

TFT 液晶驱动实验

基于 EasyARM1138

AN01010101

V0.00

Date: 2008/12/10

产品应用笔记

类别	内容
关键词	TFT-LCD、LM3S1138
摘 要	基于 EasyARM1138 的 TFT 液晶驱动实验

修订历史

版本	日期	原因
V0.00	2008/12/10	创建文档

销售与服务网络（一）

广州周立功单片机发展有限公司

地址：广州市天河区北路 689 号光大银行大厦 12 楼 F4
邮编：510630
电话：(020)38730916 38730917 38730972 38730976 38730977
传真：(020)38730925
网址：www.zlgmcu.com



广州专卖店

地址：广州市天河区新赛格电子城 203-204 室
电话：(020)87578634 87569917
传真：(020)87578842

南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室
电话：(025)83613221 83613271 83603500
传真：(025)83613271

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座
1207-1208 室（中发电子市场斜对面）
电话：(010)62536178 62536179 82628073
传真：(010)82614433

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦
（赛格电子市场）1611 室
电话：(023)68796438 68796439
传真：(023)68796439

杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室
电话：(0571) 28139611 28139612 28139613
28139615 28139616 28139618
传真：(0571) 28139621

成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室
（磨子桥立交西北角）
电话：(028)85439836 85437446
传真：(028)85437896

深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4
楼 D 室
电话：(0755)83781788（5 线）
传真：(0755)83793285

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室
（华中电脑数码市场）
电话：(027)87168497 87168297 87168397
传真：(027)87163755

上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室
电话：(021)53083452 53083453 53083496
传真：(021)53083491

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室
电话：(029)87881296 83063000 87881295
传真：(029)87880865

销售与服务网络（二）

广州致远电子有限公司

地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼

邮编：510660

传真：(020)38601859

网址：www.embedtools.com （嵌入式系统事业部）

www.embedcontrol.com （工控网络事业部）

www.ecardsys.com （楼宇自动化事业部）



技术支持：

CAN-bus：

电话：(020)22644381 22644382 22644253

邮箱：can.support@embedcontrol.com

iCAN 及模块：

电话：(020)28872344 22644373

邮箱：ican@embedcontrol.com

MiniARM：

电话：(020)28872684 28267813

邮箱：miniarm.support@embedtools.com

以太网及无线：

电话：(020)22644380 22644385 22644386

邮箱：wireless@embedcontrol.com

ethernet.support@embedcontrol.com

编程器：

电话：(020)22644371

邮箱：programmer@embedtools.com

分析仪器：

电话：(020)22644375 28872624 28872345

邮箱：tools@embedtools.com

ARM 嵌入式系统：

电话：(020)28872347 28872377 22644383 22644384

邮箱：arm.support@zlgmcu.com

楼宇自动化：

电话：(020)22644376 22644389 28267806

邮箱：mjs.support@ecardsys.com

mifare.support@zlgmcu.com

销售：

电话：(020)22644249 22644399 22644372 22644261 28872524

28872342 28872349 28872569 28872573 38601786

维修：

电话：(020)22644245

目 录

1. 适用范围.....	1
2. 原理概述.....	2
2.1 硬件接口定义.....	2
2.2 TFT-LCD底层驱动编写.....	4
2.3 TFT-LCD高级API函数编写	7
3. 开发环境.....	8
4. 技术实现.....	9
5. 参考资料.....	11
6. 免责声明.....	12

1. 适用范围

EasyARM 1138

2. 原理概述

EasyARM1138 是专门针对广大电子信息专业在校大学生而设计的一款基于ARM Cortex-M3^[1]先进内核的高性能、低价格开发板，虽然实验板没有标配显示设备，但给其扩展一块TFT彩色液晶显示器也是一件非常容易的事，使其更方便应用于教学、毕业设计、电子竞技等场合。

TFT-LCD 液晶显示屏是薄膜晶体管型液晶显示屏，具有高速度、高亮度、高对比度等特点，是目前最好的 LCD 彩色显示设备之一，其效果接近 CRT 显示器，是现代高科技电子产品中常用的屏幕显示设备之一。由于具体的 TFT 液晶显示器型号和接口比较繁杂，下面就以常见的 8 位并行接口（5 根控制线+8 根数据线）的读写时序为例，编写 TFT-LCD 的驱动程序。

2.1 硬件接口定义

假设 EasyARM1138 与 TFT-LCD 的连线如图 2.1 所示，则与 TFT-LCD 接口相关的宏定义如程序清单 2.1 所示

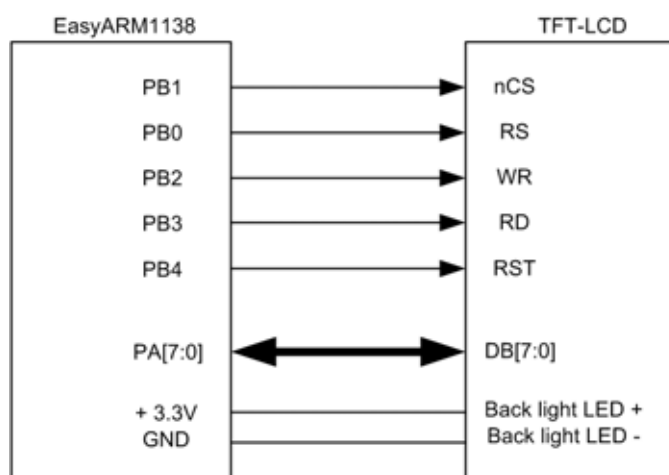


图 2.1 硬件连接示意图

程序清单 2.1 与 TFT-LCD 接口相关的宏定义

```

/*****
硬件相关的定义
*****/

#define LCD_PERI          SYSCTL_PERIPH_GPIOA | SYSCTL_PERIPH_GPIOB
// CS 定义 PB1
#define LCD_CS_BASE       GPIO_PORTB_BASE
#define LCD_CS_PIN        GPIO_PIN_1
#define LCD_CS_BASE_PIN   LCD_CS_BASE,LCD_CS_PIN
// RS 定义 PB0
#define LCD_RS_BASE       GPIO_PORTB_BASE
#define LCD_RS_PIN        GPIO_PIN_0
#define LCD_RS_BASE_PIN   LCD_RS_BASE,LCD_RS_PIN
// WR 定义 PB2
#define LCD_WR_BASE       GPIO_PORTB_BASE
#define LCD_WR_PIN        GPIO_PIN_2

```

```
#define LCD_WR_BASE_PIN    LCD_WR_BASE,LCD_WR_PIN
// RD 定义  PB3
#define LCD_RD_BASE        GPIO_PORTB_BASE
#define LCD_RD_PIN          GPIO_PIN_3
#define LCD_RD_BASE_PIN    LCD_RD_BASE,LCD_RD_PIN
// RST 定义  PB4
#define LCD_RST_BASE        GPIO_PORTB_BASE
#define LCD_RST_PIN          GPIO_PIN_4
#define LCD_RST_BASE_PIN    LCD_RST_BASE,LCD_RST_PIN
// 数据端口定义  PA
#define DATA_PORT_BASE      GPIO_PORTA_BASE
#define DATA_PORT_PIN        0xFF
#define DATA_PORT_BASE_PIN  DATA_PORT_BASE,DATA_PORT_PIN

// 用于位操作的定义
#define LCD_RS                HWREG(LCD_RS_BASE + (GPIO_O_DATA + (LCD_RS_PIN << 2)))
#define LCD_CS                HWREG(LCD_CS_BASE + (GPIO_O_DATA + (LCD_CS_PIN << 2)))
#define LCD_WR                HWREG(LCD_WR_BASE + (GPIO_O_DATA + (LCD_WR_PIN << 2)))
#define LCD_RD                HWREG(LCD_RD_BASE + (GPIO_O_DATA + (LCD_RD_PIN << 2)))
#define LCD_RST               HWREG(LCD_RST_BASE + (GPIO_O_DATA + (LCD_RST_PIN << 2)))
#define DATA_POR             HWREG(DATA_PORT_BASE + (GPIO_O_DATA + (DATA_PORT_PIN << 2)))
```

接口初始化程序如程序清单 2.2 所示。

程序清单 2.2 接口初始化函数 interfaceInit()

```
void interfaceInit(void) /* 初始化函数 */
{
    SysCtlPeriEnable(LCD_PERI); /* 使能相应的 GPIO */
    GPIOPinTypeOut(LCD_CS_BASE_PIN); /* 设置相应的控制管脚为输出 */
    GPIOPinTypeOut(LCD_WR_BASE_PIN); /* 设置相应的控制管脚为输出 */
    GPIOPinTypeOut(LCD_RS_BASE_PIN); /* 设置相应的控制管脚为输出 */
    GPIOPinTypeOut(LCD_RD_BASE_PIN); /* 设置相应的控制管脚为输出 */
    GPIOPinTypeOut(LCD_RST_BASE_PIN); /* 设置相应的控制管脚为输出 */
    GPIOPinTypeOut(DATA_PORT_BASE_PIN); /* 设置相应的数据管脚为输出 */

    LCD_RS = 0xFF; /* 初始化输出为高电平 */
    LCD_CS = 0xFF; /* 初始化输出为高电平 */
    LCD_WR = 0xFF;; /* 初始化输出为高电平 */
    LCD_RD = 0xFF; /* 初始化输出为高电平 */
    LCD_RST = 0xFF; /* 初始化输出为高电平 */
    DATA_PORT = 0xFF;

    LCD_RST = 0x00; /* 拉低复位引脚 */
    SysCtlDelay(10 * SysCtlClockGet( ) / 3000);
    LCD_RST = 0xFF; /* 拉高复位引脚 */
}
```



```

SysCtlDelay(10 * SysCtlClockGet( ) / 3000);
}

```

2.2 TFT-LCD 底层驱动编写

向地址宽度为 8 位、数据宽度为 8 位的寄存器读写操作的常见时序如图 2.2 所示。

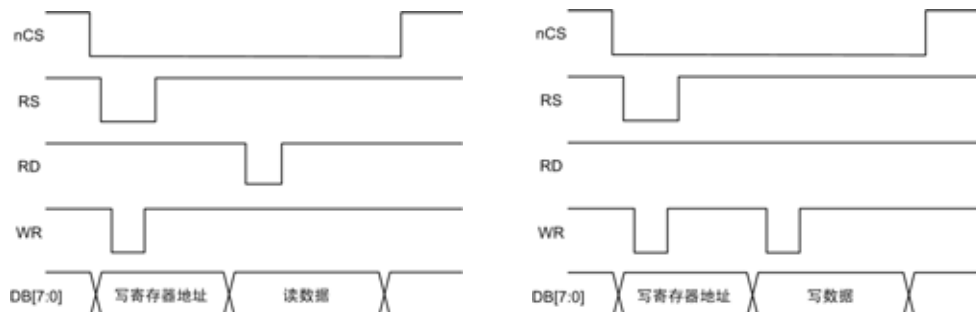


图 2.2 地址宽度为 8 位、数据宽度为 8 位的寄存器读写时序图

根据图 2.2 的时序图可知，对地址宽度为 8 位、数据宽度为 8 位的寄存器的操作只需要三个函数，即“8 位地址函数”、“8 位数据写函数”和“8 位数据读函数”，这三个函数分别如程序清单 2.3 所示。

程序清单 2.3 地址宽度为 8 位、数据宽度为 8 位的寄存器基本操作函数

```

void __writeRegAddr8(uchar ucAddr)                                /* 写 8 位寄存器地址函数 */
{
    LCD_CS = 0x00;                                                /* 拉低 CS */
    LCD_RS = 0x00;                                                /* 拉低 RS */
    DATA_PORT = ucAddr;                                          /* 写地址数据 */
    LCD_WR = 0x00;                                                /* 拉低 WR */
    LCD_WR = 0xFF;                                                /* 拉高 WR */
    LCD_CS = 0xFF;                                                /* 拉高 CS */
}

void __writeData8(uchar pd)                                       /* 写 8 位数据函数 */
{
    LCD_CS = 0x00;                                                /* 拉低 CS */
    LCD_RS = 0xFF;                                                /* 拉高 RS */
    DATA_PORT = pd;                                              /* 写数据 */
    LCD_WR = 0x00;                                                /* 拉低 WR */
    LCD_WR = 0xFF;                                                /* 拉高 WR */
    LCD_CS = 0xFF;                                                /* 拉高 CS */
}

unsigned char __readData8(void)
{
    unsigned char ucReturn = 0;
    GPIOPinTypeIn(DATA_PORT_BASE_PIN);                          /* 设置为输入端口 */
    LCD_CS = 0x00;                                                /* 拉低 CS */
}

```

```

LCD_RS = 0xFF;                                /* 拉高 RS */
LCD_RD = 0x00;                                /* 拉低 RD */
LCD_RD = 0xFF;                                /* 拉高 RD */
ucReturn = DATA_PORT;                        /* 读入数据 */
LCD_CS = 0xFF;                                /* 拉高 CS */
GPIOPinTypeOut(DATA_PORT_BASE_PIN);          /* 还原为输出端口 */
return ucReturn;                              /* 返回结果 */
}
    
```

向地址宽度为 16 位、数据宽度为 16 位的寄存器读操作的常见时序如图 2.3 所示，写操作时序如图 2.4 所示。

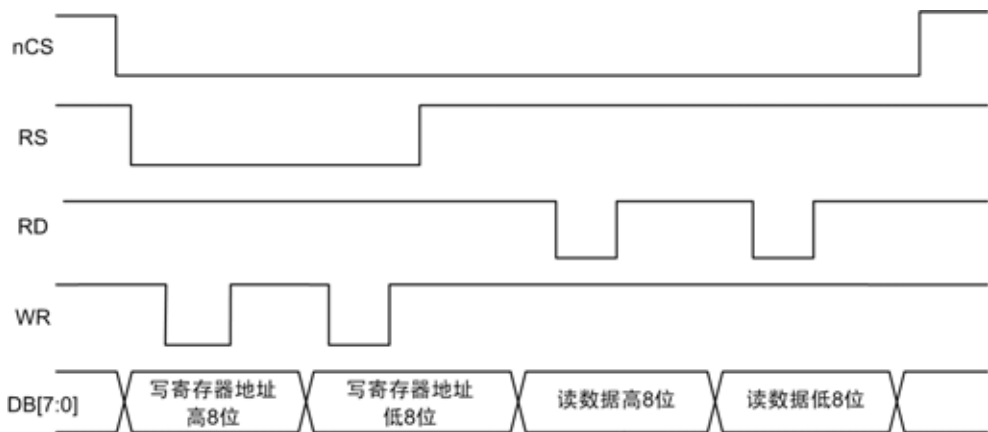


图 2.3 地址宽度为 16 位、数据宽度为 16 位的寄存器读时序图

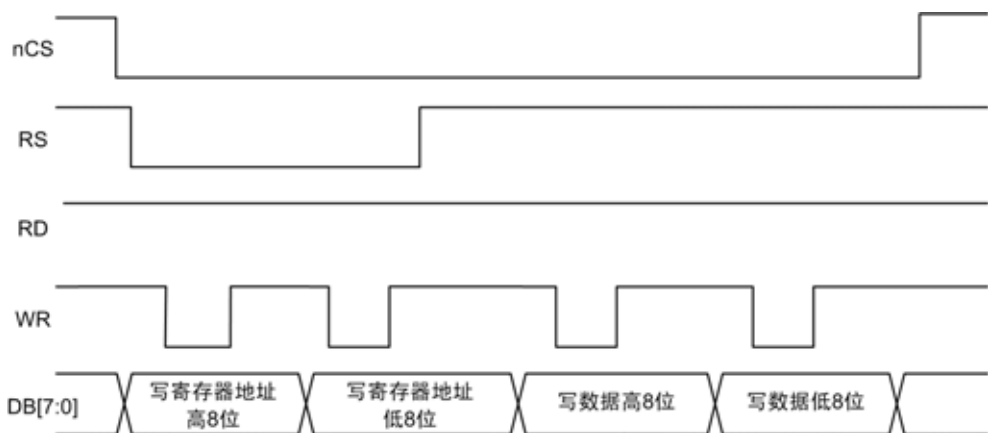


图 2.4 地址宽度为 16 位、数据宽度为 16 位的寄存器写时序图

分析图 2.3 和图 2.4 可知，对地址宽度为 16 位、数据宽度为 16 位的寄存器的读写操作也只需要三个函数即可，这三个函数分别为“16 位地址写函数”、“16 位数据读函数”、“16 位数据写函数”，如所示。

程序清单 2.4 地址宽度为 16 位、数据宽度为 16 位的寄存器基本操作函数

```

// 16 位地址写函数
void __writeRegAddr16(unsigned int uiAddr)
{
    
```

```

LCD_CS = 0x00;                                /* 拉低 CS */
LCD_RS = 0x00;                                /* 拉低 RS */
DATA_PORT = (unsigned char)(uiAddr >> 8);     /* 写地址高 8 位数据 */
LCD_WR = 0x00;                                /* 拉低 WR */
LCD_WR = 0xFF;                                /* 拉高 WR */
DATA_PORT = (unsigned char)uiAddr;            /* 写地址低 8 位数据 */
LCD_WR = 0x00;                                /* 拉低 WR */
LCD_WR = 0xFF;                                /* 拉高 WR */
//LCD_RS = 0xFF;                              /* 拉高 RS */
LCD_CS = 0xFF;                                /* 拉高 CS */
}

// 16 位数据读函数
unsigned int __readRegData16(void)
{
    unsigned int uiRetrun = 0;
    GPIOPinTypeIn(DATA_PORT_BASE_PIN);        /* 设置为输入端口 */
    LCD_CS = 0x00;                            /* 拉低 CS */
    LCD_RS = 0xFF;                            /* 拉高 RS */
    LCD_RD = 0x00;                            /* 拉低 RD */
    LCD_RD = 0xFF;                            /* 拉高 RD */
    uiRetrun = (unsigned char)DATA_PORT;       /* 读入高 8 位数据 */
    LCD_RD = 0x00;                            /* 拉低 RD */
    LCD_RD = 0xFF;                            /* 拉高 RD */
    uiRetrun = (uiRetrun << 8) + (unsigned char)DATA_PORT; /* 读入低 8 位数据 */
    LCD_CS = 0xFF;                            /* 拉高 CS */
    GPIOPinTypeOut(DATA_PORT_BASE_PIN);       /* 还原为输出端口 */
    return uiRetrun;
}

// 16 位数据写函数
void __writeData16(uint val)                  /* 写 16 位数据函数 */
{
    LCD_CS = 0x00;
    LCD_RS = 0xFF;                            /* 拉高 RS */
    DATA_PORT = (uchar)(val>>8);            /* 写高 8 位数据 */
    LCD_WR = 0x00;                            /* 拉低 WR */
    LCD_WR = 0xFF;                            /* 拉高 WR */
    DATA_PORT = (uchar)val;                  /* 写低 8 位数据 */
    LCD_WR = 0x00;                            /* 拉低 WR */
    LCD_WR = 0xFF;                            /* 拉高 WR */
    LCD_CS = 0xFF;                            /* 拉高 CS */
}

```

2.3 TFT-LCD 高级 API 函数编写

通过以上六个基本操作函数的有机组合，便可轻松完成TFT-LCD的顶层API函数的编写，如向液晶写命令、设置窗口、设置坐标、画点、画线、填充颜色、绘图等复杂的操作函数，由于这些操作涉及的寄存器等与具体的液晶型号紧密相关，所以在这里就不作具体描述了，读者可以参考配套的例程和相关的Datasheet^[2, 3]。

3. 开发环境

TFT-LCD：LTM024D130

硬件平台：EasyARM1138 套件、杜邦线若干

软件开发环境：IAR Embedded Workbench 5.11 / Keil uVision3、Stellaris Peripheral Driver
Library Edition 3223^[3]

4. 技术实现

按照图 2.1 将液晶(配套例程在 LTM024D130 TFT 液晶上测试通过)的控制线及数据线通过杜邦线连接到 EasyARM1138 上, 如图 4.1 所示, 将实验程序下载到 MCU 上后, 复位芯片, 即可看到液晶交替显示文字及图片, 效果如图 4.2 和图 4.3 所示。



图 4.1 测试电路实物

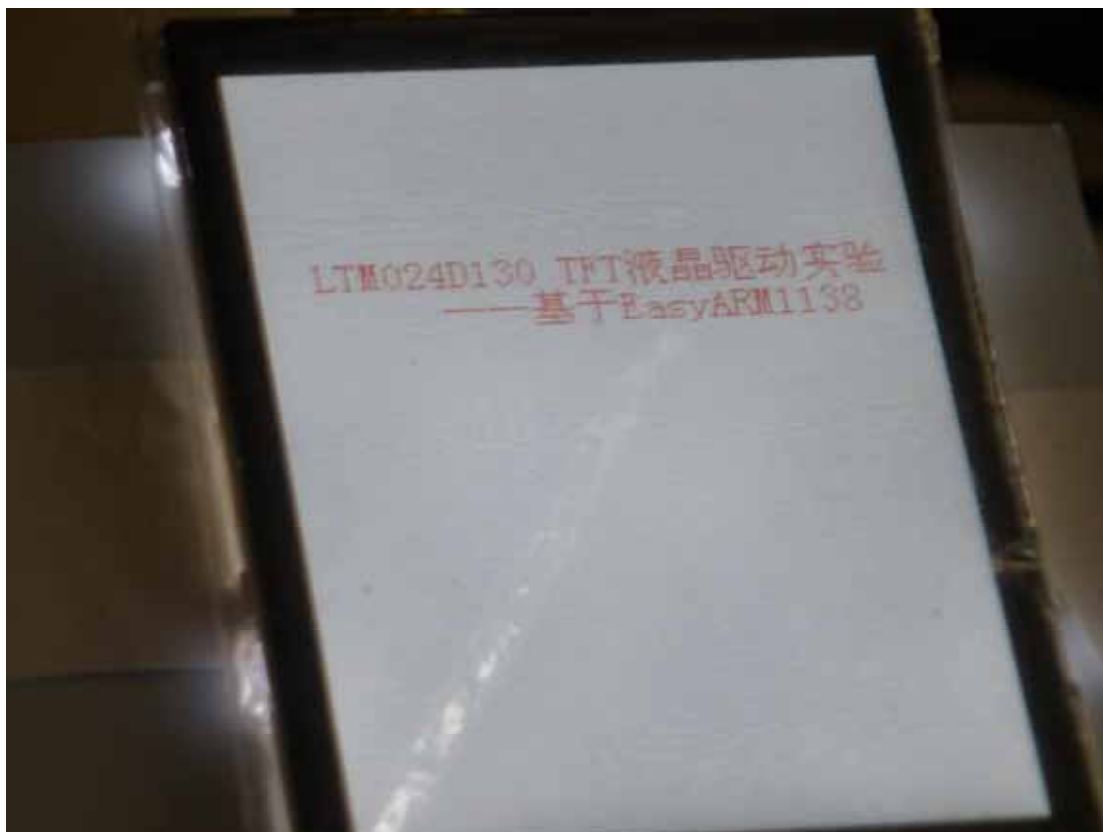


图 4.2 例程运行效果 1



图 4.3 例程运行效果 2

5. 参考资料

- 1 Arm公司. Cortex-M3技术参考手册 (Cortex-M3_trm.Pdf) . ARM公司, 2005.
- 2 Luminarymico公司. LM3S1138数据手册 (Datasheet-Lm3s1138.Pdf) . LuminaryMico公司, 2008.
- 3 Luminarymicro公司. 外设驱动库使用手册 (Pdl-Lm3s-Ug-3223.Pdf) . LuminaryMicro公司, 2008.

6. 免责声明

本应用笔记版权均属广州致远电子有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将受到国家法律的严厉制裁。

您若需要我公司产品及相关信息，请及时与我们联系，我们将热情接待。

广州致远电子有限公司保留在任何时候修订本用户手册且不需通知的权利。