



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

Enunciat de la pràctica de laboratori

Interrupcions

Interrupcions

Objectius

L'objectiu d'aquesta pràctica és familiaritzar-se amb la programació de les rutines de servei a les interrupcions (RSI) per les interrupcions externes. Comprendre la configuració dels diferents paràmetres que intervenen en la gestió de les interrupcions (habilitació, prioritats, flancs d'activació, etc). Per assolir aquests objectius es treballarà amb les interrupcions del microcontrolador PIC18F45K22 i la placa de desenvolupament EASYPIC7 de MikroElectronica.

Coneixements previs de l'alumne:

L'alumne ja coneix:

- L'arquitectura del PIC18F45K22
- El simulador PROTEUS
- La programació del PIC en C
- El funcionament dels ports d'E/S del PIC
- Les particularitats i instruccions pròpies del compilador XC8
- El funcionament de les interrupcions d'alta i baixa prioritat
- Les opcions de la placa EASYPIC7 referents a la connexió dels polsadors a VDD, VSS, configuracions de *pull-up* i *pull-down*
- Les opcions de la placa EASYPIC7 referents a la connexió dels leds als ports del micro

Pràctica base

Implementar un programa en C que compti el nombre de vegades que es premem els polsadors connectats a uns pins d'entrada del micro. El valor d'aquest comptador cal que es mostri pels 4 displays de 7-segments connectats al PORTD. Cada cop que es premi algun dels polsadors es generarà una interrupció externa. Respondrem a la pulsació de 3 botons, un per pujar el comptador en 1 unitat, un per netejar el comptador a zero, i un per desactivar la funcionalitat dels altres botons:

- Primer polsador: en deixar anar el botó (per flanc de baixada del senyal) s'executarà una rutina d'**alta prioritat** associada a la interrupció INT0 que desactivarà o activarà les interrupcions de baixa prioritat.
- Segon polsador: en prémer el botó (per flanc de pujada del senyal) s'executarà una rutina de **baixa prioritat** associada a la interrupció INT1 que incrementarà el comptador.
- Tercer polsador: en prémer el botó (per flanc de pujada del senyal) s'executarà una rutina de **baixa prioritat** associada a la interrupció INT2 que posarà el comptador a zero.

Si en algun moment el comptador és superior a **10**, sortirà un missatge d'error a la pantalla (veure figura 1) i s'esperarà l'arribada de la interrupció INT2 per tornar a zero.

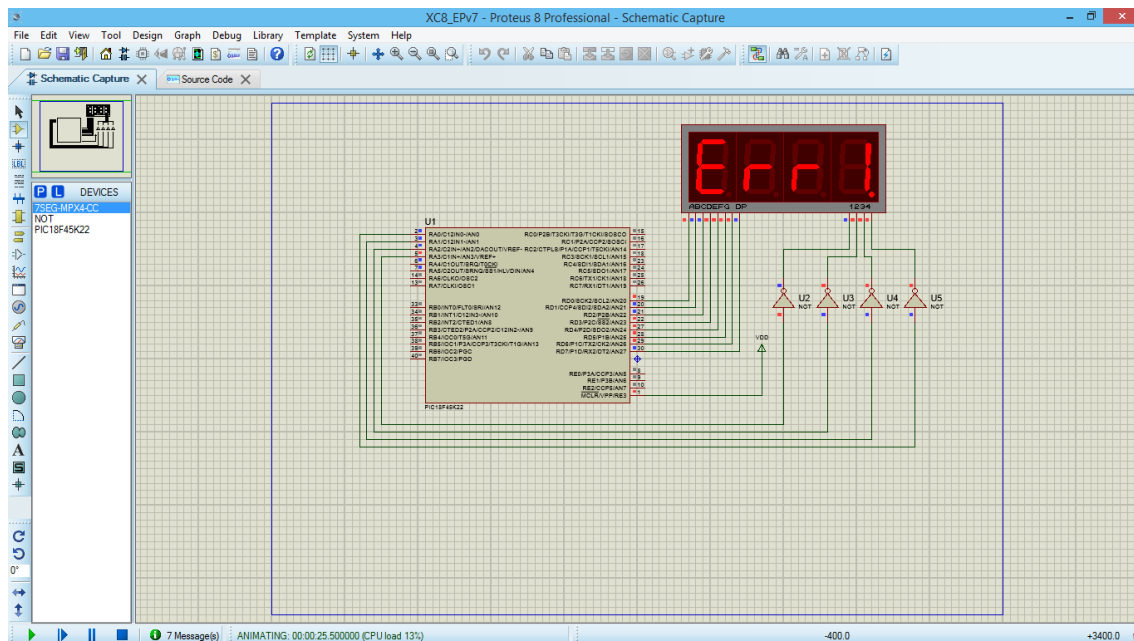


Figura 1. Missatge d'error del comptador (NUM<0, NUM>9999)

Treball previ

Temps estimat: 3 hores

- Estudar l'apartat 5.9 del *User's Guide* del compilador XC8 referent a la programació de rutines de servei a interrupcions.
- Consultar el Capítol 9 (*Interrupts*) del *DataSheet* del microcontrolador per resoldre el que es demana en aquesta pràctica.
- Esbrinar l'adreça del vector d'interrupció pel cas d'alta i baixa prioritat.
- Decidir si la interrupció es produirà al flanc de pujada o baixada.
- Estudar la manera d'assignar la prioritat a les interrupcions externes.
- Implementar el programa.
- Esbrinar en el *datasheet* del microcontrolador per quins ports i per quins bits entren les interrupcions externes INT0, INT1 i INT2, així com els corresponents pins del microcontrolador.
- Dissenyar l'esquema electrònic sobre PROTEUS usant els components que calguin per a poder simular la pràctica respectant, lògicament, els ports i els pins per on entren les interrupcions externes INT0, INT1 i INT2.
- Realitzar l'execució, test i depurat (debug) 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.

Entregueu el projecte PROTEUS (inclou diagrama esquemàtic, codi font implementat i el fitxer executable “.hex”) pel Racó, ABANS de la vostra sessió de pràctiques. Per a garantir compatibilitats de versions, lliureu sempre el vostre treball salvat en una versió compatible de PROTEUS (versió actual als laboratoris, v8.4 SP0).

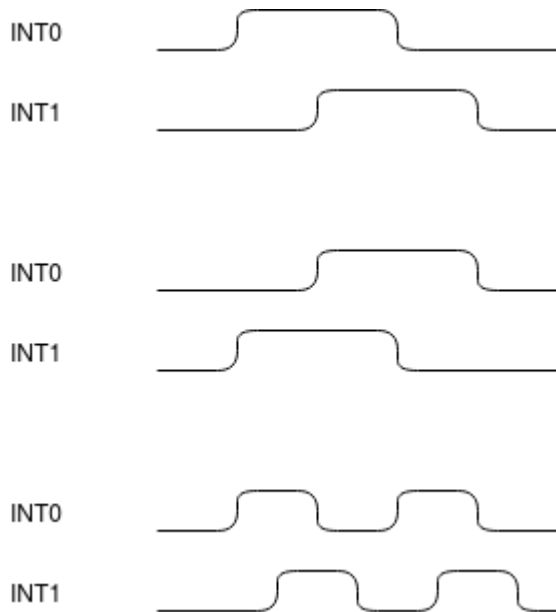
Pràctica al laboratori

- En iniciar la sessió de pràctiques al laboratori, haureu de mostrar el programa que heu implementat funcionant en PROTEUS.
- A continuació, haureu de passar la implementació a la placa EASYPIC7, per la qual cosa cal:
 - Esbrinar a l'esquema electrònic de la placa EASYPIC7 quins polsadors son els que arriben a les interrupcions externes INT0, INT1 i INT2.
 - Configurar els polsadors tal com els teniu a l'esquema de PROTEUS a partir dels *jumpers* i *switches* corresponents de la placa.
 - Connectar un extrem del polsador a VCC o GND mitjançant el *jumper* J17.
 - Connectar una resistència de *pull-up* o *pull-down* a l'altre extrem mitjançant el SW del port corresponent.
 - Programar el microcontrolador amb el *firmware* implementat i provat sobre PROTEUS.
 - Comprovar el correcte funcionament del programa en la placa.
- Comprovareu que es pot produir una lleugera diferència sobre la placa quan, a cada pulsació, s'activa més d'una interrupció. Implementar la corresponent estratègia anti-rebots.
- A continuació s'haurà de modificar el programa per a que realitzi la funció que determini el professor. Aquesta part de la pràctica s'haurà de realitzar “in situ” al laboratori, i no disposeu d'enunciat. L'enunciat us el proposarà el professor en la mateixa sessió, un cop hagueu entregat el treball previ.

Qüestionari d'interrupcions

**A ENTREGAR EN FORMAT PAPER A L'INICI DE LA
SESSIÓ DE LABORATORI**

1) Donat els següents cronogrames temporals, digueu quina acció o accions s'espera del firmware descrit a la secció Treball Previ d'aquest enunciat.



2) Indiqueu el contingut dels següents registres (en binari) just després d'haver saltat el hardware breakpoint en INT1 (segons us demanem a la secció Treball Previ d'aquest enunciat).

INTCON =

INTCON3 =

3) Quin és el *elapsed time* que us indica Proteus (temps d'execució entre dos breakpoints consecutius, indicat en la barra inferior), en el moment en que salta el 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 generar interrupcions sense perdre'n cap?