

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

Functions

- void initBoard (void)
- void initLeds (void)
- void initButton (void(∗isr)(void), uint8_t debounce)
- uint8_t readButton (void)

4.2.1 Detailed Description

Define cómo está conectado todo en la placa.

4.2.2 Function Documentation

4.2.2.1 void initBoard (void)

Inicializa todo lo vinculado a la placa. Debe ser llamada siempre al comienzo antes de usar cualquier elemento de la placa.

4.2.2.2 void initButton (void(∗)(void) *isr,* uint8_t *debounce*)

Inicializa el botón.

Parameters

isr	Puntero a la función que será llamada cuando se presione el botón.
debounce	Cantidad de milisegundos para filtrar los rebotes

4.2.2.3 void initLeds (void)

Inicializa los puertos de los leds.

4.2.2.4 uint8_t readButton (void)

Retorna el estado actual del botón.

Returns

- 0: si el botón está presionado
- 1: si el botón no está presionado

4.3 delay.h File Reference

```
#include <util/delay_basic.h>
```

Functions

- static void delay_ms (uint16_t msec)

4.3.1 Detailed Description

Cabecera para funciones de demoras temporales.

4.3.2 Function Documentation

4.3.2.1 void delay_ms (uint16_t *msec*) [inline, static]

Produce un retardo durante una cantidad entera de milisegundos.

Parameters

msec	tiempo en milisegundos. Rango de 1 a 65536.

4.4 interrupt.h File Reference

Functions

- void configExtInt (uint8_t num, uint8_t sense, void(∗isr)(void))

4.4.1 Detailed Description

Cabecera para el manejo de interrupciones en Atmega8 y Atmega88.

4.4.2 Function Documentation

4.4.2.1 void configExtInt (uint8_t *num*, uint8_t *sense*, void(∗)(void) *isr*)

Configura una interrupción externa.

Parameters

num	número de interupción 0 o 1
sense	tipo de activación para la interrupción • 0: nivel bajo • 1: cualquier cambio de nivel • 2: flanco descendente • 3: flanco ascendente
isr	puntero a la función de manejo de la interrupción. Si es NULL se desactiva la interrupción.

4.5 pin.h File Reference

```
#include "bitmanip.h"
```

Data Structures

- struct IOPIN_t

Functions

- void configPin (const IOPIN_t *pin, uint8_t dir, uint8_t pullup)
- static void setPin (const IOPIN_t *pin)
- static void clearPin (const IOPIN_t *pin)
- static void togglePin (const IOPIN_t *pin)
- static uint8_t readPort (const IOPIN_t *pin)
- static uint8_t readPin (const IOPIN_t *pin)

4.5.1 Detailed Description

Cabecera para el manejo de pines en Atmega8 y Atmega88.

4.5.2 Function Documentation

4.5.2.1 void clearPin (const IOPIN_t * *pin*) [inline, static]

Apaga el bit del puerto.

Parameters

pin	puntero a la estructura que describe al pin

4.5.2.2 void configPin (const IOPIN_t * *pin*, uint8_t *dir*, uint8_t *pullup*)

Configura un pin como entrada o salida

Parameters

pin	puntero a la estructura que describe al pin
dir	• 1: salida • 0: entrada
pullup	• 1: con pullup • 0: sin pullup

See also

setPin(), clearPin(), togglePin(), readPort()

4.5.2.3 uint8_t readPin (const IOPIN_t * *pin*) [inline, static]

Lee el valor actual del pin correspondiente (sin buffer).

Parameters

pin	puntero a la estructura que describe al pin

4.5.2.4 uint8_t readPort (const IOPIN_t * *pin*) [inline, static]

Lee el valor actual del port (buffer) para el pin correspondiente.

Parameters

pin	puntero a la estructura que describe al pin

4.5.2.5 void setPin (const IOPIN_t * *pin*) [inline, static]

Enciende el bit del puerto.

Parameters

pin	puntero a la estructura que describe al pin

4.5.2.6 void togglePin (const IOPIN_t * *pin*) [inline, static]

Invierte el bit del puerto.

Parameters

pin	puntero a la estructura que describe al pin

4.6 pwm.h File Reference

Data Structures

- struct PWM2_t

Functions

- void configPWM2 (uint8_t mode, uint8_t prescaler, void(*isr)(void), uint8_t output)
- void startPWM2 (void)
- void stopPWM2 (void)
- void setPWM2 (uint8_t x)

4.6.1 Detailed Description

Cabecera para el manejo del PWM en Atmega8 y Atmega88.

4.6.2 Function Documentation

4.6.2.1 void configPWM2 (uint8_t *mode*, uint8_t *prescaler*, void(*)(void) *isr*, uint8_t *output*)

Configura el PWM asociado al timer 2 (8 bits).

Parameters

mode	• 1: phase correct PWM • 3: fast PWM
prescaler	divisor para la frecuencia del micro • 1: fclk • 2: fclk/8 • 3: fclk/32 • 4: fclk/64 • 5: fclk/128 • 6: fclk/256 • 7: fclk/1024
isr	puntero a la función de manejo de la interrupción que se produce cada vez que se alcanza el nivel de comparación. La función no debe recibir ningún parámetro y no debe devolver nada. Ej: void mifunc(void){}. Si es NULL no se generará la interrupción.
output	• 0 sin salida • 1 salida por OC2

4.6.2.2 void setPWM2 (uint8_t *x*)

Establece el ancho de pulso.

Parameters

x	ancho del pulso

4.6.2.3 void startPWM2 (void)

Arranca el PWM a la frecuencia que fue configurada.

4.6.2.4 void stopPWM2 (void)

Detiene el PWM.

4.7 utils.h File Reference

`#include <stddef.h> #include "bitmanip.h" #include "pin.-h" #include "pwm.h" #include "interrupt.h" #include "delay.-h"`

4.7.1 Detailed Description

Cabecera para la biblioteca de utilidades generales para el Atmega8 y Atmega88.