

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2016-2017

Pràctica 1

LSHangman

| Alumnes | Login | Nom |
|---------|-------|-----|
| | | |
| | | |

| Entrega | Placa | Memòria | Nota |
|---------|-------|---------|------|
| | | | |

| Data | |
|------|--|
|------|--|

Portada de la memòria

Introducció i consideracions generals

Des de fa temps, l'empresa LSDiversiones S.L. té al cap realitzar un conjunt de jocs per entretenir als seus clients mentre esperen a ser atesos. Arrel al gran èxit que van tenir els jocs en les edicions anteriors (Tic Tac Toe, LSSimon i LSMemory), els directius s'han tornat a posar en contacte amb nosaltres per a que ens fem càrrec de l'execució del projecte.

En aquest cas, el repte proposat és el LShangman. Imitant el clàssic model de joc del Penjat, l'usuari s'haurà d'enfrontar a les plaques per a endevinar una paraula concreta. Com ja sabeu, l'usuari guanyarà la partida si aconsegueix endevinar les lletres que la conformen en menys de 8 intents; altrament quedarà penjat.

Per simplificar la realització de la pràctica, la dividirem en 3 plaques diferenciades:

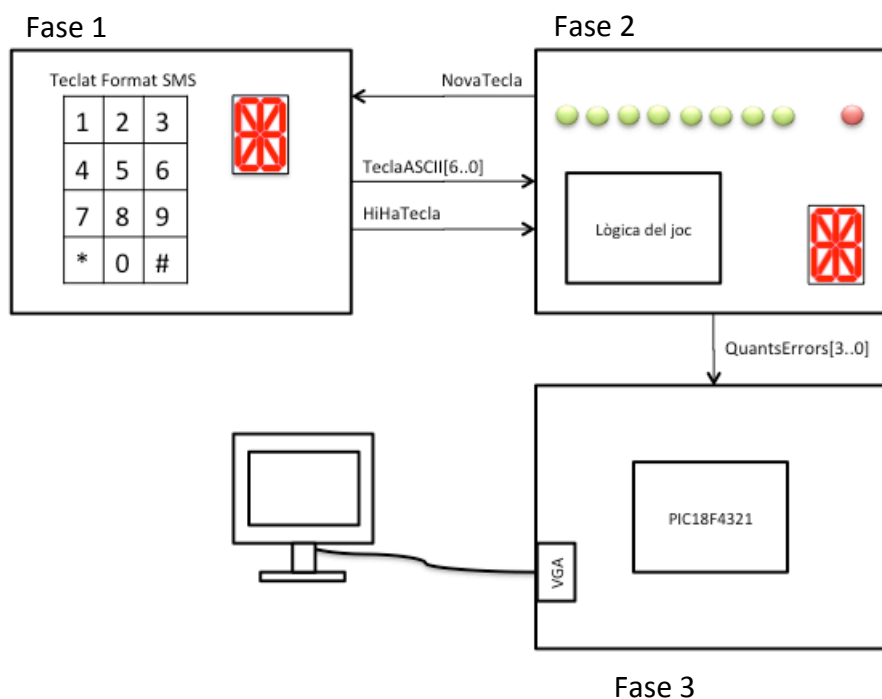


Figura 1. Diagrama de blocs general de la pràctica.

Fase 1: Interfície de connexió amb l'usuari. Aquesta placa ens servirà per introduir les lletres que l'usuari cregui que té la paraula a encertar. Ha de contenir un teclat de matriu de 3x4 i un *display* de 16 segments per mostrar les lletres que s'estiguin seleccionant.

Fase 2: Lògica del joc. Aquesta part de la pràctica serà l'encarregada tant de la generació d'una paraula aleatòria a endevinar com de mostrar el progrés de la partida.

Fase 3: Finalment, la Fase 3 mostrarà per una pantalla VGA el clàssic ninot del penjat. Concretament, fent servir un microcontrolador i un cable VGA, es mostrarà el nombre d'errors que es van cometent a la partida afegint parts al dibuix d'un ninot de Hangman.

La pràctica es realitzarà en tres fases, on cadascuna fa referència a una placa. A continuació es detallen algunes dates importants referents a la planificació d'aquesta pràctica.

| Setmana | Classe de teoria | Laboratori |
|-----------------|---|-------------------|
| 19 set - 23 set | Presentació pràctica 1 Explicació fase 1 | |
| 26 set - 30 Oct | | Fase 1 |
| 3 Oct - 7 Oct | | Fase 1 |
| 10 Oct -14 Oct | | Fase 1 |
| 17 Oct - 21 Oct | | Fase 1 |
| 23 Oct | | Lliurament fase 1 |
| 24 Oct - 28 Oct | Explicació fase 2 | Fase 2 |
| 31 Oct - 4 Nov | | Fase 2 |
| 7 Nov - 11 Nov | | Fase 2 |
| 14 Nov - 18 Nov | | Fase 2 |
| 21 Nov - 25 Nov | | Fase 2 |
| 28 Nov - 1 Des | | Fase 2 |
| 5 Des - 9 Des | Explicació fase 3 | Fase 2 |
| 11 Des | | Lliurament fase 2 |
| 12 Des - 16 Des | | Fase 3 |
| 19 Des - 23 Des | | Fase 3 |
| 26 Des - 30 Des | VACANCES NADAL | VACANCES NADAL |
| 2 Gen - 6 Gen | VACANCES NADAL | VACANCES NADAL |
| 8 Gen | | Lliurament fase 3 |

Taula 1. Calendari de la pràctica

Requeriments generals de les pràctiques

- És obligatori que tots els circuits integrats que s'utilitzin a la Fase 1 i 2 siguin de la família 74LS excepte les memòries RAM, EEPROM, GALs i oscil·lador.
- En cas de no poder localitzar algun circuit integrat de la família 74LS es pot utilitzar un que sigui equivalent d'alguna altra família prèvia autorització per part dels monitors de pràctiques. Aquest fet haurà de quedar reflectit a la memòria de la pràctica.
- El model de GAL a utilitzar és la 22V10 o 22V8.
- El model d'EEPROM a utilitzar és la 27C256 o equivalent, i de memòria RAM, la 62256 o equivalent. L'oscil·lador a utilitzar en les diverses plaques és un NE555 funcionant en mode astable.

Dates de lliurament

Les pràctiques s'hauran de pujar al pou que s'obrirà a l'estudy d'acord amb el format correcte per cada fase. Aquest pou es tancarà el diumenge abans de començar la setmana d'entregues a les 23:55. Un cop es tanca el pou no es podran realitzar modificacions d'aquest i els arxius de les GALs, EEPROMs i microcontrolador que contingui el pou són els que s'hauran d'utilitzar en el moment de l'entrevista.

Fase 1: La interfície d'usuari

Aquesta primera fase s'encarrega de la interacció amb l'usuari. És a dir, des d'aquesta placa es seleccionarà la lletra que es vol enviar cap a la Fase 2.

Els senyals que ha de contenir són:

- *HiHaTecla*: Senyal de sortida que indica que s'ha seleccionat una tecla. S'ha d'activar quan han passat 2 segons des de l'última vegada que s'ha pulsat una tecla.
- *TeclaASCII[6..0]*: Senyal de sortida que contindrà el codi ASCII de la lletra seleccionada en majúscules. Com que només farem ús del codi ASCII simplificat, només necessitem els 7 bits de menys pes del codi de la lletra.
- *NovaTecla*: senyal d'entrada que indica que la Fase 2 està preparada per rebre una nova tecla.

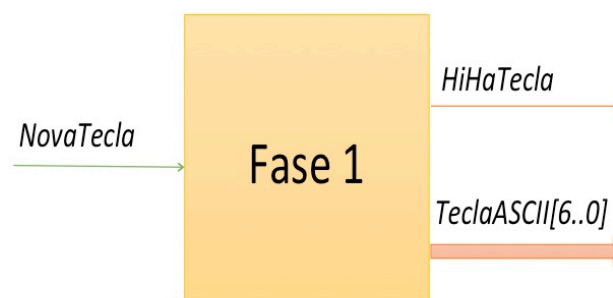


Figura 2. Diagrama d'entrades i sortides de Fase 1

Aquesta fase haurà de tenir un teclat de matriu de 4 files i 3 columnes que permetrà seleccionar la lletra desitjada.

Per passar de les 12 tecles del teclat matriu a les 26 lletres de l'abecedari, caldrà codificar les lletres de la mateixa manera que fan els telèfons mòbils per permetre enviar missatges SMS. Així, s'utilitzarà la següent distribució:

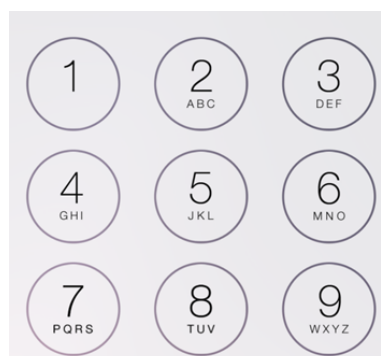


Figura 3. Teclat SMS

El funcionament de la placa serà el següent: cada cop que es prem una tecla, aquesta s'haurà de mostrar pel *display* de 16 segments (farem ús del caràcter ASCII en majúscules de la lletra). Si es prem dues vegades la mateixa tecla, haurà d'aparèixer la segona lletra corresponent a aquella tecla (tal i com s'indica a la Figura 3). Per exemple: si es polsa dues vegades consecutives la tecla "2", haurà de mostrar-se pel *display* la lletra "A" a la primera pulsació i la lletra "B" a la segona pulsació, ja que la seqüència completa de la tecla 2 és A, B, C, A, B, C.

Cal tenir en compte que si primer es polsa la tecla "2" dues vegades i després la "3", haurà de mostrar-se pel *display* la seqüència de lletres "A", "B", "D".

Tindrem 2 segons des que hem seleccionat l'última lletra per decidir si és la definitiva que volem enviar cap a la Fase 2; passat aquest temps s'activarà el senyal *HiHaTecla* i es reiniciarà la seqüència de lletra. És a dir, si seleccionem la tecla "B" i passen 2 segons, quan s'hagi activat *NovaTecla* i tornem a polsar la tecla "2" s'haurà de mostrar una "A" pel *display* de 16 segments.

Per tant, podem resumir les accions que ha de fer aquesta fase com:

1. Esperar que la Fase 2 estigui preparada per a processar una nova tecla.
2. Esperar que l'usuari premi una lletra.
3. Quan el jugador premi una tecla cal refrescar el 16 segments.
4. Si no han passat 2 segons, tornar al punt 2.
5. Si han passat 2 segons, preparar el senyal *QuinaTecla*[6..0] i activar el senyal de *HiHaTecla*.
6. Tornar al punt 1.

Ho podem veure il·lustrat a la Figura 4.

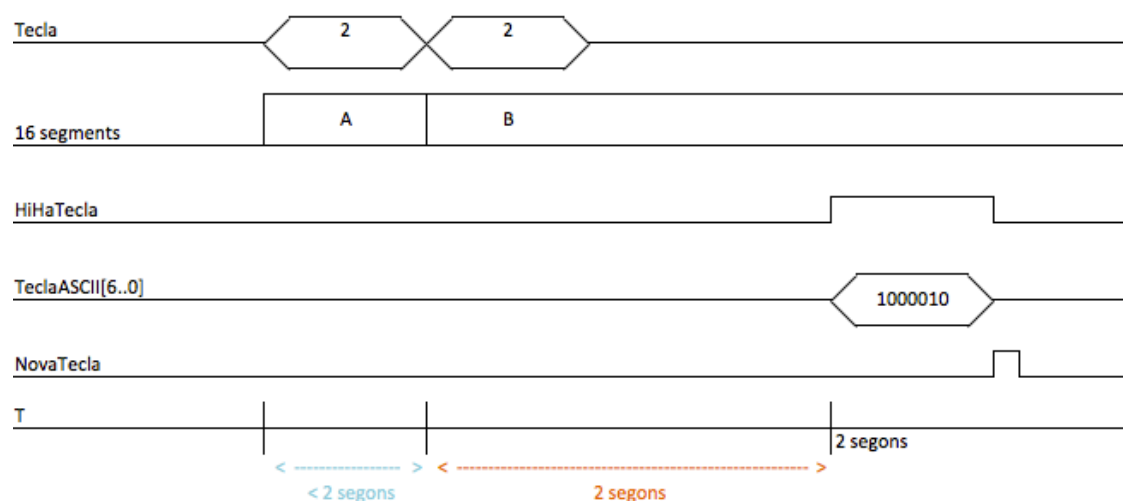


Figura 4. Diagrama de temps de la Fase 1.

Consideracions importants

- Per a la realització d'aquesta fase s'haurà de fer servir com a mínim una EEPROM.
- Els senyals de rellotge de la placa es generaran amb un NE-555. Aquest *clock* tindrà un *Duty Cycle* el més proper possible al 50% i una freqüència d'1KHz. Aquest *clock* tindrà sentit local. És a dir, no es podrà enviar cap a altres plaques.
- Al lliurament d'aquesta fase cal entregar un fitxer .rar que contingui una memòria que compleixi la normativa de pràctiques i dues fotografies de la placa, una des de sobre i una des de sota. La memòria constarà dels punts esmentats a la normativa i com a portada cal usar la portada corresponent a aquesta fase que podeu trobar a l'*estudy*. Caldrà incloure en una carpeta els fitxers *.PLD de la/es GAL/s (en el cas que s'utilitzin) i els *.hex de les EEPROM. El nom del fitxer .rar ha de seguir el següent format:
 - YYYY.MM.DD-SDM-1617-P1-F1-login1-login2.rar
 - Ex: 2016.12.03-SDM-1617-P1-F1-ls12345-ls54321.rar

Fase 2: La lògica del joc

Aquesta Fase serà l'encarregada de dur a terme el joc del penjat. Se seleccionarà una paraula prèviament emmagatzemada a una memòria de la placa de forma aleatòria i l'usuari l'haurà d'endevinar.

Disposarà de 8 intents per a endevinar la paraula, altrament haurà perdut.

Per la visualització de la partida aquesta fase disposarà de 8 LEDs que indicaran el nombre i la posició d'aquelles lletres que encara no s'han endevinat. També hi haurà un *display* de 16 segments que anirà mostrant lletra a lletra el progrés de la paraula.

Els senyals d'entrada i sortida d'aquesta fase són els següents:

- *HiHaTecla*: Senyal d'entrada que s'envia des de la Fase 1. Ens avisarà que l'usuari ha seleccionat una tecla.
- *TeclaAscii[6..0]*: Senyal d'entrada que s'envia des de la Fase 1. Ens indica el valor ASCII de la tecla que ha premut l'usuari.
- *NovaTecla*: Senyal de sortida que s'envia a la Fase 1 i indica que la Fase 2 està preparada per rebre una nova tecla.
- *NumErrors[3..0]*: Senyal de sortida que s'envia cap a la Fase 3 i indica el nombre d'errors que s'han comès fins al moment.
- *GameOver*: Senyal de sortida que s'envia cap a la Fase 3. Indica que s'ha acabat la partida perquè l'usuari ha comès 8 errors.
- *PartidaGuanyada*: Senyal de sortida que s'envia cap a la Fase 3 i indica que s'ha acabat la partida perquè l'usuari ha endevinat la paraula.

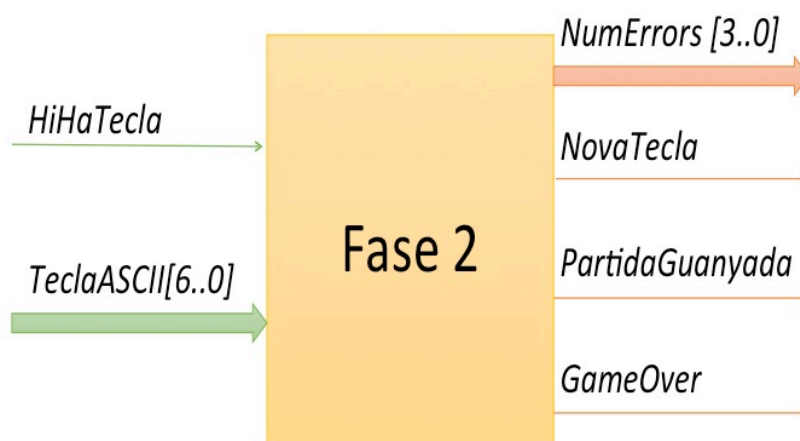


Figura 5. Diagrama d'entrades i sortides de la Fase 2.

La Fase 2 està dividida en 3 parts:

1. Selecció d'una paraula aleatòria: La placa contindrà una memòria EEPROM plena de paraules de màxim 8 caràcters. Ha d'haver-hi un mínim de 256 paraules emmagatzemades abans de començar a jugar (poden haver-hi algunes paraules repetides).
2. Zona de LEDs: A la placa hi haurà 8 LEDs que indicaran quines posicions de lletra ha endevinat l'usuari. Un LED encès indicarà que l'usuari encara no ha endevinat la lletra d'aquella posició. Inicialment els LEDs encesos indicaran la mida de la paraula. Per exemple, si la paraula a endevinar és "HOLA", si l'usuari ha endevinat només la lletra "O" tindrem la següent seqüència de LEDs:
"ENCÈS, APAGAT, ENCÈS, ENCÈS, APAGAT, APAGAT, APAGAT, APAGAT".



Figura 6. LEDs

3. Zona de *display*: Aquesta fase contindrà un *display* de 16 segments pel qual s'aniran mostrant una per una les lletres que conformen la paraula. Aquelles lletres que ja hagin sigut encertades es mostraran en majúscules, mentre que aquelles que encara no s'hagin encertat es mostraran com a "_".

Hi haurà d'haver un indicador que representi el final de la paraula. Aquest indicador pot ser: deixar tot el *display* apagat, encendre el punt, encendre tot el *display*... i haurà de durar un segon.

Per tant, en l'exemple anterior, s'hauria d'anar mostrant pel *display* "_", "O", "_", "_" en bucle fins que s'encerti una nova lletra. El temps de cada lletra és lliure sempre i quan l'usuari el pugui llegir bé (1 segon aproximadament).

Podem veure aquest exemple il·lustrat a la Figura 7.

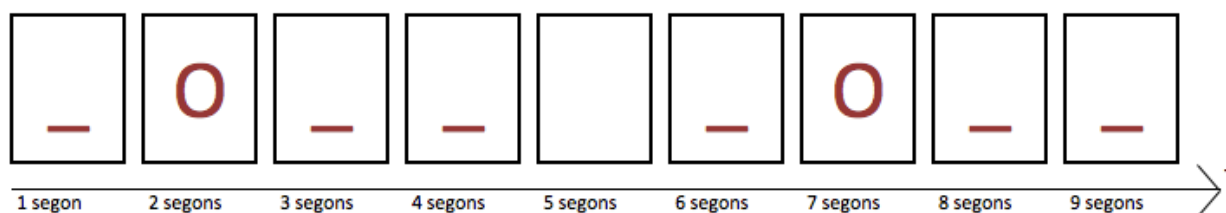


Figura 7. Exemple del que s'ha de mostrar pel display.

També es disposarà d'un LED de GameOver indicant que s'ha acabat la partida perquè s'han comès 8 errors. Per tornar a jugar estarem a l'espera que l'usuari reiniciï la placa manualment mitjançant un polsador.

De la mateixa manera, si s'han endevinat totes les lletres, s'esperarà que l'usuari reiniciï la placa i no es podrà seguir jugant (si es premen més tecles al teclat matriu, s'ignoraran i no es mostraran pel *display* de la Fase 1).

Així doncs, la seqüència d'accions que haurà de generar aquesta placa serà la següent:

1. Selecció d'una paraula aleatòria.
2. Encendre els LEDs corresponents al nombre de lletres de la paraula.
3. Activar el senyal *NovaTecla* quan la placa estigui preparada per processar una nova lletra.
4. Comparar la lletra amb totes les de la paraula seleccionada al punt 1.
5. Si la lletra hi és una o més vegades, apagar els LEDs de les posicions pertinents. Altrament, sumar un error per a enviar a la Fase 3.
6. Si el nombre d'errors és igual a 8, s'encendrà el LED de *GameOver*, s'activarà el senyal de *GameOver* i s'esperarà a que l'usuari reiniciï la placa.
7. Si el nombre d'errors és més petit que 8 però ja no queden lletres per endevinar, tots els LEDs estaran apagats, s'activarà el senyal de *PartidaGuanyada* i s'esperarà a que l'usuari reiniciï la placa.

Cal recordar que, paral·lelament a totes aquestes accions, el *display* de 16 segments sempre estarà mostrant contínuament els caràcters de la paraula.

Consideracions importants

- S'ha d'utilitzar la placa realitzada a la fase anterior.
- Els senyals de rellotge s'hauran d'implementar amb un únic NEE-555 seguint les mateixes especificacions que en la fase anterior.
- Pel que fa a memòries RAM i EEPROM, només es podran fer servir una memòria RAM 62256 i una memòria EEPROM 27C256 per resoldre aquesta segona fase.
- Cal lliurar les dues plaques corresponents a la Fase 1 i a la Fase 2.
- Al lliurament d'aquesta fase cal entregar un fitxer .rar que contingui una memòria que compleixi la normativa de pràctiques i dues fotografies per cada placa, una des de sobre i una des de sota. La memòria constarà dels punts esmentats a la normativa i com a portada cal usar la portada corresponent a aquesta fase que podeu trobar a l'*estudy*. Caldrà incloure en una carpeta els fitxers *.PLD de les GAL i els *.hex de les EEPROM. El nom del fitxer .rar ha de seguir el següent format:
 - YYYY.MM.DD-SDM-1617-P1-F2-login1-login2.rar
 - Ex: 2016.12.03-SDM-1617-P1-F2-ls12345-ls54321.rar

Fase 3: El generador de senyal VGA

Els directius de LSDiversiones han quedat molt contents amb el resultat obtingut fins al moment i han decidit que volen televisar les partides. Concretament volen col·locar una pantalla a la sala d'espera en la qual es mostrarà l'estat de la partida. Aquesta pantalla es connectarà al sistema que hem dissenyat mitjançant un connector VGA i una placa amb microcontrolador.

L'encarregat de realitzar aquesta tasca serà un PIC18F4321 funcionant a 40 MHz i programat en llenguatge *assembler*.

La pantalla mostrarà el clàssic ninot del penjat corresponent al nombre d'errors que s'han comès fins al moment. A més, també s'haurà de pintar el nombre de grup que se us va assignar en formalitzar el grup de pràctiques.

Durant la partida, el ninot serà de color blanc. Si s'acaba el joc perquè s'han comès els 8 errors (s'activa el senyal de *GameOver*), el ninot s'haurà de pintar de color vermell. Per altra banda, si el joc s'acaba perquè ja s'ha endevinat la paraula (s'activa el senyal de *PartidaGuanyada*), el ninot es pintarà de color verd.

El número de grup pot ser de qualsevol color i haurà de romandre estàtic durant tota la partida independentment del nombre d'errors.

Les entrades i sortides d'aquesta fase seran les següents:

- *NumErrors[3..0]*: Aquest senyal d'entrada ens dirà el nombre d'errors que ha comès l'usuari fins al moment. Per cada error afegirem una part al ninot del penjat.
- *GameOver*: Senyal d'entrada que ens indica que el joc ha acabat perquè l'usuari ha arribat al màxim nombre d'errors possibles (8). Quan aquest senyal estigui actiu, el ninot haurà de pintar-se de color vermell.
- *PartidaGuanyada*: Senyal d'entrada que ens indicarà que s'ha acabat la partida perquè l'usuari ha endevinat totes les lletres.
- *R, G, B*: Senyals de sortida que s'enviaran cap a la pantalla per tal de pintar cada píxel amb el color desitjat.
- *HSync* i *Vsync*: Senyals de sortida que s'enviaran cap a la pantalla. Són els senyals típics de sincronisme que necessita una pantalla analògica per funcionar. El seu funcionament està explicat més endavant dins d'aquesta memòria (Annex: Senyal de vídeo VGA).

Així, caldrà implementar un sistema com el de la Figura 8. Noteu que els senyals que s'envien de la Placa 2 a la Placa 3 per a que això sigui possible han de ser estrictament els necessaris (*NumErrors[3..0]*, *GameOver* i *PartidaGuanyada*).

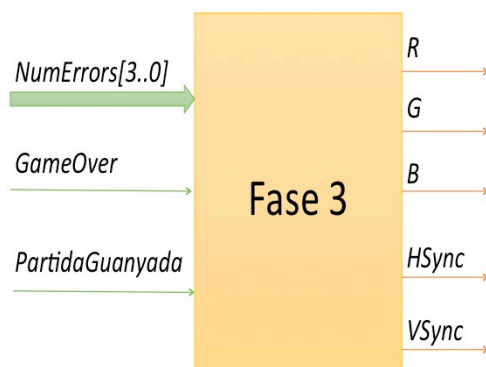


Figura 8. Entrades i sortides de la Fase 3.

La manera de representar la partida a la pantalla ha de complir els requisits explicats anteriorment. A les figures 9 i 10 es proposa un exemple de representació:

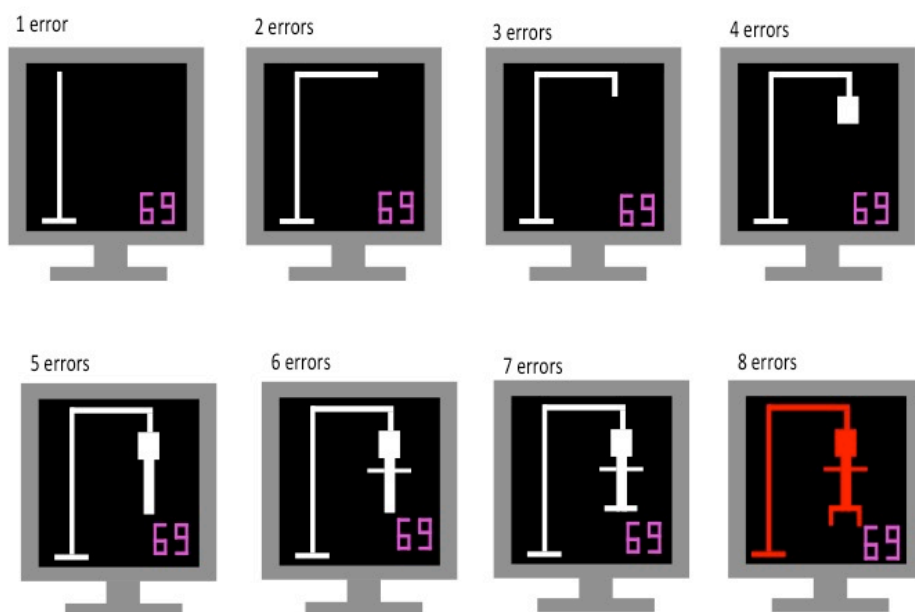


Figura 9. LSHangman orientatiu.

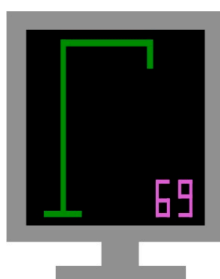


Figura 10. Partida on hi ha hagut 3 errors i s'ha endevinat la paraula.

A més, també disposeu d'un esquema elèctric bàsic que il·lustra com connectar el PIC, el circuit de *reset*, el circuit de programació i el connector VGA.

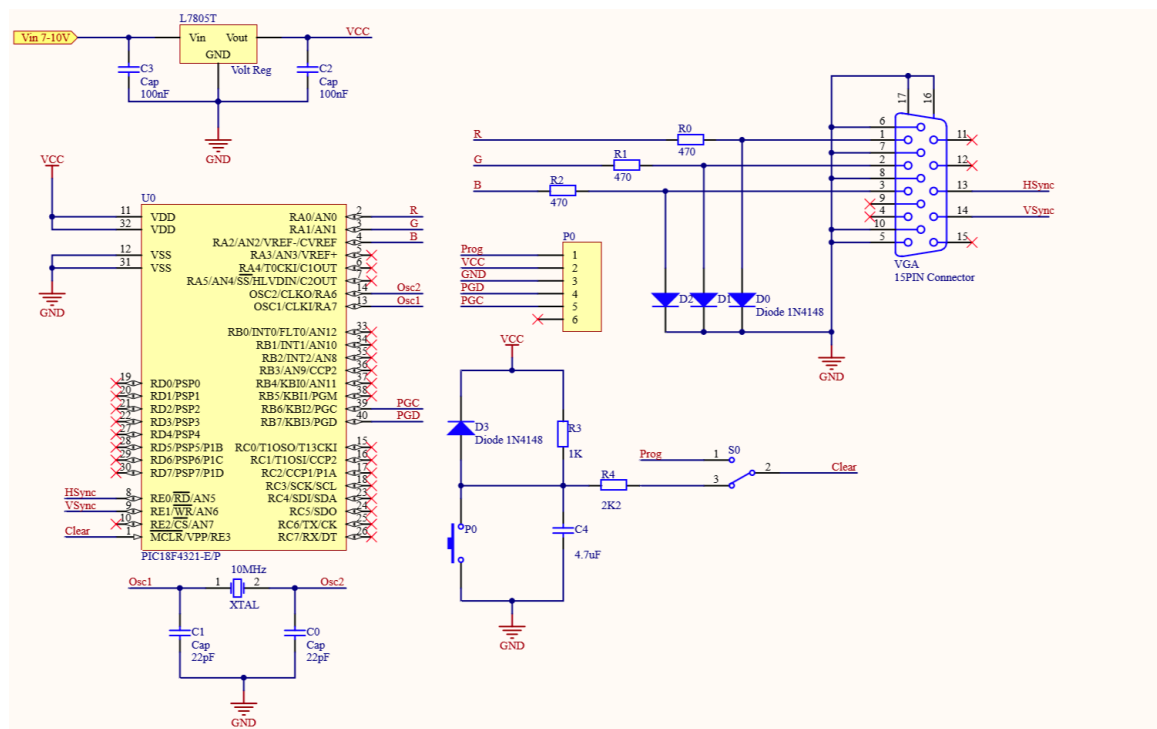


Figura 11. Proposta d'esquema elèctric per la Fase 3.

Consideracions importants

- S'han de lliurar les tres plaques.
- Al lliurament d'aquesta fase cal entregar un fitxer .rar que contingui una memòria que compleixi la normativa de pràctiques i dues fotografies per cada placa, una des de sobre i una des de sota. La memòria constarà dels punts esmentats a la normativa i com a portada cal usar la portada corresponent a aquesta fase que podeu trobar a l'estudy. Hi ha d'haver una carpeta amb el fitxer d'assembler i el fitxer compilat. El nom del fitxer .rar ha de seguir el següent format:
 - YYYY.MM.DD-SDM-1617-P1-F3-login1-login2.rar
 - Ex: 2016.12.03-SDM-1617-P1-F3-ls12345-ls54321.rar

Annex: Senyal de vídeo VGA

El port VGA s'ha utilitzat des de fa molts anys als ordinadors per tal de comunicar-se amb un monitor. El senyal VGA suporta múltiples resolucions i *frame rates* (fotogrames per segon). En el nostre cas treballarem a la resolució més bàsica: 640x480 i un *frame rate* de 60 Hz. El senyal VGA està format per 5 senyals de control: 3 components RGB (*Red, Green, Blue*) que s'envien per separat, 1 senyal de sincronisme vertical (*VSync*) i 1 senyal de sincronisme horitzontal (*HSync*). Més endavant s'explicarà el funcionament de cadascun d'aquests senyals.

Un altre factor important és la precisió de color, mesurada en bits. Com ja s'ha dit el senyal VGA és analògic. Les pantalles actuals suporten precisions superiors a 8bits per canal RGB donant una combinació de $(2^8)^3 \approx 16$ milions de colors. En el nostre esquema no tenim cap DAC de 8 bits sinó que disposem d'una precisió d'un bit per color. És a dir, cadascun dels 3 colors RGB es mostrarà saturat al màxim o bé apagat. Per tant, disposarem de 8 possibles colors: Negre (R=G=B=0), Vermell (R=1,G=B=0), Verd (R=0,G=1,B=0), Blau (R=G=0,B=1), Cyan, Magenta, Groc, i Blanc. Els quals són el resultat de mesclar els 3 colors primaris amb màxima saturació.

Com que els nivells de tensió amb els que treballa el senyal de vídeo VGA per a les línies RGB són de 0 a 0.7V, utilitzarem un díode que ens fixarà aquesta tensió entre els seus extrems, i una resistència en sèrie per limitar el corrent de sortida, però que sigui suficient per polaritzar el díode. D'altra banda, els senyals de VSync i HSync són TTL (5V) així que no caldrà adaptar-los.

El senyal VGA

És important tenir present que no tot el senyal que s'envia pel connector VGA és visible. Hi ha unes parts invisibles anomenades pòrtic vertical i pòrtic horitzontal. Les antigues pantalles CRT (tub de raigs catòdics) necessitaven un temps per moure el canó d'electrons del final de la línia fins al principi de la següent, i des del final de la pantalla fins al principi. Tot i que amb les pantalles digitals LCD d'avui en dia això no faria falta, el senyal VGA segueix mantenint els pòrtics i els polsos de sincronització tan llargs per mantenir la compatibilitat cap enrere.

A la figura que es mostra a continuació s'observa l'estructura del senyal VGA. Entendre aquesta figura és clau per poder generar adequadament el senyal i entendre el funcionament. El senyal es comença a generar per la part superior esquerra i es va fent un escombrat en sentit horitzontal. Al final de cada línia se salta a la següent per la part esquerra, i quan s'arriba al final (part inferior dreta) es torna a l'inici per pintar el següent fotograma.

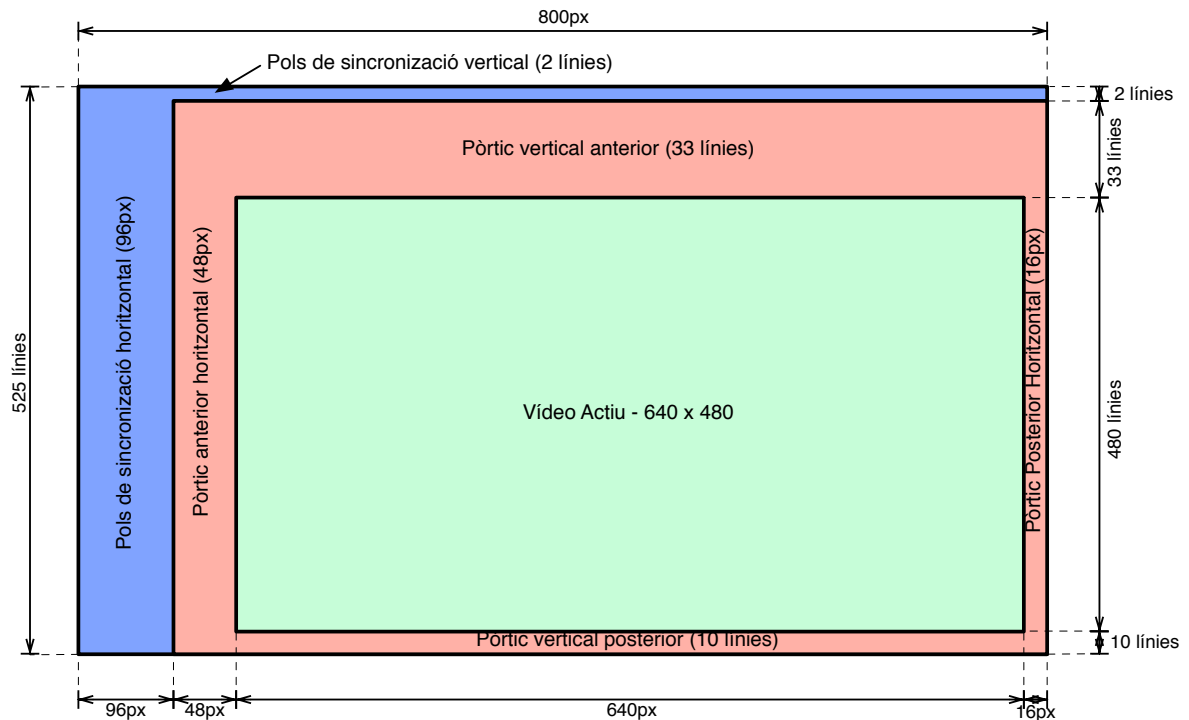


Figura 12: Diagrama del senyal VGA per un fotograma.

Sincronització vertical – VSync

El pols de sincronització vertical és necessari generar-lo quan s'acaba de pintar el fotograma sencer. Com que treballem a un *frame rate* de 60Hz, cada 1/60s s'haurà d'enviar un pols de sincronització vertical. La durada d'aquest pols serà l'equivalent a dues línies.

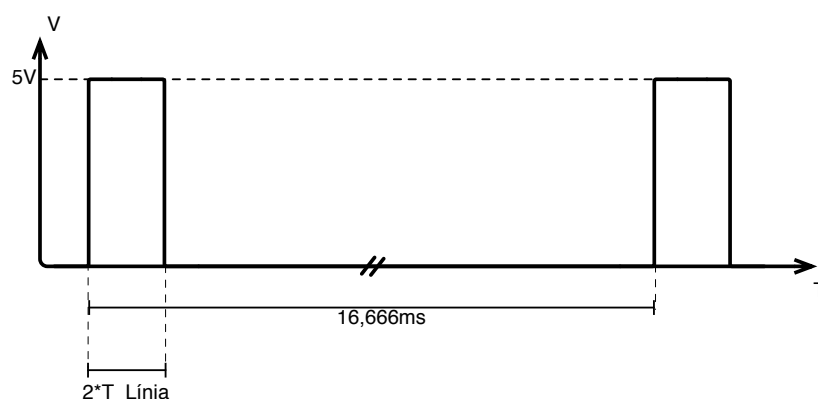


Figura 13: Senyal de sincronisme vertical.

Sincronització horitzontal - HSync

Els polsos de sincronització horitzontals indiquen al monitor quan comença una nova línia. L'amplada d'aquest pols serà l'equivalent al temps de 96 píxels.

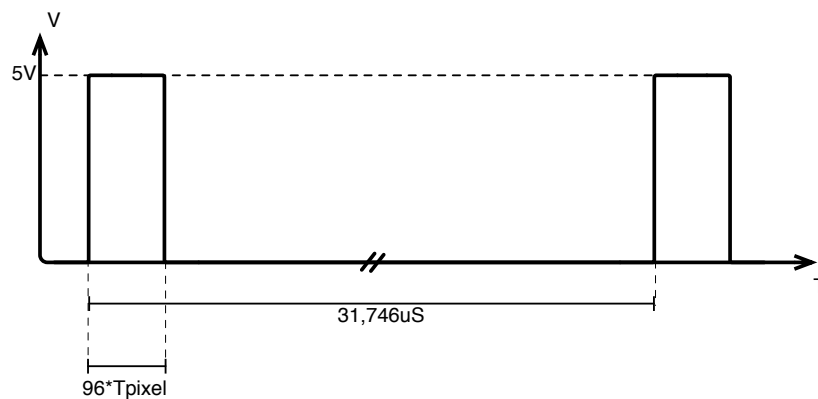


Figura 14: Senyal de sincronisme horitzontal.

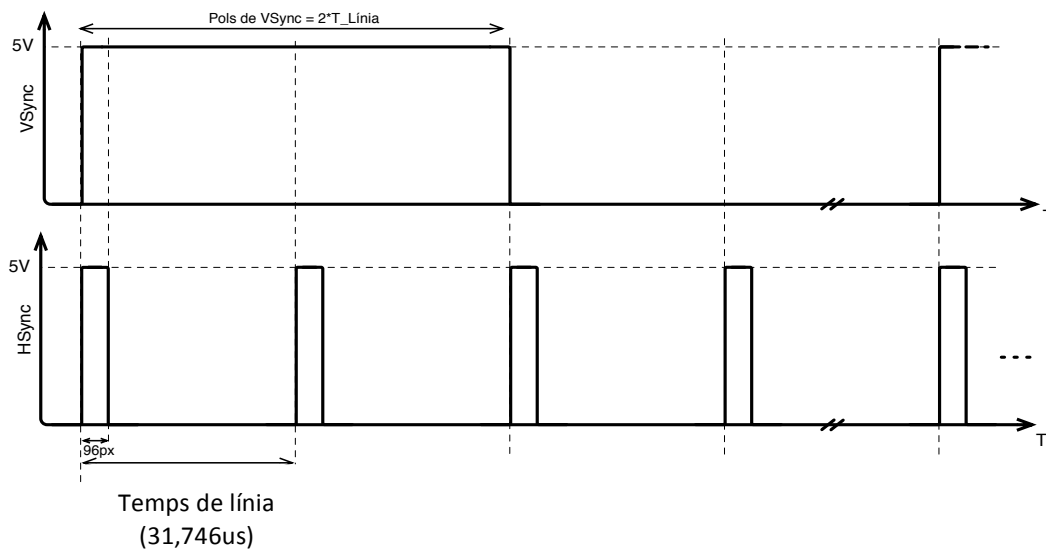


Figura 15: Pols de sincronització vertical + Polsos de sincronització horitzontal

Com s'observa a la figura 6, mentre s'envia el pols de sincronització vertical, se segueixen enviant els polsos de sincronització horitzontal tal i com mostra la figura 5. Tot i no haver-se representat a la figura, durant aquest espai de temps els senyals RGB haurien d'estar en repòs.

Pòrtics

És un espai de temps que necessita el monitor abans de començar a mostrar informació. Durant aquest espai de temps el senyal de les components RGB es mantindrà a 0. Noteu que els pòrtics s'han d'ajustar a l'esquema de la Figura 3. Així, realitzar els pòrtics únicament implica deixar en repòs les línies RGB. No s'ha de generar cap pols amb l'HSync ni VSync.

Vídeo Actiu

Aquesta zona és on es transmet la informació que volem representar per la pantalla línia a línia. Aquesta informació s'envia a través de les components RGB segons la mescla que s'ha esmentat anteriorment. Un '1' a la component significa que està activa. Un '0' significa que està apagada. A la següent figura es mostra un *timing* de com realitzar les barres EBU amb 100% de saturació.

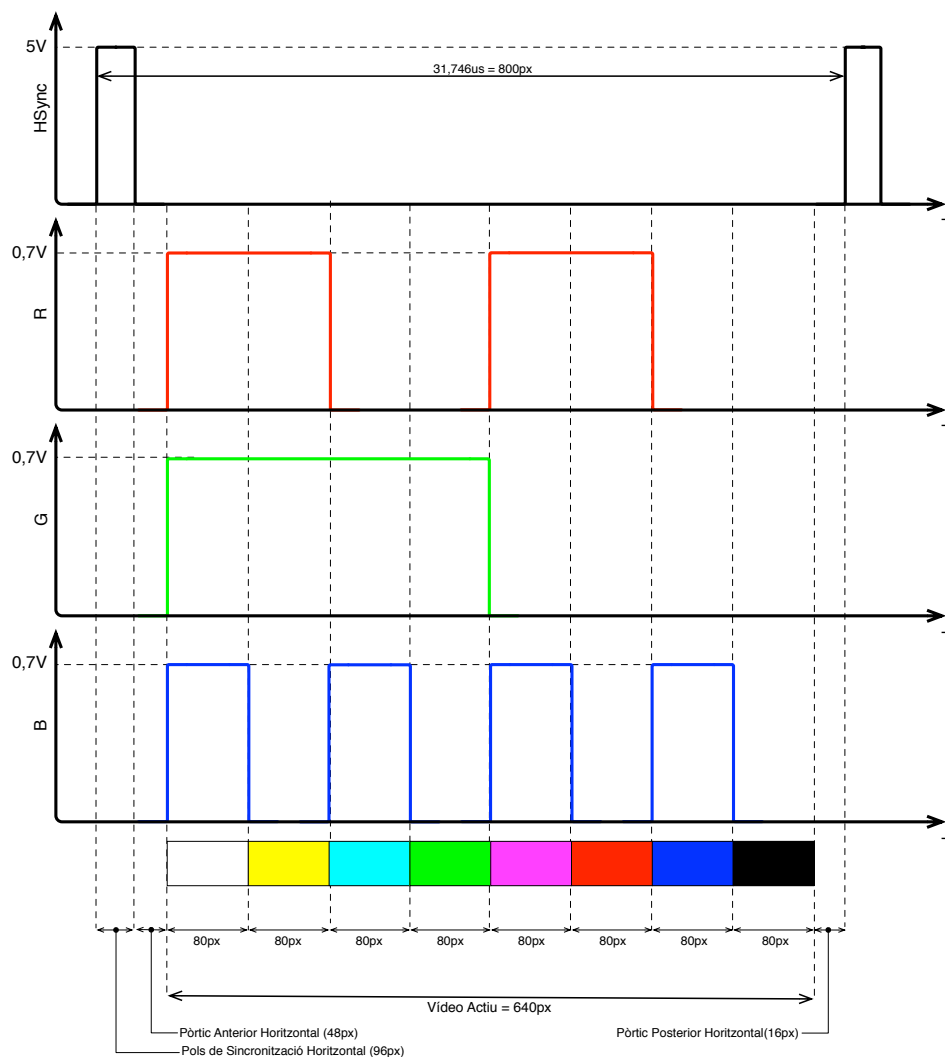


Figura 16: Diagrama de temps de la zona de vídeo actiu per mostrar les barres EBU.

Càlcul del temps

A continuació es detalla com realitzar la conversió de píxels i línies a instruccions. Primer de tot cal tenir en compte que el PIC treballa a 40MHz, i per tant executa 10 milions d'instruccions (d'1 cicle) per segon. Per tant, el temps d'instrucció és de 100ns.

El següent pas és calcular el temps de píxel. Sabem que la pantalla s'ha de refrescar 60 vegades per segon (60 Hz) i que disposem d'una àrea de 800x525 píxels per pintar (s'han de tenir en compte els actius i els no actius!!). Això ens dona que hem de pintar 25,2Mpíx/s. Per tant el temps de píxel és d'aproximadament 39,68ns.

$$F_{Max} = f_{Actualització} \cdot Res_H \cdot Res_V = 60 \cdot 800 \cdot 525 = 25,2MHz$$

$$T_{Pixel} = \frac{1}{F_{Max}} = 39,6825397ns$$

Amb aquesta informació ens adonem que una sola instrucció equival a uns 2,5 píxels aproximadament. Això té conseqüències en la resolució horitzontal màxima que podem aconseguir. També pot ser útil conèixer el temps que dura una línia. Aquest simplement serà 800 vegades el temps de píxel.

$$T_{Línia} = T_{Pixel} \cdot Res_H = 39,6825397ns \cdot 800px = 31,746\mu s$$

Noteu que a l'hora de generar els senyals de sincronisme i els pòrtics s'induirà un error degut als decimals. El vostre software ha de ser prou hàbil com per a què aquest error es noti el mínim possible a l'hora de pintar la pantalla. Evidentment, l'oscil·loscopi és una eina imprescindible per sincronitzar els temps.

Una mica d'història...

Normalment, el tractament de gràfics es realitza amb processadors i un hardware específic anomenat RAMDAC que a grans trets no és més que una memòria RAM acoblada a un DAC que va traient la informació a 25,2MHz, pinta cada píxel, i genera els senyals de sincronisme de forma "automàtica". Llavors des d'una CPU s'accedeix a aquesta memòria i s'assigna un valor a cada píxel. La velocitat de la CPU és independent de la resolució de pantalla i aquesta pot fer altres tasques a part de preocupar-se de gestionar la pantalla. Si la CPU és lenta, només donarà la sensació que la imatge es refresca d'una manera més lenta. Aquesta ve a ser la idea del funcionament més bàsic de les targetes gràfiques dels ordinadors.

Com Generar el Senyal VGA

En aquest punt es disposa dels coneixements suficients per adonar-se de la problemàtica de generar el senyal VGA amb un microprocessador que no disposa de hardware específic per a tal tasca. La primera limitació que assalta a la vista és que el temps que disposa el microcontrolador per decidir quines sortides ha d'activar és bastant reduït (per cada instrucció es perden 2.5 píxels!). Així, amb els càlculs anteriors un s'adona fàcilment que la resolució horitzontal de 800px (640 visibles) no s'aconseguirà.

Es sacrifica resolució horitzontal degut a que és l'únic element que podem modificar de l'equació: el *frame rate* ve fixat i la pantalla s'espera el sincronisme vertical a 60Hz, el nombre de línies tampoc es pot variar doncs si no enviem el nombre correcte de línies el monitor mostrarà coses errònies o simplement no reconeixerà el senyal. Per tant l'únic element que podem tocar és el nombre de píxels horitzontals. Si es fa durar més estona el valor del píxel, aquest simplement es mostrarà més 'llarg'.

Una recomanació a seguir és realitzar els sincronismes dins d'una RSI que s'activi per un *timer* a cada final de línia i en funció de a quina línia us trobeu, genereu un senyal o un altre. És molt important que aquesta RSI sempre duri el mateix independentment del camí que prengui el codi. També disposar d'un 'main' dins el qual genereu tot el referent al vídeo actiu (visible) sense preocupar-se de quan s'han de generar els senyals i simplement vigilant que no es pinti més estona del compte en una mateixa línia. Noteu que aquesta estratègia no és única i no esteu obligats a implementar-ho d'aquesta manera.

Cal ser molt curós amb els senyals HSync i VSync. Cal que la ubicació i la duració d'aquests senyals sigui correcte. Si en algun moment del codi aquest senyal pateix un retard o la RSI dura més del compte, observareu l'efecte d'aquest retard en forma de desplaçament horitzontal dels píxels de la pantalla.