Embedded Programmer

Jorge Aparicio's blog on Electronics, Robotics and more...

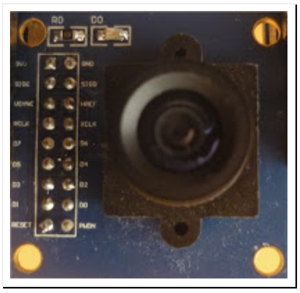
Monday, July 23, 2012

Hacking the OV7670 camera module (SCCB cheat sheet inside)

An in-depth look of the OV7670 camera module

The OV7670 is a low cost image sensor + DSP that can operate at a maximum of 30 fps and 640 x 480 ("VGA") resolutions, equivalent to 0.3 Megapixels. The captured image can be pre-processed by the DSP before sending it out. This preprocessing can be configured via the Serial Camera Control Bus (SCCB). You can see the full datasheet here.

There are many camera modules, that come with standard 0.1" spaced headers, in eBay with prices under $10. I'll be using the one shown below, it comes WITHOUT a FIFO buffer.



HARDWARE

The camera module comes with a 9x2 header, the pin diagram is shown below:

VDD	GND
SDIOC	SDIOD
VSYNC	HREF
PCLK	XCLK
D7	D6
D5	D4
D3	D2
D1	D0
RESET	PWDN

Now, I'll cover the meaning of these pins.

Pin	Type	Description
VDD**	Supply	Power supply
GND	Supply	Ground level
SDIOC	Input	SCCB clock
SDIOD	Input/Output	SCCB data
VSYNC	Output	Vertical synchronization
HREF	Output	Horizontal synchronization
PCLK	Output	Pixel clock
XCLK	Input	System clock
D0-D7	Output	Video parallel output
RESET	Input	Reset (Active low)
PWDN	Input	Power down (Active high)

****A note about supply voltage and I/O voltage.**

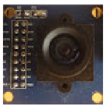
As stated in the datasheet:

Github Repositories

Check my open source projects here!

Featured Posts

Top 10 posts of this week



Hacking the OV7670 camera module (SCCB cheat sheet inside)

- VDDA can range from 2.45V to 3.00V.
- VDDC can range from 1.62V to 1.98V.
- VDDIO can range from 1.7V to 3.00V.

You can (hopefully) see here (sorry, it's buried among other files) the schematic of the model I'm using in this post. As you can see U1 and U2 are LDO regulators, one is a 2.8V regulator for VDDA and VDDIO and the other is a 1.8V regulator for VDDC. The actual regulator that gets soldered on the module seems to vary between modules.

In conclusion, for the same model I'm using:

- You can *safely* supply 3.3V (3.0V - 3.6V) to the OV7670 VDD. (I used this configuration)
- You can *safely* use a maximum of 3.0V for the I/O pins. However the module *I/O pins will work at 2.8V*.
- A 5V supply for the OV7670 VDD *might* work (try at your own risk), it depends on the maximum input voltage of the LDO regulators *your module* has.
- You can use 3.3V on the I/O pins, the internal I/O protection diodes will clamp the I/O voltage to 2.8V. *However*, this may degrade the OV7670 faster and/or cause more power loss. (I used this configuration)

STRUCTURE OF AN IMAGE

Before going into the signaling, it's necessary to understand how video and images are representend in digital format.

A video is a succession of ***frames***, a frame is a still image taken at an instant of time. A frame is compromised of ***lines***, and a line is compromised of ***pixels***. A pixel is the smallest part of a digital image, and it looks like a colored dot.

	P0	P1	P2	P3	P4
L0					
L1					
L2					
L3					
L4					

A 5x5 image

For example, the image above has 5 lines, and each line has 5 pixels. This means the image has a resolution of 5x5 pixels. This image is monochrome, there are also color image. This color can be encoded in various formats, in the next section we'll cover the most relevant formats for the OV7670.

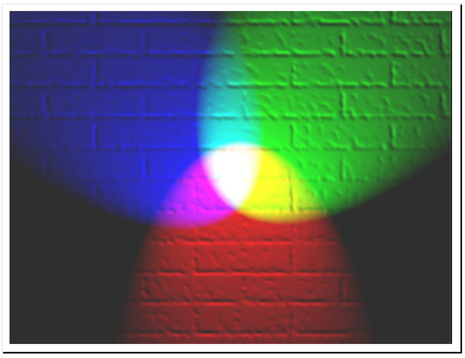
PIXEL FORMATS

Monochrome

In monochromes images, each pixel is stored as 8 bits, representing gray scale levels from 0 to 255. Where 0 is black, 255 is white and the intermediate values are grays.

RGB

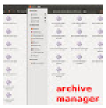
Is a fact that any color can be decomposed in red, green and blue light at different intensities. This approach is known as the RGB color model. Using this model, each pixel must be stored as three intensities of these red, green and blue lights.



RGB color model. Image from wikipedia.

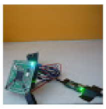
The most common format is RGB888, in this format each pixel is stored in 24 bits, the red, green and blue channels are stored in 8 bits each. This means that the intensity of each light can go from 0 to 255, where 0 is the absence of light, and 255 is the maximum intensity.

The formats used by the OV7670 are the RGB565, RGB555 and RGB444. The difference with the

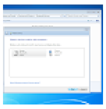


STM32F4DISCOVERY Development with GCC in Eclipse

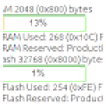
Ubuntu: Hacking the HC-06 bluetooth module



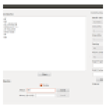
Demo: STM32F4 + OV7670 + qSerialTerm



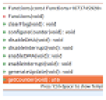
Windows: Communicating with a HC-06 bluetooth module



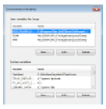
Bare metal development for ARM Cortex M microcontrolllers



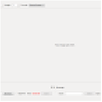
qSerialTerm: A Qt-based serial port terminal emulator



Open Source: Template Peripheral Library for the STM32 microcontrollers



Windows: Environment Variables for MinGW



qSerialTerm: Qt based Serial Port Image Viewer

Blog Archive

- ► 2013 (8)
- ▼ 2012 (27)
  - ► October (5)
  - ► September (2)
  - ► August (7)
  - ▼ July (5)
    - Hacking the OV7670 camera module (SCCB cheat sheet...
    - Windows: Communicating with a HC-06 bluetooth modu...
    - Windows: Installing Qt Creator
    - Ubuntu Tip 03: Quickly switch through applications...
    - Open Source: Template Peripheral Library for the S...
  - ► June (8)

Flag Counter

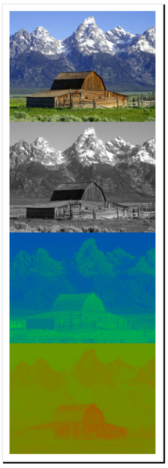
US	80782	DE	33367
RU	28917	IN	27949
PL	19101	GB	18222
FR	18127	BR	16308
IT	13531	KR	13298

Newest: AP You: RU
Today: 29
Month: 1876
Total: 502693
Supercounters.com

RGB888 format, is the number of bits assigned to each channel. For example, in the RGB565 format, the red channel is stored as 5 bits, the green channel as 6 bits and the blue channel as 5 bits. These formats take less memory when stored but in exchange sacrifice the number of colors available.

YCbCr

YCbCr is a format in which a RGB color can be encoded. The Y or luminance component is the amount of white light of a color, and the Cb and Cr are the chroma components, which respectly encode the blue and red levels relative to the luminance component.



Decomposition of an image into its Y, Cb and Cr components. Image from wikipedia.

As you can see the Y channel encodes the gray scale levels of the image. Therefore, the easiest way to get a monochrome image from the OV7670 is to extract the Y channel of the YCbCr format.

As the RGB format, the YCbCr also stores each channel as 8 bits (from 0 to 255) and we can convert from YCbCr to RGB using the following expression.

$$\begin{aligned} R &= Y && && + 1.402\cdot && (C_R - 128) \\ G &= Y - 0.34414\cdot && (C_B - 128) && - 0.71414\cdot && (C_R - 128) \\ B &= Y + 1.772\cdot && (C_B - 128) && && \end{aligned}$$

The OV7670 uses the YCbCr422 format, this format is stored as follows:

	Byte 0	Byte 1	Byte 2	Byte 3
Word 0	Cb0	Y0	Cr0	Y1
Word 1	Cb2	Y2	Cr2	Y3
Word 2	Cb4	Y4	Cr4	Y5

Data stored as words (4 bytes)

Or equivalently, the data arrives in the following order:

N	Byte
1st	Cb0
2nd	Y0
3rd	Cr0
4th	Y1
5th	Cb2
6th	Y2
7th	Cr2
8th	Y3
...	...

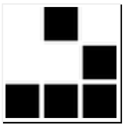
And the actual pixels are the following:

Pixel 0	Y0 Cb0 Cr0
Pixel 1	Y1 Cb0 Cr0
Pixel 2	Y2 Cb2 Cr2
Pixel 3	Y3 Cb2 Cr2
Pixel 4	Y4 Cb4 Cr4
Pixel 5	Y5 Cb4 Cr4

Notice each pixel is 3 byte long (e.g. Y0, Cb0 and Cr0), as in the RGB format. But, in the YCbCr422 format, the Cb and Cr channels are shared between two consecutive pixels (e.g. pixels 0 and 1 share Cb0 and Cr0). Therefore two pixels are "compressed" into 4 bytes or 32 bits, this means that in average each pixel is stored as 2 bytes or 16 bits. From the example above, 3 words

Visitor Map

About Me



Jorge Aparicio

G+ Follow 0

Mechatronics Engineering Undergraduate at National University of Engineering. DIY Electronics Hacker.

View my complete profile

(12 bytes) store 6 pixels.

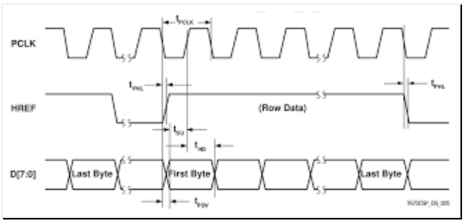
The extra advantage of YCbCr is that the Y channel is the grayscale image, whereas in RGB you'll need to average the 3 channels to get the grayscale image.

SIGNALING

The OV7670 sends the data in a parallel synchronous format. First of all, to get any data out of the OV7670, is necessary to ***supply a clock signal on the XCLK pin***. According to the datasheet, this clock must have a frequency between 10 and 48 MHz. However, I have successfully used a 8 MHz clock with some configuration via the SCCB.

If you are using a microcontroller that has clock output, you can use that to clock the OV7670, these can generally output their inner system clock prescaled by some factor. If your microcontroller doesn't have clock output capability, but you're using an external crystal, then you can connect the OSC_OUT pin to the OV7670.

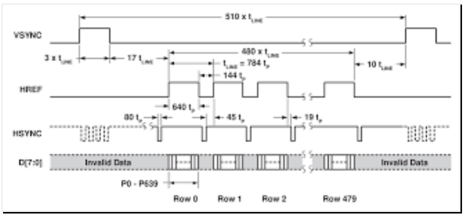
After a clock signal has been applied to the XCLK pin, the OV7670 will start driving its VSYNC, HREF and D0-D7 pins. Let's take a look at these signals.



Horizontal Synchronization

First thing to notice, the D0-D7 must be sampled at the rising edge of the PCLK signal. Number two, D0-D7 must be sampled only when HREF is high. Also, the rising edge of HREF signals the start of a line, and the falling edge of HREF signals the end of the line.

All these bytes sampled when HREF was high, correspond to the pixels in one line. Note that ***one byte is not a pixel***, it depends on the format chosen. By default, the format is YCbCr422, this means that in average two bytes correspond to a pixel.



VGA timing

The image above shows the signals for a "VGA" (640 x 480) frame. During HSYNC high state, we must capture 640 pixels, equivalent to a line. The 480 lines, equivalent to a frame, are captured during the low state of VSYNC. This means that the falling edge of VSYNC signals the start of a frame, and its rising edge signals the end of a frame.

That covers all the process of obtaining one frame, the remaining question is how fast are frames sent. By default, the PCLK will have the same frequency of XCLK, however prescalers and PPLs can be configured using the SCCB, to produce a PCLK of different frequency.

A PCLK of 24 MHz will produce 30 fps, a PCLK of 12 MHz will produce 15 fps and so on. All this is independent of the format of the image (VGA, CIF, QCIF, etc).

SCCB (Serial Camera Control Bus)

What makes the OV7670 so versatile is its inner DSP, that can pre-process the image before its sent. This DSP can be accessed via a SCCB interface. This SCCB protocol is very similar to the I2C protocol. You can see the SCCB specification here.

I couldn't get my STM32 microcontroller's I2C module to work with the OV7670's SCCB interface, so I implemented a bit bang version of the SCCB specification. This implementation is my peripheral library libstm32pp.

After making sure the SCCB is working, we can tweak the OV7670.

Changing the FPS

To change the frames per second (fps), we need to change the frequency of PCLK. And for that we need to modify the following registers via the SCCB.

Register	Address	Default	Description
CLKRC	0x11	0x80	***Bit[6]:*** ***0: Apply prescaler on input clock*** ***1: Use external clock directly*** ***Bit[0-5]: Clock prescaler***

			F(internal clock) = F(input clock) / (Bit[0-5] + 1) ***Range [0 0000] to [1 1111]***
DBLV	0x6B	0x0A	***Bit[7-6]: PLL control*** ***00: Bypass PLL*** ***01: Input clock x4*** ***10: Input clock x6*** ***11: Input clock x8*** Bit[4]: Regulator control 0: Enable internal regulator 1: Bypass internal regulator

Now that you know the involved registers, the process is straightforward. For example, say we have a 8 MHz input clock and we want a 24 MHz PCLK. The only possible configuration is prescaler by 2, and PLL x6.

- CLKRC Bit[6] must be 0, to enable prescaler.
- CLKRC Bit[0-5] must be 1, to enable prescaler by 2.
- DBLV Bit[7-6] must be 10, to enable PLL x6

Pseudocode:

```
unsigned char tmp;

/* Configuration for 30 FPS */

// CLKRC register: Prescaler = 2
tmp = readOV7670(0x11);
writeOV7670(0x11, (tmp & 0b10000000) | 0b00000001);

// DBLV register: PLL = 6
tmp = readOV7670(0x6B);
writeOV7670(0x6B, (tmp & 0b00111111) | 0b10000000);
```

Changing the frame format/resolution

The OV7670 can use various frame formats:

- VGA (640 x 480)
- QVGA (320 x 240)
- CIF (352 x 240)
- QCIF (176 x 144)
- Manual scaling

By default, the OV7670 uses the VGA format, if you want to do image processing on a microcontroller with the OV7670 output, this may be way too much data, and you might want the QCIF format instead. To change the format we need to modify the following registers.

Register	Address	Default	Description
COM3	0x0C	0x00	Bit[6]: 0: Nothing 1: Swap the data MSB and LSB. Bit[5]: On powedown 0: Tri-state the output clock 1: Do not tri-state the output clock Bit[4]: On powerdown 0: Tri-state the output data 1: Do not tri-state the output data ***Bit[3]: 0: Disable scaling*** ***1: Enable scaling*** Bit[2]: 0: Disable downsampling, cropping, windowing 1: Enable downsampling, cropping, windowing
COM7	0x12	0x00	Bit[7]: 0: Nothing 1: Reset all the registers to default values ***Bit[5]: 0: Nothing*** ***1: Use CIF format*** ***Bit[4]: 0: Nothing*** ***1: Use QVGA format*** ***Bit[3]: 0: Nothing*** ***1: Use QCIF format*** Bit[1]: 0: Disable color bar 1: Enable color bar Bit[2, 0]: 00: YUV 01: RGB 10: Bayer raw 11: Processed bayer raw

Example, say we want to use the QCIF format, we'll need to enable the scaling, and select the QCIF format.

- COM3 Bit[3] must be 1, to enable scaling
- COM7 Bit[3] must be 1, to use the QCIF format

Pseudocode:

```
unsigned char tmp;

/* Configuration for QCIF format */

// COM3 register: Enable format scaling
tmp = readOV7670(0x0C);
writeOV7670(0x0C, tmp | 0b00001000);

// COM7 register: Select QCIF format
tmp = readOV7670(0x12);
writeOV7670(0x12, (tmp & 0b11000111) | 0b00001000);
```

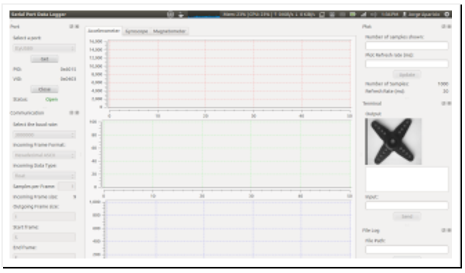
I'll add more possible configurations as I explore other features.

MY RESULTS

I have tested my OV7670 module, with a STM32F4 microcontroller. This microcontroller comes with a Digital CaMera Interface (DCMI) and a Direct Memory Access (DMA) controller, these two can capture the frames without the intervention of the processor.

I used an XCLK of 8 MHz, but configured the OV7670 to output a PCLK of 24 MHz, this means I was capturing 30 fps. I used the QCIF format, however I was receiving 174 x 144 pixels instead of 176 x 144. Color format was the default YCbCr422. One of every six frames was sent to a PC using a UART communication at 3 Mbps.

On the PC side, I received the frames using a modded version of qSerialTerm, only the Y channel (gray scale version) of the incoming frames was used. The result is shown in the following image.



A ~~ninja star~~ servo horn captured by the OV7670 camera module.

Now you can use qSerialTerm to visualize images streamed through the Serial Port. Check this post for more info.

TROUBLESHOOTING

SCCB:

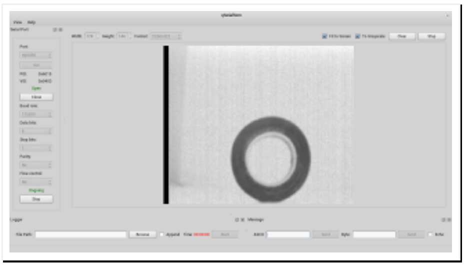
- Make sure the SCCB is working properly, the OV7670 will answer with an ACK, after it has been address properly.
- The 7 bit SCCB/I2C address is 0x21, this translates to 0x42 for write address and 0x43 for read address.
- For debugging purposes, try reading some registers and check that they contain their default values. e.g. reading the 0x01 register should return 0x80.
- Always read a register first, modify the desired bits and then write it back to the OV7670.

Image sensor

- Check wiring, pin configuration and clock configuration.
- Start only grabbing a snapshot (only one frame), this is from VSYNC falling edge to VSYNC rising edge. Repeat this procedure multiple times, and make sure the number of bytes per snapshot, is constant.
- Cover the camera lens, and verify that the snapshot have the following information (in bytes): 128 0 128 0 128 0 128 0 ... i.e. every even byte should be 128 and every odd byte should be 0. This correspond to a pitch black image.
- If the two previous experiments fail, your uC might be too slow to grab the OV7670 stream, either increase its clock speed or get a faster uC. If you are using a DMA controller, then give it high priority, clock it as fast as possible and/or dedicate it to this task.
- If you are visualizing the grabbed snapshot in a PC, for starters only use the luminance (Y) channel, i.e. only use the even bytes of the snapshot. On the PC, assign R = G = B = Y for each pixel.
- At this point, all the electrical/software part should be working. The only remaining issue is the camera focus (distance from the camera lens to the image sensor), you will have to vary the camera focus by trial and error until you get a clear image.

Varying the camera focus

UPDATE: Check this post about 3 demos that involve the STM32F4, the OV7670 and qSerialTerm. Full code available.



qSerialTerm displaying a frame sent by the STM32F4 and was captured by the OV7670

Posted by Jorge Aparicio at 4:38 PM

Labels: Cross-platform, Electronics, Open Source, Ubuntu, Windows

256 comments:

Dew@Dawn August 3, 2012 at 4:22 AM

This comment has been removed by the author.

Reply

Replies

shahbaz alam October 11, 2015 at 8:24 PM

This comment has been removed by the author.

Reply

Dew@Dawn August 3, 2012 at 4:24 AM

hi.. I used Stm32f4 and Ov7670 without FIFO same as you. However I cannot get a clear image like you. the output of the image is 177 x 148 for QCIF.

1. Could you show me some configurations that you initialize the camera thru SCCB (i2c)?

2. I dont understand this part when you config the clock.

"have a 16 MHz input clock and we want a 24 MHz PCLK"
first you set the prescaler to 2 then you get 8 MHZ
sencond you multiply by 6 so.. it should be 8x6 = 48 isn' it?

thanks

Reply

Replies

Jorge Aparicio August 3, 2012 at 12:24 PM

Hi Dew,

I added a troubleshooting section, I hope you'll find it useful.

@Question 1: I added pseudocode in the SCCB section. That's the configuration I'm using.

@Question 2: Sorry, my mistake, that configuration is for a 8 MHz input clock. The post has been corrected.

Anonymous June 24, 2015 at 8:53 PM

cyberasdf

Reply

drainelectronics August 5, 2012 at 9:14 AM

great info, thank you... :)
Please if it,s possible can you send me your modded version of qSerialTerm???

Thanks in advance

drainelectronics@gmail.com

Reply

Replies

Jorge Aparicio August 5, 2012 at 4:12 PM

You're welcome.

Right now I'm working to integrate the data plotter and the image viewer features in the main repository of qSerialTerm. Everything should be ready by next week.

edward. beer71 December 3, 2015 at 5:14 PM

im hacker and offering Folowing Services contact me if you intersted in any thing i do fair deals try me first
skype : edward.beer71
yahoo IMI : edwardbeers420@yahoo.com
Hang out : edwardbeer7@gmail.com
****
==
======
i got Wu bug New versoion that i use to make western union transfer my prize is listed i will show you each and everything
about wu bug ,
Western Union Prize :
1000$ in 275$
1500$ in 325$
2000$ in 475$
2999$ in 550$
==
======
i have all tools that you need to smap,well if you dnt know what is spam then please vist at http://hacktoladen.blogspot.com
Spamming Tools :
1:Smtp : 3$ for ip (4$ for Domain)
2:Shell : 2$ per one
3:Cpanel : 3$ per one
4:Scam page 30$ for simple (60$ for undetectable)
5:RDP : 15$ any country
6:PHP mailer : 3$ per one
****
==
======
well if you know about carding i m offerring folowing tools with fair prize .
Carding Tools :
RDP: 15$ any country
HMA: 25$ unlimited 12 month
Vip72: 25$ unlimited 6 month
card validator : 50$ (for fixing un-valide card number and bin checker)
wu Java bypass Script ,(by pass any page with your giving commands)

****
==
======

Credit cards :
random : 25$ per one
fullz : 35$ per one (with Dob + SSN + MMN + Driving license +)

****
==
======

Virus/Rate
zeus : 250$ (with fud crypted jpeg,pdf or doc file)
key loger : 150$ (for email,pm and btc logs)
ninja Rat : 130$ (with fud crypted jpeg,pdf or doc file)
cidital : 150$ (with fud crypted jpeg,pdf or doc file)

****
==
======

private Scanner
smtp scanner : 350$ (linux bassed) (ssh/root required for run)
rdp scnnaer : 400$ (linux bassed) (ssh/root required for run)edwardbeers420@yahoo.com
Smtp+rdp multi scanner : 600$ (linux bassed) (ssh/root required for run)
cpanel scanner : 500$ (linux bassed) (ssh/root + 10 cpanel or shell required for run)
root scanner : 800$ (linux bassed) (ssh/root required for run)

****
==

======
Contact me:
skype : edward.beer71
yahoo IM : edwardbeers420@yahoo.com
Hang out : edwardbeer7@gmail.com

Reply

sikbbang August 20, 2012 at 9:55 PM

hi!

very helpful information

I'm also studing about ov7670

I set up the QCIF as like above information, but it dosen't work

Could you let me know how you can set registers for qcif?

Pleas mail me to cfsang0402@hotmail.com

Reply

Replies

Jorge Aparicio August 22, 2012 at 12:55 AM

Hello sikbbang,

I'm glad you've found this information useful.

I've used the configuration shown in the pseudocodes (SCCB section). I'll recommend checking the Troubleshooting section of this post.

In particular, you should check that you are receiving correct data using the default configuration. Then, test that your SCCB interface is working, you should be able to read the OV7670 registers and see their default values.

And only if all of the above works, then you should try other configurations.

If you still have specific problems, post them here.

Reply

Dew@Dawn August 22, 2012 at 2:18 AM

sikbang,

for me I finally made it work! could you show me your output? so I can figure out your problem.

Dew

Reply

sikbbang August 27, 2012 at 1:01 AM

Dew & jorge

Thanks for your attention!

i make a QICF

it need setting vstart vstop hstop hstart!

Reply

Replies

jhon wallter November 7, 2015 at 8:18 PM

Contact me:
skype : alfaz999
yahoo IM : Sonic.programmer1@yahoo.com
Hang out : hacktorich@gmail.com
Web : http://hacktoladen.blogspot.co.uk/
****
==
======

Spamming Tools :

1:Smtp : 3$ for ip (4$ for Domain)
2:Shell : 2$ per one
3:Cpanel : 3$ per one
4:Scam page 30$ for simple (60$ for undetectable)
5:RDP : 15$ any country
6:PHP mailer : 3$ per one

****
===
======
Carding Tools :

RDP: 15$ any country
HMA: 25$ unlimited 12 month
Vip72: 25$ unlimited 6 month
card validator : 50$ (for fixing un-valide card number and bin checker)
wu Java bypass Script ,(by pass any page with your giving commands)

****
===
======

Virus/Rate
zeus : 250$ (with fud crypted jpeg,pdf or doc file)
key loger : 150$ (for email,pm and btc logs)
ninja Rat : 130$ (with fud crypted jpeg,pdf or doc file)
cidital : 150$ (with fud crypted jpeg,pdf or doc file)

****
===
======

Western Union Prize :
1000$ in 225$
1500$ in 250$
2000$ in 350$
2999$ in 450$

****
===
======
Contact me:
skype : alfaz999
yahoo IMI : Sonic.programmer1@yahoo.com
Hang out : hacktorich@gmail.com
Web : http://hacktoladen.blogspot.co.uk/

Reply

Brijesh September 6, 2012 at 12:21 PM

I am also trying to interface OV7670 to STM32f4.

Did you have to set any other registers, other than ones you mention in the blog post? I am having difficulty in capturing image, I am not seeing the 00 80 00 80 pattern.

Thanks

Reply

Replies

Jorge Aparicio September 6, 2012 at 12:43 PM

Hello Brijesh,

Is not necessary to set any register via the SCCB to get the data capture interface working.

Be sure to connect the clock output (MCO1/MCO2) of the F4 to the OV7670 XCLK pin.

Check the configuration of the MCO pin (I used 16 MHz output derived from the HSI) and the DCMI peripheral.

Be sure to check the Troubleshooting section of this post.

Good Luck :)

Anonymous June 24, 2015 at 8:54 PM

Spamming Tools :

1:Smtp : 3$ for ip (4$ for Domain)
2:Shell : 2$ per one
3:Cpanel : 3$ per one
4:Scam page 30$ for simple (60$ for undetectable)
5:RDP : 15$ any country
6:PHP mailer : 3$ per one
****
===
======

Carding Tools :

RDP: 15$ any country
HMA: 25$ unlimited 12 month
Vip72: 25$ unlimited 6 month

card validator : 50$ (for fixing un-valide card number and bin checker)
wu Java bypass Script ,(by pass any page with your giving commands)

****
===
======

Credit cards :
random : 25$ per one
fullz : 35$ per one (with Dob + SSN + MMN + Driving license +)

****
===
======

Virus/Rate
zeus : 250$ (with fud crypted jpeg,pdf or doc file)
key loger : 150$ (for email,pm and btc logs)
ninja Rat : 130$ (with fud crypted jpeg,pdf or doc file)
cidital : 150$ (with fud crypted jpeg,pdf or doc file)

****
===
======

private Scanner
smtp scanner : 350$ (linux bassed) (ssh/root required for run)
rdp scnnaer : 400$ (linux bassed) (ssh/root required for run)
Smtp+rdp multi scanner : 600$ (linux bassed) (ssh/root required for run)
cpanel scanner : 500$ (linux bassed) (ssh/root + 10 cpanel or shell required for run)
root scanner : 800$ (linux bassed) (ssh/root required for run)

****
===
======
Contact me:
skype : alfaz999
yahoo IMI : Sonic.programmer1@yahoo.com
Hang out : hacktorich@gmail.com

Reply

Brijesh September 8, 2012 at 1:24 PM

I was able to get a image, but only displayed Black & white by using just the Y value. I am not yet getting a good color picture.

Were you able to get a good color picture with default register settings and the conversion from YUV to RGB formula that you posted?

Reply

Jorge Aparicio September 12, 2012 at 5:55 PM

Hello Brijesh,

I did get a proper color image using the default register configuration. I was getting a distorted image when I used 8 MHz on XCLK, so I used 16 MHz instead.

When doing the YCbCr -> RGB conversion, be sure to use floats/double when carrying out the operations and then limit the obtained R, G, B values between 0 and 255.

You might want to use qSerialTerm to visualize the images if you are streaming the image via UART.

I'll post a F4 + OV7670 demo soon. Stay tuned.

Reply

Brijesh September 13, 2012 at 1:15 AM

"When doing the YCbCr -> RGB conversion, be sure to use floats/double..."

Dang not sure how I missed that one. I was using integer. Even with that change I did not get good color image. I had to use register settings I got from Linux driver to get a good image. Not sure why.

About qSerialTerm, it is a quite useful program. The only reason I have not used is that I would have to compile the program myself. That means I have to setup the whole QT environment. Untill now all my time was used up in debugging the camera.

Also what is protocol and data format to display a image in qSerialTerm?

Thanks for your input. I look forward to your demo code with F4 + OV7670

Reply

Replies

Jorge Aparicio September 15, 2012 at 12:33 AM

Brijesh,

I'm getting a "correct" color image using the above conversion equations, in the sense that the image doesn't get distorted. But the colors are no good, because instead of green I see magenta and so on.

I have tested the conversion equations against Octave's rgb2ycbcr function, and it works fine. Maybe the OV7670 follows another conversion standard?. I have also tried switching Cb <-> Cr without any noticeable change in the color output.

I have seen those OV7670 linux drivers, are you using any of the SCCB configurations available there?

qSerialTerm accepts uncompressed binary images in Monochrome, RGB888 or YCbCr422 formats. This means, you only need to stream the data collected from the DCMI module into the UART module.

I've already finished the demo and updated qSerialTerm. I'll post the results as soon as I find an adequate screencasting software.

Brijesh September 19, 2012 at 12:23 AM

Hello,

Yes for me also with default settings, I am getting wrong colors, instead of red, I got greyish-green.

But finally managed to get a good color image. I had to play around with the registers a lot and got some register settings from Chinese supplier.

https://lh5.googleusercontent.com/-u08R-fwq8E8/UFf3dU1gh1I/AAAAAAAAAGs/Yc9CXJDkLZ0/s639/rubic%2520cube.jpg

If you need the register settings that finally worked for me let me know.

Cheers

Christopher September 22, 2012 at 12:21 AM

Hi Brijesh

We currently have an OV7670 which correctly gives us test patterns, however gives us terribly saturated images with completely incorrect colours. Are you able to share those register settings, or some us advice about how to set this device up for normal looking images? Thanks very much,
-Chris

Brijesh September 23, 2012 at 10:27 AM

The register file is huge about 300 registers. Send me an email at brijesh at negtronics dot com

I have documented the project at
www.negtronics.com/vision-module

I have a forum thread at sparkfun
https://forum.sparkfun.com/viewtopic.php?f=14&t=32645

Chaitannya Supe March 9, 2014 at 10:37 AM

hello brijesh,
I am currently working on real time image processing using FPGA. For that i am using OV7670 camera module. I have got real time video but with wrong colors(mostly magneta color). Can you please mail me the correct color settings?

Reply

Anonymous October 6, 2012 at 8:06 AM

Hi guys,
if I want to use an external clock, do I have to connect the oscillator between XCLK and ground directly or do I need to add a capacitor in series (XCLK - oscillator - cap - GDN)? If yes, which value for the capacitor?

Thanks

Alex

Reply

Anonymous October 6, 2012 at 8:22 AM

Hi guys,
I have a ceramic resonator @ 16Mhz (http://it.rs-online.com/web/p/risonatori-ceramici/5266154/), but it has 3 pins instead of two. Searching on google I've seen the the the central pin must be connected to the ground, but I'm stuck with the others two.
Do you have some suggestions?

Reply

Replies

Jorge Aparicio October 6, 2012 at 9:39 AM

Hello Anon,

Short answer:

You can't use a ceramic resonator with the OV7670.

Long answer:

Ceramic resonator require an active oscillator circuit. These active oscillators are usually integrated inside microcontrollers, and you have access to it via two pins usually labelled OSC_IN and OSC_OUT.

The OV7670 does NOT have an internal active oscillator. You need to feed a clean periodic (usually square) signal to the XCLK pin.

Possible solution:

Ideally you should connect a clock out signal (e.g. the MCO pin of a STM32 microcontrollers) to the OV7670. But, the following MIGHT work:

+ Connect a crystal/resonator to the OSC pins of your microcontroller.
+ Connect the OSC_OUT of your microcontroller to the XCLK pin of the OV7670.
+ Obviously the microcontroller and the OV7670 already share the same ground level.

Rationale:

The integrated active oscillator of your microcontroller will output a periodic signal on its OSC_OUT pin as part of the feedback loop.

Possible problem:

Connecting the XCLK with the OSC_OUT will add more capacitance load and it might distort the periodic signal.

Worst case scenario:

The OSC_OUT pin won't have enough current drive and no clock will be produced. This implies that neither your microcontroller nor the OV7670 will work.

Good Luck

P.S. I'm curious, what microcontroller are you interfacing the OV7670 to?

Anonymous October 6, 2012 at 12:04 PM

Hi Jorge,
I'm using BeagleBone with QNX. But BeagleBone seems not provide an external clock. If I use an external oscillator (crystal) do I have connect one pin to the XCLK and one to the ground without capacitors?

Thank in advance for your kind reply.

Alessandro

Anonymous October 6, 2012 at 12:30 PM

Hi Jorge,
I've just found a pin with an external clock on BeagleBone... :) (my fault!) In this case I suppose I have to connect the pin directly to the XCLK (sharing the same GND). Right?

Thank you!

Alessandro

Jorge Aparicio October 6, 2012 at 3:02 PM

Hi Alessandro,

Yes, yhat should work. You'll need to configure the actual output frequency to something around 8-24 MHz for proper operation.

The STM32 F2/F4 microcontrollers have a DCMI peripheral that simplifies the data reception from the camera. I don't know if the BeagleBone have something similar.

Good Luck!

Reply

Haris October 16, 2012 at 10:02 AM

i have interfaced ov7690 with stm32f4 and PIC. i get data in both rgb565 and yuv422 and reconstuct the image . but the image is not so good. u can look my post at http://www.edaboard.com/thread268832.html

plz take a look.

Reply

Replies

Jorge Aparicio October 18, 2012 at 9:25 AM

Hello Haris,

I can't see your attached files in the website you provided. (I don't have an account in that discussion board)

What do you mean with "the image is not so good"? Are you getting a clear image with the wrong colors or are you getting random pixels? Does the grayscale (only the Y channel) image looks clear?

AFAIK in YUV422 mode, getting the right colors with the OV7670 requires some heavy calibration, as stated by brijesh in his/her comment; but the grayscale version should look clear like the image I posted.

You can send me your files to jorge.aparicio.r at gmail dot com.

Reply

claudiu November 4, 2012 at 10:27 AM

Hi Jorge, nice page you have here, seems you have been studying quite a fair bit on OV7670. I am also working on something similar but using AVR uC instead (Atmega16/32) with OV7670 FIFO version and works pretty well.
I am using YUV mode and using the YUV422 equations give me a pretty good looking image, a bit soft but not bad (http://thinksmallthings.wordpress.com/2012/11/03/ov7670-yuv-demystified/).
FYI, i am using Processing to capture and display the images, allows me to write my code and play with the output pretty easy.
Again, great tutorial on OV7670!

Reply

mih November 5, 2012 at 6:44 AM

Hi, that post was very useful and helped me understand how this sensor works, but i can't make it to work.

I'am unable to read proper image data, SCCB works fine for me and I can read registers, but for example when i'm reading frames with my lens covered i get some random numbers - where as you wrote there should be 128 0 pattern. I'm using camera mode with FIFO buffer embedded. Could you send me your code for reading frames ? Or will you be able to help me with this ? I'm using STM32F103VC

Thanks ;)

Reply

Replies

Jorge Aparicio November 7, 2012 at 10:11 AM

Hello mih,

I haven't used the OV7670 FIFO version and I used the STM32F4 which has a Digital Camera Interface (DCMI) that the STM32F1 doesn't have. That's the reason my code won't really help.

Make sure you are supplying a correct clock (12-24 MHz) to the OV7670 XCLK pin. The SCCB works WITHOUT the XCLK, so a working SCCB doesn't guarantee a working parallel interface (D0-D7).

For maximum throughput you should use the FSMC peripheral, but you should "debug" your hardware first by using GPIO (D0-D7) and polling (HREF, PCLK).

Good luck.

Anonymous January 5, 2013 at 4:50 PM

Hello

I've just tried to get SCCB working on my OV7670 (like yours: no FIFO) and it looks like it doesn't work at all until I connect clock source to XCLK pin. After successful test implementation (got 0x80 from 0x01 register etc) i tried to disconnect XCLK signal and it simply stopped working. So it seems to me that XCLK is necessary for SCCB operation.

Reply

zainka November 14, 2012 at 9:04 AM

Hi

I noticed that you use 3.3V, or atleast that you use this voltage in your blog. The data sheet for OV7670 says 3.0V max for DOVDD (1.7 to 3.0).

In the module I use, (ov7970 without fifo) the DVDD pad (core voltage) is simply filtered to ground via capacitor thus ov7670 will generate 1.8 from DOVDD. This requires the DOVDD to range from 2.45V to 3V.

The analog power supply, AVDD, must range from 2.45 to 3.0V. and in my device is AVDD and DOVDD connected together with a simple noise filter.

Reply

Replies

zainka November 14, 2012 at 9:05 AM

I wrote ov7970 instead of 0v7670. To bad i did not see that and to bad there is no edit opportunities...

Jorge Aparicio November 14, 2012 at 10:49 AM

Hi zainka,

The (blue) model I used in this post has internal LDO regulators. You must have the (black?) model that is wired directly. I have also used the latter with a 3.3V supply and a series Schottky diode.

I've added a notice in the Hardware section to avoid confusion.

Thanks.

Reply

Anonymous December 12, 2012 at 9:49 AM

Dear Jorge,

Do I need the XCLK to get the SCCB working? I do not get my camera working :(

And the SCCB cheat sheet you talk about where I can find it?

Thanx!
Dennis

Reply

Replies

Jorge Aparicio January 16, 2013 at 2:57 PM

Hello Dennis,

Sorry for the awfully late reply. You probably already sorted out the problem, but here it goes:

You need to drive the XCLK pin with a clock signal to make the OV7670 parallel interface (D0-D7) work to get any image data out of it. Also a user reported here that driving the XCLK pin is also necessary to get the SCCB interface working.

The SCCB "cheat sheet" only has two cheats: (1) up/down scaling the fps and (2) up/down scaling the image resolution.

Reply

Mermaldad December 14, 2012 at 11:16 PM

Wow, Jorge, it looks like your post has struck a chord with a number of users! If I may throw another question into the fray, I am thinking about using a microprocessor (thinking mbed, but not sure yet) to sample from multiple OV7670s. I don't need to capture every frame from every camera, just be able to grab a picture from camera 1, then from 2, etc. at whatever rate the processor can manage. Any thoughts on how one might do this with some sort of bus, or would this require a separate processor for each camera?

Reply

Replies

Jorge Aparicio January 16, 2013 at 5:03 PM

Hello Mermaldad,
(by the way, nice name)

Sorry for the awfully late reply. You probably already took a decision. But here is my view anyways:

tl;dr I'll go with one uC per camera, it increases throughput.

Using many cameras per uC has the following disadvantages:

(1) Hooking multiple OV7670s to the same uC will require a digital circuit to act as a multiplexer and will increase the capacitive load of the logic lines (XCLK, D0-D7), which can degrade or totally corrupt the data.
(2) You are dealing with high speed parallel interfaces, and they don't get pretty far. This will limit how far you can place the cameras.
(3) The OV7670 outputs raw data, that means each frame of 640x480 color pixels equals a minimum of 600 KiB (YUV422 format). This implies you can't keep one frame in the uC RAM memory (usually less than 200 KiB), so you'll need to stream the frame to a bigger memory as soon as possible.

I don't know what application you have in mind, but I have thought about these general schemes, all involve using one uC per camera:

(1) Each uC grabs a frame, compresses it using JPEG and stores it to its own SD card.
(2) Similar to (1), but the compressed frame is sent via ethernet to a central storage server (some PC).
(3) Similar to (2), but the central server does some useful processing and alternatively issues commands to the uCs.

Using one uC per camera has the following advantages:

(1) You can take one frame from all the cameras almost at same time.
(2) Each uC will capture frames as fast as possible.
(3) You can place the camera as far as you wish and as far as the network media allows it.

Good luck in your project!

Reply

Anonymous January 3, 2013 at 2:35 PM

Hi

I've just configured the C code for OV7670 but with error. So, may anyone give some help.

Reply

Anonymous January 5, 2013 at 11:35 PM

Excellent project!

Can u make it open source, that would be great for many beginners!

Reply

Replies

Anonymous January 7, 2013 at 6:45 AM

Thanks

I have only the following parts which are OV7670 and Arduino uno board. I wrote my code and I have to cheange the clock function code to either read or write in the registers of the camera"OV7670". But, I can't get any response from the camera.

My code has written as follows:

```
"Start"
Serial.println("Memory written");
}

void loop()
{
//
// Read the first page in EEPROM memory, a byte at a time
//
Serial.println("eeprom_read_byte, return: ");

Wire.beginTransmission(DEVADDR);
Wire.write(0x0B);
Wire.endTransmission();
Wire.requestFrom(int(DEVADDR), 1);
if (Wire.available()) {
rdata = Wire.read();
}
//return rdata;
Serial.println(rdata, HEX);

digitalWrite(13, HIGH);
delay(100);
digitalWrite(13, LOW);
delay(100);


}

void eeprom_write_byte(byte deviceaddress, int eeaddress, byte data)
{
// Three lsb of Device address byte are bits 8-10 of eeaddress
byte devaddr = deviceaddress | ((eeaddress >> 8) & 0x07);
byte addr = eeaddress;
```

```
Wire.beginTransmission(devaddr);
Wire.write(int(addr));
Wire.write(int(data));
Wire.endTransmission();
delay(10);
}
```

"End"

Thank you again and I'm looking for your help.

Regards
Saad Alasmari
salasmari@kacst.edu.sa

Jorge Aparicio January 16, 2013 at 2:37 PM

Hello Saad,

A user reported that is necessary to supply a clock signal to the XCLK pin to get the SCCB working. Make sure you are correctly driving the XCLK pin.

Good Luck.

Anonymous February 4, 2013 at 5:45 AM

You too Jorge,

Thanks a lot.

Reply

Pedro Matos January 7, 2013 at 6:08 AM

Olá Jorge!

Sou um iniciante em linux, e tou querendo usar a mesma camera que usaste com uma NGW100.

Se pudesses dar-me umas dicas básicas com algum pseudo codigo seria formidavel.

Muito obrigado,
Pedro Matos

Reply

Replies

Jorge Aparicio January 16, 2013 at 2:27 PM

Hello Pedro,

You got me there. I don't have experience with Linux when it comes to accessing peripherals, for this post I used a microcontroller without OS. AFAIK, Linux is more restrictive on specific memory/register access.

My wild guess, is that you'll need to create a C/C++ project and link/add the necessary Linux drivers (either a parallel interface or the image sensor interface + network interfaces). After that, pretty much everything is specific to your NGW100 board.

There seems to be a lot of documentation for that board, you'll have more luck there.

Reply

robert bloink January 15, 2013 at 11:03 PM

Can anyone program the SCCB interface..to change the zoom feature of a OV2720?

Reply

Alessandro Portosa January 16, 2013 at 10:05 AM

First of all, thanks a lot for your article and for all the help that I found here. :)

I get an OV7670 camera but I'm confused about this sentence into your article:

"**: On some models the VDD is directly connected to the OV7670 sensor AVDD and DOVDD, in this case VDD should be below 3.0V. With this model I have used a 3.3V supply with a series Schottky diode without problems.

**: There are other models (like the blue one I used in this post) which contain LDO regulators between the VDD and the OV7670 sensor AVDD and DOVDD. I'm not sure how much input voltage these regulators can stand, but I have used 3.3V directly without problems."

How can I know if my module has this LDO regulators?

Here you can find a front/back photos of my camera module:
FRONT: http://i49.tinypic.com/2h4w3dk.jpg
BACK: http://i48.tinypic.com/24ax6at.jpg

Could you please have a look and tell me which kind of supply I need to provide to this camera? And maybe also where are LDO regulators placed (if there are...)?

Thanks again.
Alessandro.

Reply

Replies

Jorge Aparicio January 16, 2013 at 2:02 PM

Hello Alessandro,

I updated the I/O and supply voltage part of this post. I hope it's clearer now.

Alessandro Portosa January 16, 2013 at 4:21 PM

Yes, it's clearer! Thanks a lot for your quick reply

Reply

Anonymous January 17, 2013 at 2:35 PM

Hi Jorge, I am trying to store the data in an array that captured from OV7670. I'd like to implement in Keil 4. Can you send me this project? I will surely compensate your effort (which an amount we both agree :) adnanoktar41@hotmail.com

pls contact me whoever can do it.
---cheers.

Reply

Anonymous January 19, 2013 at 2:10 AM

hi!!
I would like to know whether we can interface ov7670 with pic16f or pic18f.....plz reply as soon as possible

Reply

Replies

Jorge Aparicio January 19, 2013 at 10:17 AM

Hello anon,

I don't think you can interface this version of the OV7670 module with a 8 bit uC, because the data is streamed at very high speeds and your uC RAM is way too small.

You might have more luck with the FIFO version (I don't have experience with that version). But then again, a 8 bit uC has very little RAM so you would be able to process a very scaled down (in resolution) image. If you are using the uC as a buffer to a higher speed interface, then the lack of DMA and processing power will make the uC a bottleneck.

Anonymous July 17, 2013 at 12:59 PM

Not true I got it working with an arduino uno I get about 1fps at qvga. Also this is the version without the fifo see http://forum.arduino.cc/index.php?topic=159557.0 and check out my github project for working code https://github.com/ComputerNerd/arduino-camera-tft
The trick is to set the divider value to something slow. Also the 10mhz lower clock limit is false. It will go much slower just set the register 0x11 to whatever is slow enough for you.

Reply

Giacomo Fiorini January 21, 2013 at 11:46 AM

Hi Jorge, i'm working with the OV7670 module on an altera FPGA Board and i'm encountering some problems wich i cannot solve.
The YCbCr422 encoding is the default format used by the camera module right?
If so what are the exact data position/order that the camera is using on it's D0-7 pin?
When you explain how the data is stored for the YCbCr format you make a block scheme.
What those blocks represents? Are they Bytes? If so one block would be the information present on D0-7 pins at a certain pclk edge?

Reply

Replies

Jorge Aparicio January 22, 2013 at 12:04 PM

Hello Giacomo,

Yes, the YCbCr422 encoding is the default format used by the camera.

On every PCLK transition you should read D0-D7 (one byte), these bytes you read should be a Cb byte, then a Y byte, next a Cr byte, a Y byte again, and the cycle repeats with Cb and so on. (Assuming you are correctly synchronized by HREF and VSYNC)

I updated my diagrams, I believe it's clearer now.

Good luck on your project.

Alessandro Portosa January 23, 2013 at 8:38 AM

Thanks a lot, now it's clearer and for some reason there is no reference about this into the datasheet.

Also, you wrote that:

"A PCLK of 24 MHz will produce 30 fps, a PCLK of 12 MHz will produce 15 fps and so on. All this is independent of the format of the image (VGA, CIF, QCIF, etc)."

Yes, it's indipendent of the image format, but should be highly dipendent of the kind of pixel format... Am I right?

I mean, in my purpose I just need to sample the Y component and I can sample it only on the 2nd (and 4th and so on...) rising edge of a 24Mhz PCLK: so pratically I'm working with a 12Mhz pixel clock!

In order to have 30fps and assuming that I'm using the default parameters, I need to feed the XCLK of the camera with a 48Mhz clock... Is my idea right?

Jorge Aparicio January 23, 2013 at 6:38 PM

Alessandro,

Take a look at the Figure 6 of the datasheet. Say XCLK is 24 MHz, and using the default SCCB configuration PCLK would be 24 Mhz. From the figure, t_plck is the period of PCLK and t_p = 2 * t_pclk.

t_p is two clock cycles of PCLK, i.e. the necessary time to grab a pixel (on the formats of the OV7670, on average each pixel is equivalent to 2 bytes).

For VGA, you have 480 lines of 640 pixels. Each line lasts t_l = 784 * t_p, with HREF active only for 640 * t_p. Next, each frame lasts t_f = 510 * t_l, but you only get lines during 480 * t_l.

If you do the math: t_f = 510 * 784 * 2 / 24MHz = ~33.3 ms, equivalent to 30 fps.

I've only worked with YCbCr422, but for other formats like RGB565, the t_p = 2 * t_pclk should hold. For other image sizes t_f and t_l also hold, is the active part of HREF what gets shorter and the the numbers of transmitted lines also gets reduced.

In conclusion, XCLK should be 24 MHz and you should read the D0-D7 pins only on every even (2, 4, ...) rising edge. You'll end up with 30 fps, but each pixel will be 1 byte (Y component) long.

Reply

Anonymous January 25, 2013 at 1:19 AM

can we intrface ov7670 with pic16 and wnted to know about the rgb output format of ov7670 ? how will the camera capture the image ?will we require a video sync separator?

please do reply asap.thnku.

Reply

Replies

Jorge Aparicio January 25, 2013 at 10:13 AM

Hello Anon,

I have answered a similar question above in other comment. Please check that comment as well.

1) Can you interface the OV7670 with a PIC16?

I don't think is doable with this version of the OV7670 camera. You might want to check the FIFO version.

2) I haven't worked with the RGB format, but you will be limited to 5 bit Red, 6 bit Green and 5 Blue or similar - i.e. 2 bytes per pixel.

3) If you want to know how the camera captures the image, read the datasheet.

4) See (1)

Anonymous January 28, 2013 at 1:48 AM

thnx jorge looking forward to ur help..Culd u then sugest us a microcontroller with ov7670, will pic32 work?

Jorge Aparicio January 29, 2013 at 11:00 AM

I can't recommend the PIC32, because I haven't work with it.

I'll recommend a uC that has a dedicated Digital Camera Interface (like the DCMI peripheral found in the STM32F4)

If that's not possible then a Parallel Interface *might* be adapted to capture the data from the OV7670.

Finally, you might want to check my last post (the link is the end of this post) with 3 demos about using the STM32F4, the OV7670 and qSerialTerm, the source code is available.

Anonymous February 1, 2013 at 4:24 AM

thanks, how can we downsize the image in OV7670 , if yes what will be the minimum possible size of array?

Reply

Diogo Salgueiro January 31, 2013 at 5:13 PM

Hey Jorge,
I'm working with this Camera and with the STM32F4 and i manage to get values from the Camera but i'm having trouble making the serialTerm work. It complains about me not having the serialport.h library. I did like it says in the wiki, i got serialport first i installed the library it says there and i installed Qtcreator for ubuntu and then i just opened your serialTerm project and did "play" and i get that error. Can you help me?
Thank you and great tutorial! Thumbs up!

Reply

Replies

Jorge Aparicio January 31, 2013 at 6:55 PM

Hello Diego,

I just checked that the last commit to the qtserialport library introduced a compilation error, this is the reason why qSerialTerm can't find the library, it's because qtserialport didn't got build and didn't got installed.

As a workaround you can install the second to last commit, which works, using the following commands:

```
sudo apt-get install libudev-dev
git clone git://gitorious.org/qtplayground/qtserialport.git
cd qtserialport
git checkout 0e56cac
qmake
make
# Check if there are any errors!
sudo make install
```

Please tell me if that works.

I'll notify the maintainers of the qtserialport library about the compilation error.

Thanks

Reply

Aleksandar Jovanovic February 2, 2013 at 5:41 PM

This comment has been removed by a blog administrator.

Reply

austin ciambriello February 7, 2013 at 4:14 PM

Hi Jorge,

I'm using the same exact set up you have here for a school project and I can successfully communicate with the camera via I2C routines. The next step I'm trying to do is configure the DCMI to capture the image and store in an internal buffer to verify that the DCMI is collecting the data. My problem is I never get anything in the DCMI->DR register although the Vsync and Hsync are being triggered in the SR. I think I'm messing up in the configuration of the DCMI and DMA. Think you could help me out?

Reply

Replies

Jorge Aparicio February 7, 2013 at 4:43 PM

Hello Austin,

Please check my other post whichs covers 3 demos of the STM32F4 and the OV7670:
http://embeddedprogrammer.blogspot.com/2013/01/demo-stm32f4-ov7670-qserialterm.html

You'll find the source code there, you might not be able to directly use the code (unless you use my libraries), but the code is very readable.

If you read my code you'll see which configuration I've used for the DMA and the DCMI and understand the logic flow.

If you still have questions, ask them in the comment section of the other post.

austin ciambriello February 7, 2013 at 4:52 PM

I've read over the examples you provided there but I'm developing in .C not .cpp. I understand the translation over to C but maybe you could explain a few things. You mention that the Discovery board cannot be used because all of the DCMI pins cannot be accessed but I'm not certain that is the case. Also I assumed that the DCMI would also handle the complete capture of a frame such that it automatically determines when vsync and hsync occurs and clears the flags after capturing the data and transferring it through the dma ect. Is this true or do I need to create a routine that handles the actual capture when thos flags are set?

Jorge Aparicio February 7, 2013 at 5:12 PM

Regarding the STM32F4DISCOVERY, the DCMI pins are available in the breakout headers, but those pins are also connected to other components inside the DISCOVERY, i.e. all of them are not "free" (see the table 5 of the DISCOVERY manual). That may or not be a problem.

If you configure correctly both the DCMI and the DMA, the only thing that you need to do is: (1) start a capture on the DCMI (set the CAPTURE bit high) and (2) wait for the capture complete interrupt. After that, all the data will be in the memory region you selected for the DMA.

austin ciambriello February 7, 2013 at 7:44 PM

Even so as long as the Vsync Hsync and Pclk are not interfere with other components on the board, some sort of data should be in the buffer. Even if the data is wrong that would be better than where I am. So I can assume since we are both using the same Vsync Hsync and Pclk pins that I have something configured incorrectly with either my DCMI and or my DMA since nothing is being stored in my buffer.

Jorge Aparicio February 7, 2013 at 8:16 PM

I won't rule out a hardware problem as those signaling pins are connected to components on the DISCOVERY board.

You can do some debugging with the following flags (DCMI_RIS register):
+ Line Received (check if DCMI signaling works)
+ Frame Synchronization (check if DCMI signaling works)
+ Buffer Overrun (this flag indicates a bad configured DMA)

Good Luck!

Reply

austin ciambriello February 9, 2013 at 6:34 PM

I am not receiving anything in the RIS register at all. the only registers that change are the config registers during init. the Sr register sets the vsync and hsync after dcmi_cmd() and only the capture bit gets set when the dcmi_capturecmd() is called. So basically the stm is not recognizing the hsync and vsync signals coming in it seems... Does this sound like it could be a hardware problem. that those components are effecting the way PA4 PA6 and PB7 are supposed to be working?

Once again thanks for your time and help. I really appreciate the assistance especially being that it is fully voluntary. :)

Reply

Replies

Jorge Aparicio February 9, 2013 at 7:10 PM

Austin,

From the top of my head, you could do one or both of these tests to check if the components on the board are distorting the DCMI/OV7670 signals:

+ Leave the XCLK running, and poll the HSYNC, VSYNC and PCLK pins using the GPIO module, not the DCMI module. This would rule out the possibility of a badly configured DCMI.

+ Leave the XCLK running, hook a logic analyzer or an oscilloscope to the HSYNC, VSYNC and PCLK pins, and by looking at the signals you'll know whether the other components on the board distort those signals or not.

I'll try to help as long as my experience and time allow me. I've received a lot of help in my life, and this feels like a good way to retribute that. :)

Reply

austin ciambriello February 11, 2013 at 9:13 PM

I have solved the problem. I had a typo in my GPIO configuration.... instead of defining PCLK on PA6 i defined it as PA5. So simple I don't know how i missed it. But now i'm able to capture the images fine. Thanks for your help!

Reply

Replies

Jorge Aparicio February 11, 2013 at 9:14 PM

Austin, good to hear that it's working now.

Your welcome.

Jencie July 15, 2013 at 9:27 AM

Hello Austin!

I'd really appreciate it, if you could send me details abo...ut the discovery board and the ov7670. Pin connection and dcmi configuration.

Thank you for your help!

Reply

Anonymous February 13, 2013 at 5:23 AM

Hello Jorge,
I am fairly new to i2c and would like a quick start using the 0v7670 cam and the arduino uno. Would you happen to have code I can run to test the functionality of my cam. The simplest configuration of the cam would be nice. Perhaps being able to save to an sd card the image taken. Thanks a lot.

Reply

Replies

Jorge Aparicio February 13, 2013 at 6:49 AM

Hello Anon,

1) Sorry, I don't have arduino code, I don't even use 8 bit uC.

2) It's not I2C it's SCCB, there are minor differences. And you don't need SCCB to get frames from the camera, you need a parallel interface. The SCCB is for configuration, but the camera works well with the default configuration.

3) There is a link at the bottom of this post, that contains source code of 3 demos using the STM32F4 and the OV7670. That code won't work with your arduino but it will give you an idea of how to work with the camera.

4) If you have a question after YOU ACTUALLY TRY SOMETHING, then feel free to ask. I won't answer to more "do the work for me" requests.

Reply

Curtis Newton February 16, 2013 at 12:18 PM

hello, the ov7676.rar file seem to be missing, can you put it online ?
thx

Reply

Replies

Jorge Aparicio February 17, 2013 at 11:13 AM

Curtis,

Sorry I don't have the file with me and I don't host the link I provided in this post.

However you might find that file in the web, the sellers of the OV7670 usually put it along the description of the module in their websites.

Good Luck

Reply

Anonymous February 17, 2013 at 10:39 AM

Hey ... this goes out to everyone who have trouble with the SCCB:
- The module needs the XCLK even if you do not use the picture data.
- It seems like the interface is not very fast, so scale down your SIO_C
- To get some data out:
- - send a start
- - you have to "write"(0x42) the target register first (sub-address)
- - stop first com
- - then start a new com
- - now you can "read"(0x43) the register
- - stop second com
e.g.(start, 0x42, 0x0b, stop, start, 0x43, should get 0x73, stop)

maybe the owner will add this schematic in his main text

Reply

Replies

Jorge Aparicio February 17, 2013 at 11:20 AM

Thanks for your comment Anon, I'm sure other people will find it useful.

I also added a link to the code of my SCCB bit bang implementation in the SCCB section of this post. Hopefully it's readable.

Cheers

Reply

Den Tran April 10, 2013 at 7:11 AM

This comment has been removed by a blog administrator.

Reply

Anonymous April 24, 2013 at 3:14 AM

Hello all
Thank your for this useful page.
I'm using the OV7670 camera with fpga . but i'm having problem
the Vsync signal is always at '1' so i don't get anything in the display ? any ideas ? Need help
Thank you so much

Reply

Replies

Anonymous May 15, 2013 at 5:05 PM

Hi. We are working with the OV7670 and also the OV7725 sensor. Found out the hard way that the OV7725 MUST have an external clock fed into the sensor. Our OV7670 is the version that features an onboard 24 Mhz clock source so it was not required. Can you confirm if your version of the module needs an external clock ? Is your I2C (SCCB) bus working ? We found that even the I2C bus NACKed without the external clock source. We fixed the issue by applying 25 Mhz into pin 8 for our OV7725 module (wayengineer.com, China). Soon we will work with the "raw" OV7670 (no FIFO) after we validate the version with FIFO in our design.

Reply

Robert Murphy April 26, 2013 at 10:12 AM

I have an OV7670 on order and have a few questions. I'm using an Arduino Uno which has a 3.3V out, but not, I guess, at sufficient rating for the camera. Also, the camera has 3V IO pins while the Arduino uses 5V. Where do I get these lower voltages from? I have seen level shifters, is this what I need? Sorry about the simlest of questions, from the look of posts here this may be the least of hurdles in front of me.

Reply

Replies

Anonymous May 15, 2013 at 5:13 PM

Hi Robert. Let me try to help..

a) Did you receive your OV7670 module ? Please post the details on your sensor as they most definitely vary from vendor to vendor. Is your module the "raw" sensor (without FIFO) or with FIFO and oscillator ?

b) Likely there is enough current @ 3.3 volts from your Arduino to supply the sensor.

However, before you go there, share the details on the sensor so we can confirm if you have onboard regulators to properly lower the (multiple) voltages required by the actual OV7670 sensor itself. Next, the voltage swing for the I/O should be lowered to 3.3 volts or perhaps even lower, again depending on the version of the module you actually purchase. Our module is being interfaced with the AL422B fifo which is operating @ 3.3 volts so we are fine with I/O @ 3.3 volts. The pins we have seen on the module, except for the I2C Clock and I2C Data, are all unidirectional. So, it is possible to purchase level shifters that will lower the voltage to suit the module from 5 volts to say 2.8 volts, etc. More on this later after we know about your module :)

Reply

Anonymous May 15, 2013 at 4:59 PM

Hi Jorge. Many thanks for this great webpage. It has helped a lot to better understand the OV7670 (we have one with FIFO & 24 Mhz oscillator) we have sourced from one of the offshore vendors.

May we confirm if your extracted data stream from the sensor is 100% as follows (if covered with dust cap) = BLACK:

128 0 128 0 128 0 128 0... (if possible could you paste a dump from your sensor)

The reason is that we are noting the values of 128 = 80 (hex) and 00 but not always in the above format.

For example (with the dust cap ON),

```
80 00 80 80 00 00 00 00 00 00 00 00 80 80 00 80
00 80 80 80 00 80 80 80 00 00 00 00 00 00 00 00
00 00 80 80 80 00 80 80 80 00 00 00 00 00 00 00
00 00 80 80 80 80 00 80 80 80 00 00 00 00 00 00
00 00 00 00 80 80 80 00 80 80 80 80 00 00 00 00
00 00 80 00 80 80 80 00 80 80 00 00 02 00 00 00
```

If we remove the dust cap, then the values do change:

```
94 90 00 bb e4 ea 00 96 d7 e8 00 96 95 94 94 00
00 9f 97 8d 94 89 91 89 d8 00 81 c1 b0 00 4b b6
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8b 97 00 7e a0 9e 00 59 8a 94 00 8b 93 8b 99 00
90 00 95 98 97 a2 26 7b a0 99 00 60 8b a1 00 91
8b 92 8b 96 00 d3 8b 00 00 53 95 9c 00 ad 90 96
97 93 98 8d 98 91 9e 00 4f 8f 95 00 3f 8a 9b 85
```

If we cover the module, the values will turn into 80(hex) and 00 but again in varying order. Perhaps we are not extracting the values properly from the AL422B FIFO ?

We will map to a local LCD screen within a few days but wanted to confirm the BLACK color output. The results that are posted are without priming the sensor. Thanks again !!

Kumar

Reply

Replies

Shubham Garg June 9, 2013 at 8:51 AM

i even have the same problem
no order of getting 128 0 128 0

```
128,128,128,0,0,0,128,128,0,128,128,
128,128,128,0,128,128,128,128,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,128,128,0,128,0,0,
0,128,0,0,0,0,0,0,128,128,0,0,0,0,0,0,128,0,128,128,0,0,0,0,0,0,128,0,128,128,0,
0,0,128,128,128,128,0,128,128,0,0,0,128,0,128,128,128,128,128,0,0,0,0,0,0,0,0,0,
0,128,128,0,128,0,128,128,128,128,128,0,128,128,0,0,0,128,0,0,0,0,0,0,0,0,128,12
8,0,128,0,0,0,0,0,0,128,0,128,128,0,0,0,128,0,128,0,128,128,0,128,0,0,0,128,128,
0,128,128,128,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128,0,128,128,128,128,128,0,128,
128,0,0,0,128,0,128,128,128,0,0,128,0,0,0,0,0,0,0,0,0,0,128,0,0,0,128,0,0,0,128,
0,128,128,0,0,0,0,0,128,128,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,0,0,0,0,0,128,0,
128,128,0,0,0,0,0,0,128,128,0,128,0,128,0,0,0,128,128,0,128,128,128,0,0,128,0,0,
0,0,0,0,0,0,128,128,0,128,0,0,0,128,0,128,128,0,0,0,0,0,0,128,0,128,0,128,128,12
8,128,0,0,128,0,128,128,128,128,128,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,0,
0,128,128,0,128,128,128,128,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,128,128,128,0,0,
128,0,128,128,0,0,0,128,0,128,128,0,0,128,128,0,0,0,0,0,0,0,0,0,0,128,0,0,0,128,
0,0,0,128,0,128,128,0,0,0,0,128,0,0,0,128,128,0,128,127,128,128,128,0,128,128,12
8,128,128,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,128,0,0,0,128,128,0,128,127,128,12
8,0,0,128,0,128,128,0,128,0,0,0,128,0,0,0,128,0,128,128,0,0,0,0,0,0,128,0,0,0,0,
0,0,0,0,0,0,0,128,128,0,128,0,128,128,0,128,0,0,0,128,0,0,0,0,0,128,0,128,0,0,0,
0,0,0,128,0,128,128,0,0,0,1,0,0,128,128,0,128,0,128,0,0,0,128,128,0,128,0,128,12
8,128,128,128,0,128,128,0,0,0,128,0,128,0,0,0,0,0,128,128,0,0,0,0,0,0,0,0,0,128,
0,128,128,0,0,0,128,0,128,128,0,0,0,0,0,0,128,128,0,128,0,128,128,128,0,0,128,0,
128,128,128,128,128,0,0,0,0,0,0,0,0,0,0,0,128,128,0,128,0,128
```

Anonymous July 17, 2013 at 1:03 PM

No that is correct the ov7670 outputs yuv422 by default. If you are sending it to an lcd you may want to set it to rgb565 mode but be aware that rgb565 mode is very bad quality. If your microcontroller is fast enough try converting from yuv422 to rgb565 in real-time that results in a much better image.

Reply

Alex contreras trujillo May 26, 2013 at 6:53 PM

Hi to everyone I have buyed this camera http://dx.com/p/ov7670-al422b-fifo-cmos-camera-module-w-dupont-cable-for-smart-car-153842, im trying to connect it to the STM32F4discovery board using the bit bang version of SCCB, but I cant get the camera to work, I dont know if the delay time (500) is the problem, the STM32 is running at 144MHZ Im trying to see the waveform in the osciloscope but nothing happens, what it could be? thnks in advance

Reply

Shubham Garg May 29, 2013 at 2:52 PM

This comment has been removed by the author.

Reply

Shubham Garg June 6, 2013 at 1:09 PM

This comment has been removed by the author.

Reply

Alex cont June 18, 2013 at 10:22 PM

Hi, well the camera is working, but the colors are wrong, it shows purple instead of white, what registers should I change in order to get RGB 565? in somewhere I did read that purple color is white without green component however I can see the green bar in the test pattern I will apreciate any help thank you

Reply

prem June 19, 2013 at 2:03 AM

Hi Jorge,

we are getting distorted picture, we found is that some of the pixels are getting missed within each frame of total 640 frames (for vga mode 640x480) and hence we are neither able to get full picture size nor good colors of an image".
We are using OV5640 camera on I.MX233 (freescale) and OS is 2.6.35-8-ARCH+.

We want to implement DMA to capture all pixels thrown out by DVP port of camera. How we can proceed with this. We are using GPIO's port to grab them.

Prem

Reply

Anonymous June 22, 2013 at 11:34 PM

i want to acces the ov7670 with arduino,,anybody help me?
thank you

Reply

Blaise July 2, 2013 at 11:42 AM

Excellent blog! Thank you!

Reply

Anonymous July 21, 2013 at 2:27 PM

I'm having an inordinate amount of difficulty getting my camera module to respond to my bit-banged SCCB, which is based directly off the code helpfully provided by the author of this blog post. Could someone please point me in the right direction?

Here's my logic analyzer output:

http://i.imgur.com/eq2p6vv.png

You'll see a start condition, setup to write to 42, sub address 0, an ACK (by the master...the slave has never once ACKed or NACKed). Then a stop condition, start condition, and phase two, which is setup a read at 43, then the clock line is pulsed 8 times, at which point the slave doesn't assert the data line. This process then repeats for subaddress 1, 2, etc.

I'm also supplying a 12 MHz XCLK signal to the module.

Reply

Replies

Anonymous July 22, 2013 at 12:29 AM

It would figure that after I finally make a post about this I figure it out, even though it's been like two weeks of work to get to this point!

The problems I was having boiled down to this:

1. XCLK was necessary for me to communicate with SCCB (I'm using a 12 MHz clock, output on one of the GPIO pins on my AT32UC3A3)

2. Though I had enabled the clock out, I was attempting to communicate with SCCB *before* actually enabling it! Stupid.

3. I had a few places where I was inadvertently driving the Don't Care bits without setting my SIOD line to input mode, and I think that it could've been preventing the slave from asserting it when it wanted.

4. Originally, I was confused about when to do start and stop conditions. To read a register you need to do the following:

a. Start condition, 8 bits for the bus address for write, 1 D/C bit, 8 bits for the register memory address, 1 D/C bit, Stop condition. Start condition, 8 bits for the bus address for read, then pulse the clock to read 8 bits and then 1 NACK/ACK bit (or maybe it's a D/C). Stop condition.

5. Another big issue was that the datasheet just says the write address is 42 and the read address is 43. What they neglect to make explicit is that it's 0x42 and 0x43, not 42d and 43d! Sheesh. Maybe it was clear to everyone else, but not to me.

A logic analyzer and (to a lesser extent) an oscilloscope helped immeasurably in solving this. (The output you see in that screenshot is from the Saleae Logic analyzer.)

Reply

Jencie July 22, 2013 at 12:16 AM

Hi!

Could you please check my problem on the ST forum?
https://my.st.com/public/STe2ecommunities/mcu/Lists/cortex_mx_stm32/Flat.aspx?RootFolder=%2fpublic%2fSTe2ecommunities%2fmcu%2fLists%2fcortex_mx_stm32%2fSTM32F4Discovery%20%2b%20DCMI%20%2b%20OV7660%20%28OV7670%29&FolderCTID=0x01200200770978C69A1141439FE559EB459D7580009C4E14902C3CDE46A77F0FFD06506F5B&TopicsView=https%3A%2F%2Fmy%2Est%2Ecom%2Fpublic%2FSTe2ecommunities%2Fmcu%2FLists%2Fcortex_mx_stm32%2FAllItems%2Easpx&currentviews=9
I'd be very grateful!
Thanks!

Jeno

Reply

Replies

mark diaa92 December 14, 2015 at 3:27 PM

E r R 0 r e 404.™ (͡°□ ͡°)
im hacker and offering Folowing Services contact me if you intersted
in any thing i do fair deals try me first
skype : suzi.maan1
yahoo IMI : suzimaan@yahoo.com
Hang out : suzimaan@gmail.com
WEB:https://hackersfourzerofour.blogspot.com
icq :675452902
----
i got Wu bug New versoion that i use to make western union transfer
my prize is listed i will show you each and everythingabout wu bug ,
Western Union Prize :
1000$ in 225$
1500$ in 250$
2000$ in 350$
2999$ in 450$
----
well if you know about carding i m offerring folowing tools with fair prize .
Carding Tools :
RDP: 15$ any country
HMA: 25$ unlimited 12 month
Vip72: 25$ unlimited 6 month
card validator : 50$ (for fixing un-valide card number and bin checker)
wu Java bypass Script ,(by pass any page with your giving commands)

Fresh fullz random world wide is here now.
Credit cards :
random : 20$ per one
fullz : 30$ per one (with Dob + SSN + MMN + Driving license +)

Virus/Rate
relesed 2015 zeus : 200$ (with fud crypted jpeg,pdf or doc file)
relesedkey loger : 150$ (for email,pm and btc logs)
ninja Rat : 100$ (with fud crypted jpeg,pdf or doc file)
cidital : 150$ (with fud crypted jpeg,pdf or doc file)

Contact me:
skype : suzi.maan1
yahoo IMI : suzimaan@yahoo.com
Hang out : suzimaan@gmail.com
icq :675452902

Reply

Tirthraj July 29, 2013 at 12:44 PM

hi,

I need information on Pixel Clock.

How much min pixel clock can give good image. If my controller is slow speed and I don't want more than 1FPS then can I lower the PCLK less than 1Mhz?
I did not found any min PCLK requirement anywhere.

Reply

Replies

Anonymous July 29, 2013 at 6:28 PM

I believe the PCLK frequency is determined by the XCLK and the scaling factor in the SCALING_PCLK_DIV register. The maximum scaling value is 16, so XCLK / 16 = PCLK_min. If you want any slower than that you'll have to lower XCLK, which has a minimum of 10 MHz. So 10 MHz/16 MHz = 625 kHz = PCLK_min.

I haven't done that part yet, so I may be wrong.

Anonymous July 29, 2013 at 6:30 PM

That should be 10 MHz/16 (not 16 MHz)

Reply

Tirthraj July 30, 2013 at 11:48 AM

Thanks for information.

Reply

Shubham Garg August 2, 2013 at 12:39 PM

@Tirthraj
use 16Mhz CRYSTAL and change the value of 0x11 register to 0xBF
16MHz/(63+1)= .25MHz

decreasing the value of crystal in atmega32 with not work since the machine cycle gets slower

Reply

Anonymous August 5, 2013 at 10:04 PM

If you scale PCLK down with PCLK_SCALING_DIV, does that reduce the horizontal resolution? When I set it to be as slow as it can go, I get 480 HREFs interrupts (480 height) but 24 times less than the number of PCLK interrupts I need to cover every pixel (assuming 3 bytes per pixel, so it's actually less). So my resolution is something crazy, like 80 x 480.

Reply

ingfut August 13, 2013 at 9:28 PM

First i would like to say thanks for such a informative blog, its just what i was looking since a long time :)
Here are some of the point i would like to ask
1:- Can i connect this with ARM based board(beagle board or raspi)?
2:- If above is true then can u suggest how to do that?
3:- can we access on-board DSP of camera ?

thanks in advance

Reply

Steinberg Lin August 15, 2013 at 3:48 AM

Here are the 3.5 QVGA 4.3 WQVGA 5.7 QVGA 7 WVGA TFT LCD modules, and always trying to stay in LCD technology in the forefront of our current technology products, including COG, COF, TAB, SMT. Believe that through our cooperation, the displayed a better world.Character LCD Module . click here

Reply

DVE August 20, 2013 at 4:52 AM

Dear Jorge,

Thank you for your great forum here. I used it to get multiple things clear with my Deal Extreme ov7670 with fifo. I use the stm32F103 with a 2.4" TFT in rgb565 mode. I am past all trouble with sccb and I get good qqvga images on my TFT (I scaled to qqvga because the amount of data is less then).

My "problem" is the colors look reasonable with daylight but when it gets darker its way off. I know its a cheap camera but maybe you found a way to deal with this?

Cheers,
Dennis

Reply

Matz August 21, 2013 at 11:26 AM

Hi Jorge, thank you for the really informative post on the OV7670. I was wondering if you have any experience with similar modules, but with a higher resolution? I'd like to experiment with something >3MP.

Reply

Replies

Anonymous October 31, 2015 at 8:18 PM

hello guys to day i am talking aboute the market there is many scamers
he gona make deal with you but he not give you the things so care about
you self and if you want make any deal with any person i can get you
it is real persone are not because i am old in the this bussiness...
any thing you want to buy like..
Virus/Rate
zeus :(with fud crypted jpeg,pdf or doc file)
key loger :(for email,pm and btc logs)
ninja Rat :(with fud crypted jpeg,pdf or doc file)
cidital :(with fud crypted jpeg,pdf or doc file)
Carding Tools :
RDP:any country
1.HMA:unlimited 12 month 2.Vip72:unlimited 6 month
3.card validator:(for fixing un-valide card number and bin checker)
4.wu Java bypass Script ,(by pass any page with your giving commands)
Spamming Tools :
1:Smtp :for ip (for Domain) 2:Shell: 3:Cpanel: 4:Scam page undetectable)
5:RDP:any country 6:PHP mailer:
contact:skype : suzi.maan1

Reply

Leonardo August 24, 2013 at 3:08 AM

Hi Jorge,
I get this bad image from camera http://img208.imageshack.us/img208/5576/i94r.png
I've tried to change registers with internet found values but nothing change, the sensor may be faulty?

Reply

Shervin August 29, 2013 at 10:19 PM

Thanks for this very useful tutorial! I was thinking about buying this camera and then writing a tutorial on it, but you have already made a good tutorial, so now I just need to buy the camera :-)

Reply

Arndt Jenssen September 2, 2013 at 3:18 PM

Great post!! Helped me with some issues I had with the OV7670. I managed to get it running von an AVR uC (and more). For all the AVR freaks out there: https://github.com/arndtjenssen/ov7670

Reply

Gabriele Garavini September 11, 2013 at 5:49 AM

hello I wanted to know if you think it is possible to use a different module than the OV7670. I wanted to have a much higher resolution type 12/14 Mpx but with 1 frame per minute.
is such a thing possible, any suggestions or tutorials?

Reply

Anonymous September 18, 2013 at 3:40 AM

Hey, i'm using the OV2640, which is quite similar.
SCCB is working with a 20MHz resonator XCLK.
But i have nothing on the other outputs (ie VSYNC, HREF, D0-D7...)

Is the data coming out once it is powered ?
Or do i need to "start" the module so it can send the data ?

Reply

Replies

Anonymous November 3, 2013 at 9:56 PM

If it's like the OV7670, make sure PWDN and RESET are in proper states, supply 3V3 and GND, and pulse XCLK at a valid frequency. At that point you should start to see the OV7670 driving HREF, VSYNC, PCLK, and D0-D7. I'm not sure if it's necessary, but I think you're also supposed to pull-up SIOC and SIOD.

Reply

Shubham Garg September 28, 2013 at 1:20 AM

plz help !!!!

tell me input to XCLK should be sinusoidal or square wave.

Reply

Replies

Anonymous November 3, 2013 at 9:54 PM

I provided a 24 MHz square wave clock to my XCLK and it worked well. I imagine it's a digital clock, not analog.

-Dan

Reply

Tincho Andrade October 9, 2013 at 4:32 PM

Hi jorge, I am a last year student of Technical High School from Argentina, and I need to make a final project.

I have a OV7670 camera without FIFO. In order to test it, before try to get an image, I wish to read valid pixel data and print them trough the serial port.

I have connected output pins d0 to d7, href, vsync, to digital input pines of ARDUINO MEGA 2560 board, and the camera is powered by 3.3 V from ARDUINO board.

For clock signals (XCLK) I use a 24.8 MHz external oscillator which is powered with 5 V from ARDUINO board.

I have written a program to read data when vsync changes from low to high but I have get only zeroes.

I assume that camera is OK because when I read d0-d7 pines directly I see that bytes change their values under illumination changes

I need some help to control this camera, if you know any way to view images (don’t care the format, I’m not changing registers, I use the default configuration) only with arduino mega board and a PC it will be great, and useful to my project.

Thank you for help me,

Martín

This is the program:

```
#define href 21
#define vsync 20
#define d0 24
#define d1 25
#define d2 26
#define d3 27
#define d4 28
#define d5 29
#define d6 30
#define d7 31


void setup(){

pinMode(href, INPUT);
pinMode(vsync, INPUT);

Serial.begin(9600);

pinMode(d0, INPUT);
pinMode(d1, INPUT);
pinMode(d2, INPUT);
pinMode(d3, INPUT);
pinMode(d4, INPUT);
pinMode(d5, INPUT);
pinMode(d6, INPUT);
pinMode(d7, INPUT);


attachInterrupt (3, empiezabarrer, RISING);
}


void empiezabarrer(){

// read 8 bits (1 byte) and print to serial

int Bit1;
int Bit2;
int Bit3;
int Bit4;
int Bit5;
int Bit6;
int Bit7;
int Bit8;

Bit1= digitalRead(d0);
Bit2= digitalRead(d1);
Bit3= digitalRead(d2);
Bit4= digitalRead(d3);
Bit5= digitalRead(d4);
Bit6= digitalRead(d5);
Bit7= digitalRead(d6);
Bit8= digitalRead(d7);

Serial.print(Bit1);
Serial.print(Bit2);
Serial.print(Bit3);
Serial.print(Bit4);
Serial.print(Bit5);
Serial.print(Bit6);
Serial.print(Bit7);
Serial.print(Bit8);
Serial.println(" ");

//detachInterrupt(3);

}

void loop(){


// the flowing lines captures d0 to d7 directly (all the time)

/*
int Bit1;
int Bit2;
int Bit3;
int Bit4;
int Bit5;
int Bit6;
int Bit7;
int Bit8;

Bit1= digitalRead(d0);
Bit2= digitalRead(d1);
Bit3= digitalRead(d2);
Bit4= digitalRead(d3);
Bit5= digitalRead(d4);
Bit6= digitalRead(d5);
Bit7= digitalRead(d6);
Bit8= digitalRead(d7);

Serial.print(Bit1);
Serial.print(Bit2);
Serial.print(Bit3);
Serial.print(Bit4);
Serial.print(Bit5);
Serial.print(Bit6);
Serial.print(Bit7);
Serial.print(Bit8);
Serial.println(" ");
*/
}
```

Reply

Anonymous October 30, 2013 at 12:18 AM

Finally i made the OV7670+FIFO module work with STM32F103, it took me months and a lot of tests (trying different versions of OV7670 without FIFO i got just random data...), by now i can say that the registers are a real pain, and there are LOT of things that can get messed up just by modifying one bit of any register, and that there even are reserved registers that must be set to an adequate value, i don't know why but even the output sequence (first and second byte) gets swapped by a register that is unknown. Here I publish the register values that worked to RGB565, QVGA (XCLK signal is given by a 25MHZ osc).
-Diego

Reply

Anonymous October 30, 2013 at 12:19 AM

```
{0x3a, 0x04},
{0x40, 0x10},
{0x12, 0x14},
{0x32, 0x80},
{0x17, 0x17},
{0x18, 0x05},
{0x19, 0x02},
{0x1a, 0x7b}, /* 0x7a */
{0x03, 0x0a}, /* 0x0a */
{0x0c, 0x0c},
{0x3e, 0x00},
{0x70, 0x00},
{0x71, 0x01},
{0x72, 0x11},
{0x73, 0x09},
{0xa2, 0x02},
{0x11, 0x00},
{0x7a, 0x20},
{0x7b, 0x1c},
{0x7c, 0x28},
{0x7d, 0x3c},
{0x7e, 0x55},
{0x7f, 0x68},
{0x80, 0x76},
{0x81, 0x80},
{0x82, 0x88},
{0x83, 0x8f},
{0x84, 0x96},
{0x85, 0xa3},
{0x86, 0xaf},
{0x87, 0xc4},
{0x88, 0xd7},
{0x89, 0xe8},
{0x13, 0xe0},
{0x00, 0x00}, /* AGC */
{0x10, 0x00},
{0x0d, 0x00},
{0x14, 0x10}, /* 0x38 limit the max gain */
{0xa5, 0x05},
{0xab, 0x07},
{0x24, 0x75},
{0x25, 0x63},
{0x26, 0xA5},
{0x9f, 0x78},
{0xa0, 0x68},
{0xa1, 0x03}, /* 0x0b */
{0xa6, 0xdf}, /* 0xd8 */
{0xa7, 0xdf}, /* 0xd8 */
{0xa8, 0xf0},
{0xa9, 0x90},
{0xaa, 0x94},
{0x13, 0xe5},
{0x0e, 0x61},
{0x0f, 0x4b},
{0x16, 0x02},
{0x1e, 0x37}, /* 0x07 */
{0x21, 0x02},
{0x22, 0x91},
{0x29, 0x07},
{0x33, 0x0b},
{0x35, 0x0b},
{0x37, 0x1d},
{0x38, 0x71},
{0x39, 0x2a},
{0x3c, 0x78},
{0x4d, 0x40},
{0x4e, 0x20},
{0x69, 0x5d},
{0x6b, 0x40}, /* PLL */
{0x74, 0x19},
{0x8d, 0x4f},
{0x8e, 0x00},
{0x8f, 0x00},
{0x90, 0x00},
{0x91, 0x00},
{0x92, 0x00}, /* 0x19 0x66 */
{0x96, 0x00},
{0x9a, 0x80},
{0xb0, 0x84},
{0xb1, 0x0c},
{0xb2, 0x0e},
{0xb3, 0x82},
{0xb8, 0x0a},
{0x43, 0x14},
{0x44, 0xf0},
{0x45, 0x34},
{0x46, 0x58},
{0x47, 0x28},
{0x48, 0x3a},
{0x59, 0x88},
{0x5a, 0x88},
{0x5b, 0x44},
{0x5c, 0x67},
{0x5d, 0x49},
{0x5e, 0x0e},
{0x64, 0x04},
{0x65, 0x20},
{0x66, 0x05},
{0x94, 0x04},
{0x95, 0x08},
{0x6c, 0x0a},
{0x6d, 0x55},
{0x6e, 0x11},
{0x6f, 0x9f}, /* 0x9e for advance AWB */
{0x6a, 0x40},
/* {0x01, 0x60}, */
/* {0x02, 0x60}, */
{0x13, 0xe7},
{0x15, 0x00},
{0x4f, 0x80},
{0x50, 0x80},
```

Reply

Anonymous October 30, 2013 at 12:19 AM

```
{0x51, 0x00},
{0x52, 0x22},
{0x53, 0x5e},
{0x54, 0x80},
{0x55, 0x00}, /* ÁÁ¶È */
{0x56, 0x60}, /* ¶Ô±È¶È */
{0x57, 0x90}, /* 0x40 change according to Jim's request */
{0x58, 0x9e},
{0x41, 0x08},
{0x3f, 0x05}, /* ±ßÔµÔöÇ¿µ÷Õû */
{0x75, 0x05},
{0x76, 0xe1},
{0x4c, 0x0F}, /* ÔëÉùÒÖÖÆÇ¿¶È */
{0x77, 0x0a},
{0x3d, 0xc2}, /* 0xc0 */
{0x4b, 0x09},
{0xc9, 0xc8},
{0x41, 0x38},

{0x34, 0x11},
{0x3b, 0x02}, /* 0x00 0x02 */
{0xa4, 0x89}, /* 0x88 */
{0x96, 0x00},
{0x97, 0x30},
{0x98, 0x20},
{0x99, 0x30},
{0x9a, 0x84},
{0x9b, 0x29},
{0x9c, 0x03},
{0x9d, 0x4c},
{0x9e, 0x3f},
{0x78, 0x04},
{0x79, 0x01},
{0xc8, 0xf0},
{0x79, 0x0f},
{0xc8, 0x00},
{0x79, 0x10},
{0xc8, 0x7e},
{0x79, 0x0a},
{0xc8, 0x80},
{0x79, 0x0b},
{0xc8, 0x01},
{0x79, 0x0c},
{0xc8, 0x0f},
{0x79, 0x0d},
{0xc8, 0x20},
{0x79, 0x09},
{0xc8, 0x80},
{0x79, 0x02},
{0xc8, 0xc0},
{0x79, 0x03},
{0xc8, 0x40},
{0x79, 0x05},
{0xc8, 0x30},
{0x69, 0xaa},
{0x09, 0x00},
{0x3b, 0x42}, /* 0x82 0xc0 0xc2 night mode */
{0x2d, 0x01}, /* 0x82 0xc0 0xc2 night mode */
};

#endif

#ifdef OV7725
const uint8_t OV7725_Reg[OV7725_REG_NUM][2]=
{
{0x3a, 0x04},
{0x40, 0x10},
{0x12, 0x14},
{0x32, 0x80},
{0x17, 0x17},
{0x18, 0x05},
{0x19, 0x02},
{0x1a, 0x7b}, /* 0x7a */
{0x03, 0x0a}, /* 0x0a */
{0x0c, 0x0c},
{0x3e, 0x00},
{0x70, 0x00},
{0x71, 0x01},
{0x72, 0x11},
{0x73, 0x09},
{0xa2, 0x02},
{0x11, 0x00},
{0x7a, 0x20},
{0x7b, 0x1c},
{0x7c, 0x28},
{0x7d, 0x3c},
{0x7e, 0x55},
{0x7f, 0x68},
{0x80, 0x76},
{0x81, 0x80},
{0x82, 0x88},
{0x83, 0x8f},
{0x84, 0x96},
{0x85, 0xa3},
{0x86, 0xaf},
{0x87, 0xc4},
{0x88, 0xd7},
{0x89, 0xe8},
{0x13, 0xe0},
{0x00, 0x00}, /* AGC */
{0x10, 0x00},
{0x0d, 0x00},
{0x14, 0x10}, /* 0x38 limit the max gain */
{0xa5, 0x05},
{0xab, 0x07},
{0x24, 0x75},
{0x25, 0x63},
{0x26, 0xA5},
{0x9f, 0x78},
{0xa0, 0x68},
{0xa1, 0x03}, /* 0x0b */
{0xa6, 0xdf}, /* 0xd8 */
{0xa7, 0xdf}, /* 0xd8 */
{0xa8, 0xf0},
{0xa9, 0x90},
{0xaa, 0x94},
{0x13, 0xe5},
{0x0e, 0x61},
{0x0f, 0x4b},
{0x16, 0x02},
{0x1e, 0x37}, /* 0x07 */
{0x21, 0x02},
{0x22, 0x91},
{0x29, 0x07},
{0x33, 0x0b},
{0x35, 0x0b},
{0x37, 0x1d},
{0x38, 0x71},
{0x39, 0x2a},
{0x3c, 0x78},
{0x4d, 0x40},
{0x4e, 0x20},
{0x69, 0x5d},
{0x6b, 0x40}, /* PLL */
{0x74, 0x19},
{0x8d, 0x4f},
{0x8e, 0x00},
{0x8f, 0x00},
{0x90, 0x00},
{0x91, 0x00},
{0x92, 0x00}, /* 0x19 0x66 */
{0x96, 0x00},
{0x9a, 0x80},
{0xb0, 0x84},
{0xb1, 0x0c},
{0xb2, 0x0e},
{0xb3, 0x82},
{0xb8, 0x0a},
{0x43, 0x14},//importante
{0x44, 0xf0},
{0x45, 0x34},
{0x46, 0x58},
{0x47, 0x28},
{0x48, 0x3a},
```

-Diego

Reply

Anonymous October 30, 2013 at 12:23 AM

Sorry, i committed a mistake in my previous posts, the registers that WORK are the ones after the following statement:

```
#ifdef OV7725
const uint8_t OV7725_Reg[OV7725_REG_NUM][2]=
{
```

-Diego

Reply

Replies

Anonymous November 2, 2013 at 5:05 PM

Diego, thanks so much for pasting your registers. I, too, have spent months trying to work with this camera, and I've tried multiple OV7670s with and without FIFO buffers. I'm also trying to use RGB565 with QVGA, and your register values have gotten me closer than my previous attempts.

I'm still seeing a corrupt image, however. A typical capture looks like this: http://i.imgur.com/zwVGtsG.png

and a capture with the lens cap on looks like this: http://i.imgur.com/MTymQ1f.png

I used your values from the #ifdef onwards, but also set a bit in 0x15 so that VSYNC will be inverted (so that WRST is automatically toggled). Are you sure those are all the registers I need?

Do you know or other readers know what might be wrong with my setup?

Thanks,
Dan

(I hope this doesn't show up multiple times)

XavierJoder November 27, 2013 at 11:21 PM

Hello Dan, sorry for the late response, i know that the registers are a real pain, im happy that mine brought you closer to get an image that actually works!.
I past a lot of nights to get those registers working, i can give you some tips:
- Do NOT write more registers and more registers just to make things work, things dont fix up, they actually mess up, as an example i got the camera to work, but if i write some more registers it screw up again!
- To view a test patern, "the COLOR BAR" you need to put the lens cap, and write the register: 0x71 (the 7th bit in 1). This helps you to get the color bar that should look exactly like this: http://blog-imgs-56.fc2.com/m/i/b/mibc/colorbar.png.
- The registers that i provided should work if you feed the camera with an 24MHz clock signal, you can either provide it with a crystal, or with the MCO pin of your stm32f.
- Even when registers say "disable AEC" or whatever, AEC values do have an impact on the output, so dont just trust that disabled features wont affect. In particular case check that AEC, AGC, AWB have coherent values and same case for YUV
- I seen your images, and it seems that you have clocking issues, check the registers that take efecton the PLL, PRESCALER, PCLK, 3E, 6B, etc.
Ask anything else, because i tend to forget things with time, i just dont know why this 0v7670 works so fine all over the internet, but with some people like us it is a real pain...

Reply

fervenceslau November 8, 2013 at 7:43 AM

I'm having trouble with SCCB.

Im using the Stellaris Launchpad for that with io of 3.3V.
I tested the i2c code with an oscilloscope and it seems to be working fine.
Im feeding the XCLK with a 50% DC PWM of 10Mhz and I'm getting PCLK and HREF.

However, when I connect SDIOC to the i2c clock, the lines just stop working... everything is pulled up by the pull up resistor to the 3.3V.

Anyone knows what could be the problem? Thanks.

Reply

Replies

Marco Sebastianelli November 9, 2013 at 2:33 PM

hi I 'm using launchpad msp430f5529lp with io of 3.3V.

I think use uart i2c mode isn't correct because the uart machine (for msp430) wait acknowledge form slave but sccb don't generate ack....

I write one set of function for control sccb bus, I see at the oscilloscope and I think is correct but ov7670 don't respond.... for test my code I try to control one sensor with bus i2c and it respond....
I think there is another problem in ov7670
-vcc 3.3V
-reset 3.3V
-pwdn 0V
-xclok 16Mhz

if you want I send my code, maybe will find an error !
sorry for my english.

Marco Sebastianelli November 13, 2013 at 10:38 AM

you can try use an external resistor 10K, and disability internal msp430 pull up resistor

Reply

Anonymous November 10, 2013 at 4:21 PM

Can someone please post the registers they use to get QVGA RGB565, or VGA RGB565? I only get garbage output, like this: http://i.imgur.com/zwVGtsG.png

Thanks!

-Dan

Reply

Replies

XavierJoder November 27, 2013 at 11:25 PM

Almost forgot to tell you, make SURE that PWDN and RST pins are connected, dont just trust that they have pull up/down resistors. My registers worked, but if i dont connect this pins i get totally random data

Anonymous November 28, 2013 at 5:53 PM

Thanks for the replies, Xavier! I forgot to mention that I'm using the one with a FIFO buffer, so the RST and PWDN pins are already connected. However, I'll dump my registers and check the things you indicated. The FIFO version should feed it with 24 MHz.

I use the reset registers feature (writing a 1 to some register and then a 0, to get it to reset to default values), but considering how sensitive this stupid camera is, maybe the default values are wrong?

Diego November 29, 2013 at 12:30 PM

Hi, I'm Diego, Xavier is another name for my Google account..
I have both the FIFO version and the one without it.
My FIFO have the RST and PWDN pins available, and they must be connected. I also use the reset register feature, is the first register i write..
"how sensitive this stupid camera is" .. hahaha. yes it is a crap..
I dumped the registers and sometimes default values for 0x00 to 0x04 are wrong, so better write them, i checked all the registers, when you dump them most of them have its default values and don't worry to much about it

Reply

Anonymous December 21, 2013 at 6:01 PM

Hello,

Does anybody can comment schematics how RST and PWDN pins should be connected to STM32 pins? And what pins configuration I need to setup for those. My pilot still not starting up...

Reply

Anonymous February 13, 2014 at 4:38 PM

One thing I found is that in the modern OV7670 the RST is active low, but in an older datasheet it's active high- so be careful- for modern OV7670 chips, RST should be high for normal operation (and PWDN should be low for normal operation).

Reply

Erik Andrén Zachrisson February 17, 2014 at 9:50 AM

I'm using a ov7660 camera that is similar connected to a FPGA.
My problem is that i get a lot of noise on the inputs.
Using the internal chipscope of the FPGA I get spurious data on both Vsync and HREF.

Has anyone else experienced a similarissue?

Reply

Anonymous February 25, 2014 at 2:28 PM

Please people Im trying to make the sccb connection working with a SCCB BIT-BANG algorithm here

https://github.com/ctc8631/ov7670/blob/master/src/OV7670.c,

but I just can't get anything... I can't get any signal from the camera when I have my SIOD configured in input, like if the camera was dead. Here the sequence is simple : Start, write 0x42, write 0x02, stop, start, write, 0x43, read SIOD, stop. But nothing appears.

Heres my GPIO configuration :

```
void SCCB_SIC_GPIO_OUTPUT(void)
{

GPIO_InitTypeDef GPIO_InitStructure;

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
GPIO_InitStructure.GPIO_Pin = SCCB_SIC_BIT; //PD5
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_Speed =GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
GPIO_Init(GPIOD, &GPIO_InitStructure);
GPIOD->BSRRH = SCCB_SIC_BIT;// stay down
}

void SCCB_SID_GPIO_OUTPUT(void)
{
GPIO_InitTypeDef GPIO_InitStructure;

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = SCCB_SID_BIT; //PB4
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_Speed =GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOB, &GPIO_InitStructure);
GPIOB->BSRRH = SCCB_SID_BIT;

}
void SCCB_SID_GPIO_INPUT(void)
{

GPIO_InitTypeDef GPIO_InitStructure;
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = SCCB_SID_BIT; //PB4
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStructure.GPIO_Speed =GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
GPIO_Init(GPIOB, &GPIO_InitStructure);
}
```

I connect directly SIOD and SIOC to the GPIO pins, I tried with 10k pull ups on both lines, but nothing happened too.

RESET is to 3.3V, and PWND is to GND.

Heres my signal :

https://imageshack.com/i/15q9j8j

Don't know what to do ... Im begging for help people !!! :)

Reply

Replies

Erik Andrén Zachrisson February 26, 2014 at 12:58 AM

Are you supplying an XCLK?
Is the data pins returning any data?

You could start by getting the write working by observing that the data pins live and then write to the sleep register which should make the camera stop responding.
Then you know that the writes work properly.

Anonymous February 26, 2014 at 6:09 AM

Yep, XCLK at 16MHz, and I have all the video data outputs OK. I tried what you proposed, writing 0x11 at 0x09, but nothing works, I still have outputs on d0-d7... I tried pulling UP with 10k, with 22ohm serial to the stm32f4, but nothing... And when I simply pull up with 10k, without 22ohm serial, and if I use the I2C standard (All Open drains I/O), I have this signal :
[IMG]http://imagizer.imageshack.us/v2/xq90/842/2zxw.jpg[/IMG][/URL]
Acks are supposed to be low, not high like what I have ...

I don't know what Im doing wrong, or what Im missing...

Erik Andrén Zachrisson February 26, 2014 at 6:17 AM

What frequency is your SIOC running at?

Anonymous February 26, 2014 at 10:10 AM

22kHz or something ...

Reply

Anonymous February 27, 2014 at 3:49 PM

Ok finally I got it to work perfectly. It needed to have a 30k pull up resistor, all open drain, and my sioc signal was not good enough, in the sense it was not a good clock signal, needed to change a few delays in my bit bang sccb algorithm. Now it is perfect. :)

Reply

Replies

Sergey Dobretsov June 30, 2014 at 2:51 AM

Why did you need a pull-up resistor provided GPIO's already have such resistors (could be configured through PuPd field of init structure)?
And I wonder why did you choose namely those values for resistors (30k pull-up, 22ohm serial)?

Sergey Dobretsov June 30, 2014 at 2:59 AM

I'm trying to implement my own bit-bang SCCB version too, but can't get response from OV7670. The SCCB specifications at hand are very vague, that's why I have to make sure that what I send could be understood by OV7670. Could you please provide me with timing diagram of voltage on SIOD and SIOC pins during simple transaction (e.g. start, address 0x42, write bite, 9th bit, subaddress, 9th bit, stop)?

Reply

Anonymous March 2, 2014 at 11:53 PM

Is it possible to use subwindowing to get a higher framerate? I want capture a total of a hundred pixels with resolutions ranging from 2x50 to 10x10. I need 1000+ frames per second for my project. I looked into using a optical mouse but they are usual only 15x15 and capturing the indivual pixel data for each frame is very slow compared to a camera.

Reply

Chris Riether March 2, 2014 at 11:57 PM

Is it possible to use subwindowing to get a higher framerate? I want capture a total of a hundred pixels with resolutions ranging from 2x50 to 10x10. I need 1000+ frames per second for my project. I looked into using a optical mouse but they are usual only 15x15 and capturing the indivual pixel data for each frame is very slow compared to a camera.

Reply

Anonymous March 3, 2014 at 2:23 AM

I am interfacing omnivision's (ov5640)image sensor with the FPGA and was able to capture raw frame in YUV422 format through the digital video port (DVP) parallel output interface.Now I am trying to capture the JPEG compressed output.The datasheet doesnot provide much detail of the JPEG mode and Since I have never studied JPEG mode I am unable to understand what the sensor will output in this mode(what bytes mean e.g. in raw yuv mode I knew the bytes coming are luma and chroma intensity values) and Whether the sensor will output only the scan data or it will ouput all other necessary headers and info. that is required to make a complete JPEG file.I will be thankful for any help

Regards
Az Ahmad

Reply

Ferhat Yol March 7, 2014 at 5:07 PM

Hi.

I want to use STM32F4-Discover the OV7670 camera. What I do not run the camera.
I'm giving the camera 8Mhz clock signal from XCLK pins. But the camera does not give I2C ACK signal from the line. I send from the device address 0x42. I'm sending the camera device my address wrong.

Why do you have such a problem?

Reply

Replies

supun tharanga May 7, 2014 at 5:37 AM

What kind of wires are you using to connect camera to micro controller. due to the impedance mismatch I2C SCL line will distorted. always use thin small lengrh wires to connect camera with micro controller

Reply

Sai Sharan March 8, 2014 at 3:46 AM

sir, i m dng a project with arduino UNO..i have a camera module ov7670 which i want to connect to UNO board..for capturing and save it into the sd card i need a arduino code for it i m trying from 10 dys can any 1 help me.. my email id: sharanfrndz@gmail.com

Reply

Chris Riether April 2, 2014 at 7:53 AM

I was wondering if there was a way to get higher then 30fps by sampling a very small array of pixels (30x40). You said that that the framerate is set by the clock speed and the image resolution. Is it possible then to go beyond 30fps (I need something around 200-400FPS) by sampling a very low resolution image (at least 20x20)

Reply

vvavepacket April 12, 2014 at 8:33 AM

Im getting this image. What do you think is the problem?

http://puu.sh/85G4p.jpg

Reply

Erik Andrén Zachrisson April 14, 2014 at 5:31 AM

Are you using the pclk to save the incoming pixels?
If not, try to use the internal clock to save the image.

Reply

Replies

vvavepacket April 23, 2014 at 6:28 AM

@Erik its working now. But i have one issue. I cant write to registers using Arduino I2C

Reply

vvavepacket April 23, 2014 at 6:30 AM

Im using Arduino UNO to read the values of OV7670 registers.. Im using the Wire library..

Reading the registers works fine

```
int addressread = 0x21;
int address = 0x1;
int data = 0x3;

-------------------------------------code---------------
Wire.beginTransmission(addressread);
Wire.write(address);
Wire.endTransmission();
Wire.requestFrom(addressread,1);
while(!Wire.available())
{
}
int c = Wire.read();
Serial.println(c);
-------------------------------------------------------
```

but when I am writing registers, it doesn't work

```
--------------------------------code--------------------------------
Wire.beginTransmission(0x21); // i2c slave address
Wire.write(0x1); //register address to write
Wire.write(0x3); //data to write
Wire.endTransmission();
----------------------------------------------------------------
```

executed the above write function, and when I read back, it doesnt change.

Please help

Thanks

Reply

supun tharanga May 7, 2014 at 3:53 AM

Hello jorge,

Thank you for your post and replies for above comments. They were really useful for me to sucess with STM32F4 discovery board connecting to OV7660 camera module. OV 7660 is almost same as OV7670 . I included all the details on my blog. Thanks again for your help.
Cheers!

Reply

Replies

Anonymous June 18, 2015 at 5:51 AM

Hello
Is it sufficient to supply VDD pin of ov7670 module with 3V supply pin of

stm32f4discovery?
Thanks.

Reply

Raman Jairath May 14, 2014 at 1:23 PM

very informative post buddy

Spy Camera Supplier in Delhi

Reply

Anonymous June 29, 2014 at 6:35 AM

Hello,

Thank you very much for your post and replies for above comments.
I´m trying to interface OV7670 but I have a problem with output data. Camera still returns data on parallel bus Y: <5 Cb: >123 Cr: >123 and data wont change when I remove the cover -> still returns black image. Can anyone help please? Can be the problem, that XCLK runs only at about 2MHz?

Thanks very much for replies
Mišo

Reply

Sergey Dobretsov June 30, 2014 at 2:38 AM

Dear George!
I'm trying to implement my own bit-bang SCCB version, but can't get response from OV7670. The SCCB specifications at hand are very vague, that's why I have to make sure that what I send could be understood by OV7670. Could you please provide me with timing diagram of voltage on SIOD and SIOC pins during simple transaction (e.g. start, address 0x42, write bite, 9th bit, subaddress, 9th bit, stop)?

Reply

Ethan Smith September 1, 2014 at 2:21 AM

Buy & sell new and used iPhones.
Wide selection of new and used iPhones for sale.
More at here

Reply

Anonymous September 9, 2014 at 6:05 PM

I get this color bar: http://i58.tinypic.com/30cnhpw.png
What can be wrong?

Reply

Anonymous September 10, 2014 at 8:43 PM

PS: I improved image with RGB555, D7 line of sensor is stuck at GND

Reply

Jair Olguin September 15, 2014 at 4:22 PM

ESTOY UTILIZANDO TU LIBRERIA PARA EL SCCB, LA TRANSCRIBI A MIKROC PRO FOR ARM Y NO FUNCIONA :(NO SE SI PUDIERAS AYUDARME

Reply

Anonymous October 7, 2014 at 1:53 AM

At least my camera needs a pullup resistor on the SIOD pin, maybe that's your issue?

Reply

Igor Borges Tavares October 7, 2014 at 8:46 AM

Hi! What is the black strip in the left of the images?

Reply

Akin Yildiz November 1, 2014 at 2:28 PM

hello sir, thank you for a very in-depth post. maybe you can help me with my question; http://www.instructables.com/community/arduino-wifi-time-lapse-camera/

Reply

ACG December 1, 2014 at 12:06 AM

I can confirm that the OV7670 blue boards required input on XCLK for the SCCB to respond. No FIFO, with onboard 3.0v and 1.8v regulators. That is with default register values. There may be some value to change that but you would have to communicate with it to change it. Chicken Vs. The Egg.

After I made some bit-bang code, PIC32EP series, for write and read all seemed good. Most of which I ported over old code from another project. However the data for read operations was 50/50 consistent. 1/2 the time the camera did not ACK (drop the 9th bit low) after sending it the first phase or even second phase ID address. My issue was I did not toggle the SDA (SIOD) line before toggling the clock. The PIC32EP series is 5 times faster than my older chipset!! 14.2nanoseconds between the two. Its a 70mips pic for goodness sake. After a day it dawned on me I need to add some extra delays before toggling the clock line SCL(SDIOC)

Main tip for SCCB control. Make sure you toggle data way before toggling clock. If you have no scope around, start with 10microseconds between the two. Just send in a loop the ID address. If you dont get an ACK each time, you need to fix your code before bothering to do anything else.

Be advised, the blessing of speed from newer micro's can be a curse some times.

Reply

Replies

Anonymous December 2, 2014 at 11:16 PM

Upps, meant PIC24EP. I wish Microchip would come out with a 32bit version. Not the borrowed from someone else's 32bit micros they sell, real genuine microchip instruction set based 32bit micro.

Reply

Ionut Pop December 3, 2014 at 3:24 PM

I am confused by something:
In the timing diagrams in the datasheet, the QVGA and other alternate formats are presented as simply skipping some lines;
However, the QVGA lines are shorter than the VGA(this is not outlined in the diagrams).

How are the lines going to be read? Will the pixel clock be divided? Or will the HREF pulses be actually shorter than presented in the diagrams?(they are presented as equal to the vga setup)

And another thing: is it really true that you can't get a bigger framerate with lower image formats? Is there any way around? Could I try to take a limited number of lines from the camera and then force it to restart the pixel parsing?

Reply

Big Tuna December 5, 2014 at 1:04 AM

Greetings Jorge,

First of all I would like to say great blog! I learned a lot about pixel formats and the OV7670. Would it be possible for you to send me the windows 7 installer .exe file for this version of qSerialTerm or the most up to date one? I downloaded QT Creator and your source code only to painfully realize I had no idea what I was doing... It would be greatly appreciated!

Tuan.n604@gmail.com

Reply

gauravRaina January 15, 2015 at 10:19 AM

Thanks a lot Jorge Aparicio for sharing your work with us. I could implement this project only because of your excellent documentation :)

Reply

Pravin Xeona January 26, 2015 at 2:26 AM

Hi, can anyone please share the register values for configuring the camera in RGB555 format? Thanks in advance...

Reply

rohan raghunath March 11, 2015 at 3:13 AM

hello can any! help or gude how to capture the image from OV7725 or OV7670 n send that image as mms using sim900 gsm moddule

Reply

Anonymous March 29, 2015 at 3:00 PM

What is the SCCB Interface and how to use one; sry i'ma started in this and would like to thank all those who would help me

Reply

Sheikh Asif April 16, 2015 at 6:17 AM

Hello guys,
I need help to implement ov7670 with arduino??? i try but fail to write in ov7670 register.What can i do now?plz help.

Reply

Anonymous June 5, 2015 at 8:58 PM

If a pclk of 24 mhz gives you 30 fps what will 48 mhz produce?

Reply

aseok July 3, 2015 at 2:48 PM

Hello
I should tell Jorge Aparicio that according to ov7670's datasheet,raw RGB(888) is supported. Please correct your blog.

Reply

Replies

aseok July 3, 2015 at 3:18 PM

Sorry for my misunderstanding, RGB888 isn't supported.
What is output data format in test pattern mode?i tried to change it to RGB565, but the output didn't change,so it seems that test pattern is just in YCbCr4:2:2 mode,is it right?

Reply

Gani khan SEO July 4, 2015 at 4:44 AM

Sakshmart - Online digital camera, online sale for mobile, Bluetooth smart watch, USB rugged nylon cable and baker premium set for
USA customers.

Visit for Online Product :- **Online Digital Camera in usa**
For support please email :- sale@sakshmart.com

Reply

uffi63 July 22, 2015 at 7:05 AM

I investigated the YCbCr Output of OV7670 camera, which is erroneously named YUV Format in the specification / daata sheet.
I found out that the Byte order corresponds to the YUV2 Format which is:

Y0 Cb0 Y1 Cr0

so the information given above on YUV / YCbCr Byte order is not correct for this cam OV7670.

Reply

uffi63 July 28, 2015 at 9:36 AM

Sorry, instead of YUV2 Format I was talking about YUY2 Format according to the Four CC. So:

4:2:2 Formats, 16 Bits per Pixel

Two 4:2:2 formats are recommended, with the following FOURCC codes:
•YUY2
•UYVY

Both are packed formats, where each macropixel is two pixels encoded as four consecutive bytes. This results in horizontal downsampling of the chroma by a factor of two.

YUY2

In YUY2 format, the data can be treated as an array of unsigned char values, where the first byte contains the first Y sample, the second byte contains the first U (Cb) sample, the third byte contains the second Y sample, and the fourth byte contains the first V (Cr) sample, as shown in the following diagram.

If the image is addressed as an array of little-endian WORD values, the first WORD contains the first Y sample in the least significant bits (LSBs) and the first U (Cb) sample in the most significant bits (MSBs). The second WORD contains the second Y sample in the LSBs and the first V (Cr) sample in the MSBs.

YUY2 is the preferred 4:2:2 pixel format for Microsoft DirectX Video Acceleration (DirectX VA). It is expected to be an intermediate-term requirement for DirectX VA accelerators supporting 4:2:2 video.

UYVY

This format is the same as the YUY2 format except the byte order is reversed—that is, the chroma and luma bytes are flipped (Figure 4). If the image is addressed as an array of two little-endian WORD values, the first WORD contains U in the LSBs and Y0 in the MSBs, and the second WORD contains V in the LSBs and Y1 in the MSBs.

Reply

uffi63 July 31, 2015 at 9:24 AM

Aat least this byte order in the Format YUY2, ie. YUYV is the default output for the default register settings of the registers TSLB and COM13. Using these registers you can also swap the bytes according to your wishes.

Reply

Shiya Priya August 20, 2015 at 5:18 AM

Nice Website...
Hey JOIN now fblikesbot.com and Increase Facebook Likes your profile and websites.
Increase Facebook Likes and check your website worth worth my websites
its may be very beneficial for you also really

Reply

HealthTips September 13, 2015 at 8:59 PM

See also more and compare for best prices deals for Camera Sale here!

Reply

Clash Of Clans YT October 4, 2015 at 6:34 PM

Download This Game Hack Tool 2015 Free Working Here:

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

Reply

Clash Of Clans YT October 5, 2015 at 4:41 AM

Download This Game Hack Tool 2015 Free Working Here:

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

http://dlhack.com/download/hack-tool

Reply

Ivan 000 October 8, 2015 at 12:58 PM

Hello all
am looking few years that some guys comes into the market they called themselves hacker, carder or spammer they rip the peoples with different ways and it's a badly impact to real hacker now situation is that peoples doesn't believe that real hackers and carder scammer exists. Anyone want to make deal with me any type am available but first I'll show the proof that am real then make a deal like
Wire Bank Transfer
Western Union,
Money Gram
SSN
Hacking stuff
Shipping product. serious / needy contact about it.

lykovine@yahoo.com

Reply

Lincong Li October 10, 2015 at 1:11 AM

This comment has been removed by the author.

Reply

shahbaz alam October 11, 2015 at 8:24 PM

This comment has been removed by the author.

Reply

Ivan 000 October 20, 2015 at 1:38 AM

Hello all
am looking few years that some guys comes into the market they called themselves hacker, carder or spammer they rip the peoples with different ways and it's a badly impact to real hacker now situation is that peoples doesn't believe that real hackers and carder scammer exists. Anyone want to make deal with me any type am available but first I'll show the proof that am real then make a deal like
Wire Bank Transfer
Western Union,
Money Gram
SSN
Hacking stuff
Shipping product. serious / needy contact about it.

lykovine@yahoo.com

Reply

Dvidz Cpora October 29, 2015 at 10:31 PM

Bonjour, à tous j'offre et partage des codes Free Wifi Code Orange et SFR chaque semaine pour vous connecté a internet dans toutes la france

Mon site de partage ▄︻̷̿┻̿═━一 http://caribbeansunshine.co/

identifiant wifi gratuit

Reply

nick vjuice November 4, 2015 at 4:00 PM

contact point.business@yahoo.com for your bank logins and bank transfers, western union transfer,carding of online stores and flight tickets.
i have been doing this for two years and its 100percent guaranteed and sure. ripping and scamming is for low lives. before any transfer you pay 70percent before transfer and
30 percent after transfer

for bank transfers maximum time is 8 hours
bank transfers:
$2500 - $250
$3000 - $300
$5000 - $450
$10000 - $900
$15000 - $1200
$20000 - $1800

mode of payment: bitcoin and perfect money

For Usa BANK Transfer Info Needed:
1: Account Number
2: Routing Number
3: Bank Name:
4: Account Holder Name:
5: Account Holder Phone Number:

For Uk BANK Transfer Info Needed:
1: Account Number
2: Sort Code
3: Bank Name
4: Account Holder Name:

For International BANK Transfer info Needed:
1: Account Number:
2: Iban Or Swift Code
3: Bank Name:
4: Account Holder Name:

2)western union: maximum of three hours
2000 Transfer = 200
3000 Transfer = 250
4000 Transfer = 350
5000 Transfer = 500
7000 Transfer = 700

For Westren Union Transfer Info Needed
1: First Name
2: Last Name
3: City
4: Country
5: Email for confirmation

for carding and other services. prices varies. contact me on my email. point.business@yahoo.com

Reply

Xavier Bahlmain December 15, 2015 at 1:37 AM

http://hackerbuddy.com/edges/100 incase of any needed hacks.

Reply

Enter your comment...

Comment as: Unknown (Goo ▼ Sign out

Publish Preview □ Notify me

Load more...

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)