

# 骑飞电子 CMOS 摄像头

使

用

说

明

V1.2

骑飞电子

2012.1.9

## 目录

### 1. OV7670 一般摄像头模块

#### 1.1 简介

#### 1.2 管脚定义

#### 1.3 控制方式说明

#### 1.4 采集图像的基本方法

### 2. OV7670 带 FIFO 摄像头模块

#### 2.1 简介

#### 2.2 管脚定义

#### 2.3 控制方式说明

#### 2.4 图像采集的基本方法

### 3. 问题解答

#### 3.1 图像采集难吗

#### 3.2 学习图像方面的知识需要哪些基础

#### 3.3 初学者遇到问题该怎么解决

#### 3.4 模块提供哪些资料

#### 3.5 单片机能够真正的采集图像吗

#### 3.6 带 FIFO 和不带 FIFO 的模块到底哪个好，有什么区别

#### 3.7 模块上有晶振好，还是没晶振好

#### 3.8 摄像头寄存器该怎么设置

#### 3.9 骑飞电子模块提供的 Demo 输出的数据是什么格式的

- 3.10 如果想真正实现图像的采集并且能够处理图像数据该如何做
- 3.11 骑飞电子的那个模块能够适合飞思卡尔小车的比赛
- 3.12 骑飞电子的模块输出到底是模拟的还是数字的
- 3.13 骑飞电子模块的质量如何
- 3.14 骑飞电子几种驱动板的功能，区别是什么
- 3.15 如何检测骑飞电子摄像头模块是否损坏
- 3.16 骑飞电子摄像头模块和模组的区别是什么

## 1. OV7670 一般摄像头模块

### 1. 简介:

OV7670 一般模块指骑飞电子推出的低成本数字输出 CMOS 摄像头，其摄像头包含 30w 像素的 CMOS 图像感光芯片，3.6mm 焦距的镜头和镜头座，板载 CMOS 芯片所需要的各种不同电源（电源要求详见芯片的数据文件），板子同时引出控制管脚和数据管脚，方便操作和使用。

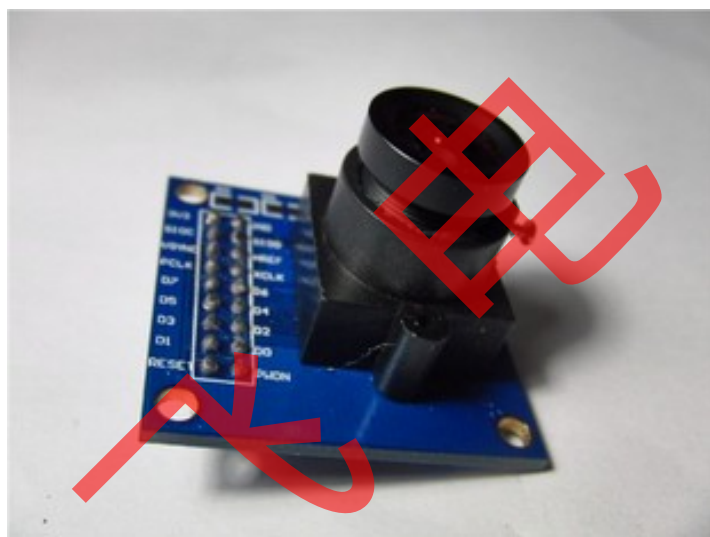


图 1. OV7670 一般模块

### 2. 管脚定义:

如图，控制传感器所需的管脚定义如下：

3V3-----输入电源电压（推荐使用 3.3，5V 也可，但不推荐使用）

GND-----接地点

SIO\_C---SCCB 接口的控制时钟（注意：部分低级单片机需要上拉控制，和 I2C 接口类似）

SIO\_D——SCCB 接口的串行数据输入（出）端（注意：部分低级单片机需要上拉控制，和 I2C 接口类似）

VSYNC——帧同步信号（输出信号）

HREF——行同步信号（输出信号）

PCLK——像素时钟（输出信号）

XCLK——时钟信号（输入信号）

D0-D7——数据端口（输出信号）

RESTE——复位端口（正常使用拉高）

PWDN——功耗选择模式（正常使用拉低）

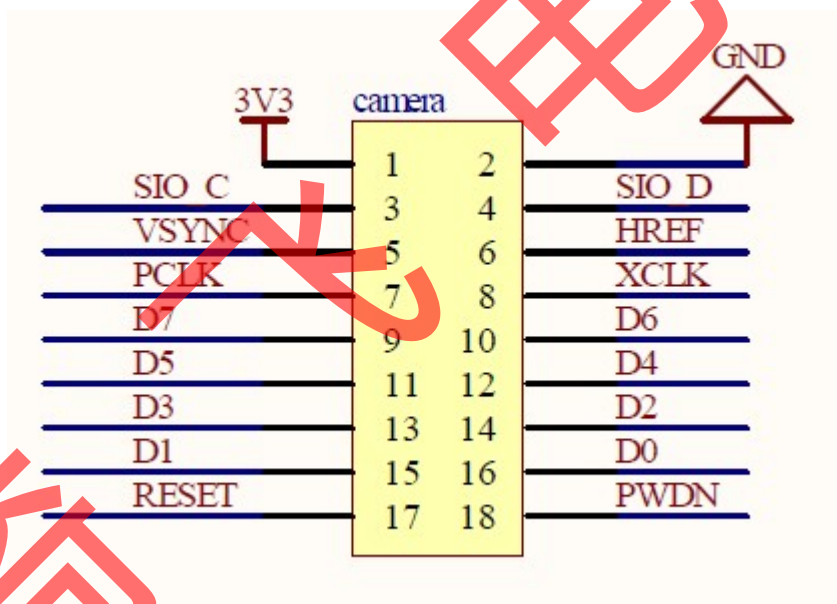


图 2. 骑飞摄像头接口定义

### 3. 控制方式说明

采集图像数据需要严格按照 OV 公司的芯片时序进行，这些时序包括：

（1）SCCB 通讯时序，其作用是设置芯片内部寄存器，以控制图像的各种所

需功能。其时序和一般的 I2C 时序相似，部分低级单片机要接上拉电阻。

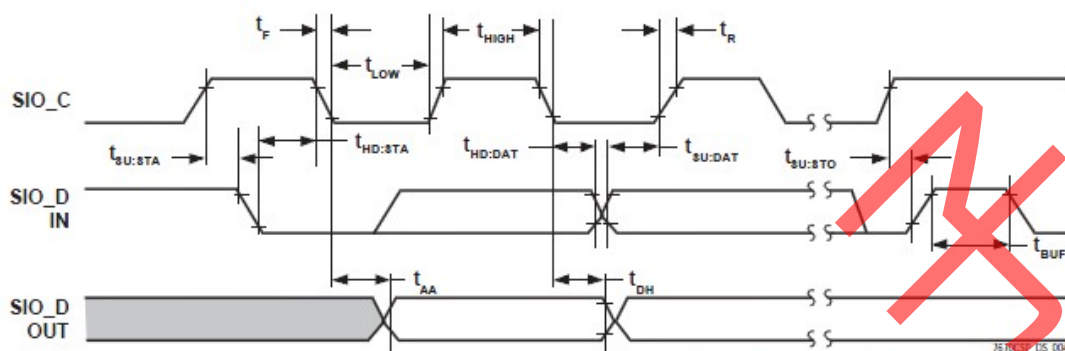


图 3. SCCB 时序图

## (2) 行输出时序

行输出时序可用来控制一行像素的输出情况，HREF 即一行输出的开始和结束信号，同时在像素时钟的同步下，输出 8 位的像素信号

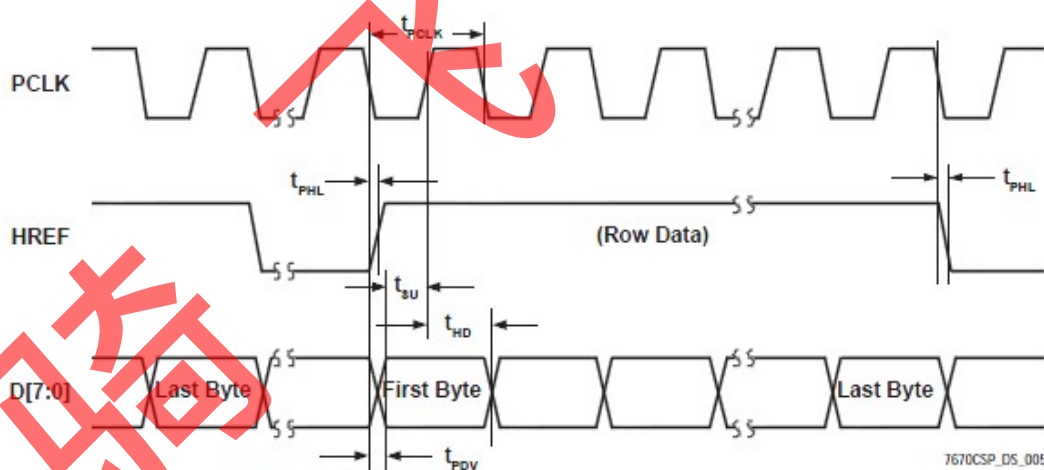


图 3. 行输出时序图

## (3) 全帧输出下的时序情况：(VGA 为例)

该图显示的是一副图像输出的情况下，各控制信号和数据信号的输出。

图中，VGA=640X480 大小情况下，帧同步信号，行同步信号（HREF 或

者 HSYNC, 注: HSYNC 在其它场合下使用, CMOS 可以设置, 更多时候用 HREF 即可) 如图:

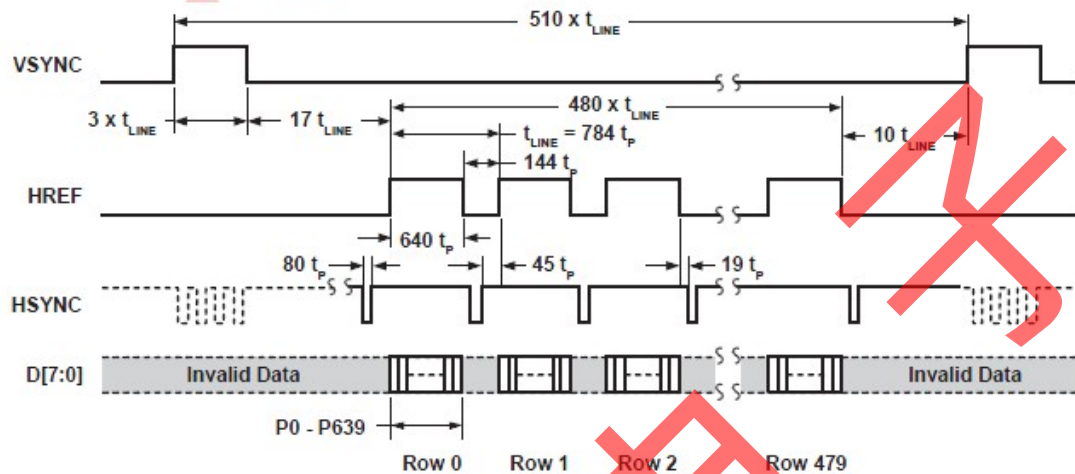


图 4. VGA 大小的帧输出情况下的时序图

#### 4. 采集图像的基本方法

##### (1) 单片机直接采集:

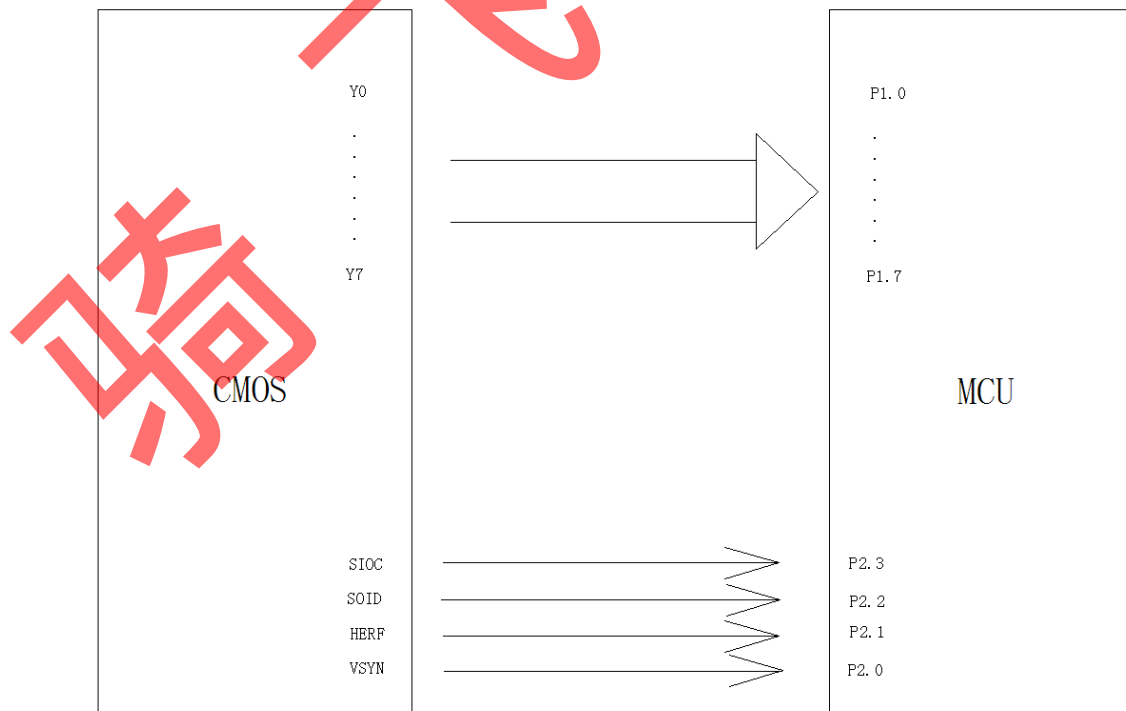


图 5 直接采集的连接方法

如图，这种方法是最简单，最直接，但也是最不好实现的方法，原因是多数的 CMOS 芯片（如 ov7670）的时钟速度可高达 24M，一般单片机的 I/O 端口速度根本不可能达到，所以需要高速 MCU。这对多数用户来讲有些不现实。

但也不是完全没有办法在低速上实现采集，方法也很简单，那么就是降低 CMOS 的输出速度，不过这要靠外部的晶振和内部的 PLL 电路以及像素时钟速度，帧速等多个寄存器共同设置，并且要和 MCU 的 I/O 速度匹配才可实现。但骑飞电子家族不建议这么做，原因是：这种寄存器设置将带来更多的学习困难和理解困难，并导致硬件图像的采集速度可能下降到 0.5 帧以下，同时带来图像失真的可能。

注：部分 CMOS 时钟速度不快，可以单片机直接采集，如 OV7660，但该芯片已经停产。

## （2）高级 32 位芯片直接使用

某些 32 位的 ARM 核，MIPS 核，x86 核直接带有 camera 接口，可以直接使用，典型的如三星的 arm9，sc2440 等。

该采集方法请参考具体的芯片器件数据手册。

## （3）DMA 方式的采集方法

这种方法一般需要具有 DMA 功能的 16 位以上的高级单片机来实现，通过 DMA 方式，直接从总线或者 I/O 口采集数据送入内存，内存中的数据可以存储，计算，保留，或者供显示使用。

## （4）间接采集并显示的方法（推荐使用）

这种方法是最容易实现，能够直接看到采集的图像内容。实现采集就显



示的功能，这种方法具有较高帧速，一般 QVGA 可保证在 30 帧左右的速度。

实际上，这种间接的办法其实就是将 CMOS 输出的数据直接送到显示屏（如 TFT 显示器的内存中）进行显示的，数据并不经过 MCU（所以对 MCU 而言是间接采集），也无法经过 MCU，因此可以达到较高的帧速。

## 2. OV7670 带 FIFO 模块

### 1. 简介:

OV7670 带 FIFO 模块，是针对慢速的 MCU 能够实现图像采集控制推出的带有缓冲存储空间的一种模块。这种模块增加了一个 FIFO（先进先出）存储芯片，同样包含 30w 像素的 CMOS 图像感光芯片，3.6mm 焦距的镜头和镜头座，板载 CMOS 芯片所需要的各种不同电源（电源要求详见芯片的数据文件），板子同时引出控制管脚和数据管脚，方便操作和使用。

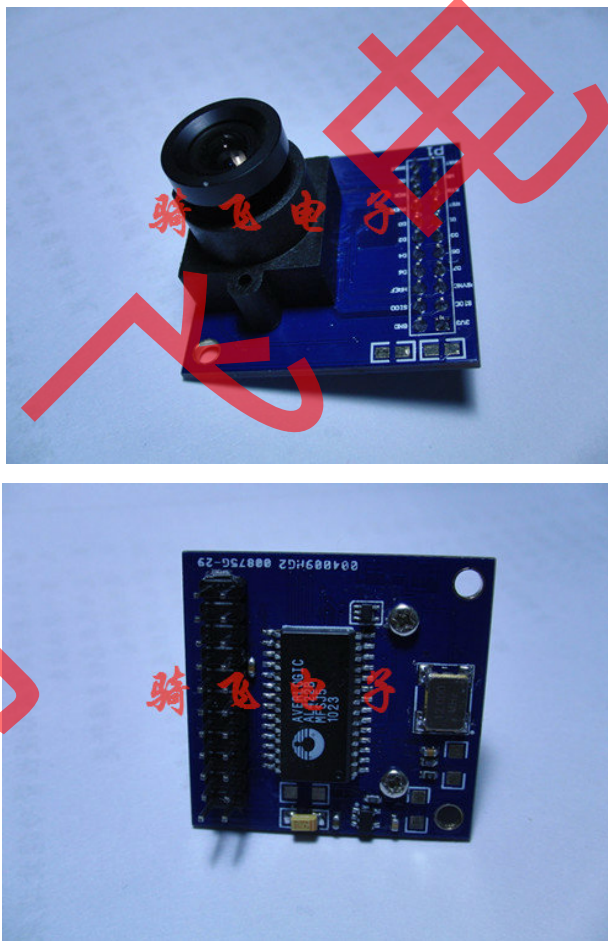


图 6. OV7670 带 FIFO 模块

### 2. 管脚定义:

如图，控制传感器所需的管脚定义如下：

3V3-----输入电源电压（推荐使用 3.3，5V 也可，但不推荐）

GDN-----接地点

SIO\_C---SCCB 接口的控制时钟（注意：部分低级单片机需要上拉控制，和 I2C 接口类似）

SIO\_D---SCCB 接口的串行数据输入（出）端（注意：部分低级单片机需要上拉控制，和 I2C 接口类似）

VSYNC---帧同步信号（输出信号）

HREF---行同步信号（输出信号）

PCLK---像素时钟（输出信号）

XCLK---时钟信号（输入信号）

D0-D7---数据端口（输出信号）

RESTE---复位端口（正常使用拉高）

PWDN---功耗选择模式（正常使用拉低）

STROBE---拍照闪光控制端口（正常使用可以不需要）

FIFO\_RCK---FIFO 内存读取时钟控制端

FIFO\_WR\_CTR---FIFO 写控制端(1 为允许 CMOS 写入到 FIFO，0 为禁止)

FIFO\_OE---FIFO 关断控制

FIFO\_WRST---FIFO 写指针服务端

FIFO\_RRST---FIFO 读指针复位端

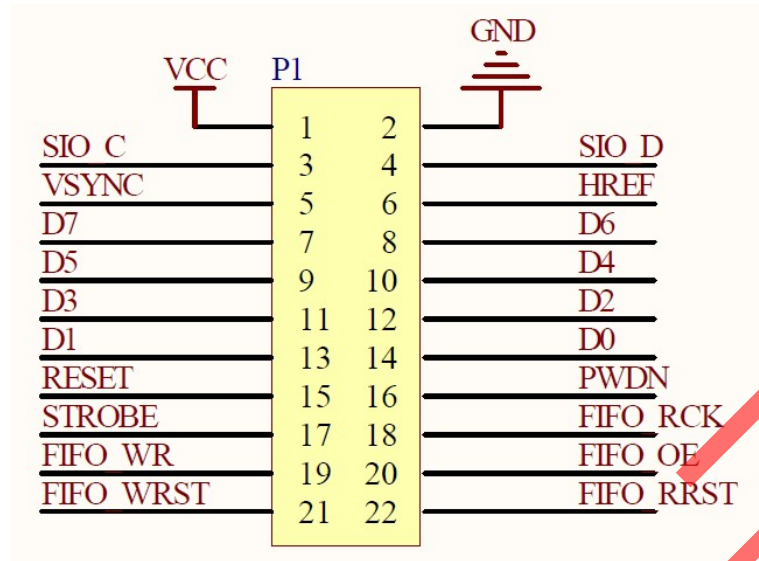
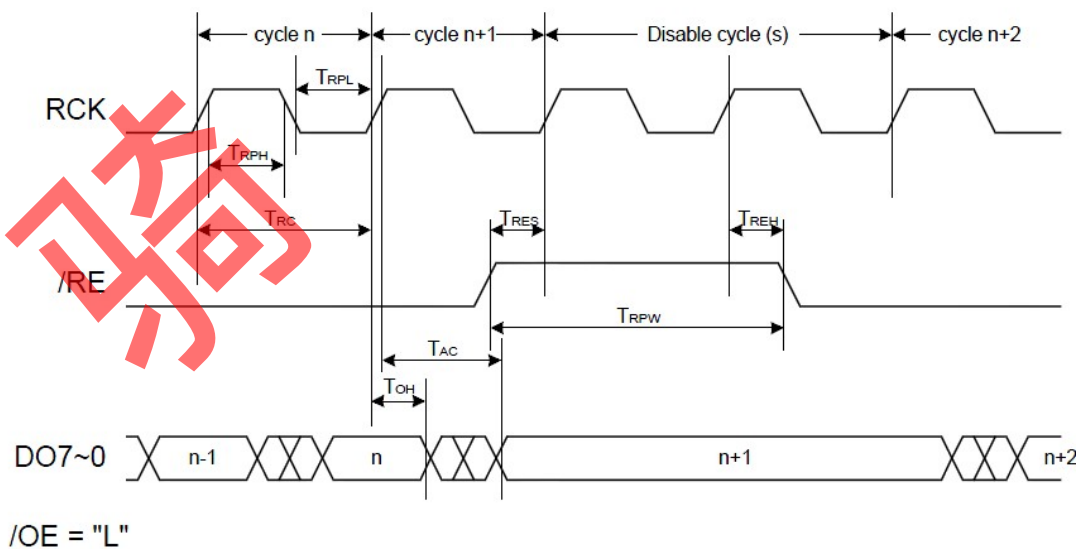


图 7. 骑飞 FIFO 摄像头接口定义

### 3. 控制方式说明

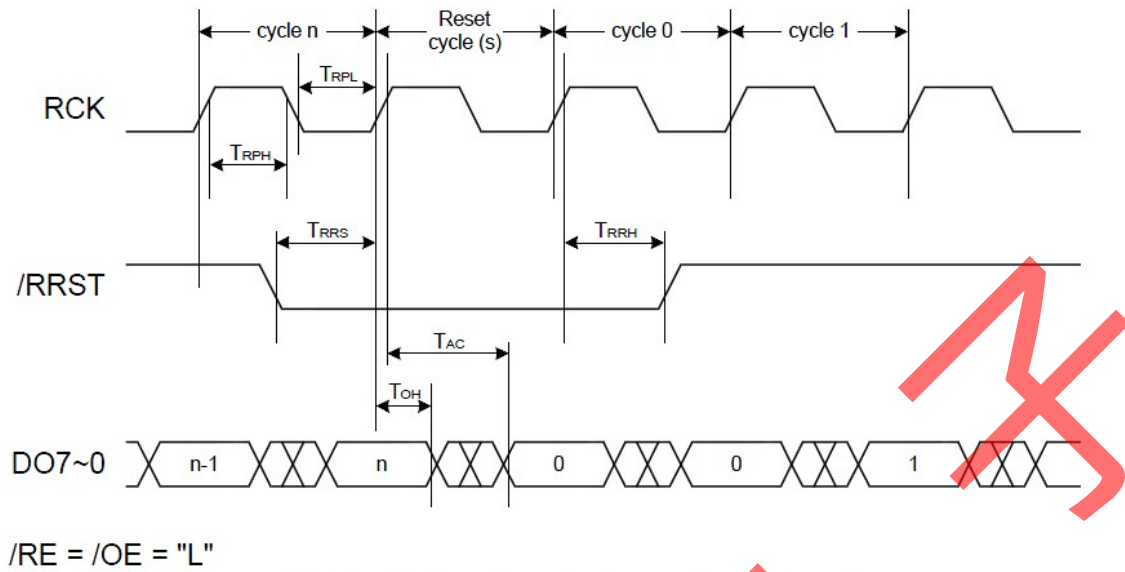
由于采用了 FIFO 做为数据缓冲，数据采集大大简便，用户只需要关心是如何读取即可，不需要关心具体数据是如何采集到的，这样可减小甚至不用关心 CMOS 的控制以及时序关系，就能够实现图像的采集。

控制时序如下：



AL422-08 Read Cycle Timing (Read Enable)

图 8. 骑飞 FIFO 摄像头读数据时序图

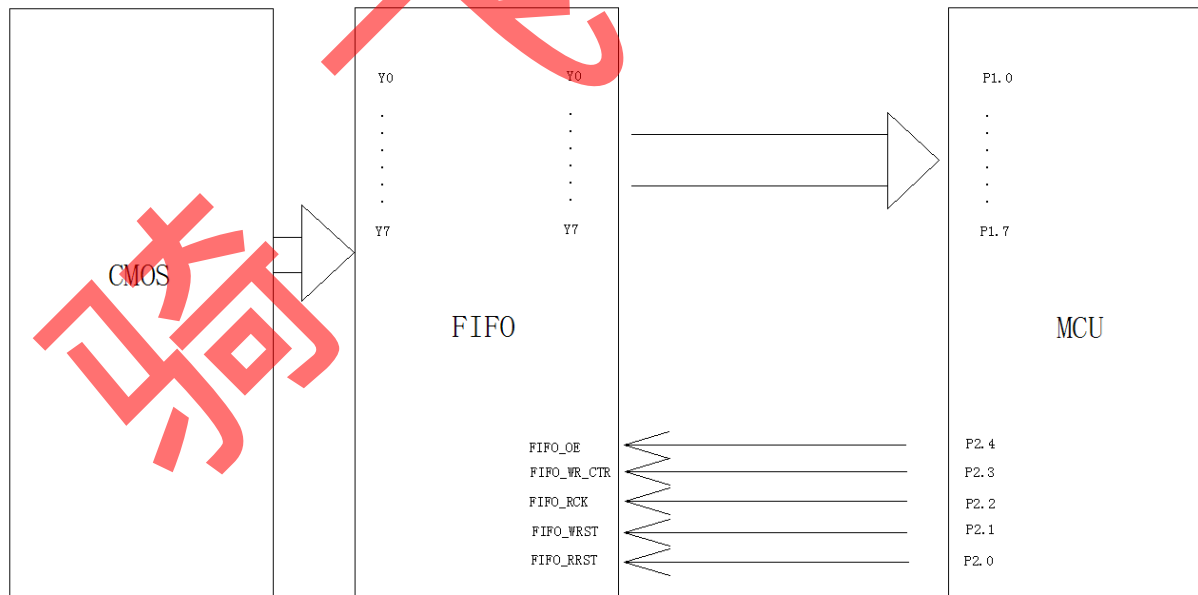


AL422-07 Read Cycle Timing (Read Reset)

图 9. 骑飞 FIFO 摄像头读数据时序图

## 4. 采集图像的基本方法

### (1) 单片机直接采集:



如图，用户只需要按上述时序图控制相关的几个控制引脚即可，可以很方便的使用在低速单片机上，另外一个好处是，可以直接 IO 口读取数据，

读出的数据可以直接送屏，也可以经过 MCU 简单处理；当然也可以不经过 MCU，直接送到屏等外围器件使用。

### 3. 问题解答

很多用户在使用本公司的模块时，遇到不少问题，这些问题既有技术问题，也和用户本身知识结构相关，针对这些集中的问题，骑飞家族对此进行问题汇总，并一一解答，希望用户在使用时能够了解。

#### 1. 图像采集难吗？

答：这个问题对初次接触的用户来讲经常会遇到。难和容易都是相对的，它和许多相关的基础知识紧密相联系。从涉及到的学科和内容上来讲，图像采集或者图像处理（硬件）可以讲是数字电子学中最难的，因为它涉及到光学，如颜色，白平衡，色空间，镜头（这些概念在设置传感器寄存器的时候会大量出现）等；涉及到传感器基本原理，如 CMOS 和 CCD 的原理和区别；涉及到电子学中的难题，如：海量数据量的传输、存储，涉及到主控制芯片的选择问题，如速度，和处理能力；涉及到电路板的高速布板问题；涉及到图像的编码和解码等算法问题。。。。。

特别是当像素不断增加的时候，这些问题会更加突出，因此客观的讲，如果和其它的传感器或者模块相比较，例如温度，无线模块等等还是有一定难度的，这些模块只要给出接口，一般都容易知道如何处理，但图像并不如此。

然而，难度是和相关基础知识的多少成反比例的，掌握的相关基础知识越多，就会越觉得简单。

#### 2. 学习图像方面的知识需要哪些基础？

答：首先，基本的光学知识要先了解，比如：什么是颜色空间，RGB, YUV



是什么，这些基础的必须了解；其次，扎实的电子学基础，比如：单片机基本上运用自如，时序图看起来没有任何难度；另外，较好的编程能力和读程序能力，这也是硬件图像的基础；最后，上位机中对图形图像的画法，处理等也有较深刻的认识。

### 3. 初学者遇到的问题该怎么解决？

答：很多用户都是初次学习图像硬件采集的相关内容，甚至刚刚开始单片机学习的用户也想学习这方面的内容。而实际上，他们遇到的问题很雷同，比如，图像中的一些基本概念如 RGB，YUV，模拟图像，数字图像等就比较模糊，电子学中如 FIFO 是什么，也经常被问到。其实，这些简单的基础知识或者概念只要 Google 或者 Baidu 一下，就可以快速找到大量的相关资料。

**友情提示：**由于骑飞电子技术人员紧缺，时间有限，所以希望能够抓住有限时间和精力开发更多、更好的受欢迎的开源产品，而不愿看到更多的时间花在相关基础知识的普及上，同时更愿意和用户一起探讨更高水平的图像技术和方案。这里骑飞电子诚恳的希望各位用户尊重知识的不易、尊重技术工作者的艰辛，谢谢理解！

### 4. 模块提供的是那些资料？

答：提供图像芯片的数据文件（这个很重要），这里要做个说明，那就是相关资料很少，得来不易。原因是：实际上这些数据文件都是 OV 公司不经意间流出的东西，做为个人，OV 公司一般根本不会提供这些内容的，它只和对应的有芯片需求的公司签订协议，提供相关资料，并且有保密



协议。我们获得的资料也是网上公开后得到的。除此之外，我们还提供中文的数据资料文件，SCCB 接口、模块定义、模块原理、模块尺寸等资料。此外，提供了一个基于 C8051F330 的驱动程序，该程序的原理是将模块数据直接送屏显示。

网上有人做过 AVR，STM32 单片机的驱动，原理类似，请自己搜索，至于用 8051 等单片机的驱动程序，我们没有做过，但触类旁通，只要理解了原理，驱动方法其实都是一样的。

#### 5. 单片机能够真正的采集图像吗？

答：这个明确的讲是不行的，特别是 8 位的单片机那更是不行的，必须寻求两种解决办法，一种是增加一个缓冲，如 FIFO，另外一个就是使用高速，如采用 32 位 MCU。8 位单片机实现的多数仅仅是提供一个 SCCB 总线读取或者设置的功能。即使是 32 位，比如大家都喜欢的 stm32，它也不能直接用 I/O 口采集，因为 I/O 速度还是不行，骑飞电子家族曾经做过实验，用 I/O 采集，实现的刷屏速率不到 1 帧，基本没有适用价值，除非使用 DMA 方式，也就是使用总线的带宽速度，可以实现采集，但这个速度也只能在 5-10 帧左右！

另外一个问题就是，即使采集到，由于一副图像的数据量很大，QVGA 的数据量是 320x240x2，你放到哪里呢？所以只能寻找 32 位以上的 MCU，骑飞电子自主开发了基于 stm32 的 DMA 方式的驱动板，解决了这个问题。其它的解决方案就是增加一个 FIFO(但不具备图像处理能力)。

## 6. 带 FIFO 和不带 FIFO 的模块到底哪个好，有什么区别？

答：按照上述几个问题，基本能够理清一个思路，那就是，如果针对 8 位单片机，图像采集是不现实的，只能用更高级的 MCU 或者采取更灵活的处理手段才能实现真正的图像采集。

那么这样就容易明白了，用那种摄像头是根据你的需求来确定的，如果你就是想显示下图像，8 位足够，不带 FIFO 的摄像头足矣，带个 TFT 显示屏即可！如果你的芯片是高级 MCU，ARM9 以上或者 DSP，那么也可以选择这个不带 FIFO 模块的，因为本身高档芯片的速度很快，存储空间也很大，足够直接采集图像数据！

如果还想对获取的图像数据做些非常简单的处理（注意是非常简单），请选择 FIFO 摄像头，由于 FIFO 不具备地址功能，因此他也就不具备数据的定位（选址）读取功能，所以不可能有真正的数据处理能力！请切记！！

另外，带 FIFO 摄像头模块避免了需要了解更多图像输入或者采集的原理，降低了学习难度，用户只需知道如何从 FIFO 中读取数据即可，因此更建议初学者使用该模块！

总结一下就是：两种模块的选择，要具体看项目的需求和目标来确定。

## 7. 模块上有晶振好，还是没晶振好？

答：晶振选择也是根据外围电路的能力来确定的。由晶振提供时钟（注：4-24M 都可，尽量是 4 的倍数，CMOS 内部有 PLL 功能）信号是可行的。

但骑飞电子认为，当前大多数的 MCU 都具有直接输出频率信号的能力，比如，有些芯片，如 CYPRESS，C8051f 等等，而且这个频率信号一

般非常稳定，所以强烈建议用这样的时钟信号直接供给摄像头。至少，如果能输出 50%占空比的 PWM，那么这个也是可以直接供给 CMOS 芯片使用的。

所以，带不带晶振根本没有本质区别，至于好坏，除了多增加成本外，别无它用。

#### 8. 摄像头寄存器该怎么设置

答：没有其它简洁的办法，只能参考 OV 公司的写的很烂数据手册，慢慢阅读理解和消化。稍快一点的办法就是按照骑飞电子提供的一个 Demo，针对其中寄存器的初始化设置和与其相对应的数据手册仔细推敲。当然也可以不去理解，直接使用即可！

#### 9. 骑飞电子模块提供的 Demo 输出的数据是什么格式的？

答：是 RGB565，QVGA 格式的。这样能够很好的和 320x240 的显示屏直接相匹配，方便使用和参考！

#### 10. 如果想真正实现图像的采集并且能够处理图像数据该如何做？

答：按下面几个方案选择：

(1) 如果想用 8 位实现采集和处理，基本不现实，简单处理的话，用带 FIFO 的摄像头；

(2) 32 位的高速芯片可以，有些直接有图像传感器接口，有些必须寻找一些办法，如骑飞电子基于 stm32f103 的 DMA 方式；

(3) 真正的图像采集和处理，使用 DSP 图像采集处理芯片，如 TI 的和 ADI 公司的芯片，但这些芯片价格超高，单片都在 80-200 元之间；

(4) FPGA 也可以实现类似功能，但如果要有速度和处理能力，价格也是几何数的增长；

11. 骑飞电子的模块那个能够适合飞思卡尔小车的比赛？

答：根据不少用户的反应，现在看来，选择带 FIFO 的头还是更现实些！

12. 骑飞电子的模块输出到底是模拟的还是数字的

答：是数字摄像头，它的输出是标准格式的，具体可以参考数据手册。

至于模拟摄像头一般如 AV 输出，有两种制式，使用时需要编码和解码才能和大多数的数字芯片或者屏连接。骑飞电子将在今后推出类似模块和驱动电路！

13. 骑飞电子模块的质量如何？

答：骑飞电子的摄像头模块除了插头手工焊接外，其它如 CMOS 芯片，贴片器件等都是机器焊接，每次成批按 1000 套制作，出厂前 100%检测，以避免坏件到用户手里产生不必要的折腾，骑飞电子和用户都惧怕这样的麻烦，所以出货前就尽量避免此类事情发生！

另外，值得一提的事，有些用户事先缺乏调研，或者缺乏相关知识架构，在使用了骑飞摄像头模块后，无法实现驱动，强行认为是骑飞模块质量问题，一般为避免矛盾，我们都给予了退货，但返回的模块经过检查后没有一个出现问题。

因此，在这里提醒极少数用户：请相互理解，相互尊重，尊重知识的不易、尊重技术工作者的艰辛、千万不要做个“无知者无谓”。

14. 骑飞电子几种驱动板的功能，区别是什么？

答：骑飞电子推出的几种驱动板，主要为了使使用者能够快速学习图像传感器的使用而推出的，目前陆续推出的有：

(1) stm32 的 DMA 方式图像采集板：

主要解决图像采集和处理问题，不但解决了直接采集的问题，也实现了在能够简单处理图像的功能；

(2) 基于 c8051f340 的多功能驱动板：

可以实现图像的显示，带有 usb，sd 卡，串口等外设，有其它多种功能；

(3) 基于 c8051f330 的简化驱动板

简化的驱动板，价格较低，也可实现图像显示；

(4) USB 图像采集板

实现通过 USB 采集图像的功能；

以上可参考淘宝店内容介绍，每种板都有相关光盘资料！

15. 如何检测骑飞电子摄像头模块是否损坏

答：通电，加时钟信号，REST 端拉高，PSWN 接地，示波器检测数据端是否有输出，如果有，说明 CMOS 是好的。

16. 骑飞电子摄像头模块和模组的区别是什么

答：模块就是将 CMOS 传感芯片，外围电路集成到一块 PCB 板子上，需要使用的控制管脚引出，装上摄像头，即称之为模块，如下图所示；





模组就是传感器和微型的镜头集成在一起，有软线 FPC 线引出管脚，但没有外围电路，使用时需要主板提供外围电路才能正常工作，如下图所示；



如果模组使用转接板，也可以达到模块的方便使用的效果，转接板上已经设计好了模组的外围电路，引出了和模块定义一样的管脚，方便控制和操作，如下图所示：

