

# Proyecto Hardware

Generado por Doxygen 1.8.6

Jueves, 17 de Diciembre de 2015 19:57:34



# Índice general

<b>1</b>	<b>Índice de estructura de datos</b>	<b>1</b>
1.1	Estructura de datos	1
<b>2</b>	<b>Índice de archivos</b>	<b>3</b>
2.1	Lista de archivos	3
<b>3</b>	<b>Documentación de las estructuras de datos</b>	<b>5</b>
3.1	Referencia de la Estructura BITMAP	5
<b>4</b>	<b>Documentación de archivos</b>	<b>7</b>
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 Bmp.h	8
4.2.1	Descripción detallada	9
4.2.2	Documentación de las funciones	9
4.2.2.1	BitmapPop	9
4.2.2.2	BitmapPush	9
4.2.2.3	BitmapView	9
4.2.2.4	CursorPush	9
4.2.2.5	CursorView	9
4.3	Referencia del Archivo Button.h	10
4.3.1	Descripción detallada	10
4.3.2	Documentación de las funciones	10
4.3.2.1	action	10
4.3.2.2	Button_init	10
4.3.2.3	Button_low_next	10
4.3.2.4	Button_low_update_screen	11
4.3.2.5	Button_next	11

4.3.2.6	Button_set_valor_actual . . . . .	11
4.3.2.7	Button_update_screen . . . . .	11
4.3.2.8	Button_valor_actual . . . . .	11
4.4	Referencia del Archivo lcd.h . . . . .	11
4.4.1	Descripción detallada . . . . .	13
4.4.2	Documentación de los 'defines' . . . . .	13
4.4.2.1	LCD_Active_PutPixel . . . . .	13
4.4.2.2	LCD_PutPixel . . . . .	13
4.5	Referencia del Archivo sudoku_2015.h . . . . .	13
4.5.1	Descripción detallada . . . . .	14
4.5.2	Documentación de las funciones . . . . .	14
4.5.2.1	celda_cambiar_candidatos . . . . .	14
4.5.2.2	celda_es_candidato . . . . .	14
4.5.2.3	celda_es_error . . . . .	14
4.5.2.4	celda_es_pista . . . . .	15
4.5.2.5	celda_leer_valor . . . . .	15
4.5.2.6	celda_poner_valor . . . . .	15
4.5.2.7	init_game . . . . .	15
4.5.2.8	sudoku_candidatos_arm . . . . .	15
4.5.2.9	sudoku_recalcular . . . . .	16
4.5.2.10	sudoku_vacia_tabla . . . . .	17
4.6	Referencia del Archivo sudoku_collection_san.h . . . . .	17
4.6.1	Descripción detallada . . . . .	17
4.6.2	Documentación de las funciones . . . . .	18
4.6.2.1	sudoku_collection_descomprime . . . . .	18
4.6.3	Documentación de las variables . . . . .	19
4.6.3.1	cuadriculaCasiResuelta . . . . .	19
4.6.3.2	cuadriculas . . . . .	19
4.7	Referencia del Archivo sudoku_graphics.h . . . . .	19
4.7.1	Descripción detallada . . . . .	20
4.7.2	Documentación de las funciones . . . . .	20
4.7.2.1	sudoku_graphics_draw_base . . . . .	20
4.7.2.2	sudoku_graphics_draw_state . . . . .	20
4.7.2.3	sudoku_graphics_fill_from_data . . . . .	20
4.7.2.4	sudoku_graphics_mark_error . . . . .	20
4.7.2.5	sudoku_graphics_print_final_screen . . . . .	20
4.7.2.6	sudoku_graphics_print_instructions . . . . .	21
4.7.2.7	sudoku_graphics_print_still_alive . . . . .	21
4.7.2.8	sudoku_graphics_print_title_screen . . . . .	21
4.7.2.9	sudoku_graphics_put_number_in_square . . . . .	21

4.7.2.10	sudoku_graphics_remark_error_in_square . . . . .	21
4.7.2.11	sudoku_graphics_remark_square . . . . .	21
4.7.2.12	sudoku_graphics_update_lcd . . . . .	22
4.8	Referencia del Archivo Timer2.h . . . . .	22
4.8.1	Descripción detallada . . . . .	22
4.8.2	Documentación de las funciones . . . . .	22
4.8.2.1	Timer2_Empezar . . . . .	22
4.8.2.2	Timer2_Inicializar . . . . .	22
4.8.2.3	Timer2_Leer . . . . .	22
4.8.2.4	Timer2_Reiniciar . . . . .	22
<b>Índice</b>		<b>23</b>



# Capítulo 1

## Índice de estructura de datos

### 1.1. Estructura de datos

Lista de estructuras con una breve descripción:

<a href="#">BITMAP</a> . . . . .	5
----------------------------------	---





## Capítulo 2

# Indice de archivos

### 2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

<a href="#">8led.h</a>	Funciones de control del display 8-segmentos . . . . .	7
<a href="#">Bmp.h</a>	Definicion mapas de bits del LCD . . . . .	8
<a href="#">Button.h</a>	Modulo que gestiona los rebotes de los botones . . . . .	10
<a href="#">lcd.h</a>	Funciones de visualizacion y control LCD . . . . .	11
<a href="#">sudoku_2015.h</a>	Modulo que contiene funciones relacionadas con el analisis de sudokus . . . . .	13
<a href="#">sudoku_collection_san.h</a>	Fichero de recursos que contiene cuadrículas . . . . .	17
<a href="#">sudoku_graphics.h</a>	Modulo que actua de capa de abstraccion entre el juego y la pantalla . . . . .	19
<a href="#">Timer2.h</a>	Modulo que gestiona un contador . . . . .	22



## 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

<code>in</code>	<code>value</code>	Valor a poner en el 8 segmentos
-----------------	--------------------	---------------------------------

## 4.2. 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.*

#### 4.2.1. Descripción detallada

Autor

#### 4.2.2. Documentación de las funciones

##### 4.2.2.1. 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

##### 4.2.2.2. 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	Stru_Bitmap	Bitmap to draw

##### 4.2.2.3. 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	Stru_Bitmap	Bitmap to draw

##### 4.2.2.4. 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

##### 4.2.2.5. 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

### 4.3. 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](#) ()

#### 4.3.1. 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

#### 4.3.2. Documentación de las funciones

##### 4.3.2.1. void action ( int *n* )

Accion asociada a la pulsacion del boton de codigo n

**Parámetros**

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

##### 4.3.2.2. void Button\_init ( int *min*, int *max* )

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

**Parámetros**

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

##### 4.3.2.3. void Button\_low\_next ( void )

Baja el flag interno de next



**4.3.2.4. void Button\_low\_update\_screen ( )**

Baja el flag interno indicando que se han realizado acciones

**4.3.2.5. 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

**4.3.2.6. 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
-----------	----------	---------------------

**4.3.2.7. 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

**4.3.2.8. int Button\_valor\_actual ( void )**

Valor actual de la cuenta interna

**Devuelve**

valor actual

**4.4. 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\_Clr** (void)
- **void Lcd\_Test** (void)
- **void Lcd\_Dma\_Trans** (void)
- **void LcdVirtualToTrue** (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\_Anti\_Disp** (INT16U usX0, INT16U usY0, INT16U usX1, INT16U usY1)
- void **Lcd\_DisplayChar** (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U ucChar)
- void **Lcd\_DisplayChar\_inverted** (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U ucChar)
- void **Lcd\_DisplayShort** (INT16 sX, INT16 sY, INT16U usInt)
- void **Lcd\_Circle** (INT8 X, INT8 Y, INT16 radius, INT8U ForeColor)
- void **Zdma0Done** (void) \_\_attribute\_\_((interrupt("IRQ")))
- void **Lcd\_DspAscll6x8** (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U \*pucChar)
- void **Lcd\_DspAscll8x16** (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\_DspHz16** (INT16U x0, INT16U y0, INT8U ForeColor, INT8U \*s)
- void **ReverseLine** (INT32U ulHeight, INT32U ulY)
- void **ReverseSquare** (INT32U ulX0, INT32U ulY0, INT32U ulX1, INT32U ulY1)

#### 4.4.1. Descripción detallada

Versión

<P6-ARM>

#### 4.4.2. Documentación de los 'defines'

##### 4.4.2.1. #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))
```

##### 4.4.2.2. #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. 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\_COLUMNAS** = 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\_COLUMNAS], uint8\_t fila, uint8\_t columna)
- int `sudoku_candidatos_arm` (CELDA cuadrícula[NUM\_FILAS][NUM\_COLUMNAS], uint8\_t fila, uint8\_t columna)
- int `sudoku_recalcular` (CELDA cuadrícula[NUM\_FILAS][NUM\_COLUMNAS])
- 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_vacia_tabla` (CELDA cuadrícula[NUM\_FILAS][NUM\_COLUMNAS])

### 4.5.1. Descripción detallada

#### Autor

Guillermo Robles Gonzalez

### 4.5.2. Documentación de las funciones

4.5.2.1. 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>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

4.5.2.2. 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

4.5.2.3. 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**

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

**Devuelve**

0 en caso de que la celda sea correcta

**4.5.2.4. 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**

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

**Devuelve**

0 si y solo si la celda no es pista

**4.5.2.5. uint8\_t celda\_leer\_valor ( CELDA *celda* ) [inline]**

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

**Parámetros**

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

**4.5.2.6. void celda\_poner\_valor ( CELDA \* *celdaptr*, uint8\_t *val* ) [inline]**

Actualiza el valor de la celda dada al valor dado

**Parámetros**

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

**4.5.2.7. void init\_game ( void )**

Inicializa el juego

**4.5.2.8. 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**

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

**Devuelve**

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

4.5.2.9. `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 vacías, o -1 en caso de que haya errores

4.5.2.10. void sudoku\_vacia\_tabla ( CELDA *cuadrícula*[NUM\_FILAS][NUM\_COLUMNAS] )

Pone cualquier casilla que no sea pista a 0

## Parámetros

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

## 4.6. 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 [cuadrículas](#) [NUM\_CUADRICULAS][COMPRESSED\_SIZE]  
*Colección de cuadrículas comprimidas.*
- char [cuadrículaCasiResuelta](#) [COMPRESSED\_SIZE]  
*Cuadrícula especial casi resuelta.*

## 4.6.1. Descripción detallada

Módulo que contiene cuadrículas en formato comprimido, además de funciones de descompresión

## Autor

Guillermo Robles Gonzalez

#### 4.6.2. Documentación de las funciones

4.6.2.1. 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>	Cuadrícula compresada
out	<i>destiny</i>	Zona de memoria en la que se descomprimirá la cuadrícula

## 4.6.3. Documentación de las variables

## 4.6.3.1. char cuadrículaCasiResuelta[COMPRESSED\_SIZE]

Cuadrícula especial en la cual solo resta introducir un valor

## 4.6.3.2. char cuadrículas[NUM\_CUADRICULAS][COMPRESSED\_SIZE]

Colección de cuadrículas en formato comprimido

## 4.7. Referencia del Archivo sudoku\_graphics.h

Módulo que actúa de capa de abstracción 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 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\\_remark\\_error\\_in\\_square](#) (INT8 x, INT8 y, INT8 error)
- void [sudoku\\_graphics\\_print\\_title\\_screen](#) ()
- void [sudoku\\_graphics\\_print\\_instructions](#) ()
- void [sudoku\\_graphics\\_print\\_still\\_alive](#) (int lineNumber)
- void [sudoku\\_graphics\\_draw\\_state](#) (int state, int number)

#### 4.7.1. Descripción detallada

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

Autor

Guillermo Robles Gonzalez

#### 4.7.2. Documentación de las funciones

##### 4.7.2.1. void sudoku\_graphics\_draw\_base ( )

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

##### 4.7.2.2. 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	state	Estado actual, en forma textual
in	number	Valor actual del botón

##### 4.7.2.3. void sudoku\_graphics\_fill\_from\_data ( CELDA cuadrícula[NUM\_FILAS][NUM\_COLUMNAS] )

Rellena el sudoku con la información de la cuadrícula dada

Parámetros

in	cuadrícula	Cuadrícula de la que se cojera la información
----	------------	---

##### 4.7.2.4. 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	x	Posición x de la casilla
in	y	Posición y de la casilla
in	error	Error a marcar (intervalo 1-9)

##### 4.7.2.5. void sudoku\_graphics\_print\_final\_screen ( int tiempo\_juego\_s, int tiempo\_calculos\_ms, int errores )

Imprime la pantalla final, si el número de errores es 0, se imprimirá la pantalla de éxito, en caso contrario se imprimirá un mensaje de fracaso.

Parámetros

in	tiempo_juego_s - ms	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

#### 4.7.2.6. void sudoku\_graphics\_print\_instructions ( )

Imprime la pantalla de instrucciones

#### 4.7.2.7. 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
----	-------------------	---

#### 4.7.2.8. void sudoku\_graphics\_print\_title\_screen ( )

Imprime la pantalla de titulo

#### 4.7.2.9. 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

#### 4.7.2.10. void sudoku\_graphics\_remark\_error\_in\_square ( INT8 x, INT8 y, INT8 error )

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

#### 4.7.2.11. 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	y	Posicion y de la casilla
----	---	--------------------------

#### 4.7.2.12. void sudoku\_graphics\_update\_lcd ( )

Actualiza el LCD con la informacion del sudoku

## 4.8. 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)

#### 4.8.1. Descripción detallada

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

Autor

Guillermo Robles Gonzalez

#### 4.8.2. Documentación de las funciones

##### 4.8.2.1. void Timer2\_Empezar ( void )

Funcion que comienza la cuenta.

##### 4.8.2.2. void Timer2\_Inicializar ( void )

Funcion que inicializa el contador

##### 4.8.2.3. uint32\_t Timer2\_Leer ( void )

Devuelve el contador interno en microsegundos

Devuelve

Tiempo en microsegundos desde el ultimo reinicio del contador

##### 4.8.2.4. void Timer2\_Reiniciar ( void )

Funcion que resetea el contador interno a 0

# Índice alfabético

- 8led.h, [7](#)
  - D8Led\_blink\_symbol, [7](#)
  - D8Led\_current\_symbol, [7](#)
  - D8Led\_init, [7](#)
  - D8Led\_symbol, [8](#)
- action
  - Button.h, [10](#)
- BITMAP, [5](#)
- BitmapPop
  - Bmp.h, [9](#)
- BitmapPush
  - Bmp.h, [9](#)
- BitmapView
  - Bmp.h, [9](#)
- Bmp.h, [8](#)
  - BitmapPop, [9](#)
  - BitmapPush, [9](#)
  - BitmapView, [9](#)
  - CursorPush, [9](#)
  - CursorView, [9](#)
- Button.h, [10](#)
  - action, [10](#)
  - Button\_init, [10](#)
  - Button\_low\_next, [10](#)
  - Button\_low\_update\_screen, [10](#)
  - Button\_next, [11](#)
  - Button\_set\_valor\_actual, [11](#)
  - Button\_update\_screen, [11](#)
  - Button\_valor\_actual, [11](#)
- Button\_init
  - Button.h, [10](#)
- Button\_low\_next
  - Button.h, [10](#)
- Button\_low\_update\_screen
  - Button.h, [10](#)
- Button\_next
  - Button.h, [11](#)
- Button\_set\_valor\_actual
  - Button.h, [11](#)
- Button\_update\_screen
  - Button.h, [11](#)
- Button\_valor\_actual
  - Button.h, [11](#)
- celda\_cambiar\_candidatos
  - sudoku\_2015.h, [14](#)
- celda\_es\_candidato
  - sudoku\_2015.h, [14](#)
- celda\_es\_error
  - sudoku\_2015.h, [14](#)
- celda\_es\_pista
  - sudoku\_2015.h, [15](#)
- celda\_leer\_valor
  - sudoku\_2015.h, [15](#)
- celda\_poner\_valor
  - sudoku\_2015.h, [15](#)
- cuadrículaCasiResuelta
  - sudoku\_collection\_san.h, [19](#)
- cuadrículas
  - sudoku\_collection\_san.h, [19](#)
- CursorPush
  - Bmp.h, [9](#)
- CursorView
  - Bmp.h, [9](#)
- 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, [15](#)
- LCD\_Active\_PutPixel
  - lcd.h, [13](#)
- LCD\_PutPixel
  - lcd.h, [13](#)
- lcd.h, [11](#)
  - LCD\_Active\_PutPixel, [13](#)
  - LCD\_PutPixel, [13](#)
- sudoku\_2015.h, [13](#)
  - celda\_cambiar\_candidatos, [14](#)
  - celda\_es\_candidato, [14](#)
  - celda\_es\_error, [14](#)
  - celda\_es\_pista, [15](#)
  - celda\_leer\_valor, [15](#)
  - celda\_poner\_valor, [15](#)
  - init\_game, [15](#)
  - sudoku\_candidatos\_arm, [15](#)
  - sudoku\_recalcular, [15](#)
  - sudoku\_vacia\_tabla, [17](#)
- sudoku\_candidatos\_arm
  - sudoku\_2015.h, [15](#)

sudoku\_collection\_descomprime  
    sudoku\_collection\_san.h, [18](#)  
sudoku\_collection\_san.h, [17](#)  
    cuadrículaCasiResuelta, [19](#)  
    cuadriculas, [19](#)  
    sudoku\_collection\_descomprime, [18](#)  
sudoku\_graphics.h, [19](#)  
    sudoku\_graphics\_draw\_base, [20](#)  
    sudoku\_graphics\_draw\_state, [20](#)  
    sudoku\_graphics\_fill\_from\_data, [20](#)  
    sudoku\_graphics\_mark\_error, [20](#)  
    sudoku\_graphics\_print\_final\_screen, [20](#)  
    sudoku\_graphics\_print\_instructions, [21](#)  
    sudoku\_graphics\_print\_still\_alive, [21](#)  
    sudoku\_graphics\_print\_title\_screen, [21](#)  
    sudoku\_graphics\_put\_number\_in\_square, [21](#)  
    sudoku\_graphics\_remark\_error\_in\_square, [21](#)  
    sudoku\_graphics\_remark\_square, [21](#)  
    sudoku\_graphics\_update\_lcd, [22](#)  
sudoku\_graphics\_draw\_base  
    sudoku\_graphics.h, [20](#)  
sudoku\_graphics\_draw\_state  
    sudoku\_graphics.h, [20](#)  
sudoku\_graphics\_fill\_from\_data  
    sudoku\_graphics.h, [20](#)  
sudoku\_graphics\_mark\_error  
    sudoku\_graphics.h, [20](#)  
sudoku\_graphics\_print\_final\_screen  
    sudoku\_graphics.h, [20](#)  
sudoku\_graphics\_print\_instructions  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_print\_still\_alive  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_print\_title\_screen  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_put\_number\_in\_square  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_remark\_error\_in\_square  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_remark\_square  
    sudoku\_graphics.h, [21](#)  
sudoku\_graphics\_update\_lcd  
    sudoku\_graphics.h, [22](#)  
sudoku\_recalcular  
    sudoku\_2015.h, [15](#)  
sudoku\_vacia\_tabla  
    sudoku\_2015.h, [17](#)  
  
Timer2.h, [22](#)  
    Timer2\_Empezar, [22](#)  
    Timer2\_Inicializar, [22](#)  
    Timer2\_Leer, [22](#)  
    Timer2\_Reiniciar, [22](#)  
Timer2\_Empezar  
    Timer2.h, [22](#)  
Timer2\_Inicializar  
    Timer2.h, [22](#)  
Timer2\_Leer  
    Timer2.h, [22](#)  
  
Timer2\_Reiniciar  
    Timer2.h, [22](#)