

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2016-2017

Pràctica 1 – Fase 3

LSHangman

Alumnes	Login	Nom
	Ls30759	Samuel Tavares da Silva
	Ls30652	Gabriel Cammany Ruiz

Entrega	Placa	Memòria	Nota

Data	28/03/2017
-------------	-------------------

Portada de la memòria

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2016-2017

Pràctica 1 – Fase 3

LSHangman

Alumnes	Login	Nom
	Ls30759	Samuel Tavares da Silva
	Ls30652	Gabriel Cammany Ruiz

Entrega	Placa	Memòria	Nota

Data	28/03/2017
-------------	-------------------

Portada de l'alumne

Índex

1. Síntesi de l'enunciat	4
2. Plantejament.....	5
3. Diagrama de mòduls	6
4. Disseny.....	8
4.1. Disseny del SSPE.....	8
4.1.1. Unitat de control	8
4.1.2. Unitat de procés	¡Error! Marcador no definido.
4.2. Implementació del SSS	¡Error! Marcador no definido.
4.3 Fitxers .pld de les GAL's	¡Error! Marcador no definido.
5. Esquemes elèctrics	10
6. Problemes observats.....	10
7. Conclusions.....	12

1. Síntesi de l'enunciat

Es requereix la implementació del LSHangman, també conegut com el clàssic model de joc el penjat, aquest només contindrà paraules de com a màxim 8 lletres, per lo que s'ha decidit que l'usuari per poder guanyar la partida tingui que completar una paraula sencera sense cometre més de 8 intents.

El primer sistema es basa en la selecció d'una lletra, pe tant per poder comunicar-se amb el sistema s'utilitzarà un teclat matriu de mida 3x3 el qual simularà un teclat alfanumèric, típicament utilitzat en els mòbils. El sistema controlarà el temps que l'usuari es mantingui amb la mateixa tecla un temps superior o igual a 2 segons, cal destacar que cada lletra s'haurà de mostrar per un display de 16 segments.

El segon sistema seleccionarà aleatòriament una paraula, dins un conjunt de com a màxim 255 paraules, seguidament indicarà mitjançant una sèrie de 8 leds quines lletres l'usuari ha encertat, també s'utilitzarà un 16 segments amb la mateixa finalitat però mostrar les lletres que l'usuari ha endevinat, cal destacar que aquest sistema també és l'encarregat de controlar les vides de l'usuari, en cas que aquest superi els 8 intents s'il·luminarà un LED que indiqui el Game Over.

Finalment el tercer sistema mitjançant l'ús d'un microcontrolador i un cable VGA mostrarem la part gràfica de la pràctica. A mesura que l'usuari vagi realitzant errors al seleccionar lletres per conèixer la paraula, s'anirà mostrant les diferents parts del penjat. En el cas que l'usuari realitzi més de 8 errors es completarà el ninot penjat i es mostrarà la figura de color vermell, amb la intenció d'indicar que s'ha perdut la partida. En el cas que l'usuari completi la paraula correctament, la figura que s'estigui mostrant per pantalla es posarà de color verd, de manera que quedi clar que ha guanyat la partida.

Cal destacar que en tot moment s'ha de mostrar el número corresponent al grup assignat a inici de curs.

2. Plantejament

Aquesta fase té un plantejament diferent a les altres dos, ja que la part de hardware és gairebé mínima i es centra més en la programació del software.

Per tant, per la programació del hardware utilitzarem el llenguatge Assembler que ens proporcionarà totes les eines necessàries per controlar el funcionament de la PIC i programar per mostrar la part corresponent del penjat per pantalla.

Aquesta fase així com les demes, s'ha de plantejar i resoldre els blocs poc a poc perquè cada un funcioni correctament. Així que el més important és assegurar que la soldadura és la correcta i que s'ha soldat tal com es mostra en el esquema elèctric que ens han proporcionat.

Un cop que ja tenim soldat i provat que funciona correctament, comencem per el més bàsic de control de pantalles, és a dir, el VSync i el HSync. Que incloïa el càlcul de el temps de 1 o 0 de cada senyal perquè la pantalla pugui reconèixer correctament el senyal VGA.

Un cop comprovat que la sincronització d'aquestes dos senyals és correcta, procedim a pintar per pantalla. Per realitzar-ho, ens basarem en el control dels colors RGB que té el senyal VGA activant el color que volem representar en cada moment per pintar en un moment donat una part o altre.

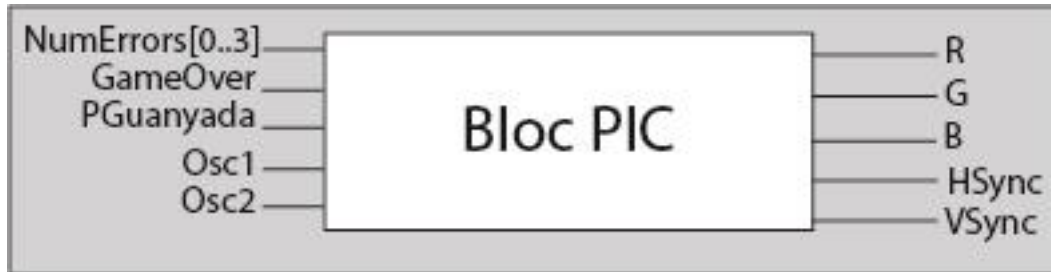
La nostra manera de pintar per pantalla es basa en comprovar depenent del moment del joc si en el lloc on estem actualment s'ha de pintar o no, és a dir, no definim cada objecte, sinó que definim que s'ha de pintar en la línia corresponent.

L'operació més utilitzada per aquesta fase és "NOP", que és un acrònim de, no operation. Ja que per poder pintar per pantalla les parts del penjat que vulguem, hem de definir quin color en cada moment i el temps que aquest color estigui actiu. Tot això serà realitzat a partir d'afegir o treure NOPs mentre pintem o esperem.

3. Diagrama de mòduls

3.1 Físic

3.1.1 Bloc PIC



- El bloc PIC, es l'encarregat de rebre les senyals provinents de les fases anteriors, relacionades amb la lògica del joc, mentre que també s'encarrega de generar les senyals per el VGA, tant colors com senyals de sincronització.

3.1.2 Bloc



XTAL

- Bloc XTAL, es el bloc encarregat de generar els oscil·ladors, senyals que actuen com la típica senyal generada per el NE555 que s'encarrega de realitzar la sincronització del sistema.

3.1.2

Bloc VGA



- Bloc XTAL, es el bloc encarregat de generar els oscil·ladors, senyals que actuen com la típica senyal generada per el NE555 que s'encarrega de realitzar la sincronització del sistema.

3.2 Lògic

3.2.1 Bloc Sincronització



- Bloc sincronització, la part més important per poder dibuixar per pantalla es la sincronització de les senyals de HSync i VSync. Així que aquest bloc serà el encarregat de poder dur a terme la sincronització, comprovació i el control .

3.2.2 Bloc Pintar



- Finalment tenim el bloc per pintar, que tal com especifica el seu nom, es l'encarregat de pintar per pantalla depenent de la línia que estigui, determinarem la zona corresponent i el color necessari, de manera que sempre surti el color desitjat.

3.2.3 Bloc Color



- Bloc color, aquest bloc es l'encarregat de que en funció del numero d'errors i l'estat de la partida , es el ninot del penjat es pinti de color blanc, vermell o verd, indicant cada color un estat de la partida diferent.

4. Disseny

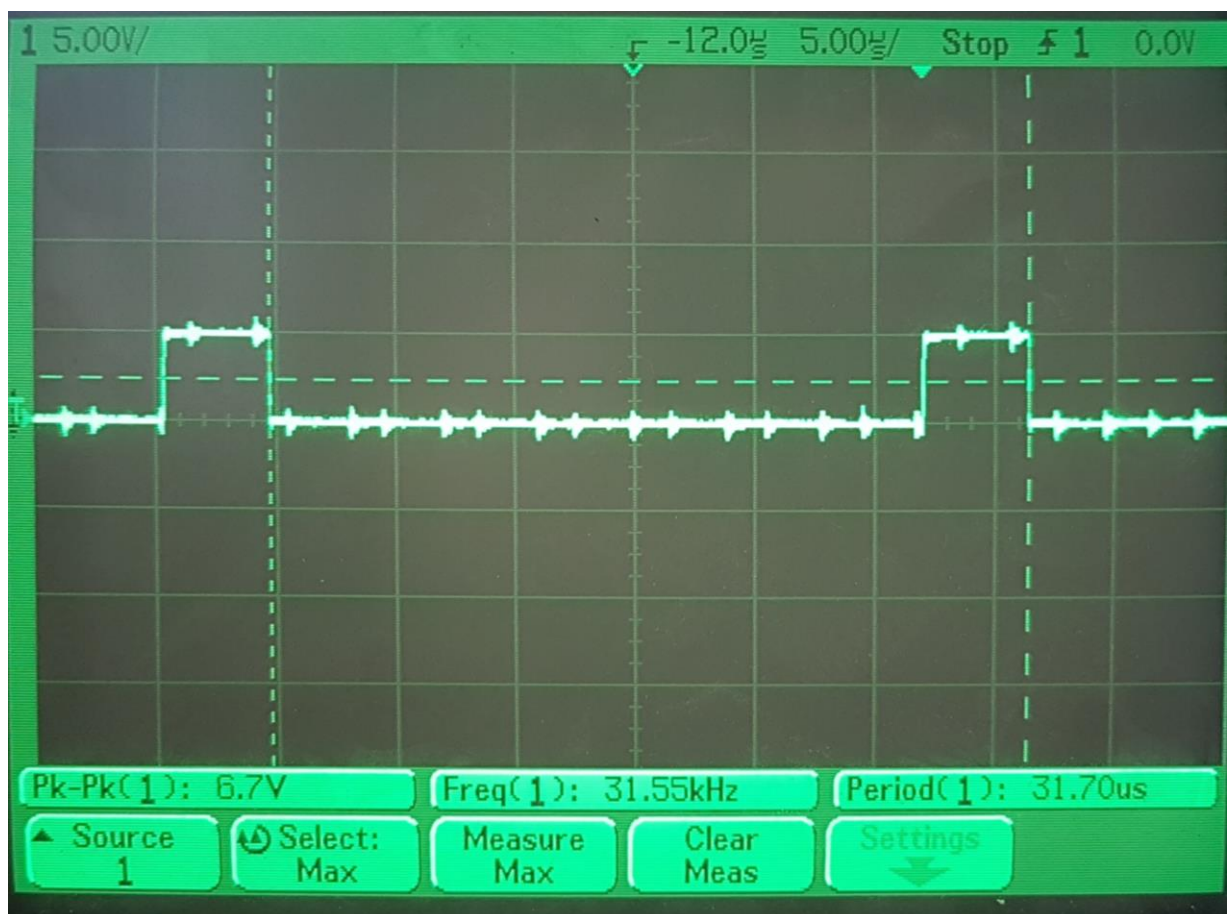
4.1. Configura el micro-controlador

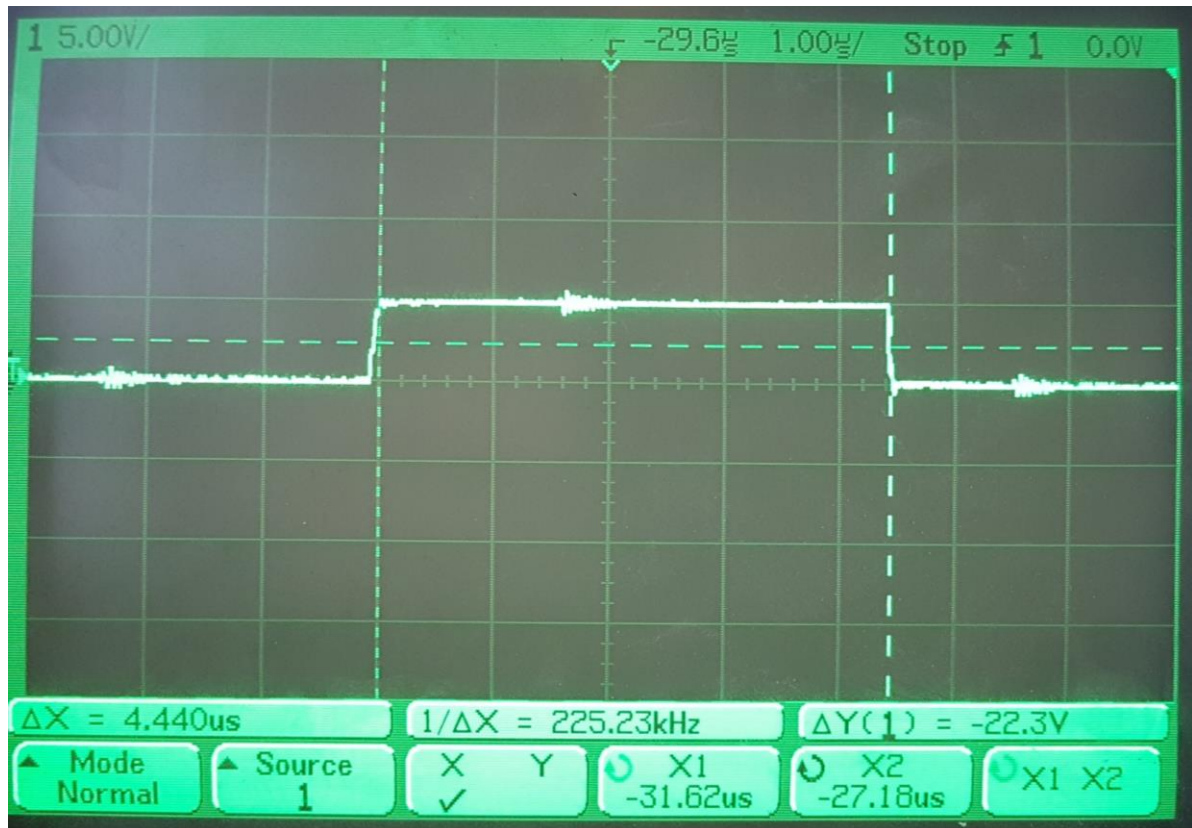
La configuració del micro-controlador es basa en diferents parts:

- Oscil·lador, per poder dur a terme tota la fase es necessita controlar la oscil·lació per determinar la velocitat de la PIC. Per tant, com que tenim un input a 10Mhz, haurem d'utilitzar la configuració de HSPLL i per tant multiplicar aquesta freqüència per 4 perquè com mes endavant aquesta es divideix entre 4 així ens quedem amb una freqüència de 10Mhz.
- Ports. La configuració dels ports ha sigut la següent:
 - Ports A. Tots aquests ports s'han configurat com a sortida, sense tenir en compte cap dels dos ports que utilitza el oscil·lador.
 - Ports B.

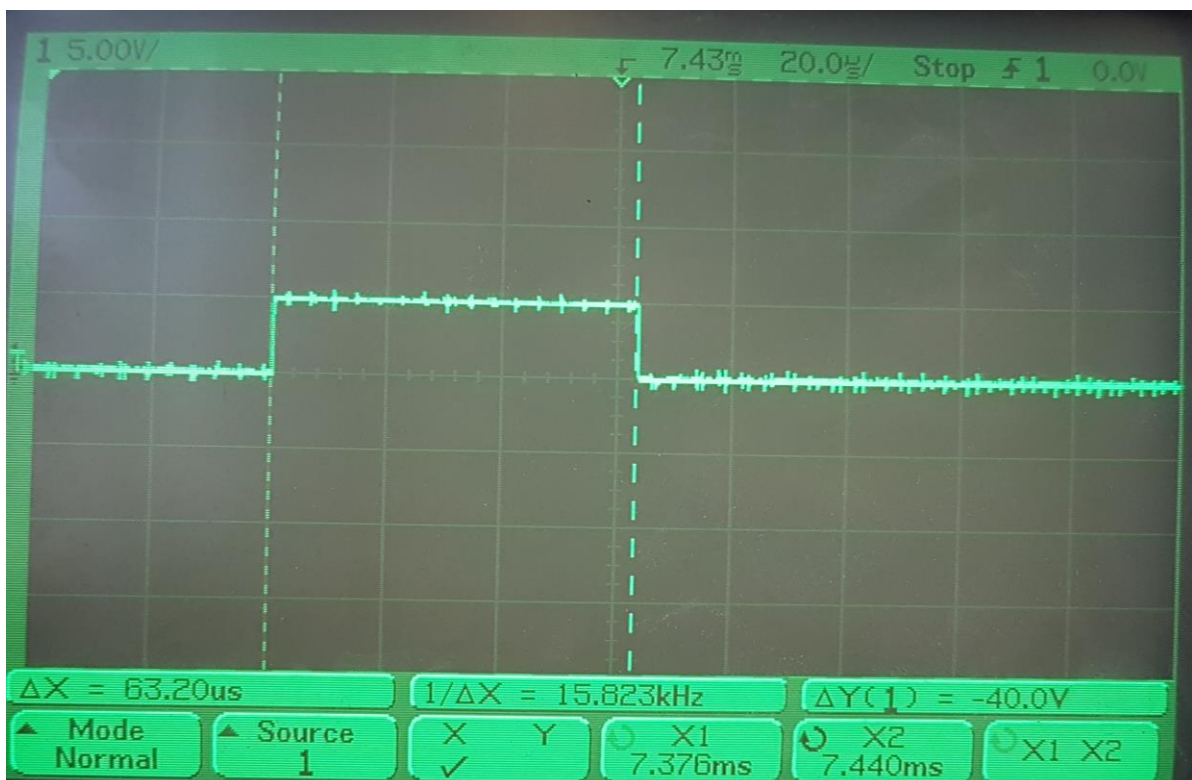
4.2. Captures de l'oscil·loscopi de les senyals HSync i VSync

HSync:

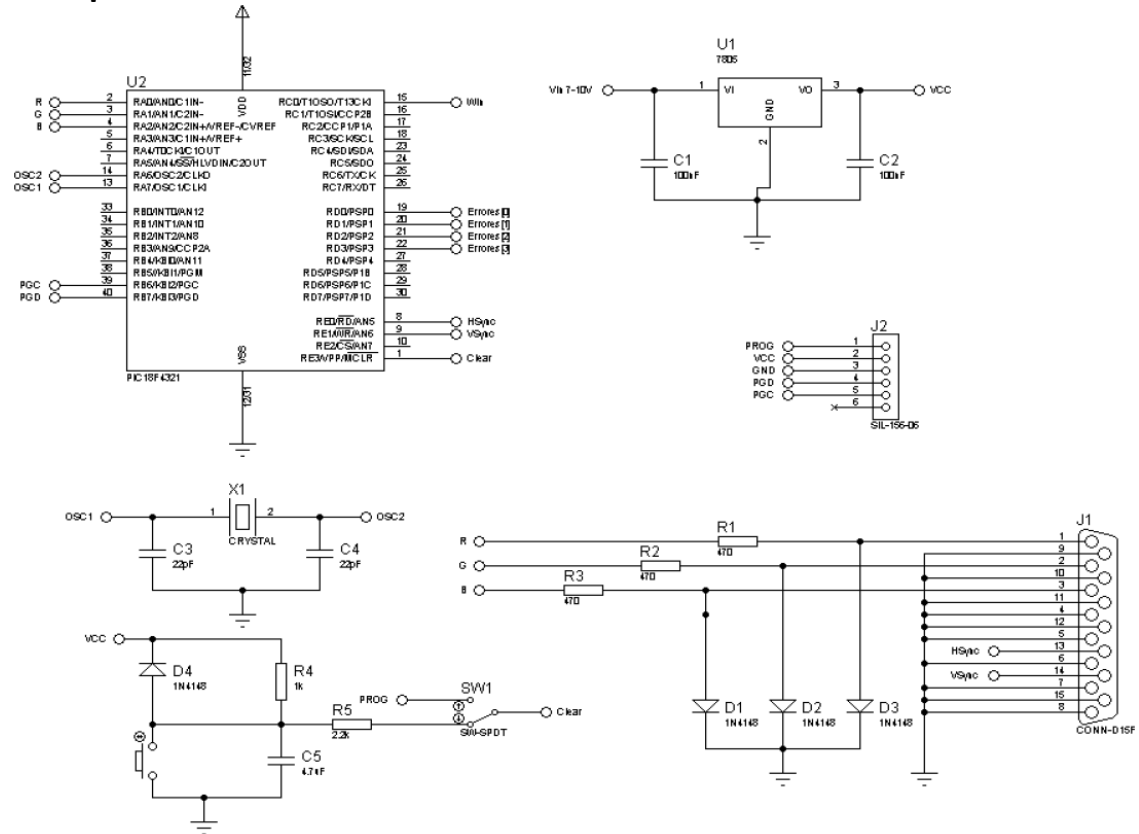




Vsync



5. Esquema elèctric



6. Problemes observats

A diferència de les dos fases anteriors per molt esquema elèctric que tinguéssim o simulació que tinguéssim el principal obstacle ha sigut els coneixements de Assembler, per el que els problemes s'han dividit en dos seccions clares, que serien Hardware i Software.

Hardware

Entenent per hardware tot el procés d'interpretació de l'esquema lliurat per l'assignatura i soldadura de la placa. Els problemes que hem tingut han sigut generats al tenir que tallar la placa, donat que vàrem soldar la fase 2 i fase 3 en la mateixa placa. Així doncs en el procés, vam fer mal bé el connector VGA. Donat que justament el pin que nosaltres connectàvem el vermell en el VGA, no donava continuïtat amb el connector. La solució va ser prou senzilla, vam tornar a comprovar les soldadures i vam poder-ho solucionar en pocs minuts. La complicació va ser més donada per el fet de que vam provar altres coses abans de realitzar la simple prova de continuïtat i per tant, vàrem perdre bastant temps per aquest petit error.

Software

Per la part de software, a diferència de Hardware, les complicacions i els problemes eren més relacionats per la falta de coneixement i habilitat en la programació amb Assembler.

El error que segurament era més comú en aquest projecte ha estat en la desincronització de HSync i VSync. Per poder definir correctament el temps a 1 i a 0 de cada senyal vàrem tenir que fer bastantes proves fins poder arribar a sincronitzar correctament.

Després de aconseguir sincronitzar-los, vam començar a la programació de cada línia o zona per mostrar el penjat.

L'error que més ens va perjudicar va ser en la no organització de les parts, es a dir, un cop programat el penjat i que es mostres correctament, no vam tenir en compte l'estat de la partida. Així doncs, el codi tenia que ser reestructurat per adaptar-se a l'estat de la partida i mostrar en cas de guanyar o perdre el color del penjat.

A part d'aquest error, durant la programació com que es basava sobretot en comptar la quantitats de NOPs que es tenia que afegir, vàrem perdre temps per acabar de definir be cada línia i el temps necessari perquè estigues sincronitzat.

7. Conclusions

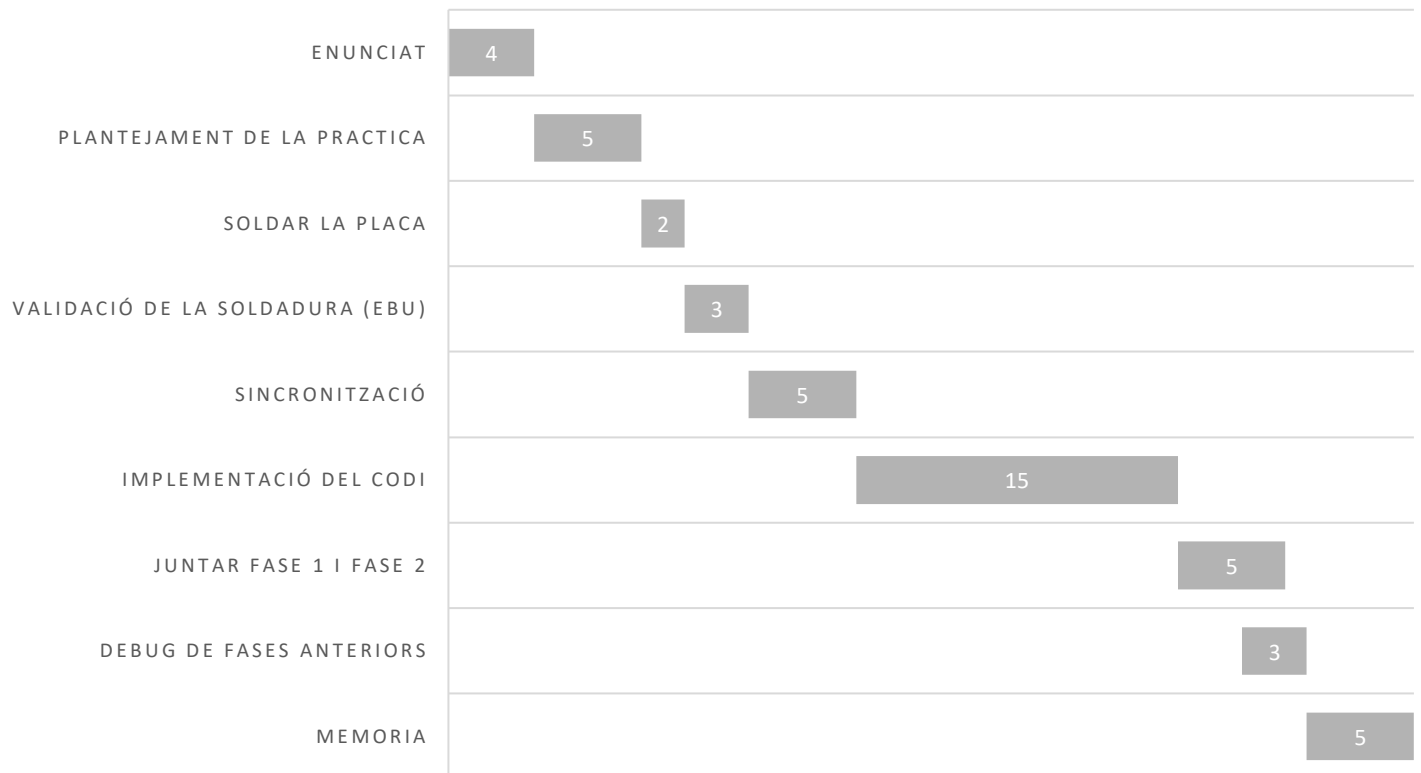
En el nostre cas considerem que un cop hem realitzat aquesta fase podem dir que tenim una bona base d'Assembler, degut a que amb els coneixements de classe només ens permet tenir una perspectiva del que hem de fer però ha sigut amb el prova i error del desenvolupament del codi de la practica on podem dir que hem après considerablement. Nosaltres hem percebut també la importància del us del datasheet de la Pic, ja que ens ha permès agafar agilitat al programar per aquesta pic concreta.

Un factor important que hem après amb aquesta practica ha sigut el tenir que adequar el codi als recursos que ens proporciona el microcontrolador, degut a que no sols utilitzem un llenguatge de molt baix nivell, directament a nivell d'instruccions, sinó que treballem amb un entorn on no comptem amb un sistema operatiu que ens ofereix una gestió dels recursos i certes facilitats, que tal com sens demana hem de ser conscients del que realitzem pas a pas, suposant així una dificultat afegida.

Per un altre costat, considerem que al desenvolupar la fase 3 ens ha donat una perspectiva real de com és utilitzar una pic amb diferents perifèrics o simplement el fet de poder connectar-ho a una pantalla i veure que podem encarregar-nos de generar contingut per el VGA a molt baix nivell.

Finalment ens ha agradat poder acabar veient com unint les tres fases hem pogut obtenir un joc complert. Ens produeix una gran satisfacció, després de superar tants errors desconeguts per nosaltres fins aleshores, veure com d'un teclat matriu hem acabat mostrant un penjat per pantalla amb un microcontrolador i ens adonem com amb l'experiència obtinguda amb aquesta practica podem abastar noves idees i projectes més diversos.

8. Planificació



9. Annex

Guanyem la partida

→ <https://youtu.be/eFqVQapf6LM>

Perdem la partida

→ <https://youtu.be/wPG5XRNauj0>