

# TRABAJO 1

Generado por Doxygen 1.8.10

Sábado, 19 de Septiembre de 2015 03:04:02



# Índice general

<b>1</b>	<b>TRABAJO1</b>	<b>1</b>
<b>2</b>	<b>Índice de estructura de datos</b>	<b>3</b>
2.1	Estructura de datos . . . . .	3
<b>3</b>	<b>Índice de archivos</b>	<b>5</b>
3.1	Lista de archivos . . . . .	5
<b>4</b>	<b>Documentación de las estructuras de datos</b>	<b>7</b>
4.1	Referencia de la Estructura ins_t . . . . .	7
4.1.1	Documentación de los campos . . . . .	7
4.1.1.1	array . . . . .	7
4.2	Referencia de la Estructura instruction_t . . . . .	7
4.2.1	Documentación de los campos . . . . .	7
4.2.1.1	mnemonic . . . . .	7
4.2.1.2	op1_type . . . . .	7
4.2.1.3	op1_value . . . . .	8
4.2.1.4	op2_type . . . . .	8
4.2.1.5	op2_value . . . . .	8
4.2.1.6	op3_type . . . . .	8
4.2.1.7	op3_value . . . . .	8
<b>5</b>	<b>Documentación de archivos</b>	<b>9</b>
5.1	Referencia del Archivo decoder.c . . . . .	9
5.1.1	Documentación de las funciones . . . . .	9
5.1.1.1	countLines(FILE *fp) . . . . .	9
5.1.1.2	decodeInstruction(instruction_t instruction, uint32_t *registros, uint32_t *Banderas) . . . . .	9
5.1.1.3	getInstruction(char *instStr) . . . . .	9
5.1.1.4	readFile(char *filename, ins_t *instructions) . . . . .	9
5.2	Referencia del Archivo decoder.h . . . . .	10
5.2.1	Documentación de las funciones . . . . .	10
5.2.1.1	countLines(FILE *fp) . . . . .	10
5.2.1.2	decodeInstruction(instruction_t instruction, uint32_t *registros, uint32_t *Banderas) . . . . .	10

5.2.1.3	getInstruction(char *instStr)	10
5.2.1.4	readFile(char *filename, ins_t *instructions)	10
5.3	Referencia del Archivo funcBanderas.c	10
5.3.1	Documentación de las funciones	11
5.3.1.1	funcBanderas(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas)	11
5.3.1.2	funcBanderas2(uint32_t *rd, uint32_t a, uint32_t *Banderas)	11
5.4	Referencia del Archivo funcBanderas.h	11
5.4.1	Documentación de las funciones	12
5.4.1.1	funcBanderas(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas)	12
5.4.1.2	funcBanderas2(uint32_t *rd, uint32_t a, uint32_t *Banderas)	13
5.5	Referencia del Archivo funcioninterfaz.c	13
5.5.1	Documentación de las funciones	13
5.5.1.1	iniciarinterfaz()	13
5.5.1.2	mostrarRegistros(uint32_t *registros, size_t t)	13
5.5.1.3	refreshScreen(void)	14
5.6	Referencia del Archivo funcioninterfaz.h	14
5.6.1	Documentación de las funciones	14
5.6.1.1	iniciarinterfaz()	14
5.6.1.2	mostrarRegistros(uint32_t *registros, size_t t)	14
5.6.1.3	refreshScreen(void)	15
5.6.1.4	terminarinterfaz()	15
5.7	Referencia del Archivo funcionregistro.c	15
5.7.1	Documentación de las funciones	15
5.7.1.1	mostrarbanderas(uint32_t *banderas, size_t t)	15
5.8	Referencia del Archivo funcionregistro.h	16
5.8.1	Documentación de las funciones	16
5.8.1.1	mostrarbanderas(uint32_t *banderas, size_t t2)	16
5.8.1.2	mostrarregistros(uint32_t *registros, size_t t)	16
5.9	Referencia del Archivo instrucciones.c	16
5.9.1	Documentación de las funciones	17
5.9.1.1	ADC(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas)	17
5.9.1.2	ADD(uint32_t *rd, uint32_t a, uint32_t b)	18
5.9.1.3	AND(uint32_t *rd, uint32_t a, uint32_t b)	18
5.9.1.4	ASR(uint32_t *rd, uint32_t a, uint32_t b)	18
5.9.1.5	BIC(uint32_t *rd, uint32_t a, uint32_t b)	18
5.9.1.6	CMN(uint32_t a, uint32_t b, uint32_t *Banderas)	19
5.9.1.7	CMP(uint32_t a, uint32_t b, uint32_t *Banderas)	19
5.9.1.8	EOR(uint32_t *rd, uint32_t a, uint32_t b)	19
5.9.1.9	LSL(uint32_t *rd, uint32_t a, uint32_t b)	19
5.9.1.10	LSR(uint32_t *rd, uint32_t a, uint32_t b)	20

5.9.1.11	MOV(uint32_t *rd, uint32_t a) . . . . .	20
5.9.1.12	MUL(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	20
5.9.1.13	MVN(uint32_t *rd, uint32_t a) . . . . .	20
5.9.1.14	NOP() . . . . .	21
5.9.1.15	ORR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	21
5.9.1.16	REV(uint32_t *rd, uint32_t a) . . . . .	21
5.9.1.17	REV16(uint32_t *rd, uint32_t a) . . . . .	21
5.9.1.18	REVSH(uint32_t *rd, uint32_t a) . . . . .	22
5.9.1.19	ROR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	22
5.9.1.20	RSB(uint32_t *rd, uint32_t a) . . . . .	22
5.9.1.21	SBC(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	22
5.9.1.22	SUB(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	23
5.9.1.23	TST(uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	23
5.10	Referencia del Archivo instrucciones.h . . . . .	23
5.10.1	Documentación de las funciones . . . . .	24
5.10.1.1	ADC(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	24
5.10.1.2	ADD(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	25
5.10.1.3	AND(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	25
5.10.1.4	ASR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	25
5.10.1.5	BIC(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	25
5.10.1.6	CMN(uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	26
5.10.1.7	CMP(uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	26
5.10.1.8	EOR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	26
5.10.1.9	LSL(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	26
5.10.1.10	LSR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	27
5.10.1.11	MOV(uint32_t *rd, uint32_t a) . . . . .	27
5.10.1.12	MUL(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	27
5.10.1.13	MVN(uint32_t *rd, uint32_t a) . . . . .	27
5.10.1.14	NOP() . . . . .	28
5.10.1.15	ORR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	28
5.10.1.16	REV(uint32_t *rd, uint32_t a) . . . . .	28
5.10.1.17	REV16(uint32_t *rd, uint32_t a) . . . . .	28
5.10.1.18	REVSH(uint32_t *rd, uint32_t a) . . . . .	29
5.10.1.19	ROR(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	29
5.10.1.20	RSB(uint32_t *rd, uint32_t a) . . . . .	29
5.10.1.21	SBC(uint32_t *rd, uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	29
5.10.1.22	SUB(uint32_t *rd, uint32_t a, uint32_t b) . . . . .	30
5.10.1.23	TST(uint32_t a, uint32_t b, uint32_t *Banderas) . . . . .	30
5.11	Referencia del Archivo main.c . . . . .	30
5.11.1	Documentación de las funciones . . . . .	31

5.11.1.1 main(void) . . . . .	31
5.12 Referencia del Archivo README.md . . . . .	31

## **Capítulo 1**

# **TRABAJO1**

Maria Camila Herrera Ramos cod. 1113308334 Jhon fredy Muñoz Rodriguez cod. 1085689993





## Capítulo 2

# Índice de estructura de datos

### 2.1. Estructura de datos

Lista de estructuras con una breve descripción:

<a href="#">ins_t</a> . . . . .	7
<a href="#">instruction_t</a> . . . . .	7



## Capítulo 3

# Indice de archivos

### 3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

<a href="#">decoder.c</a>	9
<a href="#">decoder.h</a>	10
<a href="#">funcBanderas.c</a>	10
<a href="#">funcBanderas.h</a>	11
<a href="#">funcioninterfaz.c</a>	13
<a href="#">funcioninterfaz.h</a>	14
<a href="#">funcionregistro.c</a>	15
<a href="#">funcionregistro.h</a>	16
<a href="#">instrucciones.c</a>	16
<a href="#">instrucciones.h</a>	23
<a href="#">main.c</a>	30



## Capítulo 4

# Documentación de las estructuras de datos

### 4.1. Referencia de la Estructura ins\_t

```
#include <decoder.h>
```

#### Campos de datos

- char \*\* [array](#)

#### 4.1.1. Documentación de los campos

##### 4.1.1.1. char\*\* array

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

- [decoder.h](#)

### 4.2. Referencia de la Estructura instruction\_t

```
#include <decoder.h>
```

#### Campos de datos

- char [mnemonic](#) [10]
- char [op1\\_type](#)
- char [op2\\_type](#)
- char [op3\\_type](#)
- uint32\_t [op1\\_value](#)
- uint32\_t [op2\\_value](#)
- uint32\_t [op3\\_value](#)

#### 4.2.1. Documentación de los campos

##### 4.2.1.1. char mnemonic[10]

##### 4.2.1.2. char op1\_type

4.2.1.3. uint32\_t op1\_value

4.2.1.4. char op2\_type

4.2.1.5. uint32\_t op2\_value

4.2.1.6. char op3\_type

4.2.1.7. uint32\_t op3\_value

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

- [decoder.h](#)

## Capítulo 5

# Documentación de archivos

### 5.1. Referencia del Archivo decoder.c

```
#include "decoder.h"
#include "instrucciones.h"
#include "funcBanderas.h"
```

#### Funciones

- void [decodeInstruction](#) ([instruction\\_t](#) instruction, uint32\_t \*registros, uint32\_t \*Banderas)
- [instruction\\_t getInstruction](#) (char \*instStr)  
*Obtiene la instrucción separada por partes.*
- int [readFile](#) (char \*filename, [ins\\_t](#) \*instructions)
- int [countLines](#) (FILE \*fp)

#### 5.1.1. Documentación de las funciones

5.1.1.1. int [countLines](#) ( FILE \* *fp* )

5.1.1.2. void [decodeInstruction](#) ( [instruction\\_t](#) *instruction*, uint32\_t \* *registros*, uint32\_t \* *Banderas* )

5.1.1.3. [instruction\\_t getInstruction](#) ( char \* *instStr* )

Obtiene la instrucción separada por partes.

#### Parámetros

<i>instStr</i>	cadena que contiene la instrucción.
----------------	-------------------------------------

#### Devuelve

[instruction\\_t](#) la instrucción separada por partes.

5.1.1.4. int [readFile](#) ( char \* *filename*, [ins\\_t](#) \* *instructions* )

## 5.2. Referencia del Archivo decoder.h

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include "funcBanderas.h"
```

### Estructuras de datos

- struct [ins\\_t](#)
- struct [instruction\\_t](#)

### Funciones

- void [decodeInstruction](#) ([instruction\\_t](#) instruction, uint32\_t \*registros, uint32\_t \*Banderas)
- [instruction\\_t getInstruction](#) (char \*instStr)  
*Obtiene la instrucción separada por partes.*
- int [readFile](#) (char \*filename, [ins\\_t](#) \*instructions)
- int [countLines](#) (FILE \*fp)

### 5.2.1. Documentación de las funciones

5.2.1.1. int [countLines](#) ( FILE \* *fp* )

5.2.1.2. void [decodeInstruction](#) ( [instruction\\_t](#) *instruction*, uint32\_t \* *registros*, uint32\_t \* *Banderas* )

5.2.1.3. [instruction\\_t getInstruction](#) ( char \* *instStr* )

Obtiene la instrucción separada por partes.

Parámetros

<i>instStr</i>	cadena que contiene la instrucción.
----------------	-------------------------------------

Devuelve

[instruction\\_t](#) la instrucción separada por partes.

5.2.1.4. int [readFile](#) ( char \* *filename*, [ins\\_t](#) \* *instructions* )

## 5.3. Referencia del Archivo funcBanderas.c

```
#include "funcBanderas.h"
#include "instrucciones.h"
#include "funcionregistro.h"
```



## Funciones

- void `funcBanderas` (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Funcion funcBanderas.*
- void `funcBanderas2` (uint32\_t \*rd, uint32\_t a, uint32\_t \*Banderas)  
*Funcion funcBanderas2.*

### 5.3.1. Documentación de las funciones

#### 5.3.1.1. void funcBanderas ( uint32\_t \* rd, uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

Funcion funcBanderas.

##### Parámetros

<code>uint32_t*</code>	rd valor en donde se guarda el resultado
<code>uint32_t</code>	a primer valor que se va utilizar
<code>uint32_t</code>	b segundo valor que se va a utilizar
<code>uint32_t*</code>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

#### 5.3.1.2. void funcBanderas2 ( uint32\_t \* rd, uint32\_t a, uint32\_t \* Banderas )

Funcion funcBanderas2.

##### Parámetros

<code>uint32_t*</code>	rd valor en donde se guarda el resultado
<code>uint32_t</code>	a primer valor que se va utilizar
<code>uint32_t*</code>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

## 5.4. Referencia del Archivo funcBanderas.h

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
```

## Funciones

- void `funcBanderas` (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Funcion funcBanderas.*
- void `funcBanderas2` (uint32\_t \*rd, uint32\_t a, uint32\_t \*Banderas)  
*Funcion funcBanderas2.*

#### 5.4.1. Documentación de las funciones

5.4.1.1. `void funcBanderas ( uint32_t * rd, uint32_t a, uint32_t b, uint32_t * Banderas )`

Funcion funcBanderas.

## Parámetros

<i>uint32_t*</i>	rd valor en donde se guarda el resultado
<i>uint32_t</i>	a primer valor que se va utilizar
<i>uint32_t</i>	b segundo valor que se va a utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

## Devuelve

0

5.4.1.2. void funcBanderas2 ( uint32\_t \* rd, uint32\_t a, uint32\_t \* Banderas )

Funcion funcBanderas2.

## Parámetros

<i>uint32_t*</i>	rd valor en donde se guarda el resultado
<i>uint32_t</i>	a primer valor que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

## Devuelve

0

## 5.5. Referencia del Archivo funcioninterfaz.c

```
#include "instrucciones.h"
#include "funcBanderas.h"
#include "funcionregistro.h"
```

## Funciones

- void [iniciarinterfaz](#) ()  
*Funcion iniciarinterfaz.*
- void [refreshScreen](#) (void)  
*Funcion refreshScreen.*
- void [mostrarRegistros](#) (uint32\_t \*registros, size\_t t)  
*Funcion mostrarRegistros.*

### 5.5.1. Documentación de las funciones

5.5.1.1. void iniciarinterfaz ( )

Funcion iniciarinterfaz.

## Devuelve

0

5.5.1.2. void mostrarRegistros ( uint32\_t \* registros, size\_t t )

Funcion mostrarRegistros.

**Parámetros**

<i>uint32_t*</i>	puntero, registros primera variable que se va utilizar
<i>size_t</i>	t tamaño del arreglo,segunda variable que se va utilizar

**Devuelve**

0

**5.5.1.3. void refreshScreen ( void )**

Funcion refreshScreen.

**Devuelve**

0

**5.6. Referencia del Archivo funcioninterfaz.h**

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <curses.h>
```

**Funciones**

- void [iniciarinterfaz](#) ()  
*Funcion iniciarinterfaz.*
- void [terminarinterfaz](#) ()  
*Funcion terminarinterfaz.*
- void [mostrarRegistros](#) (uint32\_t \*registros, size\_t t)  
*Funcion mostrarRegistros.*
- void [refreshScreen](#) (void)  
*Funcion refreshScreen.*

**5.6.1. Documentación de las funciones****5.6.1.1. void iniciarinterfaz ( )**

Funcion iniciarinterfaz.

**Devuelve**

0

**5.6.1.2. void mostrarRegistros ( uint32\_t \* registros, size\_t t )**

Funcion mostrarRegistros.

## Parámetros

<i>uint32_t*</i>	puntero, registros primera variable que se va utilizar
<i>size_t</i>	t tamaño del arreglo,segunda variable que se va utilizar

## Devuelve

0

## 5.6.1.3. void refreshScreen ( void )

Funcion refreshScreen.

## Devuelve

0

## 5.6.1.4. void terminarinterfaz ( )

Funcion terminarinterfaz.

## Devuelve

0

## 5.7. Referencia del Archivo funcionregistro.c

```
#include "funcionregistro.h"
#include "instrucciones.h"
#include "funcBanderas.h"
```

## Funciones

- void [mostrarbanderas](#) (uint32\_t \*banderas, size\_t t)

*Funcion que muestra banderas.*

## 5.7.1. Documentación de las funciones

## 5.7.1.1. void mostrarbanderas ( uint32\_t \* banderas, size\_t t2 )

Funcion que muestra banderas.

## Parámetros

<i>uint32_t*</i>	banderas puntero a la primera posicion del arreglo banderas
<i>size_t</i>	t2 tamaño del arreglo banderas

## Devuelve

0

## 5.8. Referencia del Archivo funcionregistro.h

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <curses.h>
```

### Funciones

- void [mostrarregistros](#) (uint32\_t \*registros, size\_t t)  
*Funcion que muestra los registros.*
- void [mostrarbanderas](#) (uint32\_t \*banderas, size\_t t2)  
*Funcion que muestra banderas.*

### 5.8.1. Documentación de las funciones

#### 5.8.1.1. void [mostrarbanderas](#) ( uint32\_t \* *banderas*, size\_t *t2* )

Funcion que muestra banderas.

##### Parámetros

<i>uint32_t*</i>	banderas puntero a la primera posicion del arreglo banderas
<i>size_t</i>	t2 tamaño del arreglo banderas

##### Devuelve

0

#### 5.8.1.2. void [mostrarregistros](#) ( uint32\_t \* *registros*, size\_t *t* )

Funcion que muestra los registros.

##### Parámetros

<i>uint32_t*</i>	registros puntero a la primera posicion del arreglo registros
<i>size_t</i>	t tamaño del arreglo registros

##### Devuelve

0

## 5.9. Referencia del Archivo instrucciones.c

```
#include "funcionregistro.h"
#include "funcBanderas.h"
#include <stdint.h>
```

### Funciones

- void [ADD](#) (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function ADD.*

- void **AND** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function AND.*

- void **EOR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function EOR.*

- void **ORR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function ORR.*

- void **SUB** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function SUB.*

- void **MUL** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function MUL.*

- void **BIC** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function BIC.*

- void **LSL** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function LSL.*

- void **LSR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function LSR.*

- void **SBC** (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)

*Function SBC.*

- void **ADC** (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)

*Function ADC.*

- void **MOV** (uint32\_t \*rd, uint32\_t a)

*Function MOV.*

- void **MVN** (uint32\_t \*rd, uint32\_t a)

*Function MVN.*

- void **RSB** (uint32\_t \*rd, uint32\_t a)

*Function RSB.*

- void **ROR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function ROR.*

- void **NOP** ()

*Function NOP.*

- void **CMN** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)

*Function CMN.*

- void **CMP** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)

*Function CMP.*

- void **TST** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)

*Function TST.*

- void **ASR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)

*Function ASR.*

- void **REV** (uint32\_t \*rd, uint32\_t a)

*Function REV.*

- void **REV16** (uint32\_t \*rd, uint32\_t a)

*Function REV16.*

- void **REVSH** (uint32\_t \*rd, uint32\_t a)

*Function REVSH.*

### 5.9.1. Documentación de las funciones

#### 5.9.1.1. void ADC ( uint32\_t \* rd, uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

Function ADC.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

**Devuelve**

0

5.9.1.2. void ADD ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function ADD.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

5.9.1.3. void AND ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function AND.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

5.9.1.4. void ASR ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function ASR.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar

**Devuelve**

0

5.9.1.5. void BIC ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function BIC.



## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.9.1.6. void CMN ( uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

## Function CMN.

## Parámetros

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

## Devuelve

0

5.9.1.7. void CMP ( uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

## Function CMP.

## Parámetros

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

## Devuelve

0

5.9.1.8. void EOR ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

## Function EOR.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.9.1.9. void LSL ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

## Function LSL.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

**5.9.1.10. void LSR ( uint32\_t \* rd, uint32\_t a, uint32\_t b )**

Function LSR.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

**5.9.1.11. void MOV ( uint32\_t \* rd, uint32\_t a )**

Function MOV.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

**Devuelve**

0

**5.9.1.12. void MUL ( uint32\_t \* rd, uint32\_t a, uint32\_t b )**

Function MUL.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

**5.9.1.13. void MVN ( uint32\_t \* rd, uint32\_t a )**

Function MVN.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

## Devuelve

0

## 5.9.1.14. void NOP ( )

Function NOP.

## Devuelve

0

## 5.9.1.15. void ORR ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function ORR.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

## 5.9.1.16. void REV ( uint32\_t \* rd, uint32\_t a )

Function REV.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

## Devuelve

0

## 5.9.1.17. void REV16 ( uint32\_t \* rd, uint32\_t a )

Function REV16.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
------------------	---

<i>uint32_t</i>	a primera registro que se va utilizaras
-----------------	---

Devuelve

0

5.9.1.18. void REVSH ( uint32\_t \* *rd*, uint32\_t *a* )

Function REVSH.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

Devuelve

0

5.9.1.19. void ROR ( uint32\_t \* *rd*, uint32\_t *a*, uint32\_t *b* )

Function ROR.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

Devuelve

0

5.9.1.20. void RSB ( uint32\_t \* *rd*, uint32\_t *a* )

Function RSB.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

Devuelve

0

5.9.1.21. void SBC ( uint32\_t \* *rd*, uint32\_t *a*, uint32\_t *b*, uint32\_t \* *Banderas* )

Function SBC.

Parámetros

---

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

5.9.1.22. void SUB ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function SUB.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

Devuelve

0

5.9.1.23. void TST ( uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

Function TST.

Parámetros

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

## 5.10. Referencia del Archivo instrucciones.h

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <urses.h>
```

### Funciones

- void **ADD** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function ADD.*
- void **AND** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function AND.*
- void **EOR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function EOR.*

- void **ORR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function ORR.*
- void **SUB** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function SUB.*
- void **MUL** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function MUL.*
- void **BIC** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function BIC.*
- void **LSL** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function LSL.*
- void **LSR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function LSR.*
- void **SBC** (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Function SBC.*
- void **ADC** (uint32\_t \*rd, uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Function ADC.*
- void **MOV** (uint32\_t \*rd, uint32\_t a)  
*Function MOV.*
- void **MVN** (uint32\_t \*rd, uint32\_t a)  
*Function MVN.*
- void **RSB** (uint32\_t \*rd, uint32\_t a)  
*Function RSB.*
- void **ROR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function ROR.*
- void **NOP** ()  
*Function NOP.*
- void **CMN** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Function CMN.*
- void **CMP** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Function CMP.*
- void **TST** (uint32\_t a, uint32\_t b, uint32\_t \*Banderas)  
*Function TST.*
- void **ASR** (uint32\_t \*rd, uint32\_t a, uint32\_t b)  
*Function ASR.*
- void **REV** (uint32\_t \*rd, uint32\_t a)  
*Function REV.*
- void **REV16** (uint32\_t \*rd, uint32\_t a)  
*Function REV16.*
- void **REVSH** (uint32\_t \*rd, uint32\_t a)  
*Function REVSH.*

### 5.10.1. Documentación de las funciones

#### 5.10.1.1. void ADC ( uint32\_t \* rd, uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

Function ADC.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

## Devuelve

0

5.10.1.2. void ADD ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b* )

Function ADD.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.10.1.3. void AND ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b* )

Function AND.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.10.1.4. void ASR ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b* )

Function ASR.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar

## Devuelve

0

5.10.1.5. void BIC ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b* )

Function BIC.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

5.10.1.6. void CMN ( uint32\_t *a*, uint32\_t *b*, uint32\_t \* *Banderas* )

Function CMN.

**Parámetros**

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

**Devuelve**

0

5.10.1.7. void CMP ( uint32\_t *a*, uint32\_t *b*, uint32\_t \* *Banderas* )

Function CMP.

**Parámetros**

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

**Devuelve**

0

5.10.1.8. void EOR ( uint32\_t \* *rd*, uint32\_t *a*, uint32\_t *b* )

Function EOR.

**Parámetros**

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

**Devuelve**

0

5.10.1.9. void LSL ( uint32\_t \* *rd*, uint32\_t *a*, uint32\_t *b* )

Function LSL.



## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.10.1.10. void LSR ( *uint32\_t \* rd*, *uint32\_t a*, *uint32\_t b* )

Function LSR.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.10.1.11. void MOV ( *uint32\_t \* rd*, *uint32\_t a* )

Function MOV.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

## Devuelve

0

5.10.1.12. void MUL ( *uint32\_t \* rd*, *uint32\_t a*, *uint32\_t b* )

Function MUL.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

5.10.1.13. void MVN ( *uint32\_t \* rd*, *uint32\_t a* )

Function MVN.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

## Devuelve

0

## 5.10.1.14. void NOP ( )

Function NOP.

## Devuelve

0

## 5.10.1.15. void ORR ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function ORR.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

## Devuelve

0

## 5.10.1.16. void REV ( uint32\_t \* rd, uint32\_t a )

Function REV.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

## Devuelve

0

## 5.10.1.17. void REV16 ( uint32\_t \* rd, uint32\_t a )

Function REV16.

## Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
------------------	---

<i>uint32_t</i>	a primera registro que se va utilizaras
-----------------	---

Devuelve

0

5.10.1.18. void REVSH ( *uint32\_t* \* *rd*, *uint32\_t* *a* )

Function REVSH.

Parámetros

<i>uint32_t</i> *	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

Devuelve

0

5.10.1.19. void ROR ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b* )

Function ROR.

Parámetros

<i>uint32_t</i> *	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

Devuelve

0

5.10.1.20. void RSB ( *uint32\_t* \* *rd*, *uint32\_t* *a* )

Function RSB.

Parámetros

<i>uint32_t</i> *	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar

Devuelve

0

5.10.1.21. void SBC ( *uint32\_t* \* *rd*, *uint32\_t* *a*, *uint32\_t* *b*, *uint32\_t* \* *Banderas* )

Function SBC.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

5.10.1.22. void SUB ( uint32\_t \* rd, uint32\_t a, uint32\_t b )

Function SUB.

Parámetros

<i>uint32_t*</i>	rd puntero a la primera direccion del arreglo
<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundo registro que se va utilizar

Devuelve

0

5.10.1.23. void TST ( uint32\_t a, uint32\_t b, uint32\_t \* Banderas )

Function TST.

Parámetros

<i>uint32_t</i>	a primera registro que se va utilizar
<i>uint32_t</i>	b segundoregistro que se va utilizar
<i>uint32_t*</i>	Banderas puntero al arreglo donde se guardan las banderas

Devuelve

0

## 5.11. Referencia del Archivo main.c

```
#include <curses.h>
#include <stdio.h>
#include "funcBanderas.h"
#include "funcionregistro.h"
#include "funcioninterfaz.h"
#include "instrucciones.h"
#include <stdint.h>
#include "decoder.h"
```

## Funciones

- int [main](#) (void)

### 5.11.1. Documentación de las funciones

5.11.1.1. `int main ( void )`

## 5.12. Referencia del Archivo README.md

