



DY2402PBL 技术手册 (完整版)

DoYoung.net 原创技术资料

2007.10.27

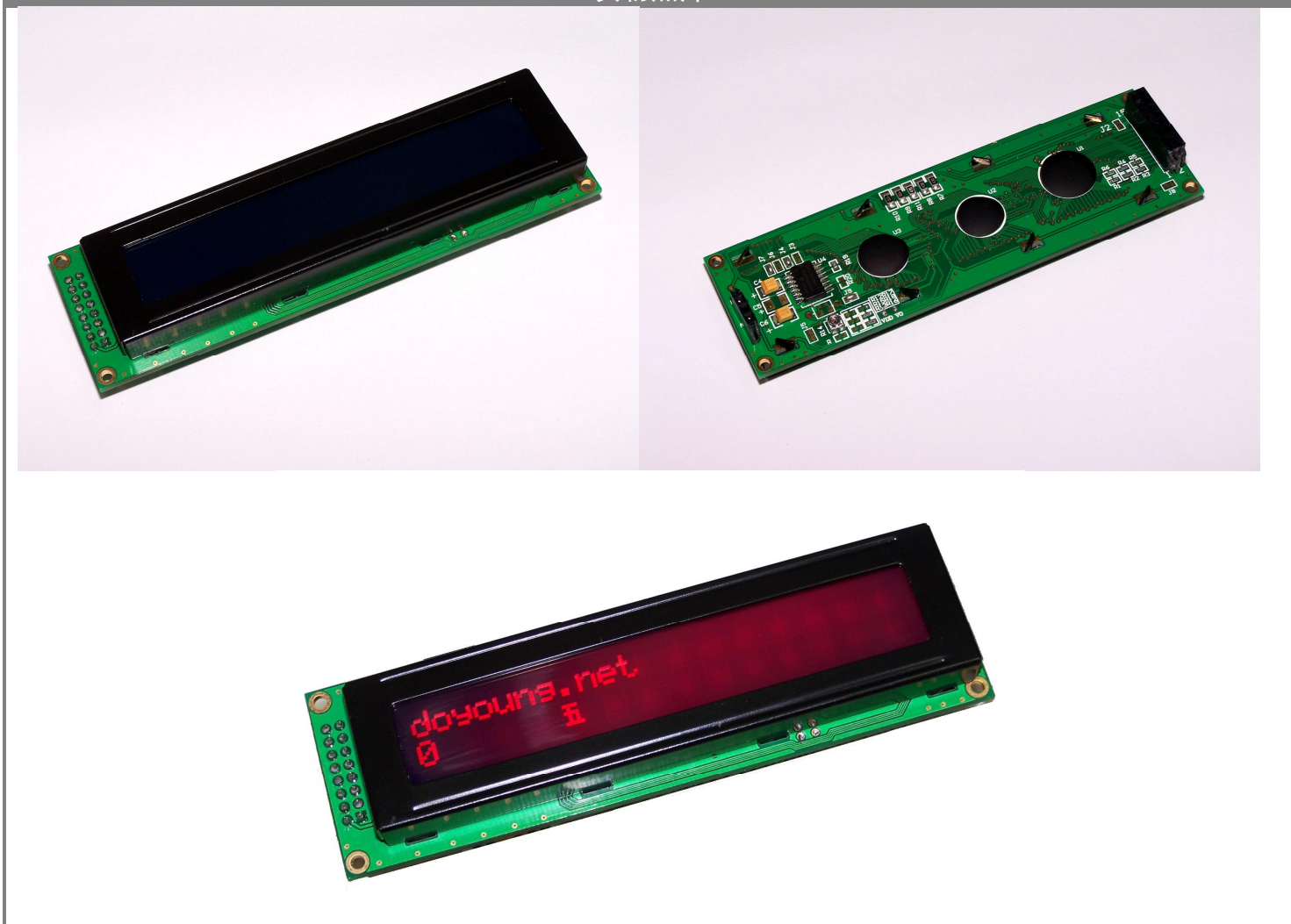
DY2402PBL

24 X 2 字符型液晶显示屏模块

- n 24 行 X 2 列字符显示, 5 X 8 单字符点阵
- n 并行 8 / 4 线数据通信
- n 红色字符, 黑色背景, 独立 LED 背光电源
- n 内置液晶灰度调整电路
- n 标准 ASCII 字符库, 可自定义 8 个用户字符

品名	字符型液晶显示屏模块
型号	MBCF24204B03 (原厂)
规格	24 X 2 STN LCD 点阵
文档制作	杜洋 (DoYoung.net)
创建日期	2007-10-27
版本信息	V1.0

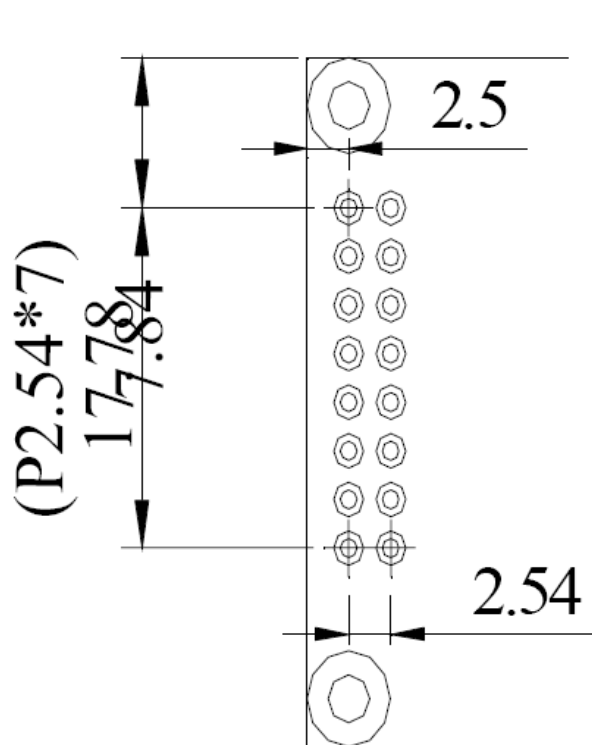
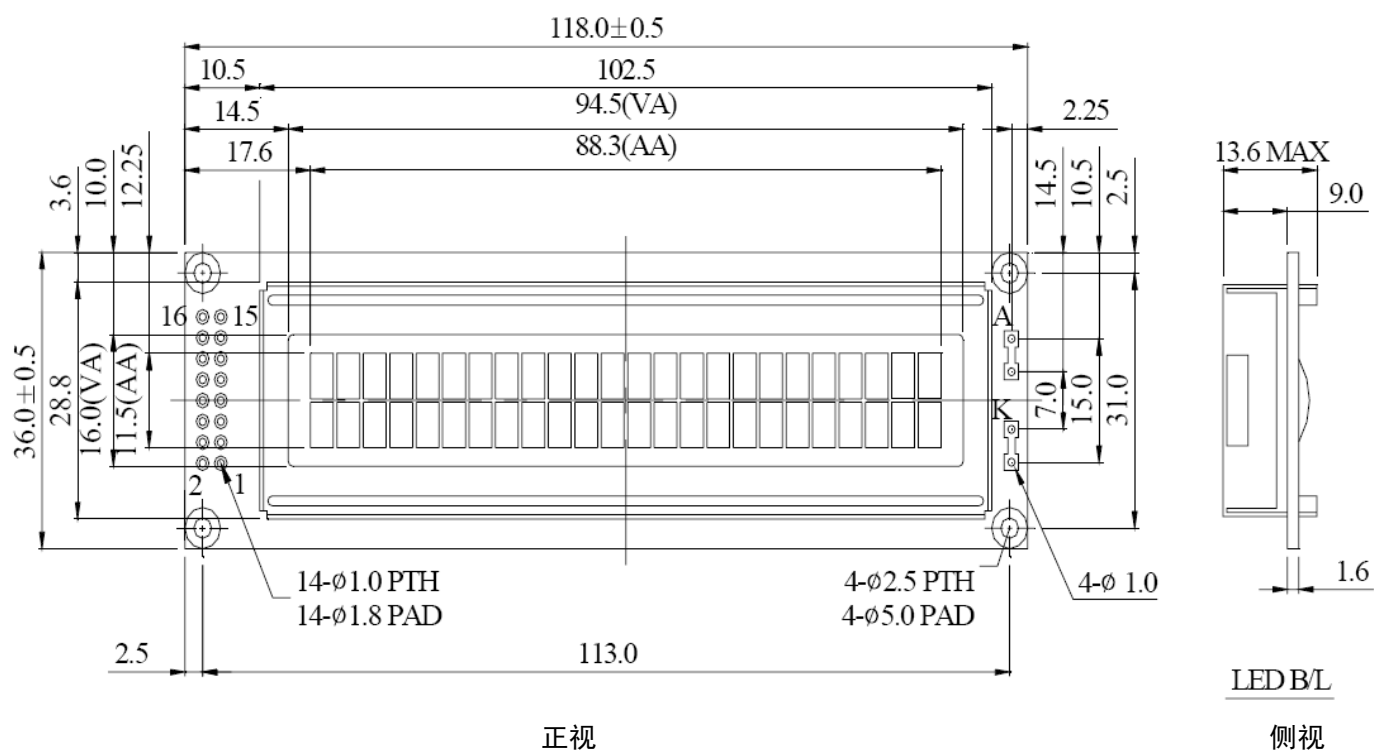
实物照片



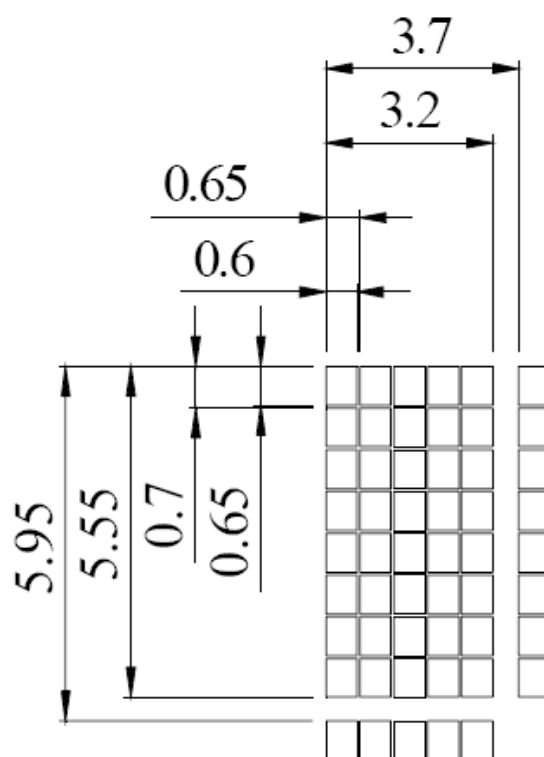
机械参数表

Item	Dimension	Unit
Number of Characters	24 characters x 2 Lines	-
Module dimension	118.0 x 36.0 x 13.6(MAX)	mm
View area	94.5 x 16.0	mm
Active area	88.3 x 11.5	mm
Dot size	0.6 x 0.65	mm
Dot pitch	0.65 x 0.70	mm
Character size	3.2 x 5.55	mm
Character pitch	3.7 x 5.95	mm
LCD type	STN , Positive , Transflective , Yellow Green	
Duty	1/16	
View direction	6 o'clock	
Backlight Type	LED , Yellow Green	

机械结构图

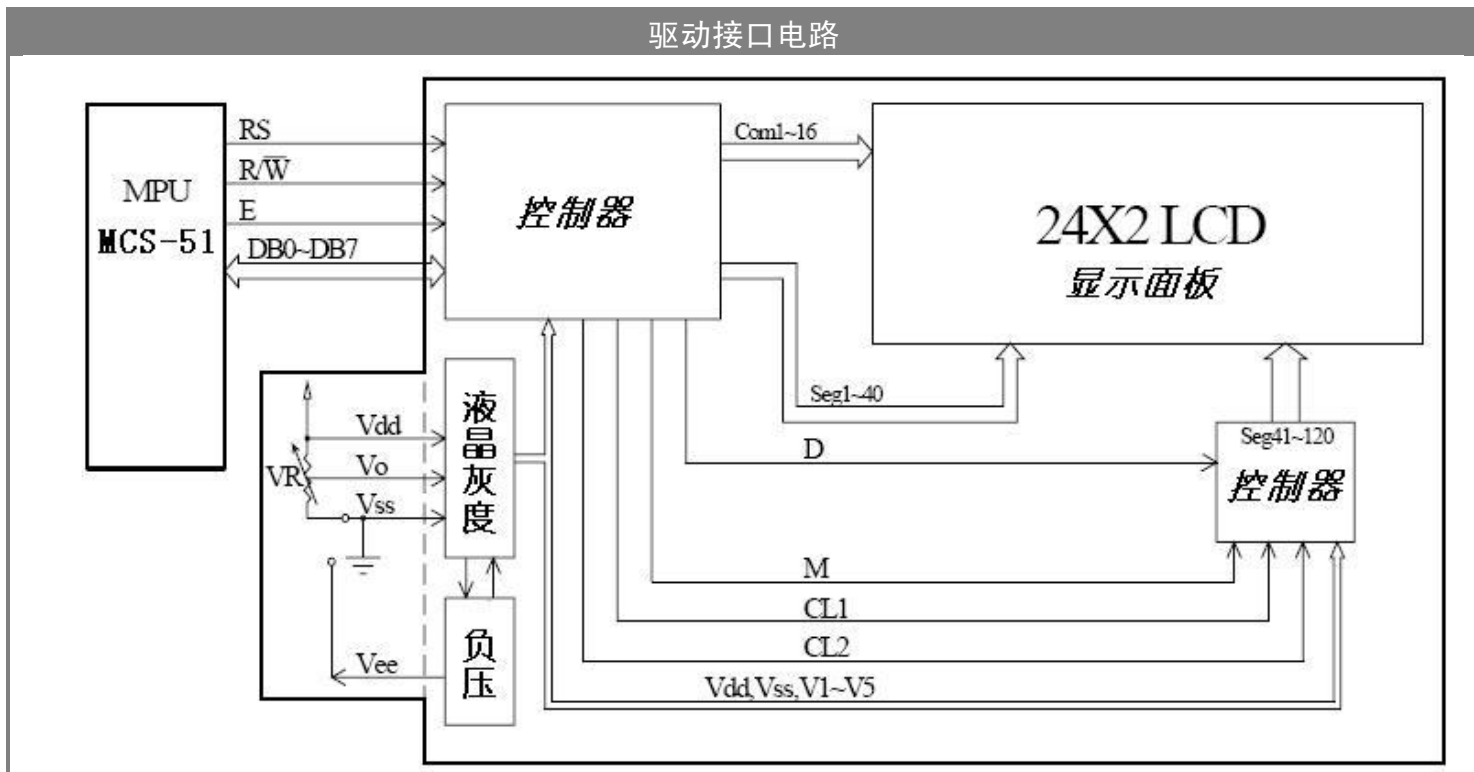


引脚位

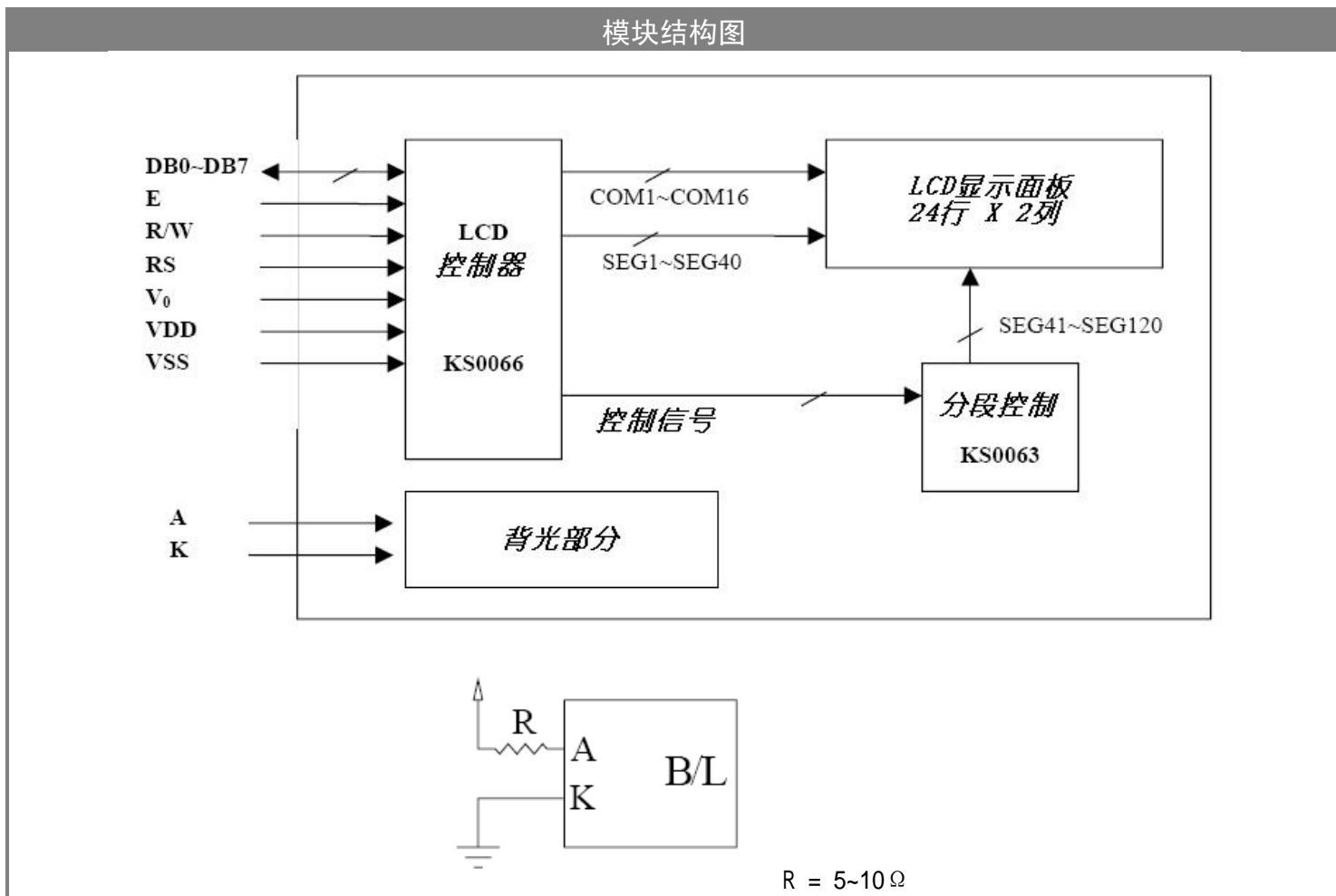


液晶点

驱动接口电路



模块结构图



ASCII 码对应表

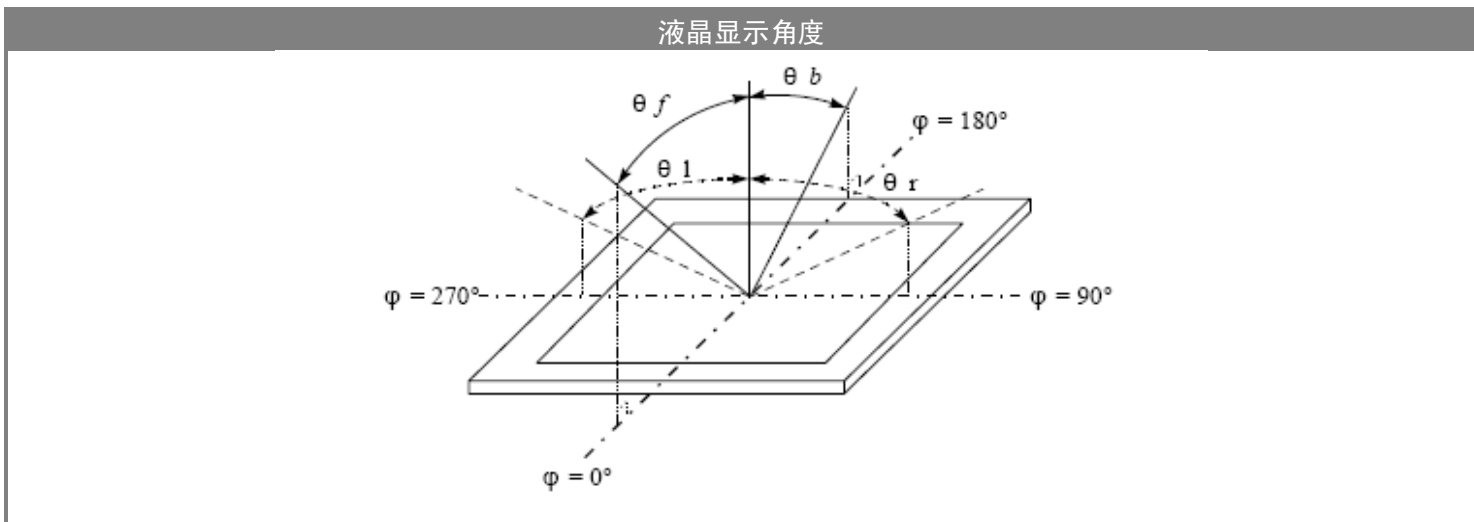
Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL (1)	CG RAM (1)			0	1	2	3	4				5	6	7	8	9
LLLH (2)		!	"	#	\$	%	&	'		()	*	+	,	-	.
LLHL (3)		:	;	<	=	>	?	@		[\]	^	_	~	`
LLHH (4)		!	"	#	\$	%	&	'		()	*	+	,	-	.
LHLL (5)		:	;	<	=	>	?	@		[\]	^	_	~	`
LHLH (6)		:	;	<	=	>	?	@		[\]	^	_	~	`
LHHL (7)		:	;	<	=	>	?	@		[\]	^	_	~	`
LHHH (8)		:	;	<	=	>	?	@		[\]	^	_	~	`
HLLL (1)		:	;	<	=	>	?	@		[\]	^	_	~	`
HLLH (2)		:	;	<	=	>	?	@		[\]	^	_	~	`
HLHL (3)		:	;	<	=	>	?	@		[\]	^	_	~	`
HLHH (4)		:	;	<	=	>	?	@		[\]	^	_	~	`
HHLL (5)		:	;	<	=	>	?	@		[\]	^	_	~	`
HHLH (6)		:	;	<	=	>	?	@		[\]	^	_	~	`
HHHL (7)		:	;	<	=	>	?	@		[\]	^	_	~	`
HHHH (8)		:	;	<	=	>	?	@		[\]	^	_	~	`

电气参数表

Item	Symbol	Min	Typ	Max	Unit
Operating Temperature	T_{OP}	-20	-	+70	°C
Storage Temperature	T_{ST}	-30	-	+80	°C
Input Voltage	V_I	V_{SS}	-	V_{DD}	V
Supply Voltage For Logic	$V_{DD}-V_{SS}$	-0.3	-	7	V
Supply Voltage For LCD	$V_{DD}-V_0$	-0.3	-	13	V

Item	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage For Logic	$V_{DD}-V_{SS}$	-	4.5	-	5.5	V
Supply Voltage For LCD	$V_{DD}-V_0$	$T_a=0^{\circ}\text{C}$	-	-	4.8	V
		$T_a=25^{\circ}\text{C}$	-	4.5	-	V
		$T_a=50^{\circ}\text{C}$	4.2	-	-	V
Input High Volt.	V_{IH}	-	2.2	-	V_{DD}	V
Input Low Volt.	V_{IL}	-	-	-	0.6	V
Output High Volt.	V_{OH}	-	2.4	-	-	V
Output Low Volt.	V_{OL}	-	-	-	0.4	V
Supply Current	I_{DD}	$V_{DD}=5\text{V}$	-	1.2	-	mA

Item	Symbol	Condition	Min	Typ	Max	Unit
View Angle	(V) θ	$CR \geq 2$	10	-	105	deg
	(H) ϕ	$CR \geq 2$	-30	-	30	deg
Contrast Ratio	CR	-	-	3	-	-
Response Time	T rise	-	-	200	300	ms
	T fall	-	-	200	300	ms



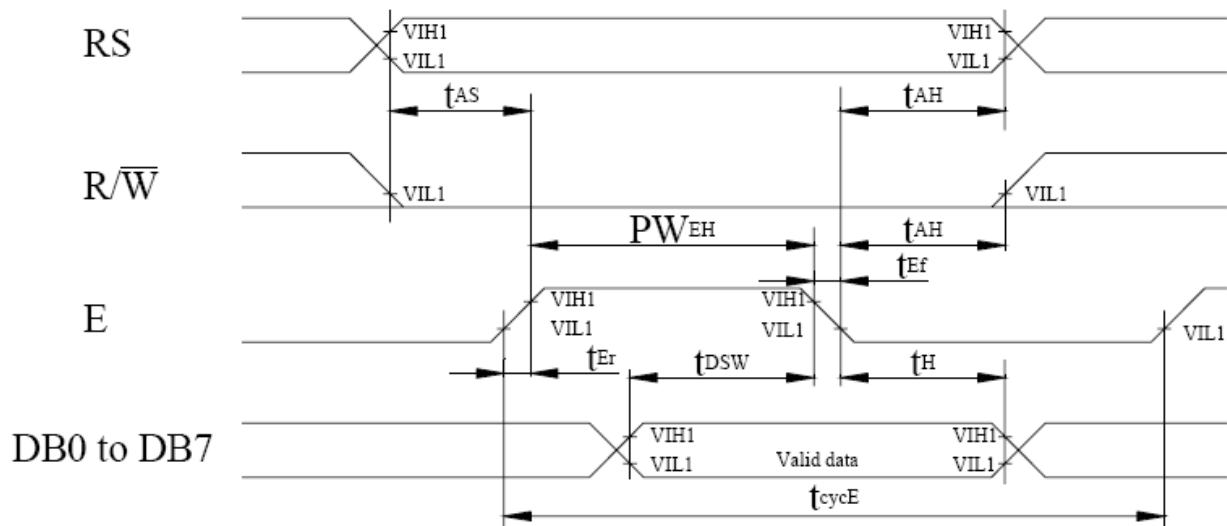
引脚说明			
标号	名称	参数	说明
1	Vss	GND 0V	模块电源 地
2	Vdd	+5V	模块电源 +5V DC
3	VO	NC(空置)	LCM 内置液晶驱动, 调整 R14 (可变电阻) 得到最佳显示效果
4	RS	H/L	H: 数据; L: 指令; (H 表示高电平, L 表示低电平)
5	R/W	H/L	H: MPU → LCM 读数据; L: MPU → LCM 写数据;
6	E	H/H → L	LCM 使能
7	DB0	H/L	数据线 0
8	DB1	H/L	数据线 1
9	DB2	H/L	数据线 2
10	DB3	H/L	数据线 3
11	DB4	H/L	数据线 4
12	DB5	H/L	数据线 5
13	DB6	H/L	数据线 6
14	DB7	H/L	数据线 7
15	LED-A	+5V	LED 背光电源 +5V DC
16	LED-K	GND 0V	LED 背光电源 地

- n 在背光电路中需要串接 5 欧姆电阻, 直接将背光接入电源可以会造成损坏。
- n 请在 DB0~DB7 数据线中加入 10K 上拉电阻, 特别是使用单片机中的三态 I/O 口时。
- n VO 在使用时可以悬空, 因为 LCM 内部置有调整电路, 调节 LCM 背板上的 R14 可变电阻得到最佳显示效果。

指令集表												
指令	数据位										功能细述	耗时 270KHZ
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
清屏	0	0	0	0	0	0	0	0	0	1	清除 DDRAM 和 AC 中的数据	1.53mS
归位	0	0	0	0	0	0	0	0	1	-	AC=0 光标、画面回到首位	1.53mS
输入方式	0	0	0	0	0	0	0	1	I/D	S	I/D=0 读写后 AC 减 1 I/D=1 读写后 AC 加 1 S=0 读写后画面不动 S=1 读写后画面平移	39uS
显示设置	0	0	0	0	0	0	1	D	C	B	D=0 关显示 D=1 开显示 C=0 关光标 C=1 开光标 B=0 光标闪烁 B=1 光标不动	39uS
光标画面	0	0	0	0	0	1	S/C	R/L	-	-	S/C=0 光标平移 1 位 S/C=1 画面平移 1 位 R/L=0 左移 R/L=1 右移	39uS
功能设置	0	0	0	0	1	DL	N	F	-	-	DL=0 4 位数据接口 DL=1 8 位数据接口 N=0 单行显示 N=1 两行显示 F=0 关显示 F=1 开显示	39uS
CGRAM 地址	0	0	0	1	A5	A4	A3	A2	A1	A0	设置 CGRAM 地址 00H~3FH	39uS
DDRAM 地址	0	0	1	A6	A5	A4	A3	A2	A1	A0	设置 DDRAM 地址 两行显示时 第一行 00H~17H 两行显示时 第二行 40H~57H	39uS
读 BF/AC 值	0	1	BF	A6	A5	A4	A3	A2	A1	A0	BF=0 LCM 正忙 BF=1 LCM 空闲 A6~A0 AC 所在地址	0uS
写数据	1	0	A7	A6	A5	A4	A3	A2	A1	A0	4/8 位数据	43uS
读数据	1	1	A7	A6	A5	A4	A3	A2	A1	A0	4/8 位数据	43uS

n BF 读忙 可以使用读忙函数或根据工作耗时使用相应延时程序

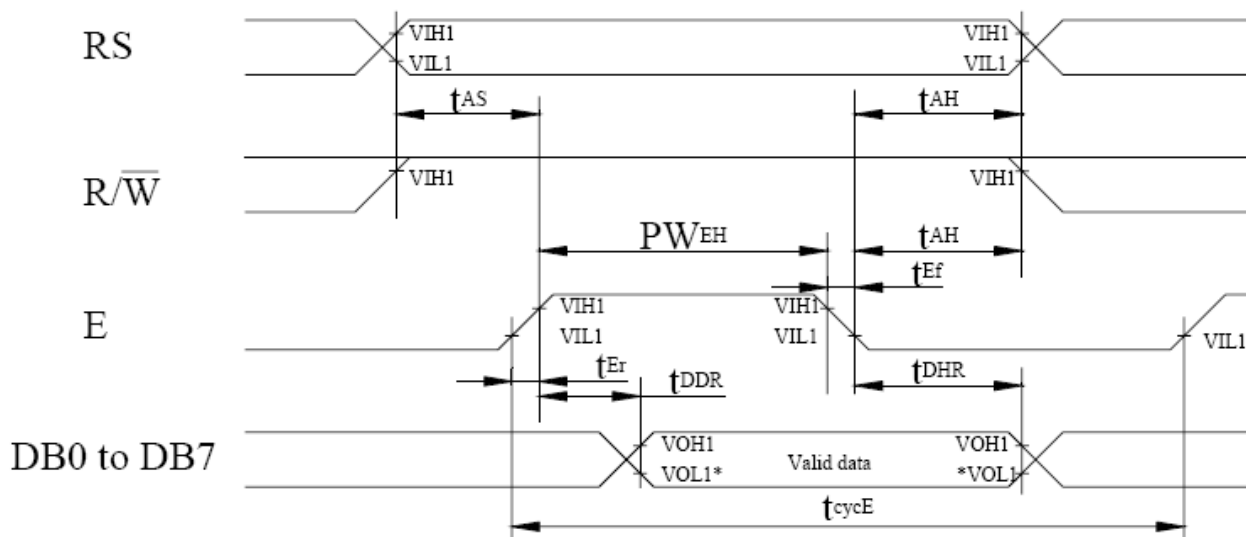
写数据时序图



$T_a=25^{\circ}C, V_{DD}=5.0\pm 0.5V$

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	t_{cycE}	400	-	-	ns
Enable pulse width (high level)	PW_{EH}	150	-	-	ns
Enable rise/fall time	t_{Er}, t_{Ef}	-	-	25	ns
Address set-up time (RS, R/W to E)	t_{AS}	30	-	-	ns
Address hold time	t_{AH}	10	-	-	ns
Data set-up time	t_{DSW}	40	-	-	ns
Data hold time	t_H	10	-	-	ns

读数据时序图



Ta=25°C, VDD=5.0± 0.5V

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	t _{cycE}	400	-	-	ns
Enable pulse width (high level)	PW _{EH}	150	-	-	ns
Enable rise/fall time	t _{Er} , t _{Ef}	-	-	25	ns
Address set-up time (RS, R/W to E)	t _{AS}	30	-	-	ns
Address hold time	t _{AH}	10	-	-	ns
Data delay time	t _{DDR}	-	-	100	ns
Data hold time	t _{DHR}	20	-	-	ns

DDRAM 位置分部图

1	2	3	4	5	6	7	21	22	23	24			
00	01	02	03	04	05	06					14	15	16	17
40	41	42	43	44	45	46					54	55	56	57

16 进制数, 每行 24 个字符的位置分部

CGRAM 自定义字符说明

For 5 * 8 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)		
7 6 5 4 3 2 1 0		5 4 3 2 1 0		7 6 5 4 3 2 1 0		
High	Low	High	Low	High	Low	
0 0 0 0 * 0 0 0		0 0 0	0 0 0	* * *	0 0 0 0	Character pattern(1)
			0 0 1	* * *	0 0 0 0	
			0 1 0	* * *	0 0 0 0	
			0 1 1	* * *	0 0 0 0	
			1 0 0	* * *	0 0 0 0	
			1 0 1	* * *	0 0 0 0	
			1 1 0	* * *	0 0 0 0	
			1 1 1	* * *	0 0 0 0	
			0 0 0	* * *	0 0 0 0	
			0 0 1	* * *	0 0 0 0	
0 0 0 0 * 0 0 1		0 0 1	0 1 1	* * *	0 0 0 0	Character pattern(2)
			1 0 0	* * *	0 0 0 0	
			1 0 1	* * *	0 0 0 0	
			1 1 0	* * *	0 0 0 0	
			1 1 1	* * *	0 0 0 0	
			0 0 0	* * *	0 0 0 0	
			0 0 1	* * *	0 0 0 0	
			0 1 0	* * *	0 0 0 0	
			0 1 1	* * *	0 0 0 0	
			1 0 0	* * *	0 0 0 0	
				* * *		Cursor pattern
				* * *		
				* * *		
				* * *		

For 5 * 10 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)		
7 6 5 4 3 2 1 0		5 4 3 2 1 0		7 6 5 4 3 2 1 0		
High	Low	High	Low	High	Low	
0 0 0 0 * 0 0 0		0 0	0 0 0 0	* * *	0 0 0 0 0 0	Character pattern
			0 0 0 1	* * *	0 0 0 0 0 0	
			0 0 1 0	* * *	0 0 0 0 0 0	
			0 0 1 1	* * *	0 0 0 0 0 0	
			0 1 0 0	* * *	0 0 0 0 0 0	
			0 1 0 1	* * *	0 0 0 0 0 0	
			0 1 1 0	* * *	0 0 0 0 0 0	
			0 1 1 1	* * *	0 0 0 0 0 0	
			1 0 0 0	* * *	0 0 0 0 0 0	
			1 0 0 1	* * *	0 0 0 0 0 0	
				* * *		Cursor pattern
				* * *		
				* * *		
				* * *		

自定义字符在 CGRAM 中从上到下，每组数据取低 5 位送入一行



驱动程序 (C 语言)

```
/*
// LCM2402_driver // LCM2402 驱动程序及测试程序 //
// 用于驱动 LCM2402 (MBCF24204B03) //
// 作者 | 杜洋 // 2007-4-12 | V1.0 // MCS-51 | 12MHZ //
// (还有更多内容在 MBCF24204B03_技术手册_完整版) //
// DoYoung.net 电子技术
// -- 创造独立资源
// http://doyoung.net/

#include <AT89X52.h> // 包含头文件 //
#include <string.h>

typedef unsigned char uint8; // 无符号 8 位整型变量 //

// 引脚定义 // (使用者根据实际更改)
#define LCM2402_DBO_DB7 P0 // 定义 LCM2402 的数据总线
sbit LCM2402_RS = P1 ^ 0; // 定义 LCM2402 的 RS 控制线
sbit LCM2402_RW = P1 ^ 2; // 定义 LCM2402 的 RW 控制线
sbit LCM2402_E = P1 ^ 1; // 定义 LCM2402 的 E 控制线
sbit LCM2402_Busy = P0 ^ 7; // 定义 LCM2402 的测忙线 (与 LCM2402_DBO_DB7 关联)

// 定义 LCM2402 指令集 // (详细请见技术手册)
#define CMD_clear 0x01 // 清除屏幕
#define CMD_back 0x02 // DDRAM 回零位
#define CMD_dec1 0x04 // 读入后 AC (指针) 减 1, 向左写
#define CMD_add1 0x06 // 读入后 AC (指针) 加 1, 向右写
#define CMD_dis_gb1 0x0f // 开显示_开光标_开光标闪烁
#define CMD_dis_gb2 0x0e // 开显示_开光标_关光标闪烁
#define CMD_dis_gb3 0x0c // 开显示_关光标_关光标闪烁
#define CMD_OFF_dis 0x08 // 关显示_关光标_关光标闪烁
#define CMD_set82 0x38 // 8 位总线_2 行显示
#define CMD_set81 0x30 // 8 位总线_1 行显示 (上边行)
#define CMD_set42 0x28 // 4 位总线_2 行显示
#define CMD_set41 0x20 // 4 位总线_1 行显示 (上边行)
#define lin_1 0x80 // 4 位总线_1 行显示 (上边行)
#define lin_2 0xc0 // 4 位总线_1 行显示 (上边行)

// 读 LCM 忙程序 [底层协议] // (所有底层协议都无需关注)
// LCM2402 测忙, 若 LCM2402 处于忙状态, 本函数将等待至非忙状态 //
void LCM2402_TestBusy(void){
```



```
LCM2402_DB0_DB7 = 0xff;    //设备读状态
LCM2402_RS = 0;
LCM2402_RW = 1;
LCM2402_E = 1;
while(LCM2402_Busy);      //等待 LCM 不忙
LCM2402_E = 0;           //
}
/*****
// 写指令程序 //
// 向 LCM2402 写命令 本函数需要 1 个指令集的入口参数 //
/***** /
void LCM2402_WriteCMD(uint8 LCM2402_command) {
    LCM2402_TestBusy();
    LCM2402_DB0_DB7 = LCM2402_command;
    LCM2402_RS = 0;
    LCM2402_RW = 0;
    LCM2402_E = 1;
    LCM2402_E = 0;
}
/*****
// 写数据程序 //
// 向 LCM2402 写数据 //
/***** /
void LCM2402_WriteData(uint8 LCM2402_data){
    LCM2402_TestBusy();
    LCM2402_DB0_DB7 = LCM2402_data;
    LCM2402_RS = 1;
    LCM2402_RW = 0;
    LCM2402_E = 1;
    LCM2402_E = 0;
}
/*****
// 打印字符串程序 // (本函数调用指针函数)
// 向 LCM 发送一个字符串,长度 48 字符之内
// 第一行位置 0x00~0x17 第二行位置 0x40~0x57
// 应用举例: print(0x80, "doyoung.net"); //在第一行第一位处从左向右打印 doyoung.net 字符串
/***** /
void print(uint8 a,uint8 *str){
    LCM2402_WriteCMD(a | 0x80);
    while(*str != '\0'){
        LCM2402_WriteData(*str++);
    }
    *str = 0;
}
/*****
// 打印单字符程序 //
// 第一行位置 0x00~0x17 第二行位置 0x40~0x57
// 向 LCM 发送一个字符,以十六进制 (0x00) 表示
// 应用举例: print(0xc0,0x30); //在第二行第一位处打印字符 "0"
/***** /
```

```
void print2(uint8 a,uint8 t){
    LCM2402_WriteCMD(a | 0x80);
    LCM2402_WriteData(t);
}
/*****
// 定义小汉字 //
// 可写入 8 个自字义字符，写入后可用其 CGRAM 代码直接提取显示。
// 字符定义方法请参考技术手册
/***** /
uint8 code Xword[]={
    0x18,0x18,0x07,0x08,0x08,0x08,0x07,0x00,    //℃，代码 0x00
    0x00,0x00,0x00,0x00,0xff,0x00,0x00,0x00,    //一，代码 0x01
    0x00,0x00,0x00,0x0e,0x00,0xff,0x00,0x00,    //二，代码 0x02
    0x00,0x00,0xff,0x00,0x0e,0x00,0xff,0x00,    //三，代码 0x03
    0x00,0x00,0xff,0xf5,0xfb,0xf1,0xff,0x00,    //四，代码 0x04
    0x00,0xfe,0x08,0xfe,0x0a,0x0a,0xff,0x00,    //五，代码 0x05
    0x00,0x04,0x00,0xff,0x00,0x0a,0x11,0x00,    //六，代码 0x06
    0x00,0x1f,0x11,0x1f,0x11,0x11,0x1f,0x00,    //日，代码 0x07
};
void CgramWrite(void) { // 装入 CGRAM //
    uint8 i;
    LCM2402_WriteCMD(0x06);    // CGRAM 地址自动加 1
    LCM2402_WriteCMD(0x40);    // CGRAM 地址设为 00 处
    for(i=0;i<64;i++) {
        LCM2402_WriteData(Xword[i]); // 按数组写入数据
    }
}
/*****
// LCM2402 初始化 // (使用者可自定义，加 * 号程序行必须保留但可修改)
/***** /
void LCM2402_Init(void){
    LCM2402_WriteCMD(CMD_set82);    /* 显示模式设置：显示 2 行，每个字符为 5*7 个像素
    LCM2402_WriteCMD(CMD_clear);    // 显示清屏
    LCM2402_WriteCMD(CMD_back);    /* 数据指针指向第 1 行第 1 个字符位置
    LCM2402_WriteCMD(CMD_add1);    // 显示光标移动设置：文字不动，光标右移
    LCM2402_WriteCMD(CMD_dis_gb3); // 显示开及光标设置：显示开，光标开，闪烁开
    CgramWrite();    // 向 CGRAM 写入自定义字符
}
/*****
//          以上是 LCM2402 驱动程序          //

/*****
// 测试用函数 //
void main (void){
    LCM2402_Init();//LCM2402 初始化
    while(1){ //主线程//
        print(0x00,"doyoung.net"); //在第一行第一位处从左向右打印 doyoung.net 字符串
        print2(0x40,0x30); //在第一行第一位处打印字符“0”
        print2(0x47,0x05); //在第一行第八位处打印自定义字符“五”
    }
}
```



```
}  
}  
/***** /  
/***** /
```

修改记录

2007-10-27 添加 校对全文