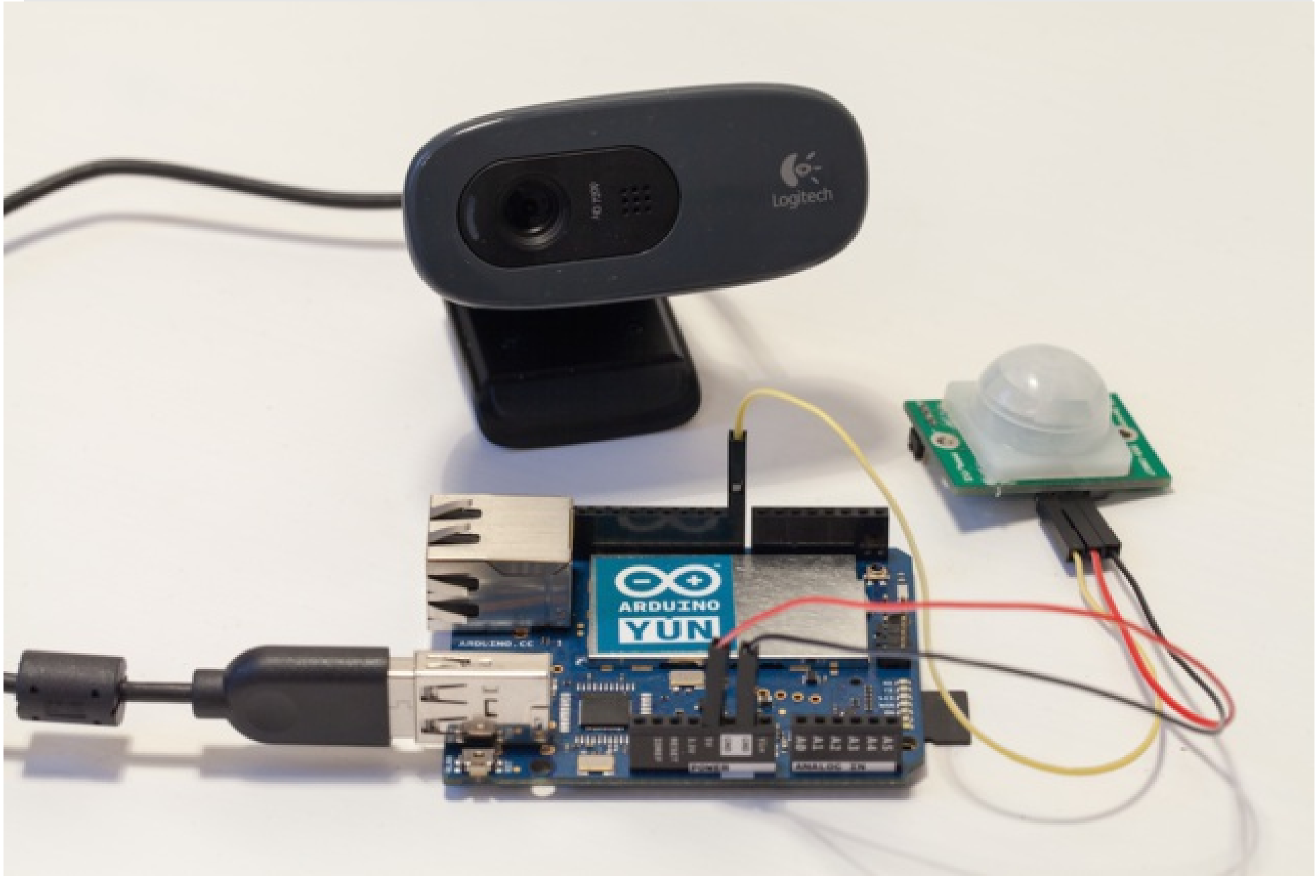
Wireless Security Camera with the Arduino Yun

Created by Marc-Olivier Schwartz

Last updated on 2014-08-13 08:30:11 AM EDT

Guide Contents

Introduction



Ever saw these wireless security cameras that you can buy off the shelf? These are devices that you can setup somewhere in your home or outside, connect to your WiFi network, and then access the video stream from anywhere. However, they are usually using the interface given by the manufacturer, which means you are quite limited with what you can do with your camera.

In this project, we are going to build our own DIY version of such devices. The project is based on the Arduino Yun, to which we are going to connect a standard USB webcam and a PIR motion detector to create some cool applications.

The first application will be a modern version of standard tasks that you want for a security camera: taking pictures when some motion is detected. The project will store pictures taken by the USB camera on an SD card inserted into the Yun, but that's not all. Because we are in the age of the Internet of Things, we also want these pictures to be automatically uploaded on a secure location. And that's exactly what we are going to do by uploading the pictures to Dropbox at the same time.

Then, we are going to stream video coming from the camera directly on Youtube. At the end, you will have your own video stream accessible from anywhere in the world so you can check what is going on in your home. This way, you can also just share the link with your family or friends, so they can monitor your home when you are not there. Excited? Let's dive in!

Connections

Let's see how to configure the hardware part of the project. The first step is to make sure you have the right hardware components. You need an Arduino Yun, a USB webcam, a PIR motion sensor, and a microSD card.

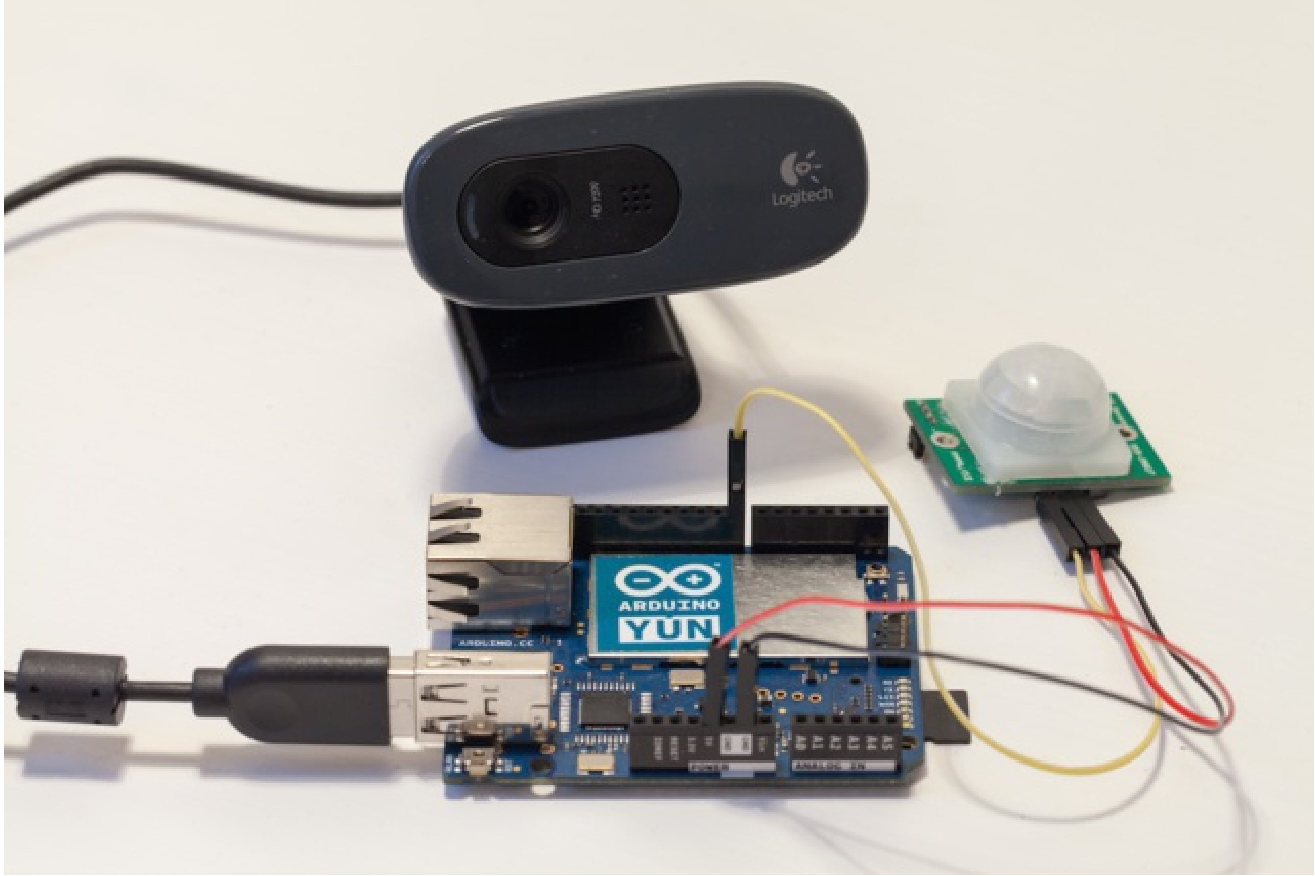
For the USB camera, you can choose any webcam that is compatible with the UVC protocol. Most of the recent webcams are compatible. I choose a Logitech C270 that can take pictures up to 720p resolution. You can find a list of compatible cameras here:

http://en.wikipedia.org/wiki/List_of_USB_video_class_devices (http://adafru.it/dfm)

The first step is to insert the SD card in the Arduino Yun board:

Then, connect the camera to the USB port of the Yun:

Finally, connect the motion sensor to the Yun. Simply connect the VCC pin to the Yun 5V pin, GND to GND, and the SIG pin to the Yun pin number 8:



Finally, just connect the project to your computer via the microUSB port, and you're good to go!

Setting up your Temboo & Dropbox accounts

Before we can build exciting applications with our hardware, we need to setup some accounts on the web services we are going to use.

The first one is Temboo, where you will need to have an account. Temboo will basically make the interface between the Arduino Yun and Dropbox. Just go over to:

https://www.temboo.com/ (http://adafru.it/dcd)

You will be prompted to create an account, and then your first app. Write down the name of your account, the name of the app and your app API key, you will need them later. There is also one more thing you need from the Temboo website: the Temboo Python SDK, that we will use later to upload pictures on Dropbox. You can get it at:

https://www.temboo.com/python (http://adafru.it/dfn)

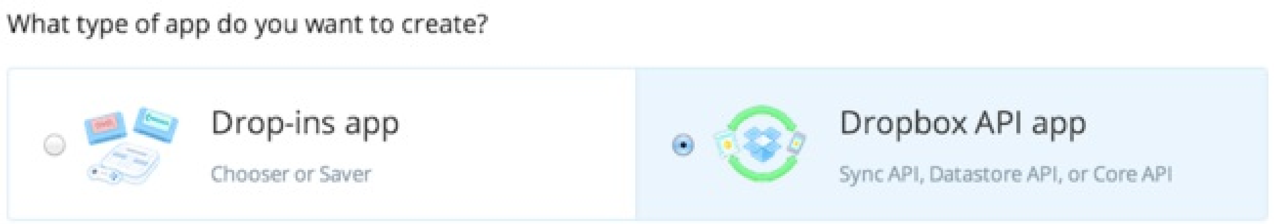
Once downloaded, simply extract the folder on the microSD card. Then, you need a Dropbox account. Go over to the Dropbox website to do so:

https://www.dropbox.com/home (http://adafru.it/dfo)

Once the account is created, you need to create an App so the Yun can upload pictures to your Dropbox folder. To create an app, you need to go to the developers section of Dropbox:

https://www.dropbox.com/developers/apps (http://adafru.it/dfp)

Then, click on “Create app”, and choose which type of app you want to create (Dropbox API app for our project):

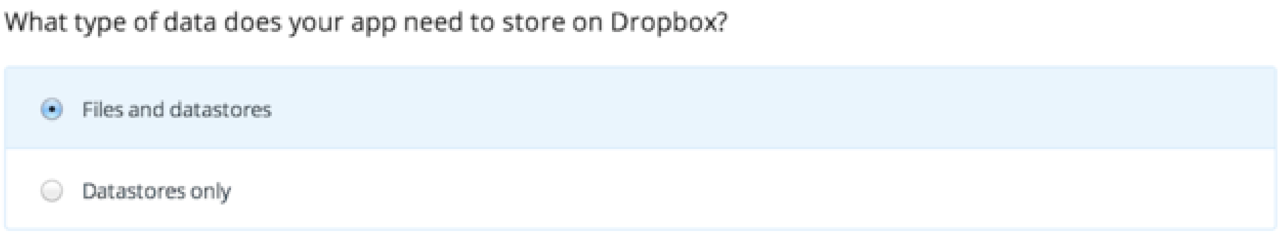


What type of app do you want to create?

○ Drop-ins app
Chooser or Saver

● Dropbox API app
Sync API, Datastore API, or Core API

Then, select "Files and datastores":



What type of data does your app need to store on Dropbox?

● Files and datastores

○ Datastores only

What you need to get now is all the keys relative to your Dropbox app, so you can enter them later in the software of our project. You will need the App Key and App Secret at this point, which are displayed on the same page as your app.

The next data we need to get is the Token Key & Token Secret key. To get them, the first step is to go to the InitialiseOAuth Choreo on the Temboo website:

https://temboo.com/library/Library/Dropbox/OAuth/InitializeOAuth/ (http://adafru.it/djg)

Here, you will need to enter the App Key and App Secret. That will generate some additional information, like a callback ID, and a temporary token secret. You'll also be asked to visit a link to Dropbox to confirm the authentification. Finally, go to the FinalizeOAuth page to finish the process. You'll be asked to enter your App Key, App Secret, callback ID and temporary token secret:

https://temboo.com/library/Library/Dropbox/OAuth/FinalizeOAuth/ (http://adafru.it/djh)

After that step, you'll be given your final Token Key and Token Secret. Write them down, you'll need them later.

You will also need to have a Google account to stream videos on Youtube. Simply create one at:

https://accounts.google.com/ (http://adafru.it/dfq)

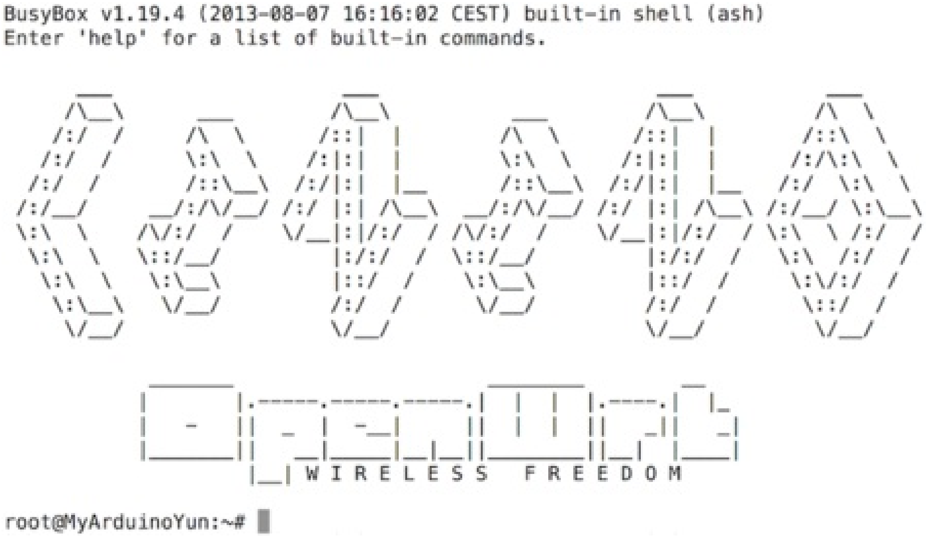
Setting up your Arduino Yun

Before we can write the software for our project, we need to install some more software on your Arduino Yun. We are also going to test the USB camera to see if the drivers are working correctly. In the rest of the project, I will assume that your Yun is already setup and connected to your local WiFi network.

First, the UVC drivers. To install them, you need to connect to your Yun via SSH. Simply open a terminal, and type:

```
ssh root@yourYunName.local
```

Where you need to insert the correct name of your Arduino Yun, that you set when you first used the Yun. You will also be prompted to enter your password. If the connection is successful, you should see some ASCII art:



```
BusyBox v1.19.4 (2013-08-07 16:16:02 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

      ___                   ___                       ___           ___
     /\__\      ___        /\__\          ___        /\__\         /\  \
    /:/  /     /\  \      /::|  |        /\  \      /::|  |       /::\  \
   /:/  /      \:\  \    /:|:|  |        \:\  \    /:|:|  |      /:/\:\  \
  /:/  /       /::\__\  /:/|:|  |__      /::\__\  /:/|:|  |__   /:/  \:\  \
 /:/__/     __/:/\/__/ /:/ |:| /\__\  __/:/\/__/ /:/ |:| /\__\ /:/__/ \:\__\
 \:\  \    /\/:/  /    \/__|:|/:/  / /\/:/  /    \/__|:|/:/  / \:\  \ /:/  /
  \:\  \   \::/__/         |:/:/  /  \::/__/         |:/:/  /   \:\  /:/  /
   \:\  \   \:\__\         |::/  /    \:\__\         |::/  /     \:\/:/  /
    \:\__\   \/__/         /:/  /      \/__/         /:/  /       \::/  /
     \/__/                 \/__/                     \/__/         \/__/

            _______                     ________        __
           |       |.-----.-----.-----.|  |  |  |.----.|  |_
           |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
           |_______||   __|_____|__|__||________||__|  |____|
                    |__| W I R E L E S S   F R E E D O M

root@MyArduinoYun:~#
```

We can now install the required packages. Start with an update of the package manager:

```
opkg update
```

Then, install the UVC drivers:

```
opkg install kmod-video-uvc
```

And the python-openssl package:

```
opkg install python-openssl
```

We also need the fswebcam utility that we will use to take pictures from the terminal:

```
opkg install fswebcam
```

For the last part of the project, we are also going to need the mjpg streaming library. You can get it with:

```
opkg install mjpg-streamer
```

We are now ready to test the webcam. Make sure that the SD card is mounted into the Yun, and go over to the SD card folder with:

```
cd /mnt/sda1
```

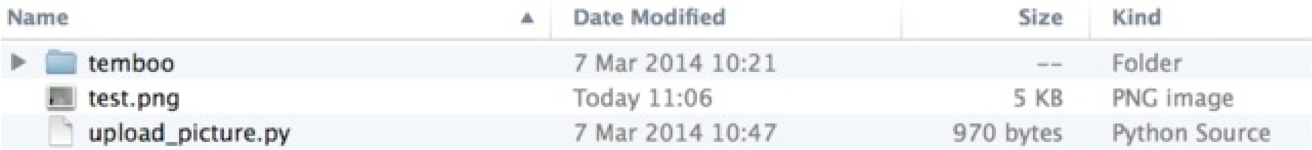
To test the camera and take a picture, it is really easy. Simply type:

```
fswebcam test.png
```

You should see some information being displayed, along with some errors, but don't worry about them. The important thing is to see these lines:

```
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
```

To check that the picture was correctly taken, remove the SD card from the Yun and read it using your computer. You should see the picture appearing at the root of the SD card:



Name	Date Modified	Size	Kind
temboo	7 Mar 2014 10:21	--	Folder
test.png	Today 11:06	5 KB	PNG image
upload_picture.py	7 Mar 2014 10:47	970 bytes	Python Source

Simply open it to make sure it was correctly taken and that it is not corrupted. If the picture looks good, you can go over to the next section and start building cool applications with the project!

Upload pictures to Dropbox

What we want to achieve in this first application is to take a picture whenever some motion is detected by the PIR motion sensor. And when that happens, store this picture locally on the SD card, and upload it to Dropbox. To do so, the code will be composed of two parts. The first one is a Python script that will connect to Dropbox, take a picture on the SD card, and then upload this picture to Dropbox. The reason to use Python for this part is that it is much easier to upload files to Dropbox using Python than directly from the Arduino sketch. The second part of the code will be the Arduino sketch itself, which will basically call the Python script to take pictures via the Bridge library of the Yun.

Let's first code the Python script. It starts by including the required libraries from the Temboo Python SDK:

```
from temboo.core.session import TembooSession
from temboo.Library.Dropbox.FilesAndMetadata import UploadFile
```

The Python script will also take the name of the picture we want to upload as an argument:

```
with open(str(sys.argv[1]), "rb") as image_file:
  encoded_string = base64.b64encode(image_file.read())
```

Remember these Temboo credentials that you created earlier ? That's where you need to enter them:

```
session = TembooSession('yourTembooName', 'yourTembooApp', 'yourTembooKey')
```

We can then create the correct Dropbox library to upload files, called a "Choreo" on Temboo:

```
uploadFileChoreo = UploadFile(session)
uploadFileInputs = uploadFileChoreo.new_input_set()
```

It's now the time to enter all the informations about your Dropbox account, like your app Key, app Secret, Access Token an Access Token Secret:

```
uploadFileInputs.set_AppSecret("appSecret")
uploadFileInputs.set_AccessToken("accessToken")
uploadFileInputs.set_FileName(str(sys.argv[1]))
uploadFileInputs.set_AccessTokenSecret("accessTokenSecret")
uploadFileInputs.set_AppKey("appKey")
uploadFileInputs.set_FileContents(encoded_string)
uploadFileInputs.set_Root("sandbox")
```

And finally, upload the file on Dropbox:

```
uploadFileResults = uploadFileChoreo.execute_with_results(uploadFileInputs)
```

You can now save this code in a file named upload_picture.py. Note that all the files are available on the GitHub repository of the project:

https://github.com/openhomeautomation/arduino-yun-camera (http://adafru.it/dfr)

We'll now work on the Arduino sketch. The sketch starts by including the required libraries:

```
#include <Bridge.h>
#include <Process.h>
```

We also have to declare a process, that we are going to use to call functions on the Linux machine of the Yun (for example the fswebcam utility we used before):

```
Process picture;
```

We are also going to build a filename for each picture the project will take, that will be stored in a string:

```
String filename;
```

We also declare the pin on which the PIR motion sensor is connected:

```
int pir_pin = 8;
```

And the path of the SD card on the Yun:

```
String path = "/mnt/sda1/";
```

Because we need to call functions on the Linux machine of the Yun, we have to start the Bridge:

```
Bridge.begin();
```

Then, in the loop() part of the sketch, we check if some motion was detected by the PIR sensor:

```
if (digitalRead(pir_pin) == true) {
```

If this is the case, we build a unique filename for the picture, with the date at which the picture was taken:

```
filename = "";
picture.runShellCommand("date +%s");
while(picture.running());

while (picture.available()>0) {
  char c = picture.read();
  filename += c;
  }
filename.trim();
filename += ".png";
```

We then make the first call to the Linux machine of the Yun, first to take a picture with the fswebcam utility. Note that here, we provide an extra argument with the -r command, which set the resolution. I used the maximum resolution of my camera, which is 720p:

```
picture.runShellCommand("fswebcam " + path + filename + " -r 1280x720");
while(picture.running());
```

We then make a second call to the Linux machine, this time calling the Python script with the name of the picture as an argument, which will upload the picture to Dropbox:

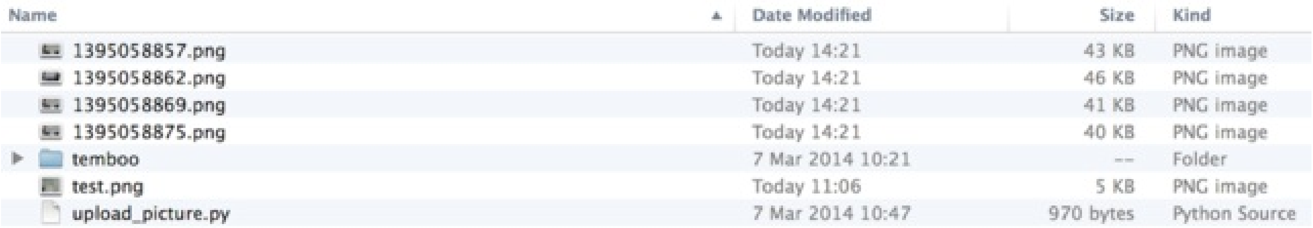
```
picture.runShellCommand("python " + path + "upload_picture.py " + path + filename);
while(picture.running());
```

You are now ready to test the project. Again, all the files are available on the GitHub repository of the project:

https://github.com/openhomeautomation/arduino-yun-camera (http://adafru.it/dfr)

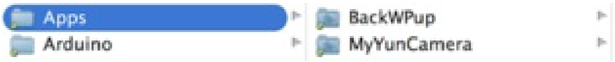
First, put the Python file at the root of the SD card, and put the SD card back into the Arduino Yun board. Then, upload the Arduino sketch to the Yun. Now, try to trigger the motion sensor, for example by waiving your hand in front of it. You should see that the webcam is being activated shortly after (for example, my webcam has a LED that turns green when it is active).

To check that the project is working correctly, after a while you can check the SD card, you should see that some pictures have been recorded:



Name	Date Modified	Size	Kind
1395058857.png	Today 14:21	43 KB	PNG image
1395058862.png	Today 14:21	46 KB	PNG image
1395058869.png	Today 14:21	41 KB	PNG image
1395058875.png	Today 14:21	40 KB	PNG image
temboo	7 Mar 2014 10:21	--	Folder
test.png	Today 11:06	5 KB	PNG image
upload_picture.py	7 Mar 2014 10:47	970 bytes	Python Source

You can also check on your Dropbox folder, where the same pictures should have been uploaded. They should be located in your Dropbox apps folder:



Stream video to Youtube

In this last part of the project, we are going to build a completely different application for our hardware: make the camera stream video live on Youtube. We are first going to make the camera stream video locally, then transmit this stream to your computer using a software called Wirecast, which will finally stream the video to a Youtube live event.

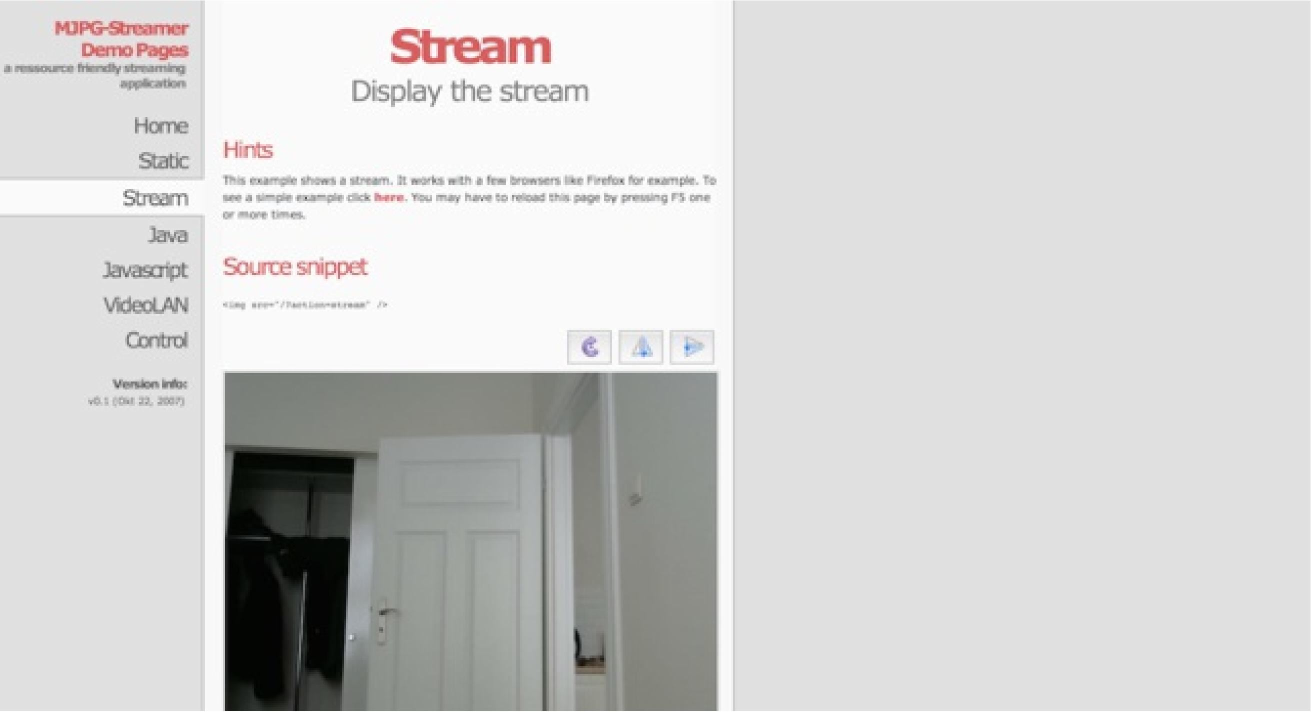
Streaming video locally is actually really easy, thanks to the software we installed earlier. Just log to the Yun again via SSH, and type:

```
mjpg_streamer -i "input_uvc.so -d /dev/video0 -r 640x480 -f 25" -o "output_http.so -p 8080 -w /www/v
```

Some explanation about this code: the -r argument sets the resolution (I used a lower resolution here because the HD video streaming was lagging) and the -p argument sets the port on which the stream will be available. To actually see the stream, just go over to (by replacing the name of your Arduino Yun board):

```
http://myarduinoyun.local:8080/stream.html
```

You should get a page which is the interface to the stream on your Arduino Yun. On this page, you should see the video from your camera being streamed in live:



Stream

Display the stream

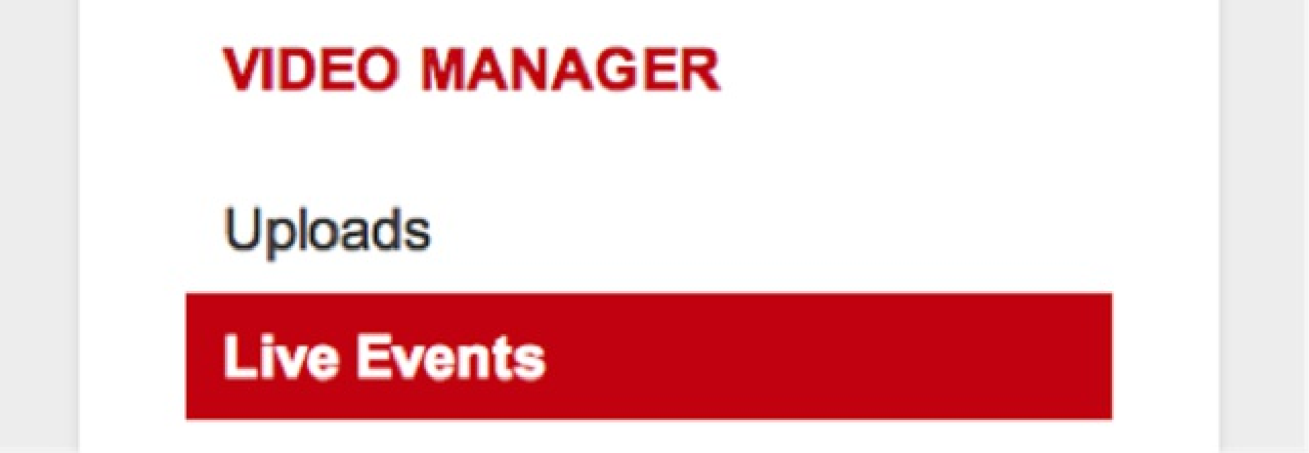
Hints

This example shows a stream. It works with a few browsers like Firefox for example. To see a simple example click here. You may have to reload this page by pressing F5 one or more times.

Source snippet

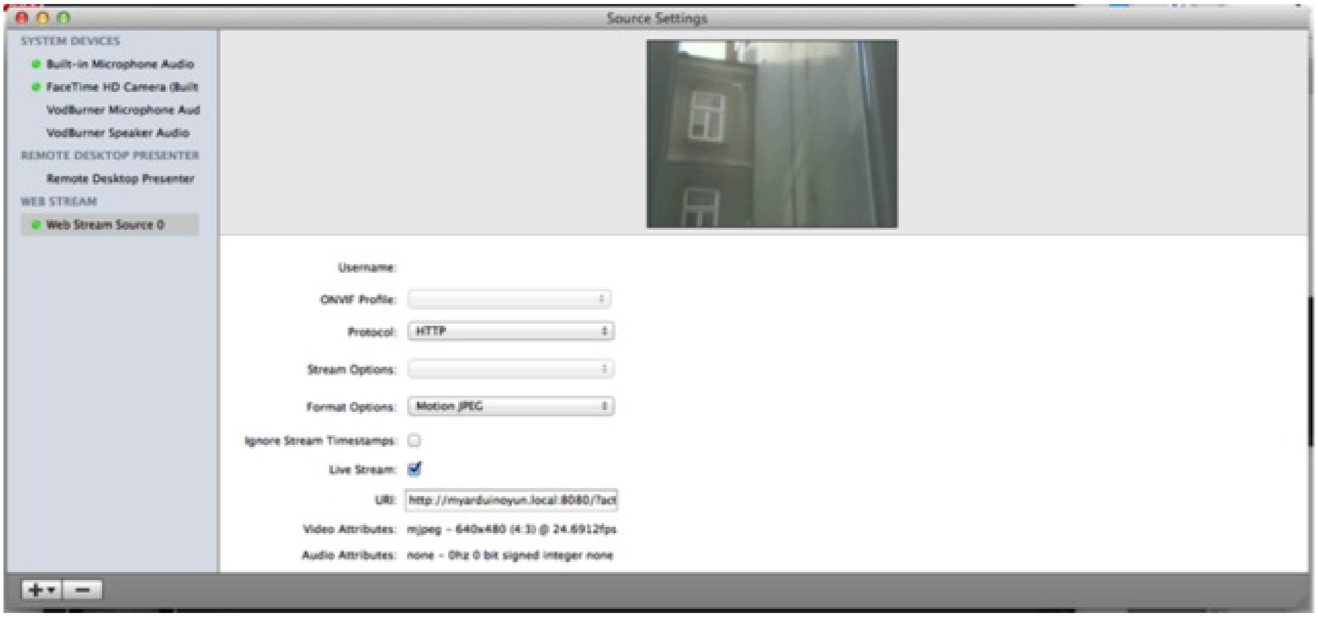
```
<img src="/?action=stream" />
```

Now, we are going to send this stream on Youtube. It starts by going to the Youtube interface, and creating a live event:



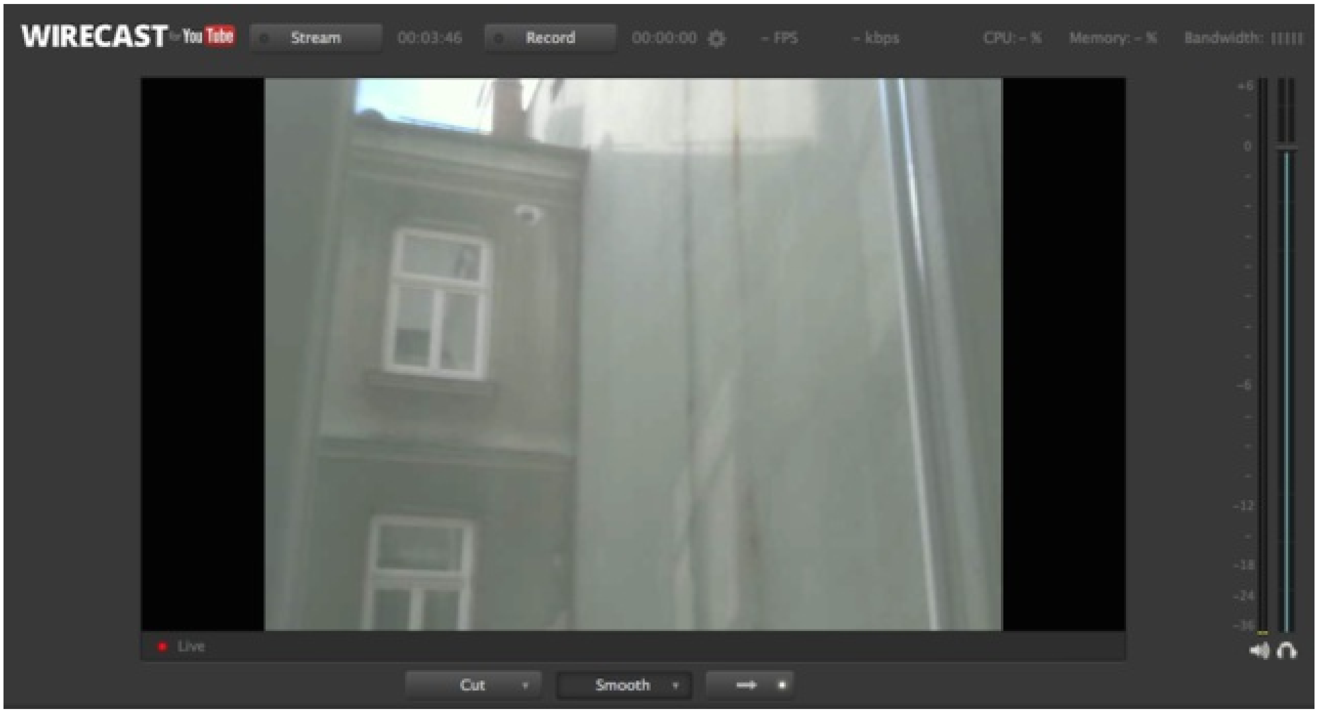
Youtube will ask you some information about your stream, and recommend some software so your computer can stream to Youtube. I used Wirecast, but it should work with other streaming software as well.

In Wirecast, you need to add a new "Web stream" with the following parameters: HTTP protocol, Motion JPEG format, and "nameofyourYun.local:8080/?action=stream" as the URI. This picture shows the parameters I used:

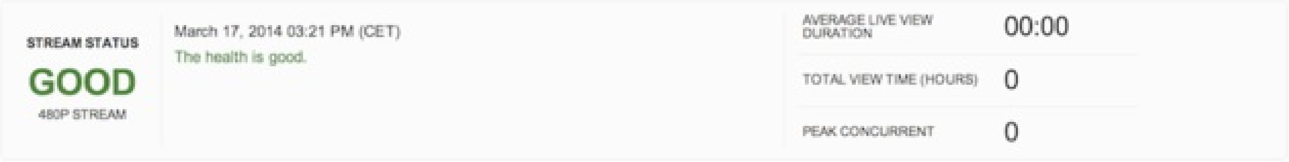


Username:
ONVIF Profile:
Protocol: HTTP
Stream Options:
Format Options: Motion JPEG
Ignore Stream Timestamps: □
Live Stream: ☑
URI: http://myarduinoyun.local:8080/?act
Video Attributes: mjpeg - 640x480 (4:3) @ 24.6912fps
Audio Attributes: none - 0hz 0 bit signed integer none

The free version of Wirecast will insert some watermarks on your stream, as the Web stream is a paid feature. But that is not really a problem to monitor your home, as they don't appear all the time. After a moment, the stream from your Yun should appear in the Wirecast interface:



Still on Wirecast, you can now press on "Stream". The software will ask for your Youtube credentials, and should automatically detect your live event. Once this is done, you can go back to the live event on Youtube. On the live event control room, you should see that Youtube is receiving some data:



STREAM STATUS
GOOD
480P STREAM

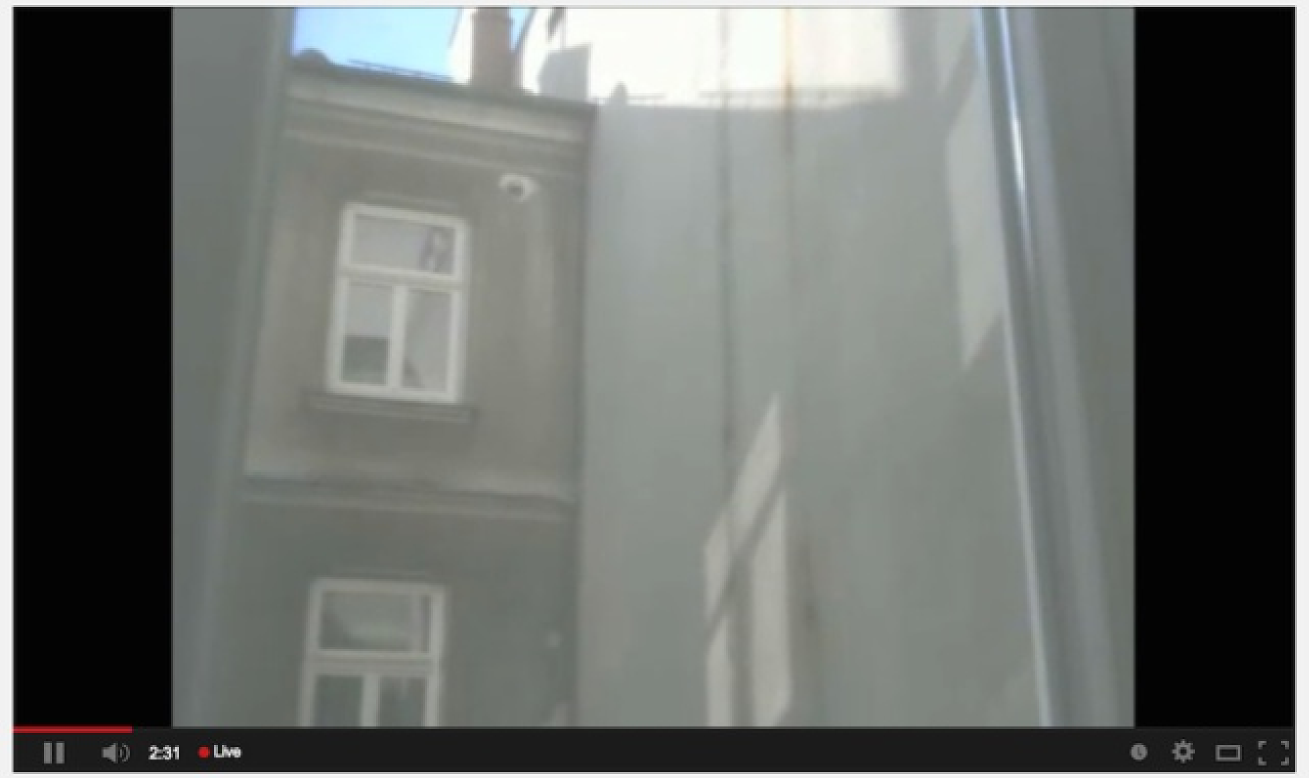
March 17, 2014 03:21 PM (CET)
The health is good.

AVERAGE LIVE VIEW DURATION	00:00
TOTAL VIEW TIME (HOURS)	0
PEAK CONCURRENT	0

If you can see the "GOOD" status, you can click on "Preview" so Youtube can prepare your stream:

Preview
Help
Sync to preview player

It will take a while (some minutes in my case), but at some point you will be able to click on "Start streaming". Once this is done, you can access your stream just like you would access any Youtube video:



Congratulations, you can now monitor your home from wherever in the world, just by having the URL to this Youtube live event! You have several options to create your live event, but I set mine as "Unlisted" so I can share the URL with people I trust, so they can also have a look on my home when I am not there.

Summary

In this project, you learned how to connect a USB camera to your Arduino Yun and build two exciting projects on top of it. The first one is a security camera that automatically upload pictures on Dropbox when motion is detected in front of the camera. The second project was a camera that streams video on a Youtube event, so you can monitor your home from anywhere in the world.

Of course, there are several ways to build other cool applications using this project. You can for example drop the motion detection part, and build a camera that take snapshots at regular intervals and upload these on Dropbox. You can for example easily create time-lapse videos with this kind of project: just collect the pictures from your Dropbox account, paste them into a time-lapse software, and done! You can also extend this project by adding more Yun + camera modules, to have a complete video monitoring system in your home!