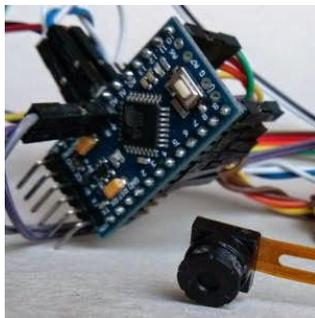


[Blog](#)[About](#)[Contact](#)

Sep 27, 2014

Arduision (I): embedded computer vision with arduino



Ok, its time to share stuff again.

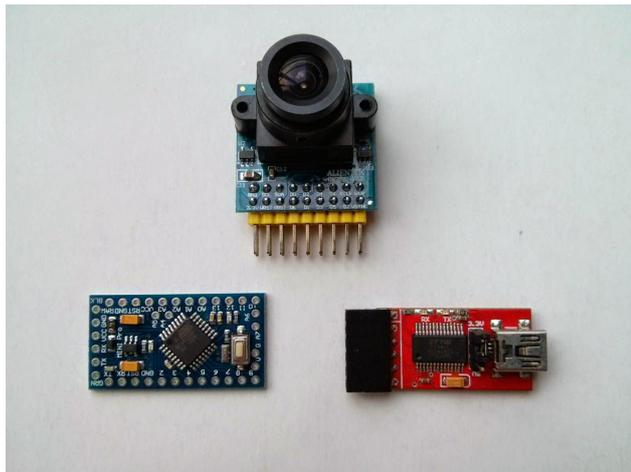
I've spent most of this year's (2014) summer learning how to use a CMOS sensor together with an arduino (ATmega328p) microcontroller, in the process of developing some personal projects.

This is the first of a series of articles to publish my experiencies, and hopefully save someone's time.

In this first post, I'll just release the minimal info that may let you start right away experimenting without reading about how things work. In further articles I'll make detailed explanations of the hardware and software, and the decisions

and reasons that led me to my current setup. However, at the end of this post, you may find links to useful documentation.

My system is based on Linux, Ubuntu 14.04 x64. You may need to adapt things to your system. You are assumed to have some knowledge of the [arduino](#) platform and environment, and also the [Processing.org](#) development environment. If so, here is what you need to clone my current setup:



WARNING:

**DO NOT USE AN ARDUINO THAT
OUTPUTS 5V ON THE DIGITAL PINS.**

HARDWARE NEEDED:

- Arduino compatible development board with ATmega 328p, rated at 3.3 volts. I use a clone



NO.1 S6 MTK6589 Quar
Core 1.2GHz Android
4.2 Smartphone w/ 5",...

DKK1,098.16

Blog history

- ▶ 2015 (1)
- ▼ 2014 (4)
 - ▼ September (1)
 - Arduision (I):
embedded computer
vision with ardu...
- ▶ April (1)
- ▶ February (2)
- ▶ 2013 (15)
- ▶ 2012 (5)

of a Sparkfun Arduino Pro Mini at 8MHz and 3.3v. This doesn't have USB, so I use a USB to serial adapter that works at 3.3v. There are other boards that should work too (not tested): Arduino Pro (3.3V version), OLIMEXINO-328, Iteduino Uno, Seeduino, Freaduino, Youduino, Bouno Lite, etc.

- A CMOS module with FIFO and an ov7670 or an ov7725 sensor (I've tried both, and setup the register parameters). At the moment I'm working only through the FIFO, but may try synchronize the microcontroller clock directly with the CMOS sensor, as int the AVRcam project.

For this setup the module must have the following available pins (numbering may differ):

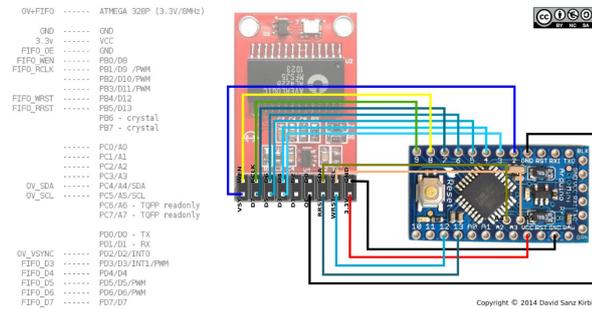
GND	---	1		2	---	VCC3.3
OV_SCL	---	3		4	---	FIFO_WRST
OV_SDA	---	5		6	---	FIFO_RRST
FIFO_D0	---	7		8	---	FIFO_OE
FIFO_D2	---	9		10	---	FIFO_D1
FIFO_D4	---	11		12	---	FIFO_D3
FIFO_D6	---	13		14	---	FIFO_D5
FIFO_RCLK	---	15		16	---	FIFO_D7
FIFO_WEN	---	17		18	---	OV_VSYNC

I've tested an ELH 30w FIFO module and the Alientek OV7670 v2.2 module. Both use a 12MHz clock. Other modules use a 24MHz clock. While this affects only the framerate selection parameters, some newer versions have a different schematic/pinout which need VSYNC signal to be inverted, these will not work with this setup.

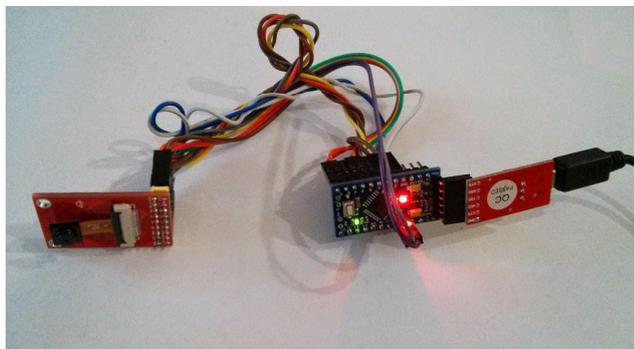
- Female to female jumper wires to connect the Arduino board to the CMOS module.

CONNECTIONS:

Connect the boards as follows:



You may edit the file IO_config.h of the arduino sketch if you want to make some changes, just pay attention on the VSYNC pin, that is tied to INT0 interruption.



SOFTWARE:

You can download from this repo the arduino sketch and the processing code to communicate with the arduino via USB:

<https://github.com/dasaki/arduision>

- ▶ 2011 (21)
- ▶ 2010 (11)

Labels

12.04 LTS 3d blob tracking 3d printing adapter adhesive tape aectir.com al422 algorithm andOSC Android android SDK Animata ardour 3 ardour3 arduino arduision arm ARToolKit ARToolKitPlus asus audio audio analysis AudioRecord augmented reality babelbots bbop bending binary boot buffer size c910 c920 camera cast casting Chip Select cinema circuit climberbot cmos cmu Color 75° color tracking computer vision computer vision cinema cooling cortex-a9 cropping cvcinema demo diy do it yourself draw drawing DSLR embedded emulator etching Euclidean Cluster Extraction exhibition experimental FFT fifo film filter forth Fourier Transform frame differencing framerate Friends With You gem.gem.pix_artoolkit gma500 GML GNU/linux golan levin GPS gspca haar finder hack hair iron hair straightener hand head pose head tracking heatpipe heatsink hot-plug impress infrared interactive installation IR light irobot irobot create java jewelry junkbot kinect laptop Lego Mindstorms lens library libre office lightpainting linux logitech macro madrid Malota Projects MalotaProjects medialab prado metal casting microfusion microphone minim mod module multitouch NITE nuigroup OCR ofxOpenNI ofxTesseract omap4 omap4430 omnivision open office open source openCV openframeworks openFrameworks openkinect OpenNI OSCeleton ov534 ov7670 ov7725 p48 packing tape pan tilt Pandaboard patch pcb pctl pd pdp_artkp pduino production pla plastic bag play play producciones pointcloud pointclouds polymorph pose estimation poulsbo power preamplifier Precise presentation press PrimeSense printed circuit board problem processing processing.org processor progress bar projection prototyping ps3 eye ps3eye puredata quad quad core Raspberry Pi real-time realtime realtime mapping recording reset RGBDtoolkit robot robotic robotics roomba RPi Samuel Beckett SD card serendipity servo shining crane shortfilm shutdown simpleopenni sketchup sketchyphysics solarbotic sound STUDIO for Creative Inquiry temperature tesseract therandomlab therandomlab.com thermal adhesive thermal paste thermography toner transfer tracking tracks triangulation u24e ubuntu udp unlimited upgrade US15W usb victorinox video vodemode wifi wishield xbee xlr

Subscribe To

- Posts
- Comments

The code will either stream a 80x60 pixels image, in different pixel depths, or detect the boundaries of the brightest/darkest blob within a threshold.

Edit: you get too much garbage in the processing console you may need to change the baudrate to match your system capabilities.

In the arduino sketch look for the line:

```
static const unsigned long _BAUDRATE = 500000;
```

And in the processing sketch, in the globalDefinitions.java tab:

```
public final static int BAUDRATE = 500000;
```

Try with substituting the 500000 for 115200.

NOTES ON THE ARDUINO SIDE:

Some people have added pullup resistors to the SDA and SCL lines and **disabled the internal pullups**. However, as I'm working with a 3.3v version, in this setup things worked straight away without these steps.

NOTES IN THE PROCESSING SIDE:

You will have to change the serial port name to match your system. In the processing code, look for the line:

```
serialPort = new Serial(this, "/dev/ttyUSB0", G_DEF.BAUDRATE);
```

And change the port name "/dev/ttyUSB0".

RESOURCES / LINKS:

I'll try that all the information published here is related only to 100% open hardware and software.

Two inspiring fully open hardware and software embedded computer vision projects:

AVRcam by John Orlando

One of the pioneers pushing the little microcontroller capabilities.

OpenMV by Ibrahim Abdelkader

google group: <https://groups.google.com/d/forum/openmvcam>

Currently in development, it will be an amazing and affordable platform with outstanding capabilities, and the only one in its class completely open.

Useful documentation

- <http://embeddedprogrammer.blogspot.com.es/2012/07/hacking-ov7670-camera-module-sccb-cheat.html>
- http://wiki.beyondlogic.org/index.php/OV7670_Camera_Module_with_AL422_FIFO_Theory_of_Operation
- http://nicolasfley.fast-page.org/?page_id=35
- <http://embeddedprogrammer.blogspot.com.es/2012/07/hacking-ov7670-camera-module-sccb-cheat.html>
- <http://embeddedprogrammer.blogspot.com.es/2013/01/demo-stm32f4-ov7670-qserialterm.html>
- <http://www.rpg.fi/desaster/blog/2012/05/07/ov7670-camera-sensor-success/>
- <http://wiki.jmoon.co/sensors/camera/ov7670/>
- <http://www.eepw.com.cn/article/113798.htm>
- <http://www.bot-thoughts.com/2010/04/gameboy-camera-prototyping.html>
- <http://sophiateam.undrgrnd.free.fr/microcontroller/camera/index.html>
- <http://www.cs.cmu.edu/~jbruce/cmvision/>

Source code related to microcontrollers and CMOS sensors/modules:

- <https://github.com/desaster/ov7670fifotest>

- <https://github.com/ComputerNerd?tab=repositories>
- <https://github.com/arndtjenssen/ov7670>
- <https://github.com/dalmirdasilva/ArduinoCamera>
- <http://www.suwa-koubou.jp/micom/NetCamera/ov7670.c>

Forum discussion threads about microcontrollers and CMOS sensors/modules

- <http://forum.arduino.cc/index.php?topic=159557.0>
- <http://forum.arduino.cc/index.php?topic=125767>
- <http://www.openedv.com/posts/list/23685.htm>
- <http://www.52rd.com/bbs/Dispbbbs.asp?BoardID=86&ID=182155>
- http://hamsterworks.co.nz/mediawiki/index.php/Zedboard_OV7670

Posted by David Sanz Kirbis 

 Recommend this on Google

Labels: [al422](#), [arduino](#), [arduision](#), [cmos](#), [computer vision](#), [fifo](#), [module](#), [omnivision](#), [ov7670](#), [ov7725](#), [processing](#), [processing.org](#)

13 comments:



Mangas e Animes October 3, 2014 at 5:34 AM

Hello ...
I copied all files and made the connections as pictured ...
and sent the .ino for Arduino ...
how do i see if it is working?

Reply

Replies



David Sanz Kirbis October 3, 2014 at 7:29 AM

Hi!

To see the stream in your computer through the USB port, you have to download the Processing environment for your platform from <https://processing.org/download/> and install it (see: <http://www.processing.org/tutorials/gettingstarted/>).

Then look for the folder "receive_stream" you got from the repo (https://github.com/dasaki/arduision/tree/master/arduision_01/processing/receive_stream)

and copy the entire folder inside your Processing sketchbook folder. Open the file receive_stream.pde with processing and run the sketch.

Once running press the space key to toggle the stream modes.

Whether you make it work or not, please give feedback! :)

Regards



Mangas e Animes October 6, 2014 at 5:04 AM

hello
I'm not finding what folder should I copy the java files

I thank



David Sanz Kirbis October 6, 2014 at 5:43 AM

The first time you run processing, it creates a folder called "sketchbook" in your computer (the place depends if you use Mac, windows or Linux). Inside the sketchbook folder is where you should copy the folder "receive_stream" and its contents. So you have the files receive_stream.pde, global_definitions.java, and simpleKalman.java inside the folder "sketchbook/receive_stream/".

Please read the [processing.org](http://processing.org/wiki) wiki and tutorials to inform yourself and learn how to use it.



Mangas e Animes October 6, 2014 at 6:44 AM

Could not find the folder ...
ran opened the screen, but that is so black ...
squeezed space and went through all modes and nothing



David Sanz Kirbis October 6, 2014 at 9:36 AM

What system are you using? Did you read the post?: "You will have to change the serial port name to match your system"

In the processing code, look for the line:
`serialPort = new Serial(this, "/dev/ttyUSB0", G_DEF.BAUDRATE);`

And change the port name "/dev/ttyUSB0" to your serial port...

try:

```
serialPort = new Serial(this, Serial.list()[0], G_DEF.BAUDRATE);
```

This kind of experiments are not a good way for a noob to start learning. As I mentioned at the beginning of the post:

"You are assumed to have some knowledge of the arduino platform and environment, and also the Processing.org development environment"



Wheberton Fialho October 6, 2014 at 10:02 AM

Yes, I did all the steps mentioned ...
arduino communicating this with the processing ..
so that it appears to envez the imagens this all black



Wheberton Fialho October 6, 2014 at 10:20 AM

Yes, I did all the steps mentioned ...
arduino communicating this with the processing ..
so that it appears to envez the imagens this all black



David Sanz Kirbis October 6, 2014 at 1:14 PM

Hmmm... What arduino are you using? Did you see in the processing console these messages?:

```
"Initializing sensor..."  
"inited OK, sensor PID: "
```



Wheberton Fialho October 6, 2014 at 1:59 PM

Arduino Nano... Atmega328p... yes the message appeared...
aparece em cima tambem a contagem de FPS, mas a tela fica preta



David Sanz Kirbis October 6, 2014 at 2:14 PM

Arduino Nano, according to the official specifications (<http://arduino.cc/en/Main/arduinoBoardNano>) works at 5v... As said in the post, the sensor can stand a maximum voltage of 3.3v... you may have destroyed the sensor...

Reply



Wheberton Fialho October 6, 2014 at 2:32 PM

can occur even connecting pin 3.3v

[Reply](#)



David Sanz Kirbis October 6, 2014 at 2:50 PM

Yes, pin 3.3v is just the power line, but the logic levels that go to the sensor (SDA, SCL and VSYNC) have to be also 3.3v or lower. A "3.3v arduino" means that the logic levels are 3.3v not 5v.

You can still try to put resistors as explained here: http://nicolasfley.fast-page.org/wp-content/uploads/2013/09/schema_arduino_FIFO_OV7670.png and see if your sensor is still alive.

[Reply](#)

Enter your comment...

Comment as:

[Publish](#)

[Preview](#)

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by [Blogger](#).