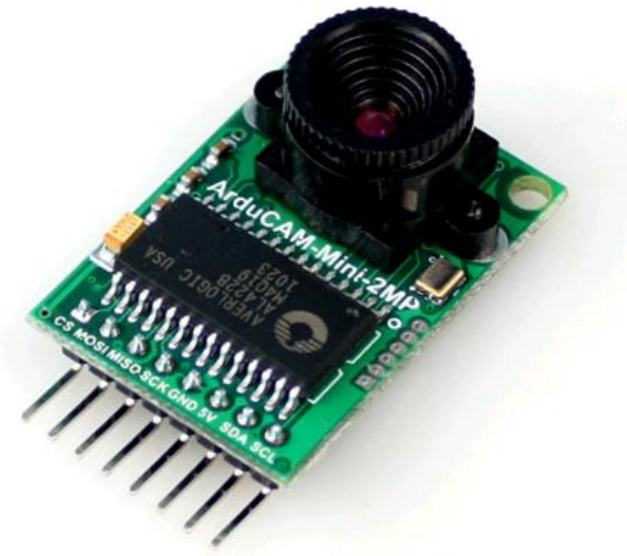
Arducam-Mini camera + ESP8266 Wi-Fi uploading pictures to a webserver using http-post file methods

For some years Adafruit makes TTL cameras that can be interfaced with Arduino, but they are quite expensive and image quality was 640x480. Recently Arducam-Mini appeared that is an amazing work. Altera CPLD chip is programmed to grab image from CMOS sensor, convert it to a jpeg and store in the RAM chip. Image size can be 1600x1200 for 2 Mpixel camera.

Arducam-Mini can be bought through Ebay. Lee has sold 400 pieces so there is a growing community.



Arducam has a well-documented website http://www.arducam.com/ and fast responces to customer questions.

Downloaded Arducam library from github on 05 March 2016. Used Arduino IDE 1.6.5.
It took some time to figure out that the library should be placed in directory arduino/libraries/ArduCAM.
If the arduino-master directory is placed under the libraries folder the compiler can not find *.h files.

Arducam-Mini + Arduino-Nano

First tried an example coming with Arducam library with Arduino streaming to Windows PC:
"**ArduCAM_Mini_OV2640_Video_Streaming**".

1. First issue was that there is no circuit diagram on the webpage. I wanted to use Arduino-Nano instead of Arduino-Uno-Rev3. It took a while to figure that SDA=A4 and SCL=A5. Arducam-Mini can run from 5V.

2. Second issue Serial.begin(921600); did not work with Arduino Serial Monitor window.
   Changed speed to 115200.

3. Third issue was that Arducam.exe downloaded from the Arducam webpage:
   http://www.arducam.com/downloads/demo/arducam_host/ArduCAM_host.zip
   did not support speed of 900kbps. Apparently that the program was not the latest version and one should use the program included with the github library example.

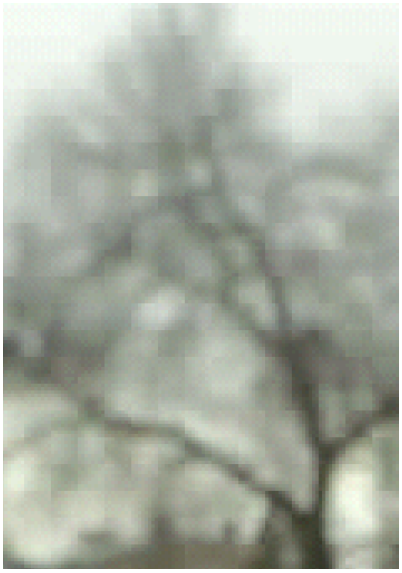
Link to a PC is useful to check that the camera module is working and that wiring is correct.
It also allows to adjust camera focus for best sharpness at a chosen distance.

When successful you should get picture on your PC screen like this:
https://www.youtube.com/watch?feature=player_embedded&v=MaOrhi6pnQk

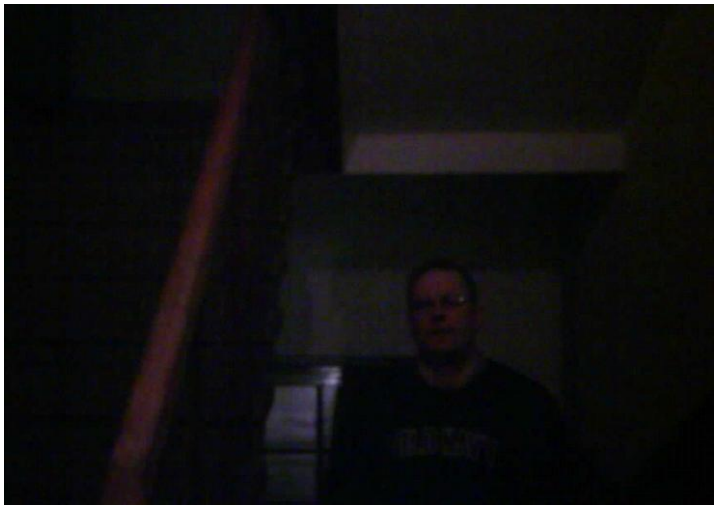
Arducam ESP8266 by Johan Kanflo runs RTOS and needs unix PC to compile. It was featured in Hackaday in 2016:
http://hackaday.com/2016/01/24/truly-versatile-esp8266-wifi-webcam-platform/

Image quality what I get

Arducam image quality is not astonishing. Below is picture at 1024x768 resolution. I can see brances of a tree by looking without camera. On the image they appear smeared out. It could be due to lens focus misadjustment. By zooming in can see that this is also due to jpeg averaging. 1024x768 picture size is only 20…40 kB. For a good quality photo size should be ca 150 kB. Unfortunately, there is no register to control jpeg compression quality.



Tested Arducam in low-light conditions – stairways illuminated by a 7W fluorescent lamp. One sees camera dark noise. As exposition is long, there is motional blur of moving objects . At same conditions Samsung Galaxy tab camera image was not darker because it has even smaller lens opening . It could help to use a larger diameter lens to collect more light like used in traditional CCD security cameras. NIR illumination is actually needed!



Some years ago I bought a 15$ keychain camera saving to SD and soldered Arduino to the buttons to automate picture taking for timelapse and security:
http://www.instructables.com/id/Keychain-camera-with-PIR-motion-detector-controlle/

Arducam-Mini controlled by ESP8266 Wi-Fi

Next tried the example with ESP8266 module ESP-12E file:
"**ArduCAM_Mini_OV2640_Capture**"
Soldered jumper wires to the needed pins of ESP module and later when everything worked fixed with some hot glue. Used 5V to 3.3 power Lipo+buck regulator board from Ebay.

ESP12-E module:
CS=GPIO16 MOSI=GPIO13 MISO=GPIO12 SCK=GPIO14 GND&GPIO15 VCC&EN=3.3V
SDA=GPIO4 SCL=GPIO5 GPIO0=button GPIO2=BlueLED

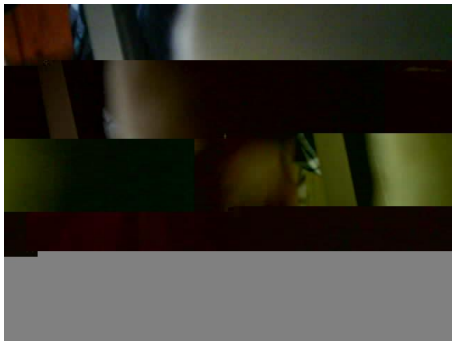
1. Serial debugging gave message Can't find OV2640 module! This was because of wrong number :
   if ((vid != 0x26) || (pid != 0x41))
   in websocket example camera id number is 0x42. Camera was found after changing to 0x42.

2. while(1); caused reset after a few seconds, as probably, the ESP8266 had no time to do other internal things.

3. In the program, I guess, it should also be nr 16 in the comment line.
   const int CS = 16; // set GPIO15 as the slave select

There is an easy way to capture a picture without the demonstration html: http://192.168.1.140/capture
Resolution can be changed with numbers between 0 and 8 in such a way: http://192.168.1.140/?ql=3

I experienced the problem with images chopped into parts described in this issue with the broken and mixed parts of the image. People report broken images. https://github.com/ArduCAM/Arduino/issues/21



Changing of buffer size to 2048 helped, but sometimes pictures still were broken. Changing to 1024 solved.

```
static const size_t bufferSize = 1024;
```

Arducam mini + ESP8266 Wi-Fi upload to server

Previous example showed that webserver that displays a photo on request. But often one would like camera to send photos to server automatically, for example, in security applications when motion detection has appeared.

I would like to make a PIR activated alarm camera automatically uploading a picture to a server.
I am a big fan of MobileWebcam:
http://forum.xda-developers.com/showthread.php?t=950933
On a server is a mobilewebcam.php script that receives the picture file and puts it into a directory with a time tag. My webcams ar at http://asi.lv/alnis
Form Linux picture upload line looks like that:
curl -F userfile=@/tmp/current.jpg asi.lv/alnis/webcammob1/mobilewebcam.php

Php upload is more convenient than FTP upload as the file name can be automatically prepared on the server side. Nice example is about the TTL camera uploading picture to server.
https://www.openhomeautomation.net/wireless-camera/

Using the example by *openhomeautomation* I have managed to adapt http POST file on ESP8266. Sketch is attached to this post named "ArduCAM_Mini_OV2640_Capture_uploadToServer".

File upload with http POST has some pecularities. One needs to calculate length of data sent. In pronciple one can send several files in same post and because of that each file has to be marked by so called boundary code for example „ AaB03x—„. One can search interet for this code and find several people posting their experiences. Several examples on uploading file to server Arduino and CC3200 did not work I guess because of header length calculation. String variables in Arduino should not be too long or they get cut. Image data are stored in Arducam-Mini RAM chip as a ready to use file and one can send it away byte for byte or store into SD card.

```
String start_request = ""; String end_request = "";

start_request = start_request +
  "\n--AaB03x\n" +
  "Content-Disposition: form-data; name=\"userfile\"; filename=\"CAM.JPG\"\n" +
  "Content-Transfer-Encoding: binary\n\n";

end_request = end_request + "\n--AaB03x--\n";



uint16_t full_length;
full_length = start_request.length() + len + end_request.length();

client.println("POST /alnis/webcammob1/mobilewebcam.php HTTP/1.1");
client.println("Host: asi.lv");
client.println("Content-Type: multipart/form-data; boundary=AaB03x");
client.print("Content-Length: "); client.println(full_length);
client.print(start_request);
```

```
// Read image data from Arducam mini and send away to internet
  static const size_t bufferSize = 1024; // original value 4096 caused split pictures
  static uint8_t buffer[bufferSize] = {0xFF};

  while (len) {
      size_t will_copy = (len < bufferSize) ? len : bufferSize;
      SPI.transferBytes(&buffer[0], &buffer[0], will_copy);
      if (!client.connected()) break;
      client.write(&buffer[0], will_copy);
      len -= will_copy;      }
     client.println(end_request);
     myCAM.CS_HIGH(); digitalWrite(led, HIGH);
}

  client.println(end_request);
```

Below is a transcript from Arduino IDE serial communication monitor.

reset
OV2640 detected.
Connecting to AP specified during programming
.....
WiFi connected IP address: 192.168.1.140

CAM captured picture length in bytes: 20480

POST /alnis/webcammob1/mobilewebcam.php HTTP/1.1
Host: asi.lv
Content-Type: multipart/form-data; boundary=AaB03x
Content-Length: 20605

--AaB03x
Content-Disposition: form-data; name="userfile"; filename="CAM.JPG"
Content-Transfer-Encoding: binary

Here are sent picture data

--AaB03x--

HTTP/1.1 200 OK
Date: Tue, 08 Mar 2016 19:28:23 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.2
X-Powered-By: PHP/7.0.2
Content-Length: 80
Content-Type: text/html; charset=UTF-8

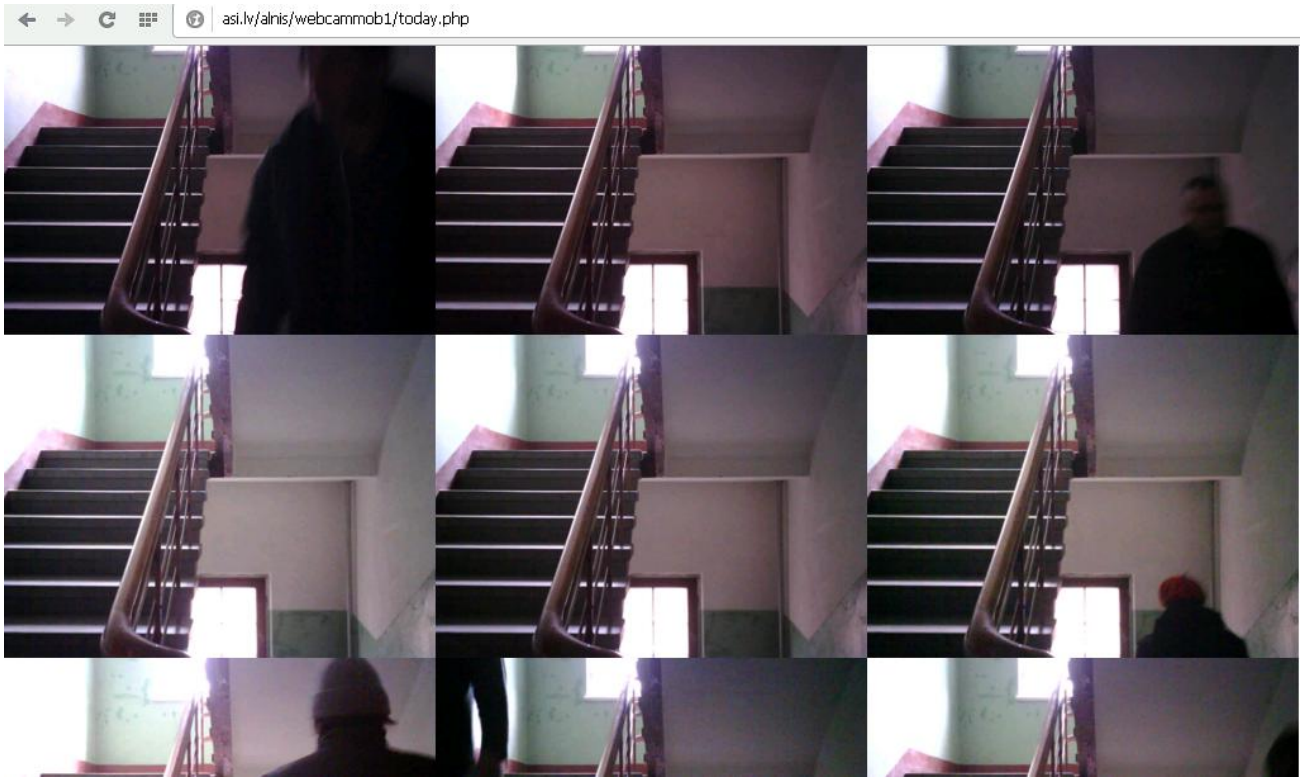
Uploading CAM.JPG Ok! Directory already exists.
saved 2016.03.08_20:28

Sleeping ...

On the server side is a php file that generates file name with the actual date and stores into a created directory. Copy the folder mobilewebcam1 to your server, or use WAMP for Windows, or my server.

```
<?php
// mobilewebcam.php  Takes caqre about uploading files to server using http POST.
$uploadfile = "";
echo "Uploading ";
echo $_FILES["userfile"]["name"];
if(strlen(basename($_FILES["userfile"]["name"])) > 0)

{
  $uploadfile = basename($_FILES["userfile"]["name"]);
  if(move_uploaded_file($_FILES["userfile"]["tmp_name"], $uploadfile))
  {
    @chmod($uploadfile,0777); echo " Ok! ";

$datum = mktime(date('H')+0, date('i'), date('s'), date('m'), date('d'), date('y'));

if (file_exists("old/".date('Y_m_d', $datum) )) {
  print("Directory already exists.\n");
  } else {
  mkdir("old/".date('Y_m_d', $datum));
  copy("index1.php","old/".date('Y_m_d', $datum)."/index.php");
  print("Directory creating.\n");
  }

echo "saved ";
copy($uploadfile,"old/".date('Y_m_d', $datum)."/".date('Y.m.d_H-i-s', $datum).".jpg");
  }
  else echo " Error copying!";
}
echo date('Y.m.d_H:i', $datum);
?>
```

Below is how pictures look in webserver. Html page is generated by index.php script that shows all the pictures in the folder.



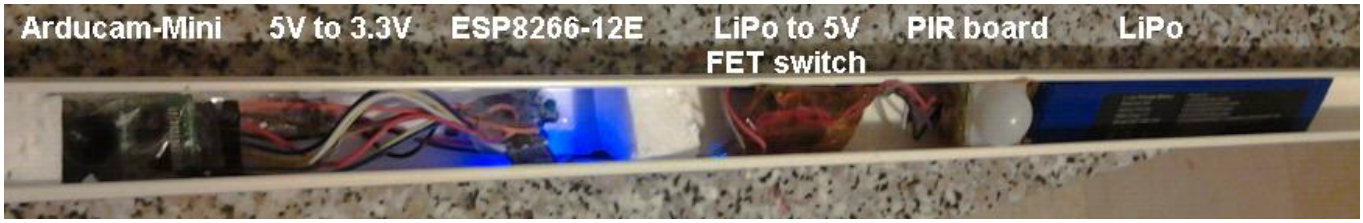
Battery-operated PIR motion detector controls Arducam-Mini ESP8266 power

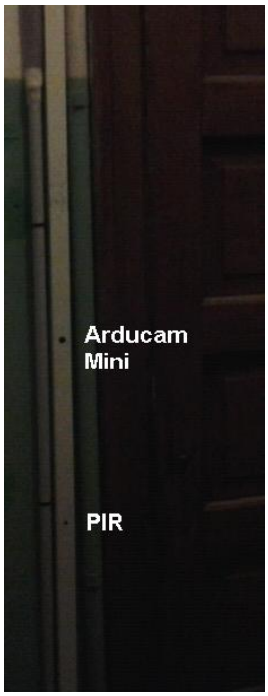
Arducam-Mini board draws nearly 100 mA. ESP8266 board draws ca 150 mA. This makes 3.3x0.25=0.8W power consumption. For battery-powered security camera that is a lot. Let's use an advantage of fast boot time.

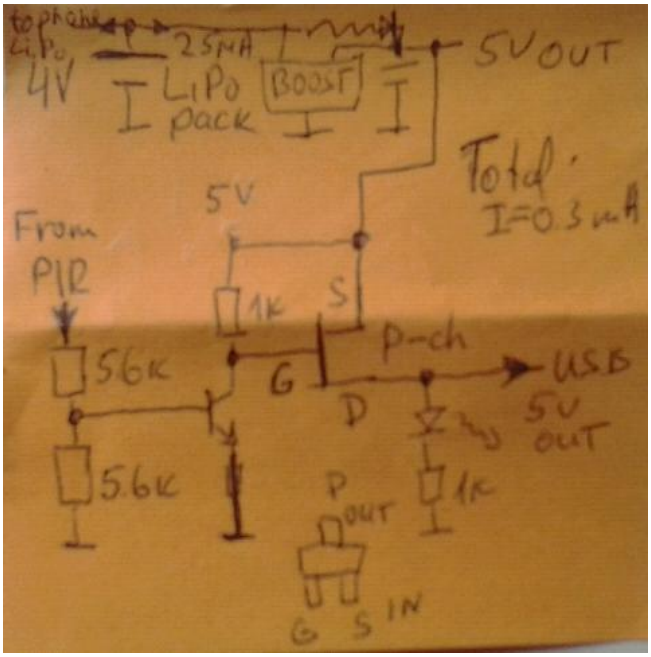
PIR detector + FET is used to switch on 3.3V to the Arducam and ESP only when the motion is detected. ESP board initializes in a couple of seconds and Arducam Mini captures an image in RAM. Then ESP needs some 10 seconds time to connect to a Wi-Fi AP and upload the captured image to a server. Upload takes ca 5 more seconds. The process can repeat and more pictures get uploaded as long as the PIR sensor is active and supplies 3.3V.

I adopted a solar charger battery pack electronics board for this purpose. It produces 5V from a single LiPo. 5V are necessary for the PIR sensor board operation. In standby mode consumption from the LiPo is 0.3mA. To supply ESP and Arducam, the 5V are converted down to 3.3V by a switching regulator board from Ebay.

It is convenient to place electronics inside a cable-channel used in electric installations. Foam bricks are used to fix things inside against sliding.







PIR-controlled power circuit idea was taken from https://www.opensmartcam.com
and my instructable http://www.instructables.com/id/GSM-Android-phone-as-HD-outdoor-webcam/

2016.03.09. Riga, Latvia. By Janis Alnis.