

# **Memoria Técnica**

## **Proyecto Hardware**

Jaime Yoldi Viguera 779057      José Marín Díez 778148

Enero 2021

# Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Librerías</b>	<b>3</b>
2.1	Botones . . . . .	3
2.1.1	EINT0 . . . . .	3
2.1.2	EINT1 . . . . .	3
2.2	Temporizadores . . . . .	3
2.2.1	TIMER0 . . . . .	3
2.2.2	TIMER1 . . . . .	3
2.3	GPIO . . . . .	3
2.4	Cola de eventos . . . . .	3
2.5	Gestión de los eventos . . . . .	3
2.5.1	Máquina de estados . . . . .	3
2.6	Comandos . . . . .	3
2.7	Power Management . . . . .	3
2.8	Modificaciones Reversi8 . . . . .	3
2.9	Modificaciones Startup . . . . .	4
2.10	RTC . . . . .	4
2.11	SWI . . . . .	4
2.12	UART0 . . . . .	4
2.13	RTC . . . . .	4
2.14	WD . . . . .	4

# 1 Introducción

En este documento se presenta la memoria técnica correspondiente a las prácticas 2 y 3 de la asignatura Proyecto Hardware. A lo largo del documento se detalla el proceso de implementación del conocido juego Reversi. Partiendo del proyecto en lenguaje c previamente optimizado en la práctica 1 se realizan una serie de modificaciones y adaptaciones, añadiendo nuevas funcionalidades y permitiendo que este pueda ser ejecutado en un emulador de máquina real con sus periféricos, como pueden ser botones, leds, teclado, pantalla, etc.

Algunas partes del código serán desarrolladas en lenguaje **ARM** y otras en **c** para finalmente ser todo ello ejecutado con la ayuda de un emulador del procesador *ARM LPC2105*. El entorno de desarrollo empleado es *uVision IDE*.

## 2 Librerías

### 2.1 Botones

#### 2.1.1 EINT0

#### 2.1.2 EINT1

### 2.2 Temporizadores

#### 2.2.1 TIMER0

#### 2.2.2 TIMER1

### 2.3 GPIO

### 2.4 Cola de eventos

### 2.5 Gestión de los eventos

#### 2.5.1 Máquina de estados

### 2.6 Comandos

### 2.7 Power Management

### 2.8 Modificaciones Reversi8

\* Utilizacion del ARM\_ARM

## 2.9 Modificaciones Startup

### 2.10 RTC

### 2.11 SWI

\* TODO: Aumentar el valor de la tabla que nos dijo para que no desborde

### 2.12 UART0

### 2.13 RTC

Para medir el tiempo transcurrido durante la partida se hace uso de uno de los dos contadores con funcionalidades específicas del procesador *ARM LPC2105*, el *Real Time Clock* (RTC). Es necesario destacar que este contador, a diferencia de los otros no genera ninguna interrupción, ya que como se ha dicho su único propósito es proporcionar información sobre el tiempo transcurrido y consumiendo poca energía.

Este es iniciado al comienzo del juego y está en funcionamiento en todo momento incluso cuando el procesador está suspendido, en powerdown o idle. Antes de esto es necesario configurarlo para adaptarlo a la frecuencia a la que trabaja el procesador (60MHz), para ello se modifica el *Prescaler Integer register* y el *Prescaler Fraction register* de la siguiente manera.

```
PREINT = 0x726;  
PREFRAC = 0x700;
```

Para que el RTC comience a contar es necesario habilitarlo, para ello se activa el bit 0 del *Clock Control Register*, además de poner a 0 la cuenta de minutos y segundos.

```
CCR=0x01;
```

Para obtener el tiempo transcurrido se hacen funciones `RTC_leer_segundos` y `RTC_leer_minutos` que se encarga de leer del registro `CTIME0` los bits correspondientes y devolver el tiempo para cada caso.

### 2.14 WD

Para ... se va a utilizar el otro contador específico, el watchdog (WD) ...