

```

// ANNEX 3: CODI DEL PROGRAMA //
#include "mcc_generated_files/mcc.h"
#include <pic18f45k22.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "flex_lcd420.c"
#include <delays.h>
#include <xc.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>
#include <stddef.h>

// ESTATS //
bit Menu_Pot = 0;
bit Menu_Vel = 0;
bit Menu_Test = 0;
bit Menu_Carrega = 0;
bit Test_OK = 0;
bit Test_Done = 0;
// CONTADORS //
unsigned char MP_cont = 1; //Menu Potencia
unsigned char MV_cont = 1; //Menu Velocitat
unsigned char MT_cont = 1; //Menu Test
unsigned char TS_cont = 1; //Testing
unsigned char MC_cont = 1; //Menu Carrega
unsigned char TC_cont = 1; //TimeingCarrega
unsigned char TC_i = 0;
unsigned char TS_i = 0;
// VARIABLES GLOBALS //
unsigned char Tipus_Bateria = 0;
unsigned char Tipus_Carrega = 0;
uint16_t I_ref = 0x0000;
unsigned char T_eval = 120; //Intervals (en sx2) d'evaluació
unsigned char T_test = 10; //Intervals (en sx2) de test
unsigned long AV_PWM = 2; //Increments del V_PWM
unsigned char E_Carrega = 0; //Estat de Carrega
//Lectures ADC //
uint16_t V_Convertidor = 0x0000;
uint16_t V_Bateria = 0x0000;
uint16_t T_Bateria = 0x0000;
//Increment de tensió per 1s de carrega
//Es considera 100 mV -> 0x0065
//Es considera 10 mV -> 0x000A
//Es considera 9 A -> 0xD2F0
uint16_t AI_Test = 0xD2F0;

//Vdc=a1*SOC(%1)+a
//SOC=Vdc/A - B (en%) )
unsigned char A = 138; // 38,4 V bateria descarregada
unsigned long B = 278; // 52 V bateria carregada
uint16_t I_Hall = 0x0000;
uint16_t I_Shunt = 0x0000;

void configura (void)
{

```

```

SYSTEM_Initialize(); // Initialize the device
lcd_init(); // Inicialitza la LCD

}

void __delay_s(unsigned char t)
{
    for(unsigned char i=0; i <= 40; i++)
    {
        __delay_ms(25);
    }
}

void LCD_string(char print[15], unsigned char j)
{
    for(unsigned char i=0; i <= 16; i++)
    {
        lcd_goto_xy(i,j);
        lcd_putc(print[i-1]);
    }
}

void LCD_int3(unsigned long a, unsigned char j)
//Permet escriure enters de 3 caracters
{
    char int_LCD[4];
    sprintf(int_LCD,"%3u", a);
    for(unsigned char i=0; i <= 3; i++)
    {
        lcd_goto_xy(i,j);
        lcd_putc(int_LCD[i-1]);
    }

}

void LCD_int5(unsigned long a, unsigned char j)
//Permet escriure enters de 5 caracters
{
    char int_LCD[5];
    sprintf(int_LCD,"%5u", a);
    for(unsigned char i=0; i <= 5; i++)
    {
        lcd_goto_xy(i,j);
        lcd_putc(int_LCD[i-1]);
    }

}

void LCD_Saludo(void)
//Salutació al inici del programa
{
    __delay_s(1);
    LCD_string("Oriol Cos Martí ",1);
    LCD_string("TFG - EEBE      ",2);
    __delay_s(40);
    __delay_s(40);
    LCD_string("Carregador Bat. ",1);
    LCD_string("LLC - 48 V      ",2);
    __delay_s(40);
    __delay_s(40);
    LCD_string("          ",1);
    LCD_string("          ",2);
}

```

```

}

void ADC_Actualitzar(unsigned char channel)
{
    ADC_SelectChannel(channel);      //Es selecciona el canal del ADC
    ADC_StartConversion();          //Començà la conversió
    while(!ADC_IsConversionDone()); //El programa no abançà fins que s'acaba
                                    // la conversió

    switch (channel)              //Amb la variable que es rep, la funció
                                  // selecciona on guarda el valor
    {
        case channel_AN1:
        {
            I_Shunt = ADC_GetConversionResult();
            break;
        }
        case channel_AN2:
        {
            I_Hall = ADC_GetConversionResult();
            break;
        }
        case channel_AN6:
        {
            V_Bateria = ADC_GetConversionResult();
            break;
        }
        case channel_AN7:
        {
            V_Convertidor = ADC_GetConversionResult();
            break;
        }
    }
}

void LCD_Menu_Inicial_Pot(unsigned char menu)
//Menu de selecció de la potència de la bateria
{
    switch (menu)
    {
        case 1:
        {
            LCD_string("Tipus Bateria  ",1);
            LCD_string("7 Ah          ",2);
            if(INTCON3bits.INT1IF == 1)      //Boto Down
            {
                MP_cont=2;
                EXT_INT1 InterruptFlagClear();
            }

            if(INTCON3bits.INT2IF == 1)      //Boto OK
            {
                Menu_Pot = 0;
                Menu_Vel = 1;
                Menu_Test = 0;
                Menu_Carrega = 0;
            }
        }
    }
}

```

```
        Tipus_Bateria = 70;
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
case 2:
{
    LCD_string("Tipus Bateria    ",1);
    LCD_string("20 Ah          ",2);
    if(INTCON3bits.INT1IF == 1)      //Boto Down
    {
        MP_cont=3;
        EXT_INT1_InterruptFlagClear();

    }

    if(INTCON3bits.INT2IF == 1)      //Boto OK
    {
        Menu_Pot = 0;
        Menu_Vel = 1;
        Menu_Test = 0;
        Menu_Carrega = 0;
        Tipus_Bateria = 200;
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
case 3:
{
    LCD_string("Tipus Bateria    ",1);
    LCD_string("50 Ah          ",2);
    if(INTCON3bits.INT1IF == 1)      //Boto Down
    {
        MP_cont=1;
        EXT_INT1_InterruptFlagClear();

    }

    if(INTCON3bits.INT2IF == 1)      //Boto OK
    {
        Menu_Pot = 0;
        Menu_Vel = 1;
        Menu_Test = 0;
        Menu_Carrega = 0;
        Tipus_Bateria = 500;
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
default:
{
    MP_cont=1;
    break;
}

}
void LCD_Menu_Inicial_Vel(unsigned char menu)
```

//Menu per seleccionar el tipus de carrega

```
{  
    switch (menu)  
    {  
        case 1:  
        {  
            LCD_string("Tipus Carrega    ",1);  
            LCD_string("10 % Pmax      ",2);  
            if(INTCON3bits.INT1IF == 1) //Boto Down  
            {  
                MV_cont=2;  
                EXT_INT1_InterruptFlagClear();  
  
            }  
  
            if(INTCON3bits.INT2IF == 1) // Boto OK  
            {  
                Menu_Pot = 0;  
                Menu_Vel = 0;  
                Menu_Test = 1;  
                Menu_Carrega = 0;  
                Tipus_Carrega = 10;  
                I_ref = ((uint16_t)Tipus_Carrega * Tipus_Bateria);  
                EXT_INT2_InterruptFlagClear();  
            }  
            break;  
        }  
        case 2:  
        {  
            LCD_string("Tipus Carrega    ",1);  
            LCD_string("15 % Pmax      ",2);  
            if(INTCON3bits.INT1IF == 1) //Boto Down  
            {  
                MV_cont=3;  
                EXT_INT1_InterruptFlagClear();  
  
            }  
  
            if(INTCON3bits.INT2IF == 1) // Boto OK  
            {  
                Menu_Pot = 0;  
                Menu_Vel = 0;  
                Menu_Test = 1;  
                Menu_Carrega = 0;  
                Tipus_Carrega = 15;  
                I_ref = ((uint16_t)Tipus_Carrega * Tipus_Bateria);  
                EXT_INT2_InterruptFlagClear();  
            }  
            break;  
        }  
        case 3:  
        {  
            LCD_string("Tipus Carrega    ",1);  
            LCD_string("20 % Pmax      ",2);  
            if(INTCON3bits.INT1IF == 1) //Boto Down  
            {  
                MV_cont=1;  
                EXT_INT1_InterruptFlagClear();  
            }  
    }  
}
```

```

    }

    if(INTCON3bits.INT2IF == 1) // Boto OK
    {
        Menu_Pot = 0;
        Menu_Vel = 0;
        Menu_Test = 1;
        Menu_Carrega = 0;
        Tipus_Carrega = 20;
        I_ref = ((uint16_t)Tipus_Carrega * Tipus_Bateria);
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
default:
{
    MV_cont=1;
    break;
}

}

void Test (unsigned char menu)
//Funció que implementa el test
{
    switch (menu)
    {
        case 1:
        {
            ADC_Actualitzar(channel_AN6);
            if ((V_Bateria > 35000) && (V_Bateria < 60000))
            {
                T_Bateria = V_Bateria;
                IO_RC3_SetLow();           //Arranquem el convertidor
                __delay_ms(10);
                IO_RA3_SetHigh();         //Tanquem el rele
                TMR0_Reload();
                TMR0_StartTimer();
                TS_cont = 2;

            }
            else
            {
                Test_Done = 1;
                Test_OK   = 0;
            }
            break;
        }
        case 2:
        {

            if(TMR0_HasOverflowOccured())
            {
                if (TS_i >= T_test)
                {
                    ADC_Actualitzar(channel_AN1);
                    IO_RA3_SetLow();           //Obrim el rele
                }
            }
        }
    }
}

```

```

        IO_RC3_SetHigh();           //Parem el convertidor
        TMR0IF = 0;
        TMR0_StopTimer();
        //__delay_ms(10);
        TS_cont = 3;
        TS_i = 0;
    }
    else
    {
        TS_i = TS_i + 1;
        TS_cont = 1;
        TMR0_StopTimer();
    }
}
break;
}
case 3:
{
    if (I_Shunt < AI_Test)
    {
        Test_Done = 1;
        Test_OK    = 1;
    }
    else
    {
        Test_Done = 1;
        Test_OK    = 0;
    }
    break;
}
default:
{
    TS_cont = 1;
    break;
}
}
}
}

void LCD_Menu_Test(unsigned char menu)
//Funció que representa els estats del Test
{
    switch (menu)
    {
        case 1:
        {
            LCD_string("Mode Test      ",1);
            LCD_string("...",2);
            Test(TS_cont);
            if (Test_Done == 1)
            {
                MT_cont = 2;
            }
            break;
        }
        case 2:
        {
            LCD_string("Mode Test      ",1);
            LCD_string("Finalitzat   ",2);
        }
    }
}

```

```

        __delay_s (5);
        if (Test_OK == 1)
        {
            MT_cont = 3;
        }
        else
        {
            MT_cont = 4;
        }
        break;
    }
    case 3:
    {
        LCD_string("Resultat Test    ",1);
        LCD_string("OK                  ",2);
        __delay_s (5);
        Menu_Pot = 0;
        Menu_Vel = 0;
        Menu_Test = 0;
        Menu_Carrega = 1;
        break;
    }
    case 4:
    {
        LCD_string("Resultat Test    ",1);
        LCD_string("Bateria No Apta ",2);
        __delay_s (20);
        __delay_s (20);
        LCD_string("Revisar Bateria ",1);
        LCD_string(" -> Reset      ",2);
        break;
    }
    default:
    {
        MT_cont = 1;
        break;
    }
}
}

void LCD_Menu_Carrega(unsigned char menu)
//Menu de representació durant la carrega
{
    switch (menu)
    {
        case 1:
        {
            ADC_Actualitzar(channel_AN1);
            LCD_string("Corrent Shunt   ",1);
            uint16_t mA_I_Shunt;
            uint32_t tmp = ((uint32_t)I_Shunt)*10;
            mA_I_Shunt = tmp/61;
            LCD_string("          mA      ",2);
            LCD_int5(mA_I_Shunt,2);
            if(INTCON3bits.INT1IF == 1) //Boto Down
            {
                MC_cont=2;
                EXT_INT1_InterruptFlagClear();
            }
        }
    }
}

```

```
        }
        break;
    }
    case 2:
    {
        ADC_Actualitzar(channel_AN2);
        LCD_string("Corrent Hall    ",1);
        uint16_t mA_I_Hall;
        uint32_t tmp = ((uint32_t)(I_Hall-6284)*10);
        mA_I_Hall = tmp/25;
        LCD_string("      mA      ",2);
        LCD_int5(mA_I_Hall,2);
        if(INTCON3bits.INT1IF == 1) //Boto Down
        {
            MC_cont=3;
            EXT_INT1_InterruptFlagClear();

        }
        break;
    }
    case 3:
    {
        ADC_Actualitzar(channel_AN6);
        LCD_string("Tensio Bat.    ",1);
        uint16_t mV_V_Bateria = V_Bateria;// - 64;
        LCD_string("      mV      ",2);
        LCD_int5(mV_V_Bateria,2);
        if(INTCON3bits.INT1IF == 1) //Boto Down
        {
            MC_cont=5;
            EXT_INT1_InterruptFlagClear();

        }
        break;
    }
    case 4:
    {
        ADC_Actualitzar(channel_AN7);
        LCD_string("Tensio Conv.    ",1);
        uint16_t mV_V_Convertidor = V_Convertidor;// - 64;
        LCD_string("      mV      ",2);
        LCD_int5(mV_V_Convertidor,2);
        if(INTCON3bits.INT1IF == 1) //Boto Down
        {
            MC_cont=5;
            EXT_INT1_InterruptFlagClear();
        }
        EXT_INT2_InterruptFlagClear();
        break;
    }
    case 5:
    {
        LCD_string("Estat de Carrega",1);
        LCD_string("      %      ",2);
        LCD_int3(E_Carrega,2);
        if(INTCON3bits.INT1IF == 1) //Boto Down
        {
```

```

        MC_cont=6;
        EXT_INT1_InterruptFlagClear();
    }
    EXT_INT2_InterruptFlagClear();
    break;
}
case 6:
{
    LCD_string("Finalitzar      ",1);
    LCD_string("Carrega        ",2);
    if(INTCON3bits.INT1IF == 1) //Boto Down
    {
        MC_cont=1;
        EXT_INT1_InterruptFlagClear();

    }

    if(INTCON3bits.INT2IF == 1) // Boto OK
    {
        Menu_Pot = 1;
        Menu_Vel = 0;
        Menu_Test = 0;
        Menu_Carrega = 0;
        Test_OK = 0;
        MC_cont=1;
        IO_RC3_SetHigh();           //Parem el convertidor
        IO_RA3_SetLow();           //Desconectem bateria
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
case 7:
{
    LCD_string("Carrega      ",1);
    LCD_string("Finalitzada  ",2);
    if(INTCON3bits.INT2IF == 1) // Boto OK
    {
        Menu_Pot = 1;
        Menu_Vel = 0;
        Menu_Test = 0;
        Menu_Carrega = 0;
        Test_OK = 0;
        MC_cont=1;
        IO_RC3_SetHigh();           //Parem el convertidor
        IO_RA3_SetLow();           //Desconectem bateria
        EXT_INT2_InterruptFlagClear();
    }
    break;
}
default:
{
    MC_cont=1;
    break;
}

}
void Time_Carrega(unsigned char menu)

```

```
//Finció que gestiona la interrupcio i
// evaluació de la carrega
{
    switch (menu)
    {
        case 1:
        {
            TMR0_Reload();           ///
            TMR0_StartTimer();
            TC_cont = 2;
            I0_RC3_SetLow();        // Arranquem el convertidor.
            I0_RA3_SetHigh();       // Tanquem el rele
            break;
        }
        case 2:
        {
            if(TMR0_HasOverflowOccured())
            {
                if (TC_i >= T_eval)
                {
                    TC_cont = 3;
                    TC_i = 0;
                }
                else
                {
                    TC_i = TC_i + 1;
                    TC_cont = 1;
                    TMR0_StopTimer();
                }
                TMR0IF = 0;
            }
            break;
        }
        case 3:
        {
            ADC_Actualitzar(channel_AN1); //Corrent Shunt
            if (I_Shunt < I_ref)
            {
                TC_cont = 4;          // Estat per incrementar PWM
            }
            I0_RA3_SetLow();           // Obrim el rele
            __delay_s(10);
            ADC_Actualitzar(channel_AN6); //Tensió Bateria
            E_Carrega = ((unsigned char)(V_Bateria/A - B)); // Actualitzem SOC
            if (E_Carrega >= 98)
            {
                I0_RC3_SetHigh();      //Parem el convertidor
                MC_cont = 7;          //Estat de finalitzar carrega
            }
            else
            {
                I0_RA3_SetHigh();      // Tanquem el rele
            }
            break;
        }
        case 4:
        {
            TC_cont = 1;
        }
    }
}
```

```

        break;
    }
    default:
    {
        TC_cont = 1;
        break;

    }
}
void main(void)
{
    configura ();

    unsigned long V_PWM = 350;
    unsigned long I_PWM = 0;

    IO_RC3_SetHigh();           //Parem el convertidor nomes començar
    IO_RA3_SetLow();           //Assegurem bateria desconectada
    EPWM2_LoadDutyValue(V_PWM);
    EPWM1_LoadDutyValue(I_PWM);
    LCD_Saludo();
    Menu_Pot = 1;

    while(1)
    {
        if (Menu_Pot == 1)
        {
            LCD_Menu_Inicial_Pot(MP_cont);
        }
        if (Menu_Vel == 1)
        {
            LCD_Menu_Inicial_Vel(MV_cont);
        }
        if (Menu_Test == 1)
        {
            LCD_Menu_Test(MT_cont);
        }
        if ((Menu_Carrega == 1) && (Test_OK == 1))
        {
            LCD_Menu_Carrega(MC_cont);
            Time_Carrega(TC_cont);

            if (TC_cont == 4)           //En cas que el corrent sigui interior
            {                          // a la consigna incrementem el PWM AV
                V_PWM = V_PWM + AV_PWM;
                if (V_PWM >= 420)
                {
                    V_PWM = 420;
                }
                EPWM2_LoadDutyValue(V_PWM);

            }
        }
        __delay_s(5);
    }
}

```