



VC0706

TV Camera Processor

使用指南

(SPI)

Preliminary[†]

Version 0.9

January 30, 2008

Notes1: The information is subject to change without notice. Before using this document, please confirm that this is the latest version.

Notes2: Not all products and/or types are available in every country. Please check with a Vimicro sales representative for availability and additional information.

Important Notice

All rights about this document belong to Vimicro Corporation (here after, refer as Vimicro). All rights are reserved.

Vimicro and its subsidiaries reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Customers should contact Vimicro's sales department before purchasing the product described in this document. All products are sold subject to Vimicro's terms and conditions of sale supplied at the time of order acknowledgment.

Vimicro does not warrant or represent that any license, either explicit or implied, is granted under any Vimicro patent right, copyright, mask work right, or other Vimicro intellectual property right relating to any combination, machine, or process in which Vimicro products or services are used. Information published by Vimicro regarding third-party products or services does not constitute a license from Vimicro to use such products or service or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Vimicro under the patents or other intellectual property of Vimicro.

Vimicro semiconductor devices are intended for standard uses (such as office equipment, computers, industrial/communications/measuring equipment, and personal/home equipment). Customers using semiconductor devices for special applications (including aerospace, nuclear, military and medical applications) in which a failure or malfunction might endanger life or limb and which require extremely high reliability must contact our Sales Department first. If damage is caused by such use of our semiconductor devices without first consulting our Sales Department, Vimicro will not assume any responsibility for the loss.

The contents of this document must not be reprinted or duplicated without written permission of Vimicro. Information and circuit diagrams in this document are only examples of device application. They are not intended to be used in actual equipment. Vimicro accepts no responsibility for infringement of patents or other rights owned by third parties caused by use of the information and circuit diagrams in this document.

Reproduction of information in Vimicro data books or data sheets is permissible only if preproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. Vimicro is not responsible or liable for such altered documentation. Resale of Vimicro products or services with statements different from or beyond the parameters stated by Vimicro for that product or service voids all explicit and any implied warranties for the associated Vimicro product or service and is an unfair and deceptive business practice. Vimicro is not responsible or liable for any such statements.

Table of Content

1. 简介	4
2. SPI 接口	4
2.1. SPI PIN	4
2.2. SPI 工作模式	4
2.3. SPI 配置	5
2.3.1. Char Length	5
2.3.2. MSB first or LSB first	5
2.3.3. Polarity & Phase	5
2.3.4. 传输速度	5
2.4. 通信协议	7
2.5. 通信命令	7
2.5.5. 命令列表	7
2.5.6. Frame Buffer	8
2.5.7. 操作步骤	8
2.5.7.1. 从 VC0706 读取图像	8
2.5.7.2. 向 VC0706 写入图像	9
2.5.8. 命令详细说明	9
2.5.8.1. GET_FBUF_LEN	9
2.5.8.2. SET_FBUF_LEN	10
2.5.8.3. FBUF_CTRL	10
2.5.8.4. READ_FBUF	10
2.5.8.5. WRITE_FBUF	11
3. REVISION HISTORY	12

1. 简介

本文档就 VC0706 的 SPI (Serial Peripheral Interface) 使用加以说明，以帮助用户在其设计中通过 SPI 与 VC0706 进行通信，以进行视频采集，或者写入图像利用 VC0706 的 JPEG 解码器进行解码并通过 VC0706 内置的 Video DAC 进行输出。

为获知 VC0706 芯片的基本说明，请参考 *VC0706 Datasheet*。

本文档以运行调试工具的外部主控 MCU 为发送端，以 VC0706 为接收端。

2. SPI 接口

2.1. SPI Pin

PIN NUMBER	NAME	TYPE	DESCRIPTION
80	UART_TXD	O	High speed UART data output
	GPIO14	O	GPIO 14 , only can be used as output mode
	SPI_MOD	I, PD	Hardware strap pin : SPI_MOD
89	SPI_SS	I/O, PU	SPI master interface slave select output
	GPIO10	I/O, PU	GPIO 10
90	SPI_SCK	I/O, PD	SPI master interface serial clock output
	GPIO11	I/O, PD	GPIO 11
91	SPI_MOSI	I/O, PD	SPI master interface master output
	GPIO12	I/O, PD	GPIO 12
92	SPI_MISO	I/O, PD	SPI master interface slave input
	GPIO13	I/O, PD	GPIO 13

2.2. SPI 工作模式

VC0706 的 SPI 接口支持两种模式：

- ! Master mode: VC0706 作为 SPI 的主设备，主要用于连接 SPI Flash 或者其他 SPI 从设备
- ! Slave mode: VC0706 作为 SPI 的从设备，主要用于连接外部控制器

VC0706 的 SPI 接口模式是通过芯片复位期间管脚 80 (UART_TXD/GPIO14/SPI_MOD) 的状态进行选择：

- ! 0: VC0706 工作于 SPI 的 master mode
- ! 1: VC0706 工作于 SPI 的 slave mode

在外部主控 MCU 通过 SPI 连接 VC0706 进行控制并读取视频采集或者写入图像的应用中，VC0706 工作于 slave mode。

在外部主控 MCU 通过 SPI 连接 VC0706 进行控制并读取视频采集或者写入图像的应用中，VC0706 的 SPI 不能再用于连接 SPI Flash 或者其他 SPI 设备。

2.3. SPI 配置

要使用外部主控 MCU 通过 SPI 连接 VC0706，除上述说明需要通过管脚 80 在芯片复位期间的状态来选择 Slave Mode 之外，还需要对 SPI 的一些参数进行设置。

2.3.1.Char Length

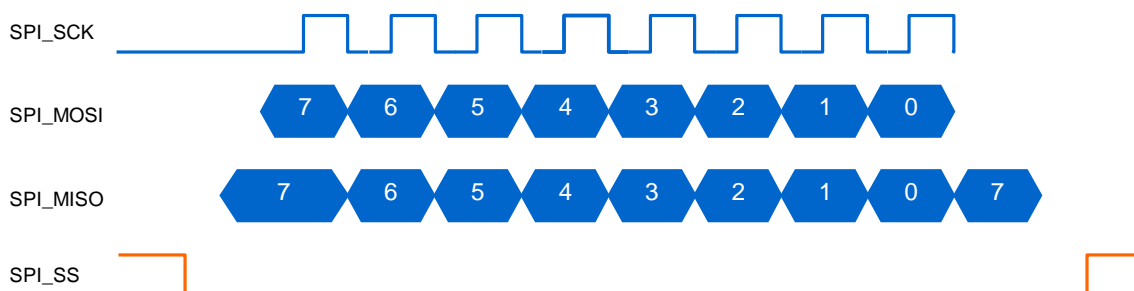
VC0706 只支持 8-bit 的 Char Length。

2.3.2.MSB first or LSB first

VC0706 只支持 MSB first 的传输方式。

2.3.3.Polarity & Phase

根据 SPI 数据有效时刻 SPI_SCK 的极性与上升/下降沿状态的不同，SPI 支持不同的 polarity & phase，VC0706 的 SPI 只支持下述模式，外部主控 CPU 需要配置为 VC0706 支持的模式：

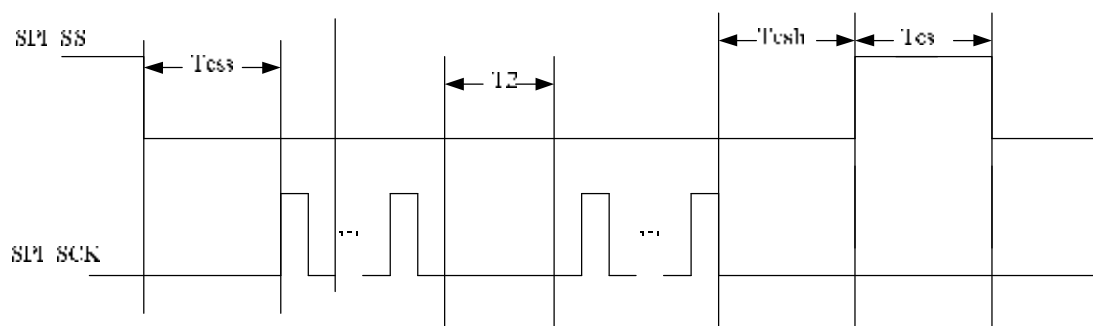


2.3.4.传输速度

VC0706 的 SPI 支持不同的传输速度，用户可以通过修改 EEPROM 中地址 0x000E 的值来进行配置。

地址	功能	长度 (字节)	缺省值	功能描述
0x000E	SPI BPS	0x04	0x000D, 0x0200	SPI 口的波特率； 前两个字节用于设置 SPI 口的 DIVIDER 寄存器； 后两个字节用于设置 SPI 口的 SS 寄存器； 计算公式如下所示：

				<div>DIVIDER = SPI_MCLK/(2*BAUD) -1</div> <div>其中：SPI_MCLK，取值为 27M</div> <div>BAUD 为要设置的 SPI 的波特率</div> <div>DIVIDER 为写入 DIVIDER 寄存器的值</div> <div>写入 SS 寄存器的值默认为 0x0200</div> <div>常用波特率的 INTER 和 FRAC 对照表如下所示：</div> <table><thead><tr><th>波特率</th><th>DIVIDER</th><th>SS</th></tr></thead><tbody><tr><td>0.844M</td><td>0x000F</td><td>0x0200</td></tr><tr><td>1.688M</td><td>0x0007</td><td>0x0200</td></tr><tr><td>3.375M</td><td>0x0003</td><td>0x0200</td></tr><tr><td>6.750M</td><td>0x0001</td><td>0x0200</td></tr><tr><td>13.500M</td><td>0x0000</td><td>0x0200</td></tr></tbody></table>	波特率	DIVIDER	SS	0.844M	0x000F	0x0200	1.688M	0x0007	0x0200	3.375M	0x0003	0x0200	6.750M	0x0001	0x0200	13.500M	0x0000	0x0200
波特率	DIVIDER	SS																				
0.844M	0x000F	0x0200																				
1.688M	0x0007	0x0200																				
3.375M	0x0003	0x0200																				
6.750M	0x0001	0x0200																				
13.500M	0x0000	0x0200																				



Tcss: SPI_SS setup time (需要满足 $>4 * T_{SPI_MCLK}$)

Tcsh: SPI_SS hold time (需要满足 $>2 * T_{SPI_MCLK}$)

Tcs: SPI_SS high time (需要满足 $>2 * T_{SPI_MCLK}$)

T_{SPI_SCK} : SPI_SCK 的周期

T2: 外部主控 MCU 通过 SPI 发送字节块的间隔时间

N1: 一帧数据需要几次间隔 (传输一帧数据所需要的字节块数-1)

N2: 一帧要传多少数据

T: 传一帧总的的时间

$$T = T_{SPI_SCK} * 8 * N2 + N1 * T2 + Tcss + Tcsh + Tcs$$

Tcss + Tcsh + Tcs: 基本上可以忽略不记

以传送 1 帧 64kB 大小的画面为例，SPI_MCLK 为 27MHz，以最高传输速度（13.5M）来传输，假定外部主控 MCU 的 SPI Buffer 为 16 字节，每次处理 SPI Buffer 所需要时间为 5us，则传输整帧画面所需要的时间为：

$$T=(1/13.5)*8*(64*1024)+((64*1024)/16-1)*5+(4+2+2)*(1/27)=59311\mu s$$

2.4. 通信协议

外部主控 MCU 通过 SPI 与 VC0706 进行通信的通信协议与 VC0703/VC0706 的 UART 串口通信协议格式完全兼容。

串口通讯协议格式如下所示：

接收的命令格式：协议标志（1 字节）+序列号（1 字节）+命令字（1 字节）+数据长度（1 字节）+数据（0~16 字节）

回复的命令格式：协议标志（1 字节）+序列号（1 字节）+命令字（1 字节）+状态字节（1 字节）+数据长度（1 字节）+数据（0~16 字节）；

其中：

协议标志：用于标志该协议为 VC0706 串口通讯协议；接收的协议标志为 0x56（‘V’），回复的协议标志为 0x76（‘v’）。

序列号：用于在多个设备同时连接时区分各个设备，只有串口命令的序列号与设备的序列号相同，设备才会接受这些命令。序列号的取值范围为 0~255。

命令字：用于标识具体的串口通讯命令。

数据长度：表示后面数据的长度，不包括协议标志、序列号、命令字和数据长度。取值范围为 0~16。

数据：命令使用到的数据，不同的命令，数据的长度和格式有所不同，最大为 16 字节。

状态字节：对于控制命令、数据写命令，数据的第一个字节表示命令是否成功，0 表示成功。0x01~0xFF 表示失败。

状态类型含义如下所示：

状态代码	错误说明
0	命令执行正确
1	非法的命令字，表示命令字系统不支持
2	非法的数据长度，表示该命令不支持该长度的数据长度值
3	数据格式非法
4	当前的系统状态下命令无法执行
5	命令接受，但是命令执行失败

说明：

- ！ 对于多字节数据类型，数据排列方式为高位数据在前，低位数据在后。
- ！ 如果是序列号错误，则系统不返回任何错误报告。
- ！ 命令中的数据的长度最大为 16 字节。
- ！ 如果命令格式或者命令执行失败，则返回的格式固定为 1 字节的错误代码，0 字节数据长度。

2.5. 通信命令

2.5.5. 命令列表

命令定义	命令字	命令含义
------	-----	------

GET_VERSION	0x11	获取 Firmware 程序版本
SET_SERIAL_NUMBER	0x21	设置序列号
SET_PORT	0x24	设置串口属性
SYSTEM_RESET	0x26	系统复位
READ_DATA	0x30	读取设备的寄存器
WRITE_DATA	0x31	向设备寄存器写入数据
READ_FBUF	0x32	从 FBUF 中读取图像
WRITE_FBUF	0x33	向 FBUF 中写入图像
GET_FBUF_LEN	0x34	获取 FBUF 中图像的长度
SET_FBUF_LEN	0x35	设置 FBUF 中图像的长度
FBUF_CTRL	0x36	对 FBUF 进行控制
COMM_MOTION_CTRL	0x37	启动或停止串口 Motion 监测
COMM_MOTION_STATUS	0x38	获取串口 Motion 监测状态
COMM_MOTION_DETECTED	0x39	启动了串口 Motion 检测时，用于串口发送检测到了 Motion
MIRROR_CTRL	0x3A	设置图像是否要 Mirror 显示
MIRROR_STATUS	0x3B	获取当前图像是否 Mirror 的状态
COLOR_CTRL	0x3C	颜色控制
COLOR_STATUS	0x3D	获取当前颜色控制模式和当前的颜色
POWER_SAVE_CTRL	0x3E	控制进入和退出节能模式
POWER_SAVE_STATUS	0x3F	检查是否进入节能模式
AE_CTRL	0x40	设置 AE 相关属性
AE_STATUS	0x41	获取 AE 相关属性
MOTION_CTRL	0x42	Motion 控制
MOTION_STATUS	0x43	获取 Motion 状态
TV_OUT_CTRL	0x44	控制打开和关闭视频输出
OSD_ADD_CHAR	0x45	添加 OSD 字符
GET_FLASH_SIZE	0x60	获取 SPI Flash 的大小
ERASE_FLASH_SECTOR	0x61	擦除 Flash 的一个块
ERASE_FLASH_ALL	0x62	擦除整个 Flash
READ_LOGO	0x70	读取 LOGO 并显示
SET_BIZTMAP	0x71	对 Bitmap 进行操作
BATCH_WRITE	0x80	批量写数据

2.5.6. Frame Buffer

在本文档中，FBUF 表示 Frame Buffer。VC0706 内部支持两个 frame buffer，分别称为当前帧和下一帧。

2.5.7. 操作步骤

外部主控 MCU 要通过 SPI 读取 VC0706 采集的视频或者通过 SPI 写入图像到 VC0706 的 FBUF，需要遵循一定的操作步骤。

2.5.7.1. 从 VC0706 读取图像

- | 发送 FBUF_CTRL 命令，参数为 0x00，来停止当前帧更新；
- | 发送 GET_FBUF_LEN 命令来获取当前帧中图像的长度；
- | 发送 READ_FBUF 命令来读取图像数据，READ_FBUF 命令中参数设置如下：
 - n FBUF 帧类型设置为 0x00
 - n 操作方式设置为 0x0F
 - n 起始地址设置为 0x00
 - n 数据长度设置为通过 GET_FBUF_LEN 命令获取的图像长度
 - n 延迟时间用于在数据和命令回复之间添加延迟，以区分数据和命令回复，默认值为 3000，可修改为其它值
- | 发送完 READ_FBUF 命令后，先等待 VC0706 的回应，回应正确后，等待接收 VC0706 发送的数据，在数据接收完毕后，VC0706 会再次发送命令回复通知外部主控 MCU，数据已经发送完毕。
- | 读取完毕后，发送 FBUF_CTRL 命令，参数为 0x02，来恢复帧的更新。

2.5.7.2. 向 VC0706 写入图像

- | 发送 FBUF_CTRL 命令，参数为 0x02，来停止所有帧更新；
- | 发送 WRITE_FBUF 命令来读取图像数据，WRITE_FBUF 命令中参数设置如下：
 - n 操作方式设置为 0x0F
 - n 起始地址设置为 0x00
 - n 数据长度设置为通过要写入的图像的字节长度
 - n 延迟时间用于在数据和命令回复之间添加延迟，以区分数据和命令回复，默认值为 3000，可修改为其它值
- | 发送完 WRITE_FBUF 命令后，先等待 VC0706 的回应，回应正确后，则向 VC0706 发送的图像数据，在数据发送完毕后，VC0706 会再次发送命令回复通知外部主控 MCU，数据已经接收完毕；
- | 发送 SET_FBUF_LEN 命令，参数为图像长度，来设置图像的长度；
- | 发送 FBUF_CTRL 命令，参数为 0x03，来将写入的图像切换到当前帧显示出来。

2.5.8. 命令详细说明

VC0706 SPI/UART 通信命令大部分与 VC0703 的 UART 串口通信命令相同，相同命令的详细说明请参考“VC0703 Reference Guide for UART”。本章节仅对新增加命令进行说明。

2.5.8.1. GET_FBUF_LEN

命令功能： 获取 FBUF 中图像的大小

命令格式： 0x56+序列号+0x34+0x01+FBUF 帧类型(1 字节)

FBUF 帧类型：表示要获取的是当前帧中的图像还是下一帧中的图像；

0：表示当前帧

1：表示下一帧。

回复格式：

如果运行正确，则回复：0x76+序列号+0x34+0x00+0x04+FBUF 长度(4 字节)

如果 FBUF 帧类型参数设置错误，则回复：0x76+序列号+0x34+0x03+0x00

举例：

0x56+0x00+0x34+0x01+0x00 获取当前帧的图像长度

0x56+0x00+0x34+0x01+0x01 获取下一帧的图像长度

说明:

该命令一般在读取 FBUF 图像前调用, 用于获取 FBUF 中图像的长度, 以便于读取 FBUF 图像。

2.5.8.2. SET_FBUF_LEN

命令功能: 设置 FBUF 的下一帧中图像的大小

命令格式: 0x56+序列号+0x35+0x04+FBUF 长度(4 字节)

FBUF 的长度最大为 65535。

回复格式: 0x76+序列号+0x35+0x00+0x00

举例:

0x56+0x00+0x35+0x04+0x00+0x00+0xA0+0xB0

设置 FBUF 中图像的长度为 0xA0B0

说明:

设置的图像大小必须与写入 FBUF 的图像大小相同, 否则可能会导致最终图像显示错误。

2.5.8.3. FBUF_CTRL

命令功能: FBUF 帧控制; 包括停止帧、切换帧和恢复帧。

命令格式: 0x56+序列号+0x36+0x01+控制标志(1 字节)

控制标志用表示要对 FBUF 进行控制的类型:

- 0: 停止当前帧
- 1: 停止下一帧
- 2: 恢复帧更新
- 3: 切换帧

回复格式:

如果运行正确, 则回复: 0x76+序列号+0x36+0x00+0x00

如果控制标志错误, 则回复: 0x76+序列号+0x36+0x03+0x00

举例:

0x56+0x00+0x36+0x01+0x00	停止当前帧
0x56+0x00+0x36+0x01+0x01	停止下一帧
0x56+0x00+0x36+0x01+0x02	恢复帧更新
0x56+0x00+0x36+0x01+0x03	切换帧

说明:**2.5.8.4. READ_FBUF**

命令功能: 从 FBUF 中读取图像

命令格式: 0x56+序列号+0x32+0x0C+ FBUF 帧类型(1 字节)+操作方式(1 字节)+起始地址(4 字节)+数据长度(4 字节)+延迟时间(2 字节)

FBUF 帧类型: 表示要获取的是当前帧中的图像还是下一帧中的图像;

- 0: 表示当前帧
- 1: 表示下一帧

操作方式: 表示传送 FBUF 数据方式

Bit0: 表示是否通过 DMA 方式传送数据

0: 通过 MCU 方式传送数据

1: 通过 DMA 方式传送数据

Bit[2:1]: 2'b11

Bit3: 1'b1

起始地址: 要读取的数据在 FBUF 中的起始地址。

数据长度: 要读取的数据的字节个数, 必须为 4 的倍数。

延迟时间: 命令与数据间的时间间隔, 单位为 0.01 毫秒

回复格式:

如果运行正确, 则回复: 0x76+序列号+0x32+0x00+0x00

然后再发送数据。

在发送完数据后, 再发送返回命令表示数据发送完毕。

如果参数设置或通讯错误, 则回复: 0x76+序列号+0x32+错误代码+0x01

举例:

1 0x56+0x00+0x32+0x0C+0x00+0x0F+0x00+0x00+0x00+0x10+0x00+0x00+0x02+0x00+0x10+0x00

通过 SPI 口以 DMA 方式读取 FBUF 当前帧的图像, 图像数据从地址 0x0100 开始, 长度为 0x0200, 命令与数据间的时间间隔为 40.96 毫秒

说明:

系统接收到命令后, 先进行命令回复, 然后再从 FBUF 中读取指定长度的数据并通过 SPI 发送给外部主控 MCU, 在发送完数据后, 再发送回复命令表示数据发送完毕。

在调用该命令之前必须先停止指定的帧, 否则 FBUF 中的图像一直在更新, 会导致读取错误的图像数据。

要读取 FBUF 中的图像, 可以通过本命令一次全部读取出来, 也可以分多次来读取。

为保证传输速度, 建议使用 DMA 方式读取数据;

2.5.8.5. WRITE_FBUF

命令功能: 往 FBUF 中写图像

命令格式: 0x56+序列号+0x33+0x0B+操作方式(1 字节)+起始地址(4 字节)+数据长度(4 字节)+延迟时间(2 字节)

操作方式: 表示传送 FBUF 数据方式

Bit0: 表示是否通过 DMA 方式传送数据

0: 通过 MCU 方式传送数据

1: 通过 DMA 方式传送数据

Bit[2:1]: 2'b11

Bit3: 1'b1

Bit4: 表示在向 FBUF 写图像的操作中本次写操作是否为第一次写操作

0: 不是第一次

1: 第一次

如果图像分多次写入, 则在第一次写时, 设置该位为 1, 在以后的操作中设置为 0。

起始地址: 要写入的数据在 FBUF 中的起始地址。

数据长度：要读取的数据的字节个数，必须为 4 的倍数。

延迟时间：命令与数据间的时间间隔，单位为 0.01 毫秒

回复格式：

如果运行正确，则回复：0x76+序列号+0x33+0x00+0x00

然后等待接收数据。

接收完数据后，再发送回复命令表示数据接收完毕。

如果参数设置或通讯错误，则回复：0x76+序列号+0x33+错误代码+0x00

举例：

I 0x56+0x00+0x33+0x0B+0x0F+0x00+0x00+0x00+0x10+0x00+0x00+0x02+0x00+0x10+0x00

通过 SPI 口以 DMA 方式向 FBUF 中写入图像，图像数据从地址 0x0100 开始，长度为 0x0200

说明：

系统接收到命令后，进行相关设置，然后进行命令回复，再等待接收数据，在接收完数据后，还要再进行命令回复。

在调用该命令之前必须先停止指定的帧。

为保证传输速度，建议使用 DMA 方式写入数据；

3. Revision History

Version No.	Remarks	Release Date
0.9	Preliminary release	2008-01-30