

longxiaowu的专栏

个人资料



小伍_

访问：190631次
积分：1609
等级：BLOG > 4
排名：千里之外

原创：14篇 转载：16篇
译文：0篇 评论：14条

文章搜索

文章分类

- linux菜鸟旅程 (4)
- LCD (15)
- CTP (0)
- sensor (0)
- git (4)
- 手机杂谈 (1)
- MSM (4)
- IT (1)
- debug (2)

文章存档

- 2015年05月 (1)
- 2015年03月 (1)
- 2014年12月 (2)
- 2014年05月 (10)
- 2014年04月 (14)

展开

阅读排行

- git stash 命令 (34114)
- MIPI DSI协议介绍 (26610)
- git commit命令的使用与 (14870)

【活动】2017 CSDN博客专栏评选 【评论送书】SQL优化、深度学习、数据科学家 CSDN日报20170526 ——《论程序员的时代焦虑与焦虑的缓解》 CSDN 日报 | 4.19-5.19 上榜作者排行出炉

MIPI DSI协议介绍

2014-04-24 14:04 26670人阅读 评论(2) 收藏 举报

分类：LCD (14)

原文地址：<http://blog.csdn.NET/qq160816/article/details/19555957>

一、MIPI

MIPI（移动行业处理器接口）是**Mobile Industry Processor Interface**的缩写。MIPI（移动行业处理器接口）是MIPI联盟发起的为移动应用处理器制定的开放标准。

已经完成和正在计划中的规范如下：

工作组	规范名称
Camera工作组	<ul style="list-style-type: none">MIPI Camera Serial Interface 1.0 specificationCamera Serial Interface 2 v1.0 (CSI-2)
Device Descriptor Block工作组	暂无
DigRF工作组	<ul style="list-style-type: none">DigRF BASEBAND/RF DIGITAL INTERFACE SPECIFICATION Version 1.12
Display工作组	<ul style="list-style-type: none">DBI-2DPI-2DSIDCS
高速同步接口工作组	<ul style="list-style-type: none">HSI 1.0
接口管理框架工作组	暂无
低速多点连接工作组	<ul style="list-style-type: none">SLIMbus
NAND软件工作组	暂无
物理层工作组	<ul style="list-style-type: none">D-PHYM-PHY
软件工作组	暂无
系统电源管理工作组	<ul style="list-style-type: none">SPMI
检测与调试工作组	暂无
统一协议工作组	<ul style="list-style-type: none">UniPro 1 point-to-pointPIE

二、MIPI联盟的MIPI DSI规范

1、名词解释

- DCS (DisplayCommandSet): DCS是一个标准化的命令集，用于命令模式的显示模组。
- DSI, CSI (DisplaySerialInterface, CameraSerialInterface)
 - DSI 定义了一个位于处理器和显示模组之间的高速串行接口。
 - CSI 定义了一个位于处理器和摄像模组之间的高速串行接口。
- D-PHY：提供DSI和CSI的物理层定义

2、DSI分层结构

DSI分四层，对应D-PHY、DSI、DCS规范、分层结构图如下：

- PHY 定义了传输媒介，输入/输出电路和和时钟和信号机制。
- Lane Management层：发送和收集数据流到每条lane。
- Low Level Protocol层：定义了如何组帧和解析以及错误检测等。
- Application层：描述高层编码和解析数据流。

LCD参数解释及计算

wm命令使用方法及LCD (13219)

qualcomm平台抓log的方法 (9965)

adb shell input的用法 (7306)

repo使用过程中遇到的一 (7267)

git commit -m注释中换 (7115)

修改source insight 工程! (7087)

评论排行

git stash 命令 (5)

MIPI DSI协议介绍 (2)

git commit命令的使用与 (2)

wm命令使用方法及LCD (1)

修改source insight 工程! (1)

adb shell input的用法 (1)

adb出现adb server is out of date (1)

arm编程中 packed的传 (1)

有关手机LCD分辨率的一 (0)

如何单独编译Android源码 (0)

推荐文章

* 5月书讯：流畅的Python，终于等到你！

*【新收录】CSDN日报——
Kotlin 专场

* Android中带你开发一款自动爆破签名校验工具kstools

* Android图片加载框架最全解析 ——深入探究Glide的缓存机制

* Android 热修复 Tinker Gradle Plugin解析

* Unity Shader-死亡溶解效果

最新评论

wm命令使用方法和LCD显示图标

adb shell input的用法
Glorevo: 请问哪个版本的系统会有这么多参数？我用 4.0 的只有这几个呢：usage: input

```
git stash 命令
从杰: $ git stash save doing
crazy things Saved working d...
```

修改source insight 工程默认存档
gust_c: good!

arm编程中__packed的使用
大新闻: 感谢分享, 学习了!

git commit命令的使用与git默认分支

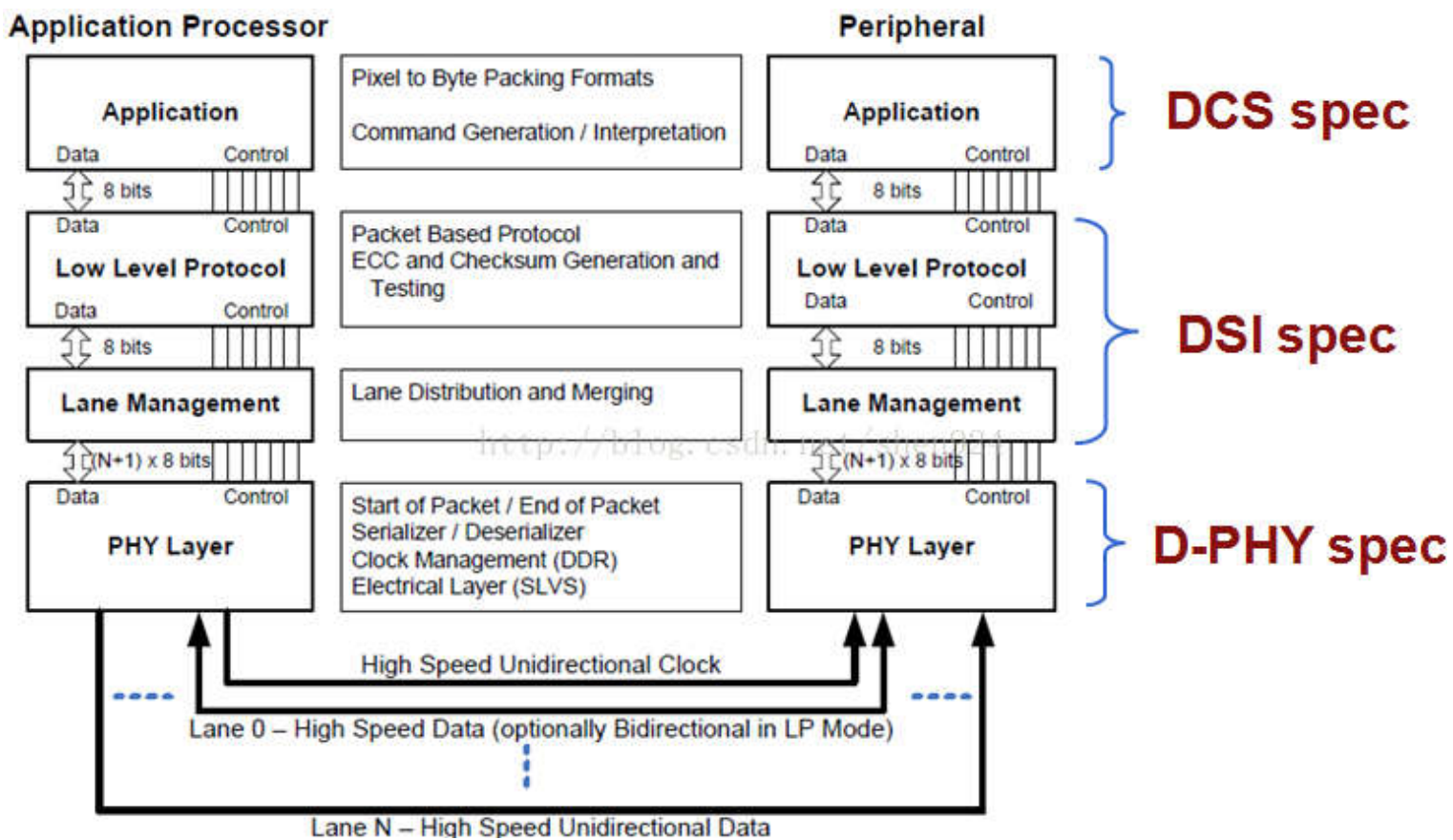
git stash 命令
没有梦想-何必远方: @iMolable:
摸楼上两位头。。。哈哈

git stash 命令
iMolable: @zhixinghe1:我也是，坑死我了；应是 git stash save -u

git stash 命令

zhixinghe1: 你这个暂存时加上-a
真是坑啊，把没改动的也加到暂
存里了，等我切换分支回来已经
没法恢复改动了

git commit命令的使用与git默认编辑器



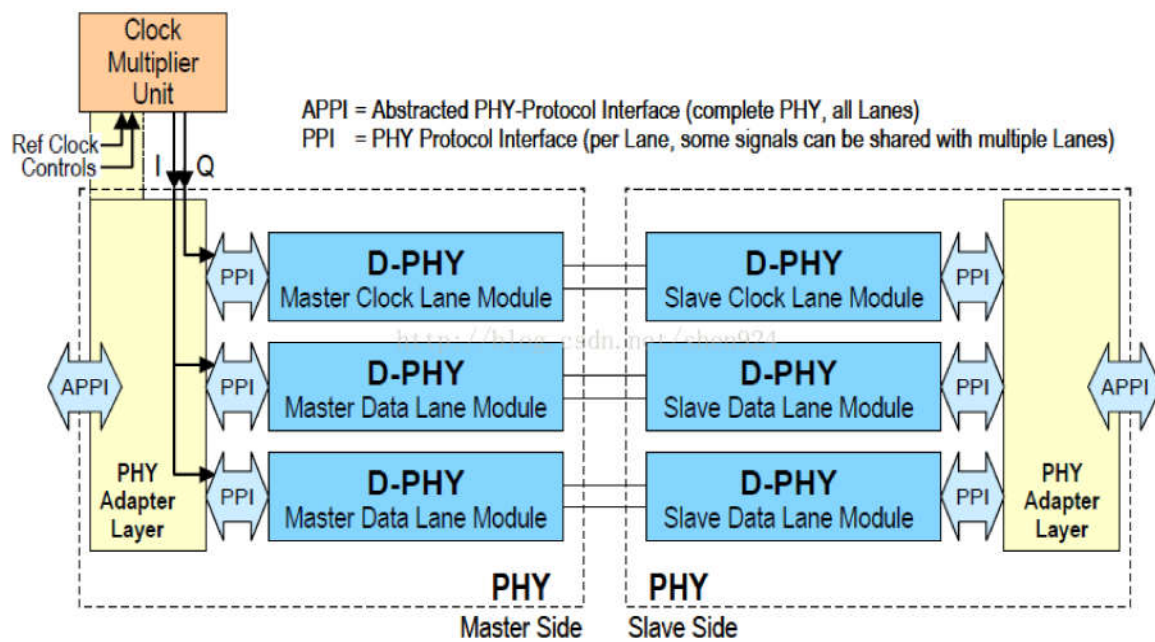
3、Command和Video模式

- DSI兼容的外设支持**Command**或**Video**操作模式，用哪个模式由外设的构架决定
- **Command**模式是指采用发送命令和数据到具有显示缓存的控制器。主机通过命令间接的控制外设。**Command**模式采用双向接口
- **Video**模式是指从主机传输到外设采用时实像素流。这种模式只能以高速传输。为减少复杂性和节约成本，只采用**Video**模式的系统可能只有一个单向数据路径

三、D-PHY介绍

1、 D-PHY 描述了一同步、高速、低功耗、低代价的PHY。

- 一个 PHY配置包括
 - 一个时钟lane
 - 一个或多个数据lane
- 两个Lanes的 PHY配置如下图



- 三个主要的lane的类型
 - 单向时钟Lane
 - 单向数据Lane
 - 双向数据Lane
- D-PHY的传输模式
 - 低功耗（Low-Power）信号模式（用于控制）：10MHz (max)
 - 高速（High-Speed）信号模式（用于高速数据传输）：80Mbps ~ 1Gbps/Lane
- D-PHY低层协议规定最小数据单位是一个字节
 - 发送数据时必须低位在前，高位在后。
- D-PHY适用于移动应用
 - DSI: 显示串行接口
 - 一个时钟lane，一个或多个数据lane
 - CSI: 摄像串行接口

2、Lane模块

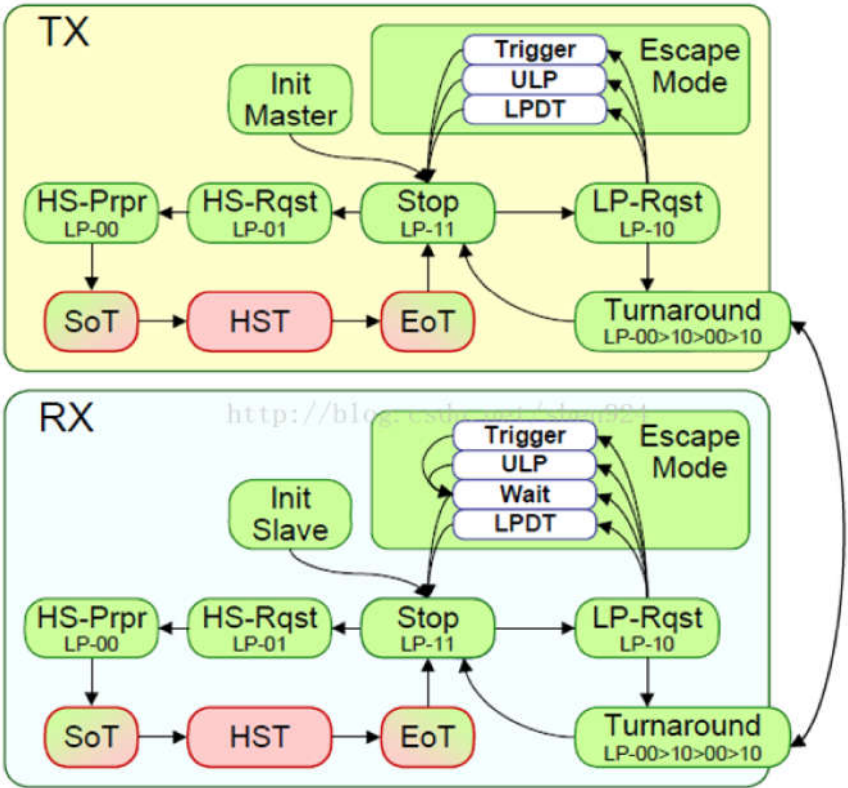
- PHY由D-PHY(Lane模块)组成
- D-PHY可能包含：
 - 低功耗发送器（LP-TX）
 - 低功耗接收器（LP-RX）
 - 高速发送器（HS-TX）
 - 高速接收器（HS-RX）
 - 低功耗竞争检测器（LP-CD）
- 三个主要lane类型
 - 单向时钟Lane
 - Master: HS-TX, LP-TX
 - Slave: HS-RX, LP-RX
 - 单向数据Lane
 - Master: HS-TX, LP-TX
 - Slave: HS-RX, LP-RX
 - 双向数据Lane
 - Master, Slave: HS-TX, LP-TX, HS-RX, LP-RX, LP-CD

3、Lane状态和电压

- Lane状态
 - LP-00, LP-01, LP-10, LP-11 (单端)
 - HS-0, HS-1 (差分)
- Lane电压（典型）
 - LP: 0-1.2V
 - HS: 100-300mV (200mV)

4、操作模式

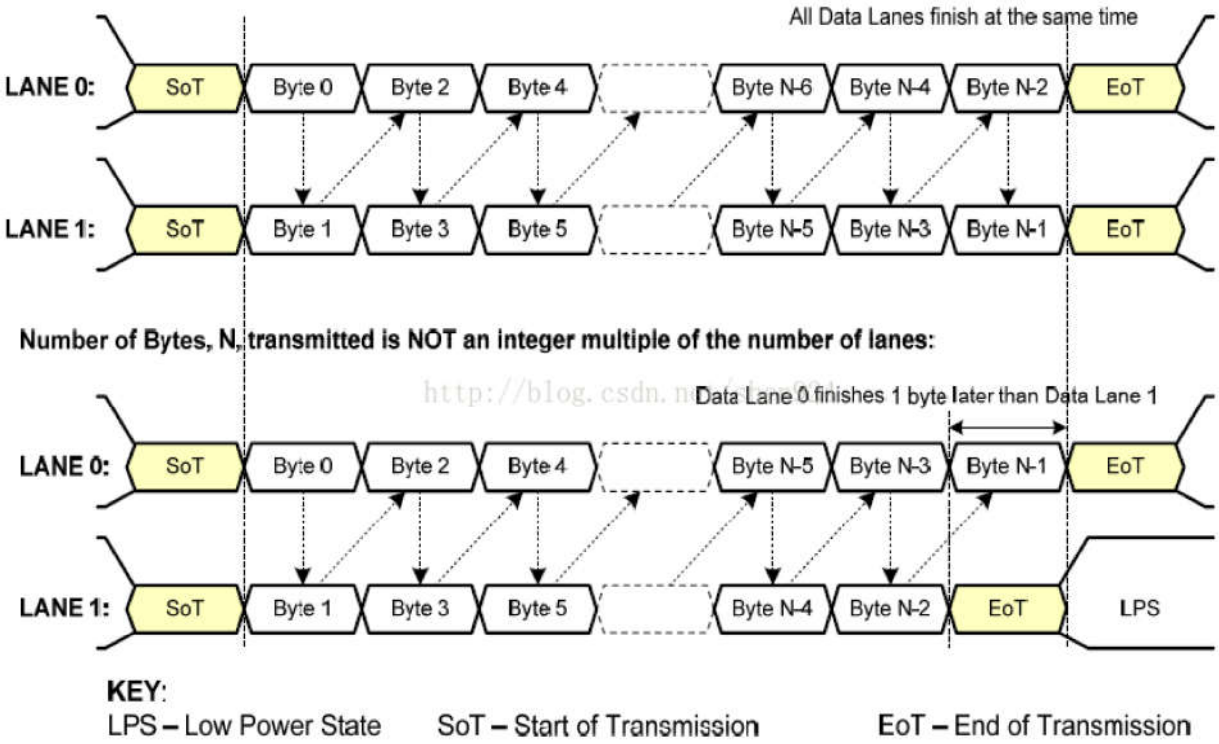
- 数据Lane的三种操作模式
 - Escape mode, High-Speed(Burst) mode, Control mode
- 从控制模式的停止状态开始的可能事件有：
 - Escape mode request (LP-11→LP-10→LP-00→LP-01→LP-00)
 - High-Speed mode request (LP-11→LP-01→LP-00)
 - Turnaround request (LP-11→LP-10→LP-00→LP-10→LP-00)
- Escape mode是数据Lane在LP状态下的一种特殊操作
 - 在这种模式下，可以进入一些额外的功能：LPDT, ULPS, Trigger
 - 数据Lane进入Escape mode模式通过LP-11→LP-10→LP-00→LP-01→LP-00
 - 一旦进入Escape mode模式，发送端必须发送1个8-bit的命令来响应请求的动作
 - Escape mode 使用Spaced-One-Hot Encoding
- 超低功耗状态（Ultra-Low Power State）
 - 这个状态下，lines处于空状态 (LP-00)
- 时钟Lane的超低功耗状态
 - 时钟Lane通过LP-11→LP-10→LP-00进入ULPS状态
 - 通过LP-10 → TWAKEUP → LP-11退出这种状态，最小TWAKEUP时间为1ms
- 高速数据传输
 - 发送高速串行数据的行为称为高速数据传输或触发（burst）
 - 全部Lanes门同步开始，结束的时间可能不同。
 - 时钟应该处于高速模式
- 各模操作式下的传输过程
 - 进入Escape模式的过程：LP-11→LP-10→LP-00→LP-01→LP-00→Entry Code → LPD (10MHz)
 - 退出Escape模式的过程：LP-10→LP-11
 - 进入高速模式的过程：LP-11→LP-01→LP-00→SoT(00011101) → HSD (80Mbps ~ 1Gbps)
 - 退出高速模式的过程：EoT→LP-11
 - 控制模式 - BTA 传输过程：LP-11→LP-10→LP-00→LP-10→LP-00
 - 控制模式 - BTA 接收过程：LP-00→LP-10→LP-11
- 状态转换关系图



四、DSI介绍

1、DSI是一种Lane可扩展的接口，1个时钟Lane/1-4个数据Lane

- DSI兼容的外设支持1个或2个基本的操作模式：
 - Command Mode（类似于MPU接口）
 - Video Mode（类似于RGB接口）- 必须用高速模式传输数据，支持3种格式的数据传输
- • Non-Burst 同步脉冲模式
- • Non-Burst 同步事件模式
- • Burst模式
- 传输模式：
 - 高速信号模式（High-Speed signaling mode）
 - 低功耗信号模式（Low-Power signaling mode）- 只使用数据lane 0（时钟是由DP，DN异或而来）。
- 帧类型
 - 短帧：4 bytes (固定)
 - 长帧：6~65541 bytes (可变)
- 两个数据Lane高速传输示例



2、短帧结构

- 帧头部（4个字节）
 - 数据标识(DI) 1个字节
 - 帧数据- 2个字节（长度固定为2个字节）
 - 错误检测(ECC) 1个字节
- 帧大小
 - 长度固定为4个字节

3、长帧结构

- 帧头部（4个字节）
 - 数据标识(DI) 1个字节

- 数据计数- 2个字节 （数据填充的个数）
- 错误检测(ECC) 1个字节
- 数据填充(0~65535 字节)
 - 长度=WC*字节
- 帧尾： 校验和 （2个字节）
- 帧大小：
 - $4 + (0\sim65535) + 2 = 6 \sim 65541$ 字节

4、帧数据类型

Data Type,		Description	Packet Size	DCS	VD PKT	GN PKT
(hex)	(binary)					
01h	00 0001	Sync Event, V Sync Start	Short		O	
11h	01 0001	Sync Event, V Sync End	Short		O	
21h	10 0001	Sync Event, H Sync Start	Short		O	
31h	11 0001	Sync Event, H Sync End	Short		O	
08h	00 1000	End of Transmission Packet	Short	O	O	O
02h	00 0010	Color Mode (CM) Off Command	Short		O	
12h	01 0010	Color Mode (CM) On Command	Short		O	
22h	10 0010	Shut Down Peripheral Command	Short		O	
32h	11 0010	Turn On Peripheral Command	Short		O	
03h	00 0011	Generic Short WRITE, no parameters	Short			NOP
13h	01 0011	Generic Short WRITE, 1 parameter	Short			O
23h	10 0011	Generic Short WRITE, 2 parameters	Short			O
04h	00 0100	Generic READ, no parameters	Short			NOP
14h	01 0100	Generic READ, 1 parameter	Short			O
24h	10 0100	Generic READ, 2 parameters	Short			O
05h	00 0101	DCS WRITE, no parameters	Short	O	O	O
15h	01 0101	DCS WRITE, 1 parameter	Short	O	O	O
06h	00 0110	DCS READ, no parameters	Short	O	O	O
37h	11 0111	Set Maximum Return Packet Size	Short	O	O	O
09h	00 1001	Null Packet, no data	Long	O	O	O
19h	01 1001	Blanking Packet, no data	Long		O	
29h	10 1001	Generic Long Write	Long			O
39h	11 1001	DCS Long Write/write_LUT Command Packet	Long	O	O	O
0Eh	00_1110	Packet Pixel Stream, 16bit RGB 5-6-5 Format (Support for 1 and 2 data lanes mode)	Long		O	
1Eh	01_1110	Packet Pixel Stream, 18bit RGB 6-6-6 Format (Support for 1 and 2 data lanes mode)	Long		O	
2Eh	10_1110	Packet Pixel Stream, 18bit RGB Loosely 6-6-6 Format (Support for 1 and 2 data lanes mode)	Long		O	
3Eh	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format (Support for 1 , 2 and 3 data lanes mode)	Long		O	
x0h&Fh,	xx 0000	DO NOT USE				
	xx 1111	All unspecified codes are reserved				

具体代码在drivers/video/msm/Mipi_dsi.h中

```
[cpp]

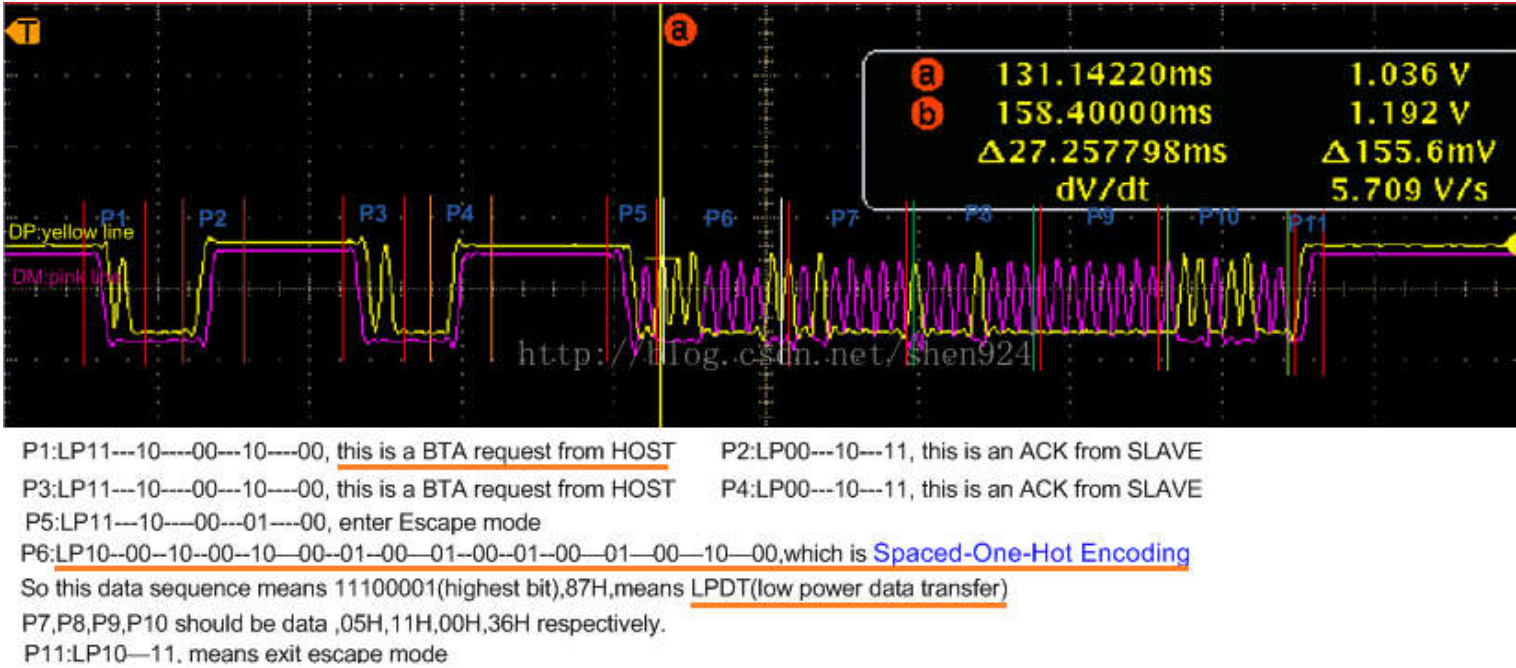
01.  /* dcs read/write */
02.  #define DTYPE_DCS_WRITE      0x05    /* short write, 0 parameter */
03.  #define DTYPE_DCS_WRITE1    0x15    /* short write, 1 parameter */
04.  #define DTYPE_DCS_READ      0x06    /* read */
05.  #define DTYPE_DCS_LWRITE    0x39    /* long write */
06.
07.  /* generic read/write */
08.  #define DTYPE_GEN_WRITE      0x03    /* short write, 0 parameter */
09.  #define DTYPE_GEN_WRITE1    0x13    /* short write, 1 parameter */
10.  #define DTYPE_GEN_WRITE2    0x23    /* short write, 2 parameter */
11.  #define DTYPE_GEN_LWRITE    0x29    /* long write */
12.  #define DTYPE_GEN_READ      0x04    /* long read, 0 parameter */
13.  #define DTYPE_GEN_READ1     0x14    /* long read, 1 parameter */
14.  #define DTYPE_GEN_READ2     0x24    /* long read, 2 parameter */
15.
16.  #define DTYPE_TEAR_ON        0x35    /* set tear on */
17.  #define DTYPE_MAX_PKT_SIZE   0x37    /* set max packet size */
18.  #define DTYPE_NULL_PKT      0x09    /* null packet, no data */
19.  #define DTYPE_BLANK_PKT     0x19    /* blanking packet, no data */
20.
21.  #define DTYPE_CM_ON          0x02    /* color mode off */
22.  #define DTYPE_CM_OFF         0x12    /* color mode on */
23.  #define DTYPE_PERIPHERAL_OFF 0x22
24.  #define DTYPE_PERIPHERAL_ON 0x32
25.
26.  /*
27.   * dcs response
28.   */
```



```
29. #define DTYPE_ACK_ERR_RESP      0x02
30. #define DTYPE_EOT_RESP          0x08    /* end of tx */
31. #define DTYPE_GEN_READ1_RESP    0x11    /* 1 parameter, short */
32. #define DTYPE_GEN_READ2_RESP    0x12    /* 2 parameter, short */
33. #define DTYPE_GEN_LREAD_RESP    0x1a
34. #define DTYPE_DCS_LREAD_RESP    0x1c
35. #define DTYPE_DCS_READ1_RESP    0x21    /* 1 parameter, short */
36. #define DTYPE_DCS_READ2_RESP    0x22    /* 2 parameter, short */
```

五、MIPI DSI信号测量实例

1、MIPI DSI在Low Power模式下的信号测量图



2、MIPI的D-PHY和DSI的传输方式和操作模式

- **D-PHY**和**DSI**的传输模式
 - 低功耗（**Low-Power**）信号模式（用于控制）：10MHz (max)
 - 高速（**High-Speed**）信号模式（用于高速数据传输）：80Mbps ~ 1Gbps/Lane
- **D-PHY**的操作模式
 - **Escape mode, High-Speed(Burst) mode, Control mode**
- **DSI**的操作模式
 - **Command Mode**（类似于**MPU**接口）
 - **Video Mode**（类似于**RGB**接口）- 必须用高速模式传输数据

3、小结论

- 传输模式和操作模式是不同的概念
- **Video Mode**操作模式下必须使用**High-Speed**的传输模式
- **Command Mode**操作模式并没有规定使用**High-Speed**或**Low Power**的传输模式，或者说
- 即使外部**LCD**模组为**Video Mode**，但通常在**LCD**模组初始化时还是使用**Command Mode**模式来读写寄存器，因为在低速下数据不容易出错并且容易测量。
- **Video Mode**当然也可以用**High-Speed**的方式来发送指令，**Command Mode**操作模式也可以使用**High-Speed**，只是没有必要这么做

顶 7 踩 0

上一篇 LABC and CABC-MTK Platform
下一篇 mipi 调试经验

相关文章推荐

- [MIPI DSI协议介绍](#)
 - [MIPI-DSI/CSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
- [MIPI DSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
 - [MIPI DSI协议介绍](#)
 - [MIPI DSI 协议介绍](#)



参考知识库



.NET知识库
3950 关注 | 839 收录

猜你在找

- [解析移动应用的身份认证，数据分析及信息推送](#)
- [移动APP测试基础到进阶](#)
- [移动应用的用户体验设计](#)
- [移动手机APP测试从零开始（初级篇）](#)
- [【魅族大赛技术公开课】移动应用开发技术精选](#)
- [如何打造移动环境下满足业务场景的高可用架构](#)
- [TalkingData研发副总裁阎志涛：Spark在TalkingData中的应用](#)
- [hbuilder开发移动app视频教程, mui 视频教程, html5 视频教程](#)
- [帆软报表FineReport 8.0移动端集成视频教程](#)
- [移动跨平台开发一步到位](#)



查看评论

2楼 [lover4ever](#) 2015-03-02 15:49发表



额，看到了，数据类型为0E,1E,2E,3E的后面就是跟的RGB数据，只是具体的还有些不太明白。显示图像的时候是不是这样的？

VSYNC->
HSYNC->图像数据->HSYNC_END
HSYNC->图像数据->HSYNC_END
HSYNC->图像数据->HSYNC_END
HSYNC->图像数据->HSYNC_END
VSYNC_END
??

1楼 [lover4ever](#) 2015-03-02 14:36发表



楼主，我看了你的帖子，还是没搞懂在高速模式下图像数据是怎么传输的

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- [全部主题](#)
- [Hadoop](#)
- [AWS](#)
- [移动游戏](#)
- [Java](#)
- [Android](#)
- [iOS](#)
- [Swift](#)
- [智能硬件](#)
- [Docker](#)
- [OpenStack](#)
- [VPN](#)
- [Spark](#)
- [ERP](#)
- [IE10](#)
- [Eclipse](#)
- [CRM](#)
- [JavaScript](#)
- [数据库](#)
- [Ubuntu](#)
- [NFC](#)
- [WAP](#)
- [jQuery](#)
- [BI](#)
- [HTML5](#)
- [Spring](#)
- [Apache](#)
- [.NET](#)
- [API](#)
- [HTML](#)
- [SDK](#)
- [IIS](#)
- [Fedora](#)
- [XML](#)
- [LBS](#)
- [Unity](#)
- [Splashtop](#)
- [UML](#)
- [components](#)
- [Windows Mobile](#)
- [Rails](#)
- [QEMU](#)
- [KDE](#)
- [Cassandra](#)
- [CloudStack](#)
- [FTC](#)
- [coremail](#)
- [OPhone](#)
- [CouchBase](#)
- [云计算](#)
- [iOS6](#)
- [Rackspace](#)
- [Web App](#)
- [SpringSide](#)
- [Maemo](#)
- [Compuware](#)
- [大数据](#)
- [apttech](#)
- [Perl](#)
- [Tornado](#)
- [Ruby](#)
- [Hibernate](#)
- [ThinkPHP](#)
- [HBase](#)
- [Pure](#)
- [Solr](#)
- [Angular](#)
- [Cloud Foundry](#)
- [Redis](#)
- [Scala](#)
- [Django](#)
- [Bootstrap](#)

