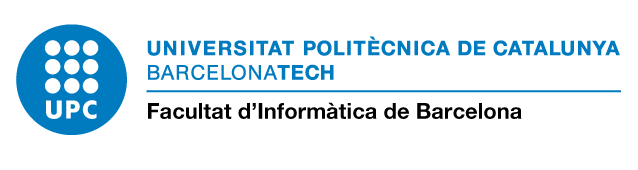
****

Enunciat de la pràctica de laboratori

**Lab 3:**

**Display 7 segments**

**L3. Display 7 segments**

**L3 (A) Multiplexat en el temps del display de 7 segments**

**L3 (B) Interrupcions generades per botons**

**1- Objectius**

L’objectiu d’aquesta pràctica és controlar un grup de 4 displays 7-segments de manera que el microcontrolador pugui presentar informació a través de les seves sortides.

Durant la primera part de la pràctica L3(A), aprendrem a gestionar botons d’entrada mitjançant el mètode d’enquesta i anirem seleccionant cada un dels 4 displays per pintar un número mitjançant la tècnica de multiplexat en el temps. Aquesta pràctica es fa la setmana del 14 al 18 de març.

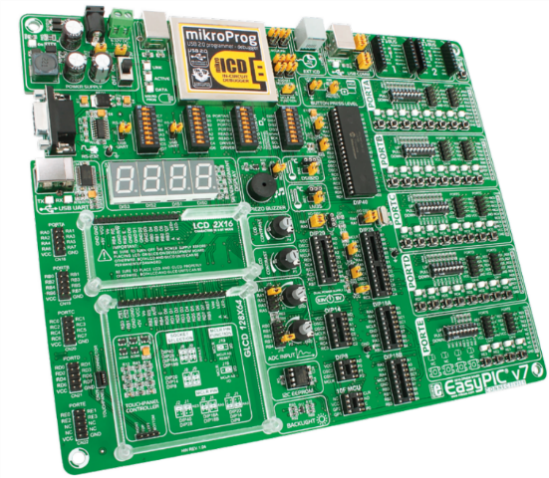
Durant la segona part de la pràctica L3(B), l’objectiu serà la programació de les rutines de servei a les interrupcions (RSI). Dels diferents tipus d’interrupcions existents, en aquesta pràctica treballarem sobre les interrupcions externes, en particular farem servir polsadors externs que al apretar-los o deixar-los anar executin codis específics a dintre del microcontrolador. Per fer-ho, s’haurà de comprendre la configuració dels diferents paràmetres que intervenen en la gestió de les interrupcions (habilitació, prioritats, flancs d’activació, etc.). Aquesta pràctica es fa la setmana del 21 al 25 de març.

En acabar la pràctica l’alumne serà capaç de:

* manegar informació tècnica donada pel fabricant de la placa de desenvolupament que utilitzem a les pràctiques.
* cercar la informació necessària dins dels manuals de referència.
* aprofundir el coneixement dels PORTS d’E/S.
* controlar l’activació d’un display de 7-segments per fer-hi un dibuix desitjat.
* entendre el concepte de multiplexat en temps, per aconseguir mostrar un nombre de diversos dígits al conjunt de 7-segments.
* entendre la diferència entre enquesta o interrupció per la gestió dels botons
* utilitzar els botons de forma que executin codi diferent quan estan apretats, quan no estan apretats, quan es produeix un flanc de pujada o quan es produeix un flanc de baixada
* entendre com configurar interrupcions en el PIC18F45K22
* entendre com implementar les rutines de servei a la interrupció
* entendre el concepte de flag de la interrupció
* entendre la utilitat d’activar o desactivar interrupcions

**2- Funcionament del display de 7-segments**

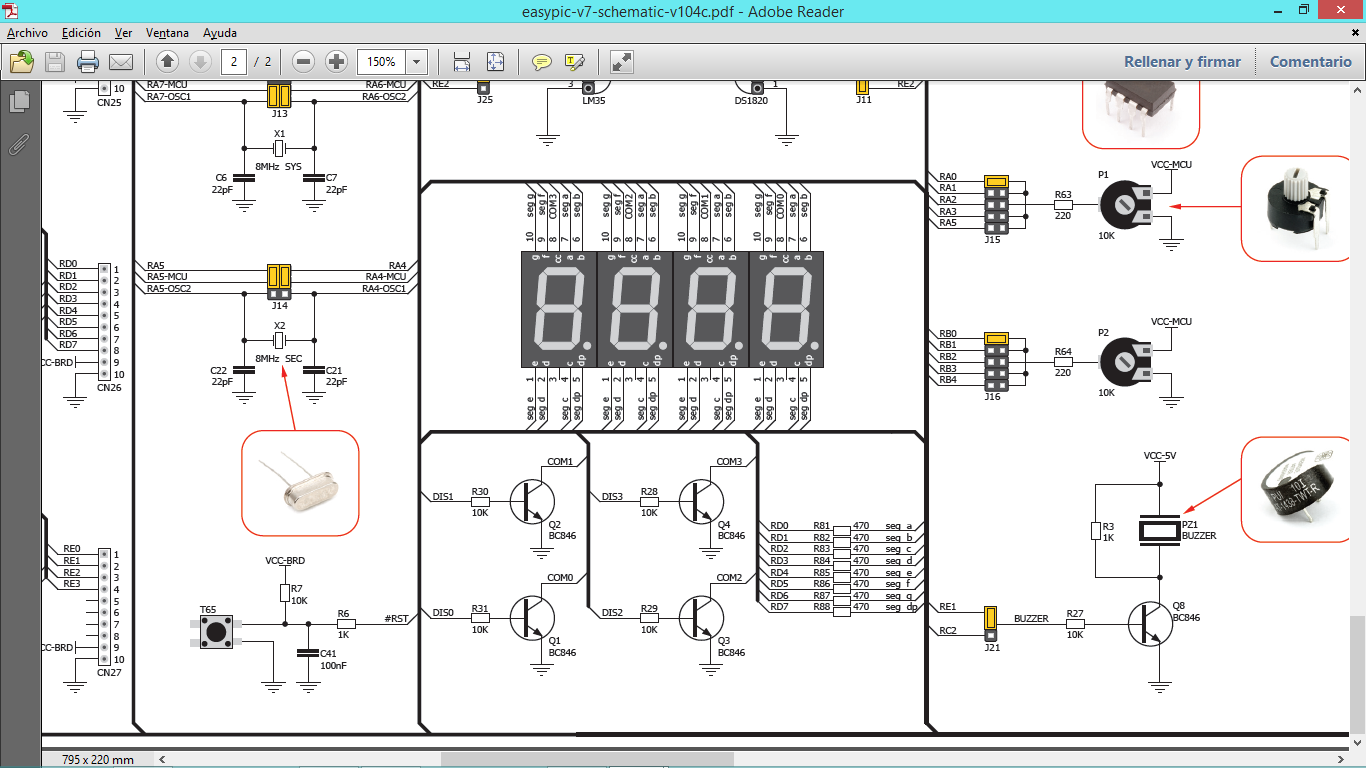
La placa de desenvolupament EasyPIC v7 (Fig.1) disposa de 4 displays de 7 segments que ens permetran mostrar informació de manera senzilla, emprant els ports d’E/S.



4 displays de 7-SEGMENTS

Fig. 1. Plataforma de desenvolupament EasyPIC v7

Aquests 4 displays de 7 segments estan connectats amb una configuració anomenada “Càtode Comú”, que vol dir que els 8 leds del display 7-segments (7 per crear la figura i un pel punt decimal) comparteixen una connexió a massa, i per tant són actius a “1”. La figura 2 mostra la connexió dels quatre displays extreta del manual de la placa.



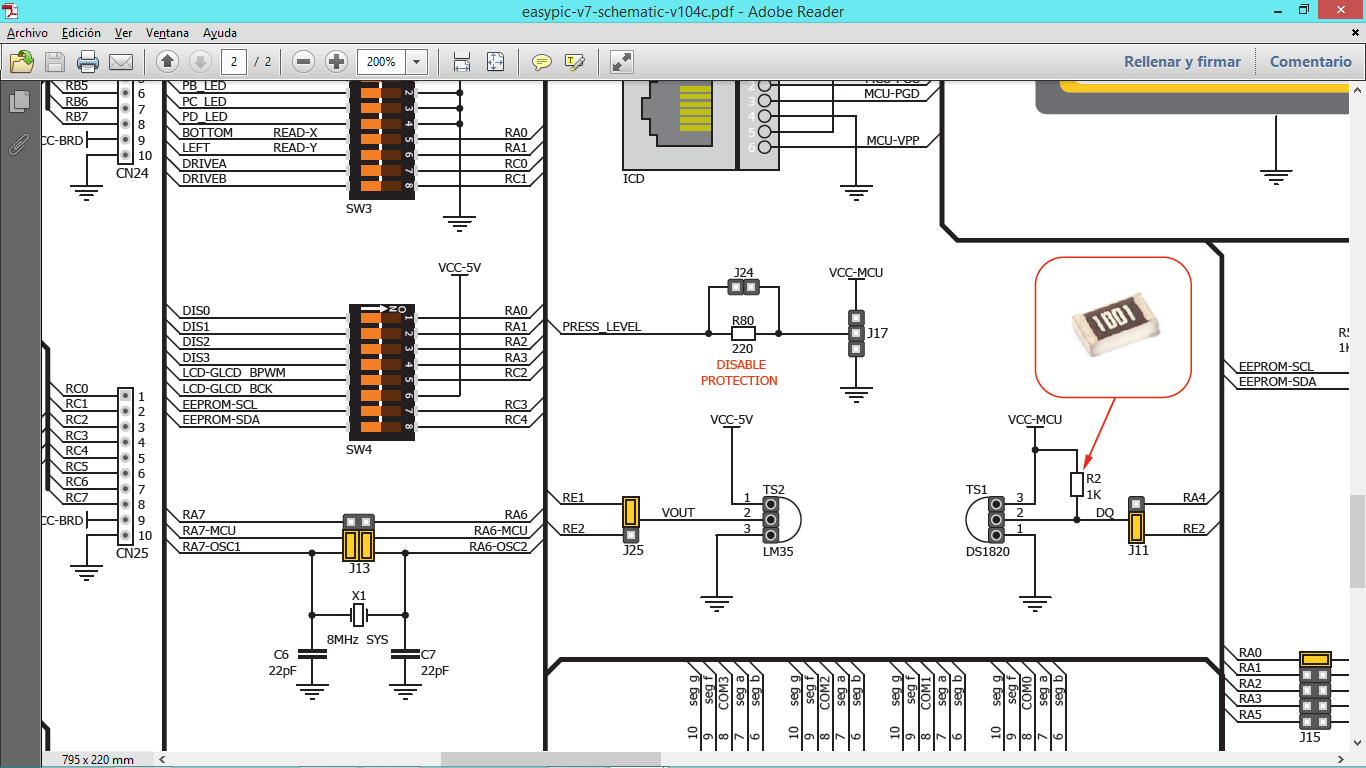


Fig. 2. Connexió a la placa dels 4 7-segments.

Observant les connexions a la placa podem concloure que:

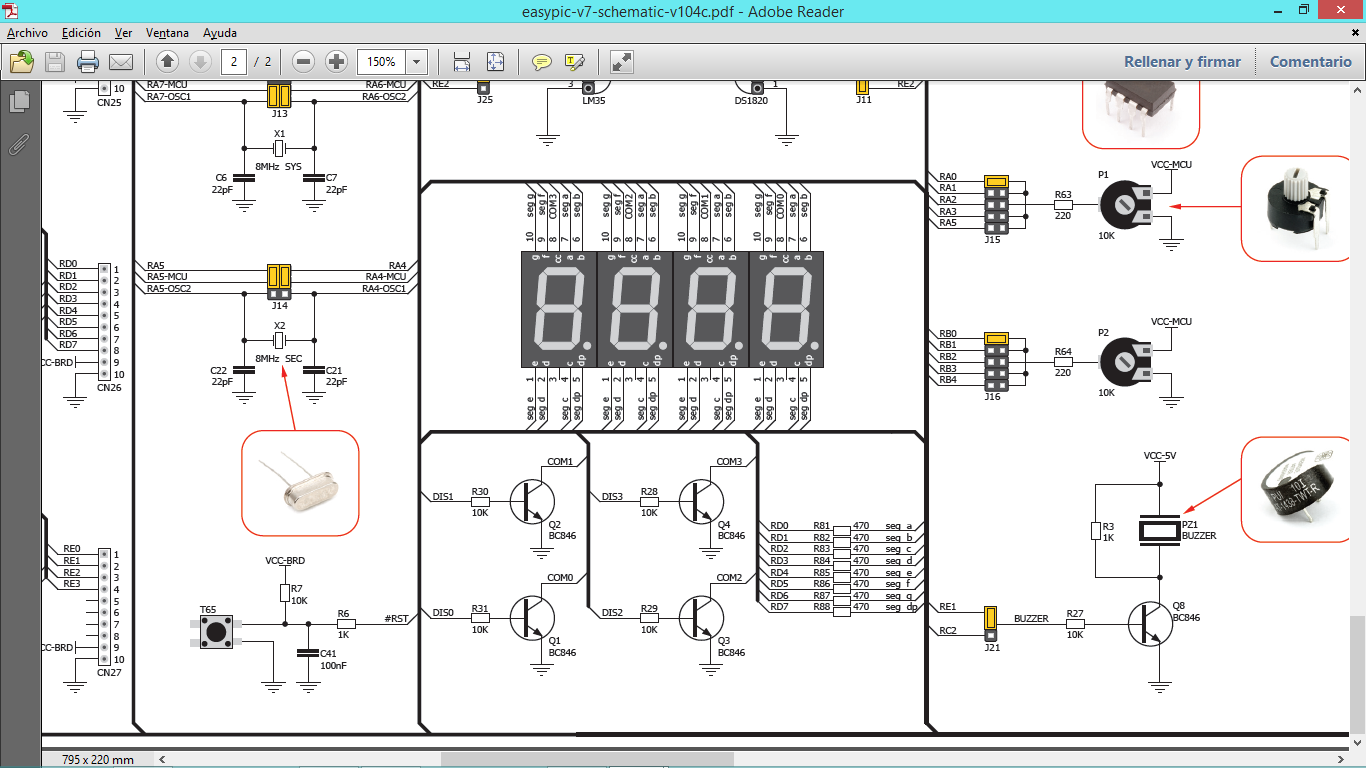
- L’activació dels segments del display depèn del PORTD.

- El PORTD està connectat als quatre displays simultàniament.

- La selecció del display actiu es fa mitjançant el càtode comú, amb uns senyals anomenats DIS0, DIS1, DIS2 i DIS3.

- Els senyals DISx es poden connectar al PORTA mitjançant el switch SW4.

Finalment, si observem amb detall el circuit de selecció del display actiu, per selecció del càtode comú, veiem que el regeix un transistor tipus NPN i que s’activarà per “1”, és a dir, quan posem un “1” a la senyal DISx es selecciona el corresponent display.



**“1”**

Fig. 3. Circuit de selecció del càtode.

Per pintar en el display s’ha de seleccionar un únic display DIS0, DIS1, DIS2 o DIS3 amb el PORTA i tot seguit enviar pel PORTD el valor a pintar. A continuació s’espera una mica, es selecciona el següent display i se li envia el nou número i així successivament. Si s’espera molt temps entre un dígit i el següent es veuran displays que s’encenen i s’apaguen donant una resultat poc atractiu. Si s’espera poc temps es barrejaran els valors pintats al conjunt. Teniu més informació disponible a Atenea a l*’Annex: Multiplexat de 7-segments. Recull d’informació addicional*.

A continuació s’inclouen una sèrie de passos per simular els 4 displays de 7-segments a Proteus:

2.1- Incloure l’arxiu config.h al projecte de Proteus.

2.2- Configurar la simulació del PIC18F45K22 a Proteus perquè funcioni a 8MHz de tal manera que la simulació i la placa EasyPicv7 funcionin a la mateixa velocitat de rellotge. Per fer-ho heu d’anar a la pestanya de l’esquemàtic i fer doble click sobre el microcontrolador. S’obrirà una finestra per editar el component i s’haurà de modificar el valor de Processor Clock Frequency a 8MHz.

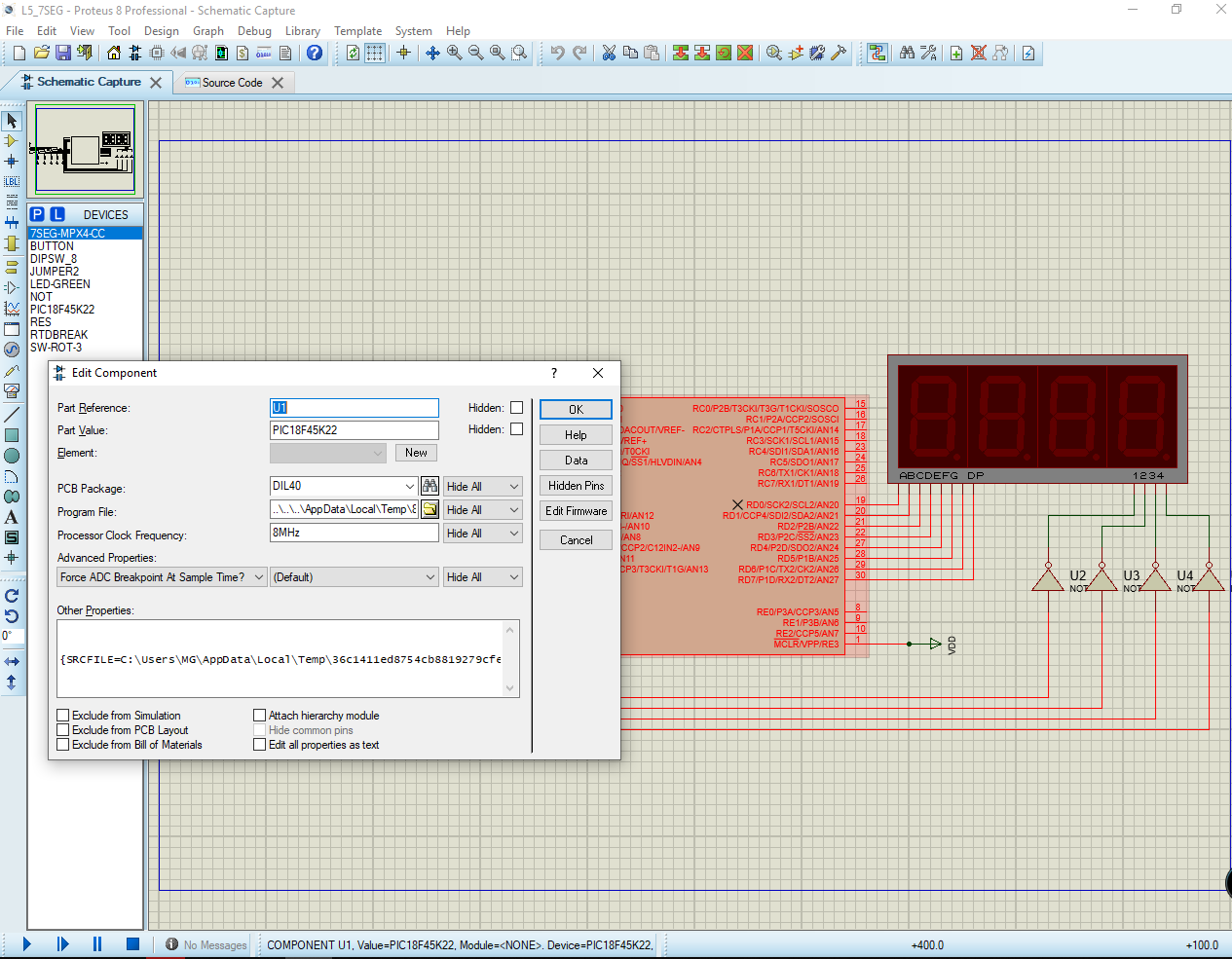


Fig. 4. Captura de pantalla pel canvi de la velocitat del clock.

2.3-Dissenyar els nombres a pintar als 7-segments; per cada un dels deu dígits {0..9} crearem un mapa de 8 bits (que posarem al PORTD) perquè el representi.

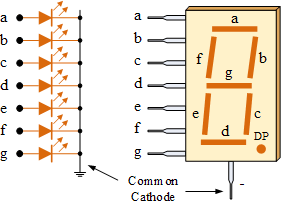


Fig. 5. Esquema dels 7-segments.

2.4-Disseny de l’esquema elèctric sobre Proteus. Farem el disseny amb Proteus amb el microcontrolador PIC18F45K22 i hi inclourem el bloc de 4 7-segments “**7SEG-MPX4-CC**” que funciona en càtode comú i inclou 4 displays, tal com necessitem. Per gestionar la selecció dels displays, canviarem el circuit real amb els transistors NPN per portes NOT. A nivell lògic funcionen igual (al posar un “1” a la porta, surt un “0” que selecciona el càtode comú d’un display) i estalviem que Proteus hagi de fer la simulació d’un circuit analògic amb els transistors, que ralentitzaria la simulació.

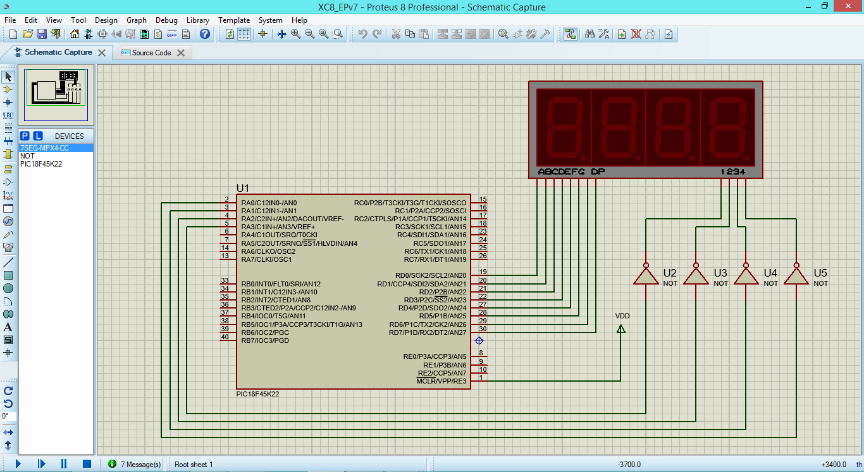


Fig. 5. Model per l’esquema de Proteus.

**3- Treball previ L3(A)**

Fer un programa en llenguatge C amb les següents funcionalitats:

3.1- Llegir l’estat del pin RB0 per enquesta. Si el botó està **apretat** es suma una unitat a un comptador. El valor d’aquest comptador es pinta al display de 7 segments. El valor del comptador augmentarà molt ràpidament, llavors si el comptador sobrepassa el valor 9999, es reseteja a 0.

3.2-Llegir l’estat del pin RB1 per enquesta. Cada cop que es detecti un **flanc de pujada** en aquest pin es suma +1 al comptador. El valor d’aquest número es pinta al display de 7 segments. Si el comptador sobrepassa el valor 9999, es queda a 9999.

3.3-Llegir l’estat del pin RB2 per enquesta. Cada cop que es detecti un **flanc de baixada** en aquest pin es resta -1 al comptador. El valor d’aquest número es pinta al display de 7 segments. Si el comptador baixa de 0, es queda a 0.

Serà clau controlar el temps per permetre la correcta visualització dels dígits al display.

Aprofiteu al màxim el codi per tal que sigui compacte, modular i fàcilment ampliable. Feu servir apropiadament variables, constants, comptadors, etc.

No hi ha qüestionari per la pràctica L3(A) però sí per la L3(B)

Podeu observar gràficament el funcionament desitjat a les següents figures:

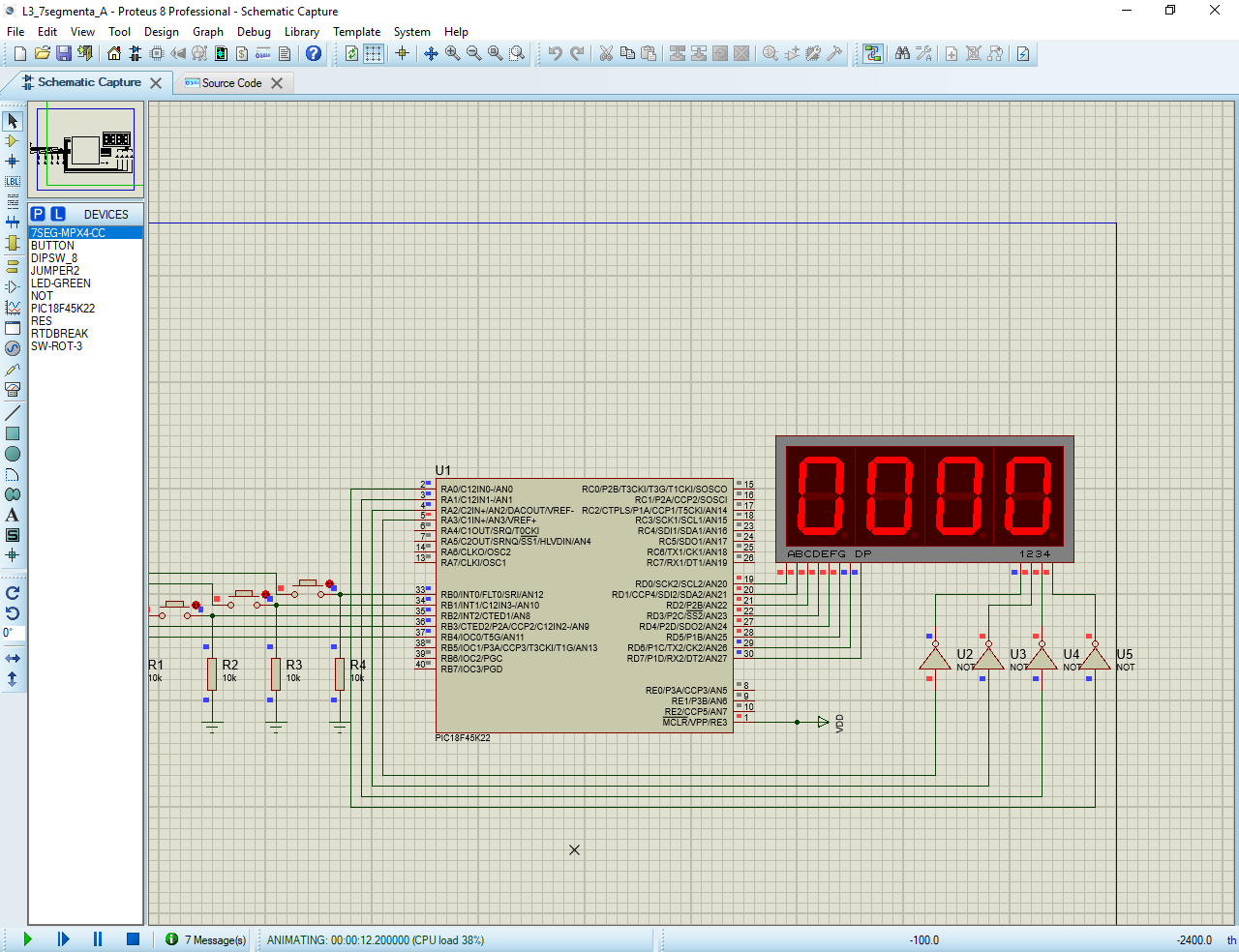


Fig. 4. Estat inicial.

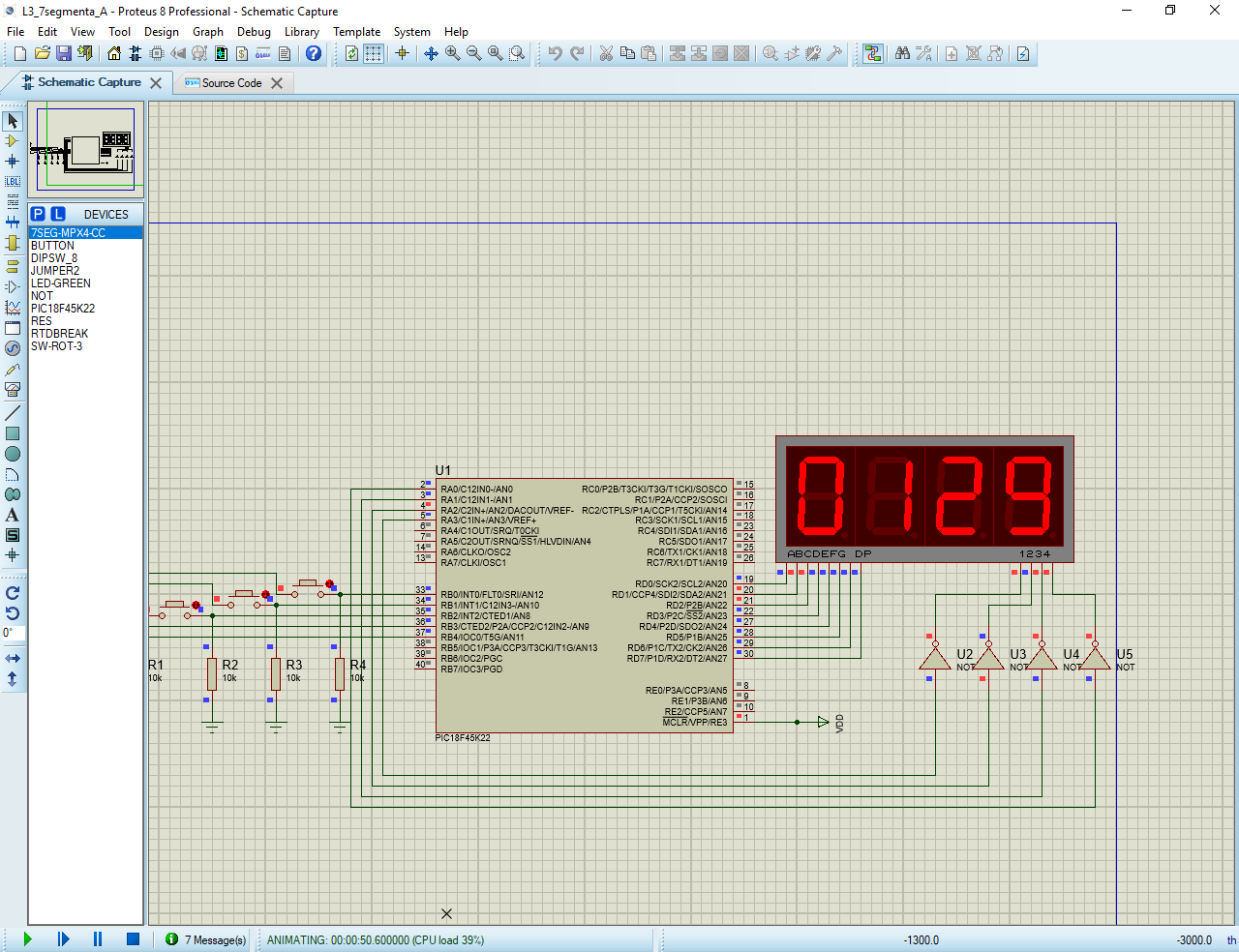


Fig. 5. Estat al apretar el botó RB0, el comptador augmenta ràpidament mentre estigui pitjat.

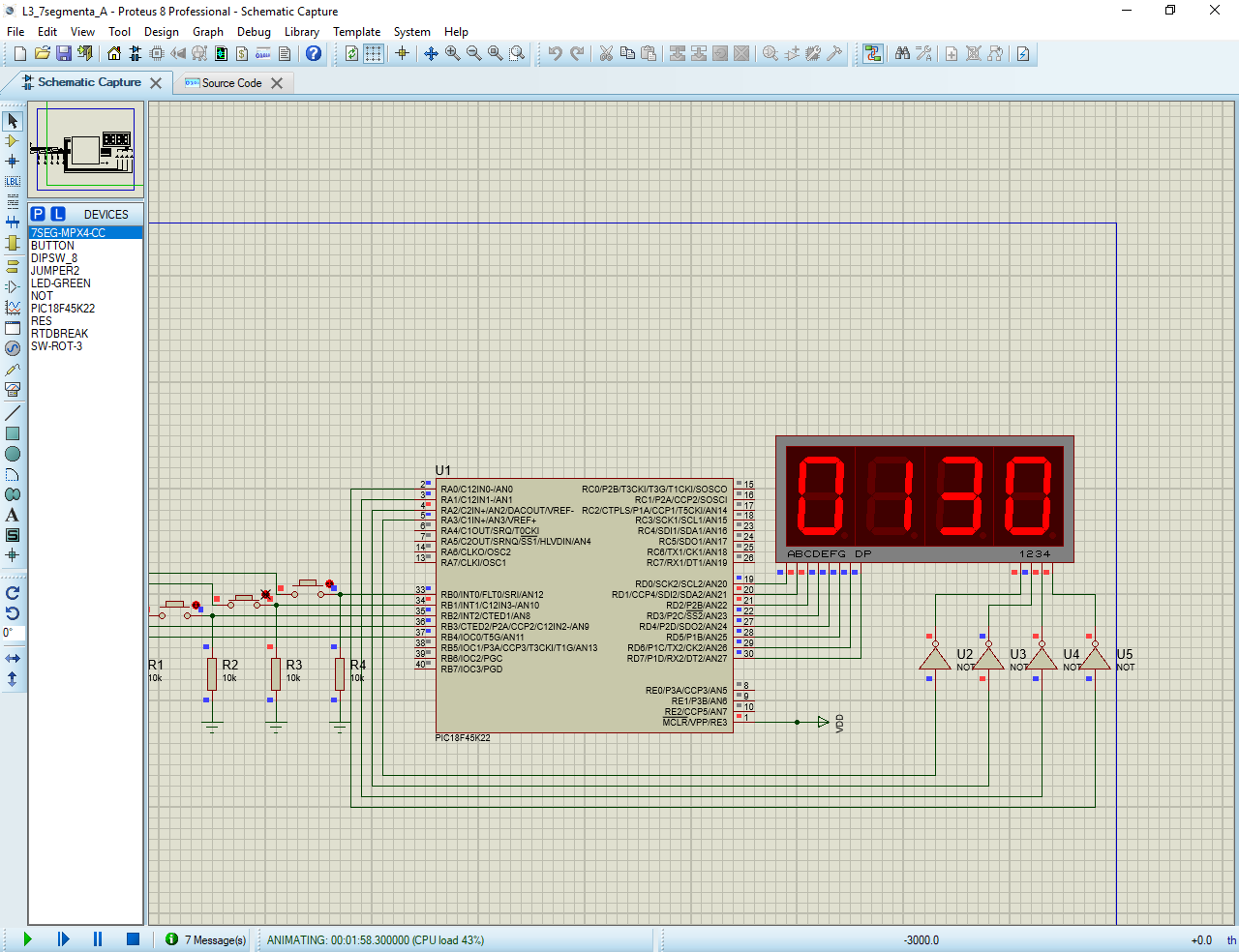


Fig. 6. Estat al apretar (flanc de pujada) el botó RB1, el comptador augmenta en una unitat.

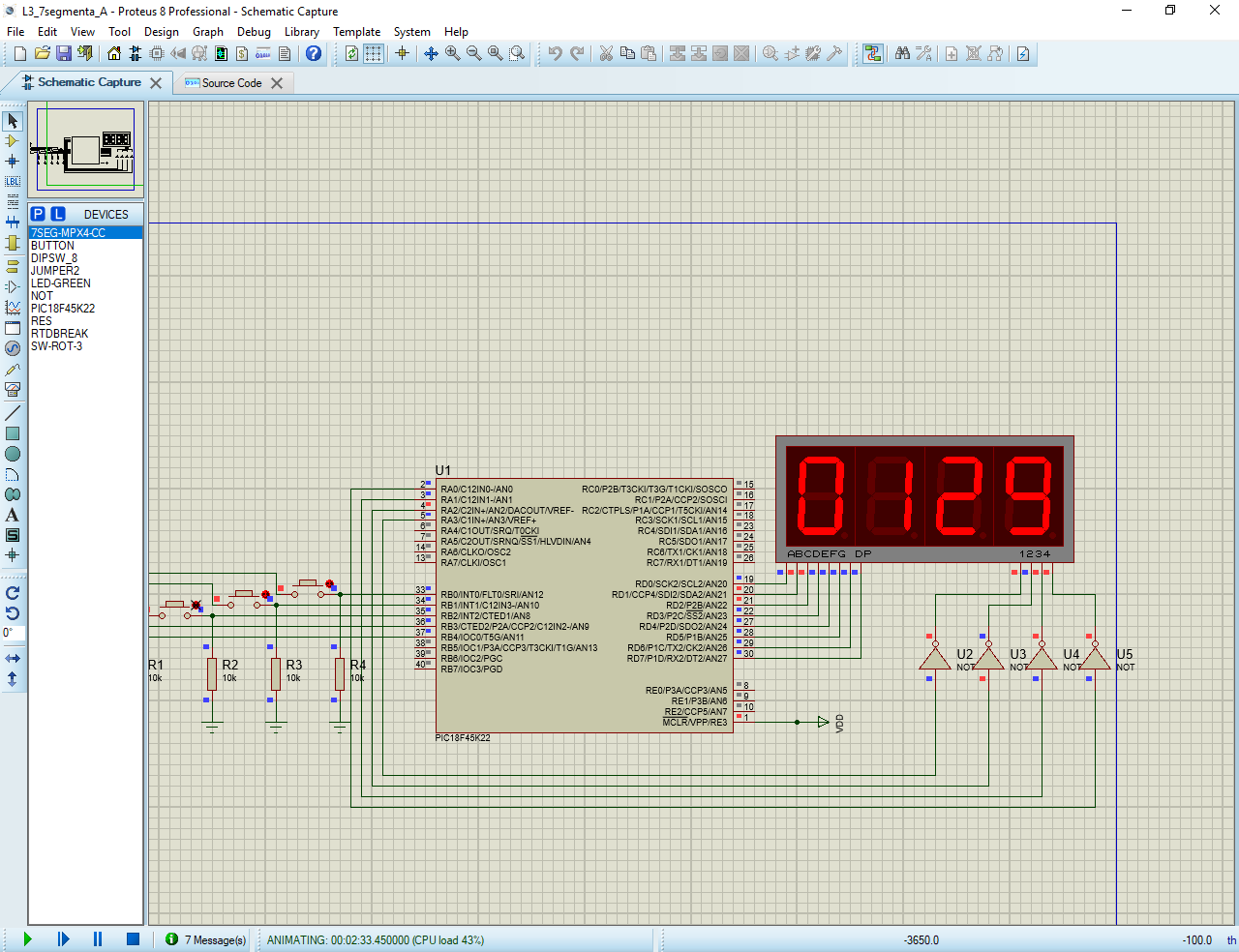


Fig. 7. Estat al apretar (flanc de baixada) el botó RB2, el comptador es decrementa en una unitat.

**4- Rúbrica treball previ L3 (A)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Iniciat  (0-2.5 punts) | En desenvolupament  (2.5-5.0 punts) | Aconseguit  (5.0-7.5 punts) | Exemplar  (7.5-10 punts) |
| Botò RB0 (2 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (apretat) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Botò RB1 (2 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (flanc de pujada) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Botò RB2 (2 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (flanc de baixada) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Display 7-segments (4 punts): | Hw mal dissenyat | Visualització incorrecta als displays | Codi poc eficient | Funciona perfectament |

**5- Treball previ L3(B)**

El programa a desenvolupar consisteix en un comptador que es pintarà al display de 7-segments i estarà controlat per 3 botons que generen interrupcions. Un dels botons es configurarà per habilitar els altres dos, que estaran encarregats de pujar o baixar el valor del comptador.

Fer un programa en llenguatge C amb les següents funcionalitats:

5.1- Configurar un polsador associat a la interrupció externa INT0 d’alta prioritat que quan es detecti un **flanc de baixada** habiliti o deshabiliti les interrupcions de baixa prioritat INT1 i INT2. Si aquestes estan habilitades llavors passen a inhabilitades i viceversa.

5.2- Configurar un polsador associat a la interrupció externa INT1 de baixa prioritat que quan es detecti un **flanc de pujada** sumi +1 al comptador i pinti el valor del comptador al display de 7-segments.

5.3- Configurar un polsador associat a la interrupció externa INT1 de baixa prioritat que quan es detecti un **flanc de pujada** resti -1 al comptador i pinti el valor del comptador al display de 7-segments.

5.4- Realitzar l’execució, test i debugat del vostre programa sobre PROTEUS. Les interrupcions són una font contínua de bugs. Aquests són molt difícils de tracejar, doncs degut a la natura asíncrona de les interrupcions, els errors de programa succeeixen de forma imprevisible. Degut a la complicació que suposa debugar el programa en fase d’execució, és recomanable ser meticulós en la fase de desenvolupament del codi. Per treballar de forma correcta, cal que seguiu els següents passos:

- Posar breakpoints a les RSI. Comproveu que sempre que feu saltar una interrupció, s’executa la RSI que toca. Comproveu que els bits de control implicats dins i fora de la RSI (flags, prioritats, emmascarament) són els esperats.

- Poseu un hardware breakpoint al pin INT1 que s’activi amb el flanc de pujada (podeu veure el vídeo tutorial *Breakpoints* i *Hardware Breakpoints* en la web de Proteus (<https://www.labcenter.com/tutorials/>), i un software breakpoint a la primera instrucció de la RSI de baixa prioritat.

- Comproveu que no salten més interrupcions que les provocades pels polsadors.

- Estresseu el sistema per a forçar l’aparició de bugs no detectats. Aquesta tècnica consisteix en provocar moltes més interrupcions per segon que les que s’esperarien en funcionament normal. Per a tal fi, substituïu els polsadors per un generador de funcions que generi interrupcions a una freqüència molt elevada. Comproveu fins a quina freqüència màxima pot arribar el generador abans de que el vostre codi deixi de funcionar correctament (els displays de 7-segments deixen de funcionar).

5.5- Omplir el qüestionari que trobareu a la última pàgina i entregueu-lo juntament amb el projecte de Proteus abans de la vostra sessió de laboratori.

**6- Rúbrica treball previ L3 (B)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Iniciat  (0-2.5 punts) | En desenvolupament  (2.5-5.0 punts) | Aconseguit  (5.0-7.5 punts) | Exemplar  (7.5-10 punts) |
| Botò INT0 (2.5 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (flanc de baixada) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Botò INT1 (2.5 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (flanc de pujada) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Botò INT2 (2.5 punts): | Hw mal dissenyat i funcionament erroni | Hw mal dissenyat o funcionament erroni (flanc de pujada) | Funcionament correcte però mal ús de recursos | Funciona perfectament |
| Display 7-segments (2.5 punts): | Hw mal dissenyat | Visualització incorrecta als displays | Codi poc eficient | Funciona perfectament |

Noms:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_GRUP:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Qüestionari – L3 (B)**

***(s’ha d’entregar en format electrònic abans de la sessió de laboratori)***

1) Dibuixa un diagrama de flux amb els estats i les transicions del programa descrit a l’apartat **5**

2) Indiqueu el contingut dels següents registres (en binari) just després d’haver saltat el hardware i el software breakpoint en INT1 (segons us demanem a la secció **5.4** d’aquest enunciat).

Hardware breakpoint: Software breakpoint:

INTCON = INTCON =

INTCON2= INTCON2=

INTCON3 = INTCON3 =

3) Quin és el *elapsed time* que us indica Proteus (temps d’execució entre dos breakpoints consecutius, indicat en la barra inferior), des de l’instant en que salta el hardware breakpoint al apretar el botó associat a INT0 fins al software breakpoint en la primera línia de la RSI? Justifica aquest retard.

4) Quan estresseu el sistema, quina és la freqüència màxima a la que podem tractar TOTES les interrupcions?