

● 电路与设计

32 段 CMOS LCD 驱动器 AY0438 及其与单片机的接口设计

空军工程大学电讯工程学院 孙中禹 王永泉

Interfacing to 32 - Segment CMOS LCD Driver and MCU

Sun Zhongyu Wang Yongquan

摘要 :AY0438 是 Microchip 公司生产的 32 段 CMOS LCD 驱动器,它结构简单,使用方便。可在单片机的控制下直接驱动 LCD 显示模块。文中介绍了 AY0438 的引脚功能和电气参数。给出了 AY0438 在 PIC16C71 单片机的控制下驱动 LCD 显示模块的实际应用电路及程序。

关键词 :LCD 显示 ;驱动器 ;单片机 ;接口 ;AY0438

分类号 :TN722 文献标识码 :B 文章编号 :1006 - 6977(2000)07 - 0032 - 04

1. 概述

AY0438 是 Microchip 公司生产的一种完整的 CMOS 显示驱动器,可在单片机或微处理器的控制下直接驱动 LCD 显示模块。它结构简单,使用方便。特别是在驱动 32 段 LCD 显示器方面,更能显示出它的精巧和方便。AY0438 只用三条控制线即可连续不断地向与它相连的 LCD 显示器输送驱动信号。该器件内含 32 位锁存器,它既可以对被显示的数据进行锁存,也可以锁存微处理器的状态或波形。

AY0438 能够驱动任何标准的 LCD 显示器,也可以驱动与标准 LCD 显示器相类似的显示模块。可显示用于 7 段、9 段、14 段或 16 段字符、十进制数的“+”和“-”号以及其它特殊字符。如果需要显示更多段位的数据或字符,则可将几个 AY0438 驱动器连起来使用。AY0438 内含时钟振荡器,可为系统提供精确的时钟输入信号,如果设计人员希望使用外部时钟,可在 LCD ϕ 和地之间串接一只电容器,根据电容值的大小来确定时钟频率的高低。

AY0438 有两种封装形式,一种采用 40 脚 PDIP 封装,另一种采用 44 脚 PLCC 形式。另外,为了方便在某些特殊情况下的使用, Microchip 公司还可以提供未经包装的 AY0438 器件。

AY0438 的主要特点如下:

- 可为任何结构的 32 段 LCD 提供驱动信号;
- 采用 CMOS 工艺技术;
- 电源电压适用范围宽;

- 工作电压低;
- 具有很高的噪声抑制能力;
- 温度范围宽;
- CMOS 信号与 TTL 输入信号完全兼容;
- 所有引脚均经过防静电的保护处理;
- 具有片内时钟振荡器;
- 控制简单,只用三条控制线即可控制芯片的全部功能。

2. 引脚功能及电气参数

2.1 引脚排列及功能

AY0438 采用 40 脚 PDIP 封装,其引脚排列如图 1 所示,表 1 所列为各引脚的说明。

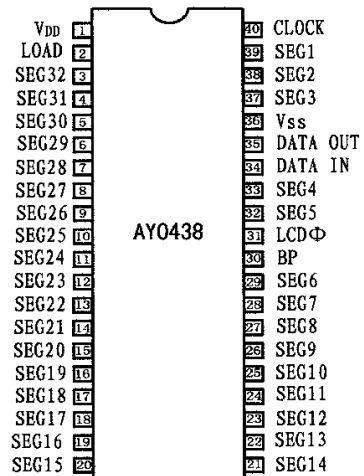


图 1 AY0438 的引脚排列

表 1 AY0438 的引脚说明

引脚号	引脚名	说明
1	V_{DD}	电源电压输入端
2	LOAD	锁存信号输入端
3~29, 32 33, 37~39	SEG1 - 32	驱动输出端,用于直接驱动 LCD 模块
30	BP	背景驱动输出端
31	LCD Φ	背景信号输入端
34	DATA IN	移位寄存器的数据输入端
35	DATA OUT	移位寄存器的数据输出端
36	V_{SS}	接地端
40	CLOCK	系统时钟输入端

2.2 电气参数

尽管 AY0438 有较宽的电气参数范围,但在设计时亦应考虑合理的电气参数,表 2 所列是 AY0438 在正常工作时的电气参数。

3. 操作过程

AY0438 LCD 显示驱动器是由 32bit 状态移位寄存器、锁存器、32 段驱动器和内部时钟发生器等电路组成的。图 2 是它的内部结构框图,现将几个主要引脚的工作原理加以说明。

3.1 DATA IN 和 CLOCK

AY0438 的内部移位寄存器是在时钟信号的下降沿进行移位和输出的。工作时,移位寄存器中的

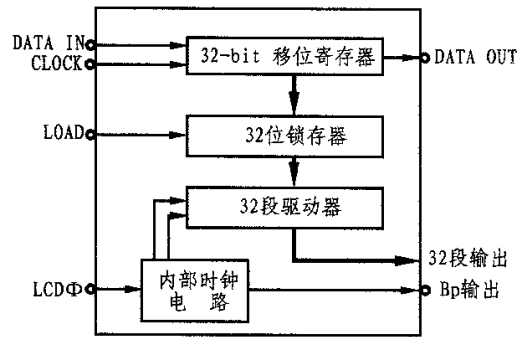


图 2 AY0438 的内部结构

数据在每个时钟信号的下降沿逻辑向左移一位。假如时钟为 32 位脉冲且 LOAD 信号已被激活,那么,第一个时钟脉冲时的数据输入信号将在 SEG 32 脚输出,而最后一个时钟脉冲时的数据信号将在 SEG 1 脚输出。这样,在每一个 32 位移位完成时,系统都会推出一个新的输出。当 DATA IN 为逻辑“1”时,如果对应的位置正在被激活或者显示,那么 Segment output 的输出采用背景输出方式进行。图 3 给出了 AY0438 的操作时序。

3.2 LOAD

当 LOAD 端的输入为逻辑“1”时,AY0438 将通过锁存器把移位寄存器中的数据锁存到与其控制所对应的器件段位上。一旦 LOAD 信号结束高电平,那么锁存器将结束锁存。这时的移位寄存器和段位驱动器将一致保持畅通地连接。

表 2 AY0438 的电气参数

名称	称号	最小值	典型值	最大值	单位	条件说明
电源电压	V_{DD}	3.0		8.5	V	
电源电流	I_{DD}		25	60	μA	LCD Φ OSC < 15kHz
			13	30	μA	LCD Φ OSC < 100kHz
输入高电平	V_{IH}	$0.5V_{DD}$		V_{DD}	V	
时钟和数据输入低电平	V_{IL1}	0		$0.1V_{DD}$	V	$3.0V \leq V_{DD} \leq 8.5V$
背景电流	I_L		0.01	± 10	μA	$V_{IN} = 0V$ 和 $-5.0V$
输入振荡电容	C_1			5.0	pF	$V_{DD} = +5.0V$
段输出电压	V_{OH}	$0.8V_{DD}$		V_{DD}	V	$I_{OH} = -100\mu A$
	V_{OL}	0		$0.1V_{DD}$	V	$I_{OL} = 100\mu A$
LCD Φ 输入高电平	V_{IN}	$0.9V_{DD}$		V_{DD}	V	
LCD Φ 输入低电平	V_{IL}	0		$0.1V_{DD}$	V	
LCD Φ 输入背景电流	I_L			10	μA	$V_{IN} = 0V$ 和 $+5.0V$
时钟速率	f	DC		1.5	MHz	
数据启动时间	t_{DS}	150			ns	从数据改变到时钟下降沿
数据保持时间	t_{DH}	50			ns	
LOAD 脉冲宽度	t_{PW}	175			ns	
数据输出延迟	t_{PD}			500	ns	$C_L = 55pF$

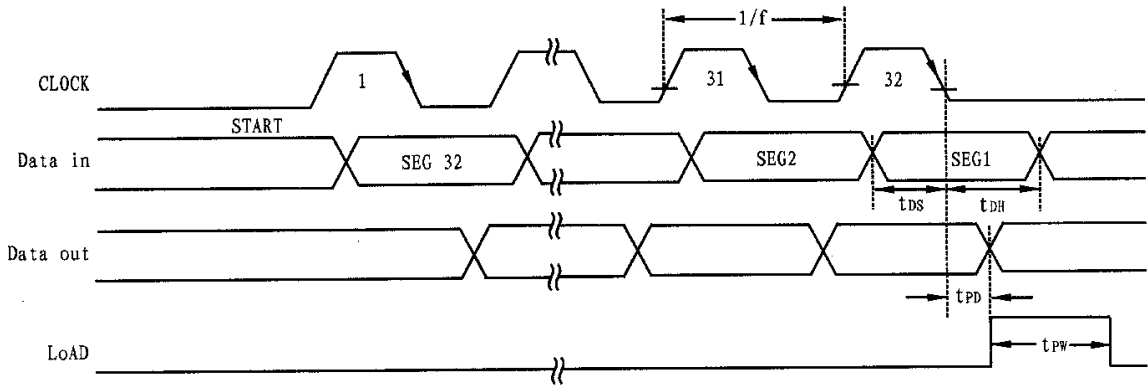


图3 操作时序

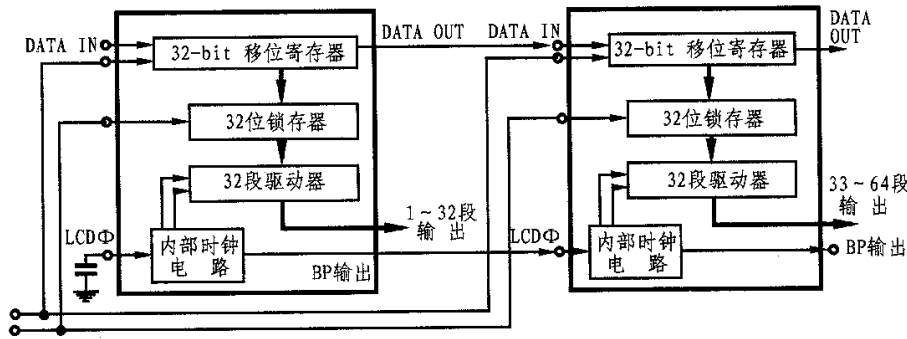


图4 两个器件的相互连接

3.3 LCDΦ

在外部 LCDΦ 与地之间接一个电容器可以产生用于驱动 LCDΦ 的外部扩展时钟信号,这样做的好处是能够通过改变电容器的值来得到不同的时钟频率。当然,也可以通过片内固有的时钟振荡器。无论是采用扩展时钟还是采用片内时钟,都能得到用于驱动背景电极的输出电压。利用片内时钟时, LCDΦ 端可以悬空不接。而采用扩展时钟时,其背景输出的状态与输入时钟有关。

在将多个 AY0438 连接在一起使用时,可在第一个 AY0438 的 LCDΦ 与地之间接上电容器,然后将其产生的背景输出接到下一个芯片的 LCDΦ 输入端,依次连接下去,最后一个芯片的背景输出用于驱动 LCD 显示模块的背景电极。图 4 是两片 AY0438 的连接图。

4. 基于 PIC16C71 单片机的应用电路及编程

为了防止和避免数据输入到负载之间因阻塞或空闲等意外情况而引起系统工作的不正常,在时钟信号的下降沿,AY0438 的 DATA IN 和 LOAD 信号的状态都不改变。图 5 是 AY0438 在 PIC16C71 单片机的控制下驱动 4 位 7 段 LCD 显示器的典型应用电路。连接时,将

CLK、DATA IN 和 LOAD 分别接到 PIC16C71 单片机的 PB0、PB1 和 PB2 上,把 SEG1~7、SEG9~15、SEG17~23 和 SEG25~31 分别依次连接到 4 位 7 段 LCD 显示模块的第一、第三、第二和第四位上。另外,在 PIC16C71 的 PB7 端连接一个微动开关,用于检查系统是否阻塞。如果发生阻塞,通过 PB7 可以重新设定 LCD 模块上从第一位到最后一位的值。下面是对 PIC16C71 单片机的编程:

```
list P=16c71, f=inhx8m
```

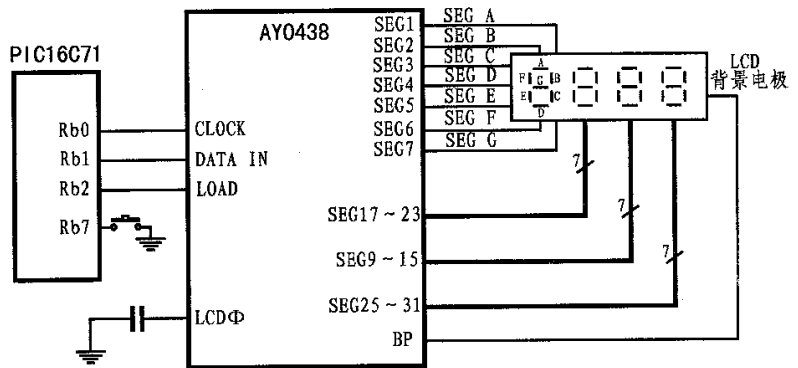


图5 基于单片机控制的显示电路

```

;
MSD equ 0X20
THRSD equ 0X21
SCNDS equ 0X22
LSD equ 0X23
Count equ 0X24
Temp equ 0X25
PORTB equ 0X06
# define CLK PORTB, 0
# define DATAIN PORTB, 1
# define LOAD PORTB, 2
# define UPDTELCD PORTB, 7
W equ 0
STATUS equ 0X03
C equ 0
RP0 equ 5
OPTION equ 0X81
RBPU equ 7
PCL equ 0X02
PCLATH equ 0X0A
;
;
org 0
goto start
org 0x10
;
;工作时,下面的真值表应该始终位于程序 page 0 的位置
;
DecodeValue
    Addwfc pcl
    retlw b'00111111' ;decode for 0
    retlw b'00000110' ;decode for 1
    retlw b'01011011' ;decode for 2
    retlw b'01001111' ;decode for 3
    retlw b'01100110' ;decode for 4
    retlw b'01101101' ;decode for 5
    retlw b'01111101' ;decode for 6
    retlw b'00000111' ;decode for 7
    retlw b'01111111' ;decode for 8
    retlw b'01101111' ;decode for 9
;
;
start
    clrf PORTB
    bsf STATUS, RP0 ;设定 PB0, 1
                    ;和 2
                    movlw B'11111000' ;
                    movwf PORTB ;
                    bcf OPTION, RBPU ;使能
                    bcf STATUS, RP0
                    wait
                    btfsc UPDTELCD ;重置开关为低时查看
                    goto wait ;为等待
                    bcf LOAD ;LOAD 异常处理
                    movf LSD, w ;设置显示器最后一位
                    clrf PCLATH ;PCH=0
                    call DecodeValue ;解码
                    call Send8 ;连续输出段值
                    movf SCNDS, w ;设定第二位
                    call DecodeValue ;解码
                    call Send8 ;连续输出
                    movf THRSD, w ;设定第三位
                    call DecodeValue ;解码
                    call Send8 ;连续输出
                    movf MSD, w ;设定第一位
                    call DecodeValue ;解码
                    call Send8 ;连续输出
                    bsf LOAD ;捕获 LOAD 线
                    bcf LOAD ;使能锁存
                    KeyReleased
                    Btfss UPDTELCD ;等待微动开关释放
                    Goto KeyReleased ;
                    Goto wait ;重复循环
;
;Send8 表示将 W 寄存器中的 8 位输出
Send8
    Movwf temp ;存入 temp
    Movlw .8 ;计数
    Movwf count ;到 8 位
Sendloop
    Bcf DATAIN ;设定 DATA IN 为低
    Rrf temp ;旋转换位
    Btfsc STATUS, C ;清理位值后跳转
    Bsf DATAIN ;设定其它数据位
    Bsf CLK ;锁存时钟
    Bcf CLK ; /
    Decfsz count ;查看 8 位是否完成
    Goto sendloop ;未完成则重新进行
    Return ;下一循环
End
                    
```