

# HT67F40 使用 C 語言實現 LCD 控制方法

文件編碼：HA0294T

## 簡介

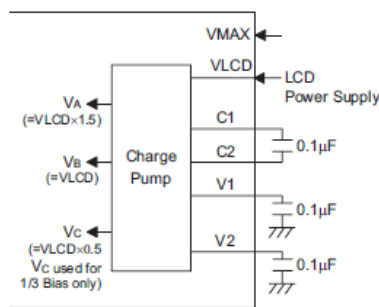
HT67Fx0 為內建有 R/C-Type LCD 功能，提供 1/2 或 1/3 Bias、4 個 COM 的驅動能力，本文以 HT67F40 為母體，給出 1/3 Bias、1/4 Duty 的 LCD 驅動使用方法。

## Theory of Operation

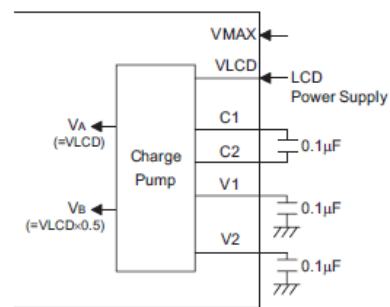
根據 LCD 的驅動原理可知，LCD 像素點上只能加上 AC 電壓，LCD 顯示器的對比度由 COM 腳上的電壓值減去 SEG 腳上的電壓值決定，當這個電壓差大於 LCD 的飽和電壓就能打開像素點，小於 LCD 閾值電壓就能關閉像素點，LCD 型 MCU 已經由內建的 LCD 驅動電路自動產生 LCD 驅動信號，因此能完成 LCD 的驅動。

驅動選擇如表：

Part No.	Duty	Drive No.	Bias	Bias Type	Wave Type
HT67F40	1/2	33x2	1/2 or 1/3	C or R	A or B
	1/3	33x3			
	1/4	32x4			

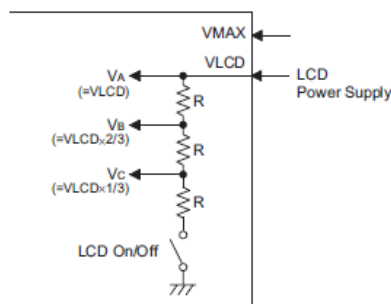


C type 1/3 Bias -  
x3 Charge Pump

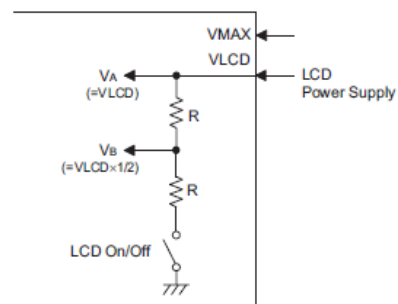


C type 1/2 Bias -  
x2 Charge Pump

C Type Bias Voltage Levels



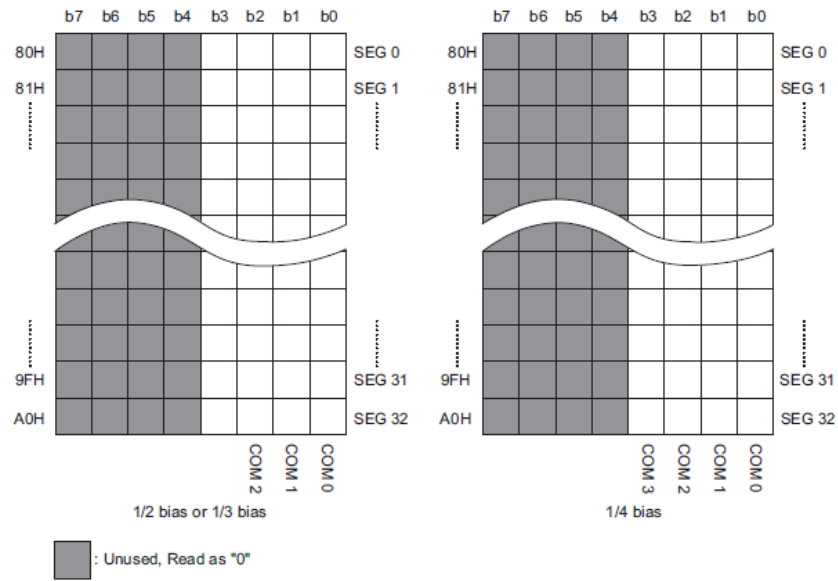
R type 1/3 Bias



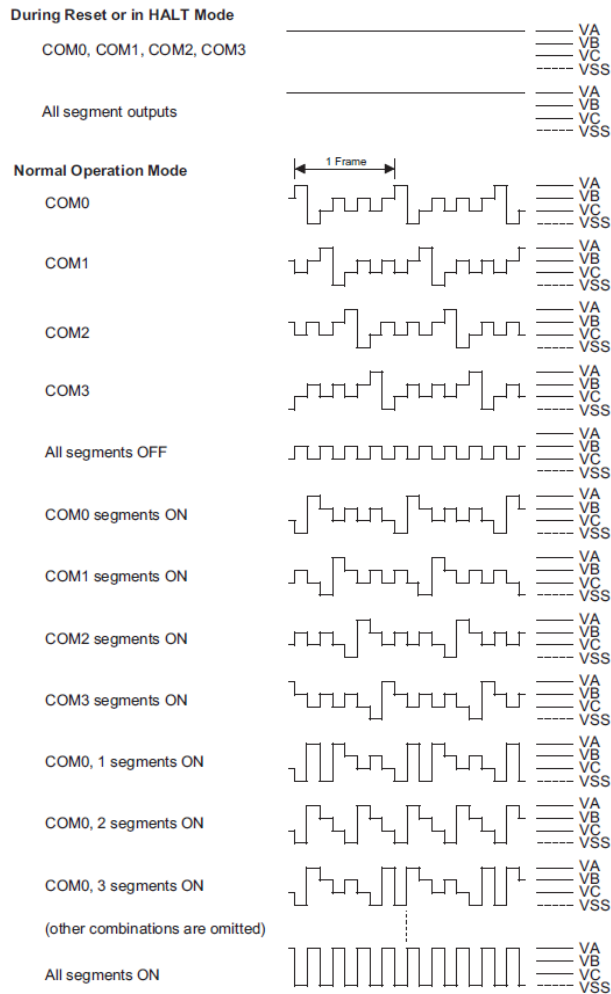
R type 1/2 Bias

R Type Bias Voltage Levels

## LCD Memory Map

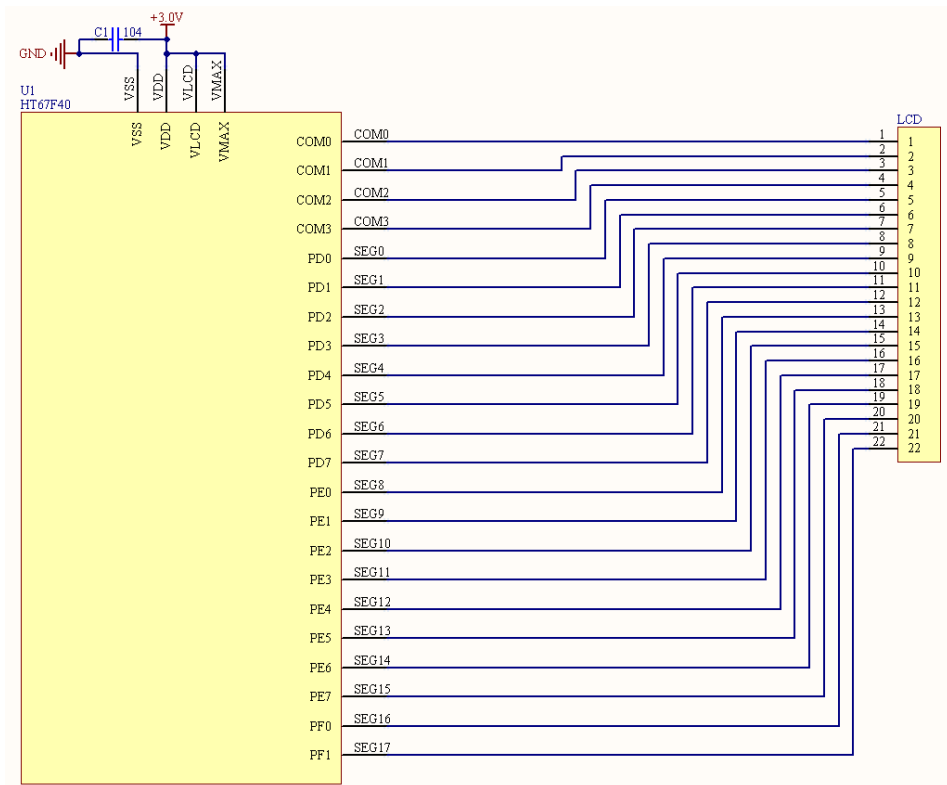


## LCD Waveform Timing Diagrams

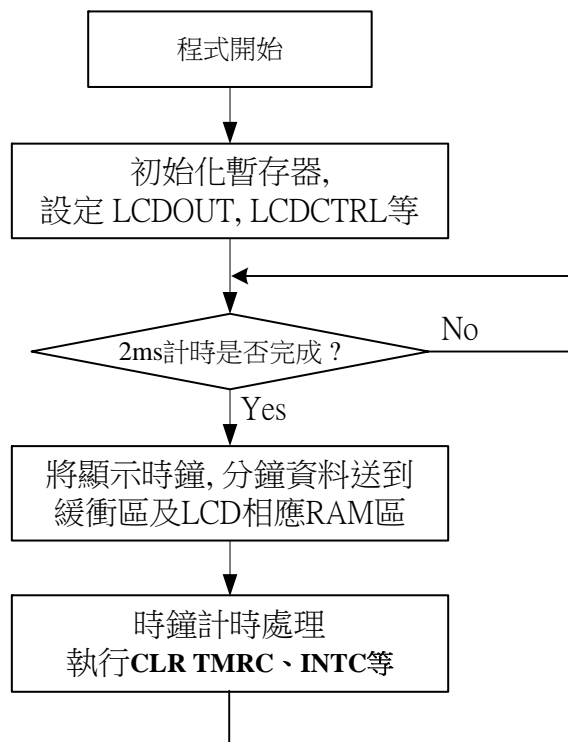


LCD Driver Output – Type A - 1/4 Duty, 1/3 Bias  
 Note: For 1/3 R type bias, the  $VA=VLCD$ ,  $VB=VLCD \times 2/3$  and  $VC=VLCD \times 1/3$ .  
 For 1/3 C type bias, the  $VA=VLCD \times 1.5$ ,  $VB=VLCD$  and  $VC=VLCD \times 1/2$ .

### Application Circuit



### S/W Flowchart



## 程式說明

本範例程式採用 4×18 段式 LCD，循環顯示實時時鐘 00:00~23:59 的資料，共使用了 9 個 Byte 的顯示緩衝 RAM (DISP8 ~ DISP16)，詳見軟體定義。用戶可以根據實際使用情況取捨 COM 和增減 SEG。將 LCD 的 RAM 某位元設定 1，相應的 LCD 段碼會點亮，相反清除為零則不亮。

### 程式範例

//以下為定義頭檔

```
//lcd.h
#ifndef _holtek_h_
#define _holtek_h_

#define u8    unsigned char    /* unsigned 8 bit type definition */
#define s8    signed char      /* signed 8 bit type definition */
#define u16   unsigned int     /* unsigned 16 bit type definition */
#define s16   signed int       /* signed 16 bit type definition */
#define u32   unsigned long    /* unsigned 32 bit type definition */
#define s32   signed long      /* signed 32 bit type definition */

#define nop _nop()
#define clrwdt1 _clrwdt1()
#define clrwdt2 _clrwdt2()

//RAM DEFINE
extern u8    DISP8;           //display buffer
extern u8    DISP9;
extern u8    DISP10;
extern u8    DISP11;
extern u8    DISP12;
extern u8    DISP13;
extern u8    DISP14;
extern u8    DISP15;
extern u8    DISP16;

#define F3_S  DISP8|=0x1 //0
#define F3_C  DISP8&=0xfe
#define G3_S  DISP8|=0x2 //1
#define G3_C  DISP8&=0xfd
#define E3_S  DISP8|=0x4 //2
#define E3_C  DISP8&=0xfb

#define B3_S  DISP9|=0x1 //0
#define B3_C  DISP9&=0xfe
#define C3_S  DISP9|=0x2 //1
#define C3_C  DISP9&=0xfd
#define D3_S  DISP9|=0x4 //2
#define D3_C  DISP9&=0xfb
#define A3_S  DISP9|=0x8 //3
#define A3_C  DISP9&=0xf7

#define F4_S  DISP10|=0x1 //0
#define F4_C  DISP10&=0xfe
#define G4_S  DISP10|=0x2 //1
#define G4_C  DISP10&=0xfd
#define E4_S  DISP10|=0x4 //2
```

```

#define E4_C DISP10&=0xfb

#define B4_S DISP11|=0x1 //0
#define B4_C DISP11&=0xfe
#define C4_S DISP11|=0x2 //1
#define C4_C DISP11&=0xfd
#define D4_S DISP11|=0x4 //2
#define D4_C DISP11&=0xfb
#define A4_S DISP11|=0x8 //3
#define A4_C DISP11&=0xf7

#define COL_S DISP12|=0x1 //0
#define COL_C DISP12&=0xfe

#define F5_S DISP13|=0x1 //0
#define F5_C DISP13&=0xfe
#define G5_S DISP13|=0x2 //1
#define G5_C DISP13&=0xfd
#define E5_S DISP13|=0x4 //2
#define E5_C DISP13&=0xfb

#define B5_S DISP14|=0x1 //0
#define B5_C DISP14&=0xfe
#define C5_S DISP14|=0x2 //1
#define C5_C DISP14&=0xfd
#define D5_S DISP14|=0x4 //2
#define D5_C DISP14&=0xfb
#define A5_S DISP14|=0x8 //3
#define A5_C DISP14&=0xf7

#define F6_S DISP15|=0x1 //0
#define F6_C DISP15&=0xfe
#define G6_S DISP15|=0x2 //1
#define G6_C DISP15&=0xfd
#define E6_S DISP15|=0x4 //2
#define E6_C DISP15&=0xfb

#define B6_S DISP16|=0x1 //0
#define B6_C DISP16&=0xfe
#define C6_S DISP16|=0x2 //1
#define C6_C DISP16&=0xfd
#define D6_S DISP16|=0x4 //2
#define D6_C DISP16&=0xfb
#define A6_S DISP16|=0x8 //3
#define A6_C DISP16&=0xf7

void Chip_Init();
void DIS_P();
void dis_p3();
void dis_p4();
void dis_p5();
void dis_p6();
void timer_p();

#endif
    
```

```

//以下為主程式內容
//lcd.c
//function: edit for segment0~17 + COM0~3 to drive LCD
//MCU:HT67F40
//option:
//SysVolt: 3.0V
//OSC: HIRC
//WDT: enable
//SysFreq: 4M
//Vlcd: 3.0V
//PB0/RES: I/O pin
//Others select by user

#include "HT67F40.h"
#include "lcd.h"

#pragma vector TM_ISR @0x14 //1ms timer
const u8 DigitCode[16] = {0x28,0x7e,0xa4,0x64,0x72,0x61,0x21,0x7c,0x20,0x60,0x30,
                          0x23,0xa9,0x26,0xa1,0xb1}; //0~f

#pragma rambank0 //下面為RAM定義
u8 DISP8; //display buffer
u8 DISP9;
u8 DISP10;
u8 DISP11;
u8 DISP12;
u8 DISP13;
u8 DISP14;
u8 DISP15;
u8 DISP16;

u8 time2ms;
u8 SENDDATA;
u8 STATE;
u8 t500ms;
u16 t1s;
u8 hour;
u8 min;
u8 index;
u16 timels_dis;
u16 timels;
u8 tmin;

bit time2msflag;
bit t500msflag;
bit t1sflag;
bit poweronflag;

//-----main-----
void main() //主程式
{
    Chip_Init();

    for(_mp0 = 0x80; _mp0 < 0xff; _mp0++) //bank0
    {
        _iar0=0;
    }

    _dmbp0=1;
    for(_mp1 = 0x80; _mp1 < 0xa0; _mp1++) //bank1
    {

```

```

        _iar1=0;
    }
    _dmbp0=0;

    poweronflag=1;

    while(1)
    {
        clrwdt1;
        clrwdt2;
        if(time2msflag)
        {
            time2msflag=0;
            DIS_P();                //顯示資料送到緩衝區

            _dmbp0=1;                //顯示緩衝區資料送到 LCD RAM 區
            _mp1 = 0x88;
            _iar1= DISP8;
            _mp1 = 0x89;
            _iar1= DISP9;
            _mp1 = 0x8a;
            _iar1= DISP10;
            _mp1 = 0x8b;
            _iar1= DISP11;
            _mp1 = 0x8c;
            _iar1= DISP12;
            _mp1 = 0x8d;
            _iar1= DISP13;
            _mp1 = 0x8e;
            _iar1= DISP14;
            _mp1 = 0x8f;
            _iar1= DISP15;
            _mp1 = 0x90;
            _iar1= DISP16;
            _dmbp0=0;

            timer_p();                //時鐘資料處理
        }
    }
}
//-----
void Chip_Init()                //初始化程式
{
    _cp0c=0;
    _cp1c=0;
    _acer1=0;

    _lcdout0=0;                //seg0~7
    _lcdout1=0;                //seg8~15
    _lcdout2=0b11111100;        //seg16~17
    _lcdctrl=0b01001111;        //1/3 bias、1/4 duty

    _t0ae=1;
    _ade=0;
    _mf0e=1;
    _tm0c0=0;
    _tm0a1=0b11101000;        //1ms
    _tm0ah=0b00000011;
    _tm0c1=0b11000001;
    _t0on=1;
}

```

```

        _emi=1;
    }
    //-----
    void timer_p()                //時鐘資料處理
    {
        if(tlsflag)
        {
            tlsflag=0;
            if(++tmin>=60)
            {
                tmin=0;
                if(++min>=60)
                {
                    min=0;
                    if(++hour>=24)
                        hour=0;
                }
            }
        }
    }
    //-----
    void DIS_P()                //顯示資料處理
    {
        if(STATE)
        {
            index=hour/10;
            SENDDATA = ~DigitCode[index];
            dis_p3();

            index=hour%10;
            SENDDATA = ~DigitCode[index];
            dis_p4();

            index=min/10;
            SENDDATA = ~DigitCode[index];
            dis_p5();

            index=min%10;
            SENDDATA = ~DigitCode[index];
            dis_p6();

            if(!t500msflag)
                COL_C;
            else
                COL_S;
        }
        else
        {
            if(poweronflag)
            {
                F3_S;
                G3_S;
                E3_S;
                DISP9=0x0f;
                F4_S;
                G4_S;
                E4_S;
                DISP11=0x0f;
                COL_S;
                F5_S;
                G5_S;
            }
        }
    }
}

```

```

        E5_S;
        DISP14=0x0F;
        F6_S;
        G6_S;
        E6_S;
        DISP16=0x0F;
    }
    else
    {
        DISP8=0;
        DISP9=0;
        DISP10=0;
        DISP11=0;
        DISP12=0;
        DISP13=0;
        DISP14=0;
        DISP15=0;
        DISP16=0;
        STATE=1;
    }
}
}
//-----
void dis_p3()                //hour 高位元顯示
{
    if(SENDDATA & 0x80)
        C3_S;
    else
        C3_C;

    if(SENDDATA & 0x40)
        E3_S;
    else
        E3_C;

    if(SENDDATA & 0x10)
        D3_S;
    else
        D3_C;

    if(SENDDATA & 0x08)
        G3_S;
    else
        G3_C;

    if(SENDDATA & 0x04)
        F3_S;
    else
        F3_C;

    if(SENDDATA & 0x02)
        A3_S;
    else
        A3_C;

    if(SENDDATA & 0x01)
        B3_S;
    else
        B3_C;
}
//-----

```

```

void dis_p4()                //hour 低位元顯示
{
    if(SENDDATA & 0x80)
        C4_S;
    else
        C4_C;

    if(SENDDATA & 0x40)
        E4_S;
    else
        E4_C;

    if(SENDDATA & 0x10)
        D4_S;
    else
        D4_C;

    if(SENDDATA & 0x08)
        G4_S;
    else
        G4_C;

    if(SENDDATA & 0x04)
        F4_S;
    else
        F4_C;

    if(SENDDATA & 0x02)
        A4_S;
    else
        A4_C;

    if(SENDDATA & 0x01)
        B4_S;
    else
        B4_C;
}
//-----
void dis_p5()                //min 高位元顯示
{
    if(SENDDATA & 0x80)
        C5_S;
    else
        C5_C;

    if(SENDDATA & 0x40)
        E5_S;
    else
        E5_C;

    if(SENDDATA & 0x10)
        D5_S;
    else
        D5_C;

    if(SENDDATA & 0x08)
        G5_S;
    else
        G5_C;

    if(SENDDATA & 0x04)

```

```

        F5_S;
    else
        F5_C;

    if(SENDDATA & 0x02)
        A5_S;
    else
        A5_C;

    if(SENDDATA & 0x01)
        B5_S;
    else
        B5_C;
}
//-----
void dis_p6()                //min 低位元顯示
{
    if(SENDDATA & 0x80)
        C6_S;
    else
        C6_C;

    if(SENDDATA & 0x40)
        E6_S;
    else
        E6_C;

    if(SENDDATA & 0x10)
        D6_S;
    else
        D6_C;

    if(SENDDATA & 0x08)
        G6_S;
    else
        G6_C;

    if(SENDDATA & 0x04)
        F6_S;
    else
        F6_C;

    if(SENDDATA & 0x02)
        A6_S;
    else
        A6_C;

    if(SENDDATA & 0x01)
        B6_S;
    else
        B6_C;
}
//-----
//1ms timer program
//-----
void TM_ISR()
{
    _t0af=0;

    if(++time2ms>=2)
    {

```

```

        time2ms=0;
        time2msflag=1;

    if(poweronflag)
    {
        if(++time1s_dis>=1000)    //啟動電源全顯 2S
        {
            time1s_dis=0;
            poweronflag=0;
        }
    }
    else
    {
        if(++t500ms>=250)
        {
            t500ms=0;
            t500msflag = ~t500msflag; //秒閃

            if(++t1s>=2)
            {
                t1s=0;
                t1sflag=1;
            }
        }
    }
}

```

## Conclusions

本範例驅動 4×18 LCD 顯示正常，用戶只需要稍加改造即可套用到所選規格的 LCD 上。