

## MBI5153 Programming Guide

(此文件仅供控制器开发使用，请勿外流)

### 控制指令

表 1. 控制指令如下：

指令名称	讯号组合		叙述
	LE	Number of DCLK LE 包含多少个 DCLK 上升缘	指令动作
停止错误侦测	High	1	停止 LED 开路强制侦测
数据栓锁	High	1	将序列数据传入缓冲存储器
VSYNC	High	2	垂直同步信号。垂直同步则会命令芯片置换新的帧数据
写入状态缓存器 1*	High	4	将序列数据传入状态缓存器 1
读取状态缓存器 1	High	5	将状态缓存器 1 的数据传入位移缓存器
执行错误侦测	High	7	执行 LED 开路强制侦测
写入状态缓存器 2*	High	8	将序列数据传入状态缓存器 2
读取状态缓存器 2	High	9	将状态缓存器 2 的数据传入位移缓存器
软件重置	High	10	热启动，软复位
前置设定(Pre-Active)	High	14	前置设定指令必须在“写入状态缓存器”指令之前传送
写入状态缓存器 3*	High	16	将序列数据传入状态缓存器 3
读取状态缓存器 3	High	17	将状态缓存器 3 的数据传入位移缓存器

上述“\*”符号表示这些指令必须在前置设定指令之后才会启动，否则皆无效。

**状态缓存器 1 及 2**

表 2. 状态缓存器 1 功能说明

默认值

高位

低位

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1	0	0	6'b101011					

位	属性	定义	值	功能说明
F	读/写	下鬼影消除	0 (默认)	0: 关闭
			1	1: 开启
E~D	保留	保留	00	保留
C~8	读/写	扫描行数	00000 00001 00010 00011 (默认) ~ 11111	00000: 1 行扫描 00001: 2 行扫描 00010: 3 行扫描 00011: 4 行扫描 ..... 11110: 31 行扫描 11111: 32 行扫描
7	读/写	SPWM 灰阶模式	0 (默认)	14 位灰阶模式。16384 GCLKs 的 14 位 PWM 工作周期可被分割为 32 个较小的 PWM 工作周期，每一周期有 512 GCLKs 用户应传递 16 位数据，该数据在最不重要位(LSB)须有 2 位 0, Ex., {14'h1234, 2'h0}
			1	13 位灰阶模式。8192 GCLKs 的 13 位 PWM 工作周期可被分割为 16 个较小的 PWM 工作周期，每一周期有 512 GCLKs 用户应传递 16 位数据，该数据在最不重要位(LSB)须有 3 位 0, Ex., {13'h1234, 3'h0}
6	读/写	GCLK 倍频	0 (默认)	0: 关闭
			1	1: 开启
5~0	读/写	电流增益调整	000000~ 111111	6'b101011 (默认) 64阶微调的电流增益功能 (增益范围: 12.5%~200%)，可适当调整输出电流。

表 3. 状态缓存器 2 功能说明

默认值

高位													低位			
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	

位	属性	定义	值	功能说明
F~B	保留	保留	保留	保留
A	读/写	双倍刷新率	0 (默认)	0: 关闭
			1	1: 开启
9~4	保留	保留	保留	保留
3~1	读/写	解决第一行扫偏暗	000(默认) ~111	000: 0 ns, 100: 20ns 001: 5 ns, 101: 25ns 010: 10 ns, 110: 30ns 011: 15 ns, 111: 35ns
0	保留	保留	保留	保留

表 4. 状态缓存器 3 功能说明

默认值

高位													低位			
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

位	属性	定义	值	功能说明
F~0	保留	保留	保留	保留

## 开发控制接口

1. 提供三组状态缓存器的字段，如下图所示，让使用者作填写。



第一组寄存器		第二组寄存器		第三组寄存器	
高字节 (Hex)	低字节 (Hex)	高字节 (Hex)	低字节 (Hex)	高字节 (Hex)	低字节 (Hex)
红芯片: 00	01	红芯片: 00	01	红芯片: 94	00
绿芯片: 70	00	绿芯片: 70	00	绿芯片: 00	00
蓝芯片: 70	10	蓝芯片: 70	10	蓝芯片: 00	00

2. 控制状态缓存器 2 位[B]要设成可控，如下图设成可(不)勾选。



去坏点设置

启用去除坏点      去除坏点

## 状态缓存器建议值

状态缓存器建议设定如下：

表 5. 状态缓存器 1 建议值

使用红/绿/蓝 LED 作负载的 IC 建议值 16'hDF2B：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1	1	0	1	1	1	1	1	0	0	1	0	1	0	1	1

表 6. 状态缓存器 2 建议值

使用红色 LED 作负载的 IC 建议值 16'h4600：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0

使用绿/蓝 LED 作负载的 IC 建议值 16'h6600：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0

表 7. 状态缓存器 3 建议值

使用红色 LED 作负载的 IC 建议值 16'hC003：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

使用绿色 LED 作负载的 IC 建议值 16'h6103：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1

使用蓝色 LED 作负载的 IC 建议值 16'h3303：

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1

## 初始化顺序

在初始化阶段，倘若缓存器之默认值不是用户所设定，就要执行写入状态缓存器指令，接着用户传送“16 x 扫描行数”数量的“数据控锁”指令以传送灰阶数据，然后传送“Vsync”指令以启动显示屏。

在用户传送完毕最后一次“数据控锁”指令之后，必须再等至少 50 GCLKs，始能再下达“Vsync”指令。然而，GCLK 务必在“Vsync”指令设定之前停止，有关时间设定限制的说明将在后续单元中详述。

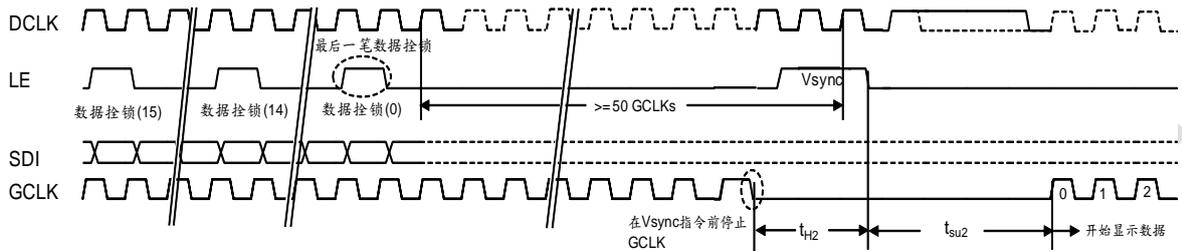


图 1. “数据控锁”与“Vsync”指令说明

## 垂直同步指令操作策略

垂直同步指令作用在于更新帧数据。以下波形图即说明更新帧数据的“Vsync”指令。

### 更新帧数据的 Vsync 指令

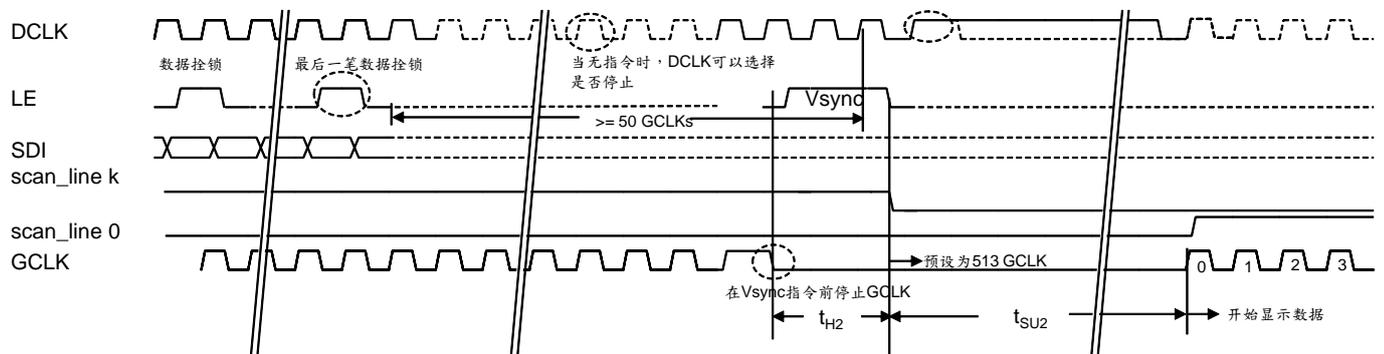


图 2. 更新帧数据的“Vsync”指令说明

### 限制条件说明:

1. 在最后一笔“数据控锁”指令之后，灰阶数据需要一段时间以预先读取来自 SRAM 的数据至内建的缓冲器中，因此在“Vsync”指令之前至少需要等 50 GCLKs。

说明: 如欲了解更多 SRAM 内存结构说明，请参考内存结构单元。

2. 建议控制器使用时保留一个 GCLK 计数器(1~513)，该 GCLK 计数器将在“Vsync”指令的 LE 下降缘时预设为 512，并且在下一笔 GCLK 时重新设定为 0。
3. LE 与 GCLK 之间有时间设定限制，也就是 GCLK 应在“Vsync”指令之前停止。另外 GCLK 与 LE 必须符合设定与保留时间，分别如 3.1 与 3.2 所示：
  - 3.1 设定时间指 LE 下降缘与 GCLK 上升缘之间  $t_{SU2}$  需大于 1200ns。
  - 3.2 保留时间指 GCLK 下降缘与 LE 下降缘之间  $t_{H2}$  需大于 300ns。

4. 消影时间(dead time)亦指扫描行数间的间隔时间，并藉由停止 GCLK 来控制。因为“Vsync”指令设定后，帧数据会更新，使用者必须将扫描行数由目前的第 K 行扫描切换回第 0 行扫描，以重新开启新的帧数据显示。
5. 当无指令时，DCLK 可以选择是否停止。
6. 在消影时间时，可选择停止 DCLK，或不要下达“数据锁”指令。
7. 在“Vsync”指令时，新数据将会储存于内建的显示缓冲器中，且在消影时间完成后显示该数据。

### 在帧数据显示期间切换扫描行数

在扫描行数的开关中，使用者应采用 MBI5153 GCLK 之计数模式，且在 MBI5153 GCLK 计数至 512 时切换扫描行数。若要了解更多扫描屏应用方式，可以参考“灰阶模式与扫描式 S-PWM”单元之叙述。消影时间是由控制器以暂停 GCLK 所控制，其间高电位  $tdth$  需大于 300ns，低电位  $tdtl$  需大于 1200ns。当 GCLK 计数值为 513 时，MBI5153 在消影时间将会关闭所有输出通道。

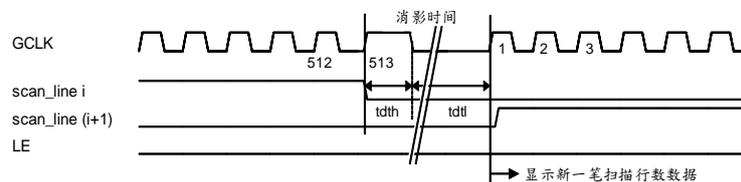


图 3. 扫描行数切换，消影时间之控制方式

控制方式如下所述：

1. 透过“写入状态缓存器”指令设定状态缓存器。
2. 传送“16 x 扫描行数”数量的“数据锁”指令以传送灰阶数据。
3. 帧频(frame rate)是指每秒钟放映或显示的帧或图像的数量。在用户传送完毕最后一次“数据锁”指令之后，必须再等至少 50 GCLKs，始能再下达“Vsync”指令。若非第一笔数据，用户应根据帧频传送“Vsync”指令。如果帧频为 60，使用者应等待 1/60 秒的时间。当用户传送“Vsync”指令时，必需遵守相关的时间限制条件。
4. 当传送“Vsync”指令时，扫描行数必须从 1 开始计算，GCLK 之计数需要预设为 513 并藉由停止 GCLK 以控制消影时间。
5. 在放映或显示的帧或图像时，用户必须维持一个 GCLK 计数(1~513)，并且在 GCLK 计数至 512 时，切换扫描行数以及进入消影时间(由停止 GCLK 控制)。
6. 在消影时间期间(包括传送“Vsync”指令或当 GCLK 之计数为 513)，不可传送“数据锁”指令。
7. 下一笔帧影像的灰阶数据必须在“Vsync”指令之后传送。
8. 强烈建议定期执行“写入控制状态缓存器 1”与“写入控制状态缓存器 2”指令以避免系统的噪声干扰。

## 视觉更新率

### 一般模式

在 14 位 S-PWM 模式下，16384 GCLKs 的工作周期可被分割为 32 个较小的 PWM 工作周期，每一周期有 512 GCLKs；在 13 位 S-PWM 模式下，8192 GCLKs 的工作周期可被分割为 16 个较小的 PWM 工作周期，每一周期有 512 GCLKs。

当使用 MBI5153 做扫描屏时，视觉更新率计算方式如下

$$F_{visual} = \frac{1}{[512 \times (T_{GCLK}) + t_{Dead}] \times N}$$

其中

$F_{visual}$ : 视觉更新率

$T_{GCLK}$ : 灰阶时钟周期(即  $1/F_{GCLK}$ )

$t_{Dead}$ : 消影时间

$N$ : 扫描行数

### 双倍刷新率

在 14 位 S-PWM 模式下，16384 GCLKs 的工作周期可被分割为 64 个较小的 PWM 工作周期，每一周期有 256 GCLKs；在 13 位 S-PWM 模式下，8192 GCLKs 的工作周期可被分割为 32 个较小的 PWM 工作周期，每一周期有 256 GCLKs。

当使用 MBI5153 做扫描屏时，视觉更新率计算方式如下

$$F_{visual} = \frac{1}{[256 \times (T_{GCLK}) + t_{Dead}] \times N}$$

### GCLK 倍频

若开启倍频功能，只需要原先一半的 GCLKs 数即可表现 PWM 工作周期。例如，在 14 位 S-PWM 模式下，完成一个行扫的工作周期仅需 8192 GCLKs，可被分割为 32 个较小的 PWM 工作周期，每一周期是 256 GCLKs；在 13 位 S-PWM 模式下，完成一个行扫的工作周期仅需 4096 GCLKs，可被分割为 16 个较小的 PWM 工作周期，每一周期是 256 GCLKs。视觉更新率计算方式如下

$$F_{visual} = \frac{1}{[256 \times (T_{GCLK}) + t_{Dead}] \times N}$$

### 强制开路错误侦测

MBI5153 之强制开路错误侦测的原理是基于实际输出端的耐受电压( $V_{DS}$ )与侦测电压 0.5V 的比较, 来判定每个输出端的 LED 负载状态。如图 4 所示, 下达“强制开路错误侦测”指令后, 错误侦测时间建议大于 100us, 侦测期间, GCLK 必需停止, MBI5153 输出端关闭, 每个通道流过极小电流, 进行侦测。当接收端收到“停止错误侦测”指令后, 驱动器将会将错误组态数据搬移到“位移寄存器”, 透过 SDO 脚位输出每个位。

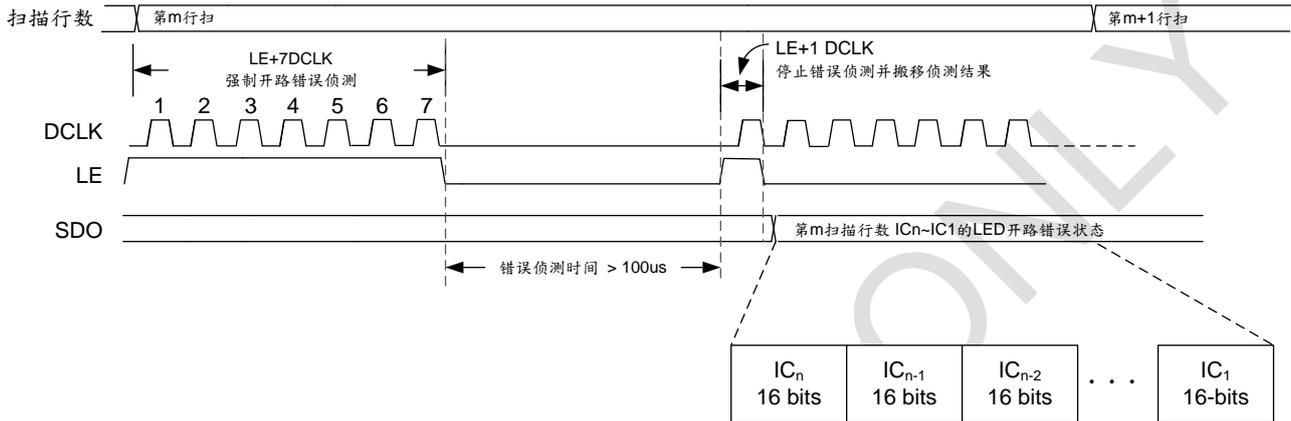


图 4. 强制开路错误侦测时序图

### 消除十字架问题的控制时序

消除十字架问题的控制程序如下：

1. 写入软件重置指令。
2. 写入控制状态缓存器 1 位[12:8]设扫描行数。(若已设定，此步骤可忽略)
3. 写入控制状态缓存器 2，将位[F]设为 0，将位[B]设为 1，其他位值维持不变。
4. 假设有 N 行扫，首先，开启第 1 行扫，写入错误侦测指令，侦测时间 200us 后停止错误侦测。
5. 接着开启第 2 行扫，重复程序 4 的步骤，依序逐行换扫，直到完成第 N 行扫。
6. 控制状态缓存器 2 位[B]维持 1，其他状态缓存器设定值则恢复显示屏设定值，接续作灰阶输入，进入一般显示状态。

时序如图 5 所示：

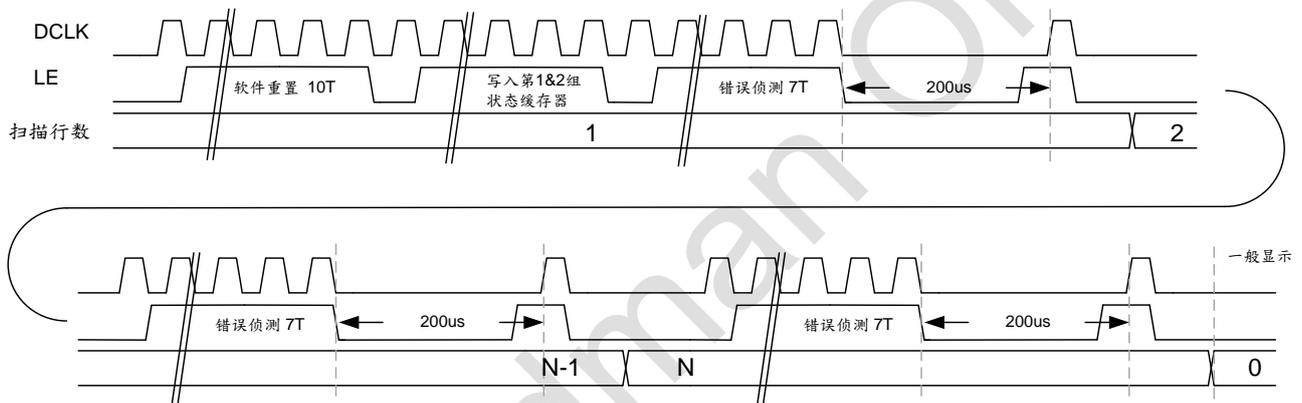


图 5. 消除十字架问题之控制时序图