

```

#include <xc.h>

#include <stdio.h>

#include <string.h>

#include "config.h"

#include "GLCD.h"

#define _XTAL_FREQ 8000000


const char * s1 = "L4A GLCD\n";

const char * s2 = "-----\n";

const char * yan = "Yanick Palacios\n";

const char * arn = "Arnau Cullell\n";

char buff[11]; //com a màxim hi haurà 11 caràcters al (x, y)

const char* buit = "      ";

const char vertical = 0xFF; //pared vertical del quadrat

const char horitzontal = 0x81; //terra i sostre horitzontals


char canal = 0;

int num0 = 0; //no són uint8_t perquè poden arribar valor de fins a 1023, i amb uint8_t només
podriem guardar fins a 255

int num1 = 0;

int resultat = 0;

int x = 0, y = 0;

int dif_x = 0, dif_y = 0;

int prev_x = 0, prev_y = 0;

char quadrat = 0;

char pos_inici_quad[4] = {48, 58, 68, 78};


//-----|-----
//-----Q0-----|-----Q1-----
//---(-dif_x,0)---|---(dif_x,0)---
//-----|-----

```

```
//~~~~~|~~~~~
//-----|-----
//-----Q2-----|-----Q3-----
//---(0,-dif_y)---|---(0,dif_y)---
//-----|-----
//    [Q0] [Q1] [Q2] [Q3]
```

```
void interrupt service_routine_HighP (){
    if (PIR1bits.ADIF && PIE1bits.ADIE) { //ENTRA QUAN ACABA CONVERSIÓ

        resultat = (ADRESH << 8) + ADRESL;

        if(canal == 0) {
            num0 = resultat;
            prev_x = x;
            x = (resultat / 8); //escalat horitzontal
            if (x >= 127) x = 127; //comprovació per que no surti dels límits
            if (x <= 0) x = 0;
            ADCON0bits.CHS = 1; //canal 0 seleccionat
            canal = 1;
        }

        else {
            num1 = resultat;
            prev_y = y;
            y = 63 - (resultat / 16); //escalat vertical i invertit
            if (y >= 63) y = 63;
            if (y <= 0) y = 0;
            ADCON0bits.CHS = 0;
            canal = 0;
        }
    }
}
```

```

    dif_x = x - prev_x;
    dif_y = y - prev_y;

    if (dif_x < 0) quadrat = 0;

    else if (dif_x > 0) quadrat = 1;

    else if (dif_y < 0) quadrat = 2;

    else quadrat = 3;

    ADCON0bits.GO = 1;
    PIR1bits.ADIF = 0;
}
}

```

```

void configPIC() {
    ANSELA=0x03; //analogic AN0 i AN1
    ANSELB=0x00;
    ANSELC=0x00;
    ANSELD=0x00;

    TRISD=0x00;
    TRISB=0x00;
    TRISA=0x03; //inputs són els dos

    PORTD=0x00;
    PORTB=0x00;
    PORTA=0x00;

    //configuració del AD

```

```

ADCON0 = 0x01; //seleccionem el canal AN0 (CHS3:CHS0 = 0000), AD encés (ADON = 1)
ADCON1 = 0x00; //referencia VDD y VSS
ADCON2 = 0xA9; //ADFM=1 (justificació a la dreta), ACQT = 101 (12 Tad), ADCS = 001 (Fosc/8)

//configuració interrupcions
RCONbits.IPEN = 1;
INTCONbits.GIEL = 1;
INTCONbits.GIEH = 1; //activem interrupcions alta prioritat
PIE1bits.ADIE = 1; //activem el ADIE
PIR1bits.ADIF = 0; //flag a 0
IPR1bits.ADIP = 1; //perquè sigui alta prioritat
}

```

```

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

```

```

void mostrar_noms(char* s1, char* s2, char* yan, char* arn) {
    writeTxt(2, 9, s1); //y, x, missatge
    writeTxt(3, 9, s2); //y, x, missatge
    writeTxt(4, 5, yan); //y, x, missatge
    writeTxt(5, 6, arn); //y, x, missatge

    __delay_ms(1000);
    clearGLCD(0,7,0,127); //Esborrem pantalla després de un 1seg
}

```

```

void crear_quadrats() {
    uint8_t comptador = 4;
    uint8_t posicio_x = 48;
    while (comptador != 0) {
        writeByte(5, posicio_x, vertical); //costat vertical
        writeByte(5, posicio_x+1, horitzontal);
        writeByte(5, posicio_x+2, horitzontal);
        writeByte(5, posicio_x+3, horitzontal);
        writeByte(5, posicio_x+4, horitzontal); //formació costat horitzontal
        writeByte(5, posicio_x+5, horitzontal);
        writeByte(5, posicio_x+6, horitzontal);
        writeByte(5, posicio_x+7, vertical); //costat vertical
        posicio_x += 10;
        --comptador;
    }
}

```

```

void omplir_quadrat(){
    writeByte(5, pos_inici_quad[quadrat]+1, 0xFF); //omple dins el quadrat
    writeByte(5, pos_inici_quad[quadrat]+2, 0xFF);
    writeByte(5, pos_inici_quad[quadrat]+3, 0xFF);
    writeByte(5, pos_inici_quad[quadrat]+4, 0xFF);
    writeByte(5, pos_inici_quad[quadrat]+5, 0xFF);
    writeByte(5, pos_inici_quad[quadrat]+6, 0xFF);
}

```

```

void main(void) {
    configPIC();
    GLCDinit();
    clearGLCD(0, 7, 0, 127);
}

```

```

setStartLine(0);

mostrar_noms(s1, s2, yan, arn); //mostrem els noms durant 1seg

crear_quadrats();

ADCON0bits.GO = 1;

while (1) {
    if (y != prev_y || x != prev_x) { //si x o y canvien actualitzem
        PIE1bits.ADIE = 0; //activem el ADIE

        writeTxt(0, 10, buit); //esborrem coordenades
        writeTxt(1, 10, buit); //esborrem diferències
        ClearDot(prev_y, prev_x); //esborrem punt

        sprintf(buff, "(%d,%d)", num0, num1); //omplim la pantalla amb coordenades
        writeTxt(0, 10, buff);

        sprintf(buff, "(%d,%d)", dif_x, dif_y); //omplim la pantalla amb diferència
        writeTxt(1, 10, buff);

        SetDot(y, x); //posem el punt

        crear_quadrats(); //posem els quadrats i borrem el previ al mateix cop
        omplir_quadrat(); //omplim quadrat corresponent
        PIE1bits.ADIE = 1; //activem el ADIE
    }
}
}

```