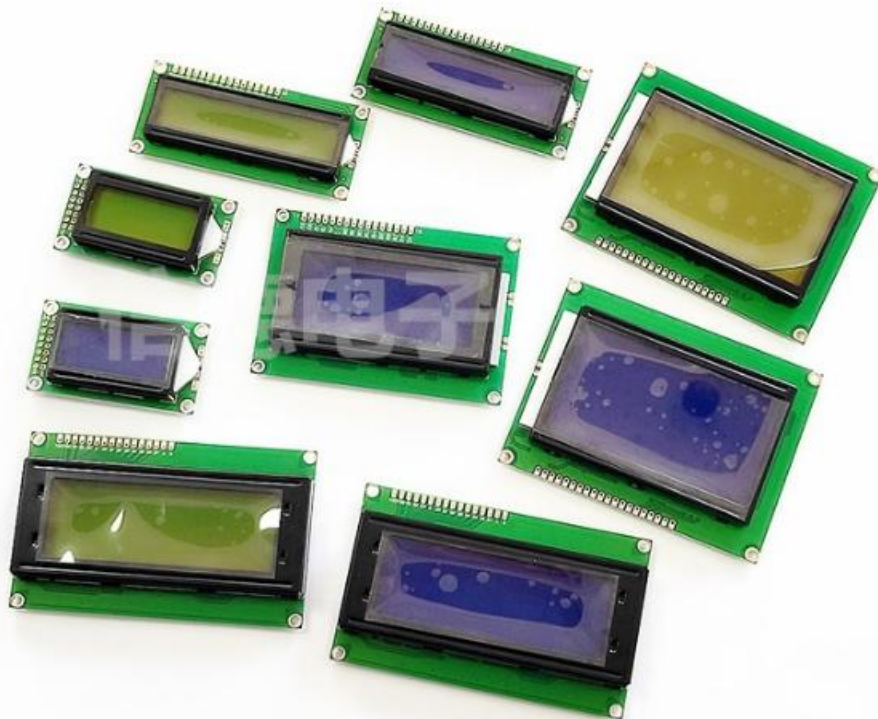


勝特力電材超市-龍山店 886-3-5773766
勝特力電材超市-光復店 886-3-5729570
勝特力电子(上海) 86-21-34970699
勝特力电子(深圳) 86-755-83298787
<http://www.100y.com.tw>

藍屏/黃綠屏 1602A/2004A/12864B 液晶屏 5V LCD 帶背光
IIC/I2C

1602A/2004A/12864B/1604B 液晶屏
藍屏/黃綠屏/帶背光液晶屏/外壳/IIC I2C
0802A 液晶显示屏



1602 採用標準的 16 腳介面，其中：

第 1 腳：VSS 為地電源

第 2 腳：VDD 接 5V 正電源

第 3 腳：V0 為液晶顯示器對比度調整端，接正電源時對比度最弱，接地電源時對比度最高，對比度過高時會產生“鬼影”，使用時可以通過一個 10K 的電位器調整對比度

第 4 腳：RS 為寄存器選擇，高電平時選擇數據寄存器、低電平時選擇指令寄存器。

第 5 腳：R/W 為讀寫信號線，高電平時進行讀操作，低電平時進行寫操作。當 RS 和 RW 共同為低電平時可以寫入指令或者顯示地址，當 RS 為低電平 RW 為高電平時可以讀忙信號，當 RS 為高電平 RW 為低電平時可以寫入數據。

第 6 腳：E 端為使能端，當 E 端由高電平跳變成低電平時，液晶模組執行命令。

第 7~14 腳：D0~D7 為 8 位雙向數據線。

第 15 腳：背光電源正極

第 16 腳：背光電源負極

1602 液晶模組內部的字元發生記憶體 (CGROM) 已經存儲了 160 個不同的點陣字元圖形，如表 1 所示，這些字元有：阿拉伯數字、英文字母的大小寫、常用的符號、和日文假名等，每一個字元都有一個固定的代碼，比如大寫的英文字母“A”的代碼是 01000001B (41H)，顯示時模組把地址 41H 中的點陣字元圖形顯示出來，我們就能看到字母“A”。

以下是在液晶模組的第二行第一個字元的位置顯示字母“A”的程式： ORG 0000H

RS EQU P3.7;確定具體硬體的連接方式

RW EQU P3.6 ;確定具體硬體的連接方式

E EQU P3.5 ;確定具體硬體的連接方式

MOV P1,#00000001B ; 清屏並游標複位

ACALL ENABLE;調用寫入命令副程式

MOV P1,#00111000B ; 設置顯示模式:8 位 2 行 5x7 點陣

ACALL ENABLE ;調用寫入命令副程式

MOV P1,#00001111B ; 顯示器開、游標開、游標允許閃爍

ACALL ENABLE ;調用寫入命令副程式

MOV P1,#00000110B ; 文字不動，游標自動右移

ACALL ENABLE ;調用寫入命令副程式

MOV P1,#0C0H ; 寫入顯示起始地址 (第二行第一個位置)

ACALL ENABLE ;調用寫入命令副程式

MOV P1,#01000001B ; 字母 A 的代碼

SETB RS ; RS=1

CLR RW ; RW=0 ;準備寫入數據

CLR E ; E=0 ;執行顯示命令

ACALL DELAY ;判斷液晶模組是否忙?
SETB E ; E=1 ;顯示完成,程式停車

AJMP \$

ENABLE:

CLR RS ; 寫入控制命令的副程式

CLR RW

CLR E

ACALL DELAY

SETB E

RET

DELAY:

MOV P1, #0FFH ; 判斷液晶顯示器是否忙的副程式

CLR RS

SETB RW

CLR E

NOP

SETB E

JB P1.7, DELAY ; 如果 P1.7 為高電平表示忙就迴圈等待

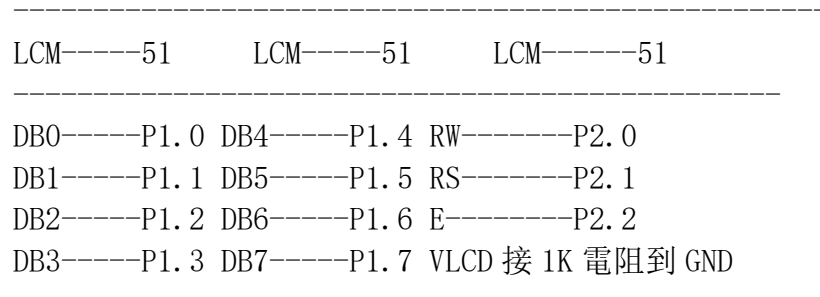
RET

END

程式在開始時對液晶模組功能進行了初始化設置，約定了顯示格式。注意顯示字元時光標是自動右移的，無需人工干預，每次輸入指令都先調用判斷液晶模組是否忙的副程式 DELAY，然後輸入顯示位置的地址 0C0H，最後輸入要顯示的字元 A 的代碼 41H。

SMC1602A (16*2) 模擬口線接線方式

連接線圖:



[注:AT89S52 使用 12M 晶振]

=====*/

```
#define LCM_RW    P2_0 //定義引腳
#define LCM_RS    P2_1
#define LCM_E     P2_2
#define LCM_Data  P1
#define Busy      0x80 //用於檢測 LCM 狀態字中的 Busy 標識

#include

void WriteDataLCM(unsigned char WDLCM);
void WriteCommandLCM(unsigned char WCLCM, BuysC);
unsigned char ReadDataLCM(void);
unsigned char ReadStatusLCM(void);
void LCMInit(void);
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData);
void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData);
void Delay5Ms(void);
void Delay400Ms(void);

unsigned char code uctech[] = {"uctech"};
unsigned char code net[] = {"uctech.icpcn.com"};

void main(void)
{
    Delay400Ms(); //啟動等待, 等 LCM 講入工作狀態
    LCMInit(); //LCM 初始化
    Delay5Ms(); //延時片刻(可不要)

    DisplayListChar(0, 5, uctech);
    DisplayListChar(0, 0, net);
    ReadDataLCM(); //測試用句無意義
    while(1);
}

//寫數據
void WriteDataLCM(unsigned char WDLCM)
{
    ReadStatusLCM(); //檢測忙
    LCM_Data = WDLCM;
```

```

    LCM_RS = 1;
    LCM_RW = 0;
    LCM_E = 0; //若晶振速度太高可以在這後加小的延時
    LCM_E = 0; //延時
    LCM_E = 1;
}

//寫指令
void WriteCommandLCM(unsigned char WCLCM, BuysC) //BuysC 為 0 時忽略忙檢測
{
    if (BuysC) ReadStatusLCM(); //根據需要檢測忙
    LCM_Data = WCLCM;
    LCM_RS = 0;
    LCM_RW = 0;
    LCM_E = 0;
    LCM_E = 0;
    LCM_E = 1;
}

//讀數據
unsigned char ReadDataLCM(void)
{
    LCM_RS = 1;
    LCM_RW = 1;
    LCM_E = 0;
    LCM_E = 0;
    LCM_E = 1;
    return(LCM_Data);
}

//讀狀態
unsigned char ReadStatusLCM(void)
{
    LCM_Data = 0xFF;
    LCM_RS = 0;
    LCM_RW = 1;
    LCM_E = 0;
    LCM_E = 0;
    LCM_E = 1;
    while (LCM_Data & Busy); //檢測忙信號
    return(LCM_Data);
}

```

```

void LCMInit(void) //LCM 初始化
{
    LCM_Data = 0;
    WriteCommandLCM(0x38, 0); //三次顯示模式設置，不檢測忙信號
    Delay5Ms();
    WriteCommandLCM(0x38, 0);
    Delay5Ms();
    WriteCommandLCM(0x38, 0);
    Delay5Ms();

    WriteCommandLCM(0x38, 1); //顯示模式設置，開始要求每次檢測忙信號
    WriteCommandLCM(0x08, 1); //關閉顯示
    WriteCommandLCM(0x01, 1); //顯示清屏
    WriteCommandLCM(0x06, 1); // 顯示游標移動設置
    WriteCommandLCM(0x0C, 1); // 顯示開及游標設置
}

//按指定位置顯示一個字元
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char DData)
{
    Y &= 0x1;
    X &= 0xF; //限制 X 不能大於 15, Y 不能大於 1
    if (Y) X = 0x40; //當要顯示第二行時地址碼+0x40;
    X = 0x80; // 算出指令碼
    WriteCommandLCM(X, 0); //這裏不檢測忙信號，發送地址碼
    WriteDataLCM(DData);
}

//按指定位置顯示一串字元
void DisplayListChar(unsigned char X, unsigned char Y, unsigned char code *DData)
{
    unsigned char ListLength;

    ListLength = 0;
    Y &= 0x1;
    X &= 0xF; //限制 X 不能大於 15, Y 不能大於 1
    while (DData[ListLength]>0x20) //若到達字串尾則退出
    {
        if (X <= 0xF) //X 座標應小於 0xF
        {

```

```
        DisplayOneChar(X, Y, DData[ListLength]); //顯示單個字元
        ListLength++;
        X++;
    }
}

//5ms 延時
void Delay5Ms(void)
{
    unsigned int TempCyc = 5552;
    while(TempCyc--);
}

//400ms 延時
void Delay400Ms(void)
{
    unsigned char TempCycA = 5;
    unsigned int TempCycB;
    while(TempCycA--)
    {
        TempCycB=7269;
        while(TempCycB--);
    };
};
```