Proyecto Hardware

Generado por Doxygen 1.8.6

Sábado, 19 de Diciembre de 2015 11:20:58

Índice general

1	Índi	ce de es	structura	de datos	1
	1.1	Estruc	tura de da	atos	1
2	India	ce de ai	chivos		3
	2.1	Lista d	le archivos	s	3
3	Doc	umenta	ción de la	as estructuras de datos	5
	3.1	Refere	ncia de la	Estructura BITMAP	5
4	Doc	umenta	ción de a	rchivos	7
	4.1	Refere	ncia del A	Archivo 8led.h	7
		4.1.1	Docume	entación 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	Refere	ncia del A	Archivo Bmp.h	8
		4.2.1	Descripo	ción detallada	9
		4.2.2	Docume	entación 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	Refere	ncia del A	Archivo Button.h	10
		4.3.1	Descripo	ción detallada	10
		4.3.2	Docume	entación 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

IV ÍNDICE GENERAL

		4.3.2.6	Button_reconfigure_range	11
		4.3.2.7	Button_set_valor_actual	11
		4.3.2.8	Button_update_screen	11
		4.3.2.9	Button_valor_actual	11
4.4	Refere	ncia del Aı	rchivo lcd.h	11
	4.4.1	Descripc	ión detallada	13
	4.4.2	Documer	ntación de los 'defines'	13
		4.4.2.1	LCD_Active_PutPixel	13
		4.4.2.2	LCD_PutPixel	13
4.5	Refere	ncia del Aı	rchivo sudoku_2015.h	13
	4.5.1	Descripc	ión detallada	14
	4.5.2	Documer	ntación 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	15
		4.5.2.4	celda_es_pista	16
		4.5.2.5	celda_leer_valor	16
		4.5.2.6	celda_poner_valor	16
		4.5.2.7	init_game	16
		4.5.2.8	sudoku_candidatos_arm	16
		4.5.2.9	sudoku_empty_grid	17
		4.5.2.10	sudoku_recalcular	18
4.6	Refere	ncia del Aı	rchivo sudoku_collection_san.h	18
	4.6.1	Descripc	ión detallada	18
	4.6.2		ntación de las funciones	19
		4.6.2.1	sudoku_collection_descomprime	19
	4.6.3	Documer	ntación de las variables	20
		4.6.3.1	cuadriculaCasiResuelta	20
		4.6.3.2	cuadriculas	20
4.7	Refere		rchivo sudoku_graphics.h	20
	4.7.1	Descripc	ión detallada	21
	4.7.2	Documer	ntación de las funciones	21
		4.7.2.1	sudoku_graphics_draw_base	21
		4.7.2.2	sudoku_graphics_draw_state	21
		4.7.2.3	sudoku_graphics_draw_time	21
		4.7.2.4	sudoku_graphics_fill_from_data	21
		4.7.2.5	sudoku_graphics_mark_error	21
		4.7.2.6	sudoku_graphics_print_final_screen	22
		4.7.2.7	sudoku_graphics_print_instructions	23
		4.7.2.8	sudoku_graphics_print_still_alive	23

ÍNDICE GENERAL v

		4.7.2.9	sudoku_graphics_print_title_screen	23
		4.7.2.10	sudoku_graphics_put_number_in_square	23
		4.7.2.11	sudoku_graphics_remark_error_in_square	23
		4.7.2.12	sudoku_graphics_remark_square	23
		4.7.2.13	sudoku_graphics_update_lcd	24
4.8	Refere	ncia del Ar	rchivo Timer2.h	24
	4.8.1	Descripci	ión detallada	24
	4.8.2	Documer	ntación de las funciones	24
		4.8.2.1	Timer2_Empezar	24
		4.8.2.2	Timer2_Inicializar	24
		4.8.2.3	Timer2_Leer	24
		4.8.2.4	Timer2_Reiniciar	24
Índice				25

Capítulo 1

Índice de estructura de datos

1.1.	1.1. Estructura de datos	
Lista d	de estructuras con una breve descripción:	
RI.	ΤΜΔΡ	,

2	Índice de estructura de datos

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

Blea.n	
Funciones de control del display 8-segmentos	7
aperture-logo-bitmap.h	
Bmp.h	
Definicion mapas de bits del LCD	8
Button.h	
Modulo que gestiona los rebotes de los botones	10
cd.h	
Funciones de visualizacion y control LCD	11
still-alive-lyrics.h	??
sudoku_2015.h	
Modulo que contiene funciones relacionadas con el analisis de sudokus	13
sudoku_collection_san.h	
Fichero de recursos que contiene cuadriculas	18
sudoku_graphics.h	
Modulo que actua de capa de abstraccion entre el juego y la pantalla	20
Timer2.h	
Modulo que gestiona un contador	24
title-bitmap.h	??
common/ 44b.h	??
common/44blib.h	??
common/ def.h	
common/ option.h	

Indice de archivos

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 ulTransColor
- INT8U * pucStart

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

■ Bmp.h

datos

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 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	у	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	у	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	Х	X coordinate of the position to draw
in	У	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	у	Y coordinate of the position to push

4.2.2.5. void CursorView (INT16U x, INT16U y)

in	X	X coordinate of the position to draw
in	у	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 ()
- void Button_reconfigure_range (int new_min, int new_max)

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	n	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	min	Valor minimo que aparecera en la pantalla
in	max	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_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

in	new_min	Nuevo minimo
in	new_max	Nuevo maximo

4.3.2.7. 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

in	n	nuevo valor a poner
----	---	---------------------

4.3.2.8. 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

4.3.2.9. int Button_valor_actual (void)

Valor actual de la cuenta interna

Devuelve

valor actual

4.4. Referencia del Archivo Icd.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) & 0x1fffff)
- #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) $\{a^{\wedge}=b; b^{\wedge}=a; a^{\wedge}=b;\}$

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 LcdCirRect (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_DspAscII6x8 (INT16U usX0, INT16U usY0, INT8U ForeColor, INT8U *pucChar)
- void Lcd_DspAscll8x16 (INT16U x0, INT16U y0, INT8U ForeColor, INT8U *s)
- void Lcd_DspAscII8x16HorizontallyCentered (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 uIX0, INT32U uIY0, INT32U uIX1, INT32U uIY1)

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 cuadricula.

Funciones

- void init_game (void)
- void celda_cambiar_candidatos (uint8_t valor, CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS], uint8_t fila, uint8_t columna)
- int sudoku_candidatos_arm (CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS], uint8_t fila, uint8_t columna)
- int sudoku_recalcular (CELDA cuadricula[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_empty_grid (CELDA cuadricula[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 cuadricula[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	valor	Pista que se eliminara
in	cuadricula	Cuadricula a modificar
in	fila	Coordenada fila de la casilla a modificar
in	columna	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 celda 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

in	celda	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

in	celda	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

out	celda	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

out	celda	Celda a cambiar
in	val	Nuevo valor

4.5.2.7. void init_game (void)

Inicializa el juego

4.5.2.8. int sudoku_candidatos_arm (CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS], uint8_t fila, uint8_t columna)

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

Parámetros

in	cuadricula	Cuadricula a modificar
in	fila	Coordenada fila de la casilla a modificar
in	columna	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. void sudoku_empty_grid (CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS])

Pone cualquier casilla que no sea pista a 0

out	cuadricuala	Cuadricula a editar
-----	-------------	---------------------

4.5.2.10. int sudoku_recalcular (CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS])

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

Parámetros

out	cuadricula	Cuadricula a modificar
-----	------------	------------------------

Devuelve

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

4.6. Referencia del Archivo sudoku_collection_san.h

Fichero de recursos que contiene cuadriculas.

```
#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_COLUMNA-S])

Variables

char cuadriculas [NUM_CUADRICULAS][COMPRESSED_SIZE]

Coleccion de cuadriculas comprimidas.

char cuadriculaCasiResuelta [COMPRESSED_SIZE]

Cuadricula especial casi resuelta.

4.6.1. Descripción detallada

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

Autor

Guillermo Robles Gonzalez

- 4.6.2. Documentación de las funciones
- $4.6.2.1. \quad void\ sudoku_collection_descomprime\ (\ char*{\it compressed},\ CELDA\ {\it destiny[NUM_FILAS][NUM_COLUMNAS]}\)$

Funcion que convierte del formato comprimido al formato normal de uso

in	compressed	Cuadricula compresa
out	destiny	Zona de memoria en la que se descomprimira la cuadricula

4.6.3. Documentación de las variables

4.6.3.1. char cuadriculaCasiResuelta[COMPRESSED_SIZE]

Cuadricula especial en la cual solo resta introducir un valor

4.6.3.2. char cuadriculas[NUM_CUADRICULAS][COMPRESSED_SIZE]

Coleccion de cuadriculas en formato compreso

4.7. 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 cuadricula[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)
- void sudoku_graphics_draw_time (int time_playing_s, int time_calculating_ms)

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 (cuadricula, numeracion de la misma, frase de informacion)

4.7.2.2. void sudoku_graphics_draw_state (int state, int number)

Dibuja el estado de seleccion 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 boton

4.7.2.3. 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	time_playing_s	Tiempo de juego, en segundos
in	time_calculating-	Tiempo de calculo, en milisegundos
	_ms	

4.7.2.4. void sudoku_graphics_fill_from_data (CELDA cuadricula[NUM_FILAS][NUM_COLUMNAS])

Rellena el sudoku con la informacion de la cuadricula dada

Parámetros

in	cuadricula	Cuadricula de la que se cojera la informacion

4.7.2.5. 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	X	Posicion x de la casilla
in	У	Posicion y de la casilla
in	error	Error a marcar (intervalo 1-9)

4.7.2.6. void sudoku_graphics_print_final_screen (int tiempo_juego_s, int tiempo_calculos_ms, 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.

in	tiempo_juego	Tiempo que ha durado el juego, en ms
	ms	
in	tiempo_calculos-	Tiempo que han durado los calculos, en ms
	_ms	
in	tiempo	Tiempo de renderizado, en ms
	renderizado_ms	
in	errores	Numero de errores actual

4.7.2.7. void sudoku_graphics_print_instructions ()

Imprime la pantalla de instrucciones

4.7.2.8. void sudoku_graphics_print_still_alive (int lineNumber)

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

Parámetros

in	lineNumber	Linea a partir de la cual comenzar a imprimir

4.7.2.9. void sudoku_graphics_print_title_screen ()

Imprime la pantalla de titulo

4.7.2.10. 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	X	Posicion x de la casilla a rellenar
in	у	Posicion y de la casilla a rellenar
in	number	Numero con el cual rellenarla
in	ucColor	Color de letra

4.7.2.11. void sudoku_graphics_remark_error_in_square (INT8 x, INT8 y, INT8 error)

Remarca una marca de error en una casilla

Parámetros

in	X	Posicion x de la casilla
in	у	Posicion y de la casilla
in	error	Error a remarcar

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

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

in	X	Posicion x de la casilla
in	У	Posicion y de la casilla

4.7.2.13. 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	sudoku_2015.h, 14
D8Led_blink_symbol, 7	celda_es_candidato
D8Led_current_symbol, 7	sudoku 2015.h, 14
D8Led_init, 7	celda_es_error
D8Led_symbol, 8	 sudoku_2015.h, 14
<u>-</u> -,, -	celda_es_pista
action	sudoku_2015.h, 16
Button.h, 10	celda_leer_valor
	sudoku_2015.h, 16
BITMAP, 5	celda_poner_valor
BitmapPop	sudoku_2015.h, 16
Bmp.h, 9	cuadriculaCasiResuelta
BitmapPush	sudoku collection san.h, 20
Bmp.h, 9	cuadriculas
BitmapView	
Bmp.h, 9	sudoku_collection_san.h, 20
Bmp.h, 8	CursorPush
BitmapPop, 9	Bmp.h, 9
BitmapPush, 9	CursorView
BitmapView, 9	Bmp.h, 9
CursorPush, 9	DOI and blinds assembled
CursorView, 9	D8Led_blink_symbol
Button.h, 10	8led.h, 7
action, 10	D8Led_current_symbol
	8led.h, 7
Button_init, 10	D8Led_init
Button_low_next, 10	8led.h, 7
Button_low_update_screen, 10	D8Led_symbol
Button_next, 11	8led.h, 8
Button_reconfigure_range, 11	
Button_set_valor_actual, 11	init_game
Button_update_screen, 11	sudoku_2015.h, 16
Button_valor_actual, 11	
Button_init	LCD_Active_PutPixel
Button.h, 10	lcd.h, 13
Button_low_next	LCD_PutPixel
Button.h, 10	lcd.h, 13
Button_low_update_screen	lcd.h, 11
Button.h, 10	LCD_Active_PutPixel, 13
Button_next	LCD_PutPixel, 13
Button.h, 11	
Button_reconfigure_range	sudoku_2015.h, 13
Button.h, 11	celda_cambiar_candidatos, 14
Button_set_valor_actual	celda_es_candidato, 14
Button.h, 11	celda_es_error, 14
Button_update_screen	celda_es_pista, 16
Button.h, 11	celda_leer_valor, 16
Button valor actual	celda_poner_valor, 16
Button.h, 11	init_game, 16
_ 5,	sudoku candidatos arm, 16
celda_cambiar_candidatos	sudoku_empty_grid, 16

sudoku_recalcular, 18 Timer2 Empezar sudoku candidatos arm sudoku_2015.h, 16 sudoku_collection_descomprime sudoku_collection_san.h, 19 sudoku collection san.h, 18 cuadriculaCasiResuelta, 20 cuadriculas, 20 sudoku collection descomprime, 19 sudoku empty grid sudoku 2015.h, 16 sudoku_graphics.h, 20 sudoku_graphics_draw_base, 21 sudoku graphics draw state, 21 sudoku_graphics_draw_time, 21 sudoku_graphics_fill_from_data, 21 sudoku graphics mark error, 21 sudoku graphics print final screen, 21 sudoku_graphics_print_instructions, 23 sudoku_graphics_print_still_alive, 23 sudoku graphics print title screen, 23 sudoku graphics put number in square, 23 sudoku_graphics_remark_error_in_square, 23 sudoku_graphics_remark_square, 23 sudoku graphics update lcd, 24 sudoku_graphics_draw_base sudoku_graphics.h, 21 sudoku graphics draw state sudoku graphics.h, 21 sudoku graphics draw time sudoku_graphics.h, 21 sudoku_graphics_fill_from_data sudoku graphics.h, 21 sudoku_graphics_mark_error sudoku_graphics.h, 21 sudoku_graphics_print_final_screen sudoku graphics.h, 21 sudoku_graphics_print_instructions sudoku_graphics.h, 23 sudoku graphics print still alive sudoku graphics.h, 23 sudoku_graphics_print_title_screen sudoku_graphics.h, 23 sudoku graphics put number in square sudoku graphics.h, 23 sudoku_graphics_remark_error_in_square sudoku_graphics.h, 23 sudoku graphics remark square sudoku graphics.h, 23 sudoku_graphics_update_lcd sudoku_graphics.h, 24 sudoku recalcular sudoku_2015.h, 18 Timer2.h, 24 Timer2 Empezar, 24 Timer2 Inicializar, 24 Timer2_Leer, 24 Timer2_Reiniciar, 24

Timer2.h, 24 Timer2 Inicializar Timer2.h, 24 Timer2 Leer Timer2.h, 24 Timer2 Reiniciar Timer2.h, 24