

CODI L5

```
#include <xc.h>
```

```
#include <string.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include "config.h"
```

```
#include "GLCD.h"
```

```
#define _XTAL_FREQ 8000000
```

```
/*
```

La barra l'he dividit en 5 troços, i a cada troç es pot emmagatzemar 20 PSI.

Les marques que hi han són indicadors. Les marques grans indiquen que has arribat fins a 20, 40, 60, 80 i 100 PSI respectivament.

De marca a marca s'arriba sumant o restant 4, i entre les marques hi han espais, on s'arriba sumant o restant 2.

```
*/
```

```
char buff[10]; //per mostrar missatges per pantalla
```

```
char prevRA[2] = {0, 0}; //per consultar l'estat previ de RA0 i RA1
```

```
const char * lab = "L5 Individual";
```

```
const char * nom = "Arnau Cullell";
```

```
char pressio = 50; //com a molt tindrà valor fins a 100 psi, llavors amb char ja basta
```

```
int puja_o_baixa = 0; //si arriba a 2, puja el progrés, si arriba a -2 baixa el progrés, int perquè permet números negatius, amb char funciona malament
```

```
char prev_pressio = -1; //per saber la pressió prèvia, char perquè només arribaré fins a 99, no cal més
```

```
char activat_pres = 0; //permet saber si el pressímetre ha sigut premut abans o no, char perquè només serà 0 o 1
```

```
//      PSI =  0  1  2  3  4  5  6  7  8  9
```

```
char press2bit[101] = {63, 63, 62, 62, 61, 61, 60, 60, 59, 59,  
//      10 11 12 13 14 15 16 17 18 19  
          58, 58, 57, 57, 56, 56, 55, 55, 54, 54,  
//      20 21 22 23 24 25 26 27 28 29  
          53, 53, 52, 52, 51, 51, 50, 50, 49, 49,  
//      30 31 32 33 34 35 36 37 38 39  
          48, 48, 47, 47, 46, 46, 45, 45, 44, 44,  
//      40 41 42 43 44 45 46 47 48 49  
          43, 43, 42, 42, 41, 41, 40, 40, 39, 39,  
//      50 51 52 53 54 55 56 57 58 59  
          38, 38, 37, 37, 36, 36, 35, 35, 34, 34,  
//      60 61 62 63 64 65 66 67 68 69  
          33, 33, 32, 32, 31, 31, 30, 30, 29, 29,  
//      70 71 72 73 74 75 76 77 78 79  
          28, 28, 27, 27, 26, 26, 25, 25, 24, 24,  
//      80 81 82 83 84 85 86 87 88 89  
          23, 23, 22, 22, 21, 21, 20, 20, 19, 19,  
//      90 91 92 93 94 95 96 97 98 99  
          18, 18, 17, 17, 16, 16, 15, 15, 14, 14,  
//      100  
          13};
```

```
void writeTxt(byte page, byte y, char * s) {  
    int i=0;  
    while (*s!='\n' && *s!='\0')  
    {  
        putchGLCD(page, y+i, *(s++));  
        i++;  
    };  
}
```

```
void actualitzar_flancs() {  
    prevRA[0] = PORTAbits.RA0;  
    prevRA[1] = PORTAbits.RA1;  
}
```

void dibuixar_barra(char y, char x) { //en el dibuix, entre cada barreta la diferència és de 4 psi, i les barres grans marquen: 0, 20, 40, 60, 80, 100

```
//nivell 0  
writeByte(y, x, 0x80);  
writeByte(y, x+1, 0x80);  
writeByte(y, x+2, 0xAA);  
writeByte(y, x+3, 0xAA);  
writeByte(y, x+4, 0x1F);  
writeByte(y, x+5, 0x20);  
writeByte(y, x+6, 0x40);  
for (int i = 7; i < 21; ++i) writeByte(y, x+i, 0x80);  
writeByte(y, x+21, 0x40);  
writeByte(y, x+22, 0x20);  
writeByte(y, x+23, 0x1F);  
  
//nivell 1  
writeByte(y-1, x, 0x20);  
writeByte(y-1, x+1, 0x20);  
writeByte(y-1, x+2, 0xAA);  
writeByte(y-1, x+3, 0xAA);  
writeByte(y-1, x+4, 0xFF);  
writeByte(y-1, x+23, 0xFF);  
  
//nivell 2  
writeByte(y-2, x, 0x08);
```

```
writeByte(y-2, x+1, 0x08);  
writeByte(y-2, x+2, 0xAA);  
writeByte(y-2, x+3, 0xAA);  
writeByte(y-2, x+4, 0xFF);  
writeByte(y-2, x+23, 0xFF);
```

```
//nivell 3
```

```
writeByte(y-3, x, 0x02);  
writeByte(y-3, x+1, 0x02);  
writeByte(y-3, x+2, 0xAA);  
writeByte(y-3, x+3, 0xAA);  
writeByte(y-3, x+4, 0xFF);  
writeByte(y-3, x+23, 0xFF);
```

```
//nivell 4
```

```
writeByte(y-4, x+2, 0xAA);  
writeByte(y-4, x+3, 0xAA);  
writeByte(y-4, x+4, 0xFF);  
writeByte(y-4, x+23, 0xFF);
```

```
//nivell 5
```

```
writeByte(y-5, x, 0x80);  
writeByte(y-5, x+1, 0x80);  
writeByte(y-5, x+2, 0xAA);  
writeByte(y-5, x+3, 0xAA);  
writeByte(y-5, x+4, 0xFF);  
writeByte(y-5, x+23, 0xFF);
```

```
//nivell 6
```

```
writeByte(y-6, x, 0x20);  
writeByte(y-6, x+1, 0x20);
```

```
writeByte(y-6, x+2, 0xA0);
writeByte(y-6, x+3, 0xA0);
writeByte(y-6, x+5, 0x80);
writeByte(y-6, x+6, 0x40);
for (int i = 7; i < 21; ++i) writeByte(y-6, x+i, 0x20);
writeByte(y-6, x+21, 0x40);
writeByte(y-6, x+22, 0x80);

//omplir barra inicialment
int aux = press2bit[pressio] -1;
for (int i = 63; i != aux; --i) {
    if (i == 63 || i == 12) {} // ja està dibuixat, no cal fer res

    else if (i == 13 || i == 62) { //part de la forma irregular
        for (int j = 7; j < 21; ++j) SetDot(i, x+j);
    }

    else if (i == 14 || i == 61) { //part de la forma irregular
        SetDot(i,x+6);
        SetDot(i,x+21);
        for (int j = 7; j < 21; ++j) SetDot(i, x+j);
    }

    else for (int j = 4; j < 23; ++j) SetDot(i, x+j); //tota la resta del mig
}
}

void empuixar_pressiometre(char y, char x) { //quan es prem un botó, el pressiometre s'activarà
    writeByte(y, x, 0x18);
    writeByte(y, x+1, 0x3C);
    for (int i = 2; i < 6; ++i) writeByte(y, x+i, 0x7E);
    for (int j = 6; j < 10; ++j) writeByte(y, x+j, 0x5A);
```

```
writeByte(y, x+10, 0x7E);  
writeByte(y, x+11, 0x18);  
writeByte(y, x+12, 0x18);  
writeByte(y, x+13, 0xFF);  
for (int k = 14; k < 18; ++k) writeByte(y, x+k, 0x00);  
}
```

void pressiometre_en_repos(char y, char x) { //quan no es prem un botó, el pressiometre es desactiva

```
writeByte(y, x, 0x18);  
writeByte(y, x+1, 0x34);  
writeByte(y, x+2, 0x76);  
for (int i = 3; i < 9; ++i) writeByte(y, x+i, 0x72);  
writeByte(y, x+9, 0x7E);  
writeByte(y, x+10, 0x7E);  
for (int j = 11; j < 17; ++j) writeByte(y, x+j, 0x18);  
writeByte(y, x+17, 0xFF);  
}
```

void dibuixar_bot(char y, char x, char * s, char estat) { //si estat = 1, llavors botó premut, altra banda no estarà premut

```
writeTxt(y, x, s); //dibuixem RA0/RA1  
if (estat == 0) { // no premut  
writeByte(y, x+20, 0x40);  
writeByte(y, x+21, 0x40);  
writeByte(y, x+22, 0x7E);  
writeByte(y, x+23, 0x02);  
writeByte(y, x+24, 0x2A);  
writeByte(y, x+25, 0x12);  
writeByte(y, x+26, 0x2A);  
writeByte(y, x+27, 0x02);  
writeByte(y, x+28, 0x7E);  
}
```

```
    writeByte(y, x+29, 0x40);  
    writeByte(y, x+30, 0x40);  
}
```

```
else { //premut  
    writeByte(y, x+20, 0x40);  
    writeByte(y, x+21, 0x40);  
    writeByte(y, x+22, 0x60);  
    writeByte(y, x+23, 0x20);  
    writeByte(y, x+24, 0x24);  
    writeByte(y, x+25, 0x28);  
    writeByte(y, x+26, 0x24);  
    writeByte(y, x+27, 0x22);  
    writeByte(y, x+28, 0x60);  
    writeByte(y, x+29, 0x40);  
    writeByte(y, x+30, 0x40);  
}  
}
```

```
void canviar_prog(char x) {  
    int y = press2bit[pressio]; //ens donarà la y corresponent del GLCD y:[0,63]  
  
    //pujar  
    if (puja_o_baixa == 2) {  
        puja_o_baixa = 0;  
        if (y == 63 || y == 12) {} // ja està dibuixat, no cal fer res  
  
        else if (y == 13 || y == 62) { //part de la forma irregular  
            for (int j = 7; j < 21; ++j) SetDot(y, x+j);  
        }  
    }  
}
```

```
else if (y == 14 || y == 61) { //part de la forma irregular
    SetDot(y,x+6);
    SetDot(y,x+21);
    for (int j = 7; j < 21; ++j) SetDot(y, x+j);
}

else for (int j = 4; j < 23; ++j) SetDot(y, x+j); //tota la resta del mig
}

//baixar
else if (puja_o_baixa == -2) {
    puja_o_baixa = 0;
    if (y == 14) {} //no cal esborrar perquè sinó treuria el sostre

else if (y == 63 || y == 15) { //part de la forma irregular
    for (int j = 7; j < 21; ++j) ClearDot(y-1, x+j);
}

else if (y == 62 || y == 16) { //part de la forma irregular
    ClearDot(y-1,x+6);
    ClearDot(y-1,x+21);
    for (int j = 7; j < 21; ++j) ClearDot(y-1, x+j);
}

else for (int j = 5; j < 23; ++j) ClearDot(y-1, x+j); //tota la resta del mig
}
}

void mostrar_PSI(char y, char x) {
    sprintf(buff, "%d PSI ", pressio);
    writeTxt(y, x, buff);
}
```



```
}
```

```
void updateGLCD(char y, char x) {
```

```
    mostrar_PSI(0, 10);
```

```
    if (prev_pressio != pressio) canviar_prog(x);
```

```
    if (prevRA[0] == 1 && PORTAbits.RA0 == 1) { //RA0 premut
```

```
        dibuixar_bot(6, 0, "RA0", 1);
```

```
        if (activat_pres == 0) empuixar_pressiometre(y, x+24);
```

```
        activat_pres = 1;
```

```
    }
```

```
    else { //RA0 no premut
```

```
        dibuixar_bot(6, 0, "RA0", 0);
```

```
        if (activat_pres == 0) pressiometre_en_repos(y, x+24);
```

```
        activat_pres = 0;
```

```
    }
```

```
    if (prevRA[1] == 1 && PORTAbits.RA1 == 1) { //RA1 premut
```

```
        dibuixar_bot(7, 0, "RA1", 1);
```

```
        if (activat_pres == 0) empuixar_pressiometre(y, x+24);
```

```
        activat_pres = 1;
```

```
    }
```

```
    else { //RA1 no premut
```

```
        dibuixar_bot(7, 0, "RA1", 0);
```

```
        if (activat_pres == 0) pressiometre_en_repos(y, x+24);
```

```
        activat_pres = 0;
```

```
    }
```

```
}
```

```
void mostrar_nom() {  
    writeTxt(2, 7, lab);  
    writeTxt(4, 7, nom);  
    __delay_ms(1000);  
    clearGLCD(0,7,0,127);  
}
```

```
void configPIC() {  
    ANSELA=0x00; //RA0 i RA1 seran digitals  
    ANSELB=0x00;  
    ANSELC=0x00;  
    ANSELD=0x00;  
  
    TRISD=0x00;  
    TRISB=0x00;  
    TRISA=0x03; //RA0 i RA1 seran inputs  
  
    PORTD=0x00;  
    PORTB=0x00;  
    PORTA=0x00; //netejem ports  
}
```

```
void main(void) {  
    configPIC();  
    GLCDinit();          //Inicialitzem la pantalla  
    clearGLCD(0,7,0,127); //Esborrem pantalla  
    setStartLine(0);     //Definim linia d'inici
```

```
mostrar_nom();  
dibuixar_barra(7, 49); //barra a posició y = 7 i x = 49  
while (1) {  
    if (prevRA[0] == 0 && PORTAbits.RA0 == 1) { //flanc ascendent  
        prev_pressio = pressio;  
        if (pressio == 100) {} //no passarà de 100  
        else {  
            ++pressio;  
            ++puja_o_baixa;  
        }  
    }  
  
    if (prevRA[1] == 1 && PORTAbits.RA1 == 0) { //flanc descendent  
        prev_pressio = pressio;  
        if (pressio == 0) {} //no baixarà de 0  
        else {  
            --pressio;  
            --puja_o_baixa;  
        }  
    }  
  
    if (prev_pressio != pressio || (prevRA[0] == PORTAbits.RA0 || prevRA[1] == PORTAbits.RA1))  
        updateGLCD(7, 49);  
  
    actualitzar_flancs();  
    __delay_ms(75);  
}  
}
```