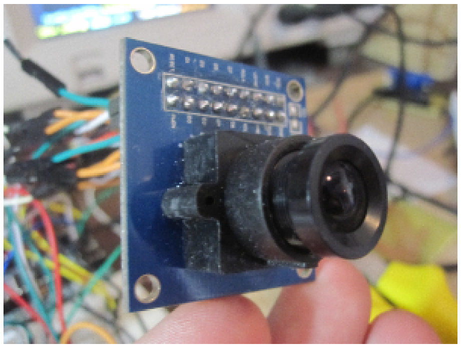
qrfnfgre'f oybt

Whfg nabgure JbeqCerff.pbz fvgr

ov7670 camera sensor success!

Posted on May 7, 2012



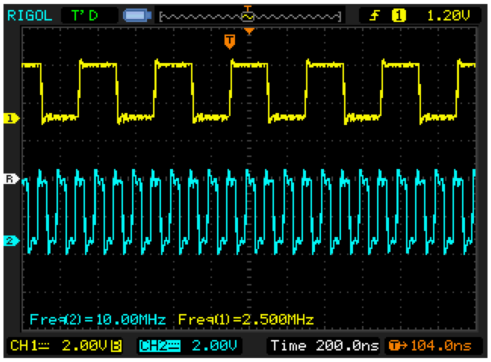
The ov7670 camera sensor, cheaply available from ebay, has been the bane of my existence during my short electronics hobby career. I bought my first ov7670 sensor maybe a year ago, and have been tearing my hair out trying to get it to respond. It seemed completely dead. It didn't respond to any form of i2c communication, and showed no signs of life in the oscilloscope.

Every now and then, I tried to get it to work using various approaches, but to no avail, and was ready to completely give up. However, now after seeing some recent success stories with the ov7670 here and here, I decided to give it one more go.

I'm pretty new to electronics, and I'm learning new stuff every day. On my most recent attempt at breathing life into the wretched thing, my attention got directed towards a feature I had not even considered before – the **XCLK** pin. This pin labeled "external clock", always seemed to me like it was something optional, as I've gotten used to all my other toys having internal RC oscillators, or no need for oscillators. But no, in this case it's not optional, and my oversight seems so obvious now!

I've read pretty much every document available on the internet about the ov7670 sensor, but to this day, I'm not sure it's clearly stated anywhere that feeding the XCLK is an important step to getting the ov7670 up and running. It's probably listed in every schematic out there, but my newbie mind ignored it.

So, to feed the camera some clockage, I used the CLKOUT feature of the LPCXpresso to generate a frequency of ~10Mhz, and lo and behold, it's alive! **Alive!**

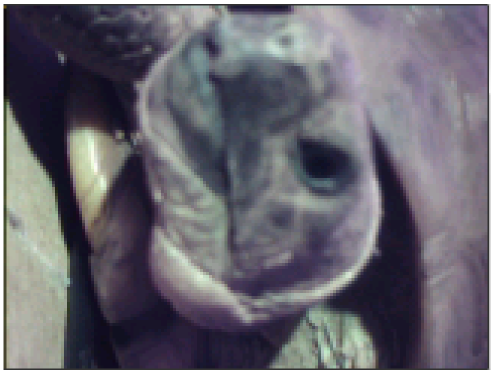


From this point on, working with the camera was pretty straightforward. Analyzing the sync signals and reading the digital color signals with the aid of the provided pixel clock was easy as a pie. Some minor headaches with the clock running so fast, but that was possible to tune down to more workable levels.

The smallest resolution the camera supports is QQVGA (160×120), and with rgb565 color space, that fits quite neatly in the 64K of memory that LPC1769 has.

To pump out the image to my computer for viewing, I initially used the semihosting feature of LPCXpresso – just for testing. Printing out the full image with printf's took nearly an hour! But after a few attempts I managed to conjure a sensible image out of the data, crowning my success with a picture of my stupid face.

Next step was to output the data via UART at more convenient speeds. Initially I didn't seem to be able to pump the data to my computer at speeds more than ~57600. Somewhere the data got buffered, and when it buffered enough, it went nuts. So I adjusted the serial protocol a little, so the client sends commands to request lines one at a time, and the lpcxpresso responds with data. This keeps both parties happy about the rate of data flowing, and I was able to increase the baudrate to 921600.



My good friend Oona. The colors could use some tuning, but I quite like the result already.

Now I was able to view a new image from the camera every 1-2 seconds. This almost feels like watching a video instead of a slideshow :) I could do some tricks to get faster framerates, but actually I won't bother. My initial thoughts with the camera were to use it for still images taken on request, and to transfer them wirelessly via XBee, so in future I'll use the camera at a much slower pace.

In case someone reads this page one day, looking for advise with the ov7670, I'll list some notes here that might be useful:

- Needs a clock source
- i2c is dodgy, I had to add some delay to read routines
- different versions exist on ebay, some take 2.8V instead of 3.3V, and some have FIFO

I've dumped the project source to github at https://github.com/desaster/ov7670test. At the time of writing, the pygame + pyserial client isn't there, but I might add it later after I work on it a bit.

Share this: Twitter Reddit Print

Like this: ★Like One blogger likes this.

This entry was posted in electronics and tagged arm, camera, cortex, lpc1768, lpc1769, lpcxpresso, m0, mcu, microcontroller, nxp, ov7670. Bookmark the permalink.

37 Responses to *ov7670 camera sensor success!*

Ad Se *says:*

July 25, 2012 at 11:13

Can you post the wiring diagram wit voltages and IO names both sides?

Reply

desaster *says:*

July 29, 2012 at 13:15

Well I've taken apart the stuff long time ago already.. but I think the connections should be pretty self explanatory at least on the camera side. And on on the MCU side, it's really up to you.

I think the source code also has most of the pins explained.

If you're working on the camera yourself, I'd be very interested in hearing about your experiences with it! :)

Reply

diego *says:*
August 8, 2012 at 15:04

hello, good information. But indon know how i will can see the captures images in pc, sended with uart, can you help me?

Regards

Reply

desaster *says:*
August 10, 2012 at 18:54

For my application, I wrote a python app using pyserial that reads the raw data by using a simple command/response protocol, and then displays the data by using pygame. You can see an example of how I did it in my github repository.

Reply

diego *says:*
August 10, 2012 at 22:22

Thank you,
Pytom it,s a stranger for me:) i will seach about this in google..

In a week i will receibe my camera, :)

Thanks, and regards from spain

diego *says:*
August 21, 2012 at 22:40

Hello,
the last friday receibed the camera and today i,m test comonication with sccb bus, with any dificults, but work i think…

Can you help me any more with how to use your aplication for pythom?

Best regards

Reply

JS *says:*

September 29, 2012 at 14:12

Hello! I am new to lpc1769 and i have been trying to get the ov7670 camera working. I followed your instructions and used your project source from github. However, when i debug and run the code, I realised that the code doesnt run beyond "while (I2CEngine() == I2CSTATE_SLA_NACK);" under the ov7670_get function. Would appreciate it if you can assist me with the troubleshooting.
Thanks!
Best Regards

Reply

desaster *says:*

September 29, 2012 at 14:44

I spent a *lot* of time trying to get the i2c working using various microcontrollers, timings, etc. In the end however it wasn't really the fault of the i2c code, but the way the camera was connected and powered.

Like I wrote in the blog, my biggest obstacle was that I wasn't providing a clock to the camera (non-FIFO verison that is). Without it, even i2c will not work. However there could be many more reasons, like the i2c connections themselves (pullup resistors).

Maybe if you provided a schematic of your setup, I could see if there's some obvious fault.

Also, for me, it was very important to have a simpler i2c device to verify that my i2c code and the bus was working properly. I used a simple foolproof i2c eeprom for that.

Reply

JS *says:*

October 2, 2012 at 16:36

Thank you for your prompt reply! I have tried figuring out for the past few days but to no avail.

This is the link to the schematic of my setup:
https://docs.google.com/document/d/1htmgQO7R7Ofnz7y8e_Dm-VtTWBKdlKjJUCyPvKx6RB8/edit?pli=1

This is the eeprom I am using (24lc512) :
http://sg.element14.com/microchip/24lc512-i-p/eeprom-serial-512k-24lc512-dip8/dp/9758020

This is the camera which I am using(VGA OV7670):
http://www.ebay.com/itm/-/230684824130?item=230684824130&ViewItem=#ht_6726wt_1053

Thank you again for assisting with my troubleshooting. Look forward to your reply soon.

desaster *says:*

October 2, 2012 at 17:09

Hey,

I don't see SIOC and SIOD from the camera connected to the lpc1769. They are the i2c bus that you were having problems with. Or did I misinterpret the drawing?

Your eeprom seems to be connected, but is lacking pullup resistors? Also seems the SCL and SDA should be swapped around (but check this from the lpc1769 datasheet). Did you have your EEPROM working, though?

JS *says:*
October 2, 2012 at 17:35

Thanks! you didnt misinterpret. I forgot to connect those pins! May I know where should SIOC and SIOD be connected to?

What values should I put for my pull-up resistors? I would need to place pull up resistors for both SDA and SCL pins of the eeprom right?

I think I connected the SCL and SDA properly but I drew it wrongly. Sorry about that! I am new to lpc1769, hence thank you for the patience.

Thanks!

Reply

desaster *says:*
October 2, 2012 at 17:59

I suggest you look up i2c bus on wikipedia for example to learn some basics about what it is about. In ov7670 you use it to modify settings of the camera, an on eeprom you use it to write and read data.

Ov7670 calls the i2c bus “SCCB”.

Your i2c bus should have a pull-up on both wires, and they can be something like 2.2k. I used 10k myself.

i2c bus has two wires: clock and data.

on eeprom: SCL = clock, SDA = data.
on ov7670: SIOC = clock, SIOD = data.

You can connect both these devices to the same i2c bus, and to the same connections in your lpc1769.

Reply

JS *says:*
October 7, 2012 at 08:49

Heyy. Thanks for assisting me with the camera! Cam test is working fine now.

For the python code, I installed python 2.7.3 and the dependencies as required. When I run the code, i was shown with this error.

——

```
Traceback (most recent call last):
File “C:\Users\acer\Desktop\desaster-ov7670test-14756aa\utils\camview.py”, line 148, in
Application()
```

```
File “C:\Users\acer\Desktop\desaster-ov7670test-14756aa\utils\camview.py”, line 108, in __init__
self.serial = SerialPort(self.ov7670, 5, reactor, baudrate=921600)
File “C:\Python27\lib\site-packages\twisted\internet\_win32serialport.py”, line 35, in __init__
xonxoff=xonxoff, rtscts=rtscts)
File “C:\Python27\lib\site-packages\serial\serialwin32.py”, line 30, in __init__
SerialBase.__init__(self, *args, **kwargs)
File “C:\Python27\lib\site-packages\serial\serialutil.py”, line 260, in __init__
self.open()
File “C:\Python27\lib\site-packages\serial\serialwin32.py”, line 56, in open
raise SerialException(“could not open port %s: %s” % (self.portstr, ctypes.WinError()))
SerialException: could not open port COM6: [Error 2] The system cannot find the file specified.
>>>
```

———

Would appreciate if you can assist me with this error, perhaps letting me know where I have gone wrong. Appreciate your help so far! Thanks alot!!

desaster *says:*
October 9, 2012 at 13:59

The error is that it can’t open COM6.. is that the correct com port you want to use?

geo *says:*
October 4, 2012 at 10:11

Hi!

I have followed the above entries and your correspondence with JS to fix up my camera. I have installed python and the various dependencies needed in your source code. However, I have experienced this even after reinstalling twisted and pyserial. Did you encounter any of this during your troubleshooting? The link has the image of the error after running the python code.

https://docs.google.com/document/d/1Po1i0hg2q1ypGkwjbADxnX3686LGtpxnF9uCBeDVALk/edit

I have installed the .egg and other packages from .exe files smoothly but just can’t get the python code that you’ve supplied to run. It would be good if you could shed some light on the error.

Thanks for the help!

Reply

desaster *says:*
October 4, 2012 at 10:20

I’m not sure, but maybe make sure your python2.x and all the dependencies are 32bit versions.

Reply

geo *says:*

October 4, 2012 at 11:01

Hi there! Thanks alot for the help. Does it matter if I am using Windows 7 64bit version? Could that be the source of the problem?

desaster *says:*
October 4, 2012 at 11:10

The problem could be that you are mixing 64bit python with 32bit libraries. I would just use 32bit version of everything.

haris *says:*
October 5, 2012 at 02:54

hi. nice blog to see. i am using ov7690 with stm32f4. i have manage to read write register settings.
i am using hardware sync.
but the problem is with image data output. I am receiving continuously 0×02.
what can be the problem? it may be clk polarity settings of vsync,hsync, pclk or what?
thanks

Reply

desaster *says:*
October 5, 2012 at 08:48

I'm not sure how to help you here. How are you reading the image data? Do you have the same non-FIFO model of the camera as I do?

Reply

haris *says:*
October 5, 2012 at 16:45

there is dedicated data register in STM32F4 i.e DCMI_DR and dedicated pins for digital output DATA from camera.
I can read and write register settings of camera.and SCCB works fine.

I am continously transmitting the DCMI_DR register every after some micro seconds through USART to PC. (i know i would lost lot of data by this method and I have to use DMA… but the for checking purposes I am directly transmitting the DCMI_DR)

I have checked by manually pull-up and pull-down the different data PINS at STM32F4 . and I got the correct results on my serial. and I got the change in run-time.

but when I connect these PINS to data output of camera, it contnously gives 0×02 on my serial.

may be i am setting the camera register incorrectly OR there is difference between clock polarity settings (vsync,hsync, pclk) of camera and STM32F4?

even if am not setting the register correctly, i think there should be some DATA at least variable DATA on data output PORT of the camera. not only constant 0×02.

=> stm32F4 has bulit-in FIFO.
=> (if i would use DMA) DMA picks up data from DCMI_DR in STM32F4.

thanks. help required.

if u say i can send main routines of my code.

Reply

desaster *says:*
October 9, 2012 at 14:01

I don't have a STM32F4, and I'm not exactly sure what you are doing, but I would be very interested in seeing your code.

Reply

JS *says:*
October 10, 2012 at 07:18

Hey thanks for the reply. COM6 is not the port I want to read the values from. Instead, its COM4. Hence, i change your camview.py code at line 108 to self.serial = SerialPort(self.ov7670, 3, reactor, baudrate=921600) as I realised that it will always +1 to the number i put in the line of the code and open the corresponding port. Eg: I put in 3 it will open port 4 which is (3+1). Having successfully open my desired port of which the uart serial usb cable is connected to, I am however still unable to get any image. I am not sure what I miss out, Perhaps asking python to read another port requires more changes to the code than just that line? All it shows is a blank/black pygame screen. You can go to this link to view the pictorial image of what I see.

https://docs.google.com/document/d/1htmgQO7R7Ofnz7y8e_Dm-VtTWBKdlKjJUCyPvKx6RB8/edit?pli=1

I am not sure what could be the problem now, as I have gotten the camtest successful stage as seen in the pic in the link above.

For my uart transfer, I am using a FT232R chip of which I only two connections beside the usb cable to my laptop: I connect TXD of the chip to RXD0/P0.3/(J6-22) of the lpc1769 and RXD of the chip to TXD0/P0.2/(J6-21) of the lpc1769.

I got a feeling my uart0 is not transmitting, hence receiving a blank image. Maybe I miss out on something. Appreciate your help if you can help me spot any obvious errors or let me know what might have gone wrong.

Thank you so much!

Reply

desaster *says:*
October 10, 2012 at 13:30

Not sure if it matters, but I ran my lpc1769 at 120mhz, while yours seems to be running at 100mhz. Search google for how to change the clock speed, IIRC it requires a small change to system_LPC17xx.c.

Anyways, before even touching the python script, you should verify that the UART communication is working properly by using a serial console manually. Study main.c to learn how the commands work (getimage, getline, regr, regw).

Also, it sounds wrong that your serial chip only has two connections – GND should be connected too!

Reply

JS *says:*

October 27, 2012 at 18:34

Heyy thanks for all your help!! I managed to get the image on the pygame display and I have been studying the source code and lpc1769 for the past week to help me understand more about how it works. My camera is getting a new image every 2-3 seconds and I was wondering if you are able to advise if there is any way to increase the frame rate to lets say maybe to the extent of like when you are watching a video/less than 1second per frame? Just wondering if that is possible as I am with limited knowledge and was thinking about whether this is possible or not. Thanks again!!

desaster *says:*

October 30, 2012 at 10:45

Well, I increased the FPS by bumping up the UART speed, however at some point you are bound to hit a limit on that.

Generally UART is not the best way to transfer video, though, and I've only used it for testing purposes. For a real life application you'd want to use some high speed interface.

haris *says:*

October 11, 2012 at 20:09

hi:
how can I display image receive on hyper terminal?

Reply

Peter *says:*

October 15, 2012 at 15:54

Hi. nice blog to see. I am using OV7670 moudle(AL422 FIFO witj LD0) with Arduino Mega2560

I just want my camera taken photo and save in registers

I search from net from very long time I can't found the ans i wanted

Could you help me to solve my problem

THX

Reply

desaster *says:*

October 15, 2012 at 16:27

Hi Peter,

I now have a FIFO version too, but unfortunately I haven't gotten around to trying it yet.

Maybe sometime in future I will try it and write a small blog entry about it.. these cameras seem to have gotten really popular lately.

Reply

desaster *says:*
October 20, 2012 at 19:19

Wrote something on the FIFO version now:

http://qrfnfgre.wordpress.com/2012/10/20/ov7670-fifo-msp430-launchpad/

Reply

Peter *says:*
October 15, 2012 at 16:40

THX for ur ans i very grateful

if my camera like yours how should i do

do u know hw to do that

THX

Reply

desaster *says:*
October 15, 2012 at 16:47

For camera like mine, I don't think any atmega will run fast enough to be able to process the signals. However, I'm not sure how slow the camera clock can be driven..

Reply

Peter *says:*
October 15, 2012 at 17:34

Here i had the code and data from who sell the camera

could u like to see it

http://www.mediafire.com/?8yr9tgetjabr3ls

THX

Reply

ycmoh *says:*

October 15, 2012 at 17:59

Hi. I'm YC and I'm from spain.

I was inspired by your blog and I went ahead to purchase one from ebay to try it out with the lpc1769 that I have lying around. Fortunately (unfortunately), the version I bought came with the FIFO. (Whats the difference btw the FIFO and non-FIFO version? What is the FIFO used for??????????)

I shall try to figure it out too. Looking forward to your future post!

Reply

desaster *says:*

October 20, 2012 at 19:19

Wrote something on the FIFO version now:

http://qrfnfgre.wordpress.com/2012/10/20/ov7670-fifo-msp430-launchpad/

Reply

Peter *says:*

October 17, 2012 at 13:18

hi there did you found the way to conquer the question ,yet

if you found the way to solved the question please must be that us know

I really desperate to konw

THX for the helping

I will very greatful

Reply

Pingback: *OV7670-FIFO + MSP430 Launchpad | qrfnfgre'f oybt*

qrfnfgre'f oybt

Theme: Twenty Ten *Blog at WordPress.com.*