

## 第22章 LCD驱动程序

---

一个 LCD 驱动程序支持一个具体系列的 LCD 控制器，而所有的 LCD 都配备一个或多个这些控制器。驱动程序本质上是通用的，意思是它可以通过修改配置文件 LCDConf.h 来进行配置。这些文件包含所有可配置的选项用于驱动程序，以及多重的定义用于硬件如何访问及控制器如何与 LCD 连接。

本章提供 $\mu$ C/GUI的LCD控制器的概述。它针对每一个驱动程序说明以下这些内容：

- 哪一个 LCD 控制器能被访问，及支持的颜色深度和接口类型。
- 额外的 RAM 需求。
- 附加函数。
- 如何访问硬件。
- 指定配置开关。
- 特定的 LCD 控制器的特别需求。

## 22.1 支持的LCD控制器及各自的驱动程序

下表列出了驱动程序及那些控制器支持它们：

驱动程序	宏 LCD_CONTROLLER 的值	LCD 控制器	支持的位/像素 (bps)
LCD07X1	711	Samsung KS0711	2
	741	Samsung KS0741	
LCD13XX	1352	Epson SED1352, S1D13502	1, 2, 4, 8, 16
	1354	Epson SED1354, S1D13504	
	1356	Epson SED1356, S1D13506	
	1374	Epson SED1374, S1D13704	
	1375	Epson SED1375, S1D13705	
	1376	Epson SED1376, S1D13706	
	1386	Epson SED1386, S1D13806	
	1300	Epson S1D13A03, S1D13A04	
LCD159A	0x159A	Epson SED159A	8
LCD15E05	0x15E05	Epson S1D15E05	2
LCD15XX	713	Samsung KS0713	1
	1560	Epson SED1560	
	1565	Epson SED1565	
	1566	Epson SED1566	
	1567	Epson SED1567	
	1568	Epson SED1568	
	1569	Epson SED1569	
	1575	Epson SED1575	
LCD6642X	66420	Hitachi HD66420	2
	66421	Hitachi HD66421	
LCDMem	0	无控制器，写入 RAM (单色显示)	2
LCDMemC	0	无控制器，写入 RAM (彩色显示)	3, 6
LCDPage1bpp	8811	Philips PCF8810, PCF8811	1
LCDSLin	1330	Epson SED1330	1
	1335	Epson SED1335	
	6963	Toshiba T6963	

## 选择一个驱动程序

如第 20 章“底层配置”中所描述的那样，宏 LCD\_CONTROLLER 定义所使用的 LCD 控制器。一个控制器由上表列出它的适当的值指定。

下面部分分别讨论每一个有效的驱动程序。

## 22.2 LCD07X1

### 支持的硬件

#### 控制器

该驱动程序在下列控制器身上测试通过：

- Samsung KS0711
- Samsung KS0741

应该能假设它也能够任何与这两种控制器结构相类似的控制器上工作。

#### 每像素的位

支持颜色深度为 2 bpp。

#### 接口

芯片支持三种类型接口：

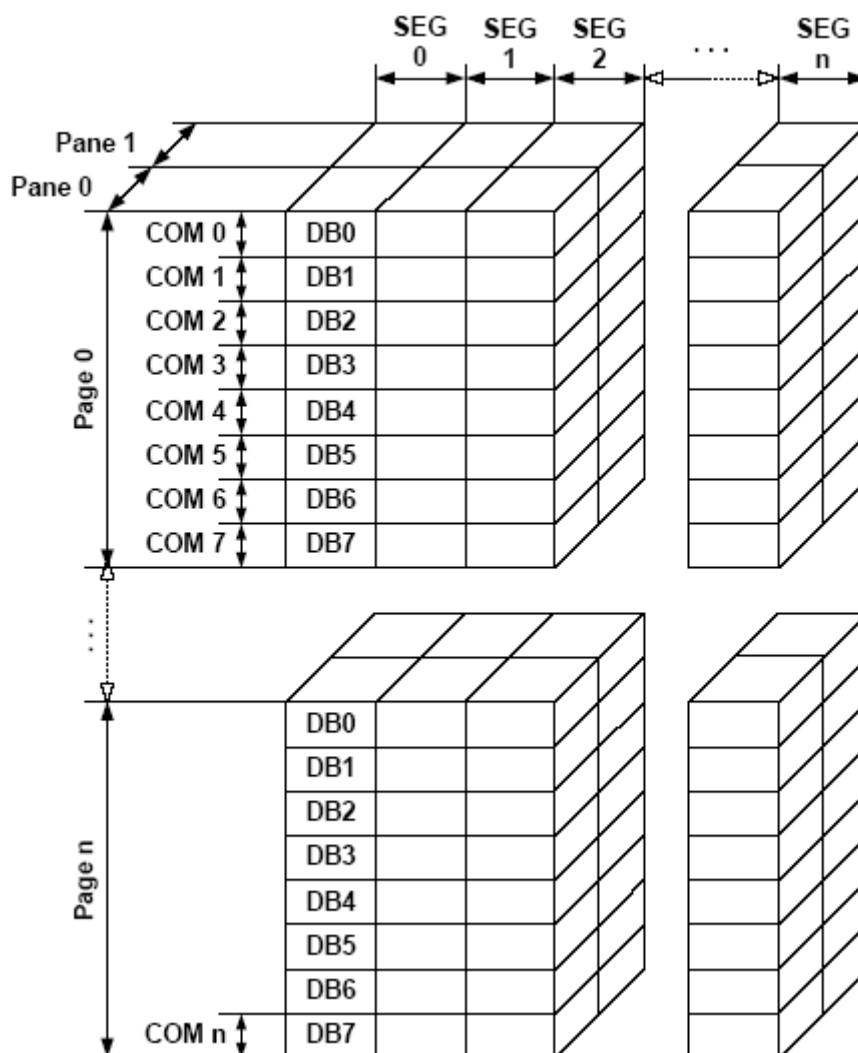
8 位并行（简单总线）接口

4 脚串行外围设备接口（SPI）

3 脚 SPI。

当前版本的驱动程序支持并行或 4 脚 SPI 模式。

## 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。对于每个像素，显示存储器被分成两个方框。每个像素较低的位存储在 pane0 中而较高的位存储在 pane 1 中。

### 驱动程序的额外 RAM 要求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果高速缓存未使用，则没有额外的 RAM 需求。

推荐使用这些驱动程序时，一起使用一个数据高速缓存，以获得更快的 LCD 访问速度。用于高速缓存的内存的数值可以由以下公式计算：

RAM 的大小 (字节) = ( LCD\_YSIZE+7)/8\*LCD\_XSIZE\*2

## 附加的驱动函数

### LCD\_L0\_ControlCache

有关这个函数的信息, 请参阅第 23 章“LCD 驱动程序 API”。

## 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏:

### 并行模式

宏	说明
<a href="#">LCD_INIT_CONTROLLER</a>	初始化 LCD 控制器序列。
<a href="#">LCD_READ_A0</a>	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
<a href="#">LCD_READ_A1</a>	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
<a href="#">LCD_WRITE_A0</a>	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
<a href="#">LCD_WRITE_A1</a>	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

### 串行模式

宏	说明
<a href="#">LCD_INIT_CONTROLLER</a>	初始化 LCD 控制器序列。
<a href="#">LCD_WRITE_A0</a>	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
<a href="#">LCD_WRITE_A1</a>	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。
<a href="#">LCD_WRITEM_A0</a>	A 线 (A-line) 为低电平时向 LCD 控制器写入多个字节。
<a href="#">LCD_WRITEM_A1</a>	A 线 (A-line) 为高电平时向 LCD 控制器写入多个字节。

## 附加的配置开关

无。

## 某些LCD控制器的特定要求

无。

## 22.3 LCD13XX

### 支持的硬件

#### 控制器

该驱动程序通过了下列 LCD 控制器的测试：

- Epson SED1352, S1D13502
- Epson SED1354, S1D13504
- Epson SED1356, S1D13506
- Epson SED1374, S1D13704
- Epson SED1375, S1D13705
- Epson SED1376, S1D13706
- Epson SED1386, S1D13806
- Epson S1D13A03, S1D13A04

应该能假设它也能够任何与以上控制器结构相类似的控制器上工作。

#### 每像素的位

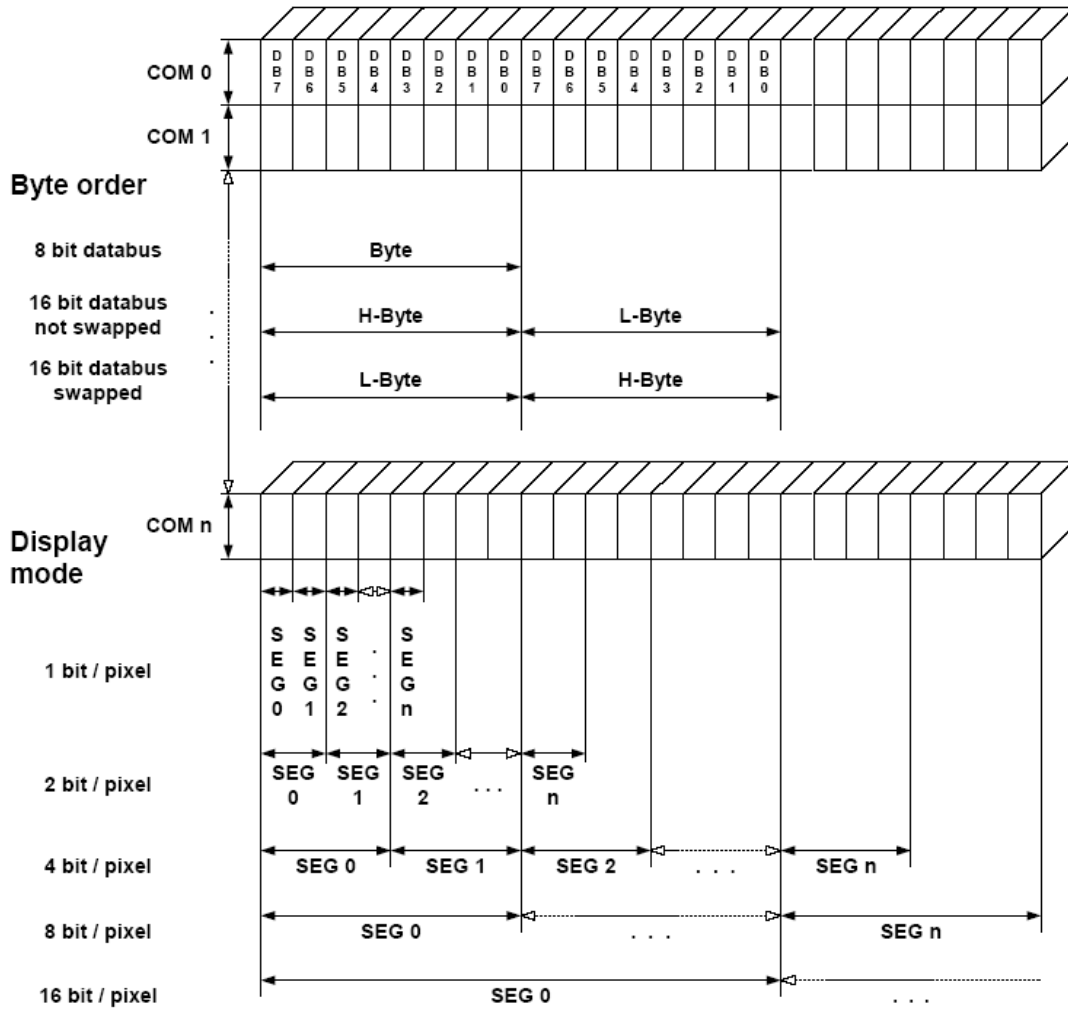
支持的颜色深度是 1, 2, 4, 8 和 16 bpp。

#### 接口

由这个驱动程序支持的芯片，能够以 8/16 位并行（完全总线）模式进行连接。

驱动程序同时支持两种接口。请参阅该各自的 LCD 控制器手册以确定是否你的芯片能够以 8 位模式连接。

### 显示屏数据 RAM 的结构



上图展示了依据颜色深度，显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

### 驱动程序的附加 RAM 要求

无。

### 附加的驱动函数

无。

### 硬件配置

该驱动程序需要使用一个如第20章“低层配置”所描述的完全的总线接口来进行硬件访

问。下表列出了必须为硬件访问所定义的宏：

宏	说明
LCD_INIT_CONTROLLER	初始化 LCD 控制器序列。
LCD_READ_MEM	读控制器的图像存储器的内容。
LCD_READ_REG	读控制器的一个配置寄存器的内容。
LCD_WRITE_MEM	向控制器的图像存储器（显示数据随机 RAM）写入数据。
LCD_WRITE_REG	向控制器的一个配置寄存器写入数据。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说明
LCD_BUSWIDTH	选择 LCD 控制器/ CPU 接口的总线宽度（8/16）。默认值是 16。
LCD_ENABLE_MEM_ACCESS	切换 M/R 信号到存储器访问。仅仅用于 SED1356 和 SED1386 LCD 控制器。
LCD_ENABLE_REG_ACCESS	切换 M/R 信号到寄存器访问。仅仅用于 SED1356 和 SED1386 LCD 控制器。
LCD_SWAP_BYTE_ORDER	当使用一个 16 位总线接口时，反转 CPU 和 LCD 控制器间的头端模式（高低字节交换）。
LCD_USE_BITBLT	如果设置为 0，禁止 BitBLT 引擎。如果设置为 1（缺省值），驱动程序将使用全部有效的硬件加速。
LCD_ON	LCD 切换到“开”的功能置换宏。
LCD_OFF	LCD 切换到“关”的功能置换宏。

## 某些LCD控制器SED1386或者S1 D13806的特殊要求

### LCD\_SWAP\_RB

该配置开关 LCD\_SWAP\_RB（交换红和蓝色部分）必须通过向 LCDConf.h 插入下面一行而激活（设置为 1）：

```
#define LCD_SWAP_RB (1)          /* 必须设置 */
```

### LCD\_INIT\_CONTROLLER

当写或者修改初始化宏时，要考虑下列问题：

初始化嵌入 SDRAM，寄存器 20 的第 7 位（SDRAM 初始化位）必须设置为 1（至少在复位



200 $\mu$ s 后)。

• 当 SDRAM 初始化位设置后，实际的初始化序列发生在第一个 SDRAM 刷新周期。该初始化序列需要大约 16 个 MCLK 来完成，并且在初始化在进行中的时候不能有存储器访问的操作。

更多信息，请参见 LCD 控制器技术资料。

#### LCD\_READ\_REG, LCD\_WRITE\_REG

为了使 BitBLT (位块传送) 引擎工作，偏移量的数据类型必须是无符号长整数 (unsigned long)。这要用配置宏 LCD\_READ\_REG 和 LCD\_WRITE\_REG 作如下所示进行设置：

```
#define LCD_READ_REG(Off) *((volatile U16*) (0x800000+(((U32) (Off)) <<1)))  
#define LCD_WRITE_REG(Off,Data) *((volatile U16*) (0x800000+(((U32) (Off)) <<1)))=Data
```

## 22.4 LCD159A

### 支持的硬件

#### 控制器

该驱动程序已经通过下列 LCD 控制器的测试：

- Epson SED159A

应假定它也能工作在任何与些结构相类似的控制器上。

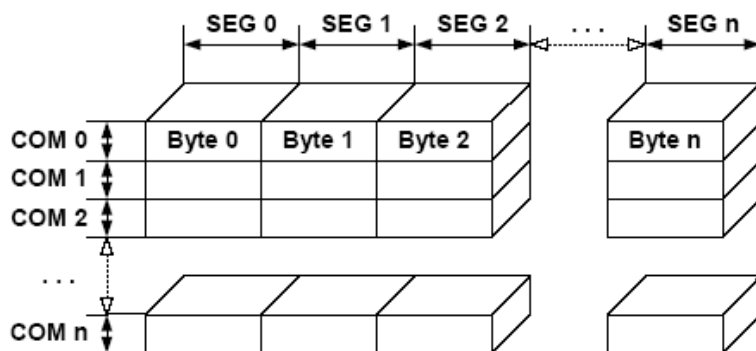
#### 每像素的位

支持 8bpp 颜色深度。

#### 接口

该驱动程序支持 8 位并行 (简单总线) 接口。

#### 显示屏数据 RAM 的结构



上图展示了显示内存和 LCD 的 SEG 和 COM 线之间的关系。

### 额外的 RAM 要求

无。

### 附加的驱动函数

无。

### 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说明
<code>LCD_INIT_CONTROLLER</code>	初始化 LCD 控制器序列。
<code>LCD_READ_A0</code>	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
<code>LCD_READ_A1</code>	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
<code>LCD_WRITE_A0</code>	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
<code>LCD_WRITE_A1</code>	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

### 附加的配置开关

无。

### 某些 LCD 控制器的特殊要求

无。

## 22.5 LCD15E05

### 支持的硬件

#### 控制器

该驱动程序已经通过下列 LCD 控制器的测试：

- Epson S1D15E05

应假定它也能工作在任何与些结构相类似的控制器上。

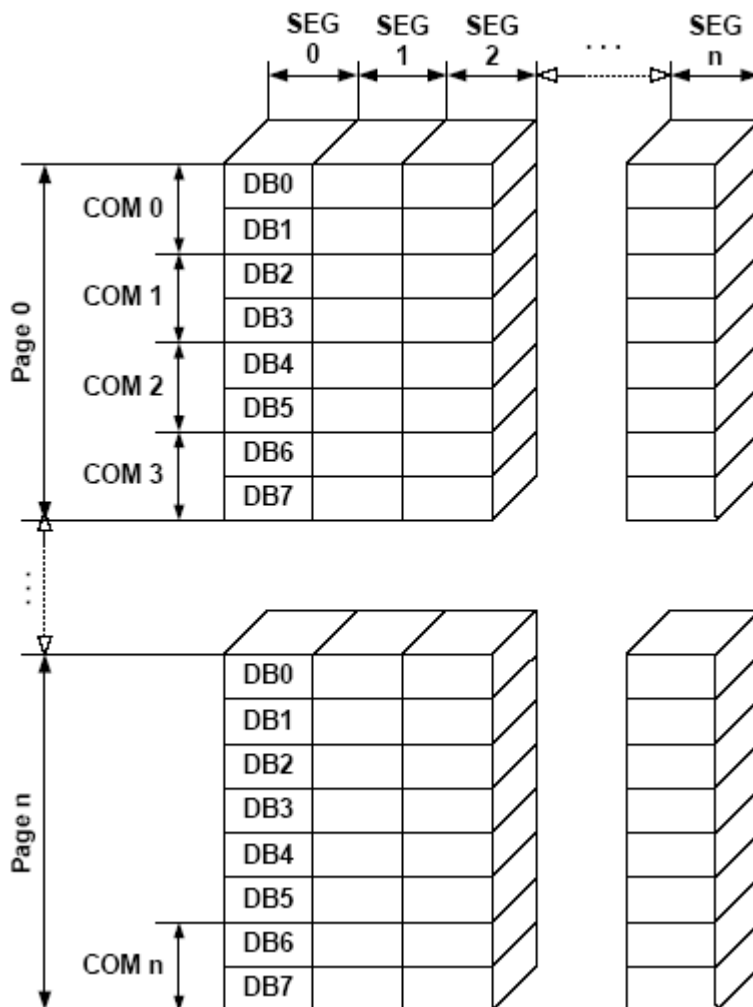
#### 每像素的位

支持的颜色深度是 2bpp。

#### 接口

同时支持 8 位并行（简单总线）和串联（SPI）接口。

## 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

## 驱动程序额外的RAM需求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果不使用高速缓存，则没有额外的 RAM 需求。

推荐使用这些驱动程序时，一起使用一个数据高速缓存，以获得更快的 LCD 访问速度。用于高速缓存的内存的数值可以由以下公式计算：

$$\text{RAM 的大小 (字节)} = (\text{LCD\_YSIZE} + 7) / 8 * \text{LCD\_XSIZE}$$

## 附加的驱动函数

无。

## 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说 明
LCD_INIT_CONTROLLER	初始化 LCD 控制器序列。
LCD_READ_A0	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
LCD_READ_A1	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
LCD_WRITE_A0	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
LCD_WRITE_A1	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说 明
LCD_CACHE	当设置为 0 时，没有使用显示数据高速缓存，会降低驱动程序的速度。默认值是 1（高速缓存激活）。

## 某些LCD控制器的特殊要求

无。

## 22.6 LCD15XX

### 支持的硬件

#### 控制器

该驱动程序已经通过下列 LCD 控制器的测试：

- Samsung KS0713
- Epson SED1560
- Epson SED1565

- Epson SED1566
- Epson SED1567
- Epson SED1568
- Epson SED1569
- Epson SED1575

应假定它也能工作在任何与些结构相类似的控制器上。

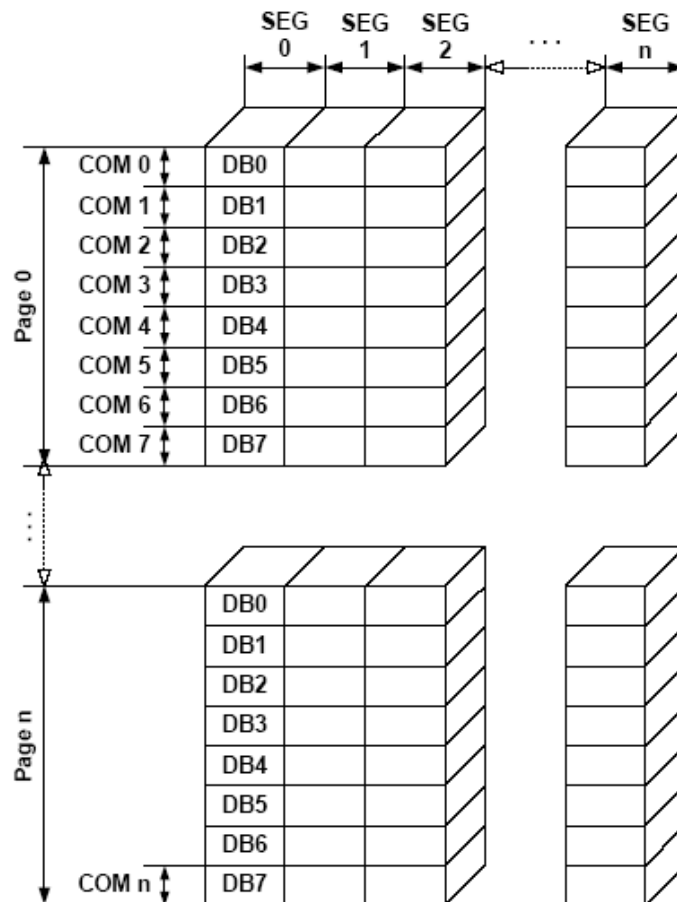
### 每像素的位

支持的颜色深度是 1bpp。

### 接口

同时支持 8 位并行（简单总线）和串联（SPI）接口。

### 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

## 驱动程序额外的 RAM 需求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果不使用一个高速缓存，则没有额外的 RAM 需求。

推荐使用这些驱动程序时，一起使用一个数据高速缓存，以获得更快的 LCD 访问速度。用于高速缓存的内存的数值可以由以下公式计算：

$$\text{RAM 的大小 (以字节为单位)} = (\text{LCD\_YSIZE} + 7) / 8 * \text{LCD\_XSIZE}$$

## 附加的驱动函数

### LCD\_L0\_ControlCache

有关这个函数的信息，请参阅第 23 章“LCD 驱动程序 API”。

## 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说明
LCD_INIT_CONTROLLER	初始化 LCD 控制器序列。
LCD_READ_A0	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
LCD_READ_A1	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
LCD_WRITE_A0	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
LCD_WRITE_A1	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说明
LCD_CACHE	当设置为 0 时，没有使用显示数据高速缓存，会降低驱动程序的速度。默认值是 1（高速缓存激活）。
LCD_SUPPORT_CACHECONTROL	当设置为 0 时，LCD_L0_ControlCache () 驱动程序 API 的高速缓存控制功能被禁止。

## 某些LCD控制器的特殊要求

无。

## 22.7 LCD6642X

### 支持的硬件

#### 控制器

该驱动程序已经通过下列 LCD 控制器的测试：

- Hitachi HD66420
- Hitachi HD66421

应假定它也能工作在任何与些结构相类似的控制器上。

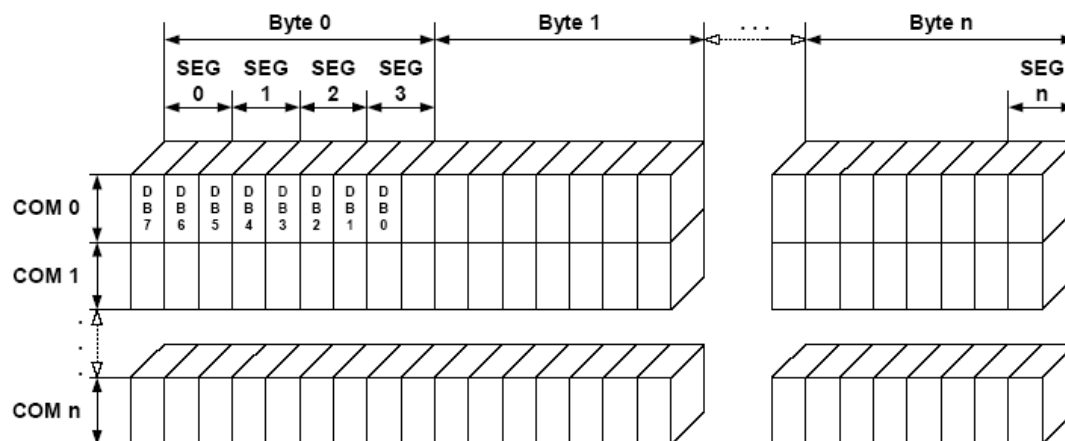
#### 每像素的位

支持的颜色深度是 2bpp。

#### 接口

该驱动程序支持 8 位并行（简单总线）接口。

### 显示屏数据 RAM 的结构





上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

## 驱动程序额外的RAM需求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果不使用高速缓存，则没有额外的 RAM 需求。

可选择的（而不是推荐），使用这些驱动程序时，同时使用一个数据高速缓存，以获得更快的LCD访问速度。用于高速缓存的内存的数值可以由以下公式计算：

$$\text{RAM的大小 (字节)} = (\text{LCD\_YSIZE}+7)/8 * \text{LCD\_YSIZE} * 2$$

## 附加的驱动函数

无。

## 硬件配置

这个驱动程序使用一个如第20章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说明
<code>LCD_INIT_CONTROLLER</code>	初始化 LCD 控制器序列。
<code>LCD_READ_A0</code>	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
<code>LCD_READ_A1</code>	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
<code>LCD_WRITE_A0</code>	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
<code>LCD_WRITE_A1</code>	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说明
<code>LCD_CACHE</code>	当设置为 0 时，没有使用显示数据高速缓存，会降低驱动程序的速度。默认值是 1（高速缓存激活）。

## 某些 LCD 控制器的特殊要求

无。

## 22.8 LCDMem

### 使用CPU作为LCD控制器

在相对快速的 CPU 和小的（四分之一 VGA 或者更小）LCD 中，没有必要使用一个 LCD 控制器。微型控制器（CPU）能兼做 LCD 控制器的工作，在一中断服务程序中刷新屏幕。该 CPU 的内存被用作图像存储器。

该方法包括下列优点：

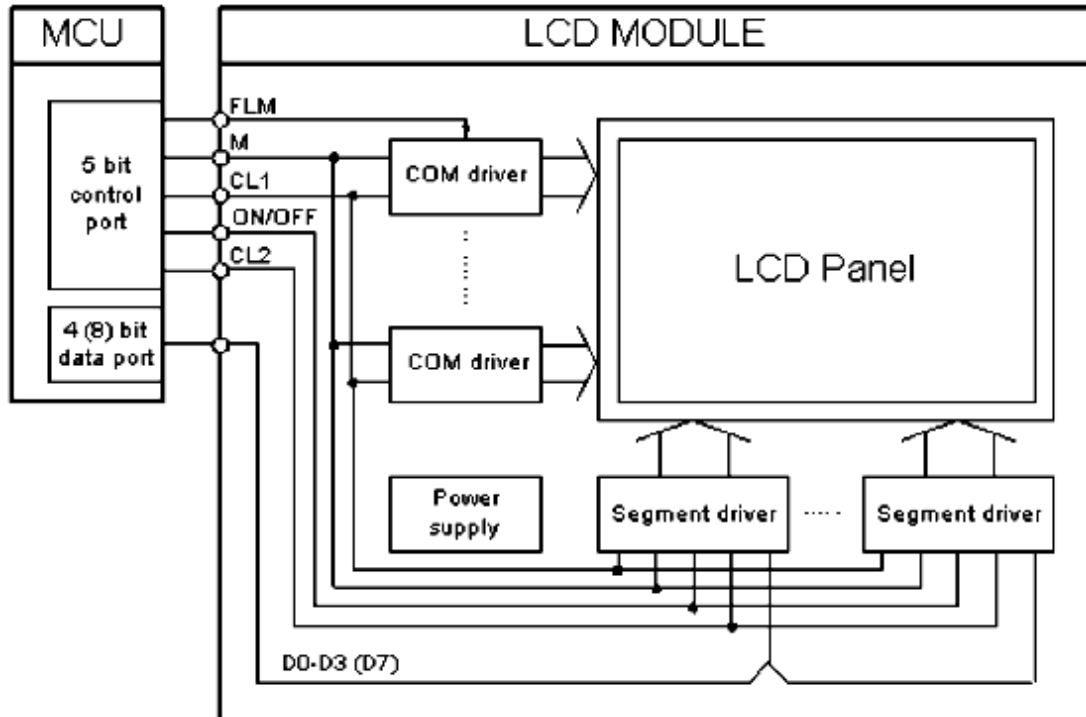
- 可以做到非常快速的屏幕更新。
- 除去了 LCD 控制器（和它的外部 RAM），减少了硬件成本。
- 简化了硬件设计。
- 可以显示 4 级灰阶。

缺点是会占用很大一部分 CPU 计算时间。根据不同的 CPU，这会占到 CPU 的开销的 20% 到几乎 100%之间；对于较慢的 CPU，它根本是极不合理的。

这类接口不需要一个特殊的 LCD 驱动程序，因为 uC/GUI 简单地将所有显示数据放入 LCD 高速缓存中。你自己必须写硬件相关部分软件，周期性地将数据从高速缓存的内存传递到你的 LCD。对于 M16C 和 M16C/80，传递图像到显示屏中的范例代码可以用“C”和最佳化的汇编程序实现。

### 如何连接 CPU 到 行/列 驱动程序

连接微型控制器到行/列驱动程序是相当容易的。需要 5 根控制线，以及 4 根或 8 根数据线（取决于列驱动程序是否能工作在 8 位模式中）。建议使用 8 位模式，它更有效率，节约 CPU 的计算时间。全部数据线应该在单个端口上，使用端口位 0..3 或者 0..7 以保证高效率的访问。该设置说明如下：



### CPU 占用率

CPU 负荷取决于硬件和使用的控制器，以及显示屏的尺寸。例如：

Mitsubishi M16C62 控制器，16MHz，160×100 显示屏，8 位接口，80 Hz 刷新频率 = 大约 12% 的 CPU 占用率。

Mitsubishi M16C62 控制器，16MHz，240×128 显示屏，8 位接口，80 Hz 刷新频率 = 大约 22% 的 CPU 占用率。

### 支持的硬件

#### 控制器

无。

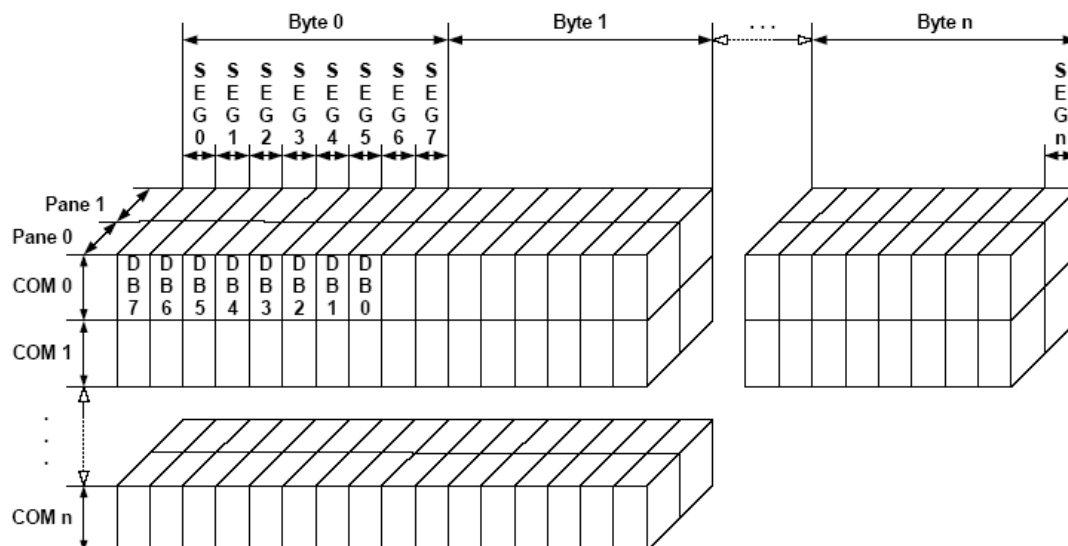
#### 每像素的位

支持的颜色深度是 2bpp。

#### 接口

该驱动程序支持 1/4/8 位的 CPU 连接 LCD 的接口。

## 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。对于每个像素，显示存储器被分成两个方框。每个像素较低的位存储在 pane0 中而较高的位存储在 pane 1 中。该方法的优点是显示数据的输出能够立即得到执行。

## 驱动程序的RAM需求

该驱动程序仅仅使用一个包含显示数据的存储区。所需显示存储器的大小可以计算如下：

$$\text{RAM 的大小 (字节)} = (\text{LCD\_YSIZE} + 7) / 8 * \text{LCD\_YSIZE} * 2$$

## 附加的驱动函数

无。

## 硬件配置

通常，该硬件接口是一个更新 LCD 的中断服务程序 (ISR)。一个用“C”代码编写的输出程序随 uC/GUI 一道发布。该程序仅仅作为一个例子提供。为了最优化执行速度，它必须改编为汇编程序代码。

对于如何写输出程序的详细信息，请看一下随驱动程序一道提供的例子或与我们联系。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说 明
LCD_TIMERINIT0	用于显示 pane 0 的 ISR 的时间值。
LCD_TIMERINIT1	用于显示 pane 1 的 ISR 的时间值。
LCD_ON	LCD 切换到“开”的功能置换宏。
LCD_OFF	LCD 切换到“关”的功能置换宏。

## 22.9 LCDMemC

该驱动程序，如 LCDMem，用于一个没有 LCD 控制器的系统的设计。差异是 LCDMemC 支持彩色显示器。有关使用 CPU 代替 LCD 控制器的更多信息，请参见该上述的 LCDMem 驱动程序部分。

### 支持的硬件

#### 控制器

无。

#### 每像素的位

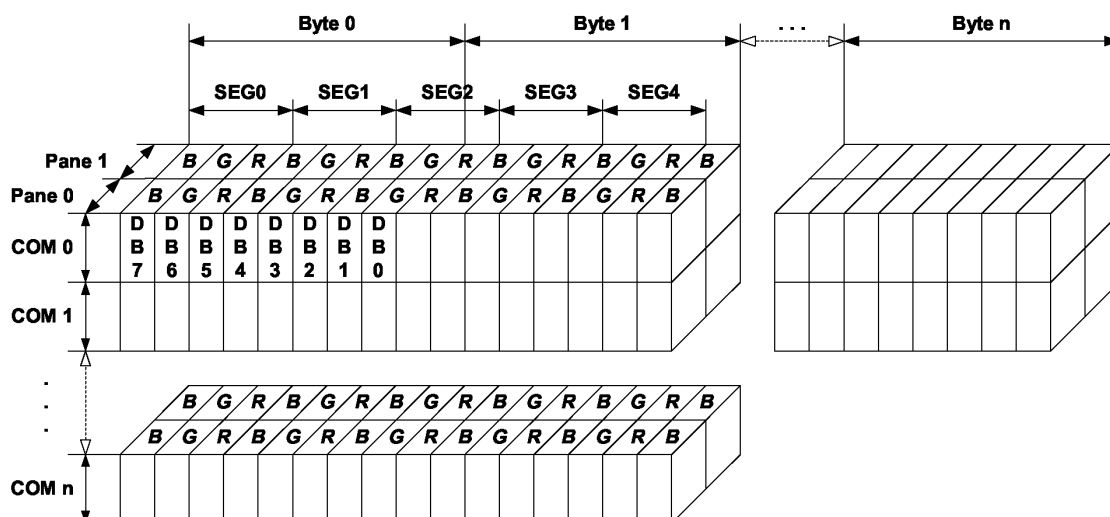
支持的颜色深度是 3 和 6bpp。

#### 接口

该驱动程序支持 1/4/8 位的 CPU 连接 LCD 的接口。

### 显示屏数据 RAM 的结构

每像素 6 位，固定调色板 = 222



**每像素 3 位，固定调色板 = 111**

该驱动程序支持一个 3 或 6 bpp 存储区用于彩色显示器。上图展示了依据颜色深度，驱动程序和 LCD 的 SEG 和 COM 引线之间的相关性。

**每像素 6 位，固定调色板模式 222**

当使用 6 bpp 模式时，显示存储器被分成每像素 2 个方框。每个像素较低的位存储在 pane0 中而较高的位存储在 pane 1 中。该方法的优点是显示数据的输出能够立即得到执行。

**每像素 3 位，固定调色板模式 111**

当使用这个模式时，每个像素仅仅存在一个方框。

## 驱动程序的RAM需求

该驱动程序仅仅使用一个包含显示数据的存储区。所需显示存储器的大小可以计算如下：

**每像素 6 位，固定调色板模式 222**

RAM 的大小 (字节) = ( LCD\_XSIZE+7)/8\*3\*2

**每像素 3 位，固定调色板模式 111**

RAM 的大小 (字节) = ( LCD\_XSIZE+7)/8\*3

## 附加的驱动函数

无。

## 硬件配置

通常，该硬件接口是一个更新LCD的中断服务程序（ISR）。一个用“C”代码编写的输出程序随 $\mu$ C/GUI一道发布。该程序仅仅作为一个例子提供。为了最优化执行速度，它必须改编为汇编程序代码。

对于如何写输出程序的详细信息，请看一下随驱动程序一道提供的例子或与我们联系。

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说明
LCD_TIMERINIT0	用于显示 pane 0 的 ISR 的时间值。
LCD_TIMERINIT1	用于显示 pane 1 的 ISR 的时间值(仅仅用于 6 bpp 模式)。
LCD_ON	LCD 切换到“开”的功能置换宏。
LCD_OFF	LCD 切换到“关”的功能置换宏。

## 22.10 LCDPage1bpp

### 支持的硬件

#### 控制器

该驱动程序已经通过下列 LCD 控制器的测试：

- Philips PCF8810
- Philips PCF8811

应假定它也能工作在任何与些结构相类似的控制器上。

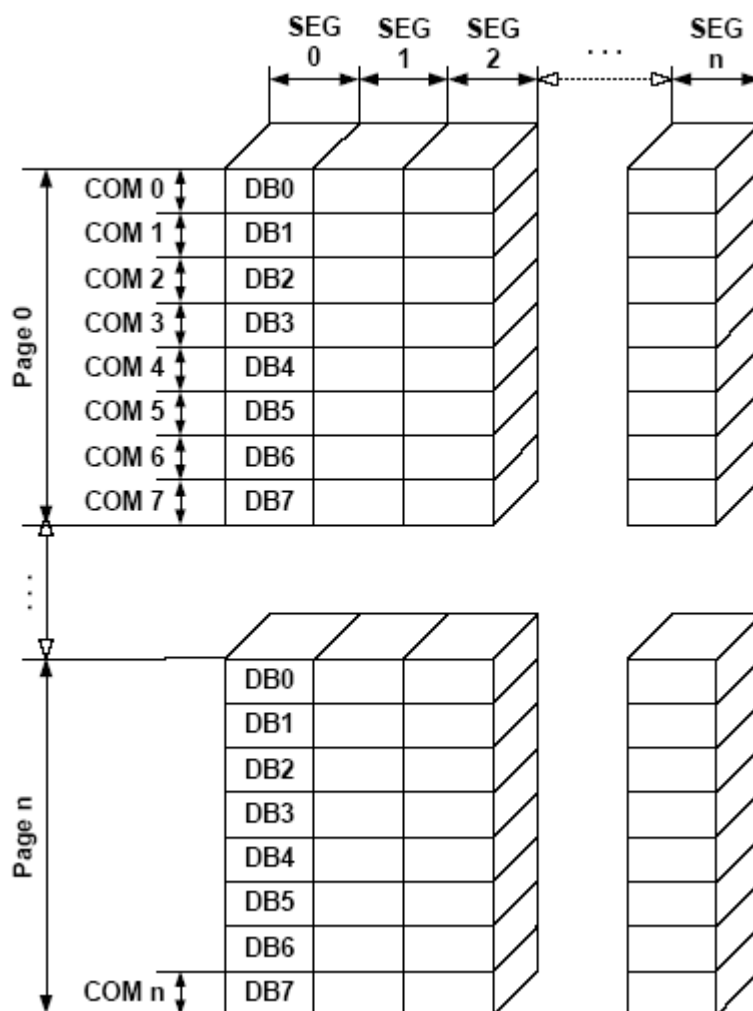
#### 每像素的位

支持的颜色深度是 1bpp。

## 接口

同时支持 8 位并行（简单总线）和串联（SPI）接口。

## 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

## 驱动程序额外的RAM需求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果不使用高速缓存，则没有额外的 RAM 需求。

推荐使用这些驱动程序时，一起使用一个数据高速缓存，以获得更快的 LCD 访问速度。用于高速缓存的内存的数值可以由以下公式计算：



RAM 的大小 (字节) = ( LCD\_YSIZE+7)/8\* LCD\_XSIZE

## 附加的驱动函数

### LCD\_L0\_ControlCache

有关这个函数的信息，请参阅第 23 章“LCD 驱动程序 API”。

## 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说 明
LCD_INIT_CONTROLLER	初始化 LCD 控制器序列。
LCD_READ_A0	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
LCD_READ_A1	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
LCD_WRITE_A0	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字
LCD_WRITE_A1	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字

## 附加的配置开关

下表展示了对于这个驱动程序有效的可选择配置开关：

宏	说 明
LCD_CACHE	当设置为 0 时，没有使用显示数据高速缓存，会降低驱动程序的速度。默认值是 1 (高速缓存激活)。
LCD_SUPPORT_CACHECONTROL	当设置为 0 时，LCD_L0_ControlCache () 驱动程序 API 的高速缓存控制功能被禁止。

## 某些LCD控制器的特殊要求

无。

## 22.11 LCDSLIn

### 支持的硬件

#### 控制器

该驱动程序已经通过下列LCD控制器的测试：

- Epson SED1330
- Epson SED1335
- Toshiba T6963

应假定它也能工作在任何与些结构相类似的控制器上。

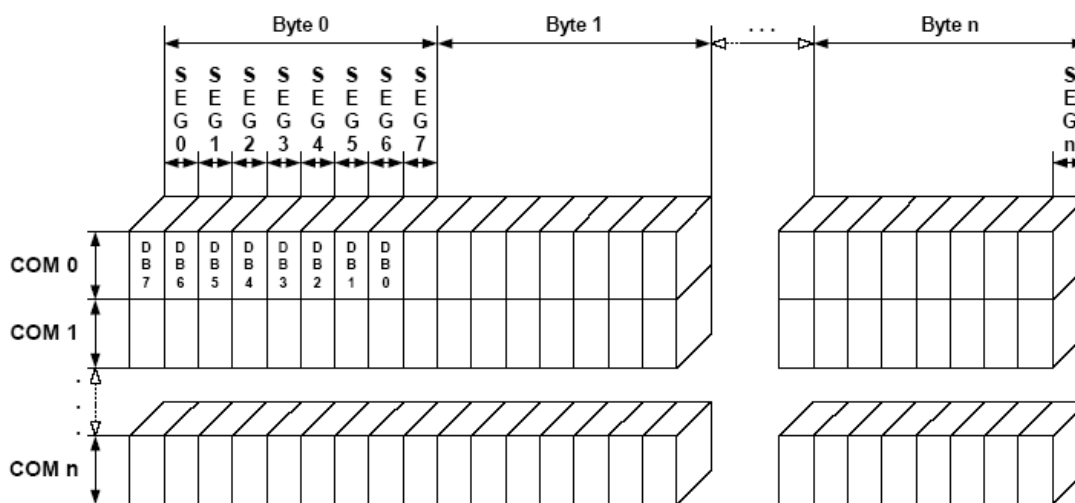
### 每像素的位

支持的颜色深度是 1bpp。

### 接口

该驱动程序支持 8 位并行（简单总线）接口。

### 显示屏数据 RAM 的结构



上图展示了显示存储器和 LCD 的 SEG 和 COM 引线之间的关系。

### 额外的RAM需求

这些 LCD 驱动程序可以使用或不使用一个显示数据高速缓存，包含一个 LCD 数据 RAM 的容量的完全拷贝。如果没有使用调整缓存，则没有额外的 RAM 需要。

推荐使用这些驱动程序时，一起使用一个数据高速缓存，以获得更快的 LCD 访问速度。用

于高速缓存的内存的数值可以由以下公式计算：

$$\text{RAM 的大小 (字节)} = (\text{LCD\_YSIZE}+7)/8 * \text{LCD\_YSIZE}$$

### 附加的驱动函数

无。

### 硬件配置

这个驱动程序使用一个如第 20 章“低层配置”所描述的简单总路线接口访问硬件。下表列出了必须为硬件访问所定义的宏：

宏	说 明
LCD_INIT_CONTROLLER	初始化 LCD 控制器序列。
LCD_READ_A0	A 线 (A-line) 为低电平时从 LCD 控制器读一个字节。
LCD_READ_A1	A 线 (A-line) 为高电平时从 LCD 控制器读一个字节。
LCD_WRITE_A0	A 线 (A-line) 为低电平时向 LCD 控制器写入一个字节。
LCD_WRITE_A1	A 线 (A-line) 为高电平时向 LCD 控制器写入一个字节。

### 附加的配置开关

无。

### 某些 LCD 控制器的特殊要求

无。