



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

VIDEOJOCS RETRO



Memòria i Annexos

Autor: Pau Cuevas Carrasco
Director: Pérez, Poch, Antoni
Convocatòria: Octubre 2017

Resum

Aquest treball consisteix en el disseny i prototipat d'un model de consola de videojocs "retro" de 8 bits i la programació d'un joc per a la plataforma creada. Per a la comprensió total del treball fet, explicarem diversos conceptes que han estat clau en el desenvolupament de la consola, com ara l'aprofundiment en les capacitats dels microcontroladors de 8 bits (en concrets els ATMEGA), la comprensió sobre diverses senyals de vídeo i com es conformen, el disseny de perifèrics (en aquest cas integrats) necessaris per a una consola, programació de videojocs (en aquest cas mitjançant el llenguatge C i imatges convertides en matrius de valors hexadecimals), tot veient com es va arribar a aquest punt en el desenvolupament de consoles i quins han estat els camins que s'han pres posteriorment. Durant aquest procés s'explicaran les eleccions de hardware fetes i els càlculs requerits, així com el codi desenvolupat.

Resumen

Este proyecto consiste en el diseño y prototipado de un modelo de consola “retro” de 8 bits y la programación de un juego para la plataforma creada. Para la comprensión total del trabajo realizado, explicaremos diversos conceptos que han sido clave en el desarrollo de la consola, como la profundización en las capacidades de los microcontroladores de 8 bits (en concreto los ATMEGA), la comprensión de diversos tipos de señales de vídeo y qué las conforma, el diseño de periféricos (en el caso que nos ocupa éstos son integrados) necesarios en una consola, programación de videojuegos (mediante C e imágenes convertidas en matrices de hexadecimales), viendo cómo se llegó a este nivel en el desarrollo de las consolas y los caminos tomados posteriormente. Durante el proceso, se explicarán las elecciones de hardware realizadas y los cálculos requeridos, así como el código desarrollado.

Abstract

This project consists of the design and prototyping of an 8 bit “retro” videogame console and the programming of a game for this console. In order to understand the work done we will explain different concepts that have been key for the development, like the study of the capabilities of 8 bit microcontrollers (ATMEGA on this case), the understanding of various types of video signals, the design of peripherals needed, videogame programming (using C Language and hexadecimal arrays as images), the history of retro videogames; and finally explaining the hardware choices made and the developed code.

Agraïments

Agraeixo l'ajuda per fer aquest treball a la comunitat de professors de la UPC ja que he rebut consells i assistència de molts d'ells, així com a Victor Mañas que m'ha aconsellat en diversos camps necessaris a l'elaboració d'aquest projecte; finalment agraeixo també a tothom que ha tingut la paciència necessària amb mi en els moments en que només parlava i pensava coses per tirar endavant aquest projecte.



Índex

RESUM	I
RESUMEN	II
ABSTRACT	III
AGRAÏMENTS	IV
1. PREFACI	1
1.1. Origen del treball	1
1.2. Motivació.....	1
1.3. Requeriments previs.....	1
2. INTRODUCCIÓ	3
3. BREU HISTORIA DELS VIDEOJOCs.	5
4. DISSENY DEL HARDWARE	9
4.1. El microcontrolador	9
4.2. L'oscil·lador.....	11
4.3. L'alimentació	11
4.4. La pantalla	12
4.5. El convertidor de video.....	13
4.6. El Joystic.....	15
4.7. Esquemàtic final.....	17
4.8. PCB dissenyada	18
4.9. La carcassa.....	19
4.10. Problemes i errors sorgits durant el muntatge.....	21
4.10.1. AD725.....	21
4.10.2. L'oscil·lador de quars	23
4.11. Prototip final	24
5. LA SENYAL DE VIDEO	25
6. EL SOFTWARE	26
6.1. El nucli.....	26
6.2. Dibuix de la pantalla	27

6.3.	Senyal de sincronisme vertical.....	28
6.4.	Controls	28
6.5.	Funcions API.....	28
6.6.	Apartat gràfic	28
6.7.	Codi del joc.....	32
7.	ANÀLISI DE L'IMPACTE AMBIENTAL	36
8.	CONCLUSIONS	38
8.1.	Futures millores	38
8.2.	Valoració personal	39
9.	PRESSUPOST	41
9.1.	Cost material	41
9.2.	Cost associat a les hores.....	43
9.3.	Cost total	44
	BIBLIOGRAFIA	45
	ANNEX A	47

1. Prefaci

1.1. Origen del treball

Aquest treball va ser presentat pel tutor com a opció de tria a l'hora d'escollir un TFG, però es una opció que sense dubte podria haver plantejat de fer ja que, encara que no parlem necessàriament d'una consola, la creació d'una interfície gràfica mitjançant microcontroladors porta ja un temps generant-me interès per las possibilitats com a complement a altres equipaments, visualització de control, etc.

1.2. Motivació

Com he comentat abans, el tema de crear una interfície gràfica ja m'atreia, però el fet de sumar-li el disseny d'un videojoc retro feia aquest treball perfecte, des de petit els primers jocs que vaig jugar ja eren d'una estètica píxel, sent l'època de consoles com 'MegaDrive' i 'GameBoy' encara una generació abans de l'arribada dels polígons; i tot i que actualment el nivell gràfic de les consoles ha crescut d'una forma increïble, es segueixen fent jocs, sobretot a nivell 'indie' amb estètica píxel-art; aquets jocs tot sovint trien explicar una historia diferent i arriscar-se amb conceptes que jocs de plataformes més populars no fan, sent així un estil de joc que sempre m'ha atret.

1.3. Requeriments previs

De requeriments previs només hi ha hagut un coneixement bàsic de microcontroladors, i conceptes de senyal de video VGA, fent que la part d'investigació i estudi personal fos una part important del treball.

2. Introducció

Actualment el mercat de les consoles està en ple auge, amb una competitivitat extremadament alta; tot i això, de cara a cada cop millorar més els gràfics, augmentar la mida dels escenaris i facilitar la jugabilitat, la majoria d'empreses han optat per encadenar una sèrie de consoles de les quals poques són portables. Tenint això en compte, s'ha optat per fer la consola portàtil, ja que cap de les necessitats normals del mercat s'apliquen al model a dissenyar; no hi ha necessitat d'una gran pantalla per a gaudir d'un joc de 8 bits, tampoc una necessitat exorbitant de memòria ni una potència gaire alta.

Per tant, els nostres objectius són:

- Disseny d'una consola:
 - o De 8 bits.
 - o Portable.
 - o Baixa necessitat energètica.
 - o Sense necessitat de perifèrics externs.
- Programació d'un videojoc:
 - o Adaptat a les capacitats que ofereix la consola dissenyada.

El treball comprèn tots els aspectes del disseny de la consola, des de la tria d'una font d'energia adequada, el disseny dels controls, la tria d'una pantalla (de la qual depèn el tractament de la senyal de video a fer), l'aspecte extern, etc.



3. Breu historia dels videojocs.

El primer videojoc va ser creat el 1952, anomenat 'Naught and Crosses' era una versió del tres en ratlla per a un ordinador EDSAC; posteriorment, utilitzant gràfics vectorials, apareix Spacewar, joc per a PDP-1. Aquets casos, desenvolupats en un àmbit universitari no es van fer gaire populars, durant aquest anys es creen varis jocs del mateix estil que no arriben a la població; tot canvia el 1972 amb la fundació d'Atari i la comercialització de Pong, en la que només es podia jugar a un joc que venia programat als propis circuits de la consola. Altres consoles contemporànies són la Coleco Telstar (de Coleco) i la Magnavox Odyssey (de Magnavox).



Figura 1. Llibre d'instruccions, comandament del model domèstic (1975), i exemple d'una pantalla de Pong. (TLP-Tenerife.com).

El gran èxit de Pong comporta l'aparició de noves consoles, nous jocs que poc a poc presenten millores, utilització de microprocessadors i unitats de memòria i l'aparició de jocs en color.

El 1977 Atari crea la Atari 2600, la primera consola de 8-bits, amb milers de jocs i múltiples gèneres es converteix en la consola més popular del mercat fins l'aparició de Nintendo. Durant aquesta època apareixen els principals competidors, la Odissey 2, molt similar a l'Atari 2600, i la Intellivision de Mattel. L'Atari 2600 conta d'una unitat central de processament completa, un circuit que controla els gràfics i el so (TIA), un circuit subministrador de memòria i control entrada/sortida i un buffer CMOS.



Figura 2. Atari2600 (wikipedia.org).

Posteriorment comença una escalada en la creació de noves consoles arribant a la NES, venuda per primer cop el 1983 al Japó; revolucionant el mercat amb jocs (que podien ser creats per terceres companyies) que han creat modes que encara avui segueixen vigents, com ara Mario Bros, Donkey Kong o Zelda.

La NES utilitza un microprocessador de 8 bits que produïa Ricoh basat en un nucli MOS Technology 6502, 2 KB de RAM utilitzats per al processament de les imatges, 256 bytes de RAM per a sprites (mapes de bits).



Figura 3. Nintendo Entertainment System (Google imatges).

La NES controla el mercat fins a l'aparició el 1988 de la Sega Mega Drive i junt amb altres consoles posteriors com la Neo-Geo, Super Nintendo o Mega CD incorporen xips de gràfics afegits als cartutxos i ampliacions de hardware. Posterior a aquesta època arribem a l'era dels 32 bits, amb gràfics 3D i consoles amb característiques semblants a un PC, l'evolució serà constant i segueix fins al present, nosaltres, però, ens aturem en aquesta època de 8 bits.

4. Disseny del hardware

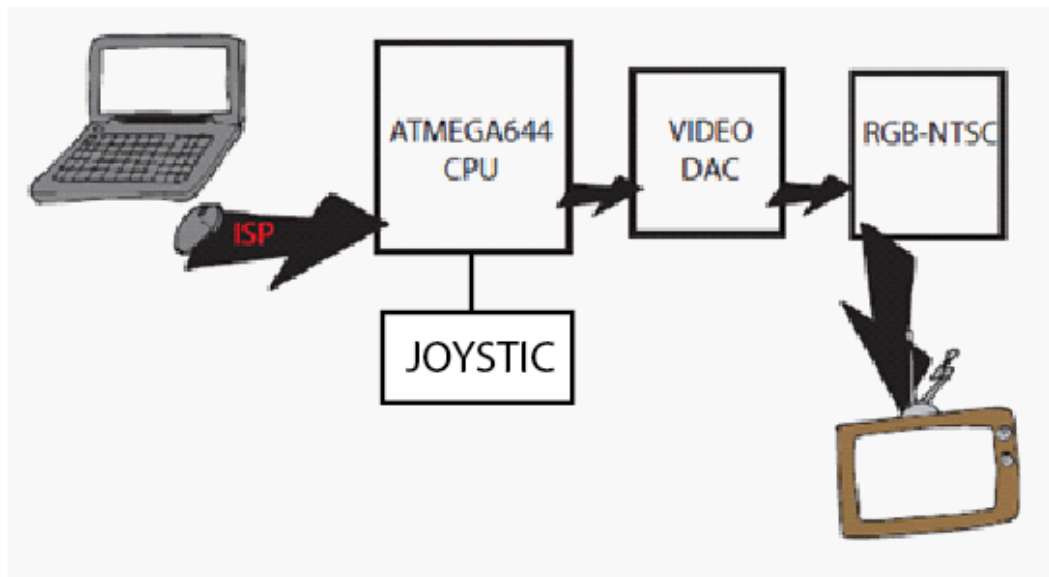


Figura 4. Esquema bàsic del hardware.

4.1. El microcontrolador

Des de el principi es te clar que el cor del muntatge serà un microcontrolador de 8 bits; al triar el microcontrolador abans de començar a dissenyar els possibles jocs, s'ha buscat un microcontrolador amb unes especificacions per sobre del que podria ser necessari en un principi, donant opció així a possibles futures millores en els jocs.

Els principals microcontroladors de 8 bits en ment a l'hora de començar són l'ATMEGA 328, ATMEGA644 i ATMEGA1284, els tres són força coneguts per estar implementats en varies plaques Arduino i tots tres tenen les especificacions mínimes sobrades per al projecte en cap, sent el més limitat l'ATMEGA328 amb 32 kB de memòria de programa, 2 bytes de RAM i 1024 bytes de EEPROM.

Finalment es tria continuar el disseny amb l'ATMEGA644 amb 64 kB de memòria de programa, 2048 bytes de EEPROM i 4 bytes de RAM, encara que també s'adquireix un ATMEGA1284 aprofitant que

els dos s'han d'importar i que el seu idèntic "patillatge" ens permet mantenir disseny tot i anar intercanviant els microcontroladors.

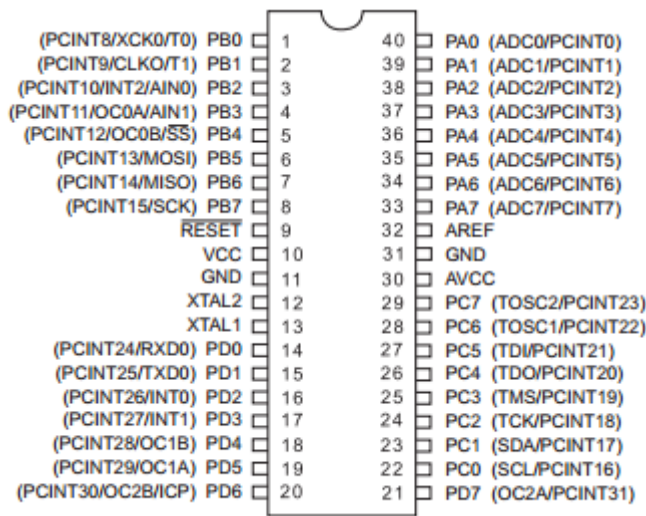


Figura 5. Pin out de l'ATMEGA644. (<http://www.atmel.com/images/2593s.pdf>)

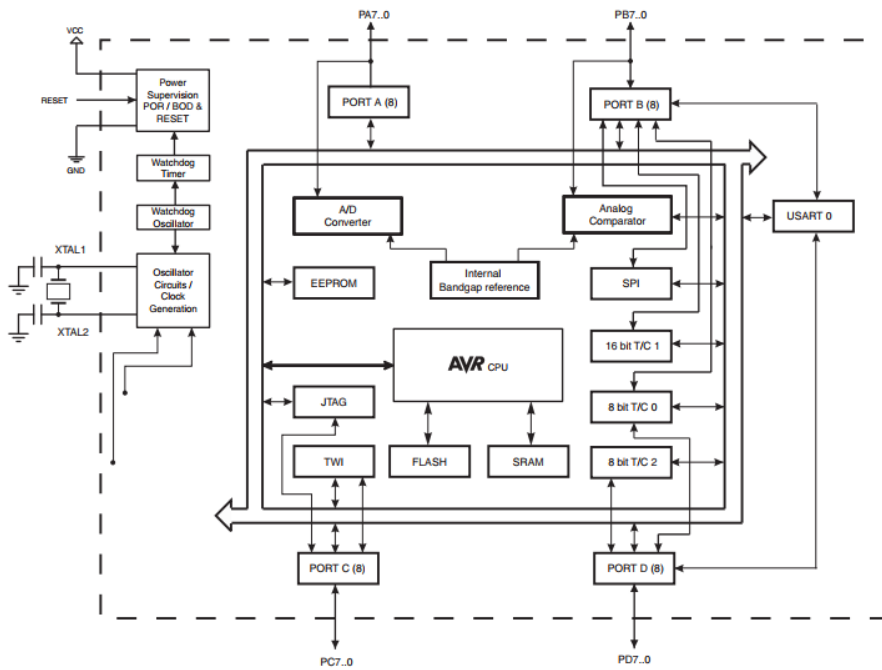


Figura 6. Diagrama de blocs de l'ATMEGA644. (<http://www.atmel.com/images/2593s.pdf>)

Tenint en compte la seva configuració es pensa en el muntatge bàsic necessari pel microcontroladors, es a dir, el port d'entrada per a la programació del microcontrolador, l'oscil·lador, l'interruptor de reset i de power, l'alimentació i un LED que ens servirà per a comprovacions.

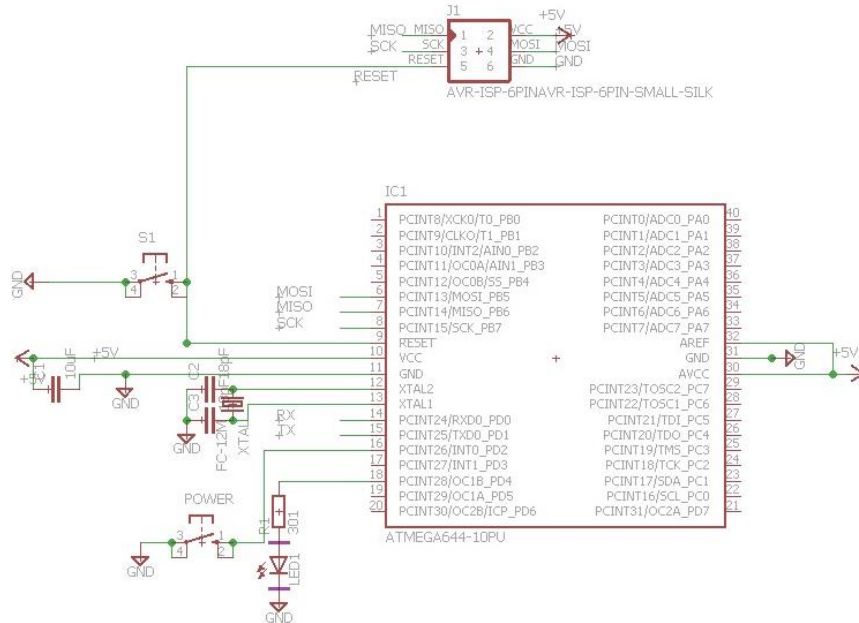


Figura 7. Muntatge bàsic.

4.2. L'oscil·lador

L'ATMEGA644 pot operar amb oscil·lador de fins a 20 MHz, però, per a generar una senyal NTSC necessitem un rellotge de 14,32 MHz, que obtindrem incloent en el muntatge un oscil·lador de 28.6363 MHz i dividint-lo.

4.3. L'alimentació

Una altra part bàsica del projecte és l'alimentació, al voler fer una consola portàtil, descartem l'ús de transformadors i adoptem la solució més típica en aquest àmbit, alimentat mitjançant piles (en aquest cas de 9 V per a que puguin alimentar la pantalla a la vegada) i afegim dos reguladors, un de 5 V i un altre de 3,3 V.

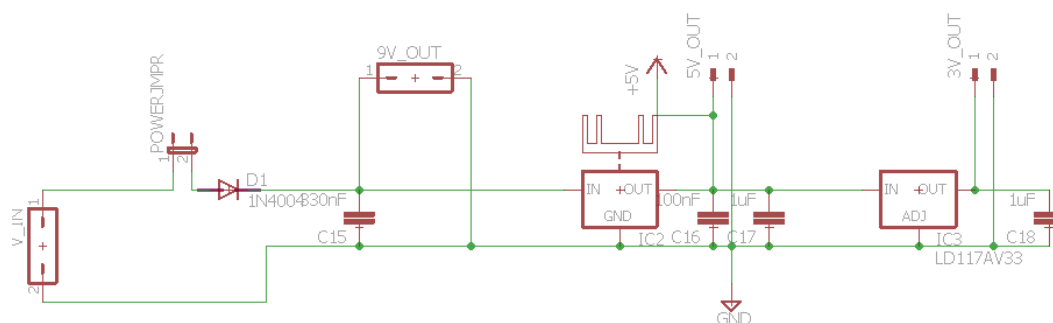


Figura 8. Alimentació del circuit.

4.4. La pantalla

Abans de prosseguir cal triar una pantalla, ja que depenent del model triat la senyal d'imatge s'haurà de tractar d'una forma o altra variant així el disseny.

Actualment hi ha forces pantalles petites de diferent característiques (mides, tàctils o no, amb capacitat per a memòria SD), ens decantem per un display TFT NTSC/PAL; en el moment de triar la pantalla, la senyal de video NTSC m'era totalment desconeguda, però la opció de trobar integrats que fan la pròpia conversió, traient-li feina així al microcontrolador fa d'aquesta pantalla una bona opció. Pot ser alimentada de 12 a 6 V, necessitant 150 i 350 mA respectivament, te un botons a la placa de control que permeten ajustar la lluminositat, el color i el contrast, es connecta directament a l'alimentació i a una senyal de video NTSC.

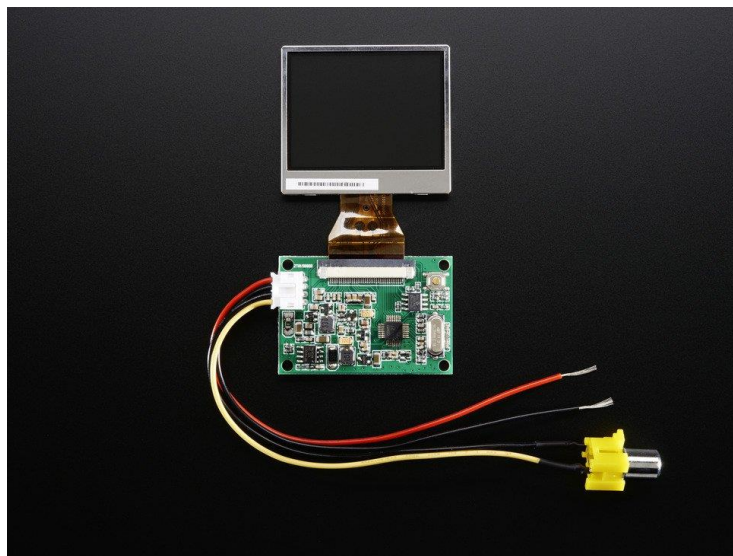


Figura 9. NTSC/PAL TFT Display. (<https://www.adafruit.com/product/911>).

4.5. El convertidor de video

Anteriorment hem comentat que una de las avantatges d'utilitzar una pantalla NTSC/PAL era l'existència de convertidors integrats, traient-li feina així al microcontrolador, buscant, del que s'ha trobat més informació ha estat l'AD725 utilitzat per projectes d'ARDUINO en que cal passar de RGB a NTSC (encara que ARDUINO ho pot fer per programació, igual que en el nostre cas se li treu pes al programa principal), i en el projecte UZEBOX (a partir de la busca de l'integrat AD725 s'ha trobat el projecte UZEBOX que es molt semblant al que s'està desenvolupant i oferirà alguna ajuda a l'hora de programar com podrem veure posteriorment).



Figura 10. Xip AD725. (https://es.aliexpress.com/price/ad725_price.html).

semblant al que s'està desenvolupant i oferirà alguna ajuda a l'hora de programar com podrem veure posteriorment).

Mitjançant el datasheet i exemples s'ha estudiat el seu comportament, l'AD725 necessita de tres senyals pel color (senyal RGB), primerament hem d'adaptar la sortida del nostre microcontrolador (ja que es de 5 V a nivell alt i l'entrada de cada color a l'AD725 es de 0,7 V), la senyal vermella i la verda estan formades per tres bits cada una i la blava per dos; tenint en compte que a la senyal RGB el bit més important haurà de valdre el doble que l'anterior i que segons el datasheet hi ha d'haver un divisor de tensió amb una resistència de 75 Ω abans de cada entrada com es veu a l'equació 1.

$$0,7 = 5 * \frac{75}{\frac{1}{R} + \frac{1}{2R} + \frac{1}{4R} + 75} \quad (eq.1)$$

$$R = 806,2 \Omega$$

Així, un cop obtinguts els valors de les resistències necessàries entre el microcontrolador i l'AD725 es pot seguir amb la connexió, parant atenció als pins 3 (entrada del rellotge), 5 (enable), 10 (sortida del senyal de video compost) i 16 (senyal de sincronisme horitzontal).

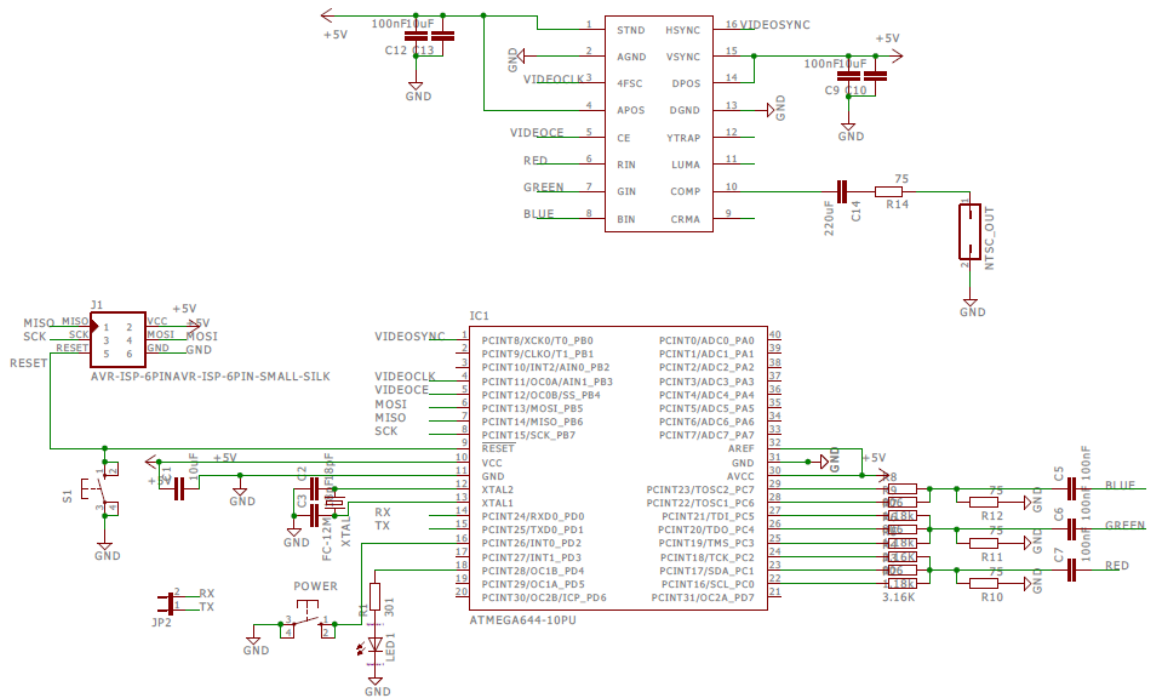


Figura 11. Connexió entre microcontrolador i AD725.

4.6. El Joystic

Pel control del jugador s'ha optat per fer un joystic integrat igual que els de la consola NES o S-NES (són iguals, però els de la S-NES tenen el doble de botons, necessitant per tant dos convertidors paral·lel sèrie).



Figura 12. Interior d'un comandament de NES. (<http://circusmoonday.blogspot.com.es>)

Els joystick d'una NES consten dels botons, que són una entrada paral·lel connectats a un CD4021B, que es un registre de conversió estàtic a 8 bits asíncron, que ens permet transformar l'entrada paral·lel a sèrie; necessita apart de les connexions amb els botons, una entrada de rellotge que obtenim del microcontrolador, i una senyal de *latch*, aquesta senyal, a l'estar activa fa que el CD4021 llegeixi l'estat dels botons, i a l'estar a nivell baix, el CD4021B llegeix la senyal de rellotge i transmet la informació en sèrie.

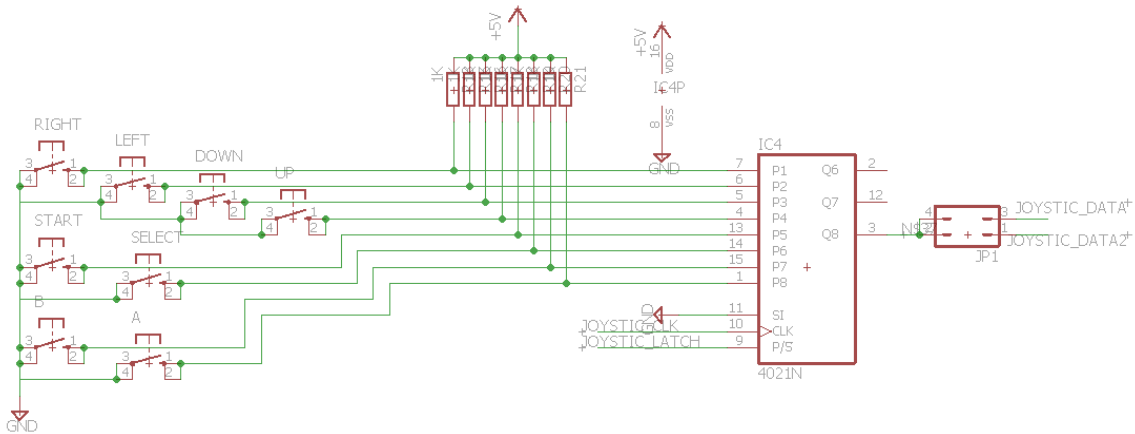
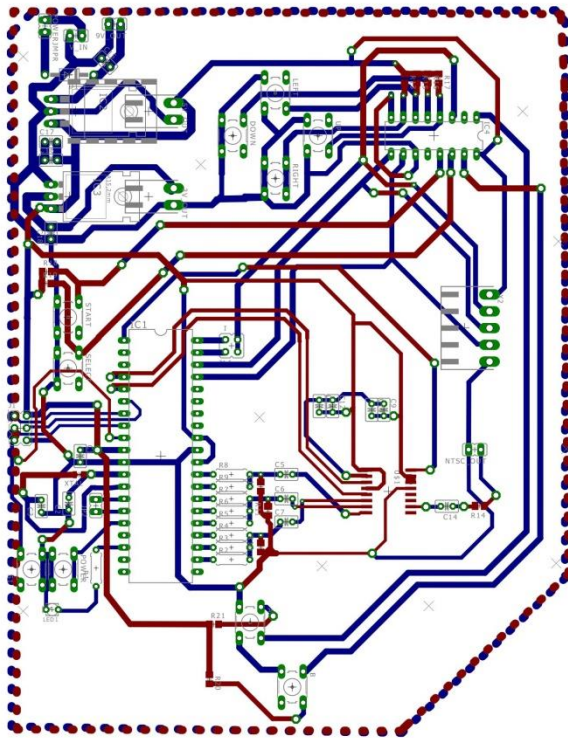


Figura 13. Connexió del CD4021B i els botons.

4.8. PCB dissenyada

18/05/2017 0:08 C:\Users\user\Desktop\TFG\eaagleftg\consola.brd



4.9. La carcassa

En el moment d'haver acabat tot el muntatge de la PCB, ens trobem que físicament es una estructura fràgil, ja que la pantalla no te cap suport i les piles tampoc, fent que perillin els cables que ho uneixen tot a la PCB; per aquesta raó es procedeix al disseny d'una carcassa.

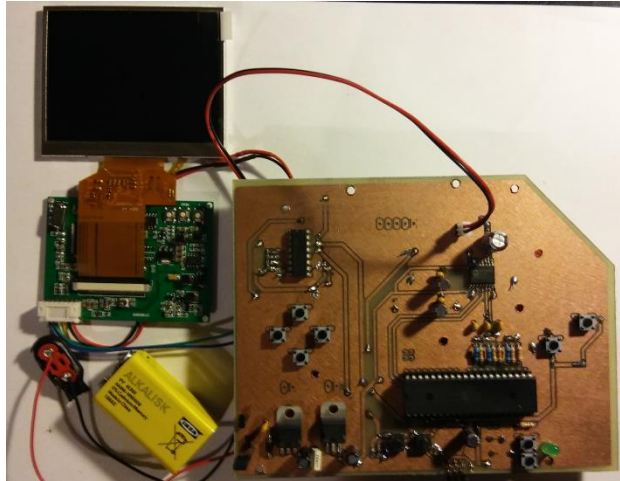


Figura 14. PCB, pantalla i pila sense suport.

Mitjançant el software Autodesk de 3d Studio Max s'ha dissenyat una carcassa amb el suport bàsic per a mantenir la PCB, la pantalla i la pila subjectades.

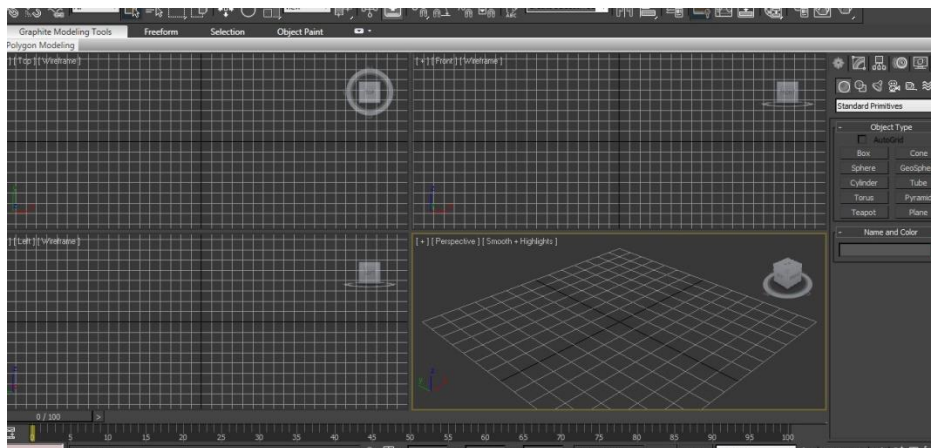


Figura 15. Interfície d'usuari de l'Autodesk.

Com es pot observar a la figura anterior, la interfície d'usuari de l'Autodesk consta de 4 subdivisions on podem veure la peça des de els tres eixos i en perspectiva.

La carcassa consta de tres peces. La primera, anomenada bottom, es la part inferior de la carcassa, conté un separador per a la pila, i unes subjectcions que posteriorment es foradaran per a subjectar la PCB.

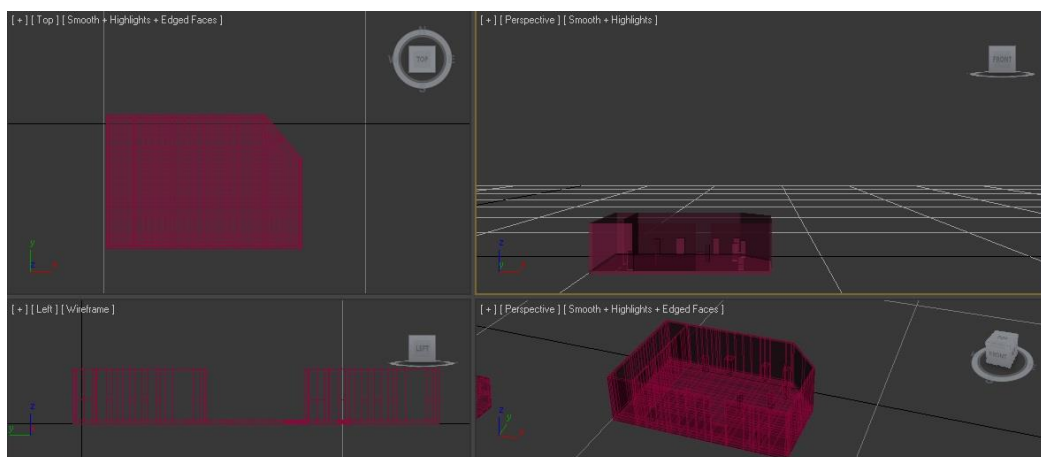


Figura 16. Vista de la peça bottom al programa Autodesk.

La *top* o tapa, es simplement una tapa amb els forats per a posar la pantalla i els botons.

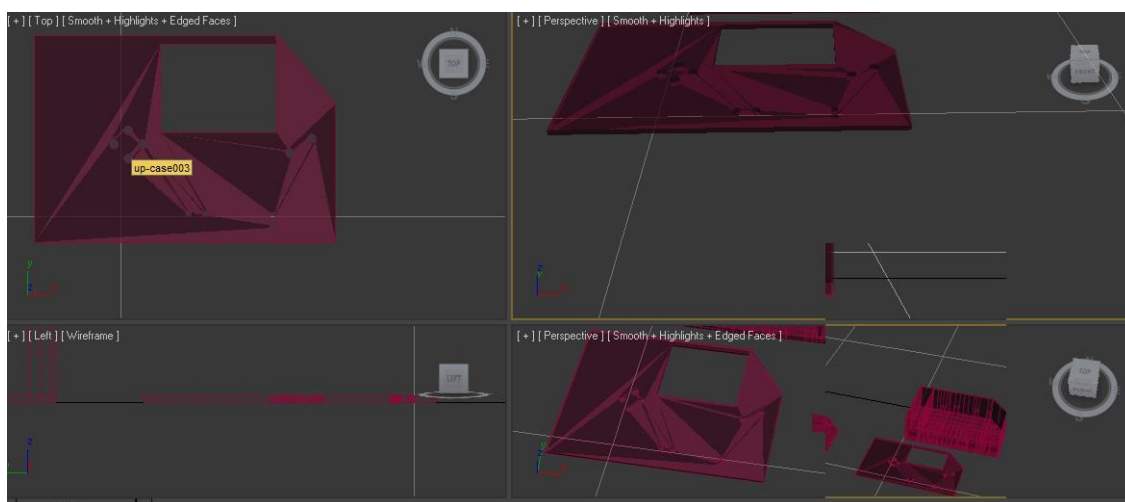


Figura 17. Vista de la peça top al programa Autodesk.

Els botons, són 6 de 3 mm de radi per als controls i quatre de 2 mm de radi per als botons de select, start, power, i reset.

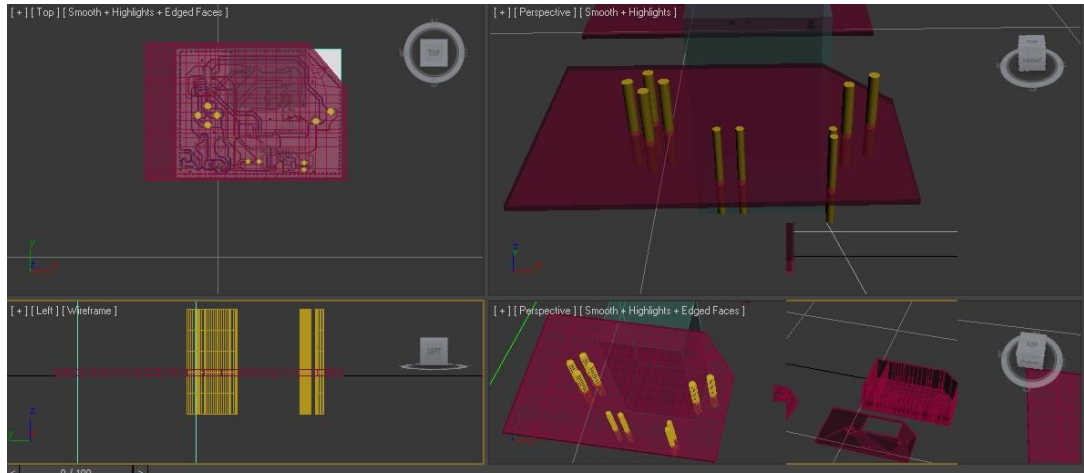


Figura 18. Vista dels botons amb la peça top.

4.10. Problemes i errors sorgits durant el muntatge

Durant el muntatge han aparegut errors i problemes que es comentaran en aquesta secció.

4.10.1. AD725

Primerament es va decidir de crear un prototip en una “placa de topos” en comptes d’una PCB, el problema era que diversos components eren SMD, així que es va haver de buscar una solució alternativa, utilitzant plaques de muntatge superficial, el problema amb aquestes es que venen pre-dissenyades per a components de certes mides.

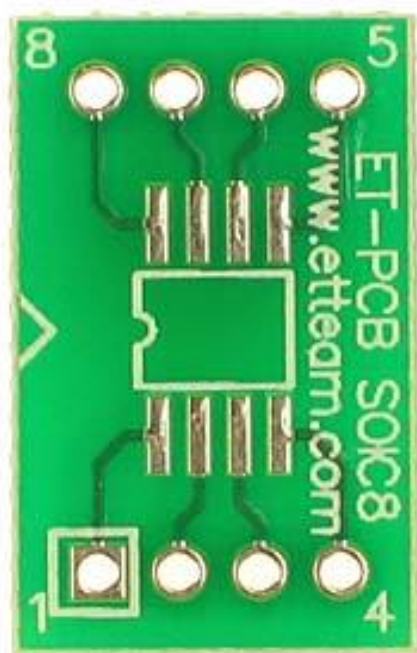


Figura 19. Exemple d'una placa de muntatge superficial.

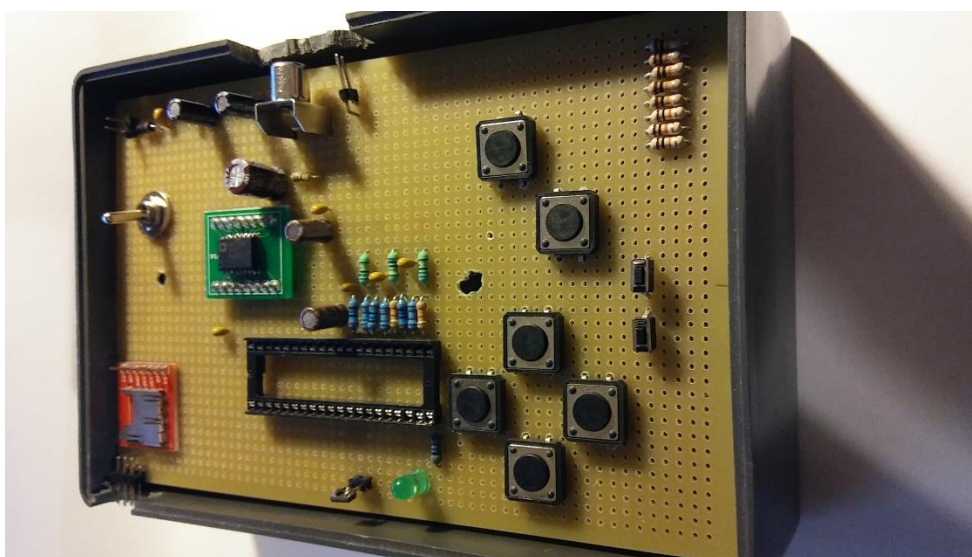


Figura 20. Prototip en una "placa de tops".

El problema al prototip ha estat que tot i utilitzar una placa de muntatge superficial, l'AD725 és 1,5 mm més ample que la placa (no havent-hi plaques de la mida de l'AD725 al mercat), raó per la que la soldadura es va complicar, havent de doblegar les potes i fent que finalment l'AD725, que es un xip molt sensible es fes malbé.

Després de comprovar que tot i el mal funcionament de l'AD725 la resta de connexions eren correctes i les senyals les adequades, es va optar per dissenyar directament la PCB on el problema d'adaptació de l'AD725 desapareixeria.

4.10.2. L'oscil·lador de quars

Per a la correcta creació de la senyal NTSC necessària per a obtenir un sincronisme correcte amb la pantalla, és necessari una freqüència de clock de 14,32 MHz, que obtenim d'un oscil·lador de quars de 28,64 MHz i un divisor de freqüència al microcontrolador. El problema de l'oscil·lador de quars es que te una tolerància, que fa que s'hagi d'estabilitzar; en el muntatge que podem observar el vibrador s'ha realitzat mitjançant dos condensadors de 18 pF, resultant una freqüència de rellotge lleugerament més baixa de la necessària i que per tant distorsiona la senyal a la sortida. Per tal de solucionar-lo s'han provat diverses solucions, des de, a partir d'un inversor d'alta velocitat rectificar els flancs de pujada i de baixada a proba runa tretzena d'oscil·ladors de quars diferents. Investigant podem veure que els oscil·ladors utilitzats en projectes d'aquest caire són de 63.63636 MHz i a l'actualitat han deixat de fer-se, sent substituïts pels de 28.636MHz. En el capítol de futures millores, podrem observar el circuit necessari per a solucionar aquest problema.

4.11. Prototip final



Figura 21. Diferents vistes del prototip final.

5. La senyal de video

La senyal de video amb la que opera la pantalla és una senyal NTSC (National Television Standards Committee), avui dia és un tipus de senyal cada cop més en desús, triant sobretot abans que una senyal NTSC senyals més senzilles de modular com pot ser VGA; el cas es que com a anteriors assignatures ja havia treballat amb VGA i una part important del TFG considero que es aprendre coses noves, es va triar treballar amb una senyal NTSC.

La senyal NTSC es especialment difícil de modular per raons de sincronisme, a continuació intentarem entendre una mica millor que és i com funciona la senyal NTSC.

La senyal NTSC està formada per 525 línies, integrant a la senyal tota la informació necessària, com els color i la sincronització.

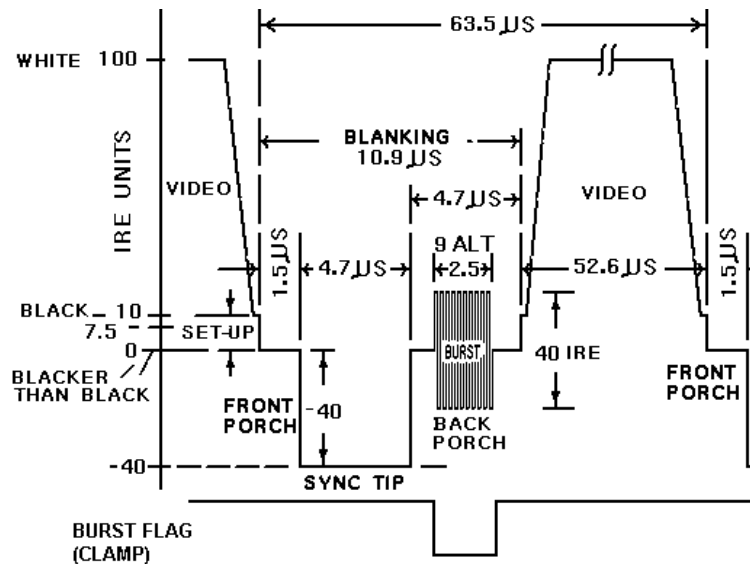


Figura 22. Senyal NTSC. (<http://fieldeffect.info>)

A la figura podem observar algunes de les següents parts de la senyal:

- Front Porch: És un pols pla de 0.3 V amb una durada d'uns 1,5 us.
- Horitzontal Sync: La senyal de sincronisme horitzontal que dura uns 4,7 us a 0 V.
- Back Porch: Una senyal de 2,5 us a 0,3 V.
- Video data: Senyal de 52,6 us, conté el valor de la luminància, descriu la imatge d'esquerra a dreta ; com més alt és el voltatge més brillant és el píxel.
- Vertical Sync Pulses: és una combinació de senyals de 0,3 i 0 V.

6. El software

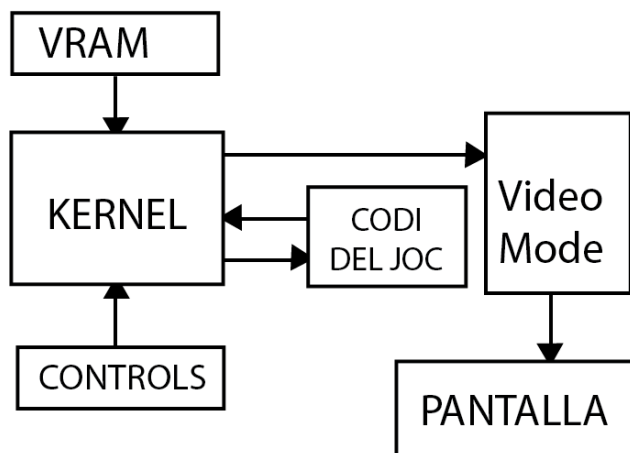


Figura 23. Esquema simplificat de l'organització i relació del software que conforma el joc.

El joc pensat, serà un joc de naus espacials que es mouen i desapareixen; serà un joc senzill per a una primera prova de contacte amb l'àmbit.

6.1. El nucli

El nucli (o kernel) és el software fonamental que permet a la resta de programes accedir al hardware, gestionar els recursos i controlar els temps.

Per al joc present s'ha utilitzat el nucli creat pel col·lectiu d'Uzebox, ja que aquest va ser dissenyat per a la finalitat de fer jocs basats en el mateix microcontrolador emprat en el projecte (encara que es pot fer servir per a molts més projectes visuals basats en el mateix hardware).

El nucli comentat s'encarrega de inicialitzar els ports, comptadors i perifèrics; generar la senyal de sincronisme necessitada pel convertidor de video AD725; llegir els controls. Funciona mitjançant interrupcions, a cada una de les quals guarda l'estat del programa que posteriorment a la tasca necessària per cada interrupció s'actualitza. Podem entendre el procés d'interrupcions a la següent figura:

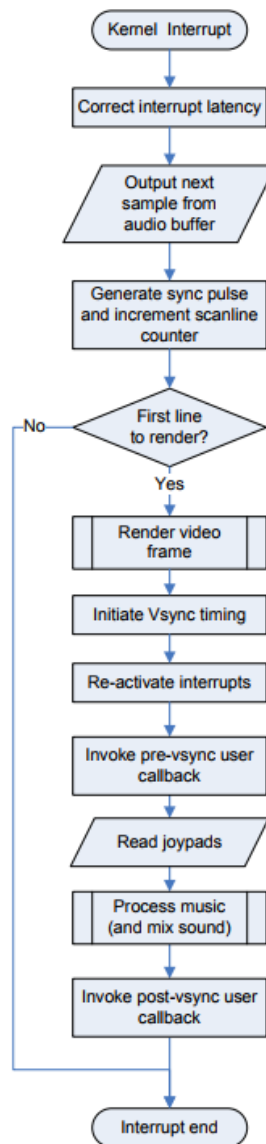


Figura 24. Procés d'interrupcions del nucli (amb *correct interrupt latency* es fa referència a la correcció dels temps de les interrupcions al no tenir l'AVR interrupcions deterministes. (uzebox.org).

6.2. Dibuix de la pantalla

Per a mostrar el video, s'ensenya una imatge completa cada sexagèsima part de segon, això s'ha fet utilitzant els "video modes" dissenyats per a funcionar amb el kernel utilitzat per una consola Uzebox. Aquest es basa en la tècnica que es feia servir a les consoles de primera generació, com ara la NES, que al no tenir els 64K de RAM que per exemple es necessiten per mostrar una sola imatge amb VGA

necessita de mètodes més enginyosos; basant el mostratge de la imatge en “tiles”, com es fa per exemple en el mostratge de pantalla basat en text, com a calculadores o displays LCD on la imatge es crea a partir de matrius de caràcters repetides en comptes de cada píxel per separat.

En concret mostrem una imatge de 240x224 píxels basada en “tiles” de 6x8 píxels (el número i creació d’aquest “tiles” es pot veure a l’apartat gràfic).

6.3. Senyal de sincronisme vertical

Quan s’acaba de generar la imatge sencera es genera la senyal de sincronisme vertical; en aquest moment es passarà a llegir els controls i la preparació de la següent imatge feta pel nucli.

6.4. Controls

Mentre la senyal de sincronisme vertical és alta es llegeix l’estat de la “botonera” i es guarda en dos *integers* que es poden llegir mitjançant la funció API *READJOYPAD()*.

6.5. Funcions API

En el programa s’han utilitzat les següents funcions API (Application Programming Interface):

- Drawmap: Dibuixa un mapa a les coordenades especificades.
- SetTileTable: Defineix el joc de “tiles” a utilitzar.
- ReadJoypad: Llegeix l’estat dels controls.
- WaitVsync: Espera que passin una quantitat concreta de mostrejos.

6.6. Apartat gràfic

La part més senzilla, al ser un joc de 8 bits es creen *tiles*, o parts d’una imatge que es col·loquen generant la imatge global; per a estalviar memòria ho idoni es poder reciclar els *tiles*, es a dir, amb 4 o 5 poder generar moltes ambientacions.

Per al dibuix es pot utilitzar qualsevol programa de dibuix en píxel, però s’ha triat *TileStudio*, ja que aquest facilita la posterior conversió del dibuix en una matriu de números hexadecimals.

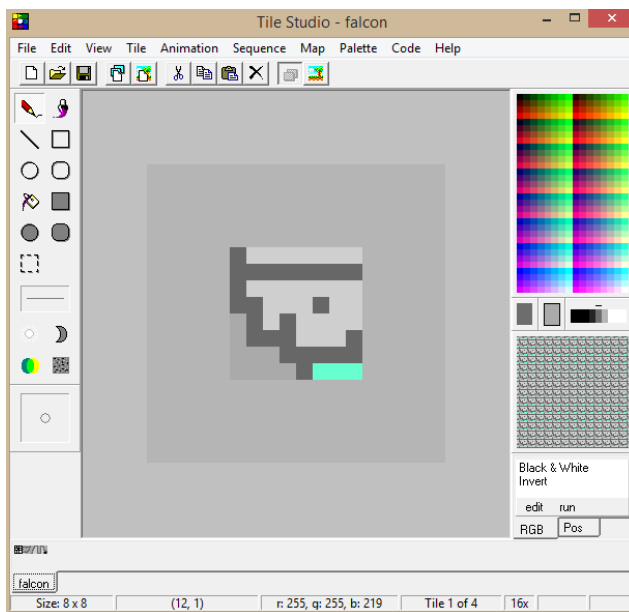


Figura 25. Finestra de *TileStudio* amb un dels quatre *tiles* que formen la nau principal.

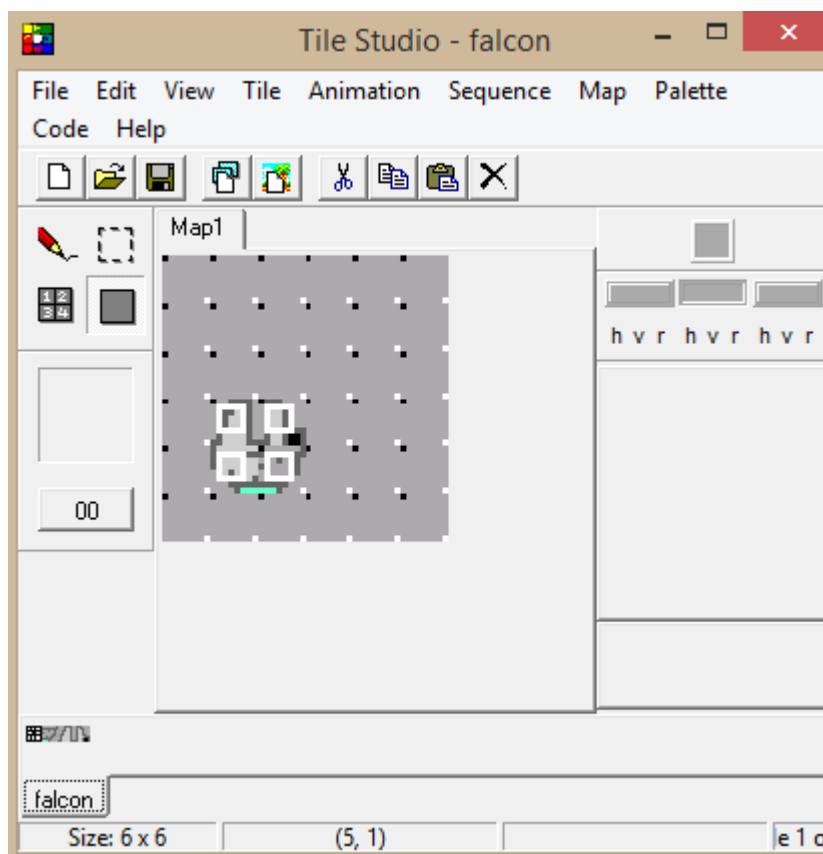


Figura 26. Finestra de *TileStudio* on s'endrecen els quatre *tiles* que formen la nau.

Posteriorment la imatge de la nau es transforma en quatre matrius (una per a cada *tile*) i una matriu on es defineix l'ordre d'aquestes matrius.

```
// falcon
// TileWidth:8
// TileHeight:8
// TileCount:4

const char falcon[] PROGMEM ={
/* tile 0 */

    0x00,0x5B,0x5B,0x5B,0x5B,0x00,
    0x00,0x5B,0xF6,0xF6,0x5B,0x00,
    0x5B,0x5B,0xF6,0xF6,0x5B,0x00,
    0x5B,0xF6,0xF6,0xF6,0x5B,0x00,
    0x5B,0xF6,0xF6,0xF6,0x5B,0x00,
    0x5B,0xF6,0xF6,0xF6,0x5B,0x00,
    0x5B,0xF6,0xF6,0xF6,0x5B,0x00,
    0x5B,0xF6,0xF6,0xF6,0x5B,0x5B,
/* tile 1 */
    0x5B,0x5B,0x00,0x00,0x00,0x00,
    0xF6,0x5B,0x5B,0x00,0x00,0x00,
    0xF6,0xF6,0x5B,0x00,0x00,0x00,
    0xF6,0xF6,0x5B,0x00,0x00,0x00,
    0xF6,0xF6,0x5B,0x5B,0x00,0x00,
    0xF6,0xF6,0xF6,0x5B,0x5B,0x5B,
    0xF6,0xF6,0x5B,0x00,0x00,0x5B,
    0xF6,0xF6,0x5B,0x00,0x00,0x5B,
```

Figura 27. Primeres dos matrius que formen el *tile set* de la nau.

```
#define falcon_Map1_Width    (2)
#define falcon_Map1_Height  (2)

const int falcon_Map1[] PROGMEM=
{falcon_Map1_Width,falcon_Map1_Height,
  0, 1,
  2, 3

};
```

Figura 28. Matriu on s'endrecen les matrius dels *tiles*.

Posteriorment es fa el mateix amb l'escenari, la nau enemiga i la imatge del "làser" que dispara la nau.

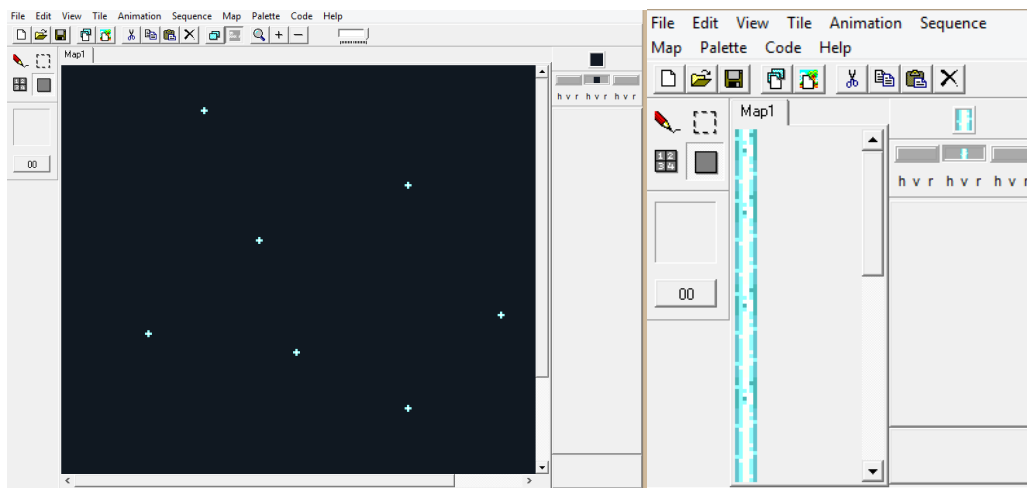


Figura 29. Escenari i imatge del làser.

6.7. Codi del joc

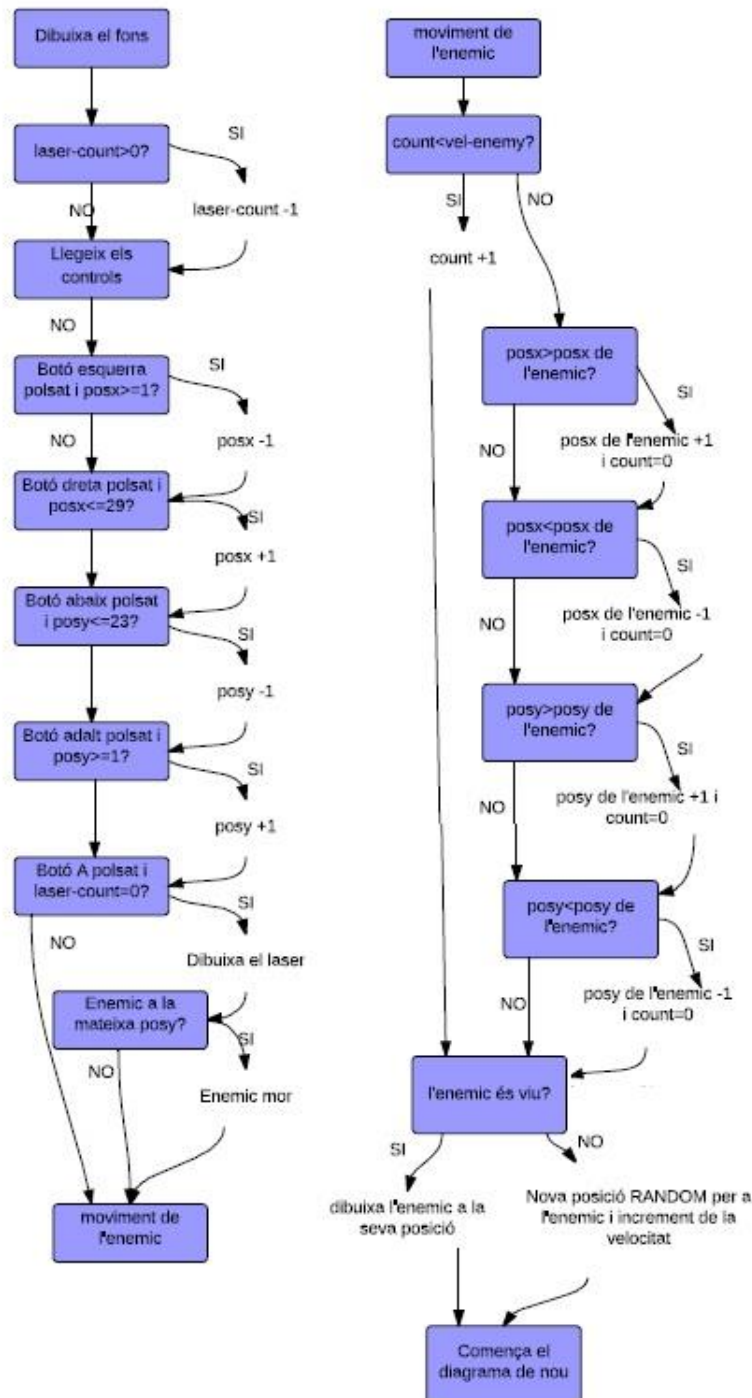


Figura 30. Diagrama de blocs amb el funcionament bàsic del joc.

El funcionament de les físiques del joc es força senzill, s'ha programat en llenguatge C, utilitzant WIN AVR de AVR STUDIO com a compilador.

Primer es dibuixa l'escenari a la pantalla.

```
SetTileTable(Tiles1); // estableix el conjunt de gràfics a utilitzar
DrawMap(0, 0, Tiles1_Map1); // dibuixa a la pantalla el mapa inclòs en el conjunt Tiles 1
    // contador del tret laser.
if (laser_count > 0) {
    laser_count--;
}
```

Figura 31. Codi referent al dibuix de l'escenari.

S'estableix el moviment de la nau del jugador llegint les entrades i es dibuixa la nau a la posició actual.

```
void moviment () {
    SetTileTable(falcon); // estableix el conjunt de gràfics a utilitzar
    // aquest cop el de la nau

    c = ReadJoypad(0); // Llegeix les entrades del joystick
    if (c & BTN_LEFT && posx >= 1) {
        posx--;
    }
    if (c & BTN_RIGHT && posx <= 29) {
        posx++;
    }
    if (c & BTN_DOWN && posy <= 23) {
        posy++;
    }
    if (c & BTN_UP && posy >= 1) {
        posy--;
    }
    DrawMap(posx, posy, falcon_Map1); // Dibueixa la nau a la posició actual
}
```

Figura 32. Codi del dibuix i moviment de la nau del jugador.

Quan es polsa el botó A, la nau ataca, això succeeix si ha passat el temps suficient des de l'últim tret, per a evitar un efecte "metralleta".

```

//Codi de la funció d'atac
void attack () {
    SetTileTable(laser); //Estableix el conjunt de gràfics del làser
    c=ReadJoypad(0); //Si s'apreta el botó A i ha passat el temps suficient des de
                    //l'últim tret, dispara seguint una vertical des de la posició de la nau.
    if (c&BTN_A && laser_count==0) {
        DrawMap(posx, posy-23,laser_Map1);
        laser_count=30;
        if (posx_enemy==posx && posy_enemy<posy) {
            enemy_alive=false;
        }
    }
}

```

Figura 33. Codi pertanyent a la funció d'atac de la nau del jugador.

Finalment es fa aparèixer la nau enemiga que persegueix la nau del jugador i cada cop que el jugador destrueix la nau enemiga, una nova apareix sent més ràpida que l'anterior, augmentant així la dificultat.

```

//Codi de la funció de la nau enemiga
void enemymov(){
    SetTileTable(tie); //Estableix el conjunt de gràfics de la nau enemiga
    //La posició de la nau enemiga segueix a la nau principal.
    if (count < vel_enemy) {
        count++;
    }
    else if (count >= vel_enemy) {
        if (posx > posx_enemy){
            posx_enemy++;
            count = 0;
        }
        if (posx < posx_enemy){
            posx_enemy--;
            count = 0;
        }
        if (posy > posy_enemy){
            posy_enemy++;
            count = 0;
        }
        if (posy < posy_enemy){
            posy_enemy--;
            count = 0;
        }
        //Si l'enemic està viu, el dibuixa, si ha mort augmenta la velocitat del següent.
        if (enemy_alive==true) {
            DrawMap(posx_enemy, posy_enemy,tie_Map1);
        }
        else {
            posx_enemy=rand() % (32);
            posy_enemy=rand() % (15);
            enemy_alive=true;
            if (vel_enemy>1) {
                vel_enemy--;
            }
        }
    }
}

```

Figura 34. Codi referent al moviment de la nau enemiga.

Finalment s'han afegit algunes millores un cop la base del joc ha estat finalitzada, un exemple és la pantalla d'inici, que apareix fins que es polsa el botó de disparar.



7. Anàlisi de l'impacte ambiental

L'anàlisi de l'impacte ambiental ha estat basat en les idees de 'L'economia del bé comú' de Christian Felber, puntuant els factors de la producció de l'1 al 10 sent un 10 un treball sostenible respecte als criteris de cada apartat, la vida útil de 0 a 20 sent un 20 un treball sostenible al llarg de tota la seva vida útil en cada camp i els riscos de -20 a 0, sent 0 un treball on no s'han trobat riscos possibles i -20 un treball amb riscos potencials i perillosos:

	Producció	Vida útil	Riscs
Ambiental.	Consum del disseny 8	Petjada ecològica 14	Riscs ambientals 0
Econòmic.	Factura 7	Pla de viabilitat 18	Riscs econòmics -3
Social.	Impacte personal 9	Impacte social 9	Riscs socials 0
Rang de sostenibilitat.			62/90

Consum del disseny: El consum es considerablement baix, ja que el prototip és alimentat per una pila recarregable de 9 V; deixant espai suficient per a la millora d'adaptar-li una bateria recarregable.

Petjada ecològica: La petjada ecològica, com la de tots els productes electrònics, és mitjanament alta, considerant per exemple, per a la fabricació d'un xip de memòria Ram, es gasten 1 Kg de combustibles fòssils, 20 l d'aigua i uns 70 g de diversos productes químics. Apart productes com a ara una consola, tenen una mitjana de vida curta degut a la ràpida evolució del camp; no obstant, s'ha de tenir en compte que la consola dissenyada, es un model obert, on qui vulgui pot programar jocs i instal·lar-los, fent-la perfecta per ambients formatius o per a estudi i millora de la programació, augmentant així dràsticament la seva vida útil. Per una altra part la carcassa, al ser dissenyada per a imprimir-se amb una impressora 3D es altament reciclable, com es pot veure a projectes com el de re-filament (<http://www.re-filament.com/>).

Riscs ambientals: Els únics riscos ambientals són el mal tractament en cas de llençar-se, es a dir no reciclar les piles o els components electrònics.

Factura: El preu total del material i de les hores pagades al treballador per la realització del projecte, fan una factura de 7016.44 €.

Pla de viabilitat: És un projecte plenament viable, especialment per a l'àmbit formatiu.

Riscs econòmics: El risc econòmic més important és un que ja ha succeït durant l'elaboració del primer prototip, i és el mal funcionament de l'AD725, un integrat especialment sensible a temperatura i descarregues d'estàtica, a més, l'AD725 és el component més car, amb un cost d'11 euros.

Impacte personal: Ha tingut varis i diferents impactes sobre l'estudiant; ha aprofundit força el seu coneixement sobre microcontroladors, en C# i en senyals de video, ha causat altes dosis d'estrès en els moments en que el primer prototip no funcionava, i finalment l'orgull d'haver creat un projecte així.

Impacte social: L'impacte social es baix, a no ser que s'implanti en alguna assignatura com a eina per a pràctiques.

Riscs socials: La idea d'utilitzar-la com a eina educativa, en un país on cada cop s'inverteix menys en l'àmbit educatiu, fa que si algun centre cometés l'error de fer pagar a cada alumne d'una classe els costos de la fabricació d'un prototip poguessin sorgir problemes de caire personal a l'alumnat.

8. Conclusions

S'ha portat a terme el disseny i prototipat d'una consola de 8 bits, basada en les consoles de primera generació NES; basada en un microcontrolador ATMEGA644 i s'ha triat fer servir una senyal de video NTSC per a comprendre de més a prop com es feien aquestes consoles en el seu moment.

La consola consta de l'alimentació, el microcontrolador, l'adaptació de la senyal de video i el joystick integrat.

Apart s'ha programat un senzill joc de naus, d'un estil similar al clàssic Space Invaders que segueix obert a futures millores.

La realització d'aquest treball ha comportat diverses i grans dificultats que m'han apropat a les diverses problemàtiques que sorgeixen al dissenyar un prototip, com ara, a la realització d'un primer prototip que no funcionava, haver d'analitzar el funcionament per etapes de la consola, comprenent amb total exactitud quin havia de ser el comportament esperat en tot moment de cada etapa per a poder trobar la que fallava; i en el moment de trobar-la analitzar quina serà la resposta més adient per a corregir l'error, arribant a la conclusió que la millor via era la repetició total del prototip basant-se en una PCB en comptes de en una "placa de topes".

En alguns moments ha resultat frustrant, especialment al comprovar que el que fallava eren components que no es poden adquirir a les botigues presencialment i s'aturava el muntatge 2 o 3 setmanes a l'espera del recanvi. Tot i això valoro molt positivament l'experiència i la sensació que deixa tot treball (especialment un on finalment tens un prototip funcional) que ha requerit de tant esforç.

8.1. Futures millores

La primera i més important de les millores es la substitució de l'oscil·lador de cristall de quars emprat en el projecte, que degut a la seva falta de precisió afecta lleument la pantalla, per un oscil·lador programable.



Figura 35. Oscil·lador programable DS1077 (todoelectronica.com).

Un bon exemple és l'oscil·lador DS1077, el qual es pot programar per anar de 16.2 KHz a 133 MHz.

La millora no s'ha fet en el moment d'entrega per considerar-se que els costos en material general durant el projecte ja han superat el pressupostat (al necessitar components que actualment no es venen aquí els costos d'enviament han suposat un important augment del pressupost) .

La segona millora plantejada consisteix a afegir un mòdul *bluetooth* (la placa s'ha deixat preparada per a que no fos necessari refer-la al voler-lo afegir) que permetria jugar en plataforma multi jugador o fer servir la consola com a comunicació amb altres objectes amb comunicació *bluetooth*.

8.2. Valoració personal

La meua valoració final és molt positiva, tot i diversos moments de tensió i/o por que ha pogut suposar el projecte, també m'ha generat moltes ganes de seguir dissenyant els meus propis prototips basats en microcontroladors, fent-me patent com podem fer gairebé qualsevol cosa començant des de una escala tan petita, i veient la diversió que pot comportar triar aquesta via en comptes de només fer servir mòduls comercials com ara Arduino o Raspberry.

També acabo el projecte amb la certesa que els meus coneixements en diversos camps s'han vist incrementats de forma important; des de el coneixement més pràctic sobre l'ús de

microcontroladors, les senyals de video alternatives, la programació en C# a l'extensió del catàleg de components per a les mil i una possibles necessitats que poden sorgir.

Finalment crec que sobretot he tret profit de les opcions d'autoaprenentatge que oferia fer aquest projecte.

9. Pressupost

9.1. Cost material

Quantitat	Nom	Informació extra	Preu	Preu total
1	ATMEGA644-10PU	ATMEL	5.83 €	5.83 €
1	NTSC/PAL (Television) TFT Display -3.5" Diagonal	ADAFRUIT NTSC/PAL (television) TFT Display	44.95 €	44.95 €
1	AD725	Conversor RGB a NTSC/PAL	11.6 €	11.6 €
1	Placa de circuit imprès positiva de dues cares.		7.27 €	7.27 €
1	XTAL smd 28, 6464 MHz	Farnell	3.22 €	3.22 €
3	Condensador 10 uF	C-EU025-025X050	0.02 €	0.06 €
2	Condensador 18 pF	C-EU025-025X050	0.04 €	0.08 €
6	Condensador 100nF	C-EU025-025X050	0.1 €	0.6 €
1	Condensador 220 uF	C-EU025-025X050	0.77 €	0.77 €
1	Condensador 330	C-EU025-	0.65 €	0.65 €

	nF	025X050		
2	Condensador 1 uF	C-EU025-025X050	0.06 €	0.12 €
1	Diode 1N4004		0.4 €	0.4 €
10	Polsadors	switch-omron	0.15 €	1.5 €
1	Diode LED verd		0.24 €	0.24 €
1	7805CV3	Regulador 5 v	0.23 €	0.23 €
1	LD33V	Regulador 3.3 V	0.2 €	0.2 €
1	4021N	Registre de conversió estàtic a 8 bits asíncron	0.3 €	0.3 €
2	Tires de pins		1 €	2 €
1	301 Ω R		0.02 €	0.02 €
2	3.16K Ω R		0.01 €	0.02 €
3	1.18K Ω R		0.02 €	0.06 €
3	806 Ω R		0.04 €	0.12 €
4	75 Ω R	smd	0.56 € (pack de 25)	0.56 €
8	1K Ω R	smd	0.64 € (pack de 25)	0.64 €
1	Impressió peces 3d	Imprés al FabLab de l'EEBE.	35 €	35 €
				Total: 116.44 €

9.2. Cost associat a les hores

A l'hora de calcular el cost total del projecte s'ha de tenir en compte els sous dels treballadors que hi ha participat; tenint present que el sou mig d'un enginyer és de 15 € la hora, calcularem el cost de les diferents activitats dutes a terme pels treballadors, en aquest cas, l'alumne realitzador del projecte:

Activitat	Hores	Cost
Recerca d'informació.	50 h	750 €
Estudi de les diferents opcions i components.	35 h	525 €
Disseny de l'esquemàtic	30 h	450 €
Aprenentatge del diferent software utilitzat (AVRdude, AVRstudio, ATMELstudio, Autodesk 3d max, GIMP)	50 h	750 €
Muntatge del prototip	35 h	425 €
Busca d'errors i correcció	40 h	600 €
Muntatge del model final	25 h	375 €
Busca d'errors del model final	15 h	225 €
Programació de les múltiples versions del joc	100 h	1500 €
Simulacions del joc	20 h	300 €
Disseny de la carcassa	10 h	150 €
Busca de millores i correccions	20 h	300 €
Escrit de la memòria	30 h	450 €
Total:	460 h	6900 €

9.3. Cost total

Per tant, estaríem parlant d'un cost total de 6900 € + 116.44 €

És a dir 7016.44 €.

Bibliografia

The Black art of video game console design. André LaMothe. Adquirible a: ic0nstrux.com

Tecnologias libres para síntesis de imagen digital tridimensional. Javier Alonso, Albusae Jiménez, Fernando Arroba Rubio, Javier BelancheAlonso, etc. Adquirible a: < books.google.es>.

A game programming library. Web de la llibreria Allegro. <liballeg.org>.

AVR programación en C. Tutorial en varios cursos sobre programación de AVR. <vidaembebida.wordpress.com>.

7" TFT display controlled by PIC. <microchip.com/fòrums>.

Learn C++. Tutorial de C++. <tutorialspoint.com>.

Graphics Library. Secció de llibreries, tutorials i eines de Microchip. <microchipdeveloper.com>.

Almacenamiento en memòria externa EEPROM. Sergio Castaño. <controlautomaticoeducacion.com>

Gameduino. Exemples i tutorials per programar una Gameduino. <excamera.com>

Guía aprender programar videojuegos con C++. Adrigm. <razonartificial.com>

Lenguajes de Programación para PICs. Luis Roberto Arteaga. Blog sobre proyectos de Hardware, Firmware y Software. <electrònica.de.desarrollo.blogspot.com.es>

Single chip game console. Voja Antonic. [The simple DIY retro game controller with PIC MCU which generates VGA signal, music and sound effects].<hackaday.io/projects/>.

Reading an SD card with an ATMEGA 168. <elasticsheep.com>.

Ejemplos en C para 16F648A. Zener. <todopic.com>



Annex A

1.Esquemàtic.

2.Circuit.