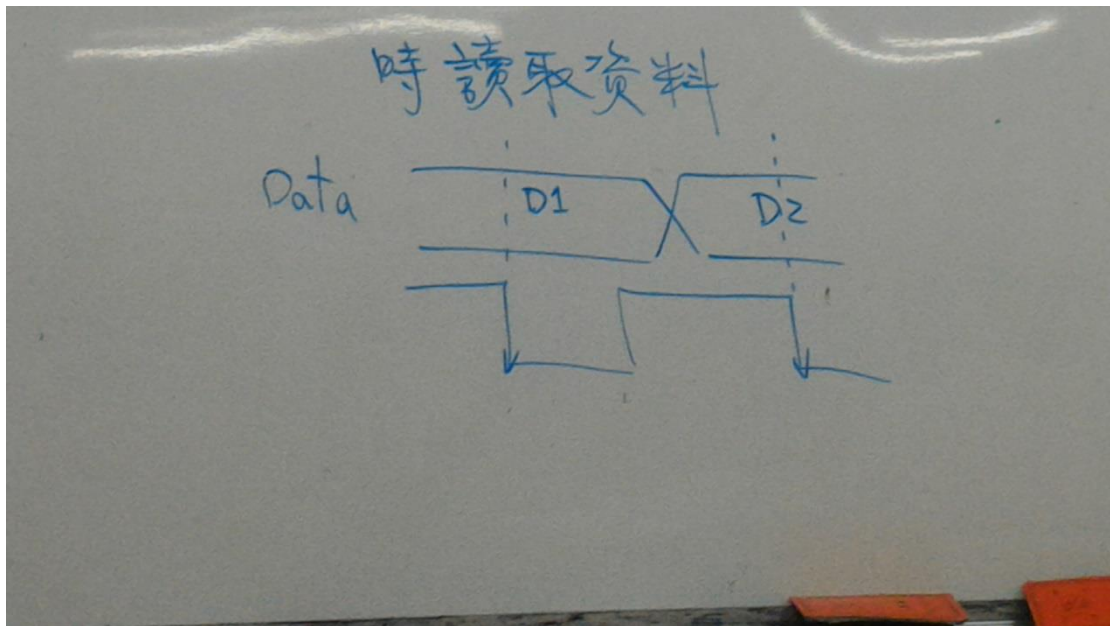


## LCD 模組

1.使用 4-bit 模式

2.Enable 腳位在負緣時讀取資料



3.8 位元資料/指令需分送給 LCD 模組高 4 位元(high nibble)先送,低四位元(low nibble)後送

4.8051 模擬器 P1.7~P1.4 接到 LCD 模組 DB7~DB4

P1.3 接 LCD 模組 RS 腳

RS=1 為資料

RS=0 為指令

P1.2 接 LCD 模組之 enable

5.LCD 模組指令(RS=0)

清除螢幕 0x01

回歸原點 0x02

輸入設定 0x04~0x07

4:游標往前顯示不動

5:游標往前顯示移動

6:游標往後顯示不動

7:游標往後顯示移動

顯示亮滅控制 0x08~0x0f

D C V (0001DCB)2

D=1 螢幕亮

D=0 螢幕滅

C=1 有游標

C=0 無游標

B=1 游標閃爍

B=0 游標不閃爍

例:LCD 模組螢幕亮游標閃爍

此指令=0x0f

游標/移動控制 0x10~0x1f

(0001 S/C R/L XX)2

S/C=1 顯示移動

S/C=0 游標移動

R/L=1 往右移(往後移)

R/L=0 往左移(往前移)

功能設定 0x20~0x2f

(001 DL N F XX)

DL=1 8 位元模式

DL=0 4 位元模式

N=1 2 行

N=0 1 行

F=1 5x10 字體

F=0 5x8 字體

例:使用 4 位元模式 2 行顯示 5x8 字體之指令 0x28

設定顯示資料 RAM 位址(DDRM)

0x80~0xff

(1 A6 A5 A4 A3 A2 A1 A0)2

A6~A0 為 DDRAM 位址

0x00 為第 1 行起始位址

0x40 為第 2 行起始位址

LCD 模組起始化

功能設定 4 位元,2 行,5x8 字體(0x28)

↓

輸入設定 游標往後,顯示不能(0x06)

↓

螢幕亮滅控制 螢幕亮,游標閃爍(0x0f)

LCD 顯示單字元

設定顯示資料位址(RS=0)

↓

寫入完資料(RS=1)

接腳設定

```

__sfr __at (0x97) P1_7;
__sfr __at (0x96) P1_6;
__sfr __at (0x95) P1_5;
__sfr __at (0x94) P1_4;
__sfr __at (0x93) P1_3;
__sfr __at (0x92) P1_2;
#define DB7 P1_7
#define DB6 P1_6
#define DB5 P1_5
#define DB4 P1_4
#define RS P1_3
#define E P1_2

```

只令函式

```

void inst(unsigned char val)
{
RS=0;
DB7=val>>7;
DB6=(val&0x40)>>6;
DB5=(val&0x20)>>5;
DB4=(val&0x10)>>4;
E=0;
E=1;
DB7=(val&0x8)>>3;
DB6=(val&0x4)>>2;
DB5=(val&0x2)>>1;
DB4=(val&0x1);
E=0;
E=1;
}

```

功能設定在 4 位元模式需將高 4 位元,設定兩次,更改 inst 函式

```

void inst(unsigned char val ,char rpt)
{
RS=0;
E=1;
DB7=val>>7;
DB6=(val&0x40)>>6;
DB5=(val&0x20)>>5;
DB4=(val&0x10)>>4;

```

```

if(rpt){E=0;E=1;}
DB7=(val&0x8)>>3;
DB6=(val&0x4)>>2;
DB5=(val&0x2)>>1;
DB4=(val&0x1);
E=0;
E=1;
}

```

寫入單元

```

sendchar(char ch)
{
RS=1;
DB7=ch>>7;
DB6=(ch&0x40)>>6;
DB5=(ch&0x20)>>5;
DB4=(ch&0x10)>>4;
E=0; E=1;
DB6=(ch&0x08)>>3;
DB5=(ch&0x04)>>2;
DB4=(ch&0x02)>>1;
DB4=(ch&0x01)>>0;
E=0; E=1;
}

```

在#include <8051.h>下加入

```

int _sdcc_external_startup(){return 1;}
***

```

\_\_sbit \_\_at (0x97) DB7;(改正)

將所有的\_\_sfr 改善 \_\_sbit

新增 delay 副程式

```

void delay(){
char c;
for(c=0;c<50;c++);
}

```

\*\*\*

取出 val 位元 7 之值

```
bitval=(val>>7)&1;
```

取出 val 位元 6 之值

```
bitval=(val>>6)&1;
```

寫成副程式

```
char getbit(char val, char pos){return ((val>>pos)&1);}
```