

# Taximetre Digital

SDMI



Robert Millan Hernandez  
Felip Moll Marquès  
SDMI12A  
Q1 - 2008/09

## **Index:**

1. Enunciat de la pràctica
2. Estructura del programa
3. Rutina RSI
4. Altres coses a destacar
5. Diagrama de flux del programa
6. Codi del programa
7. Bibliografia

## 1. Enunciat de la pràctica:

Hem realitzat un taxímetre que disposa de les següents funcionalitats:  
Per part del taxista:

- Introducció de preus de tres tipus de tarifa, concretament es permeten els valors següents:

Tarifa 1 i 2: Preu baixada bandera, preu per KM recorregut, preu per temps.

Tarifa 3: Preu baixada bandera, preu per km recorregut, preu per temps i preu del suplement d'horari nocturn.

- Introducció de l'hora del sistema en format AM/PM.

- Consulta de la facturació, km's recorreguts i mitja de combustible als 100km durant el dia.

- Protecció mitjançant un password modificable de les dades anteriors.

- Càrrega de suplementes al preu del recorregut, els suplementes tenen un preu fix i estan definits a l'enunciat, són:

- suplement d'aeroport, de maleta, de moll, nit especial, fira o gos.

- Decisió de mantenir el taxi en qualsevol dels estats possibles. Es descriuen més endavant.

Per part del client:

- Podrà veure en tot moment l'import de la carrera que està fent i com es va incrementant.

- Podrà veure l'hora durant la carrera.

- Al final veurà el temps passat i l'import total, amb els possibles suplementes que el taxista decideixi carregar en aquell moment.

(Totes les introduccions de dades tenen possibilitat de cancel·lació, o rectificació, veure el diagrama de flux o codi font)

## Comentaris rellevants sobre l'enunciat:

Vam tenir alguns problemes ja que se'ns demanava que utilitzéssim algunes sortides o entrades que a la pràctica no hem pogut fer servir degut a que estaven multiplexades amb algun altre element crític del sistema. Un exemple el podem trobar al sw1, que està multiplexat amb l'RA0/AN0, l'entrada que es fa servir per el control de consum de combustible. Un altre exemple pot ser el dels sw2:4 que serveixen per decidir quina tarifa aplicar i que estan multiplexats amb 2 bits de l'LCD. Finalment podem trobar també que el display de set segments està connectat al PORTB i el teclat l'utilitza constantment, fent impossible de marcar-hi la tarifa actual tal com se'ns demanava.

Hem solucionat com hem pogut tots aquests i altres problemes de l'enunciat. Es troba tot detallat i comentat al codi.

## 2. Estructura del programa:

Fitxers i directoris:

Hem estructurat el codi font del programa en diversos fitxers.

'include' : Al directori include hi tenim els fitxers de capçaleres

'doc' : Documentació i notes del programa

"main.c": Codi principal del programa

"teclat\_lab.c" : Driver del teclat del Pic Laboratory

"lcd\_lab.c" : Driver del LCD del Pic Laboratory

"include/constants.h" : Fitxer amb totes les constants calculades que fem servir.

Codi font:

Hem seguit l'esquema que se'ns proposa a l'enunciat de fer els estats del taxímetre en blocs de codi. Aquests blocs els hem contingut dins un while(1) que s'executarà sempre i que segons una variable "Bloc" i un switch dins el while, anirà a parar a un bloc o a un altre.

Els blocs són els definits a l'enunciat, i que també tenim definits a constants.h:

#define LLIURE	0
#define REPOS	1
#define CONTROLS	2
#define OCUPAT	3
#define IMPORT_	4

La manera de canviar d'un bloc a un altre està definit al diagrama de flux que hem inclòs a l'informe, tot i així hem respectat l'esquema de l'enunciat permetent només aquells canvis d'estat que se'ns demanava.

### 3. Rutina RSI

Volem destacar el "hack" que hem hagut de fer a la rutina de servei a les interrupcions ja que vam tenir problemes des de gairebé el principi de la pràctica.

El problema era que quan la mida del codi sobrepassava l'espai del primer banc (25% de codi) i passava a ocupar el segon, si rebíem una interrupció i entràvem a la RSI del banc 0, després no podíem tornar al lloc d'on veníem.

Ho vam arreglar en varies ocasions definint variables com a constants amb el modificador "constant". També vam eliminar algunes funcions inline, però a mesura que el codi creixia el problema reapareixia.

El cas és que després de investigar perquè se'ns produïa vam veure que era degut a que vam haver d'utilitzar la directiva #int\_global per practicar amb les interrupcions i haver-les de gestionar manualment comprovant els bits d'enable i els flags d'interrupció (p.ex INTE i INTF).

#int\_global, segons diu al manual del compilador:

PG112:

#INT\_GLOBAL

Syntax: #int\_global

Elements: None

Purpose: This directive causes the following function to **replace the compiler interrupt dispatcher**. The function is normally not required and should be used with great caution. When used, **the compiler does not generate start-up code or clean-up code, and does not save the registers**.

Ens vam adonar que no es guardava el context, i quan saltàvem de un banc a un altre perdiem l'acumulador, el pclath i l'status. Vam cercar alguna manera d'arreglar-ho i la solució va ser crear un dispatcher per salvar el context just en el moment d'haver entrat a la interrupció. El dispatcher el vam treure del manual del compilador i una mica d'ajuda de google:

```
#int_global

void
interrupcions ()
{
#define W_OLD      0x20
#define STATUS_OLD 0x21
#define PCLATH_OLD 0x22
```

```

#asm
    movwf W_OLD;
    //Guardem el valor de l'acumulador
    W_OLD es trobarà a 0x20, 0xa0, 0x120 o 0x1a0 depenent del bank en
    que ens trobem

    swapf STATUS, W;    //Guardem el valor de l'STATUS a l'acumulador
    clrf STATUS;        //Netejem STATUS, cosa que fa que canviem al bank0
    movwf STATUS_OLD;   //Guardam l'STATUS anterior a 0x21 de bank0
    movf PCLATH, W;     //Guardem el PCLATH
    movwf PCLATH_OLD;   //a 0x22 del bank0
    clrf PCLATH;
#endasm

.... Aquí el codi dels handlers de cada interrupció ....

#asm //Restauem tot en l'ordre invers.
    movf PCLATH_OLD, W;
    movwf PCLATH;

    swapf STATUS_OLD, W;
    movwf STATUS;

    swapf W_OLD, F;
    swapf W_OLD, W;
#endasm
}

```

Al fitxer 16F876\_CCS.h hi vam incloure la sentència:

```

/* Reservem l'espai que necessitem per guardar W, STATUS i PCLATH.
   0x20, 0xa0, 0x120 i 0x1a0 serviràn per guardar l'W en el bank
   corresponent. Posteriorment es canviarà de bank i es salvarà
   a la posició 0x21 i 0x22 del bank0 l'STATUS i
   el PCLATH respectivament.
   Per retornar es farà de manera inversa i es recollirà W d'on calgui.
*/
#reserve 0x20, 0x21, 0x22, 0xa0, 0x120, 0x1a0

```

D'aquesta manera vam poder solucionar el problema del 25% de codi i seguir programant fins a arribar al final del programa, que ens ha ocupat un 80% de la ROM.

#### 4. Altres coses a destacar

##### Reimplementació de MUL, DIV i MOD:

Degut a un problema que encara no hem aconseguit identificar, les operacions de resta, divisió i mòdul ens reescrivien registres aleatoriament i se'ns corrompia la memòria. Per això vam reescriure els algorismes citats fent que fossin iteratius i ocupessin poca memòria. El resultat és que ja no tenim cap tipus de problema al fer una operació aritmètica, encara que s'hagi incrementat una mica el temps de càlcul, cosa que no importa ja que la diferència és despreciable.

##### Ús de strings com a paràmetre:

Quan teniem un string i l'haviem d'imprimir per pantalla, ens veiem obligats a fer un for per passar cada caràcter al lcd amb `lcd_putc(char c)`. Vam trobar al manual del compilador que podem passar un string directament a qualsevol funció que rebés un caràcter per paràmetre i el compilador se n'encarregaria de fer el bucle corresponent.

##### Macro `printf_xy`:

Per no haver d'escriure cada cop `lcd_gotoxy` vam crear una macro que es diu `printf_xy` amb la següent definició:

```
#define printf_xy(x,y,s)  { lcd_gotoxy(x,y); lcd_putc(s); }
```

##### Us de funcions inline:

Hem utilitzant bastant el modificador inline per fer que algunes funcions molt curtes no es comportessin com a funcions sinó que es copiés el codi d'aquestes a on correspongués. Això ho hem fet per evitar excessives instruccions de crida i perdre temps en salvar context i saltar d'un lloc a un altre. Hem tingut algun descuid en el que la funció inline no era petita i la utilitzàvem molts cops, fent que la ROM s'omplís fins al 100%, ho hem arreglat eliminant l'inline.

##### Paràmetre `uint16_t`:

Ens hem definit un nou tipus al fitxer `constants.h` que ens ha permès fer més llegible el codi, un `uint16_t` és un unsigned int de 16 bits.

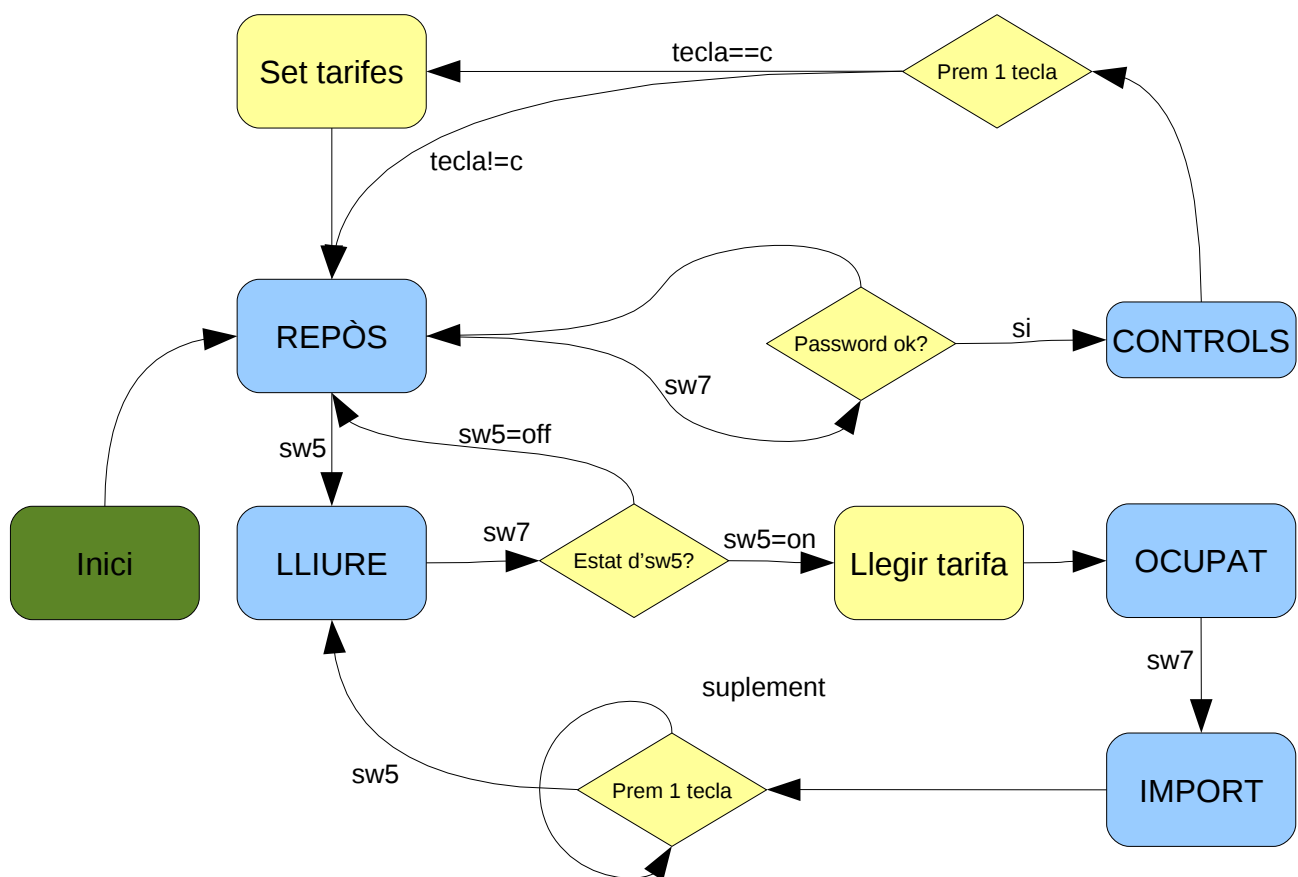
##### Variables mida bit:

El manual del compilador ens va revelar que si teniem varies variables que només havien d'ocupar un bit, les podem declarar i ell mateix s'encarregava d'ajuntar-les en un sol byte, per això, per optimitzar l'espai i la memòria RAM hem declarat tot el que són flags com a short (1bit).

##### Modificació de `teclat_lab.c`:

`teclat_lab.c` no guardava tots els registres que modificava i se'ns generaven interrupcions indesitjades. Vam haver de modificar el codi guardant i posteriorment restaurant tot el que podia generar un comportament erroni al sortir de `teclat_lab`. El codi corregit es pot trobar al mateix `teclat_lab.c`

## 5. Diagrama de flux del programa



## 6. Codi Font

A continuació presentem el codi font del programa:  
fitxers constants.h i main.c.

## 7. Bibliografia

<http://www.google.com>

<http://www.microchip.com>

PIC16F87x DataSheet

PICmicro™ Mid-Range MCU Family Reference Manual

<http://www.ccsinfo.com>

CCS Manual

Manual Pic Laboratory

## Llicència del Taximetre GPLv3

This file is part of Taximetre SDMI12A - Q1 - 2008-09.

"Taximetre SDMI12A - Q1 - 2008-09" is free software:  
you can redistribute it and/or modify it under the terms  
of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.

Taximetre is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with Taximetre. If not, see <<http://www.gnu.org/licenses/>>.