

L I T E

Autor: Julio Fernández Lasheras

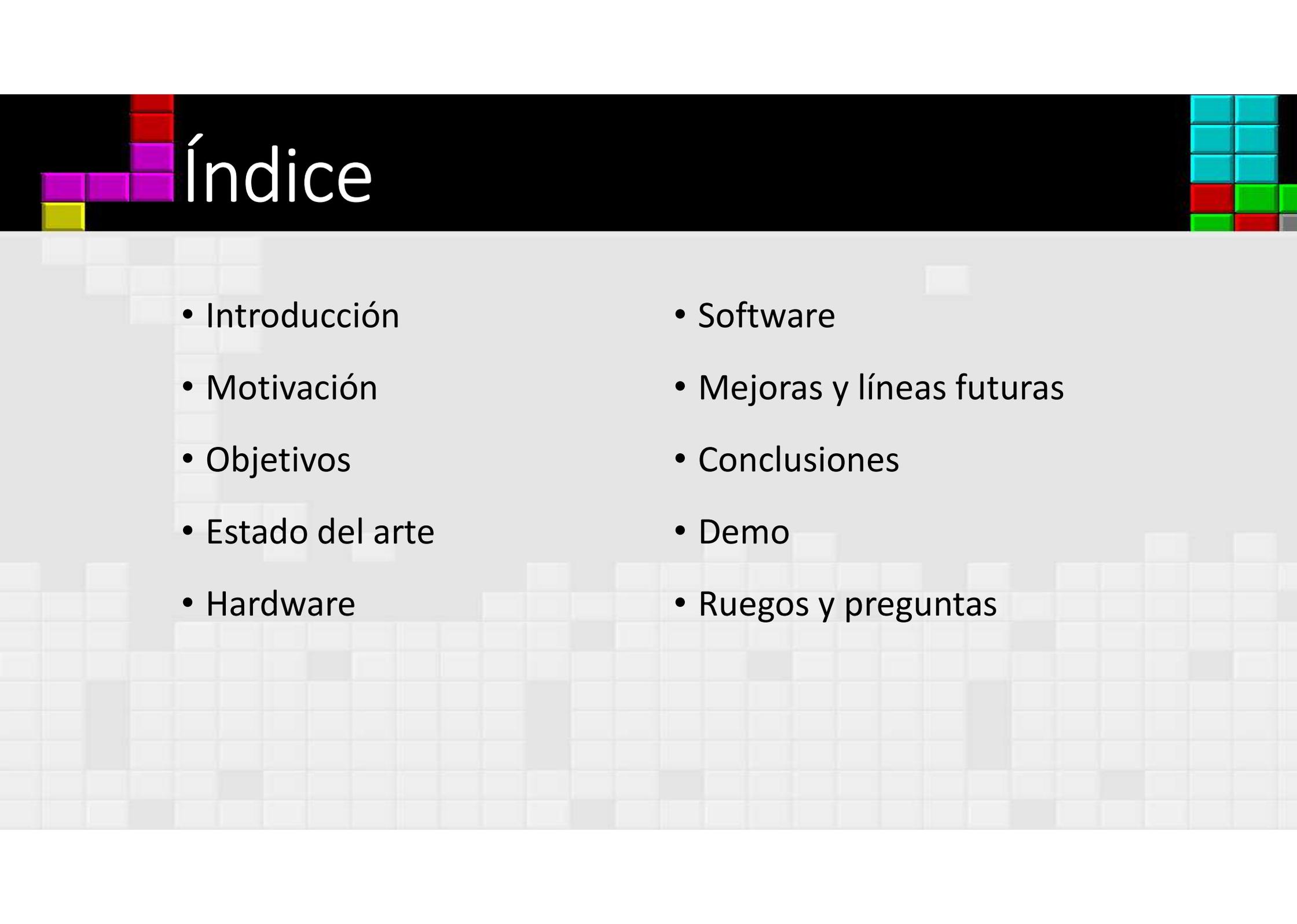
Consultor: Jordi Bécares Ferré

Master Universitario en Ingeniería Informática

Sistemas Encastados

Enero 2018





Índice

- Introducción
- Motivación
- Objetivos
- Estado del arte
- Hardware
- Software
- Mejoras y líneas futuras
- Conclusiones
- Demo
- Ruegos y preguntas

Introducción (I)

- ¿Qué son los sistemas embebidos?

embeber [Conjugar](#)

Del lat. *imbibere*.

1. *tr.* Dicho de un cuerpo sólido: Absorber a otro líquido.
2. *tr.* Empapar, llenar de un líquido algo poroso o esponjoso.
3. *tr.* Dicho de una cosa: Contener o encerrar dentro de sí a otra.
4. *tr.* Dicho de una cosa inmaterial: Incorporar o incluir dentro de sí a otra.
5. *tr.* Encajar, embutir o meter algo dentro de otra cosa.
6. *tr.* Recoger parte de una cosa en ella misma, reduciéndola o acortándola.
7. *intr.* Encogerse, apretarse, tupirse.
8. *prnl.* **embebecerse**.
9. *prnl.* Instruirse con rigor y profundidad en una doctrina, teoría, *etc.*
10. *prnl.* Entregarse con vivo interés a una tarea, sumergirse en ella.
11. *prnl.* *Taurom.* Dicho de un toro: Quedarse parado y con la cabeza alta cuando recibe la estocada.

Real Academia Española © Todos los derechos reservados

- ¿Dónde tenemos están los sistemas embebidos?



Descodificador IPTV



GPS



Termostato



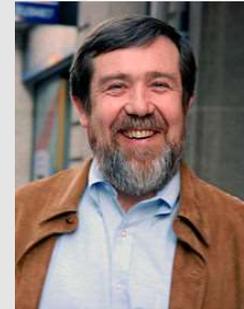
Máquina Recreativa



Launchpad
MSP432P401R

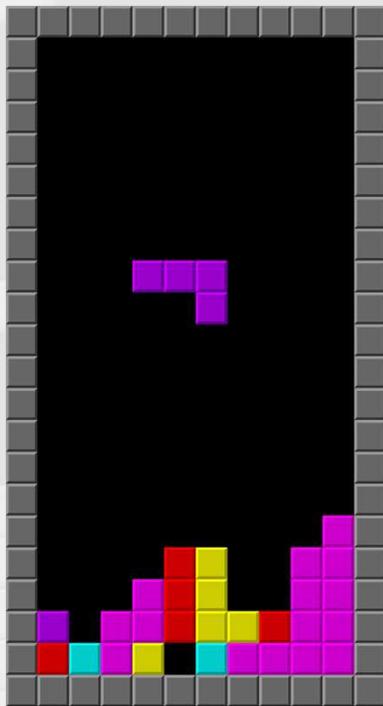
Introducción (II)

- Tetris. Orígenes
 - Creado por Alekséi Pázhitnov en 1984
- Esta basado en el uso de tetrominós

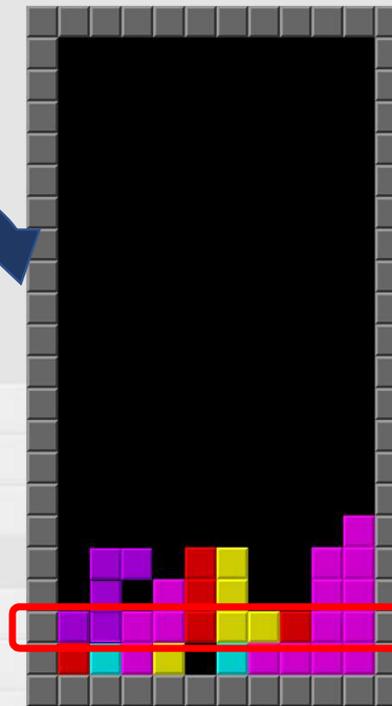


Introducción (III)

- ¿Cómo se juega?



Rotar y bajar

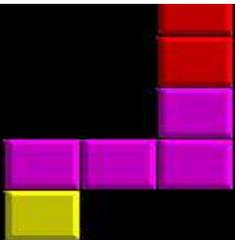


Fila completa

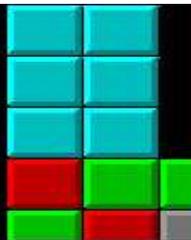
Motivación

- ¿De dónde viene la idea?
- ¿Qué se puede hacer con un sistema embebido?
- Cómo hacer un juego





Objetivos



- Objetivos principales
 - ✓ Tener una interfaz grafica basada en LCD
 - ✓ Dotar de interfaz de usuario vía Joystick y botones
 - ✓ Implementar la lógica del juego Tetris
- Objetivos secundarios o mejoras
 - ✓ Implementar una pantalla inicial
 - ✓ Implementar histórico de mayor puntuación
 - ✓ Añadir audio al sistema
 - ✓ Funcionalidad de pausa
 - ✗ Funcionalidad de guardado
 - ✓ Creación de niveles

Estado del arte (I)

- Evolución del Tetris
 - Sega System E 1988

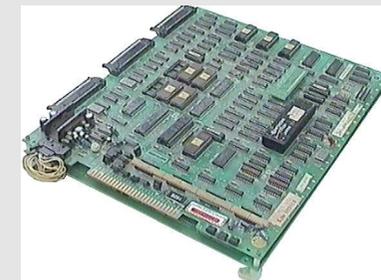


HARDWARE DESCRIPTION

CPU : Z-80B @ 8MHz
ROM : 32K for code + up to 4 x 32/64K for banks
RAM : 16K
Video RAM : 64K (2 banks of 16K for each VDP)
VDP : 2 x 315-5124 (Same as the SMS except the are 2. Note that the 315-5124 is a TMS9928 VDP combined with a SN76489 sound chip!)
Colors : 64 (32 for each VDP. Each VDP picks its palette from a possible 64 colors)
Resolution : 256 x 192
Screen Scroll : X, Y, partial
Audio : 6 tone channels and 2 noise channels
Characters : 8 x 8 (448 definitions per VDP)
Sprites : 8x8, 8x16, 16x16 (up to 128 on screen at a time)

Estado del arte (II)

- Evolución del Tetris
 - Sega System 16A 1988

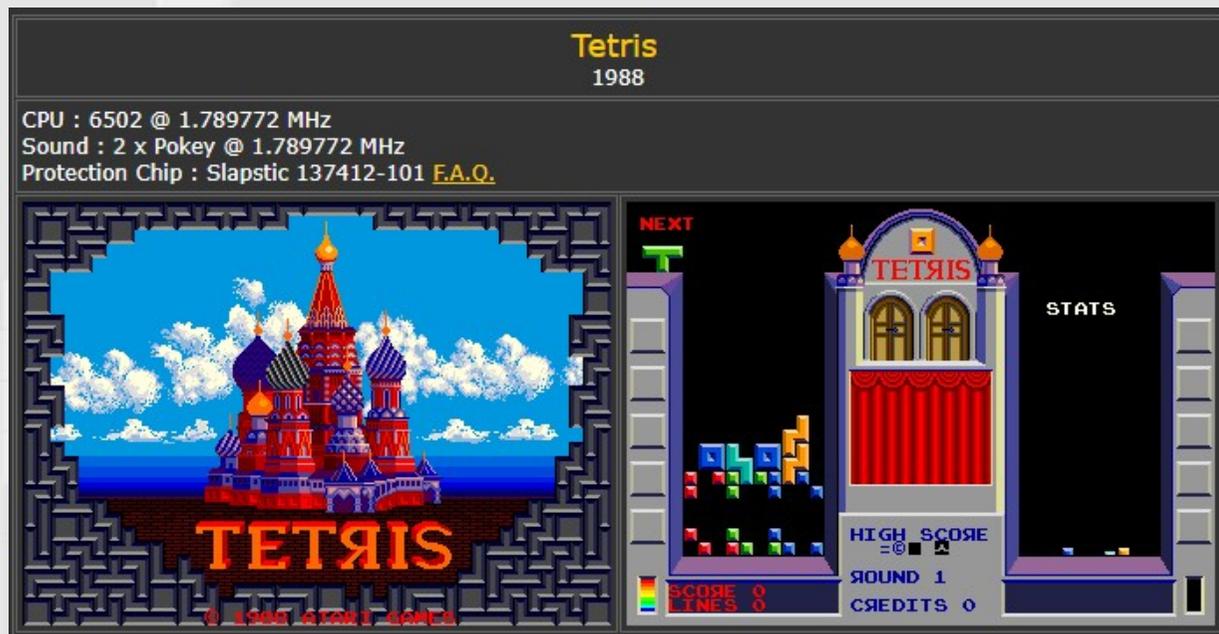


HARDWARE DESCRIPTION

Main CPU : MC68000 @ 10 MHz
Sound CPU : Z80 @ 4 MHz
Sound chip : Yamaha YM2151 @ 4 MHz & NEC uPD7751 ADPCM Decoder
Video resolution : 320 x 224 (vertical)
Colour's : 4096
Board composition : CPU board + Video board
Hardware Features : 128 Sprites on screen at one time, 2 tile layers, 1 text layer, 1 sprite layer with hardware sprite zooming, translucent shadows
Pictures : [Mother Board - Side B](#)

Estado del arte (III)

- Evolución del Tetris
 - Sega Atari 6502 1988



Estado del arte (IV)

- Evolución del Tetris
 - SONY ZN-2 HARDWARE 1998



HARDWARE DESCRIPTION

Main CPU : R3000A 32 bit RISC processor, Clock - 50MHz?, Operating performance - 30 MIPS, Instruction Cache - 4KB
BUS : 132 MB/sec.
OS ROM : 512 Kilobytes
Sound CPU : Z80 (Encrypted Kabuki Model)
Sound Chips : Capcom Q Sound (PSX Sound chip is ignored)
Main RAM: 2/4/8 Megabytes depending on game.
Video RAM: 2/4/8 Megabytes depending on game.
Sound RAM : 512 Kilobytes
Graphical Processor : 360,000 polygons/sec, Sprite/BG drawing, Adjustable frame buffer, No line restriction, 4,000 8x8 pixel sprites with individual scaling and rotation, Simultaneous backgrounds (Parallax scrolling)
Sprite Effects : Rotation, Scaling up/down, Warping, Transparency, Fading, Priority, Vertical and horizontal line scroll
Resolution : 256x224 - 740x480
Colours : 16.7 million colors, Unlimited CLUTs (Color Look-Up Tables)
Other Features : custom geometry engine, custom polygon engine, MJPEG decoder
Control Chip : NEC uPD78081 MCU @ 5MHz for handling analog controls and trackballs.

Estado del arte (V)

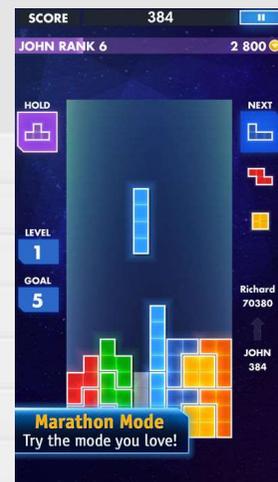
- Evolución del Tetris
 - Tetris EA 2008
 - Resolución 1080p y audio 5.1
 - 6 modos de juego en multijugador
 - Ver estadísticas, repeticiones, de jugadas, puntuaciones de los mejores jugadores...



Samsung Galaxy S7 edge specs overview

8-core, 2.3 GHz	3600 mAh
CPU Exynos 8 Octa 8890 / Snapdragon 820	LTE HSPA, HSUPA, UMTS, EDGE, GPRS NFC, Wi-Fi 802.11, Bluetooth 4.2
GPU ARM Mali-T880MP14 / Adreno 530	
RAM 4 GB RAM	12 megapixels REAR F1.7 aperture, sensor 1/2.5" Front: 5 Megapixels Video capture: 3840x2160 (4K)
STORAGE 32 / 64 GB	
5.5 inches 1440 x 2560 pixels 534 ppi, S Amoled	Android (6.0) TouchWiz UI
5.94" (150.9 mm)	0.30" (7.7 mm) thickness
2.86" (72.6 mm)	5.54 oz (157 g)

phoneArena.com

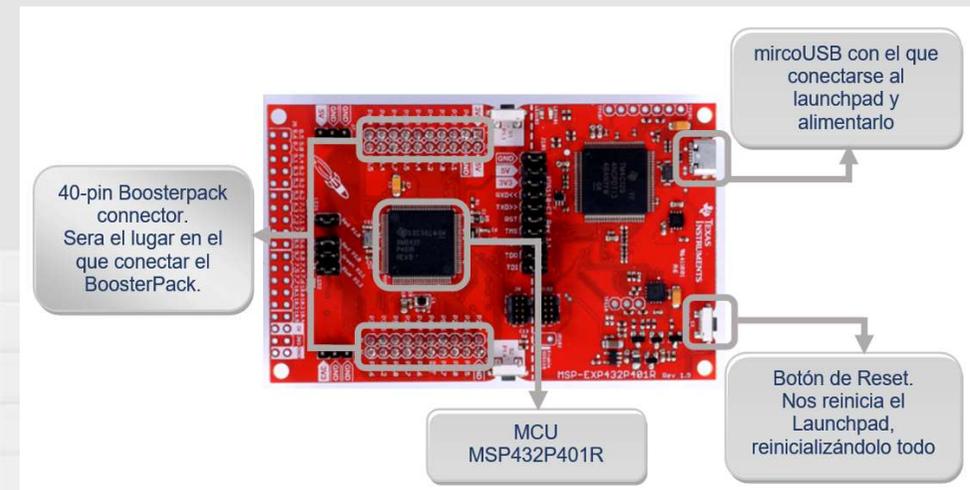


Hardware (I)

- Launchpad MSP432P401R

Features

- Low-power, high performance MSP432P401R MCU
 - 48MHz 32-bit ARM Cortex M4F with Floating Point Unit and DSP acceleration
 - Power consumption: 80uA/MHz active and 660nA RTC standby operation
 - Digital: Advanced Encryption Standard (AES256) Accelerator, CRC, DMA, HW MPY32
 - Memory: 256KB Flash, 64KB RAM
 - Timers: 4 x16-bit, and 2 x 32-bit
 - Communication: Up to 4 I2C, 8 SPI, 4 UART
- 40 pin BoosterPack Connector, and support for 20 pin BoosterPacks
- Onboard XDS-110ET emulator featuring EnergyTrace+ Technology
- 2 buttons and 2 LEDs for User Interaction
- Back-channel UART via USB to PC

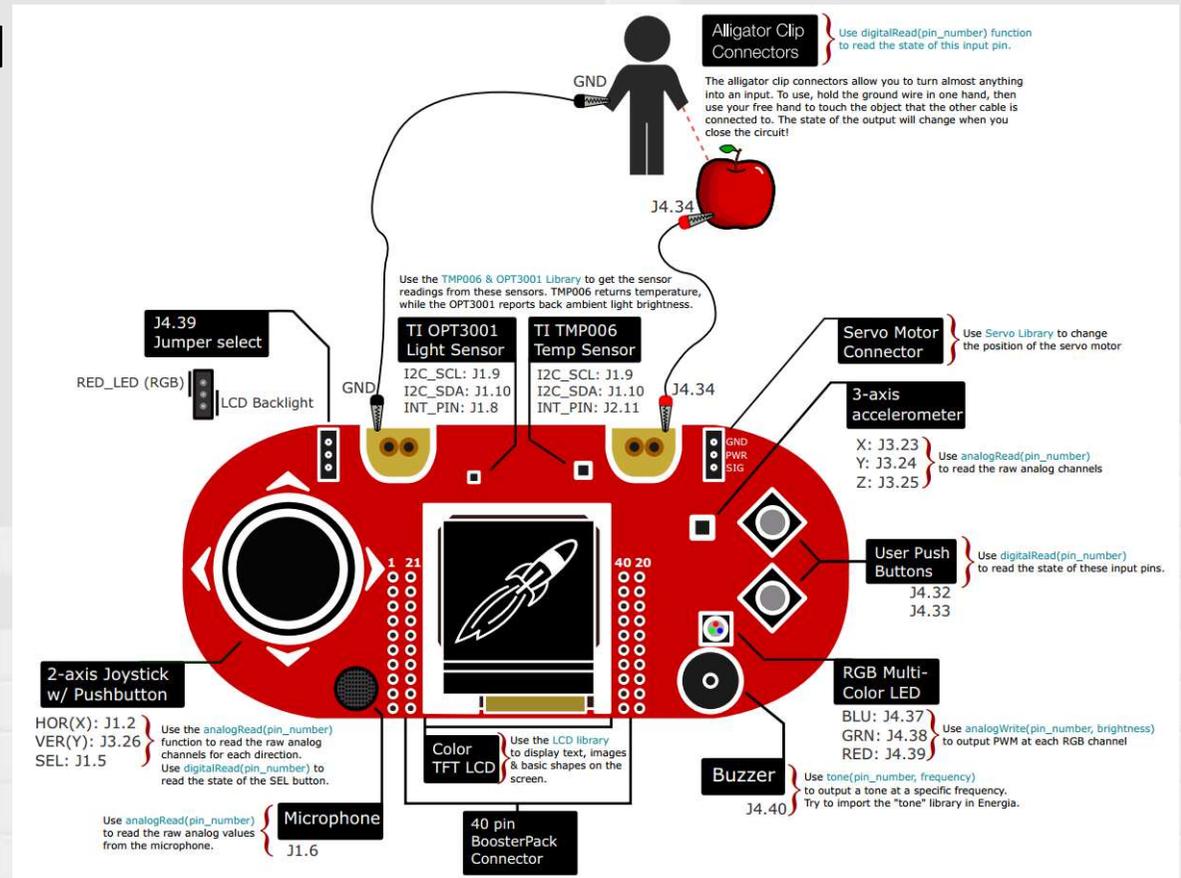


Hardware (II)

• Educational BoosterPack MKII

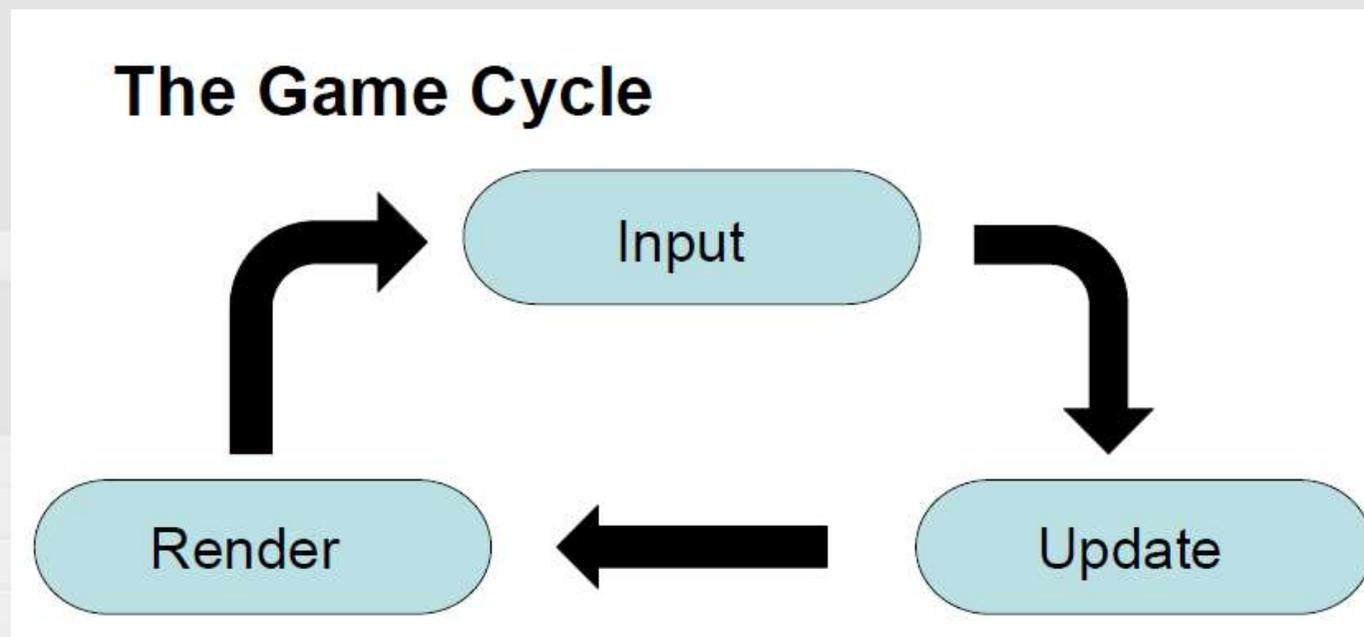
Features

- TI OPT3001 Light Sensor
- TI TMP006 Temperature Sensor
- Servo Motor Connector
- 3-Axis Accelerometer
- User Push Buttons
- RGB Multi-color LED
- Buzzer
- 40-pin Stackable BoosterPack Connector
- Color TFT LCD Display
- Microphone
- 2-Axis Joystick with Pushbutton



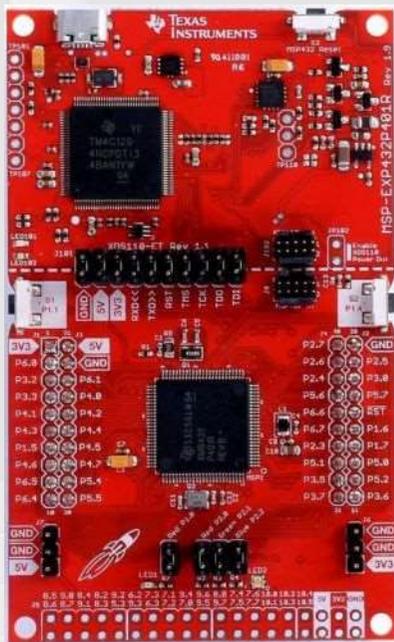
Software (I)

- Ciclo de vida de un juego



Software (II)

- Esquema funcional



El BoosterPack, envía los datos del joystick para que los procese la lógica del Tetris

Desde el Launchpad según la lógica del Tetris, se le mandan las figuras o textos que tienen que aparecer en el display

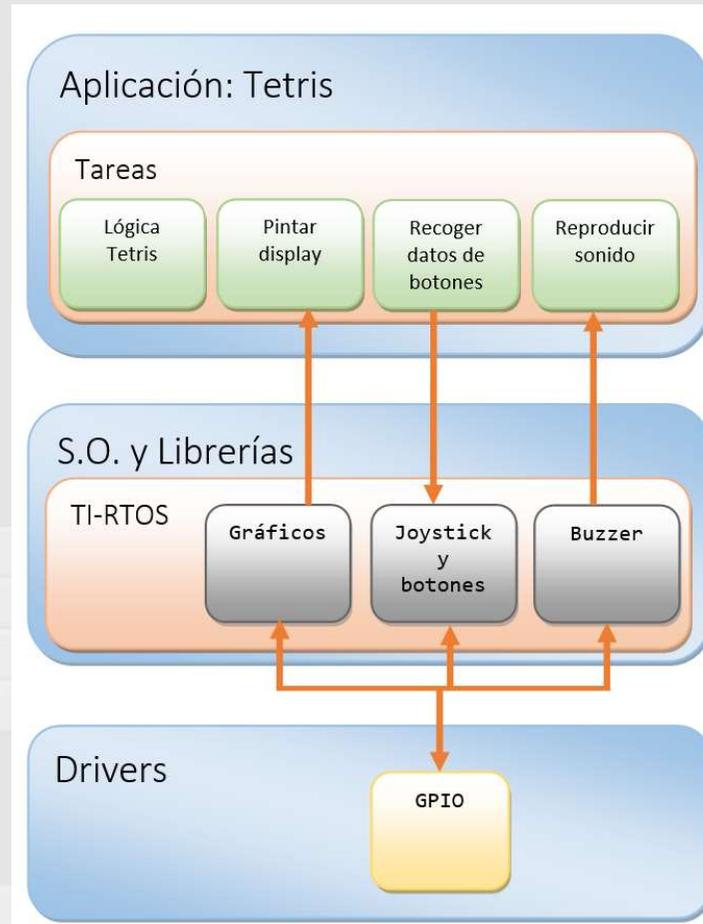
Al igual que con el LCD, según la lógica del juego, se le mandaran diferentes tonos al Buzzer para que los reproduzca.

Desde el Launchpad, se recogen los botones que haya presionado el usuario, la lógica del Tetris lo procesara, y puede resultar en la rotación de una pieza o una pausa.



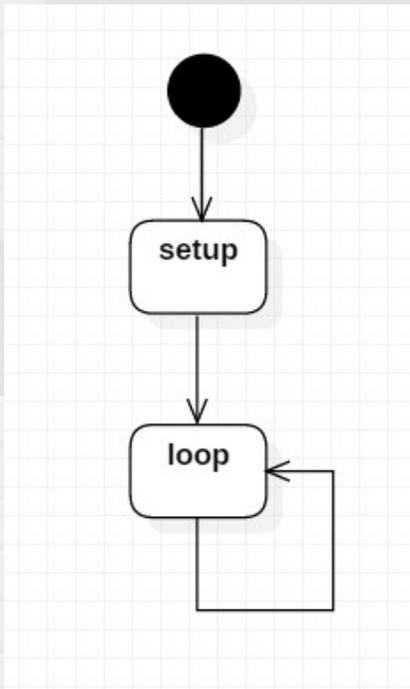
Software (III)

- Esquema funcional



Software (IV)

- Energia



```
tetrisLite | Energia 1.6.10E18
Archivo Editar Programa Herramientas Ayuda
tetrisLite joystick.cpp joystick.h music.cpp music.h pitches.h
/*
 *
 *TFM-TetrisLite
 *Master en Ingenieria Informatica
 *Julio Fernandez Lasheras
 *02/01/2017
 */

// Include application, user and local libraries
#include "SPI.h"
#include "Energia.h"

#include "joystick.h"
#include "music.h"

#include "Screen_HX8353E.h"
Screen_HX8353E myScreen;

//////////////////////////////////////////Defines//////////////////////////////////////////
#define LCD_WIDTH 120
#define LCD_HEIGHT 120

#define MIN(X, Y) ((X) < (Y) ? (X) : (Y))

#define BOARD_WIDTH 9
#define BOARD_HEIGHT 16

#define BLOCK_SIZE MIN( (LCD_WIDTH-1) / BOARD_WIDTH, (LCD_HEIGHT-1) / BOARD_HEIGHT )

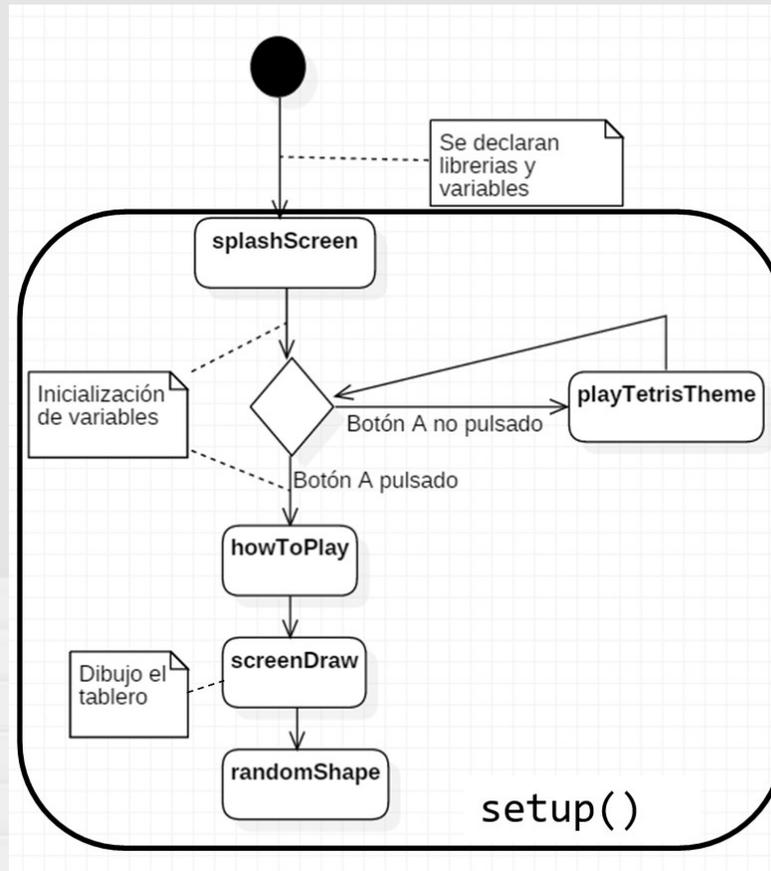
#define BOARD_LEFT (LCD_WIDTH - BOARD_WIDTH * BLOCK_SIZE)/4*3
#define BOARD_RIGHT (BOARD_LEFT + BLOCK_SIZE * BOARD_WIDTH)
#define BOARD_TOP (LCD_HEIGHT - BOARD_HEIGHT * BLOCK_SIZE) / 2
#define BOARD_BOTTOM (BOARD_TOP + BOARD_HEIGHT * BLOCK_SIZE)
#define newBlockx 5
#define newBlocky 70

#define BUTTON_A 33
#define BUTTON_B 32

#define BLUE_LED 37
```

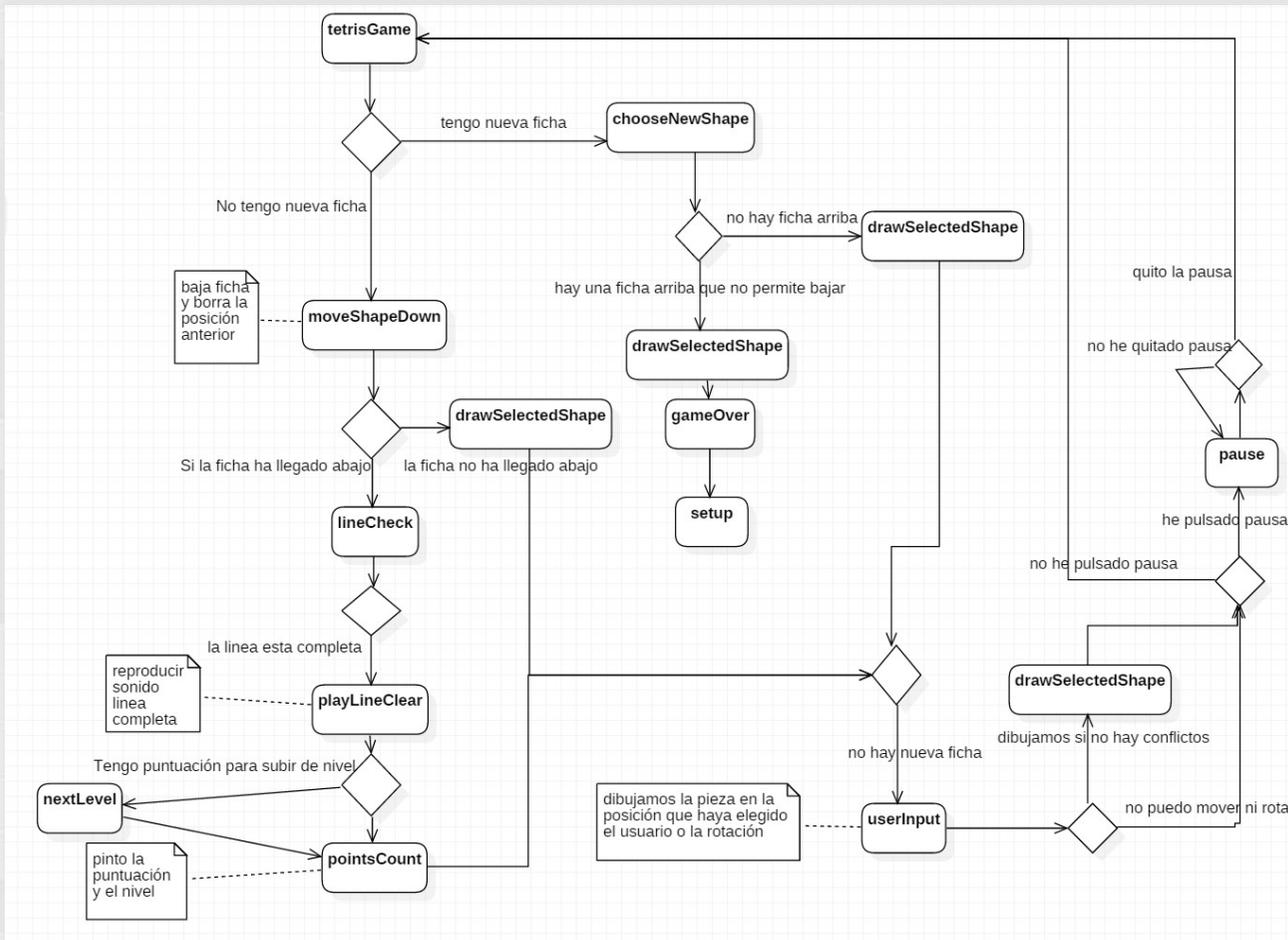
Software (V)

- Setup



Software (VI)

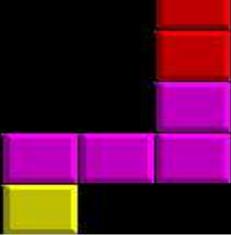
- Loop



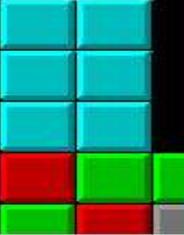
Mejoras y líneas futuras

- Guardar partida
- Selección de nivel
- Estadísticas
- Almacenar puntuaciones en cloud
- Tiempo de juego
- Mejorar interfaz

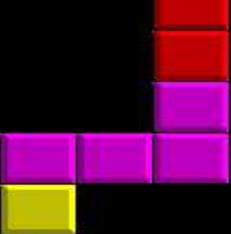




Conclusiones



- Se han cumplido los objetivos principales marcados
- Se han marcado líneas futuras con posible realización
- El kit formado por el Launchpad y BoosterPack es una gran herramienta de prototipado o proyectos domésticos
- Se ha descubierto una nueva plataforma con la que explorar sus posibilidades
- A nivel personal, gran satisfacción por haber superado este reto y al final haber disfrutado desarrollándolo.



Demo



Ruegos y preguntas

Muchas gracias por su atención

¿Ruegos y preguntas?

