

PRÀCTIQUES DE L'ASSIGNATURA

INTRODUCCIÓ ALS ORDINADORS



UNIVERSITAT DE BARCELONA



Índex

PRÀCTIQUES DE L'ASSIGNATURA.....	1
INTRODUCCIÓ ALS ORDINADORS.....	1
1. Normes de les Pràctiques.....	3
Avaluació.....	3
Sessions Pràctiques.....	3
Normativa.....	3
Pràctica 8: Exemple d'entorn real.....	4
PART I:Objectius.....	4
PART II: Entorn de programació de Microchip: Introducció.....	4
Creació d'un projecte.....	4
Entorn de simulació.....	5
Entorn de depuració.....	5
Entorn de programació.....	6
Annexe: Programació del Microcontrolador.....	6
Pràctica 9. Gestionant Interrupcions.....	8
Objectiu de la pràctica.....	8
Realització de la pràctica.....	8

1. Normes de les Pràctiques

Avaluació

Les pràctiques d'Introducció als Ordinadors són una part integrant de l'assignatura. Aquestes pràctiques es realitzen a l'aula IE de la facultat de Matemàtiques i comporten la realització de 10 pràctiques avaluables, un miniprojecte i un examen pràctic final. El resultat de l'avaluació d'aquestes pràctiques constitueix el 50% de la nota de l'assignatura. Per poder aprovar l'assignatura la nota mitjana de pràctiques ha de ser superior o igual a 5 i caldrà presentar-se a totes les sessions i entregar tota la documentació requerida.

Sessions Pràctiques

L'assignatura consta de un total de 10 pràctiques dividides en 10 sessions i un miniprojecte com a treball a realitzar per l'alumne a casa. S'establiran períodes d'entrega a través del Campus Virtual de l'assignatura. Si l'entrega no es realitza en el període establert, la pràctica constarà com suspesa.

Normativa

- Els alumnes treballaran a l'aula individualment (si hi ha ordinadors suficients). En cas de que es disposi d'un ordinador portàtil propi, pot portar-se a l'aula per a la realització de les pràctiques si el alumne així ho vol.
- Les pràctiques es realitzaran utilitzant el sistema operatiu Windows XP. La contrasenya d'entrada és l'assignada per accedir als ordinadors de la facultat
- Queda totalment prohibit instal·lar o utilitzar programes propis als ordinadors de l'aula
- No està permesa la utilització dels ordinadors per realitzar treballs d'altres assignatures durant les pràctiques.

Pràctica 8: Exemple d'entorn real

(1 sessió +1/2 si és necessari)

PART I: Objectius

Utilitzar les eines de depuració i programació d'un microcontrolador en un entorn real

Recórrer tot un procés complet de la programació d'un microcontrolador des de l'origen fins al final

PART II: Entorn de programació de Microchip: Introducció

L'entorn de depuració i programació desenvolupat per Microchip: MPLAB IDE, és una eina que ens permet realitzar tot el procés complet de programació, depuració i gravació directament sobre la memòria del microcontrolador escollit.

La principal diferència amb les simulacions que hem estat fent anteriorment és que a partir d'ara serà el mateix microcontrolador qui executarà les ordres que li hem programat, sense la intervenció del microprocessador de la placa base del nostre PC. Per poder realitzar la pràctica necessitarem una placa electrònica amb el microprocessador i una interfície que ens permeti comunicar-nos amb aquesta placa des del nostre PC. Utilitzarem la placa de evaluació de Microchip Explorer 16, amb un microcontrolador de la família PIC24F i la interfície de programació ICD2 (In Circuit Debugger 2).

Creació d'un projecte

Tal i com es treballa actualment amb els llenguatges d'alt nivell, la programació amb l'entorn MPLAB requereix també la creació d'un espai de treball que englobarà tots els arxius necessaris. Aquest espai de treball s'anomena **projecte**.

La creació d'un projecte nou es pot fer senzillament seguint les següents indicacions:


- ⇒ Al menú **Project** escollim la opció **Project Wizard** on trobarem una guia que ens ajudarà amb tots els passos necessaris per aconseguir la creació d'un projecte.
 - El microcontrolador que s'ha d'escollir és el p24fj128ga010
 - El següent pas serà escollir el nom del projecte. Donada la natural creativitat dels alumnes es proposa posar el nom Fonaments
 - En el mateix pas s'ens demanarà la ubicació on s'ha de posar el projecte, es recomana crear una carpeta amb el mateix nom que el projecte on es desaran tots els fitxers.
 - Tot seguit el programa ens preguntarà quin llenguatge de treball s'utilitzarà i quin serà el compilador escollit. Seleccionem la opció Microchip MPLAB C30 compiler. Noteu que instantàniament el programa detecta la localització dels executables. Si no és el cas, els trobareu a: C:\Archivos de Programa\Microchip\MPLAB C30\BIN
 - Finalment, per programar el processador el compilador requereix de unes determinades llibreries també suministrades per Microchip. Els fitxers que s'han de incloure són:
 - C:\Archivos de Programa\Microchip\MPLAB

C30\support\h\p24fj128ga010.h i

- C:\Archivos de Programa\Microchip\MPLAB
C30\support\gld\p24fj128ga010.gld

- El projecte està creat

- ⇒ Un cop tenim el projecte descarreguem de dossiers el fitxer en C per realitzar la pràctica. El trobareu dins la carpeta de **Pràctiques** amb el nom de main.c
- ⇒ Guardeu el fitxer a la carpeta on s'ha generat el projecte
- ⇒ Dins la finestra Fonaments.mcw trobareu la carpeta source files. Cliqueu aquesta carpeta i cliqueu un altre cop però ara amb el botó de la dreta. Escolliu la opció Add files i seleccioneu el fitxer main.c. Aquest fitxer serà inclòs al projecte.
- ⇒ Fent doble clic sobre el fitxer s'obrirà una finestra on podeu veure el codi.


Pregunta 1: Qué fa aquest programa? 


Entorn de simulació

Abans de simular res cliqueu el botó de **Built all**. Donada la complexitat de trobar aquest botó a l'entorn de treball podeu demanar ajut al professor. El que permet aquest botó és compilar el programa, linkar tots els fitxers i generar el fitxer .hex que s'utilitzarà per programar el microcontrolador. Un cop fet això aneu a la finestra **Debugger** i dins la opció **select tool** escolliu la **MPLAB SIM**. Aquesta opció permet simular el projecte, al igual que feiem amb el SIMR i el i8085. Abans d'executar res obriu la pestanya **view** i seleccioneu:

- ⇒ Watch que ens permetrà veure com varien els registres i les variables que introduïm en el programa
- ⇒ Program Memory, ens permetrà veure la traducció de C a ensamblador. Podem veure d'aquesta manera com es desglossa les diferents instruccions.

Executeu el programa pas a pas clicant la opció on surt una fletxa entre dos corxets. Veieu el seu funcionament.


Pregunta 2. A quantes línies de codi en ensamblador s'hi correspon la instrucció **for?** 


Pregunta 3. En quina posició de la memòria es troba? Qui tradueix de C a ASM? 

Entorn de depuració

Un cop vist que el programa funciona correctament amb el simulador, procedim a debugar el codi a la placa. Primer de tot connectem el programador (aquesta capsula rodona que sembla una pastilla gegant) fent servir el port USB. El programa instal·larà els drivers.

1. Seleccioneu la opció de detecció automàtica i digueu després sí a tot.
2. Alimenteu la placa amb la font que vé per fer tal efecte i connecteu el programador.
3. Finalment, escollim la pestanya Debugger novament però ara, dins la opció select tool seleccionem la opció MPLAB ICD2.
4. Construïm novament el projecte amb la opció **built all** i
5. programem el microcontrolador. Això es fa amb la opció **program target device**.
6. Executem pas a pas i veiem que és el que fa.

Pregunta 4: Funciona igual la placa que el simulador? 

Pregunta 5: On està carregat el programa que s'està executant? 

Entorn de programació

Un cop depurat el codi i fetes les modificacions que considerem oportunes arriba l'hora de programar el microcontrolador. Els passos són els següents:

1. Cliqueu la pestanya de **Debugger** i dins de **select tool** escolliu la opció **none**
2. Cliqueu la pestanya de **Programmer** i dins de **select tool** escolliu la opció MPLAB ICD2.
3. Feu un Built all
4. Programeu el micro amb la opció **program target device**
5. Desconecteu el ICD2

Pregunta 6: Què fa la placa?



Pregunta 7: Com es pot millorar el programa?



Milloreu el programa. Entregueu totes les preguntes dintre de l'informe que haureu de pujar al Campus Virtual.

Annexe: Programació del Microcontrolador

El PIC24 és un micro de 16 bits, que pot programar-se en C i en ASM. A partir d'aquestes pràctiques començarem a programar en C, tot i que treballarem en un C "a baix nivell", operant sobre els registres que proporciona el microcontrolador.

Per exemple, suposem que volem programar un dels ports com una entrada. Busquem el datasheet del microcontrolador, que trobareu a la web:

www.microchip.com

i aneu a la part de ports d'entrada-sortida.

Veureu que hi ha dos registres bàsicament per configurar un port. El TRISx i el PORTx. En aquest sentit, si volem configurar el port D com a sortida farem:

TRISD = 0x00; //si posem els bits del registre TRIS a 0 el configurem com sortides

PORTD = 0xFF; //així el que fem es posar tots els bits a 1.

Si volem per exemple posar el bit 0 com sortida i la resta de bits com entrades farem:

TRISD = 0xFE;

La majoria dels ports no només poden treballar com entrades-sortides digitals, sinó que poden tenir altres opcions. Si la nostra intenció és que treballi com entrades-sortides digitals, haurem de deshabilitar les altres possibles opcions. Per això haurem de fixar-nos en el datasheet i llegir-lo detalladament.

A la pàgina web de microchip podem trobar exemples de codi per la majoria de les aplicacions que necessiteu, però es recomana atacar als registres ja que és la forma més senzilla tenint en compte els coneixements que teniu actualment. En posteriors assignatures podreu treballar amb altres nivells d'abstracció proporcionades pels fabricants.

Pel que fa a la programació en C, podríem dir que és molt similar al Java, amb la gran diferència que en C treballem amb punters i que no hi ha una màquina virtual que ens elimina problemes de memòria. Els tipus de dades són iguals:

char; int; unsigned int; byte; long; unsigned long; float;

No tenim el boolean, per tant, hauríeu de definir-ho vosaltres. Per exemple:

```
typedef char bool;
#define TRUE      0xFF;
#define FALSE     0x00;
```

i a partir d'aquí

```
bool a = TRUE;
bool b = FALSE;
```

Control de flux:

```
int contador = 0;
for(contador = 0; contador < 0xFF; contador++){
    TODO
}
while (contador < 0xFF){
    TODO
}
while(1){
    //bucle infinit
}
if(contador != 0){
    TODO
}
else if(...){
}
else{
}
```

altres tipus de comparacions booleanes són:

==; >; >=; <; <=; ...

Pràctica 9. Gestionant Interrupcions

(1 sessió +1/2 si és necessari)

Objectiu de la pràctica

L'objectiu d'aquesta darrera pràctica és fer servir les interrupcions del microprocessador PIC24F per :

- 1.- Executar una interrupció fent servir el **Timer** cada 1 segon
- 2.- Estudiar la Rutina d'Atenció a les Interrupcions
- 3.- Transmetre dades de forma sèrie

Realització de la pràctica

Feu servir la tarja de desenvolupament Explorer16 amb el microprocessador PIC24F. Connecteu la placa a la font d'alimentació de 9V i connecteu el programador ICD2. Comproveu que la connexió és correcta seleccionant al IDE MPLAB la opció debug i seleccionant el programador ICD2. El MPLAB hauria de detectar tant la Explorer16 com el microcontrol·lador. En cas contrari aviseu al professor de pràctiques per solucionar el problema.

Al campus virtual trobareu el codi que serveix per programar el temporitzador, introduïu-lo a l'entorn de programació i comproveu el seu funcionament.

Sempre que es produeix una interrupció el programa salta a una determinada posició de la memòria on s'executa la rutina d'atenció a les interrupcions. Aquesta rutina, la definirem d'aquesta forma en els nostres programes:

```
//*****//
//*****Gestió Interrupcions*****//
#pragma code HighInterruptVector = 0x08
void InterruptVector(void){
    _asm
        goto SaltaInterrupcio
    _endasm


#pragma code LowInterruptVector = 0x18
void InterruptVector2(void){
    _asm
        goto SaltaInterrupcio
    _endasm
```

D'aquesta forma, sempre que es produeixi una interrupció, el gestor d'interruptors saltarà a una funció anomenada SaltaInterrupció que tindrem definida en el nostre codi:


```
void SaltaInterrupcio(void){  
    TODO  
}
```

en aquesta funció mirarem quina interrupció s'ha activat, esborrarem el flag i tornarem a activar les interrupcions.

Tasca 1

Quin és el bit que s'activa quan salta la interrupció? 

Que feu a la Rutina d'Atenció a la Interrupció? 

Amb els coneixements adquirits a la pràctica anterior, escolliu dos leds de la placa Explorer16, un ens indicarà les dades que volem transmetre i l'altre el senyal de rellotge.

El senyal de rellotge anirà encenent i apagant alternativament el led seleccionat cada cop que salti la interrupció del temporitzador.

El senyal de dades s'encendrà si el bit que volem transmetre és un “1” i s'apagarà si el bit transmès és un “0”. Transmeteu la frase “hola mon”.

Finalment, guardeu la frase “hola mon” a la EEPROM que teniu a la Explorer16 de la placa. Apagueu l'alimentació i comproveu que la frase continua fent una lectura de la EEPROM.

Penjeu al campus virtual l'informe de la pràctica, amb tot el codi creat així com la resposta a les preguntes que es formulen a tota la pràctica.