

UPC

# SDMI

---

## Pràctica Final

Autors:

Alejandro Marquez

Miquel Perelló

Jordi Piqueras

**02/06/2010**

## Contingut

1. Descripció detallada de les funcionalitats implementades .....	3
2. Diagrama de flux .....	5
2.1. Control del rellotge .....	6
2.2. Control del Contacte .....	7
2.3. Control LEDs.....	8
2.4. Control LCD .....	9
2.5. Control Teclat .....	10
2.6. Interrupcions .....	11
3. Problemes trobats .....	12
4. Codi comentat .....	13

# 1. Descripció detallada de les funcionalitats implementades

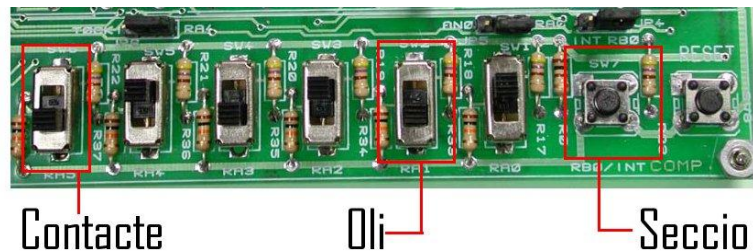
- **Funcionalitats del sistema**

La pràctica consisteix en implementar el panell de control d'una moto amb les següents funcionalitats:

- El sistema permetrà visualitzar a través de un lcd diferents dades que s'aniran alternant premen un botó.

- Un led indicarà el nivell de l'oli.

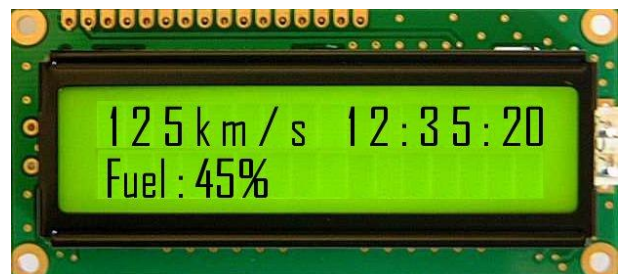
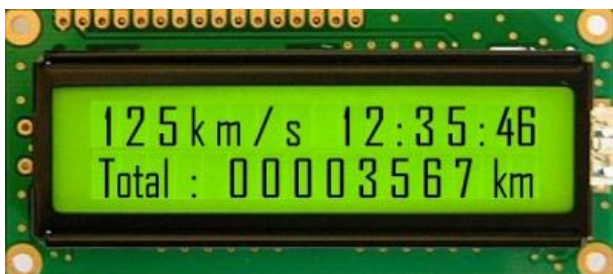
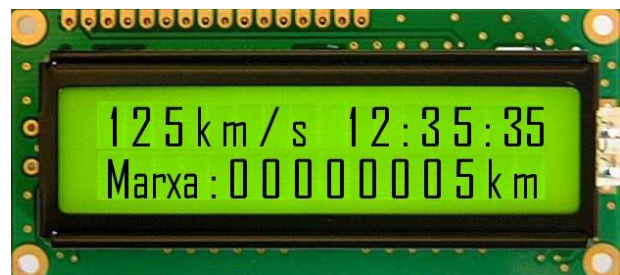
- Al engegar el contacte es mostraran tots els leds encesos durant 5 segons.



- El sistema ha de permetre canviar l'hora: el canvi d'hora s'inicia prement # A continuació s'introdueix l'hora amb el teclat numèric, si és correcte s'actualitza.

Dades a mostrar al lcd:

- Velocitat instantània
- Hora actual
- Velocitat mitjana des de la posta en marxa
- Kilòmetres recorreguts des de la posta a zero
- Kilòmetres totals recorreguts
- Nivell de combustible
- Estimació dels kilòmetres



- **Millores incorporades**

- Hem fet que el sistema mostri sempre la velocitat instantània i l'hora i que l'alternança de dades es produeixi només entre la resta de dades.
- Quan el nivell de benzina és inferior al 10% s'activa un led de reserva.
- Quan s'activa el contacte, un símbol \* es mou per la pantalla per comprovar que funciona correctament tot el panell lcd.

- **Detall del càlcul**

Càlcul de la distància en funció de la velocitat

Cada segon calculem la distància recorreguda en aquell segon.

$$\text{Distància(en mm)} = (\text{velocitat} * 1000) / 3.6$$

Anem acumulant la distància i quan arriba a 1km augmentem el còmput de kilòmetres i resetejem la variable que guarda els mil·límetres.

Càlcul de la benzina restant

Obtenim un valor analògic entre 0 i 255 del potenciòmetre P1.

Si ho normalitzem i ho multipliquem per 100 obtenim el percentatge de quantitat de benzina.

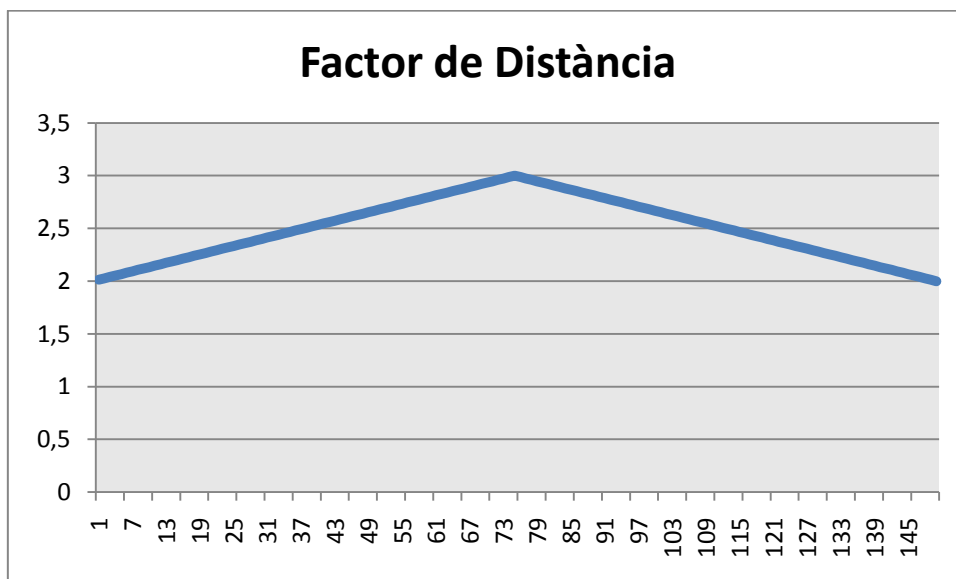
$$\text{Benzina(\%)} = (\text{ValorP1} / 255) * 100$$

Càlcul de l'autonomia

El càlcul de l'autonomia el fem a partir del nivell de benzina i la velocitat mitjana.

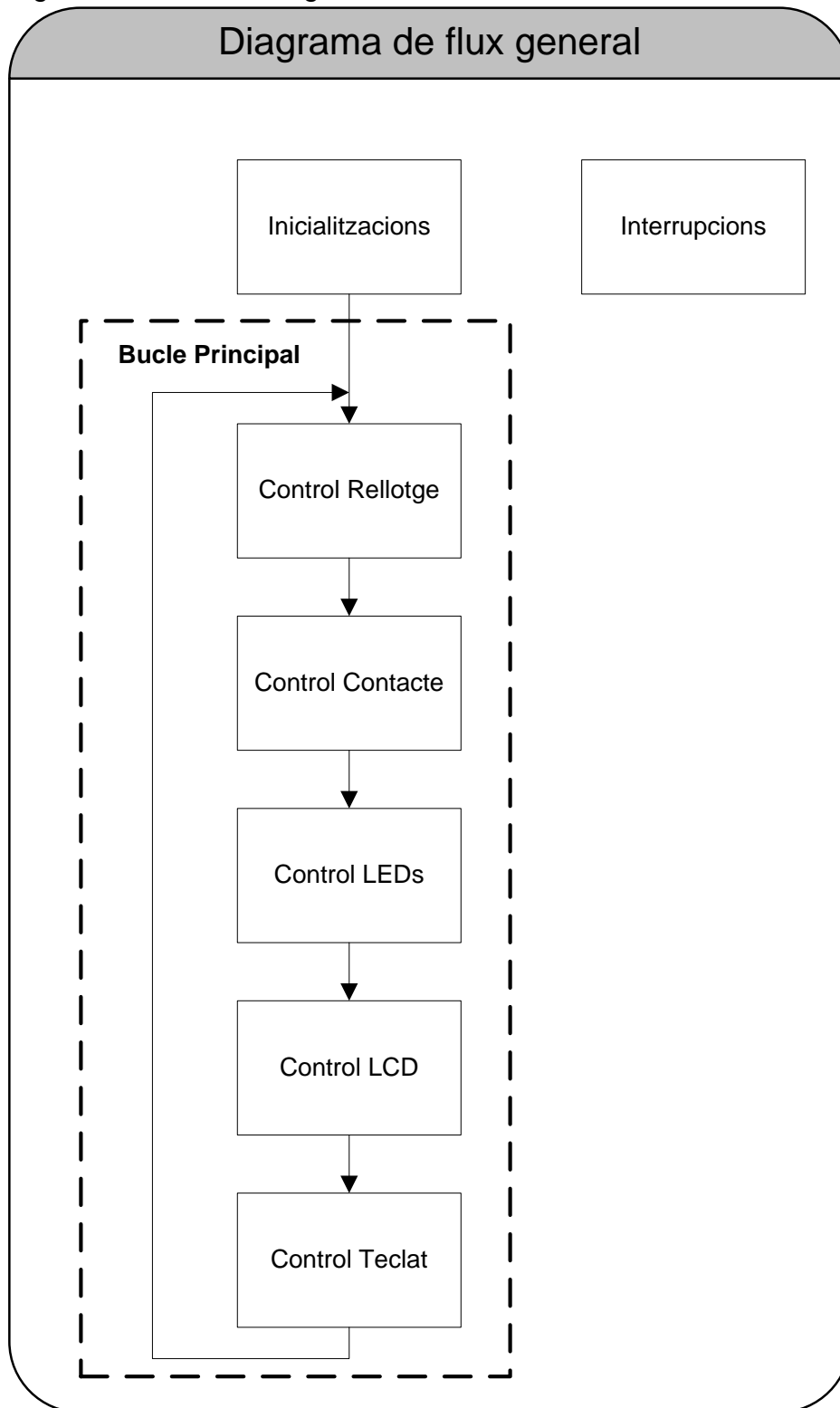
De la velocitat mitjana obtenim un factor de distància (més petit en els límits i més gran al centre) que multiplicat per el percentatge de benzina ens dóna la distància que encara es pot recórrer.

$$\text{Distància} = \text{benzina(\%)} * \text{factor\_distancia(vel\_mitjana)}$$



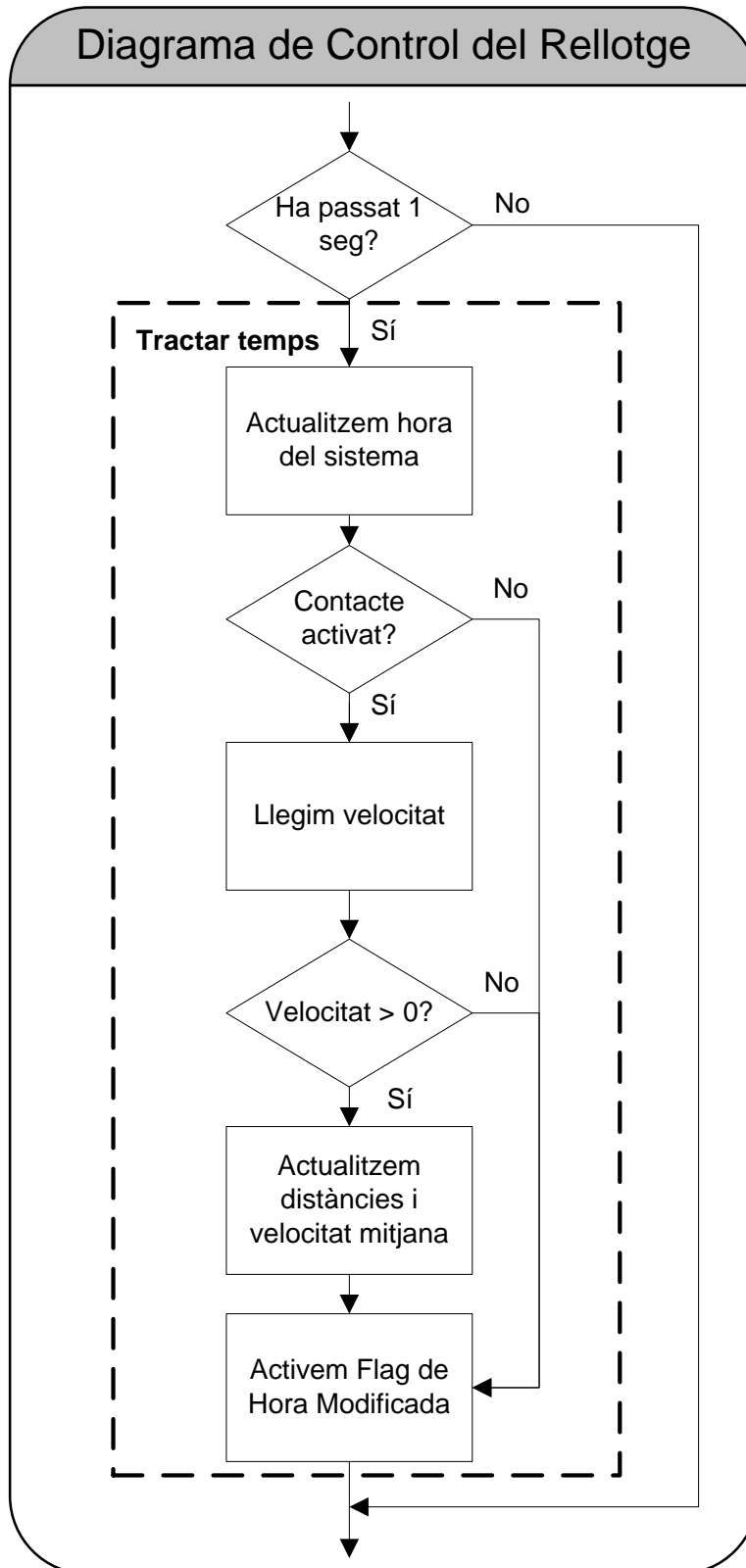
## 2. Diagrama de flux

El següent diagrama mostra el flux general sense entrar en detalls:



## 2.1. Control del rellotge

En el bloc de control del rellotge cada cop que passa un segon es miren els canvis en els sensors que depenen del temps i es recalculen les variables del sistema.

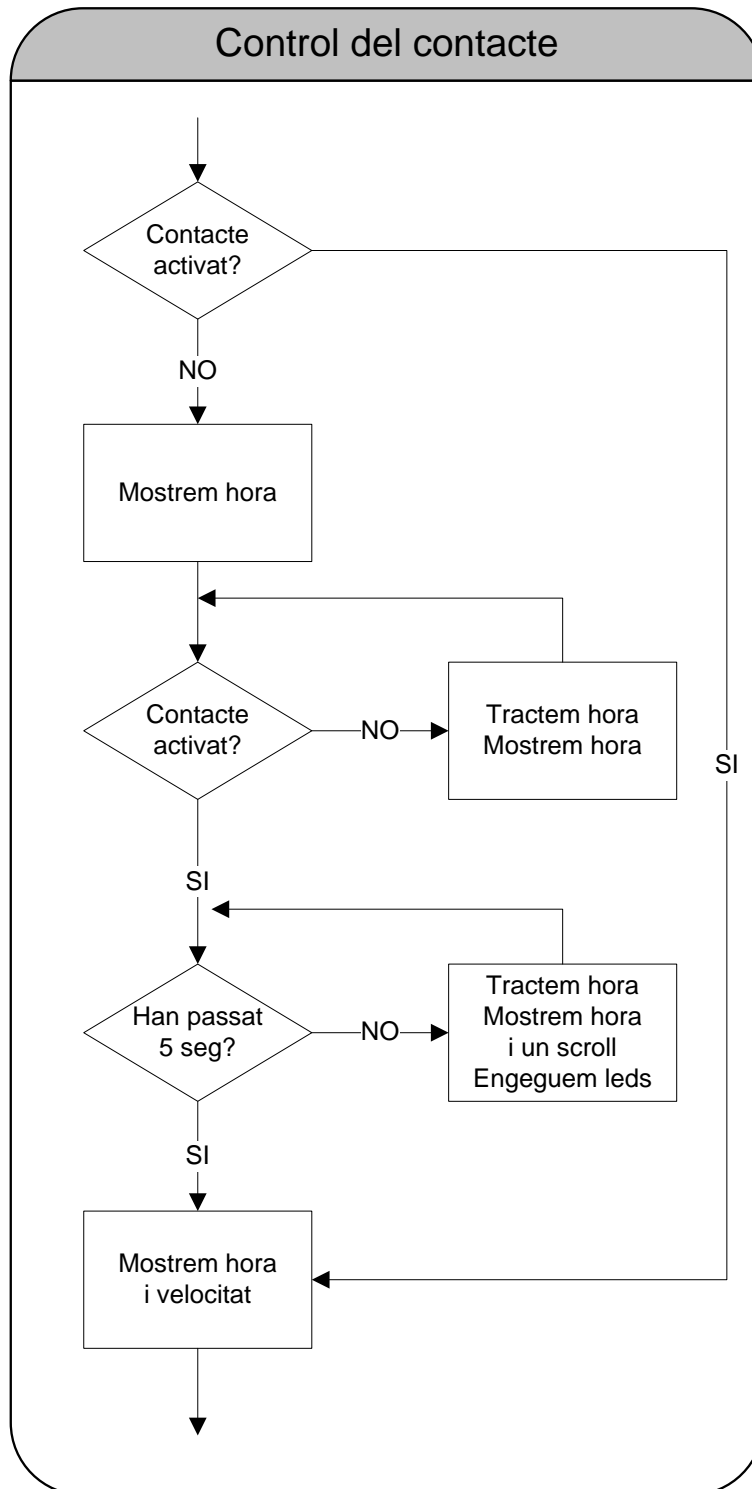


## 2.2. Control del Contacte

En el bloc de control de contacte controlem quan el contacte està apagat.

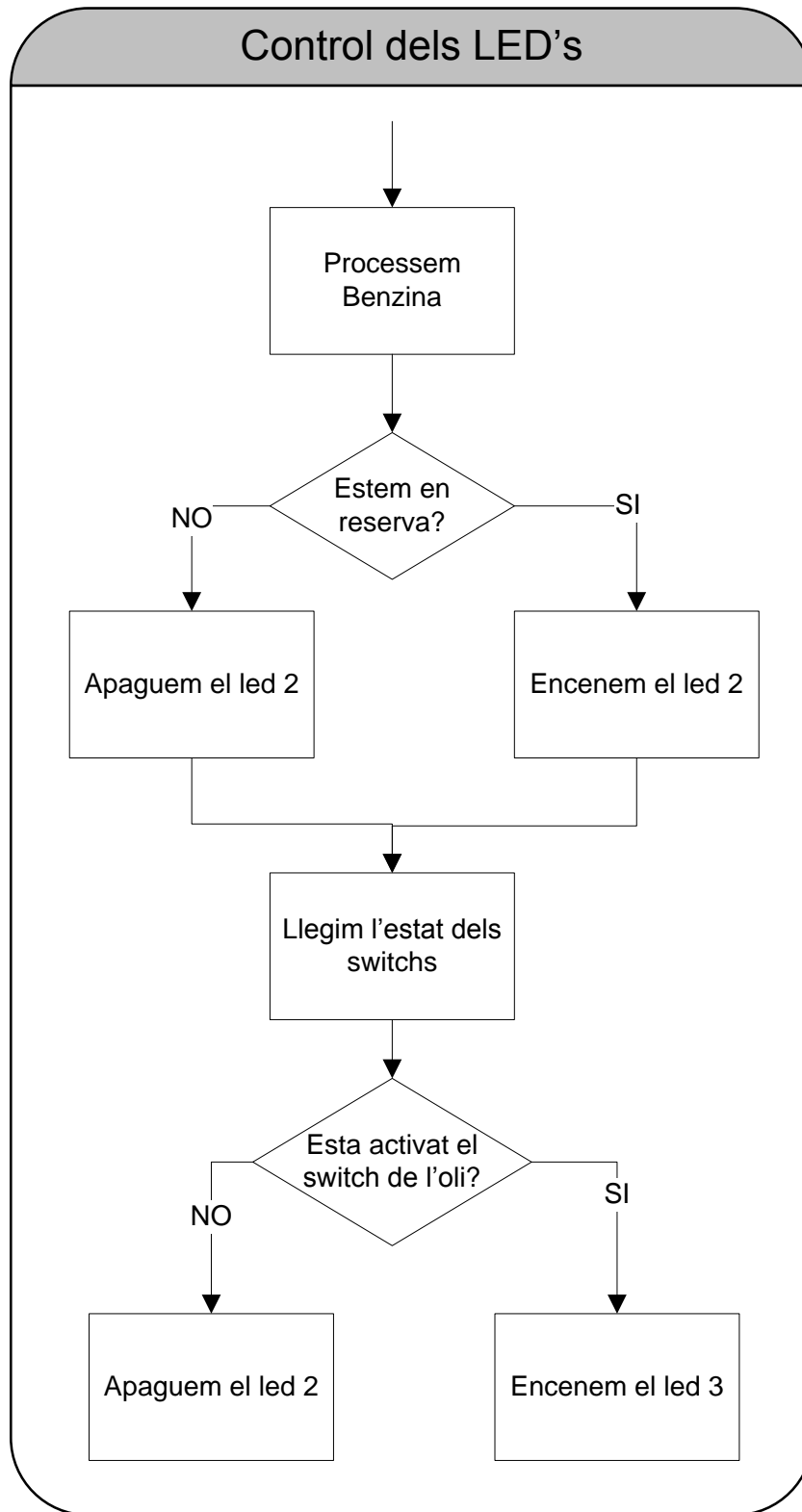
En aquest cas l'aplicació queda en bucle actualitzant el rellotge.

Al activar el contacte s'activen els leds i un scroll a la pantalla durant 5 segons i segueix el flux.



## 2.3. Control LEDs

En el bloc de Control de LEDs s'activen, si correspon, els leds d'oli i de reserva.

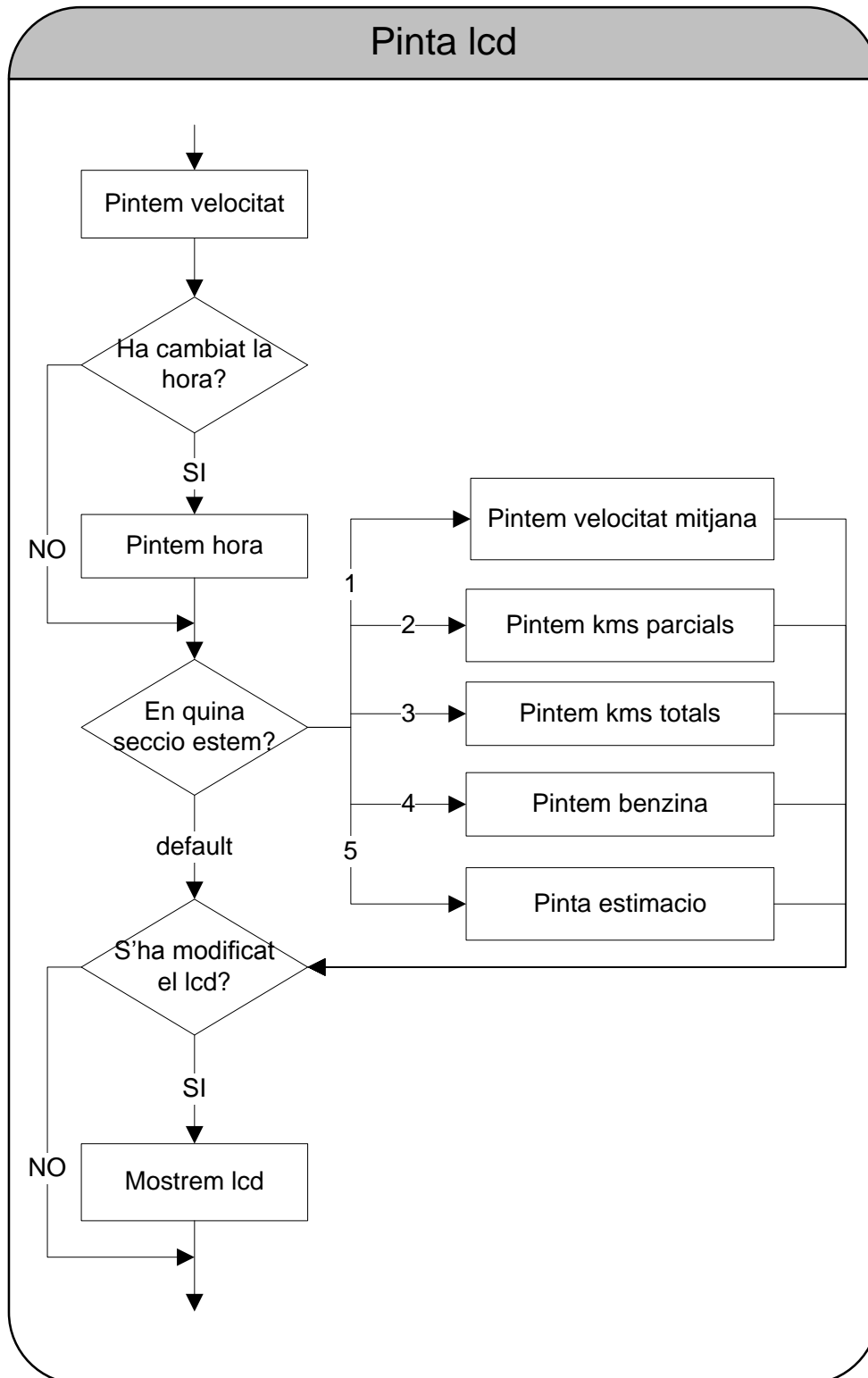




## 2.4. Control LCD

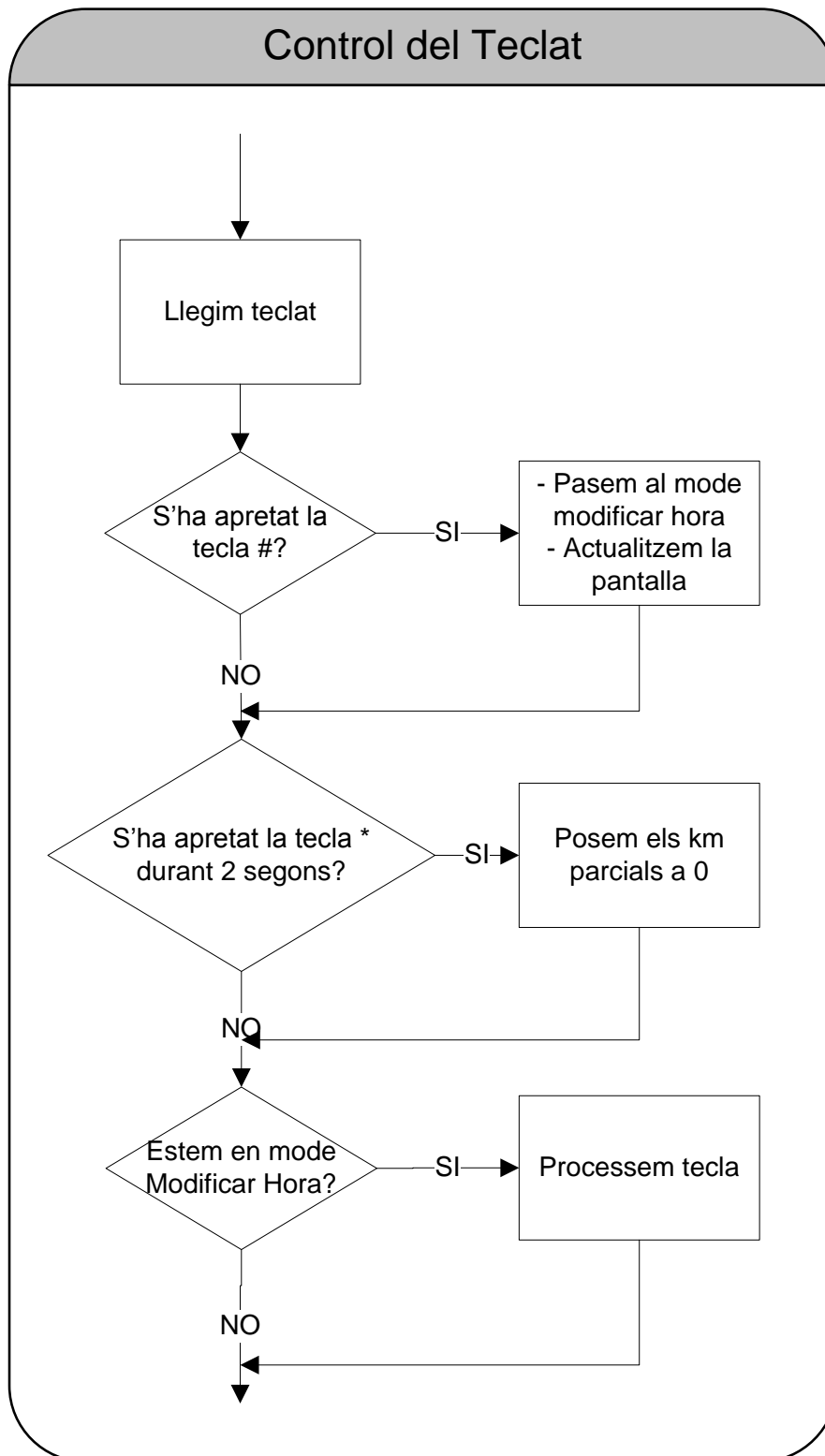
En el bloc de Control del LCD actualitzem les dades de la pantalla LCD.

Primer s'actualitza la hora si ha canviat i després la dada corresponent a la secció seleccionada actualment.



## 2.5. Control Teclat

En el bloc de control de teclat llegim el teclat per a resetejar la distància parcial o fer el canvi d'hora.

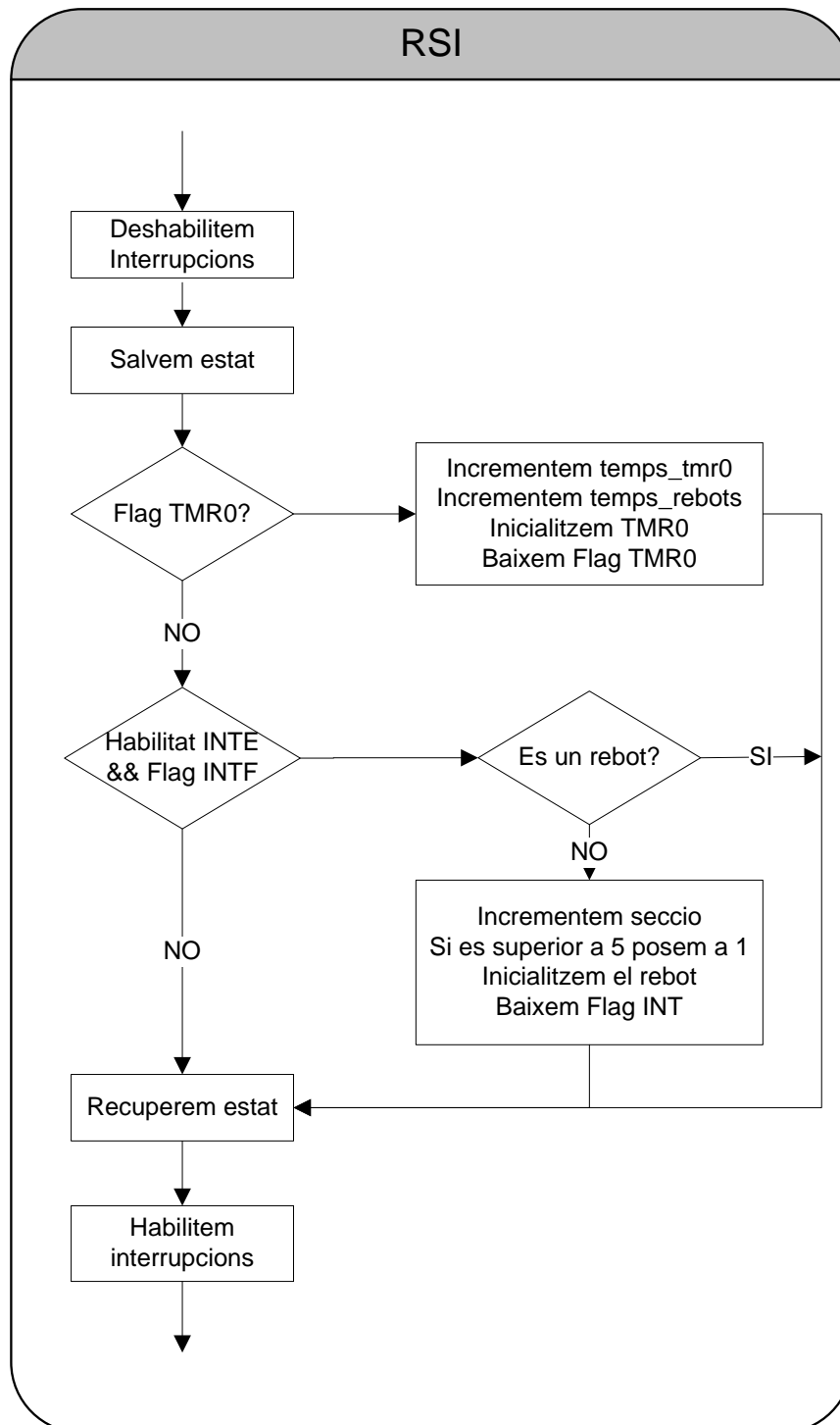


## 2.6. Interrupcions

En el bloc de control de les interrupcions processem les dades del timer 0 i la interrupció externa.

Després de gravar l'estat, comprovem quina de les dues interrupcions ha llançat el bloc.

Si és el tmr0 incrementem els comptadors de temps temporals i si és la interrupció externa actualitzem la secció actual.



### 3. Problemes trobats

Aquests són alguns dels problemes que ens hem trobat:

1. El `read_eeprom` i el `write_eeprom` tracten només un byte.  
Necessitàvem gravar 32bits per tant hem hagut de fer 4 lectures i 4 escriptures.
2. Al superar el 25% de la memòria del microcontrolador començava a fallar tot.  
Resulta que no guardàvem correctament l'estat al saltar les interrupcions. Hem hagut d'inserir en la RSI codi en ensamblador que guardi l'estat.
3. Al tractar el Timer 0 resetejarem els polsos que hi havia acumulats i per tant podia ser que perdéssim precisió en el càlcul del temps.  
Ho hem resolt fent el mòdul 100 dels polsos acumulats.
4. Al processar la distància recorreguda en un segon com a metres enters (ignorant la part decimal) perdiem part important de la informació sobre la distància.  
Ara recollim la informació de distancia recorreguda per segon en mil·límetres de manera que ja no hi ha pèrdua significativa.

## 4. Codi comentat

```
#include <16F876_CCS.h>

#include <lcd_lab.c>
#include <teclat_lab.c>
//Definicions dels registres i valor inicial de la memoria
#org 0x1F00, 0x1FFF void loader16F876(void) {} //protect bootloader code for the
8k 16F876/7
#BYTE PCLATH = 0x0A
#BYTE STATUS = 0x03
#BYTE W= 0x00
#org 0x0070, 0x0070 int W_TEMP; // Necesaris per el codi en ASM de la RSI
#org 0x0071, 0x0071 int PCLATH_TEMP; // Necesaris per el codi en ASM de la RSI
#org 0x0072, 0x0072 int STATUS_TEMP; // Necesaris per el codi en ASM de la RSI

// CONSTANTS
char linial[16];
char linia2[16];
char lcdModificat;
char seccio; // 1: VelInstant 2: VelMitj 3: KmParc 4: kmTotal
// 5: Benzina 6: Estim 7: Hora

char benzina;
char velocitat;
char velocitat_diferent;
int temps_tmr0;
int32 tempsMarxa;
int32 tempsMarxaAux;
int32 metresMarxa; //en mil·limetres
int32 kmsTotals;
int32 kmsParcials;
char modificant_hora;
long rebot;
long rebotINTE;
char seccio_anterior;
char horaModificada;
int32 suma_vel;
long mostres_vel;
int pasCanviHora;
char flagSleep;
// FI CONSTANTS

// INTERRUPCIONS
#int_global
void int_rsi(){
GIE = 0;

#asm
MOVWF W_TEMP //Copy W to TEMP register
SWAPF STATUS,W //Swap status to be saved into W
CLRF STATUS //bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF STATUS_TEMP //Save status to bank zero STATUS_TEMP register
MOVF PCLATH, W //Only required if using pages 1, 2 and/or 3
MOVWF PCLATH_TEMP //Save PCLATH into W
CLRF PCLATH //Page zero, regardless of current page
#endasm

if (TMR0IF == 1){
    temps_tmr0++;
    rebot++;
    rebotINTE++;
    TMR0 = ~195;
    TMR0IF = 0;
    OPTION_REG = 0x07;
```

```

}
else if (INTE==1 && INTF == 1) {
    if (rebotINTE > 20){
        seccio++;
        if (seccio >= 6)
            seccio = 1;
        rebotINTE = 0;
    }
    INTF = 0;
}

#asm
MOVF PCLATH_TEMP, W    //Restore PCLATH
MOVWF PCLATH           //Move W into PCLATH
SWAPF STATUS_TEMP,W   //Swap STATUS_TEMP register into W
//(sets bank to original state)
MOVWF STATUS           //Move W into STATUS register
SWAPF W_TEMP,F        //Swap W_TEMP
SWAPF W_TEMP,W        //Swap W_TEMP into W
#endasm

GIE = 1;
}
// FI INTERRUPCIONES

// FUNCIONS
void interrupciones_init();
void porta_init();
void portb_init();
void pinta_lcd();
void neteja_linia(char linia, char x, char y);
void pinta_benzina();
void pinta_hora(int32 segons);
void pinta_velocitat();
void pinta_velocitat_mitjana();
void pinta_estimacio();
void pinta_kms_parcials();
void pinta_kms_totals();
void processar_benzina();
void tractem_tmr0();
void pinta_oli();
int32 llegeixKmTotals();
void escriuKmTotals(int32 valor);
void pinta_start(int i);
// FI FUNCIONS

void main( void)
{
    int i = 0;
    char lectura;
    int32 novaHora;
    int32 nousMinuts;
    int32 tmpMinuts;
    int32 nouTempsMarxa;
    char flagRebot;
    lcdModificat = 0;
    benzina = 0;
    tempsMarxa = 0;
    seccio = 4;
    seccio_anterior = 0;
    temps_tmr0 = 100;
    velocitat = 0;
    velocitat_diferent = 1;
    modificant_hora = 0;

```

```

metresMarxa = 0;
kmsTotals = 0;
kmsParcials = 0;
horaModificada = 1;
suma_vel = 0;
mostres_vel = 0;
flagRebot = 0;
flagSleep = 0;

kmsTotals = llegeixKmTotals();

//Init
lcd_init();
porta_init();
portb_init();
interrupciones_init();

GO = 1;
//Main Loop
while(1)
{
/*****
/** DETECCIO 1 SEGON *****/
/*****
    if (temps_tmr0 >= 100){
        tractem_tmr0();
    }
/*****
/** FI DETECCIO 1 SEGON *****/
/*****

/*****
/** CONTROL DEL CONTACTE *****/
/*****

    porta_init();
    if (PORTA_5 == 0)
    {
        flagSleep=1;
        //Netejem pantalla
        neteja_linia(1, 0, 7);
        neteja_linia(2, 0, 15);
        pinta_lcd();
        while (PORTA_5 == 0)
        {
            if (temps_tmr0 >= 100){
                tractem_tmr0();
            }
            if (horaModificada == 1)
            {
                pinta_hora(tempsMarxa);
                pinta_lcd();
            }
            porta_init();
        }
        //Esperem 5 segons
        tempsMarxaAux = tempsMarxa + 5;
        i = 0;
        while (tempsMarxaAux>=tempsMarxa)
        {
            pinta_start(i++);
            if (temps_tmr0 >= 100){
                tractem_tmr0();
            }
            if (horaModificada == 1)
            {

```

```

        pinta_hora(tempsMarxa);
    }
    pinta_lcd();
    //LEDS!
    TRISB = 0x01;
    PORTB = 0xFE;
    delay_ms(100);
}
//Reengeguem la moto
flagSleep=0;
seccio = 4;
seccio_anterior = 0;
pinta_velocitat();
}
/*****
** FI CONTROL DEL CONTACTE *****/
/*****

/*****
** CONTROL BENZINA *****/
/*****
    if (ADIF == 1) processar_benzina();
    TRISB = TRISB && 0x7F;
    PORTB_2 = (benzina < 10);          // INDIQUEM SI ESTEM EN RESERVA
/*****
** FI CONTROL BENZINA *****/
/*****

/*****
** CONTROL DE L'OLI *****/
/*****
    porta_init();
    if (PORTA_1 == 1)
    {
        TRISB = 0x01;
        PORTB_3 = 1; //PORTB = 0x08;
    }
    else
    {
        TRISB = 0x01;
        PORTB_3 = 0; //PORTB = 0x00;
    }
/*****
** FI CONTROL DE L'OLI *****/
/*****

/*****
** MODIFICACIO VECTOR LCD *****/
/*****
////////////////////////////////////// LINIA 1 ////////////////////////////////////////
    pinta_velocitat();
    if (horaModificada == 1)
    {
        if (modificant_hora == 1)
            pinta_hora(nouTempsMarxa);
        else
            pinta_hora(tempsMarxa);
    }
////////////////////////////////////// FI LINIA 1 ////////////////////////////////////////

////////////////////////////////////// LINIA 2 ////////////////////////////////////////
    switch(seccio)
    {
        case 1:                // VELOCITAT MITJANA
            pinta_velocitat_mitjana();
            break;

```



```

        case 2:                                // KILOMETRES PARCIALS
            pinta_kms_parcial();
            seccio_anterior = seccio;
            break;
        case 3:                                // KILOMETRES TOTALS
            pinta_kms_totals();
            seccio_anterior = seccio;
            break;
        case 4:                                // BENZINA
            pinta_benzina();
            seccio_anterior = seccio;
            break;
        case 5:                                // ESTIMACIO
            pinta_estimacio();
            seccio_anterior = seccio;
            break;
    }
    //////////////////////////////////////////////////////////////////// FI LINIA 2 ////////////////////////////////////////////////////////////////////
    /*******
    /** FI MODIFICACIO VECTOR LCD *****/
    /*******

    /*******
    /** MOSTRAR PER LCD *****/
    /*******

    if (lcdModificat == 1){
        pinta_lcd();
        lcdModificat = 0;
    }
    /*******
    /** FI MOSTRAR PER LCD *****/
    /*******

    /*******
    /** TECLA CANVI D'HORA *****/
    /*******

    INTE = 0;
    lectura = keyScan();
    if (lectura == 0x80) //Si no s'apreta no contem rebot
    {
        rebot=0;
        flagRebot = 0;
    }

    if (rebot > 50 && lectura == '#'){ // he afegit la comprobacio del rebot
        modificant_hora = (modificant_hora+1)%2;
        pasCanviHora = 1;
        novaHora = 0;
        nousMinuts = 0;
        rebot = 0;
        nouTempsMarxa = 0;
        pinta_hora(nouTempsMarxa); // Al canviar la hora es modifica sol
        pinta_lcd(); // al modificarse sol ja es pinta
    }

    INTF = 0;
    INTE = 1;
    /*******
    /** FI TECLA CANVI D'HORA *****/
    /*******

    /*******
    /** RESET KMS PARCIALS *****/
    /*******

    if (seccio == 2 && rebot > 200 && lectura == '*') // en principi ha d'estar
    pulsat 2 segons
    {

```

```

        kmsParcials = 0;
        rebot = 0;
    }
    /*****
    /** FI RESET KM PARCIALS *****/
    /*****/

    /*****
    /** CANVI HORA *****/
    /*****/
    if (modificador_hora == 1 && rebot > 5 && flagRebot==0)
    {
        if (lectura>='0' && lectura<='9')
        {
            //Recollim el digit alla on toqui
            if (pasCanviHora==1)
            {
                novaHora = (lectura-0x30) * 10; //Decenes de hora
                nouTempsMarxa = (int32)((novaHora) * 3600);
            }
            else if (pasCanviHora==2)
            {
                novaHora += (lectura-0x30); //Unitats de hora
                nouTempsMarxa = (int32)(novaHora * 3600);
            }
            else if (pasCanviHora==3)
            {
                nousMinuts = (lectura-0x30) * 10; //Decenes de minut
                nouTempsMarxa = nouTempsMarxa + (nousMinuts * 60);
            }
            else if (pasCanviHora==4)
            {
                tmpMinuts = (lectura-0x30); //Unitats de minut
                nousMinuts += (lectura-0x30); //Unitats de minut
                nouTempsMarxa = nouTempsMarxa + (tmpMinuts * 60);
            }
            pinta_hora(nouTempsMarxa);
            pinta_lcd();
            //Actualitzem el pas perquè el digit següent es grabi on toca
            pasCanviHora++;
        }
        if (pasCanviHora==5) //Ja hem entrat els 4 números
        {
            //Validem hores i minuts
            if (novaHora>=0 && novaHora <= 23 && nousMinuts>=0 &&
nousMinuts <= 59)
            {
                //Calculem nova hora i la refresquem
                tempsMarxa = nouTempsMarxa;
                pinta_hora(tempsMarxa);
            }
            pasCanviHora = 0;
            modificador_hora = 0; //Tant si ha anat bé com si no deixem
d'estar en mode edició
        }
        rebot = 0;
        flagRebot = 1;
    }
    /*****
    /** CANVI HORA *****/
    /*****/
}
}

void porta_init()
{

```

```

    ADCON1 = 0x0E;                // LEFT JUSTIFIED && RAX: DDDDDDDA
    TRISA = 0xFF;                 // RA0-RA7 entradas
    ADCON0 = ADCON0 | 0x81; //10000001 // Fosc=32 && CHS=RA0 && GO=0 && ADON=1
    ADCON0 = ADCON0 & 0x85; //10000101
}

void portb_init()
{
    TRISB = 0xFF;
}

void interrupciones_init()
{
    T1CON = 0x02;                // prescale 00, externa i apagada
    TMR1H = 0x00;
    TMR1L = 0x00;
    TMR1ON = 1;                 // encenem TMR1

// INTCON
//GIE = 1;                      // Global Interrupt Al final l'habilitem
PEIE = 1;                      // Peripheral Interrupt
TMR0IE = 1;                     // Overflow Interrupt TMR0
INTE = 1;                      // Interrupcions RB0
RBIE = 0;                      // Interrupcions port RB
// FI_INTCON
// OPTION_REG 0x07
RBPUR = 0;
INTEDG = 0;                    //Interrupcions RB0 en flanc de baixada
TOCS = 0;                      // Internal instruction cycle clock
TOSE = 0;
PSA = 0;
PS0 = 1;                      // Prescaler
PS1 = 1;                      // Prescaler
PS2 = 1;                      // Prescaler
// FI_OPTION_REG
// PIE1
TMR1IE = 0;                    // desactivem TMR1
// FI_PIE1

    GIE = 1;                    // Global Interrupt
}

void pinta_lcd()
{
    int i;
    GIE = 0;
    for (i = 0; i < 16; i++){
        lcd_gotoxy(i,1);
        lcd_putc(linia1[i]);
        lcd_gotoxy(i,2);
        lcd_putc(linia2[i]);
    }
    OPTION_REG = 0x07;
    GIE = 1;
}

void neteja_linia(char linia, char x, char y){
    while (x <= y){
        if (linia == 1)
            linia1[x] = 0x20;
        else
            linia2[x] = 0x20;
        x++;
    }
}

```

```

void pinta_velocitat(){
    int i;
    char auxChar;
    auxChar = velocitat;
    // Comprobem si ha canviat el valor o si entrem a la seccio per primer cop
    if (seccio != seccio_anterior || velocitat_diferent == 1){
        linial[3] = 'k';
        linial[4] = 'm';
        linial[5] = '/';
        linial[6] = 'h';
        linial[7] = 0x20;
        for (i = 2; i != -1; i--){
            linial[i] = auxChar%10 + 0x30;
            auxChar /= 10;
        }
        velocitat_diferent = 0;
        lcdModificat = 1;
    }
}

```

```

void pinta_hora(int32 segons)
{
    char hores, minuts;
    hores = segons/3600;
    segons = segons%3600;
    minuts = segons/60;
    segons = segons%60;
    linial[8] = (hores/10)%10 + 0x30;
    linial[9] = hores%10 + 0x30;
    linial[10] = ':';
    linial[11] = (minuts/10)%10 + 0x30;
    linial[12] = minuts%10 + 0x30;
    linial[13] = ':';
    linial[14] = (segons/10)%10 + 0x30;
    linial[15] = segons%10 + 0x30;
    if (modificant_hora == 1){
        switch(pasCanviHora)
        {
            case 1:
                linial[8] = '_';
                break;
            case 2:
                linial[9] = '_';
                break;
            case 3:
                linial[11] = '_';
                break;
            case 4:
                linial[12] = '_';
                break;
            default:
                break;
        }
    }
    lcdModificat = 1;
    horaModificada = 0;
}

```

```

void pinta_velocitat_mitjana(){
    int i;
    char auxChar;
    auxChar = suma_vel/mostres_vel;
    linia2[0] = 'v';
    linia2[1] = 'e';
    linia2[2] = 'l';
    linia2[3] = 0x20;
}

```

```

linia2[4] = 'm';
linia2[5] = 'i';
linia2[6] = 't';
linia2[7] = 'j';
linia2[8] = ':';
linia2[12] = 'k';
linia2[13] = 'm';
linia2[14] = '/';
linia2[15] = 'h';
for (i = 11; i >= 9; i--){
    linia2[i] = auxChar%10 + 0x30;
    auxChar /= 10;
}
lcdModificat = 1;
}

void processar_benzina(){
    long auxLong;

    TRISA = 0xFF;    // RA0-RA7 entradas
    //Processar dada
    auxLong = (long)ADRESH*100;
    benzina = auxLong/255;
    ADIF = 0;
}

void pinta_benzina(){
    char i, auxChar;
    auxChar = benzina;
    // Comprobem si entrem a la seccio per primer cop
    linia2[0] = 'F';
    linia2[1] = 'u';
    linia2[2] = 'e';
    linia2[3] = 'l';
    linia2[4] = ':';
    linia2[8] = '%';
    for (i = 7; i >= 5; i--) {
        linia2[i] = auxChar%10 + 0x30;
        auxChar /= 10;
    }
    neteja_linia(2, 9,15);
    lcdModificat = 1;
}

void pinta_start(int i)
{
    char j;
    for (j = 0; j < 15; j++)
        if (j == i%16) linia2[j] = '*';
        else linia2[j] = 0x20;
}

void pinta_estimacio(){
    char i;
    char VelMitj;
    long auxLong;
    float factor;

    VelMitj = suma_vel/mostres_vel;
    if (VelMitj<=75)
        auxLong = 75 - VelMitj;
    else
        auxLong = VelMitj - 75;

    factor = 3 - ((float)auxLong / (float)75);
}

```

```

auxLong = (long)((float)benzina*factor); // mirar de usar la velocidad
linia2[0] = 'E';
linia2[1] = 's';
linia2[2] = 't';
linia2[3] = 'i';
linia2[4] = 'm';
linia2[5] = ':';
linia2[9] = 'k';
linia2[10] = 'm';
for (i = 8; i > 5; i--){
    linia2[i] = auxLong%10 + 0x30;
    auxLong /= 10;
}
neteja_linia(2, 11,15);
lcdModificat = 1;
}

void pinta_kms_parcials(){
    char i;
    int aux;
    aux = kmsParcials;
    linia2[0] = 'M';
    linia2[1] = 'a';
    linia2[2] = 'r';
    linia2[3] = 'x';
    linia2[4] = 'a';
    linia2[5] = ':';
    linia2[14] = 'k';
    linia2[15] = 'm';
    for (i = 13; i > 5; i--){
        linia2[i] = aux%10 + 0x30;
        aux /= 10;
    }
    lcdModificat = 1;
}

void pinta_kms_totals(){
    char i;
    int32 aux;
    aux = kmsTotals;
    linia2[0] = 'T';
    linia2[1] = 'o';
    linia2[2] = 't';
    linia2[3] = 'a';
    linia2[4] = 'l';
    linia2[5] = ':';
    linia2[14] = 'k';
    linia2[15] = 'm';
    for (i = 13; i > 5; i--){
        linia2[i] = aux%10 + 0x30;
        aux /= 10;
    }
    lcdModificat = 1;
}

void tractem_tmr0(){
    tempsMarxa++;
    //Posem a 0 quan arriben les 12 de la nit
    if (tempsMarxa>86399) //((24*3600)-1)
    {
        tempsMarxa = 0;
    }
    temps_tmr0 = temps_tmr0%100; // amb el modul no ens menjem petits errors
    if (flagSleep==0)
    {
        if (velocitat != TMR1L)

```

```

        velocitat_diferent = 1;
    velocitat = TMR1L;
    if (velocitat>0)
    {
        suma_vel += velocitat;
        mostres_vel++;
    }
    // Guardem els metres recorreguts aquest segon, si poso 10/36 no se porque
no funciona
    metresMarxa += (velocitat*1000)/3.6;
    if (metresMarxa >= 1000000){
        kmsParcials++;
        kmsTotals++;
        escriuKmTotals(kmsTotals);
        metresMarxa = 0;
    }
    TMR1L = 0;
    TMR1H = 0;
    GO = 1;        // CADA SEGON PERMETEM FER UNA CONVERSIO A/D
}
horaModificada = 1;
}

int32 llegeixKmTotals()
{
    int32 valor;
    if (read_eeprom(0x00)==0xEE)
    {
        valor = read_eeprom(0x01);
        valor *= 256;
        valor += read_eeprom(0x02);
        valor *= 256;
        valor += read_eeprom(0x03);
        valor *= 256;
        valor += read_eeprom(0x04);
        return valor;
    }
    else
    {
        escriuKmTotals(0);
        write_eeprom(0x00, 0xEE);
        return 0;
    }
}

void escriuKmTotals(int32 valor)
{
    write_eeprom(0x04, valor%256);
    valor /= 256;
    write_eeprom(0x03, valor%256);
    valor /= 256;
    write_eeprom(0x02, valor%256);
    valor /= 256;
    write_eeprom(0x01, valor%256);
}

```