

Proyecto Hardware

Generado por Doxygen 1.8.6

Martes, 5 de Enero de 2016 20:31:34

Índice general

1	Índice de estructura de datos	1
1.1	Estructura de datos	1
2	Índice de archivos	3
2.1	Lista de archivos	3
3	Documentación de las estructuras de datos	5
3.1	Referencia de la Estructura BITMAP	5
4	Documentación de archivos	7
4.1	Referencia del Archivo 8led.h	7
4.1.1	Documentación de las funciones	7
4.1.1.1	D8Led_blink_symbol	7
4.1.1.2	D8Led_current_symbol	7
4.1.1.3	D8Led_init	8
4.1.1.4	D8Led_symbol	8
4.2	Referencia del Archivo aperture-logo-bitmap.h	8
4.2.1	Descripción detallada	8
4.3	Referencia del Archivo Bmp.h	8
4.3.1	Descripción detallada	9
4.3.2	Documentación de las funciones	9
4.3.2.1	BitmapPop	9
4.3.2.2	BitmapPush	9
4.3.2.3	BitmapView	10
4.3.2.4	CursorPush	10
4.3.2.5	CursorView	10
4.4	Referencia del Archivo Button.h	10
4.4.1	Descripción detallada	10
4.4.2	Documentación de las funciones	11
4.4.2.1	action	11
4.4.2.2	Button_init	11
4.4.2.3	Button_low_next	11

4.4.2.4	Button_low_update_screen	11
4.4.2.5	Button_next	11
4.4.2.6	Button_reconfigure_range	11
4.4.2.7	Button_set_valor_actual	11
4.4.2.8	Button_update_screen	12
4.4.2.9	Button_valor_actual	12
4.5	Referencia del Archivo lcd.h	12
4.5.1	Descripción detallada	14
4.5.2	Documentación de los 'defines'	14
4.5.2.1	LCD_Active_PutPixel	14
4.5.2.2	LCD_PutPixel	14
4.5.3	Documentación de las funciones	14
4.5.3.1	Lcd_Active_Clr	14
4.5.3.2	Lcd_Circle	14
4.5.3.3	Lcd_Circle_Filled	14
4.5.3.4	Lcd_Clr	14
4.5.3.5	Lcd_DisplayChar	15
4.5.3.6	Lcd_DisplayChar_inverted	16
4.5.3.7	Lcd_Dma_Trans	16
4.5.3.8	Lcd_Dma_Trans_non_block	16
4.5.3.9	Lcd_Dma_Trans_wait	16
4.5.3.10	Lcd_Draw_Box	16
4.5.3.11	Lcd_Draw_Box_inverted	16
4.5.3.12	Lcd_Draw_Filled_Box	17
4.5.3.13	Lcd_Draw_HLine	17
4.5.3.14	Lcd_Draw_HLine_inverted	17
4.5.3.15	Lcd_Draw_HLine_pointed	17
4.5.3.16	Lcd_Draw_Line	18
4.5.3.17	Lcd_Draw_VLine	18
4.5.3.18	Lcd_Draw_VLine_inverted	18
4.5.3.19	Lcd_Draw_VLine_pointed	18
4.5.3.20	Lcd_DspAscll8x16	19
4.5.3.21	Lcd_DspAscll8x16_inverted	20
4.5.3.22	Lcd_DspAscll8x16HorizontallyCentered	20
4.5.3.23	Lcd_DspAscll8x16HorizontallyCentered_inverted	20
4.5.3.24	LCD_GetPixel	20
4.5.3.25	Lcd_Init	21
4.5.3.26	LcdClrRect	21
4.5.3.27	ReverseLine	21
4.5.3.28	ReverseSquare	21

4.6	Referencia del Archivo Persistence.h	21
4.6.1	Descripción detallada	22
4.6.2	Documentación de las funciones	22
4.6.2.1	Persistence_read_char	22
4.6.2.2	Persistence_read_int	22
4.6.2.3	Persistence_save_char	22
4.6.2.4	Persistence_save_int	22
4.7	Referencia del Archivo still-alive-lyrics.h	22
4.7.1	Descripción detallada	23
4.8	Referencia del Archivo sudoku_2015.h	23
4.8.1	Descripción detallada	23
4.8.2	Documentación de las funciones	24
4.8.2.1	celda_cambiar_candidatos	24
4.8.2.2	celda_es_candidato	25
4.8.2.3	celda_es_error	25
4.8.2.4	celda_es_pista	25
4.8.2.5	celda_leer_valor	25
4.8.2.6	celda_poner_valor	25
4.8.2.7	init_game	26
4.8.2.8	sudoku_candidatos_arm	26
4.8.2.9	sudoku_empty_grid	26
4.8.2.10	sudoku_recalcular	26
4.9	Referencia del Archivo sudoku_collection_san.h	26
4.9.1	Descripción detallada	27
4.9.2	Documentación de las funciones	27
4.9.2.1	sudoku_collection_descomprime	27
4.9.3	Documentación de las variables	27
4.9.3.1	cuadrículaCasiResuelta	27
4.9.3.2	cuadrículas	27
4.10	Referencia del Archivo sudoku_graphics.h	27
4.10.1	Descripción detallada	28
4.10.2	Documentación de las funciones	28
4.10.2.1	sudoku_graphics_draw_base	28
4.10.2.2	sudoku_graphics_draw_state	28
4.10.2.3	sudoku_graphics_draw_time	29
4.10.2.4	sudoku_graphics_fill_from_data	29
4.10.2.5	sudoku_graphics_mark_error	29
4.10.2.6	sudoku_graphics_mark_error_in_square	29
4.10.2.7	sudoku_graphics_print_final_screen	29
4.10.2.8	sudoku_graphics_print_instructions	30

4.10.2.9	sudoku_graphics_print_record	30
4.10.2.10	sudoku_graphics_print_still_alive	30
4.10.2.11	sudoku_graphics_print_title_screen	30
4.10.2.12	sudoku_graphics_put_number_in_square	30
4.10.2.13	sudoku_graphics_remark_square	30
4.10.2.14	sudoku_graphics_update_lcd	30
4.11	Referencia del Archivo Timer2.h	31
4.11.1	Descripción detallada	31
4.11.2	Documentación de las funciones	31
4.11.2.1	Timer2_Empezar	31
4.11.2.2	Timer2_Inicializar	31
4.11.2.3	Timer2_Leer	31
4.11.2.4	Timer2_Reiniciar	31
4.12	Referencia del Archivo title-bitmap.h	31
Índice		33

Capítulo 1

Índice de estructura de datos

1.1. Estructura de datos

Lista de estructuras con una breve descripción:

BITMAP	5
----------------------------------	---

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

8led.h	Funciones de control del display 8-segmentos	7
aperture-logo-bitmap.h	Archivo de recursos que contiene el logo de Aperture	8
Bmp.h	Definicion mapas de bits del LCD	8
Button.h	Modulo que gestiona los rebotes de los botones	10
lcd.h	Funciones de visualizacion y control LCD	12
Persistence.h	Modulo que permite una forma simple de persistencia	21
still-alive-lyrics.h	Archivo de recursos con los creditos finales	22
sudoku_2015.h	Modulo que contiene funciones relacionadas con el analisis de sudokus	23
sudoku_collection_san.h	Fichero de recursos que contiene cuadrículas	26
sudoku_graphics.h	Modulo que actua de capa de abstraccion entre el juego y la pantalla	27
Timer2.h	Modulo que gestiona un contador	31
title-bitmap.h	Archivo de recursos con la imagen del titulo	31
common/44b.h	??
common/44blib.h	??
common/def.h	??
common/option.h	??

Capítulo 3

Documentación de las estructuras de datos

3.1. Referencia de la Estructura BITMAP

Campos de datos

- INT8U **ucFlags**
- INT8U **ucBitsPix**
- INT16U **usWidth**
- INT16U **usHeight**
- INT32U **ulTransparentColor**
- INT8U * **pucStart**

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [Bmp.h](#)

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo 8led.h

Funciones de control del display 8-segmentos.

Funciones

- void `D8Led_init` (void)
Inicializacion del sistema del 8 segmentos.
- void `D8Led_symbol` (int value)
Cambia el valor en el 8 segmentos.
- int `D8Led_current_symbol` (void)
Devuelve el simbolo actualmente en el 8 segmentos.
- void `D8Led_blink_symbol` (int value, int ms)
Pone temporalmente un nuevo simbolo el 8 segmentos.

4.1.1. Documentación de las funciones

4.1.1.1. void `D8Led_blink_symbol` (int value, int ms)

Cambia temporalmente el simbolo en el 8 segmentos al valor value (este parametro ha de cumplir las mismas condiciones que para su uso en la funcion `D8Led_symbol`). El nuevo simbolo permanecera en el 8 segmentos durante los siguientes ms milisegundos. A continuacion se volvera a poner el valor inicial. Es una funcion bloqueante.

Parámetros

in	value	Valor a poner en el 8 segmentos
in	ms	Milisegundos que permanecera el simbolo en el 8 segmentos

4.1.1.2. int `D8Led_current_symbol` (void)

Devuelve el valor en el 8 segmentos, interpretado como un numero decimal en el rango [0,15]

Devuelve

Simbolo actualmente en el 8 segmentos

4.1.1.3. void D8Led_init (void)

Inicializa el sistema del 8 segmentos, y pone el simbolo 0 en el mismo

4.1.1.4. void D8Led_symbol (int value)

Cambia el simbolo en el 8 segmentos al numero pasado, interpretado en hexadecimal. En caso de que el valor no pueda ser mostrado con un unico digito hexadecimal, no cambia el numero en el 8 segmentos.

Parámetros

in	value	Valor a poner en el 8 segmentos
----	-------	---------------------------------

4.2. Referencia del Archivo aperture-logo-bitmap.h

Archivo de recursos que contiene el logo de Aperture.

```
#include "lcd.h"
#include "bmp.h"
```

'defines'

- #define **APERTURE_LOGO_HEIGHT** 100
- #define **APERTURE_LOGO_WIDTH** 100

Variables

- const INT8U **apertureBitmap** [APERTURE_LOGO_HEIGHT *APERTURE_LOGO_WIDTH]
- [STRU_BITMAP](#) **Stru_Bitmap_aperture**

4.2.1. Descripción detallada

Autor

Guillermo Robles Gonzalez

4.3. Referencia del Archivo Bmp.h

Definicion mapas de bits del LCD.

```
#include "def.h"
```

Estructuras de datos

- struct [BITMAP](#)

'defines'

- #define **BLACK** 0xf
- #define **WHITE** 0x0
- #define **LIGHTGRAY** 0x5
- #define **DARKGRAY** 0xa
- #define **TRANSPARENCY** 0xff

'typedefs'

- typedef struct **BITMAP** **STRU_BITMAP**
- typedef struct **BITMAP** * **pSTRU_BITMAP**

Funciones

- void **BitmapView** (INT16U x, INT16U y, **STRU_BITMAP** Stru_Bitmap)
Display bitmap in virtual buffer.
- void **BitmapViewHorizontallyCentered** (INT16U y, **STRU_BITMAP** Stru_Bitmap)
- void **BitmapViewVerticallyCentered** (INT16U x, **STRU_BITMAP** Stru_Bitmap)
- void **BitmapViewCentered** (**STRU_BITMAP** Stru_Bitmap)
- void **BitmapPush** (INT16U x, INT16U y, **STRU_BITMAP** Stru_Bitmap)
Push bitmap data into LCD active buffer.
- void **BitmapPop** (INT16U x, INT16U y, **STRU_BITMAP** Stru_Bitmap)
Pop bitmap data into LCD active buffer.
- void **CursorInit** (void)
Cursor init.
- void **CursorView** (INT16U x, INT16U y)
Cursor display.
- void **CursorPush** (INT16U x, INT16U y)
Cursor push.
- void **CursorPop** (void)
Cursor pop.

4.3.1. Descripción detallada

Autor

4.3.2. Documentación de las funciones**4.3.2.1. void BitmapPop (INT16U x, INT16U y, STRU_BITMAP Stru_Bitmap)****Parámetros**

in	x	X coordinate of the position to pop into
in	y	Y coordinate of the position to pop into
in	Stru_Bitmap	Bitmap to pop

4.3.2.2. void BitmapPush (INT16U x, INT16U y, STRU_BITMAP Stru_Bitmap)

Parámetros

in	x	X coordinate of the position to draw
in	y	Y coordinate of the position to draw
in	<i>Stru_Bitmap</i>	Bitmap to draw

4.3.2.3. void BitmapView (INT16U x, INT16U y, STRU_BITMAP Stru_Bitmap)

Parámetros

in	x	X coordinate of the position to draw
in	y	Y coordinate of the position to draw
in	<i>Stru_Bitmap</i>	Bitmap to draw

4.3.2.4. void CursorPush (INT16U x, INT16U y)

Parámetros

in	x	X coordinate of the position to push
in	y	Y coordinate of the position to push

4.3.2.5. void CursorView (INT16U x, INT16U y)

Parámetros

in	x	X coordinate of the position to draw
in	y	Y coordinate of the position to draw

4.4. Referencia del Archivo Button.h

Modulo que gestiona los rebotes de los botones.

Funciones

- int [Button_valor_actual](#) (void)
- int [Button_next](#) (void)
- void [Button_init](#) (int min, int max)
- void [action](#) (int n)
- void [Button_set_valor_actual](#) (int n)
- void [Button_low_next](#) (void)
- int [Button_update_screen](#) ()
- void [Button_low_update_screen](#) ()
- void [Button_reconfigure_range](#) (int new_min, int new_max)

4.4.1. Descripción detallada

Modulo que se encarga de gestionar los botones y su uso en el proyecto, inicializarlos y sincronizar el 7 segmentos con el valor interno del boton.

Autor

Guillermo Robles Gonzalez

4.4.2. Documentación de las funciones

4.4.2.1. void action (int *n*)

Accion asociada a la pulsacion del boton de codigo n

Parámetros

<i>in</i>	<i>n</i>	Codigo de boton pulsado
-----------	----------	-------------------------

4.4.2.2. void Button_init (int *min*, int *max*)

Iniciar el sistema de botones, poniendo la pantalla al valor minimo pasado

Parámetros

<i>in</i>	<i>min</i>	Valor minimo que aparecera en la pantalla
<i>in</i>	<i>max</i>	Valor maximo que aparecera en la pantalla

4.4.2.3. void Button_low_next (void)

Baja el flag interno de next

4.4.2.4. void Button_low_update_screen ()

Baja el flag interno indicando que se han realizado acciones

4.4.2.5. int Button_next (void)

Informa de si se ha de avanzar

Devuelve

1 en caso de que el flag este activo, 0 en caso contrario

4.4.2.6. void Button_reconfigure_range (int *new_min*, int *new_max*)

Reconfigura el rango que maneja el boton, solo seran validos valores en el rango [0,15]

Parámetros

<i>in</i>	<i>new_min</i>	Nuevo minimo
<i>in</i>	<i>new_max</i>	Nuevo maximo

4.4.2.7. void Button_set_valor_actual (int *n*)

Ajusta el valor de la cuenta interna del boton, actualizando el display. Ha de pertenecer al rango al cual esta actualmente configurado el boton.

Parámetros

<i>in</i>	<i>n</i>	nuevo valor a poner
-----------	----------	---------------------

4.4.2.8. int Button_update_screen ()

Indica si ha sido realizada alguna accion que afecte al estado interno del boton, tanto la pulsacion de un boton como la repeticion de una accion

Devuelve

1 en caso de que el flag este activo, 0 en caso contrario

4.4.2.9. int Button_valor_actual (void)

Valor actual de la cuenta interna

Devuelve

valor actual

4.5. Referencia del Archivo lcd.h

Funciones de visualizacion y control LCD.

```
#include "def.h"
```

'defines'

- **#define TLCD_160_240** (0)
- **#define VLCD_240_160** (1)
- **#define CLCD_240_320** (2)
- **#define MLCD_320_240** (3)
- **#define ELCD_640_480** (4)
- **#define SLCD_160_160** (5)
- **#define LCD_TYPE** MLCD_320_240
- **#define SCR_XSIZE** (320)
- **#define SCR_YSIZE** (240)
- **#define LCD_XSIZE** (320)
- **#define LCD_YSIZE** (240)
- **#define MODE_MONO** (1)
- **#define MODE_GREY4** (4)
- **#define MODE_GREY16** (16)
- **#define MODE_COLOR** (256)
- **#define Ascii_W** 8
- **#define XWIDTH** 6
- **#define BLACK** 0xf
- **#define WHITE** 0x0
- **#define LIGHTGRAY** 0x5
- **#define DARKGRAY** 0xa
- **#define TRANSPARENCY** 0xff
- **#define HOZVAL** (LCD_XSIZE/4-1)

- `#define HOZVAL_COLOR (LCD_XSIZE*3/8-1)`
- `#define LINEVAL (LCD_YSIZE -1)`
- `#define MVAL (13)`
- `#define M5D(n) ((n) & 0x1ffff)`
- `#define MVAL_USED 0`
- `#define ARRAY_SIZE_MONO (SCR_XSIZE/8*SCR_YSIZE)`
- `#define ARRAY_SIZE_GREY4 (SCR_XSIZE/4*SCR_YSIZE)`
- `#define ARRAY_SIZE_GREY16 (SCR_XSIZE/2*SCR_YSIZE)`
- `#define ARRAY_SIZE_COLOR (SCR_XSIZE/1*SCR_YSIZE)`
- `#define CLKVAL_MONO (12)`
- `#define CLKVAL_GREY4 (12)`
- `#define CLKVAL_GREY16 (12)`
- `#define CLKVAL_COLOR (10)`
- `#define LCD_BUF_SIZE (SCR_XSIZE*SCR_YSIZE/2)`
- `#define LCD_ACTIVE_BUFFER (0xc300000)`
- `#define LCD_VIRTUAL_BUFFER (0xc300000 + LCD_BUF_SIZE)`
- `#define LCD_PutPixel(x, y, c)`
- `#define LCD_Active_PutPixel(x, y, c)`
- `#define GUISWAP(a, b) {a^=b; b^=a; a^=b;}`

Funciones

- `INT8U LCD_GetPixel (INT16U usX, INT16U usY)`
- `void Lcd_Init (void)`
- `void Lcd_Active_Clr (void)`
- `void Lcd_Clr (void)`
- `void LcdClrRect (INT16 usLeft, INT16 usTop, INT16 usRight, INT16 usBottom, INT8U ucColor)`
- `void Lcd_Draw_Box (INT16 usLeft, INT16 usTop, INT16 usRight, INT16 usBottom, INT8U ucColor)`
- `void Lcd_Draw_Box_inverted (INT16 usLeft, INT16 usTop, INT16 usRight, INT16 usBottom, INT8U ucColor)`
- `void Lcd_Draw_Filled_Box (INT16 usLeft, INT16 usTop, INT16 usRight, INT16 usBottom, INT8U ucColor)`
- `void Lcd_Draw_Line (INT16 usX0, INT16 usY0, INT16 usX1, INT16 usY1, INT8U ucColor, INT16U usWidth)`
- `void Lcd_Draw_HLine (INT16 usX0, INT16 usX1, INT16 usY0, INT8U ucColor, INT16U usWidth)`
- `void Lcd_Draw_VLine (INT16 usY0, INT16 usY1, INT16 usX0, INT8U ucColor, INT16U usWidth)`
- `void Lcd_Draw_HLine_pointed (INT16 usX0, INT16 usX1, INT16 usY0, INT8U ucColor, INT16U usWidth)`
- `void Lcd_Draw_VLine_pointed (INT16 usY0, INT16 usY1, INT16 usX0, INT8U ucColor, INT16U usWidth)`
- `void Lcd_Draw_HLine_inverted (INT16 usX0, INT16 usX1, INT16 usY0, INT16U usWidth)`
- `void Lcd_Draw_VLine_inverted (INT16 usY0, INT16 usY1, INT16 usX0, INT16U usWidth)`
- `void Lcd_DspAscll8x16 (INT16U x0, INT16U y0, INT8U ForeColor, INT8U *s)`
- `void Lcd_DspAscll8x16_inverted (INT16U x0, INT16U y0, INT8U ForeColor, INT8U *s)`
- `void Lcd_DspAscll8x16HorizontallyCentered (INT16U y0, INT8U ForeColor, INT8U *s)`
- `void Lcd_DspAscll8x16HorizontallyCentered_inverted (INT16U y0, INT8U ForeColor, INT8U *s)`
- `void Lcd_DisplayChar (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U ucChar)`
- `void Lcd_DisplayChar_inverted (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U ucChar)`
- `void ReverseLine (INT32U ulHeight, INT32U ulY)`
- `void ReverseSquare (INT32U ulX0, INT32U ulY0, INT32U ulX1, INT32U ulY1)`
- `void Lcd_Circle (INT8 X, INT8 Y, INT16 radius, INT8U ForeColor)`
- `void Lcd_Circle_Filled (INT8 X, INT8 Y, INT16 radius, INT8U ForeColor)`
- `void Lcd_Dma_Trans (void)`
- `void Lcd_Dma_Trans_non_block (void)`
- `void Lcd_Dma_Trans_wait (void)`

4.5.1. Descripción detallada

Versión

<P6-ARM>

4.5.2. Documentación de los 'defines'

4.5.2.1. #define LCD_Active_PutPixel(x, y, c)

Valor:

```
(* (INT32U *) (LCD_ACTIVE_BUFFER + (y) * SCR_XSIZE / 2 + (319 - (x)) / 8 * 4)) = \
  (((*(INT32U *) (LCD_ACTIVE_BUFFER + (y) * SCR_XSIZE / 2 + (319 - (x)) / 8 * 4)) & \
    (~ (0xf0000000 » (((319 - (x)) % 8) * 4)))) | ((c) « (7 - (319 - (x)) % 8) * 4))
```

4.5.2.2. #define LCD_PutPixel(x, y, c)

Valor:

```
(* (INT32U *) (LCD_VIRTUAL_BUFFER + (y) * SCR_XSIZE / 2 + ( (x)) / 8 * 4)) = \
  (((*(INT32U *) (LCD_VIRTUAL_BUFFER + (y) * SCR_XSIZE / 2 + ( (x)) / 8 * 4)) & \
    (~ (0xf0000000 » ((( (x)) % 8) * 4)))) | ((c) « (7 - ( (x)) % 8) * 4))
```

4.5.3. Documentación de las funciones

4.5.3.1. void Lcd_Active_Clr (void)

Limpia el buffer real de la pantalla, actualizandola

4.5.3.2. void Lcd_Circle (INT8 X, INT8 Y, INT16 radius, INT8U ForeColor)

Dibuja una circunferencia

Parámetros

in	X	Coordenada X del centro
in	Y	Coordenada Y del centro
in	radius	Radio del circulo
in	ForeColor	Color a usar

4.5.3.3. void Lcd_Circle_Filled (INT8 X, INT8 Y, INT16 radius, INT8U ForeColor)

Dibuja un circulo. Es bastante impreciso

Parámetros

in	X	Coordenada X del centro
in	Y	Coordenada Y del centro
in	radius	Radio del circulo
in	ForeColor	Color a usar

4.5.3.4. void Lcd_Clr (void)

Limpia el buffer virtual de la pantalla

4.5.3.5. void Lcd_DisplayChar (INT16U *usX0*, INT16U *usY0*, INT8U *ForeColor*, INT8U *ucChar*)

Escribe la caracter dado, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas seran las de la esquina superior izquierda del primer caracter.

Parámetros

in	<i>usX0</i>	Coordenada X del caracter
in	<i>usY0</i>	Coordenada Y del caracter
in	<i>ForeColor</i>	Color en el que se escribira el caracter
in	<i>s</i>	Caracter a escribir

4.5.3.6. void Lcd_DisplayChar_inverted (INT16U *usX0*, INT16U *usY0*, INT8U *ForeColor*, INT8U *ucChar*)

Escribe la cadena dada, con los caracteres invertidos, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas seran las de la esquina superior izquierda del primer caracter. El caracter sera escrito en transparencia, mientras que el espacio no ocupado en el caracter se rellenara del color dado.

Parámetros

in	<i>usX0</i>	Coordenada X del caracter
in	<i>usY0</i>	Coordenada Y del caracter
in	<i>ForeColor</i>	Color en el que se rellenara el espacio libre
in	<i>ucChar</i>	Caracter a escribir

4.5.3.7. void Lcd_Dma_Trans (void)

Copia el buffer virtual al buffer real mediante DMA, bloqueante hasta terminar el mismo.

4.5.3.8. void Lcd_Dma_Trans_non_block (void)

Copia el buffer virtual al buffer real mediante DMA, no bloqueante

4.5.3.9. void Lcd_Dma_Trans_wait (void)

Espera a la finalizacion del DMA de pantalla, bloqueante

4.5.3.10. void Lcd_Draw_Box (INT16 *usLeft*, INT16 *usTop*, INT16 *usRight*, INT16 *usBottom*, INT8U *ucColor*)

Dibuja un rectangulo (vacío) con las coordenadas dadas, con el borde del color de linea dado

Parámetros

in	<i>usLeft</i>	Coordenada X de la esquina superior izquierda
in	<i>usTop</i>	Coordenada Y de la esquina superior izquierda
in	<i>usRight</i>	Coordenada X de la esquina inferior derecha
in	<i>usBottom</i>	Coordenada Y de la esquina inferior derecha
in	<i>ucColor</i>	Color a usar para el borde

4.5.3.11. void Lcd_Draw_Box_inverted (INT16 *usLeft*, INT16 *usTop*, INT16 *usRight*, INT16 *usBottom*, INT8U *ucColor*)

Invierte los colores a lo largo del borde del rectangulo dado

Parámetros

in	<i>usLeft</i>	Coordenada X de la esquina superior izquierda
in	<i>usTop</i>	Coordenada Y de la esquina superior izquierda
in	<i>usRight</i>	Coordenada X de la esquina inferior derecha
in	<i>usBottom</i>	Coordenada Y de la esquina inferior derecha
in	<i>ucColor</i>	

4.5.3.12. void Lcd_Draw_Filled_Box (INT16 *usLeft*, INT16 *usTop*, INT16 *usRight*, INT16 *usBottom*, INT8U *ucColor*)

Rellena el rectangulo determinado en el lugar dado con el color dado

Parámetros

in	<i>usLeft</i>	Coordenada X de la esquina superior izquierda
in	<i>usTop</i>	Coordenada Y de la esquina superior izquierda
in	<i>usRight</i>	Coordenada X de la esquina inferior derecha
in	<i>usBottom</i>	Coordenada Y de la esquina inferior derecha
in	<i>ucColor</i>	Color a usar para el relleno

4.5.3.13. void Lcd_Draw_HLine (INT16 *usX0*, INT16 *usX1*, INT16 *usY0*, INT8U *ucColor*, INT16U *usWidth*)

Dibuja un linea horizontal

Parámetros

in	<i>usX0</i>	Coordenada X del punto inicial
in	<i>usX1</i>	Coordenada X del punto final
in	<i>usY0</i>	Coordenada Y del punto inicial
in	<i>ucColor</i>	Color a usar
in	<i>usWidth</i>	Anchura de la linea

4.5.3.14. void Lcd_Draw_HLine_inverted (INT16 *usX0*, INT16 *usX1*, INT16 *usY0*, INT16U *usWidth*)

Invierte el color a lo largo de una linea horizontal

Parámetros

in	<i>usX0</i>	Coordenada X del punto inicial
in	<i>usX1</i>	Coordenada X del punto final
in	<i>usY0</i>	Coordenada Y del punto inicial
in	<i>usWidth</i>	Anchura de la linea

4.5.3.15. void Lcd_Draw_HLine_pointed (INT16 *usX0*, INT16 *usX1*, INT16 *usY0*, INT8U *ucColor*, INT16U *usWidth*)

Dibuja un linea vertical punteada

Parámetros

in	<i>usX0</i>	Coordenada X del punto inicial
in	<i>usX1</i>	Coordenada X del punto final
in	<i>usY0</i>	Coordenada Y del punto inicial
in	<i>ucColor</i>	Color a usar

<i>in</i>	<i>usWidth</i>	Anchura de la linea
-----------	----------------	---------------------

4.5.3.16. `void Lcd_Draw_Line (INT16 usX0, INT16 usY0, INT16 usX1, INT16 usY1, INT8U ucColor, INT16U usWidth)`

Dibuja un linea

Parámetros

<i>in</i>	<i>usX0</i>	Coordenada X del punto inicial
<i>in</i>	<i>usY0</i>	Coordenada Y del punto inicial
<i>in</i>	<i>usX1</i>	Coordenada X del punto final
<i>in</i>	<i>usY1</i>	Coordenada Y del punto final
<i>in</i>	<i>ucColor</i>	Color a usar
<i>in</i>	<i>usWidth</i>	Anchura de la linea

4.5.3.17. `void Lcd_Draw_VLine (INT16 usY0, INT16 usY1, INT16 usX0, INT8U ucColor, INT16U usWidth)`

Dibuja un linea vertical

Parámetros

<i>in</i>	<i>usY0</i>	Coordenada Y del punto inicial
<i>in</i>	<i>usY1</i>	Coordenada Y del punto final
<i>in</i>	<i>usX0</i>	Coordenada X del punto inicial
<i>in</i>	<i>ucColor</i>	Color a usar
<i>in</i>	<i>usWidth</i>	Anchura de la linea

4.5.3.18. `void Lcd_Draw_VLine_inverted (INT16 usY0, INT16 usY1, INT16 usX0, INT16U usWidth)`

Invierte el color a lo largo de una linea vertical

Parámetros

<i>in</i>	<i>usY0</i>	Coordenada Y del punto inicial
<i>in</i>	<i>usY1</i>	Coordenada Y del punto final
<i>in</i>	<i>usX0</i>	Coordenada X del punto inicial
<i>in</i>	<i>usWidth</i>	Anchura de la linea

4.5.3.19. `void Lcd_Draw_VLine_pointed (INT16 usY0, INT16 usY1, INT16 usX0, INT8U ucColor, INT16U usWidth)`

Dibuja un linea horizontal punteada

Parámetros

<i>in</i>	<i>usY0</i>	Coordenada Y del punto inicial
<i>in</i>	<i>usY1</i>	Coordenada Y del punto final
<i>in</i>	<i>usX0</i>	Coordenada X del punto inicial
<i>in</i>	<i>ucColor</i>	Color a usar
<i>in</i>	<i>usWidth</i>	Anchura de la linea

4.5.3.20. void Lcd_DspAscll8x16 (INT16U x0, INT16U y0, INT8U *ForeColor*, INT8U * s)

Escribe la cadena dada, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas seran las de la esquina superior izquierda del primer caracter

Parámetros

in	<i>x0</i>	Coordenada X del primer caracter de la cadena
in	<i>y0</i>	Coordenada Y del primer caracter de la cadena
in	<i>ForeColor</i>	Color en el que se escribirán los caracteres
in	<i>s</i>	Vector de caracteres a escribir, terminados en '\0'

4.5.3.21. void Lcd_DspAscll8x16_inverted (INT16U *x0*, INT16U *y0*, INT8U *ForeColor*, INT8U * *s*)

Escribe la cadena dada, con los caracteres invertidos, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas serán las de la esquina superior izquierda del primer caracter. El caracter será escrito en transparencia, mientras que el espacio no ocupado en el caracter se rellenará del color dado.

Parámetros

in	<i>x0</i>	Coordenada X del primer caracter de la cadena
in	<i>y0</i>	Coordenada Y del primer caracter de la cadena
in	<i>ForeColor</i>	Color en el que se rellenará el espacio libre
in	<i>s</i>	Vector de caracteres a escribir, terminados en '\0'

4.5.3.22. void Lcd_DspAscll8x16HorizontallyCentered (INT16U *y0*, INT8U *ForeColor*, INT8U * *s*)

Escribe la cadena dada, centrado horizontalmente en la pantalla, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas serán las de la esquina superior izquierda del primer caracter.

Parámetros

in	<i>y0</i>	Coordenada Y del primer caracter de la cadena
in	<i>ForeColor</i>	Color en el que se escribirá el caracter
in	<i>s</i>	Vector de caracteres a escribir, terminados en '\0'

4.5.3.23. void Lcd_DspAscll8x16HorizontallyCentered_inverted (INT16U *y0*, INT8U *ForeColor*, INT8U * *s*)

Escribe la cadena dada, centrado horizontalmente en la pantalla, con los caracteres invertidos, en una fuente de 8x16, en la coordenada dada. Las coordenadas dadas serán las de la esquina superior izquierda del primer caracter. El caracter será escrito en transparencia, mientras que el espacio no ocupado en el caracter se rellenará del color dado.

Parámetros

in	<i>y0</i>	Coordenada Y del primer caracter de la cadena
in	<i>ForeColor</i>	Color en el que se rellenará el espacio libre
in	<i>s</i>	Vector de caracteres a escribir, terminados en '\0'

4.5.3.24. INT8U LCD_GetPixel (INT16U *usX*, INT16U *usY*)

Devuelve el color en las coordenadas dadas

Parámetros

in	<i>usX</i>	Coordenada X del punto
in	<i>usY</i>	Coordenada Y del punto

Devuelve el color en el punto de la pantalla dado

Parámetros

in	<i>usX</i>	Coordenada X a observar
in	<i>usY</i>	Coordenada Y a observar

Devuelve

Color en el punto dado, o indeterminado en caso de que el punto no pertenezca a la pantalla

4.5.3.25. void Lcd_Init (void)

Inicializa el sistema de la pantalla

4.5.3.26. void LcdClrRect (INT16 *usLeft*, INT16 *usTop*, INT16 *usRight*, INT16 *usBottom*, INT8U *ucColor*)

Rellena el rectangulo determinado en el lugar dado con el color dado

Parámetros

in	<i>usLeft</i>	Coordenada X de la esquina superior izquierda
in	<i>usTop</i>	Coordenada Y de la esquina superior izquierda
in	<i>usRight</i>	Coordenada X de la esquina inferior derecha
in	<i>usBottom</i>	Coordenada Y de la esquina inferior derecha
in	<i>ucColor</i>	Color a usar para el relleno

4.5.3.27. void ReverseLine (INT32U *ulHeight*, INT32U *ulY*)

Invierte los colores a lo largo de lineas de la pantalla

Parámetros

in	<i>ulHeight</i>	Numero de lineas a invertir
in	<i>ulY</i>	Coordenada Y de la primera linea

4.5.3.28. void ReverseSquare (INT32U *ulX0*, INT32U *ulY0*, INT32U *ulX1*, INT32U *ulY1*)

Invierte los colores en el rectangulo dado

Parámetros

in	<i>ulX0</i>	Coordenada X de la esquina superior izquierda
in	<i>ulY0</i>	Coordenada Y de la esquina superior izquierda
in	<i>ulX1</i>	Coordenada X de la esquina inferior derecha
in	<i>ulY1</i>	Coordenada Y de la esquina inferior derecha

4.6. Referencia del Archivo Persistence.h

Modulo que permite una forma simple de persistencia.

```
#include "44b.h"
```

Funciones

- int [Persistence_read_int](#) (void)
- char [Persistence_read_char](#) (int n)
- void [Persistence_save_int](#) (int n)
- void [Persistence_save_char](#) (int n, char c)

4.6.1. Descripción detallada

Modulo que gestiona persistencia entre resets (Y perdidas de corriente) mediante el almacenado de informacion en los registros de alarma (rALM...), por lo que no se pueden usar a la vez que esta libreria. El espacio que ocupa el entero y los caracteres es compartido, de tal forma que los caracteres 2-5 representan los bytes del entero, respectivamente, de mas significativo a menos significativo.

Autor

Guillermo Robles Gonzalez

4.6.2. Documentación de las funciones

4.6.2.1. char Persistence_read_char (int n)

Devuelve el char almacenado

Parámetros

in	n	Numero de char a leer, rango [0,5]
----	---	------------------------------------

4.6.2.2. int Persistence_read_int (void)

Devuelve el entero almacenado ALMSEC = LSB ALMDAY = MSB

4.6.2.3. void Persistence_save_char (int n, char c)

Guarda el char dado en la posicion de memoria dada

Parámetros

in	n	Posicion en la que se guardara, rango [0,5]
in	c	Char a guardar

4.6.2.4. void Persistence_save_int (int n)

Guarda el entero pasado, sobrescribiendo el valor anterior

Parámetros

in	n	Entero a guardar
----	---	------------------

4.7. Referencia del Archivo still-alive-lyrics.h

Archivo de recursos con los credits finales.

'defines'

- `#define STILL_ALIVE_SIZE 117`

Variables

- `char * still_alive [STILL_ALIVE_SIZE]`

4.7.1. Descripción detallada**Autor**

Guillermo Robles

4.8. Referencia del Archivo sudoku_2015.h

Modulo que contiene funciones relacionadas con el analisis de sudokus.

```
#include <inttypes.h>
```

'defines'

- `#define CELDA uint16_t`

Enumeraciones

- `enum { NUM_FILAS = 9, NUM_COLUMNS = 16, TAM_REGION = 3, NUM_REGION = 3 }`

Informacion de la cuadrícula.

Funciones

- `void init_game (void)`
- `void celda_cambiar_candidatos (uint8_t valor, CELDA cuadrícula[NUM_FILAS][NUM_COLUMNS], uint8_t fila, uint8_t columna)`
- `int sudoku_candidatos_arm (CELDA cuadrícula[NUM_FILAS][NUM_COLUMNS], uint8_t fila, uint8_t columna)`
- `int sudoku_recalcular (CELDA cuadrícula[NUM_FILAS][NUM_COLUMNS])`
- `int celda_es_error (CELDA celda)`
- `int celda_es_pista (CELDA celda)`
- `int celda_es_candidato (CELDA celda, uint8_t valor)`
- `void celda_poner_valor (CELDA *celdaptr, uint8_t val)`
- `uint8_t celda_leer_valor (CELDA celda)`
- `void sudoku_empty_grid (CELDA cuadrícula[NUM_FILAS][NUM_COLUMNS])`

4.8.1. Descripción detallada**Autor**

Guillermo Robles Gonzalez

4.8.2. Documentación de las funciones

4.8.2.1. `void celda_cambiar_candidatos (uint8_t valor, CELDA cuadrícula[NUM_FILAS][NUM_COLUMNAS], uint8_t fila, uint8_t columna) [inline]`

Funcion que modifica los posibles candidatos de una celda, eliminando una pista dada

Parámetros

in	<i>valor</i>	Pista que se eliminara
in	<i>cuadricula</i>	Cuadricula a modificar
in	<i>fila</i>	Coordenada fila de la casilla a modificar
in	<i>columna</i>	Coordenada columna de la casilla a modificar

4.8.2.2. int celda_es_candidato (CELDA *celda*, uint8_t *valor*) [inline]

Devuelve un numero mayor que cero en caso de que la celda sea pista inicial, devuelve 0 en caso contrario

Parámetros

in	<i>celda</i>	Celda a comprobar
----	--------------	-------------------

Devuelve

0 si y solo si la celda no es pista

4.8.2.3. int celda_es_error (CELDA *celda*) [inline]

Devuelve un numero mayor de 0 si y solo si la celda esta marcada como error, devuelve 0 en caso contrario

Parámetros

in	<i>celda</i>	Celda a comprobar
----	--------------	-------------------

Devuelve

0 en caso de que la celda sea correcta

4.8.2.4. int celda_es_pista (CELDA *celda*) [inline]

Devuelve un numero mayor que cero en caso de que la celda sea pista inicial, devuelve 0 en caso contrario

Parámetros

in	<i>celda</i>	Celda a comprobar
----	--------------	-------------------

Devuelve

0 si y solo si la celda no es pista

4.8.2.5. uint8_t celda_leer_valor (CELDA *celda*) [inline]

Devuelve el numero en la celda dada (0 si es vacia)

Parámetros

out	<i>celda</i>	Celda a consultar
-----	--------------	-------------------

4.8.2.6. void celda_poner_valor (CELDA * *celdaptr*, uint8_t *val*) [inline]

Actualiza el valor de la celda dada al valor dado

Parámetros

out	<i>celda</i>	Celda a cambiar
in	<i>val</i>	Nuevo valor

4.8.2.7. void init_game (void)

Inicializa el juego

4.8.2.8. int sudoku_candidatos_arm (CELDA *cuadrícula*[NUM_FILAS][NUM_COLUMNAS], uint8_t *fila*, uint8_t *columna*)

Funcion que dado una casilla en una cuadrícula, actualiza sus candidatos

Parámetros

in	<i>cuadrícula</i>	Cuadrícula a modificar
in	<i>fila</i>	Coordenada fila de la casilla a modificar
in	<i>columna</i>	Coordenada columna de la casilla a modificar

Devuelve

0 en caso de que la casilla este vacia, !=0 en caso contrario

4.8.2.9. void sudoku_empty_grid (CELDA *cuadrícula*[NUM_FILAS][NUM_COLUMNAS])

Pone cualquier casilla que no sea pista a 0

Parámetros

out	<i>cuadrícula</i>	Cuadrícula a editar
-----	-------------------	---------------------

4.8.2.10. int sudoku_recalcular (CELDA *cuadrícula*[NUM_FILAS][NUM_COLUMNAS])

Funcion que dado una cuadrícula, actualiza todos sus candidatos y reajusta los errores como sea necesario, usando como hoja una funcion ARM.

Parámetros

out	<i>cuadrícula</i>	Cuadrícula a modificar
-----	-------------------	------------------------

Devuelve

Numero de casillas vacias, o -1 en caso de que haya errores

4.9. Referencia del Archivo sudoku_collection_san.h

Fichero de recursos que contiene cuadrículas.

```
#include "sudoku_2015.h"
```

```
'defines'
```

```
■ #define SUDOKU_COLLECTION_SAN_H_
```


- #define **NUM_CUADRICULAS** 10
- #define **COMPRESSED_SIZE** 41

Funciones

- void [sudoku_collection_descomprime](#) (char *compressed, CELDA destiny[NUM_FILAS][NUM_COLUMNAS])

Variables

- char [cuadriculas](#) [NUM_CUADRICULAS][COMPRESSED_SIZE]
Coleccion de cuadriculas comprimidas.
- char [cuadriculaCasiResuelta](#) [COMPRESSED_SIZE]
Cuadricula especial casi resuelta.

4.9.1. Descripción detallada

Modulo que contiene cuadriculas en formato comprimido, ademas de funciones de descompresion

Autor

Guillermo Robles Gonzalez

4.9.2. Documentación de las funciones

4.9.2.1. void [sudoku_collection_descomprime](#) (char * *compressed*, CELDA *destiny*[NUM_FILAS][NUM_COLUMNAS])

Funcion que convierte del formato comprimido al formato normal de uso

Parámetros

in	<i>compressed</i>	Cuadricula compresada
out	<i>destiny</i>	Zona de memoria en la que se descomprimira la cuadricula

4.9.3. Documentación de las variables

4.9.3.1. char [cuadriculaCasiResuelta](#)[COMPRESSED_SIZE]

Cuadricula especial en la cual solo resta introducir un valor

4.9.3.2. char [cuadriculas](#)[NUM_CUADRICULAS][COMPRESSED_SIZE]

Coleccion de cuadriculas en formato compresado

4.10. Referencia del Archivo sudoku_graphics.h

Modulo que actua de capa de abstraccion entre el juego y la pantalla.

```
#include "sudoku_2015.h"
#include "44blib.h"
#include "44b.h"
#include "def.h"
```

'defines'

- #define SUDOKU_NUM_CUADS 9
- #define SUDOKU_SQUARE_SIZE 18
- #define SUDOKU_X0 20
- #define SUDOKU_Y0 20
- #define SUDOKU_FONT_HEIGHT 16
- #define SUDOKU_FONT_LENGTH 8
- #define ASCII_NUMBER_BASE 48

Funciones

- void [sudoku_graphics_draw_base](#) ()
- void [sudoku_graphics_fill_from_data](#) (CELDA cuadrícula[NUM_FILAS][NUM_COLUMNAS])
- void [sudoku_graphics_update_lcd](#) ()
- void [sudoku_graphics_print_final_screen](#) (int tiempo_juego_s, int tiempo_calculos_ms, int last_record_s, int errores)
- void [sudoku_graphics_put_number_in_square](#) (INT8 x, INT8 y, INT8 number, INT8U ucColor)
- void [sudoku_graphics_remark_square](#) (INT8 x, INT8 y)
- void [sudoku_graphics_mark_error](#) (INT8 x, INT8 y, INT8 error)
- void [sudoku_graphics_mark_error_in_square](#) (INT8 x, INT8 y, INT8 error)
- void [sudoku_graphics_print_title_screen](#) ()
- void [sudoku_graphics_print_instructions](#) ()
- void [sudoku_graphics_print_record](#) ()
- void [sudoku_graphics_print_still_alive](#) (int lineNumber)
- void [sudoku_graphics_draw_state](#) (int state, int number)
- void [sudoku_graphics_draw_time](#) (int time_playing_s, int time_calculating_ms)

4.10.1. Descripción detallada

Modulo que contiene un conjunto de funciones que interactuan con el aspecto grafico del sudoku, y que abstraen las operaciones sobre la pantalla en un conjunto de operaciones de alto nivel

Autor

Guillermo Robles Gonzalez

4.10.2. Documentación de las funciones**4.10.2.1. void [sudoku_graphics_draw_base](#) ()**

Dibuja la base del sudoku (cuadrícula, numeración de la misma, frase de información)

4.10.2.2. void [sudoku_graphics_draw_state](#) (int *state*, int *number*)

Dibuja el estado de selección en el que nos encontramos state=0 => esperando fila state=1 => esperando columna state=2 => esperando valor

Parámetros

in	<i>state</i>	Estado actual, en forma textual
in	<i>number</i>	Valor actual del boton

4.10.2.3. void sudoku_graphics_draw_time (int *time_playing_s*, int *time_calculating_ms*)

Dibuja el tiempo, tanto el pasado como el de los calculos

Parámetros

in	<i>time_playing_s</i>	Tiempo de juego, en segundos
in	<i>time_calculating_ms</i>	Tiempo de calculo, en milisegundos

4.10.2.4. void sudoku_graphics_fill_from_data (CELDA *cuadrícula*[NUM_FILAS][NUM_COLUMNAS])

Rellena el sudoku con la informacion de la cuadrícula dada

Parámetros

in	<i>cuadrícula</i>	Cuadrícula de la que se cojera la informacion
----	-------------------	---

4.10.2.5. void sudoku_graphics_mark_error (INT8 *x*, INT8 *y*, INT8 *error*) [inline]

Marca el error dado en la casilla dada (Negando los colores en su interior)

Parámetros

in	<i>x</i>	Posicion x de la casilla
in	<i>y</i>	Posicion y de la casilla
in	<i>error</i>	Error a marcar (intervalo 1-9)

4.10.2.6. void sudoku_graphics_mark_error_in_square (INT8 *x*, INT8 *y*, INT8 *error*) [inline]

Remarca una marca de error en una casilla

Parámetros

in	<i>x</i>	Posicion x de la casilla
in	<i>y</i>	Posicion y de la casilla
in	<i>error</i>	Error a remarcar

4.10.2.7. void sudoku_graphics_print_final_screen (int *tiempo_juego_s*, int *tiempo_calculos_ms*, int *last_record_s*, int *errores*)

Imprime la pantalla final, si el numero de errores es 0, se imprimira la pantalla de exito, en caso contrario se imprimira un mensaje de fracaso.

Parámetros

in	<i>tiempo_juego_ms</i>	Tiempo que ha durado el juego, en ms
----	------------------------	--------------------------------------

in	<i>tiempo_calculos_ _ms</i>	Tiempo que han durado los calculos, en ms
in	<i>tiempo_ renderizado_ms</i>	Tiempo de renderizado, en ms
in	<i>errores</i>	Numero de errores actual

4.10.2.8. void sudoku_graphics_print_instructions ()

Imprime la pantalla de instrucciones

4.10.2.9. void sudoku_graphics_print_record ()

Imprime la pantalla de instrucciones

4.10.2.10. void sudoku_graphics_print_still_alive (int lineNumber)

Imprime 6 lineas de Still Alive a partir de la linea dada

Parámetros

in	<i>lineNumber</i>	Linea a partir de la cual comenzar a imprimir
----	-------------------	---

4.10.2.11. void sudoku_graphics_print_title_screen ()

Imprime la pantalla de titulo

4.10.2.12. void sudoku_graphics_put_number_in_square (INT8 x, INT8 y, INT8 number, INT8U ucColor) [inline]

Rellena una casilla con el numero dado

Parámetros

in	<i>x</i>	Posicion x de la casilla a rellenar
in	<i>y</i>	Posicion y de la casilla a rellenar
in	<i>number</i>	Numero con el cual rellenarla
in	<i>ucColor</i>	Color de letra

4.10.2.13. void sudoku_graphics_remark_square (INT8 x, INT8 y) [inline]

Remarca una casilla de forma visible al usuario (Aplicando un enmarcado)

Parámetros

in	<i>x</i>	Posicion x de la casilla
in	<i>y</i>	Posicion y de la casilla

4.10.2.14. void sudoku_graphics_update_lcd ()

Actualiza el LCD con la informacion del sudoku

4.11. Referencia del Archivo Timer2.h

Modulo que gestiona un contador.

```
#include "44b.h"
```

Funciones

- void [Timer2_Inicializar](#) (void)
- void [Timer2_Empezar](#) (void)
- void [Timer2_Reiniciar](#) (void)
- uint32_t [Timer2_Leer](#) (void)

4.11.1. Descripción detallada

Modulo que gestiona un timer de precision milisegundos, usa el timer 2

Autor

Guillermo Robles Gonzalez

4.11.2. Documentación de las funciones

4.11.2.1. void [Timer2_Empezar](#) (void)

Funcion que comienza la cuenta.

4.11.2.2. void [Timer2_Inicializar](#) (void)

Funcion que inicializa el contador

4.11.2.3. uint32_t [Timer2_Leer](#) (void)

Devuelve el contador interno en microsegundos

Devuelve

Tiempo en microsegundos desde el ultimo reinicio del contador

4.11.2.4. void [Timer2_Reiniciar](#) (void)

Funcion que resetea el contador interno a 0

4.12. Referencia del Archivo title-bitmap.h

Archivo de recursos con la imagen del titulo.

```
#include "lcd.h"  
#include "bmp.h"
```

'defines'

- `#define TITLE_BITMAP_HEIGHT 47`
- `#define TITLE_BITMAP_WIDTH 195`

Variables

- `const INT8U titleBitmap [TITLE_BITMAP_HEIGHT *TITLE_BITMAP_WIDTH]`
- [STRU_BITMAP](#) `Stru_Bitmap_title`

Índice alfabético

8led.h, [7](#)
 D8Led_blink_symbol, [7](#)
 D8Led_current_symbol, [7](#)
 D8Led_init, [7](#)
 D8Led_symbol, [8](#)

action
 Button.h, [11](#)

aperture-logo-bitmap.h, [8](#)

BITMAP, [5](#)

BitmapPop
 Bmp.h, [9](#)

BitmapPush
 Bmp.h, [9](#)

BitmapView
 Bmp.h, [10](#)

Bmp.h, [8](#)
 BitmapPop, [9](#)
 BitmapPush, [9](#)
 BitmapView, [10](#)
 CursorPush, [10](#)
 CursorView, [10](#)

Button.h, [10](#)
 action, [11](#)
 Button_init, [11](#)
 Button_low_next, [11](#)
 Button_low_update_screen, [11](#)
 Button_next, [11](#)
 Button_reconfigure_range, [11](#)
 Button_set_valor_actual, [11](#)
 Button_update_screen, [12](#)
 Button_valor_actual, [12](#)

Button_init
 Button.h, [11](#)

Button_low_next
 Button.h, [11](#)

Button_low_update_screen
 Button.h, [11](#)

Button_next
 Button.h, [11](#)

Button_reconfigure_range
 Button.h, [11](#)

Button_set_valor_actual
 Button.h, [11](#)

Button_update_screen
 Button.h, [12](#)

Button_valor_actual
 Button.h, [12](#)

celda_cambiar_candidatos
 sudoku_2015.h, [24](#)

celda_es_candidato
 sudoku_2015.h, [25](#)

celda_es_error
 sudoku_2015.h, [25](#)

celda_es_pista
 sudoku_2015.h, [25](#)

celda_leer_valor
 sudoku_2015.h, [25](#)

celda_poner_valor
 sudoku_2015.h, [25](#)

cuadrículaCasiResuelta
 sudoku_collection_san.h, [27](#)

cuadrículas
 sudoku_collection_san.h, [27](#)

CursorPush
 Bmp.h, [10](#)

CursorView
 Bmp.h, [10](#)

D8Led_blink_symbol
 8led.h, [7](#)

D8Led_current_symbol
 8led.h, [7](#)

D8Led_init
 8led.h, [7](#)

D8Led_symbol
 8led.h, [8](#)

init_game
 sudoku_2015.h, [26](#)

LCD_Active_PutPixel
 lcd.h, [14](#)

LCD_GetPixel
 lcd.h, [20](#)

LCD_PutPixel
 lcd.h, [14](#)

lcd.h, [12](#)
 LCD_Active_PutPixel, [14](#)
 LCD_GetPixel, [20](#)
 LCD_PutPixel, [14](#)
 Lcd_Active_Clr, [14](#)
 Lcd_Circle, [14](#)
 Lcd_Circle_Filled, [14](#)
 Lcd_Clr, [14](#)
 Lcd_DisplayChar, [14](#)
 Lcd_DisplayChar_inverted, [16](#)
 Lcd_Dma_Trans, [16](#)

- Lcd_Dma_Trans_non_block, 16
- Lcd_Dma_Trans_wait, 16
- Lcd_Draw_Box, 16
- Lcd_Draw_Box_inverted, 16
- Lcd_Draw_Filled_Box, 17
- Lcd_Draw_HLine, 17
- Lcd_Draw_HLine_inverted, 17
- Lcd_Draw_HLine_pointed, 17
- Lcd_Draw_Line, 18
- Lcd_Draw_VLine, 18
- Lcd_Draw_VLine_inverted, 18
- Lcd_Draw_VLine_pointed, 18
- Lcd_DspAscll8x16, 18
- Lcd_DspAscll8x16_inverted, 20
- Lcd_DspAscll8x16HorizontallyCentered, 20
- Lcd_DspAscll8x16HorizontallyCentered_inverted, 20
- Lcd_Init, 21
- LcdClrRect, 21
- ReverseLine, 21
- ReverseSquare, 21
- Lcd_Active_Clr
 - lcd.h, 14
- Lcd_Circle
 - lcd.h, 14
- Lcd_Circle_Filled
 - lcd.h, 14
- Lcd_Clr
 - lcd.h, 14
- Lcd_DisplayChar
 - lcd.h, 14
- Lcd_DisplayChar_inverted
 - lcd.h, 16
- Lcd_Dma_Trans
 - lcd.h, 16
- Lcd_Dma_Trans_non_block
 - lcd.h, 16
- Lcd_Dma_Trans_wait
 - lcd.h, 16
- Lcd_Draw_Box
 - lcd.h, 16
- Lcd_Draw_Box_inverted
 - lcd.h, 16
- Lcd_Draw_Filled_Box
 - lcd.h, 17
- Lcd_Draw_HLine
 - lcd.h, 17
- Lcd_Draw_HLine_inverted
 - lcd.h, 17
- Lcd_Draw_HLine_pointed
 - lcd.h, 17
- Lcd_Draw_Line
 - lcd.h, 18
- Lcd_Draw_VLine
 - lcd.h, 18
- Lcd_Draw_VLine_inverted
 - lcd.h, 18
- Lcd_Draw_VLine_pointed
 - lcd.h, 18
- Lcd_DspAscll8x16
 - lcd.h, 18
- Lcd_DspAscll8x16_inverted
 - lcd.h, 20
- Lcd_DspAscll8x16HorizontallyCentered
 - lcd.h, 20
- Lcd_DspAscll8x16HorizontallyCentered_inverted
 - lcd.h, 20
- Lcd_Init
 - lcd.h, 21
- LcdClrRect
 - lcd.h, 21
- Persistence.h, 21
 - Persistence_read_char, 22
 - Persistence_read_int, 22
 - Persistence_save_char, 22
 - Persistence_save_int, 22
- Persistence_read_char
 - Persistence.h, 22
- Persistence_read_int
 - Persistence.h, 22
- Persistence_save_char
 - Persistence.h, 22
- Persistence_save_int
 - Persistence.h, 22
- ReverseLine
 - lcd.h, 21
- ReverseSquare
 - lcd.h, 21
- still-alive-lyrics.h, 22
- sudoku_2015.h, 23
 - celda_cambiar_candidatos, 24
 - celda_es_candidato, 25
 - celda_es_error, 25
 - celda_es_pista, 25
 - celda_leer_valor, 25
 - celda_poner_valor, 25
 - init_game, 26
 - sudoku_candidatos_arm, 26
 - sudoku_empty_grid, 26
 - sudoku_recalcular, 26
- sudoku_candidatos_arm
 - sudoku_2015.h, 26
- sudoku_collection_descomprime
 - sudoku_collection_san.h, 27
- sudoku_collection_san.h, 26
 - cuadrículaCasiResuelta, 27
 - cuadrículas, 27
 - sudoku_collection_descomprime, 27
- sudoku_empty_grid
 - sudoku_2015.h, 26
- sudoku_graphics.h, 27
 - sudoku_graphics_draw_base, 28
 - sudoku_graphics_draw_state, 28
 - sudoku_graphics_draw_time, 29

- sudoku_graphics_fill_from_data, [29](#)
- sudoku_graphics_mark_error, [29](#)
- sudoku_graphics_mark_error_in_square, [29](#)
- sudoku_graphics_print_final_screen, [29](#)
- sudoku_graphics_print_instructions, [30](#)
- sudoku_graphics_print_record, [30](#)
- sudoku_graphics_print_still_alive, [30](#)
- sudoku_graphics_print_title_screen, [30](#)
- sudoku_graphics_put_number_in_square, [30](#)
- sudoku_graphics_remark_square, [30](#)
- sudoku_graphics_update_lcd, [30](#)
- sudoku_graphics_draw_base
 - sudoku_graphics.h, [28](#)
- sudoku_graphics_draw_state
 - sudoku_graphics.h, [28](#)
- sudoku_graphics_draw_time
 - sudoku_graphics.h, [29](#)
- sudoku_graphics_fill_from_data
 - sudoku_graphics.h, [29](#)
- sudoku_graphics_mark_error
 - sudoku_graphics.h, [29](#)
- sudoku_graphics_mark_error_in_square
 - sudoku_graphics.h, [29](#)
- sudoku_graphics_print_final_screen
 - sudoku_graphics.h, [29](#)
- sudoku_graphics_print_instructions
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_print_record
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_print_still_alive
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_print_title_screen
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_put_number_in_square
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_remark_square
 - sudoku_graphics.h, [30](#)
- sudoku_graphics_update_lcd
 - sudoku_graphics.h, [30](#)
- sudoku_recalcular
 - sudoku_2015.h, [26](#)
- Timer2.h, [31](#)
 - Timer2_Empazar, [31](#)
 - Timer2_Inicializar, [31](#)
 - Timer2_Leer, [31](#)
 - Timer2_Reiniciar, [31](#)
- Timer2_Empazar
 - Timer2.h, [31](#)
- Timer2_Inicializar
 - Timer2.h, [31](#)
- Timer2_Leer
 - Timer2.h, [31](#)
- Timer2_Reiniciar
 - Timer2.h, [31](#)
- title-bitmap.h, [31](#)