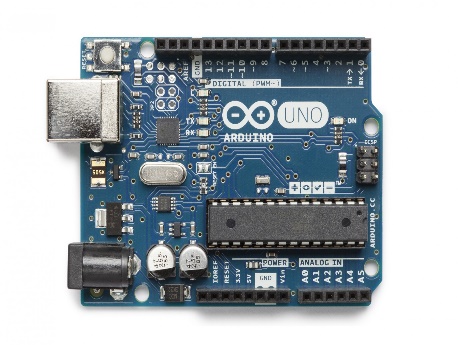
COMPLETE EXPLANATION

In this project, you needed these parts

1.[Arduino Uno R3](http://s.click.aliexpress.com/e/jm2vJaM) (you can also use the other version of Arduino)



2. OV7670 camera sensor module [without](http://s.click.aliexpress.com/e/jyJMJMn) FIFO RAM 1pc



3.Arduino IDE ( you can download it from [here](https://www.arduino.cc/en/Main/Software) ) and output software (you can download it [here](https://drive.google.com/file/d/1d7pbPzsq8qYXYxAN-j2S7LDBFwa94iYg/view?usp=sharing). if it will not work - download and install microsoft.net 3.5 framework)

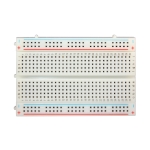
4.Jumper cables F-M, M-M



5. Resistors 4 pcs (2 pcs 10 KOhm, 2 pcs 4.7 KOhm)



6.[Breadboard](http://s.click.aliexpress.com/e/7mAAe62) half size or small size 1 pc



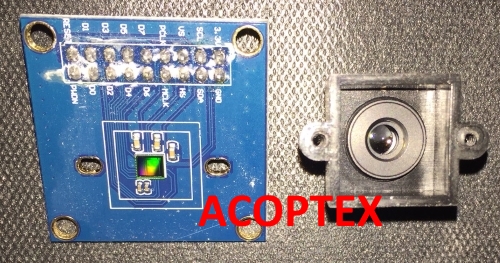
General

We will learn how to connect OV7670 camera sensor module to Arduino board and use it.

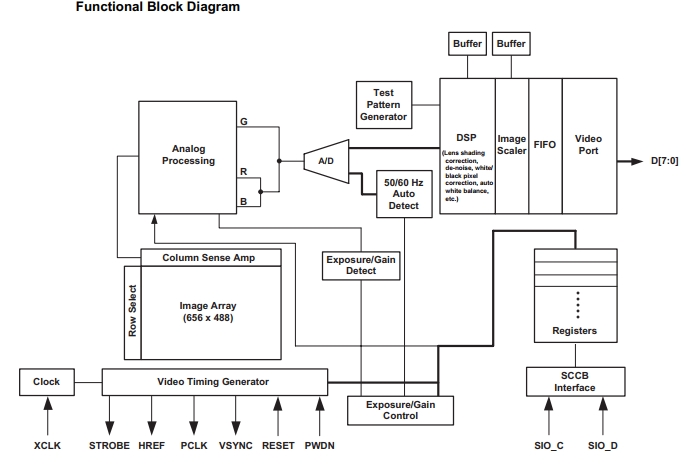
Understanding the OV7670 camera sensor module

0V7670 camera module can be used to take a pictures only. No video streaming.

The OV7670/OV7171 CAMERACHIPTM is a low voltage CMOS image sensor that provides the full functionality of a single-chip VGA camera and image processor in a small footprint package. The OV7670/OV7171 provides full-frame, sub-sampled or windowed 8-bit images in a wide range of formats, controlled through the Serial Camera Control Bus (SCCB) interface. This product has an image array capable of operating at up to 30 frames per second (fps) in VGA 640x480 resolution with complete user control over image quality, formatting and output data transfer. Allrequired image processing functions, including exposure control, gamma, white balance, colorsaturation, hue control and more, are also programmable through the SCCB interface. In addition, OmniVision CAMERACHIPs use proprietary sensor technology to improve image quality byreducing or eliminating common lighting/electrical sources of image contamination, such as fixed pattern noise (FPN), smearing, blooming, etc., to produce a clean, fully stable color image.



The OV7670 camera module is a low cost 0.3 mega pixel CMOS color camera module.



Features:

High sensitivity for low-light operation

Low operating voltage for embedded portable apps

Standard SCCB interface compatible with I2C interface

Supports VGA, CIF, and resolutions lower than CIF for RGB (GRB 4:2:2, RGB565/555/444), YUV (4:2:2) and YCbCr (4:2:2) formats

VarioPixel® method for sub-sampling

Automatic image control functions including: Automatic Exposure Control (AEC), Automatic Gain Control (AGC), Automatic White Balance (AWB), Automatic Band Filter (ABF), and Automatic Black-Level Calibration (ABLC)

Image quality controls including color saturation, hue, gamma, sharpness (edge enhancement), and anti-blooming

ISP includes noise reduction and defect correction

Supports LED and flash strobe mode

Supports scaling

Lens shading correction

Flicker (50/60 Hz) auto detection

Saturation level auto adjust (UV adjust)

Edge enhancement level auto adjust

De-noise level auto adjust

Application:

Cellular and picture phones

PDAs

Toys

PC multimedia

Digital still cameras

Other battery-powered products

Can be used in Arduino, Maple, ChipKit, STM32, ARM, DSP, FPGA platforms

Specification:

Array element (resolution): 640x480 VGA

Lens size: 1/6"

Power supply Digital core: 1.8VDC+/- 10%

Power supply Analog: 2.45 V to 3V

Power supply I/O: 1.7V to 3V

Power consumption working : 60 mw / 15fps VGA YUV

Power requirements Active: TBD

Power requirements Standby:<20μA

Temperature Operation: from - 30° to 70°C

Temperature Stable image : from 0° to 50℃

Output format (8 bit) : RawRGB, RGB(GRB4: 2:2, RGB565/555/444), YUV(4:2:2) and YCbCr(4:2:2)

Chief ray angle : 24 degrees

Max image transfer rate (frame rate) : 30fps VGA

Sensitivity : 1.3 V / (Lux - SEC)

S/N ratio : 40 dB

Dynamic range : TBD

Scan mode: Progressive

Electronic exposure: up to 510:1 (for selected fps)

Pixel size : 3.6 μm x 3.6 μm

Dark current : 12 mV/s at 60 ℃

Well capacity : 17 K e

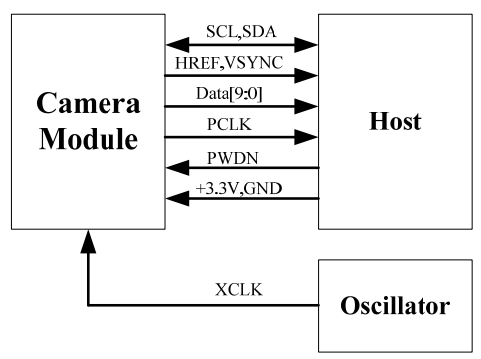
Image area: 2.36 mm x 1.76 mm

Package dimensions: 3785 μm x 4235 μm

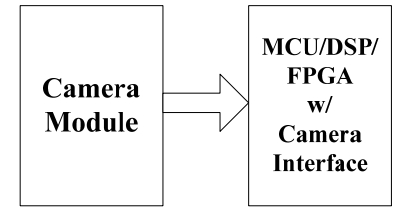
You can find the datasheet [here](http://acoptex.com/uploads/ov7670_cmos_camera_module_revc_ds.pdf) and [here](http://acoptex.com/uploads/OV7670.pdf). Schematic is [here](http://acoptex.com/uploads/OV7670SCH.pdf).

Omnivision Serial Camera Control Bus functional specification can be found [here](http://acoptex.com/uploads/SCCB.pdf).

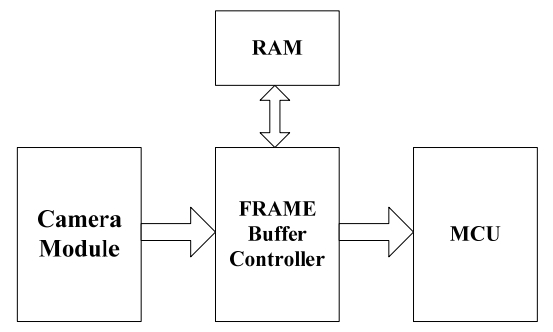
A basic of camera based system:



The camera module is powered from a single +3.3V power supply. An external oscillator provide the clock source for camera module XCLK pin. With proper configuration to the camera internal registers via I2C bus, then the camera supply pixel clock (PCLK) and camera data (Data[9:0]) back to the host with synchronize signal like HREF and VSYNC. The host may have integrate camera interface like STM32F2 or STM32F4 series MCUs, or ARM9/11 which has dedicate camera port, and DPS like TI TMS320DM series, as well as FPGAs that user can design special logic for camera application. The typical connection between these system and camera module would show like following diagram.



For the host that doesn’t have a dedicate camera interface, additional hardware is needed. User need to buffer a entire frame before read them out with low speed MCUs. For example OV7670 camera module is an additional hardware that can be connected to Arduino UNO/Mega board and user can take a photo or something like that easily. The following diagram show the system without dedicate camera interface.



Signals and connections of the OV7670 module

3.3V (or VDD or DOVDD) - digital power supply for I/O (1.7 - 3V)

SCL (or SDIOC) - SCCB serial interface clock input

VS (or VSYNC) - vertical synchronization output (Active High: Frame Valid; indicates active frame)

PCLK (or PCK) - pixel clock output from sensor

MCLK (or XCLK or XCK) - master/system clock input to sensor

STROBE - LED/strobe control output

D7-D0 - YUV/RGB video component  output (pixel output)

RESET - Clears all registers and resets them to their default values (normal mode or reset mode)

PWDN - Power down mode selection (normal mode or power down mode)

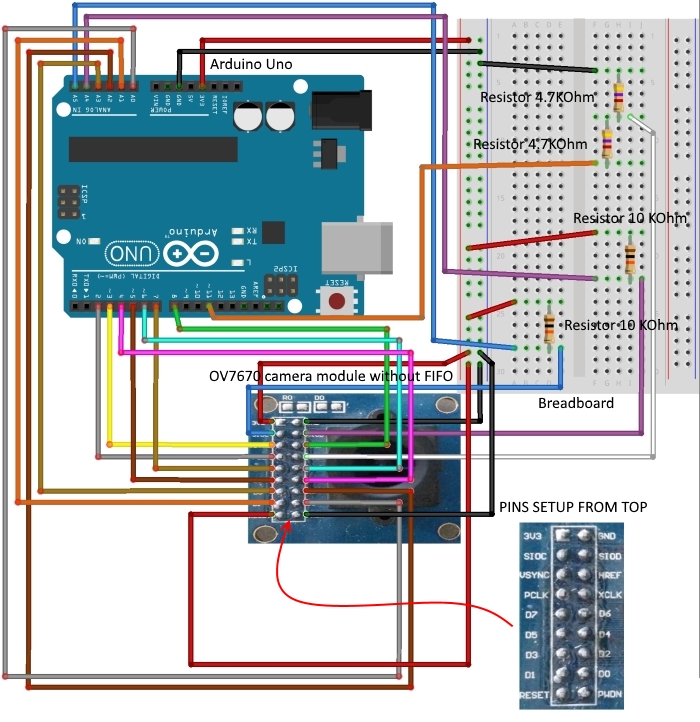
HS (or HREF) - horizontal synchronization output (Active High: Line/Data Valid; indicates active pixels)

SDA (or SDIOD) - SCCB serial interface data I/O

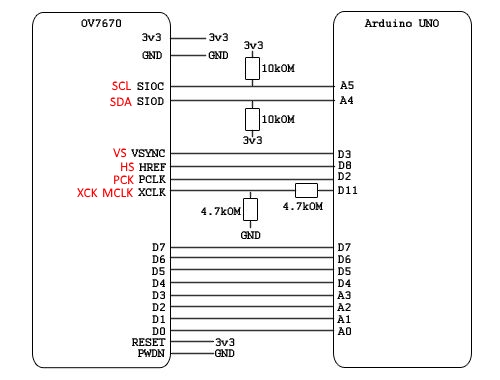
GND (or DOGND) - digital ground. Connected to Arduino board GND pin.

Wiring

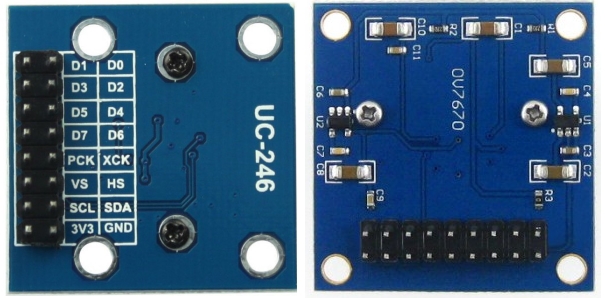
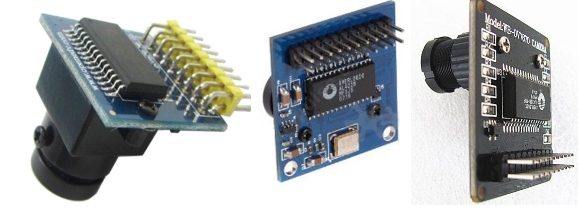
The following picture shows the needed connections with the Arduino Uno



Wiring schematic:



Step by Step instruction

Identify Your Camera - check the back side: a) OV7670 camera module without FIFO:b) OV7670 camera module with FIFO:

Do wiring.

Open Arduino IDE.

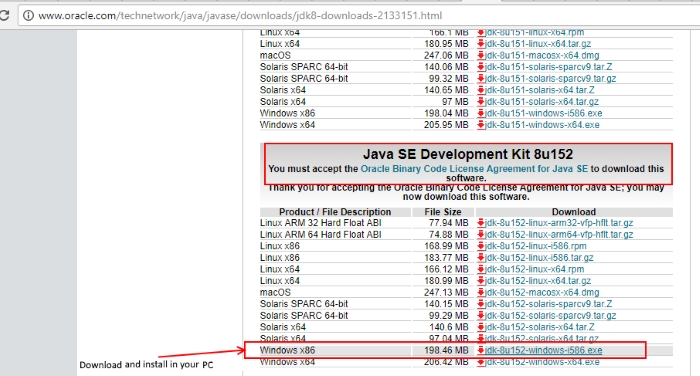
Plug your Adruino Uno board into your PC and select the correct board and com port

Modify the sketch.

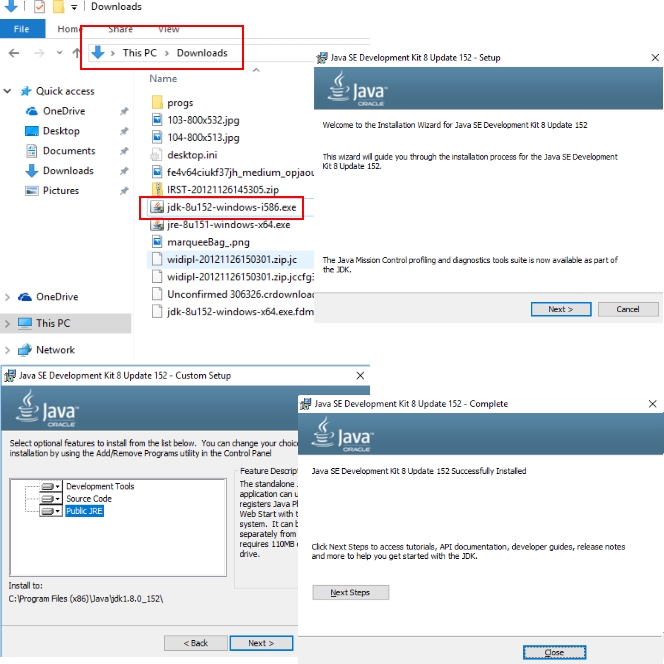
Verify and upload the the sketch to your Adruino Uno

Verify and upload the the sketch to your Adruino Uno

You have to download and install Java JDK in your PC. You will need Java JDK for Windows x86 (32 bit system). We have used PC with Windows 10 64 Bit software for this project. Don't install the 64 bit version even if your Windows system is 64 bit.

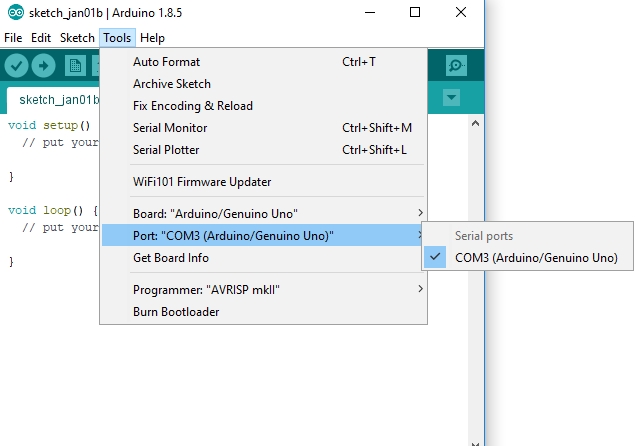


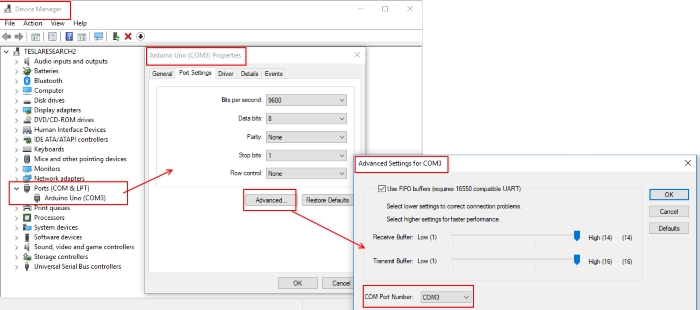
This jdk-8u152-windows-i586.exe file is quite big so it will take some time to download it.



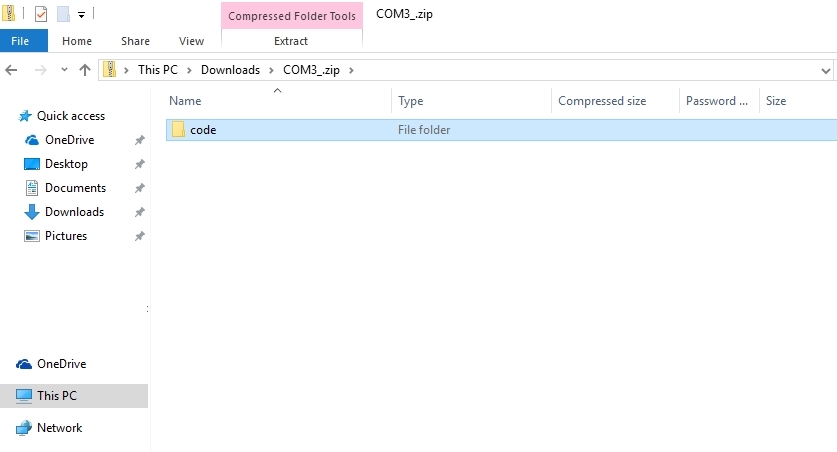
The files were installed in C:\Program Files (x86)\Java\jdk1.8.0\_152\ folder successfully.

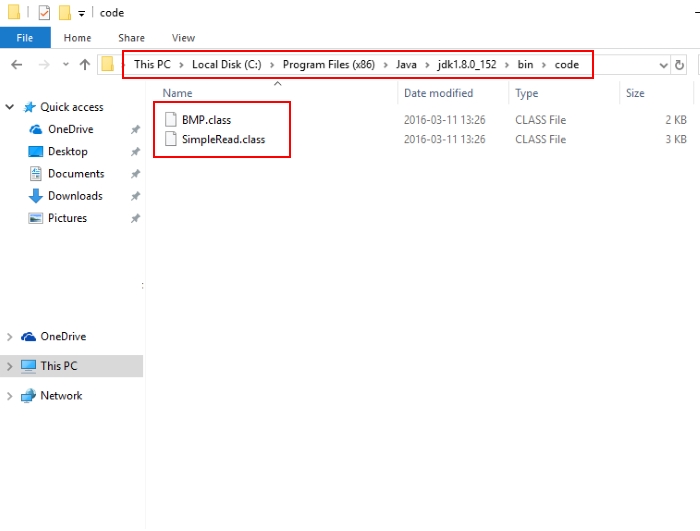
Find out which USB port your Arduino board connected to (it can be COM1, COM2,COM3,COM4,COM5). As you can see it is COM3 in our case. You can check this in Device manager of your PC or in Arduino IDE program:



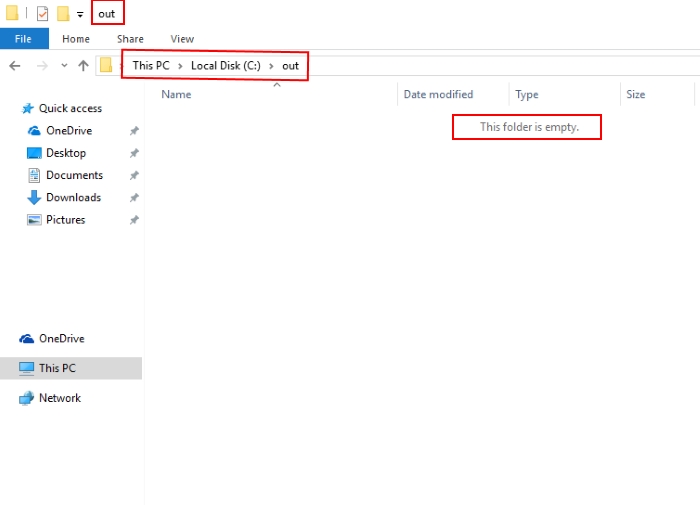


Depending on which port number your Arduino board connected to, download the corresponding file:  [COM1\_.zip](http://acoptex.com/uploads/COM1_.zip), [COM2\_.zip](http://acoptex.com/uploads/COM2_.zip), [COM3\_.zip](http://acoptex.com/uploads/COM3_.zip), [COM4\_.zip](http://acoptex.com/uploads/COM4_.zip),  [COM5\_.zip](http://acoptex.com/uploads/COM5_.zip). For example, we had COM3 port number so downloaded COM3\_.zip file. Unzip your file, copy the 'code' folder and paste it to C:\Program Files(x86)\Java\jdk1.8.0\_152\bin\ (it might ask for adm. permission).

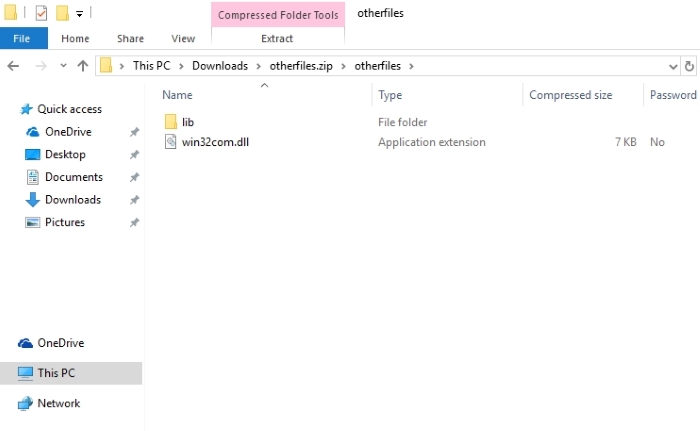


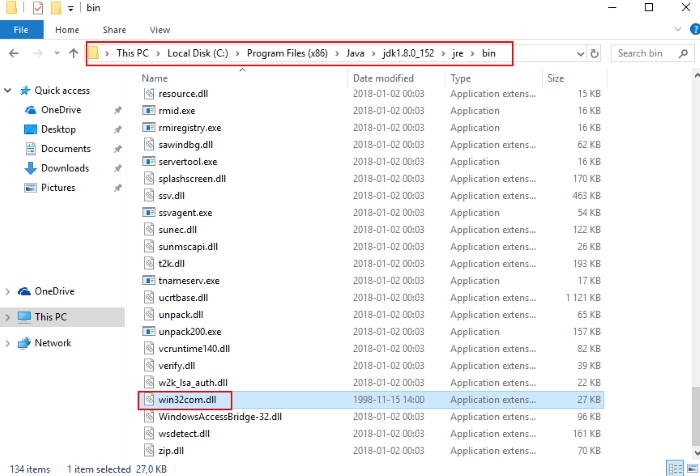


Create a new folder on your C drive called out, like C:\out. This will be your camera module sensor's taken picture's destination.

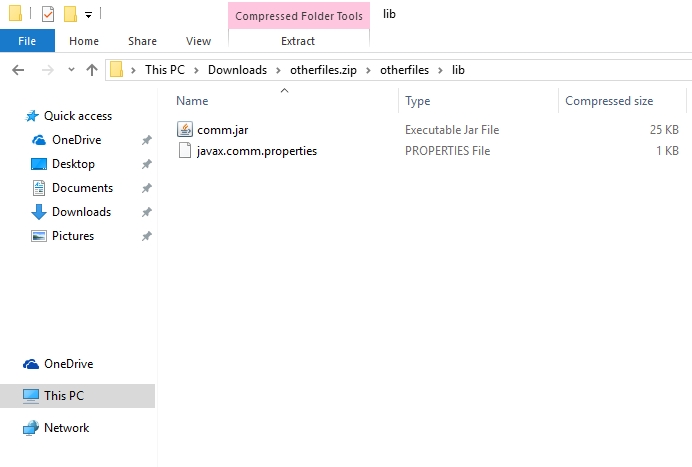


Download and unzip the [otherfiles.zip](http://acoptex.com/uploads/otherfiles.zip) file.You will find lib folder and win32com.dll file inside. Copy win32com.dll file to the C:\Program Files(x86)\Java\jdk1.8.0\_152\jre\bin\ folder (it may ask for administrator permission, just proceed anyway).

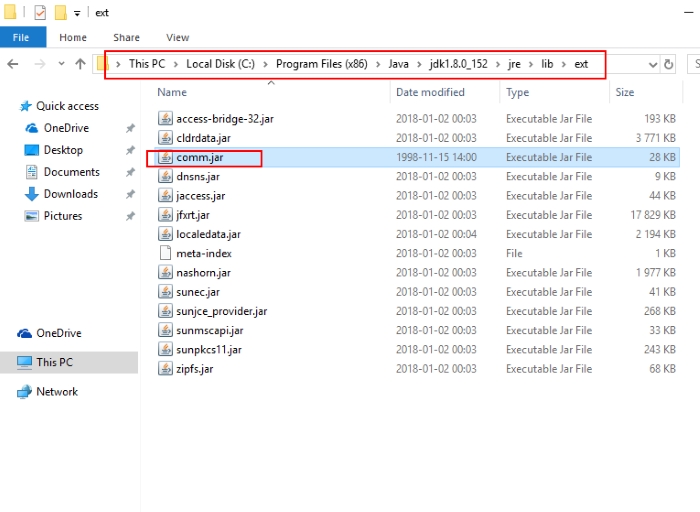




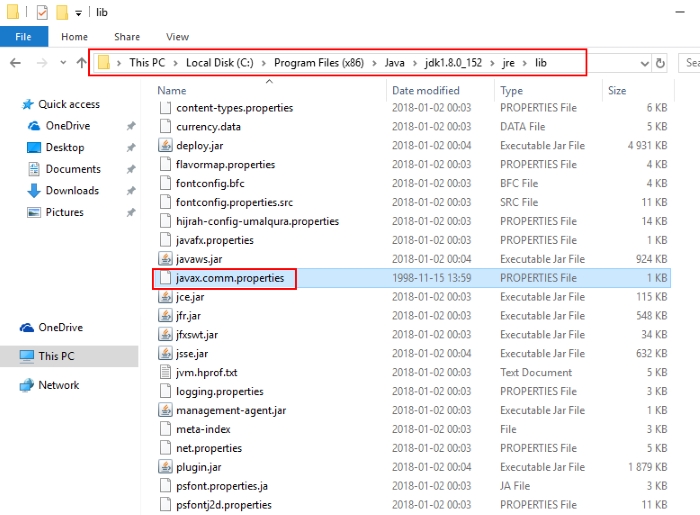
Open the lib folder in otherfiles.zip and you will see the comm.jar and javax.comm.properties files inside.



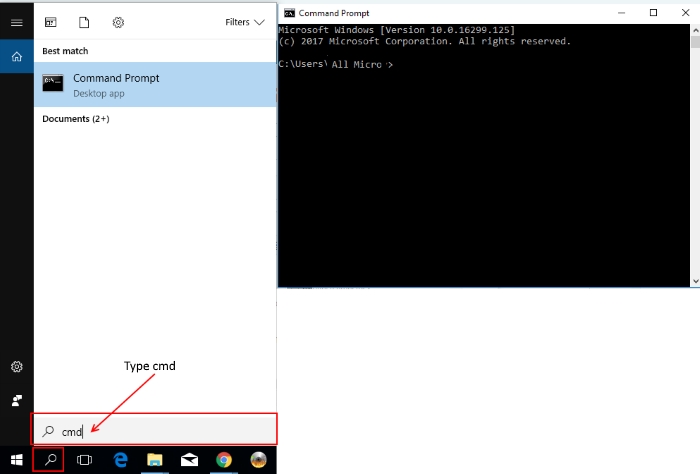
Copy and paste the comm.jar file to C:\Program Files(x86)\Java\jdk1.8.0\_152\jre\lib\ext\ folder (it may ask for administrator permission, just proceed anyway).

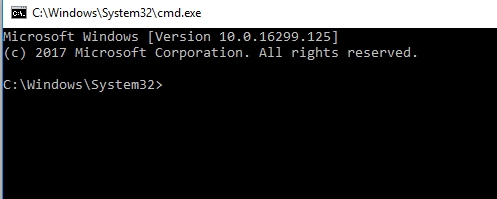


Copy and paste the javax.comm.properties file to  C:\Program Files(x86)\Java\jdk1.8.0\_152\jre\lib\ folder (it may ask for administrator permission, just proceed anyway).



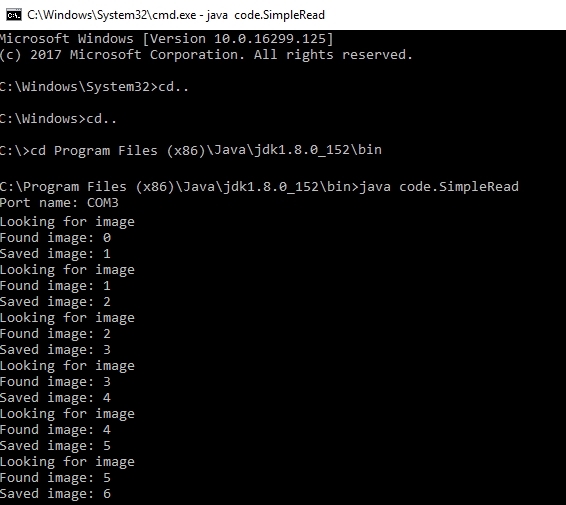
Open Command Prompt. There are two ways to do it - Search Windows ->Type cmd -> or find the cmd.exe file (C:\Windows\System32\cmd.exe).



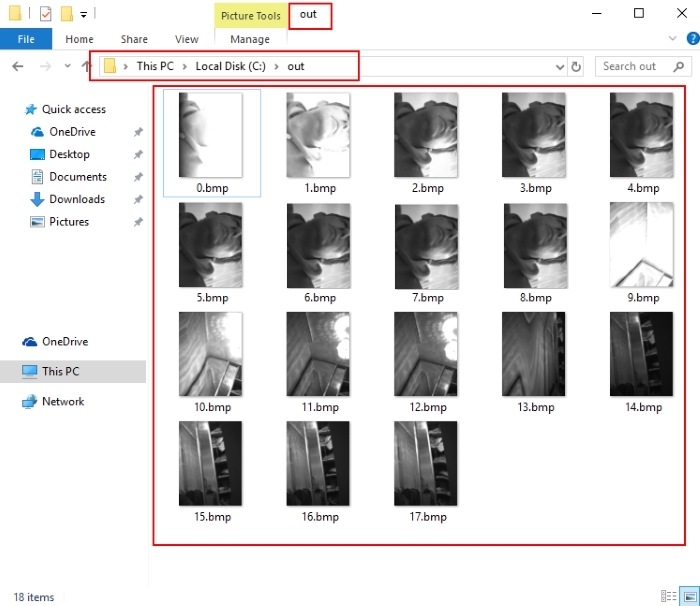


Go back to C:\ , then open the folder where the code is by writing its address. For example, our PC have we need to type cd C:\Program Files (x86)\Java\jdk1.8.0\_152\bin and press Enter button. You will see C:\Program Files (x86)\Java\jdk1.8.0\_152\bin>.

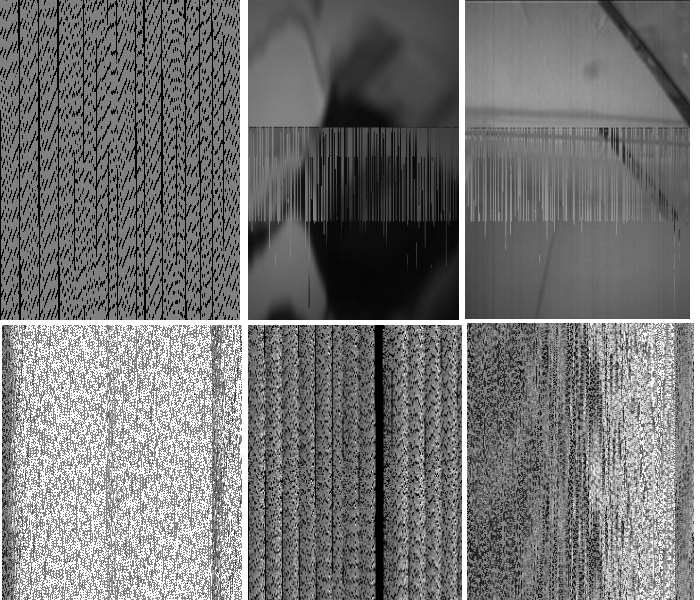
Type java code.SimpleRead and press Enter button. You will see C:\Program Files (x86)\Java\jdk1.8.0\_152\bin>java code.SimpleRead



You can see pictures taken from your OV7670 camera module in C:\out\ folder.



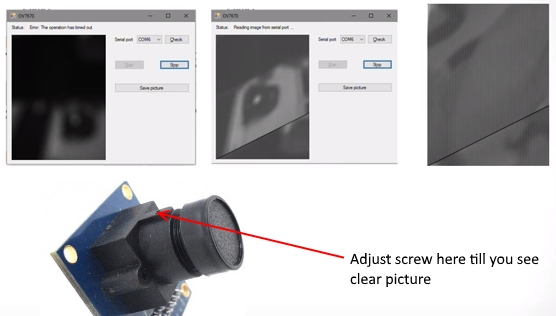
Troubleshooting



If you run Command Prompt but it cannot find images: a) check out the wiring (90% problem because of the wrong wiring); b)try switching the USB of the arduino from a USB2.0 to a USB3.0 (blue socket) or vice versa; c) check if you downloaded, extracted/copied correct files to all folders.

If you receive scrambles pictures, you need to modify the sketch. Change the number marked bold in the line wrReg(0x11, 12); try changing the 12 to 9/10/11/13. Upload modified sketch to the Arduino board and run the cmd again.

If you are not able to view a clear image and your image is very distrupted like the below. Simply adjust the screw of your camera lens and view different images till they seem to be clear



Output software

You can download it [here](https://drive.google.com/file/d/1d7pbPzsq8qYXYxAN-j2S7LDBFwa94iYg/view?usp=sharing). If it will not work - download and install [microsoft.net 3.5](https://www.youtube.com/redirect?q=https%3A%2F%2Fwww.microsoft.com%2Fen-in%2Fdownload%2Fdetails.aspx%3Fid%3D21&v=8V-XBA2qAKY&redir_token=pq_DsRNpUGIesFVGCmFb6GN_4UJ8MTUxNTA5ODM3NkAxNTE1MDExOTc2&event=video_description) framework. You need to have your Arduino board connected to your PC in order to use this software. When you save picture - give a proper name for example.example.bmp and press Enter button.



Code

1. [Original code](http://acoptex.com/uploads/originalOV7670code.ino) from Computer Nerd. You can find out more about it [here](https://github.com/ComputerNerd/ov7670-no-ram-arduino-uno).

2. [Modified code](http://acoptex.com/uploads/modifiedOV7670code.ino) from [here](http://privateblog.info/arduino-uno-i-kamera-ov7670-primer-ispolzovaniya/). It is the modified original code from Computer Nerd. A little about the code:

a) Setup of PWM (pulse width modulation) to get 8 mHz on digital pin 11 of Arduino Uno board:

DDRB |= (1 << 3); //pin 11

ASSR &= ~(\_BV(EXCLK) | \_BV(AS2));

TCCR2A = (1 << COM2A0) | (1 << WGM21) | (1 << WGM20);

TCCR2B = (1 << WGM22) | (1 << CS20);

OCR2A = 0;//(F\_CPU)/(2\*(X+1))

DDRC &= ~15; //low d0-d3 camera

DDRD &= ~252; //d7-d4 and interrupt pins

b) Setup of I2C interface:

TWSR &= ~3;//disable prescaler for TWI

TWBR = 72;//set to 100khz

c) Setup of RS232:

UBRR0H = 0;

UBRR0L = 1;//0 = 2M baud rate. 1 = 1M baud. 3 = 0.5M. 7 = 250k 207 is 9600 baud rate.

UCSR0A |= 2;//double speed aysnc

UCSR0B = (1 << RXEN0) | (1 << TXEN0);//Enable receiver and transmitter

UCSR0C = 6;//async 1 stop bit 8bit char no parity bits

d) Camera setup:

wrReg(0x12, 0x80);

\_delay\_ms(100);

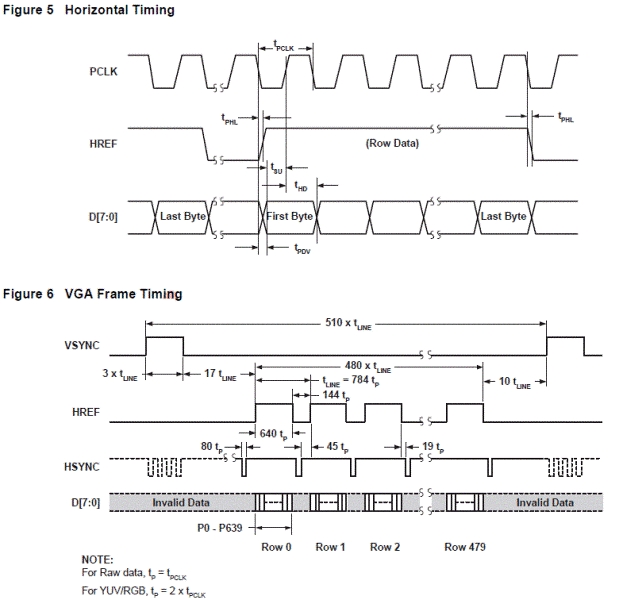
wrSensorRegs8\_8(ov7670\_default\_regs);

wrReg(REG\_COM10, 32);//PCLK does not toggle on HBLANK.

e) Picture reception. The operating mode of the camera image transmission was set to YUV. In this case, each pixel is encoded with two bytes. The first byte encodes the gradation of gray, the second - color-difference component.

http://acoptex.com/uploads/OV7670codeexplained0.jpg

Our task is to get at least a black and white image, so the second byte can be discarded. Next, we need to bring an oscillogram from which it becomes clear what signals we expect with high and low values.



StringPgm(PSTR("\*RDY\*"));

//VSYNC

while (!(PIND & 8));//wait for high

while ((PIND & 8));//wait for low

y = hg;

while (y--){

   x = wg;

   while (x--){

      //PCLK

      while ((PIND & 4));//wait for low

      UDR0 = (PINC & 15) | (PIND & 240);

      while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit

      while (!(PIND & 4));//wait for high

      while ((PIND & 4));//wait for low

      while (!(PIND & 4));//wait for high

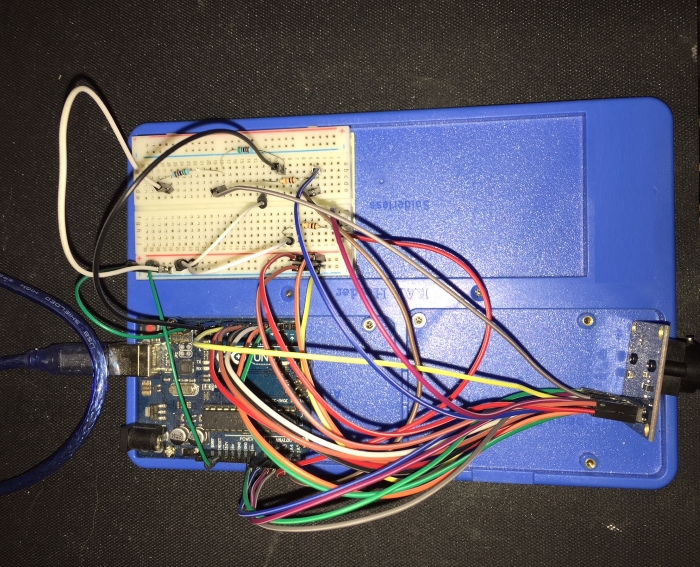
   }

}

\_delay\_ms(100);

Summary

We learnt how to connect OV7670 camera module to Arduino board and use it.



Library

No library required for this project

Sketch

See attachment on the begining of this project