OpenCV Computer Vision Kit

User Guide

Qi Yao

Table of Contents

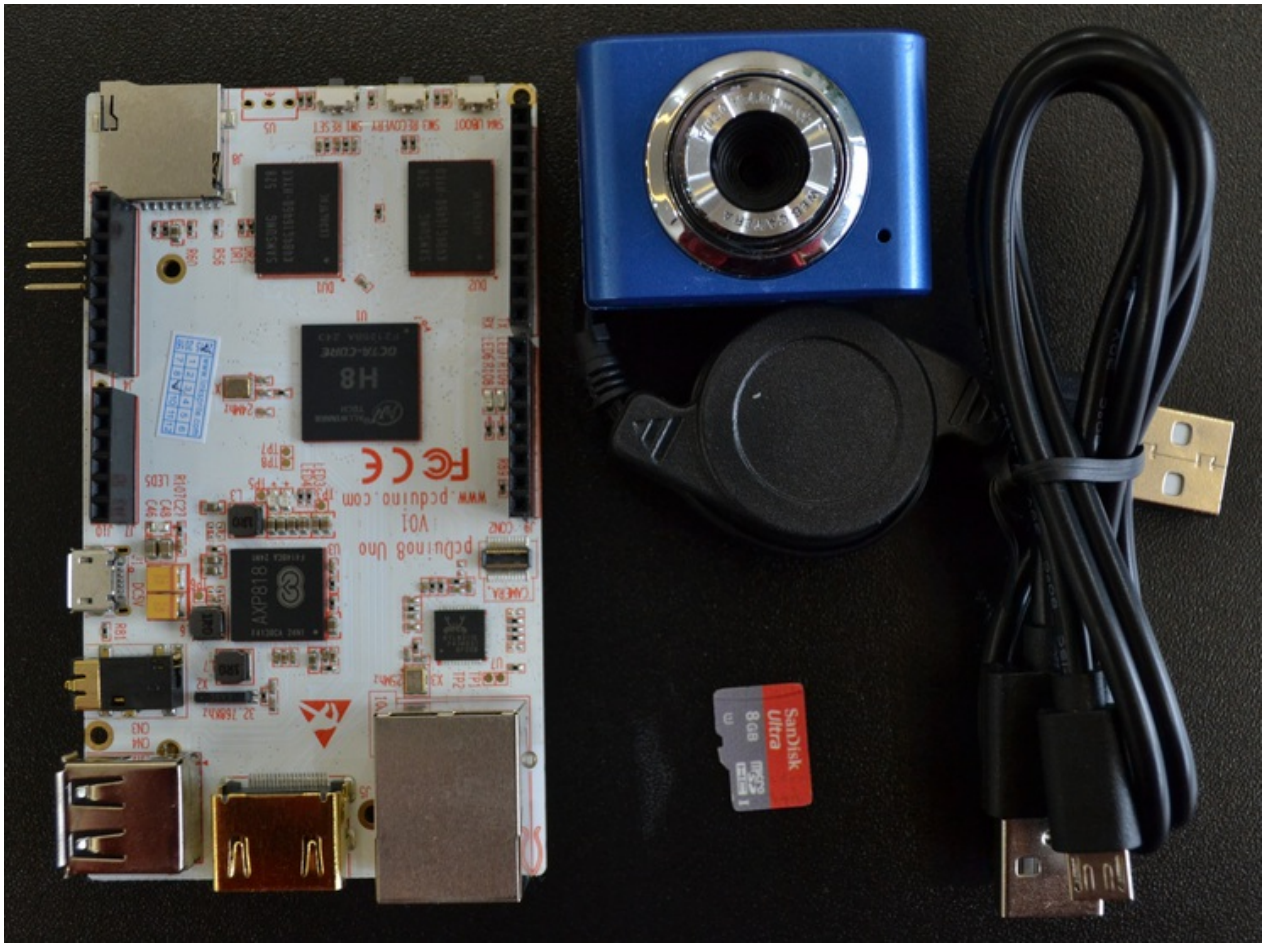
Introduction of OpenCV computer vision kit

pcDuino is pin to pin compatible with Arduino so that existing Arduino shields can be installed on pcDuino with a simple translation board (T-board). By being compatible with Arduino ecosystem, pcDuino is a platform that bridges the power of open software linux and the power of open hardware.

Recently, LinkSprite released a powerful mini PC platform pcDuino8 Uno which is powered by Allwinner H8 SoC chip. It has 8 Cortex-A7 ARM cores whose operating frequency is up to 2.0GHz.

Base on this powerful platform, we have constructed a new kit called OpenCV computer vision kit which users can use to quick start OpenCV computer vision and do lost of fun stuff including but not limited to the follows:

- Learn or teach programming
- Learn Ubuntu Linux
- Work with hardware part
- OpenCV computer vision
- Implement a network video monitoring system
- DIY a simple camera
- Motion detection
- Face detection



Specifications

1. Hardware

- pcDuino8 Uno x 1
- USB WebCam x 1
- USB micro data line x 1

- 8G TF card x 1

2. Software

We has created a Ubuntu14.04 image file for OpenCV computer vision kit, which has pre-installed the following things:

- git
- vim
- Python 2.7
- OpenCV 2.4.11
- motion
- guvcview
- ffmpeg
- demo source code(at home directory)

More information

If you want to get more documents and demos' source code for pcDuino8 Uno, please check the following websites.

1. OpenCV Computer Vision Kit Guide

This is the github repository collecting documents and demos about OpenCV computer vision kit and let user quick start on OpenCV computer vision. Use git to download:

```
$ git clone https://github.com/pcduino/pcduino8-uno-guide
```

2. Image for pcDuino8 Uno

This website has collected different system image for pcDuino8 Uno, including Ubuntu and Android.

3. Learning Center

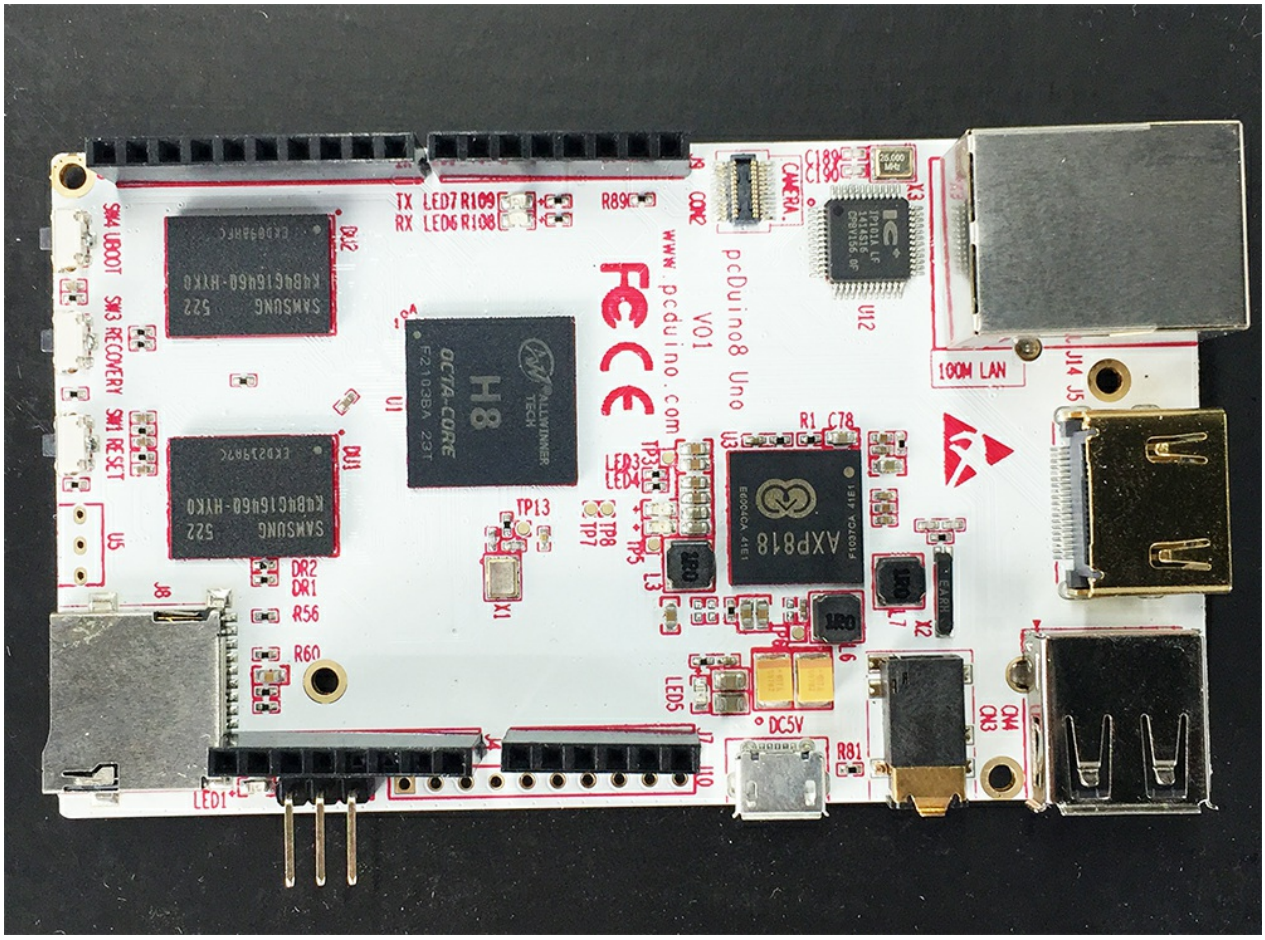
This website has been collected many posts on pcDuinos, not only pcDuino8 Uno.

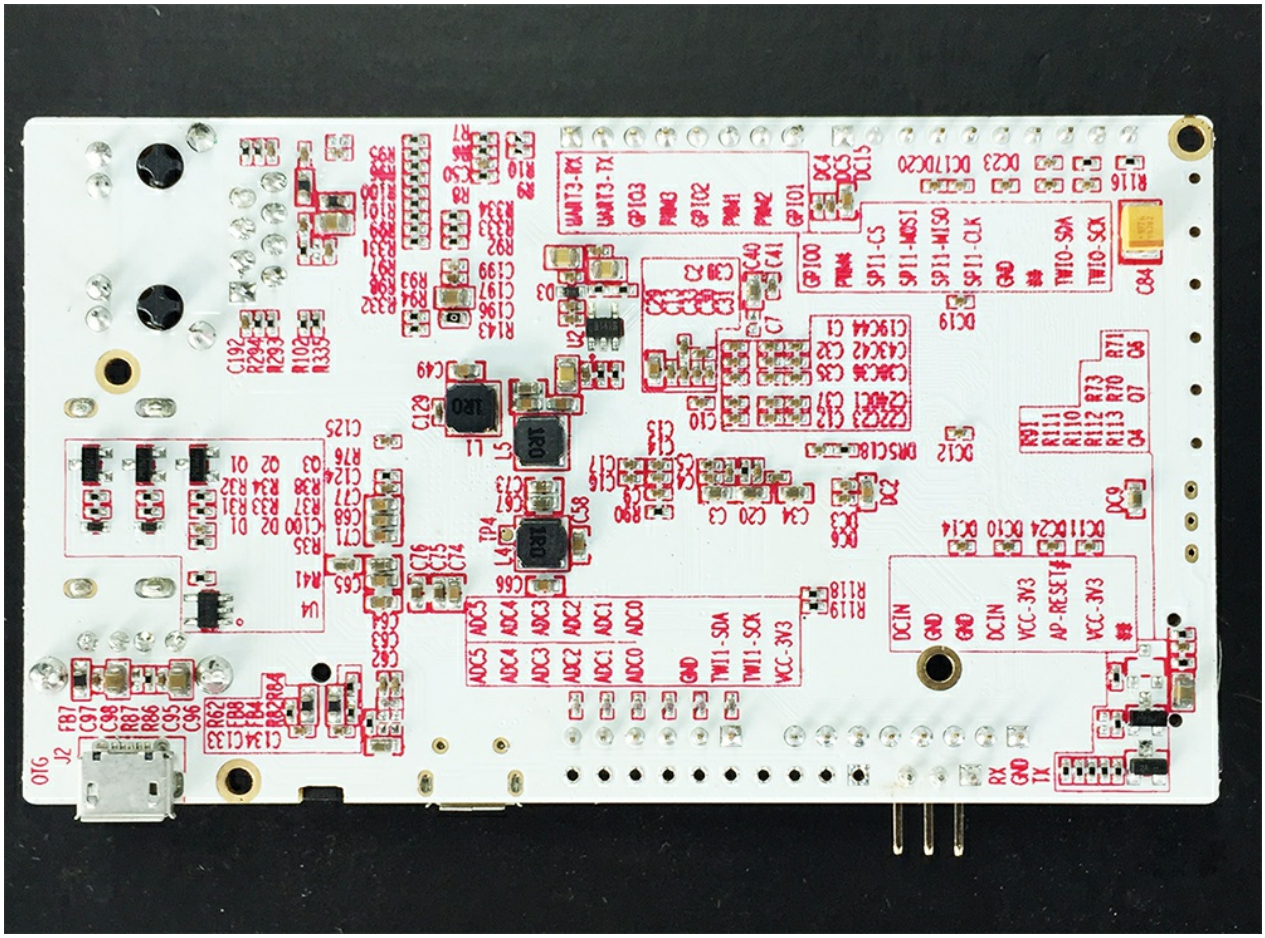
4. Forum

If you have any question, please post it there, LinkSprite Engineers will try their best to answer your questions.

What is pcDuino8 Uno?

pcDuino8 Uno is a high performance, cost effective single board computer. It runs operation systems such as Ubuntu Linux and Android. pcDuino8 Uno has HDMI interface to output its graphic desktop screen. It could support multi-format 1080p 60fps video decoder and 1080p 30fps H.265/HEVC video encoder with its built-in hardware video processing engine. It targets specially the fast growing demands from the open source community. pcDuino8 Uno provides easy-to-use tool chains and is compatible with the popular Arduino ecosystem such as Arduino Shields.



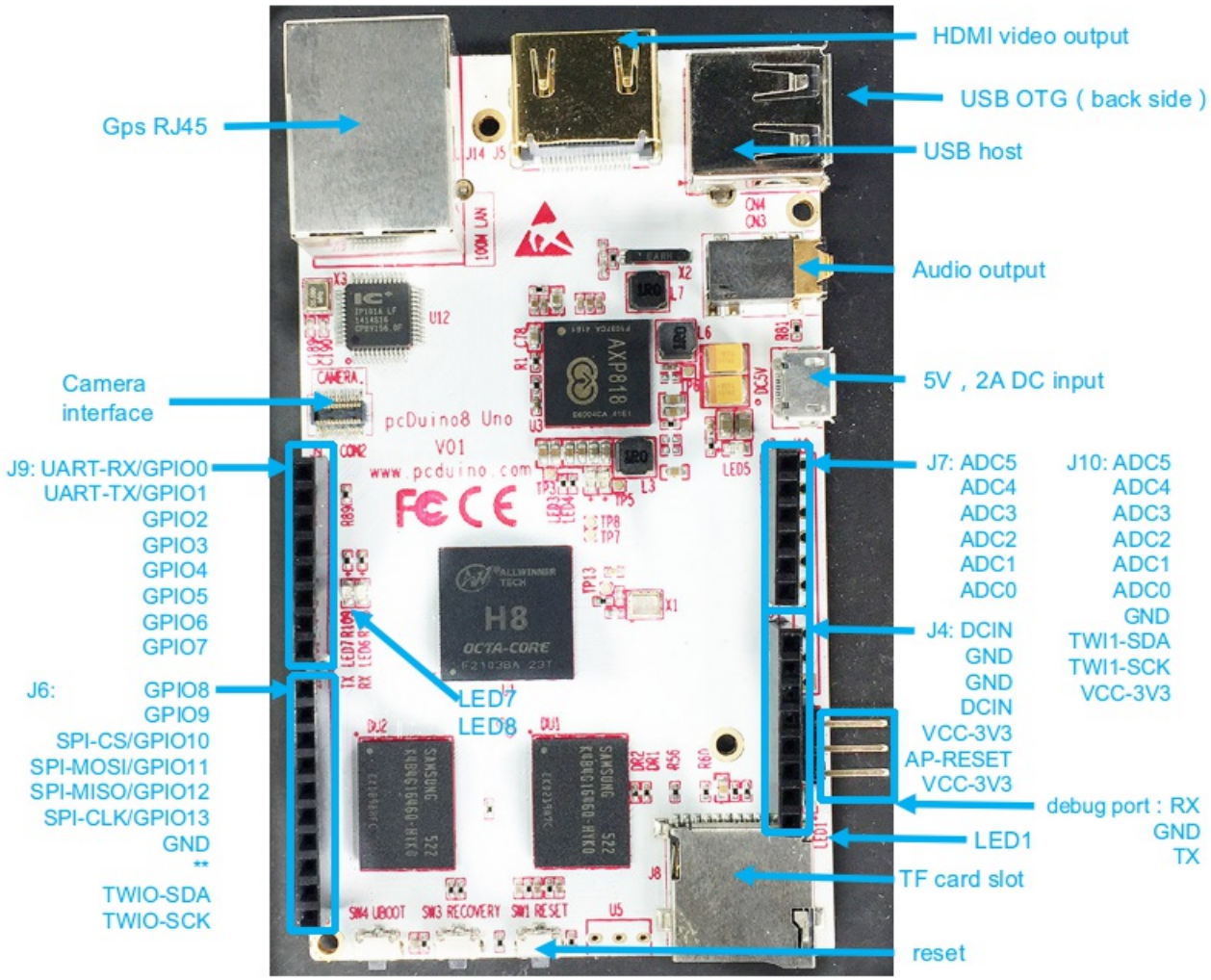


Hardware Specifications

Items	Details
CPU	AllWinner H8 8-Core Cortex-A7 @ 2.0GHz
GPU	Power VR SG544 @720MHz
DRAM	DDR3 1GB
Onboard Storage	microSD card slot for up to 32GB
Network Interface	Gbps RJ45
OS	Ubuntu 14, Android 4.4
Video Output	HDMI 1.4 with HDCP support
Arduino extension interface	Arduino sockets, same as Arduino Uno 14x GPIO, 2xPWM, 1x UART, 1x SPI 1x I2C, 6xADC (extra module needed to provide ADC)
HDMI	1 x HDMI 1.4
Audio Output	1 x 3.5mm analog audio interface
IR	1 x Infrared Receiver (Not populated, user can post-install)
Camera	MIPI
USB	1 x USB Host, 1 x USB OTG
Power	5V, 2000mA Micro USB
Overall Size	3 5/8″ x 2 1/8″

Arduino extension interface

If you want to program Arduino extension interface, you should know the pin map shown as follows:



Quick Start

This section will tell you how to quick start pcDuino8 Uno and the following topics will be discussed:

- Create bootable SD card for pcDuino8 Uno
- Use VNCViewer to login Ubuntu via USB OTG
- Arduino programming quick start
- Python programming to control GPIO
- Webcam quick start

Create bootable TF card for pcDuino8 Uno(Ubuntu)

If you want to install or refresh the Ubuntu system on pcDuino8 Uno, you can just take the following steps to create the bootable TF card for pcDuino8 Uno:

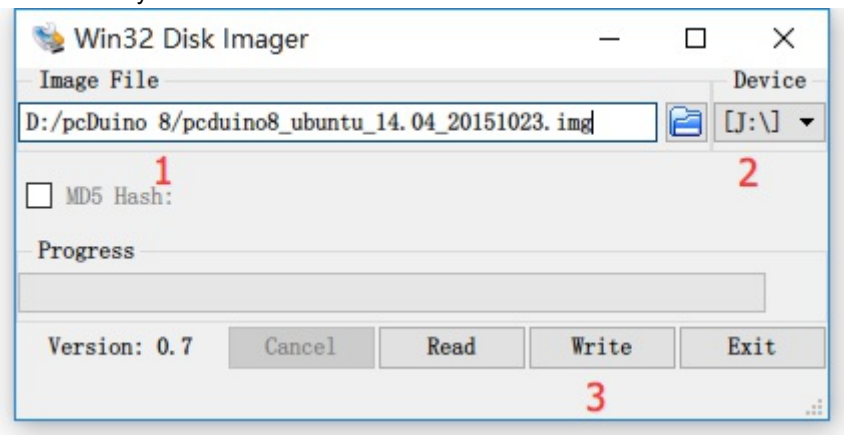
Prepare

- 8G (or bigger) TF card(Speed class is recommended for Class 6 or higher)
- TF card reader
- Image for pcDuino8 Uno
- Windows tool: Win32DiskImager

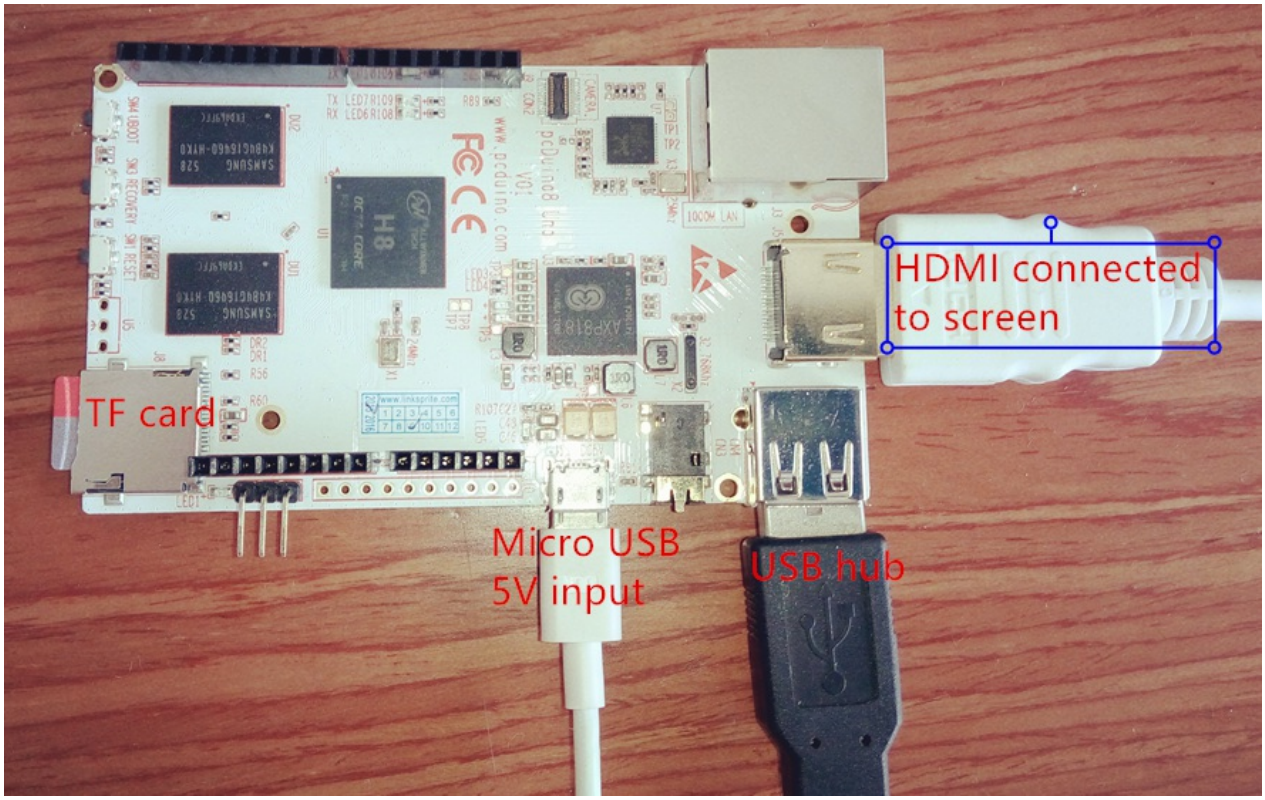
Create

Plug TF Card Reader into Windows PC and open Win32DiskImager as administrator.

- Select the **Image File** that you unzip from the downloaded File
- Select the **Device** that the PC recognized
- Click the **write** button.
- It will say **Write Successful** at the end.



Run



- Plug the TF card into pcDuino8 Uno
- Connect 5V DC power input via Micro USB port
- Connect keyboard and mouse with USB hub
- Connect screen with HDMI cable.
- Power on and run

Note: **For the first time boot, system initialization may take some time (about 2-3 minutes), and USB port is disabled. Please do not shutdown before system restart.**

VNC to pcDuino8 Uno through USB-OTG in Win8 system

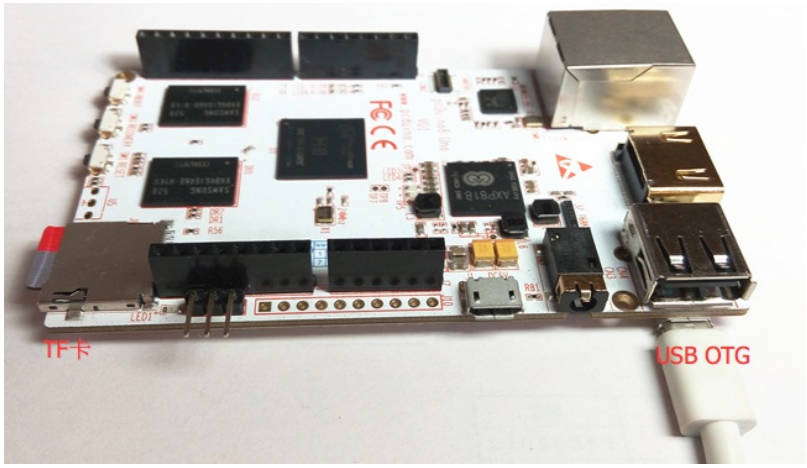
This tutorial will tell you how to quick access Ubuntu running on pcDuino8 Uno , just using an USB data line,without keyboard, mouse, and HDMI screen.

Prepare

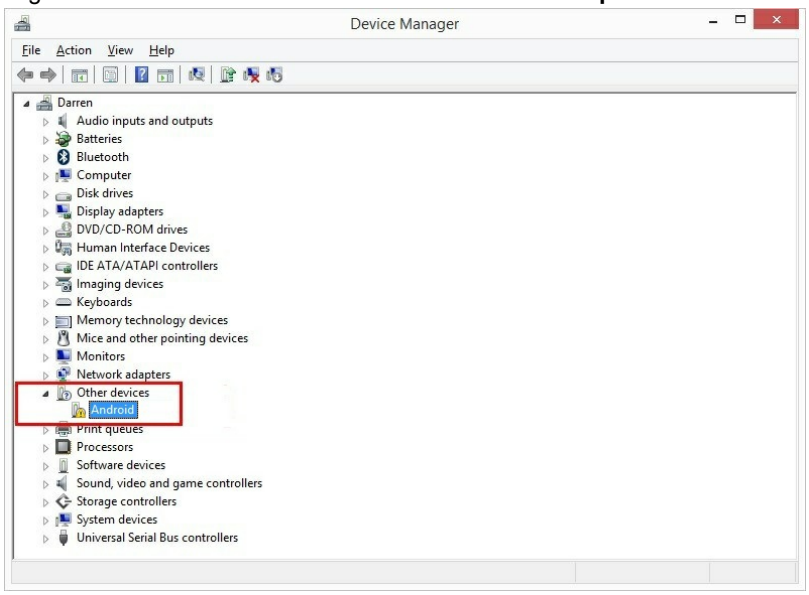
- pcDuino8 Uno
- USB micro USB line
- Windows PC(XP/Win7/Win8)
- Windows tool: vncviewer

Install Driver(Take Win7 for example)

1. First connect pcDuino8 Uno to the computer's USB port through USB OTG.



2. Open **Device Manager** in Windows.
3. Right click **Android** under **other devices** and choose **Update Driver Software**.



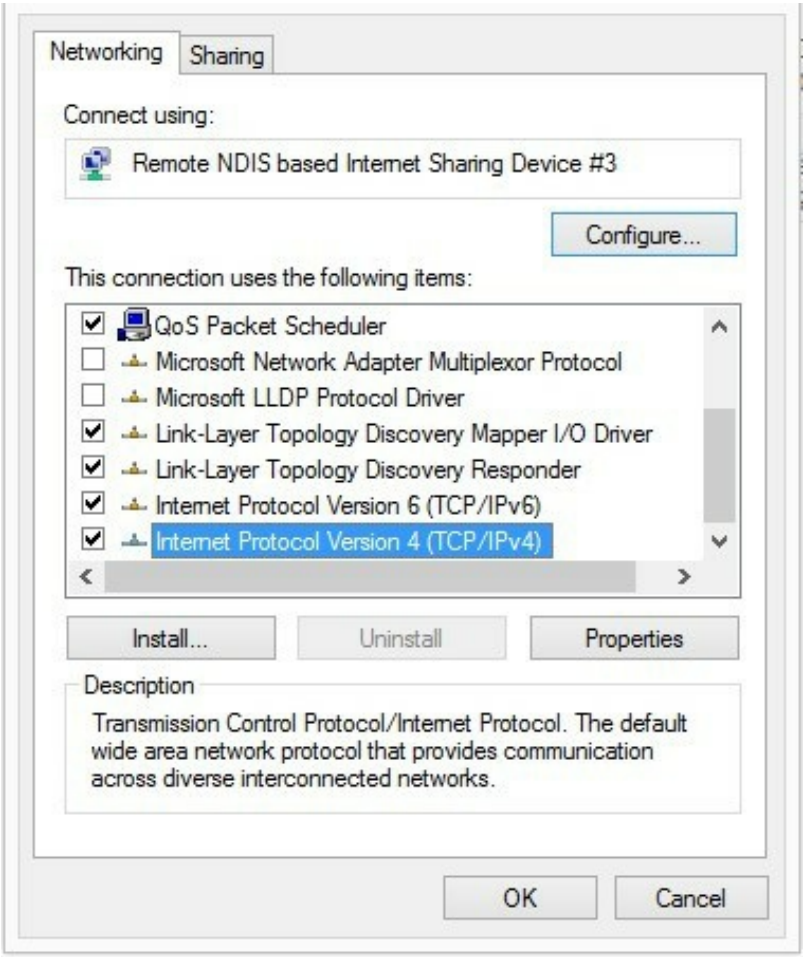
4. Choose the **browse my computer for driver software**.
5. Under **Common hardware types**, choose **Network adapters**.
6. Under Manufacturer choose **Microsoft Corporation** , and Network Adapter choose **Remote NDIS based**

Compatible Device.

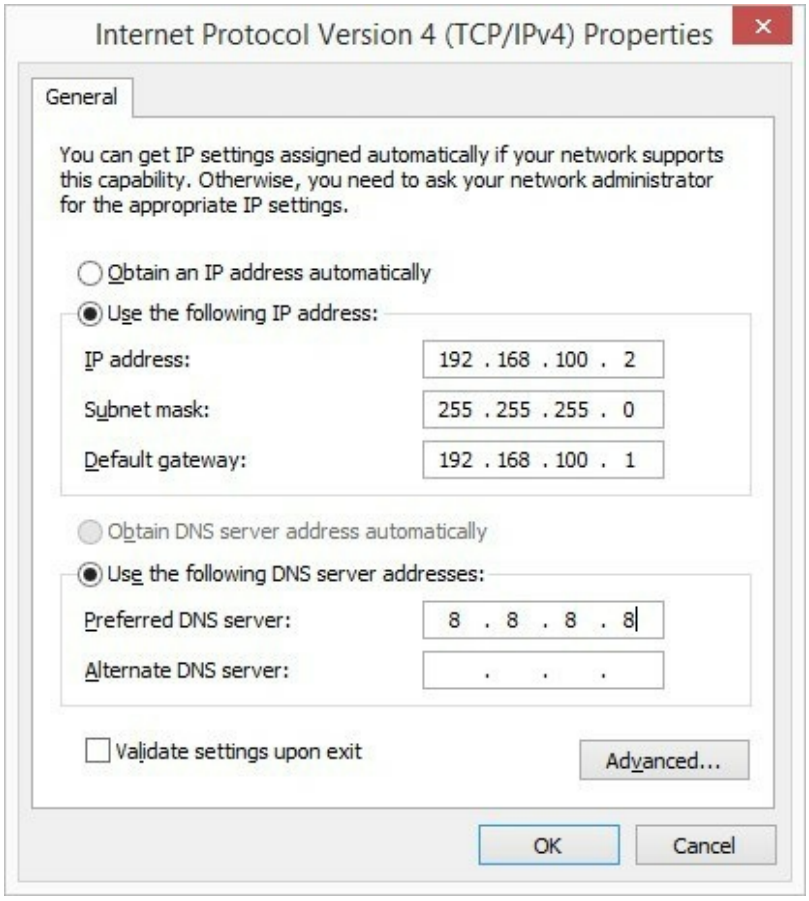
7. Windows will prompt the warning information, click Yes, after a while it will prompt that the Windows has successfully updated your driver software.

Set the Static Address

1. Right click the option(or win +x), choose **Network Connections**.
2. Right click the **Remote NDIS based Compatible Device**, choose properties.
3. Double click Internet Protocol version 4 (TCP/IPv4).

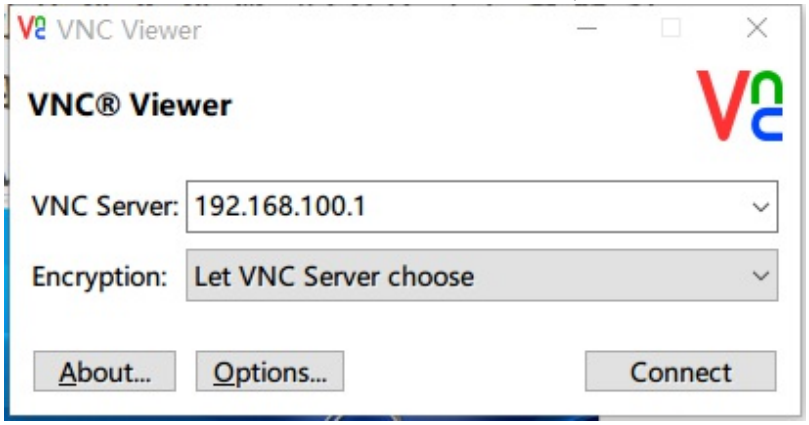


4. General set options as follow:



Use VNC to connect to pcDuino:

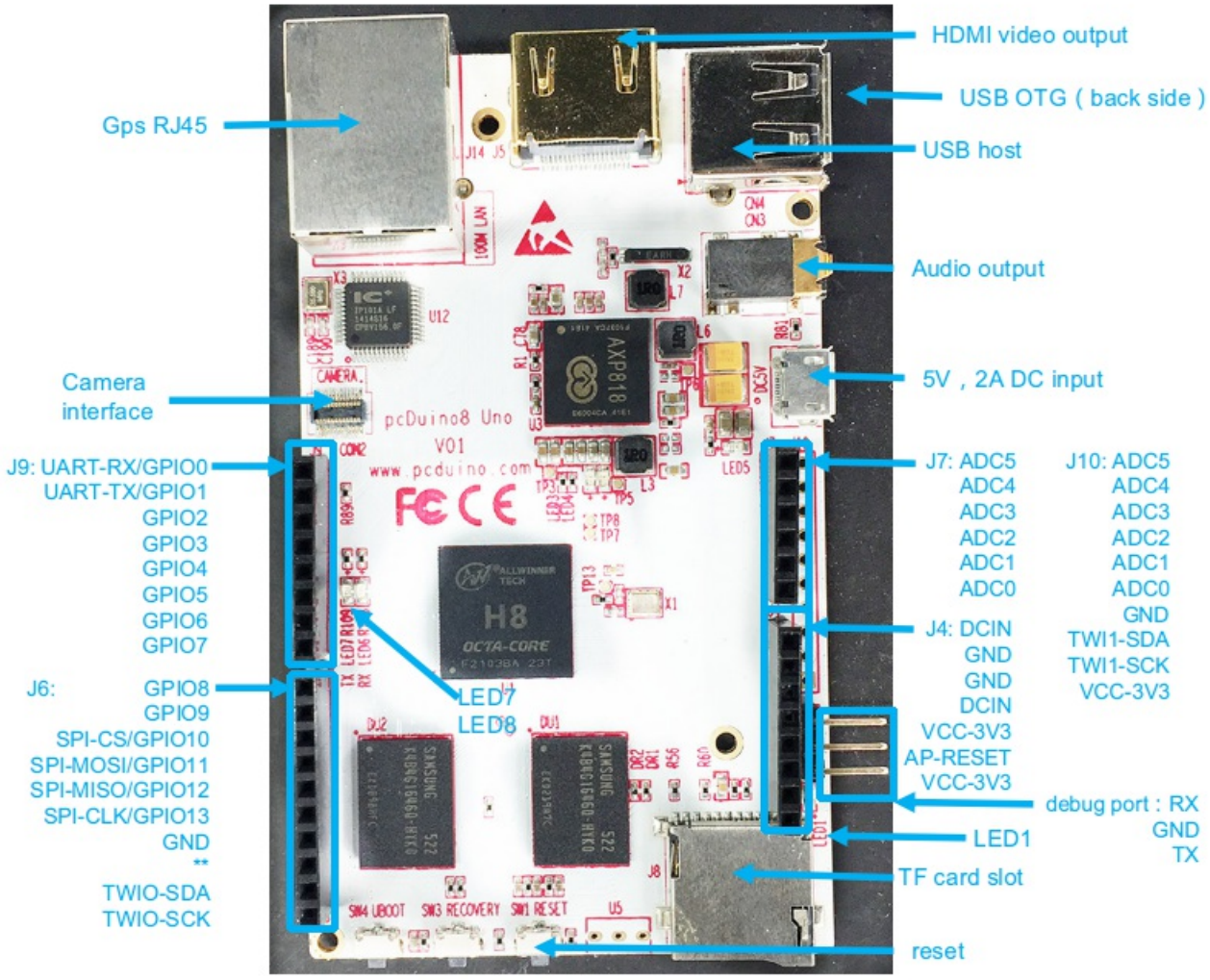
1. Open the VNC Viewer, input the: 192.168.100.1 (which is the IP address of the pcDuino8 Uno) in the server:



1. pcDuino desktop will successfully show at the VNC from OTG.

Arduino Programming Quick Start

pcDuino8 Uno is pin to pin compatible with Arduino so that existing Arduino shields can be installed on pcDuino and many Arduino libraries can run in pcDuino programming environment.

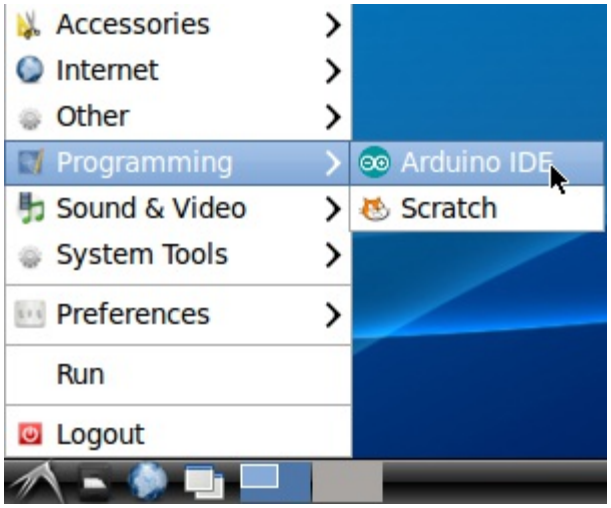


pcDuino8 Uno Image has Arduino IDE built-in.This tutorial will tell how to program GPIO using Arduino IDE.

Steps(Arduino IDE)

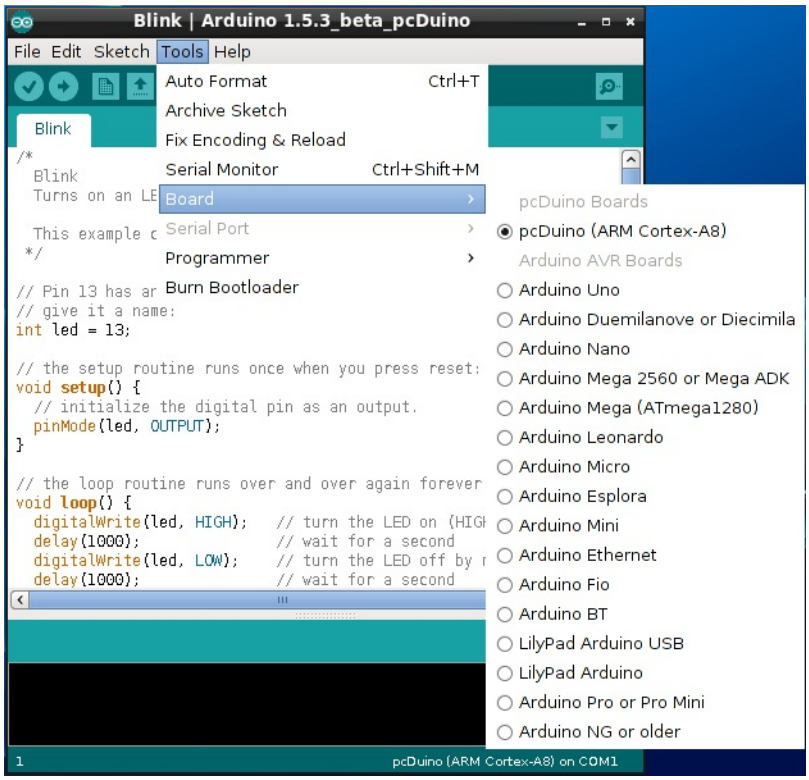
1. Open Arduino IDE

Click start menu and open **Programming --> Arduino IDE**.

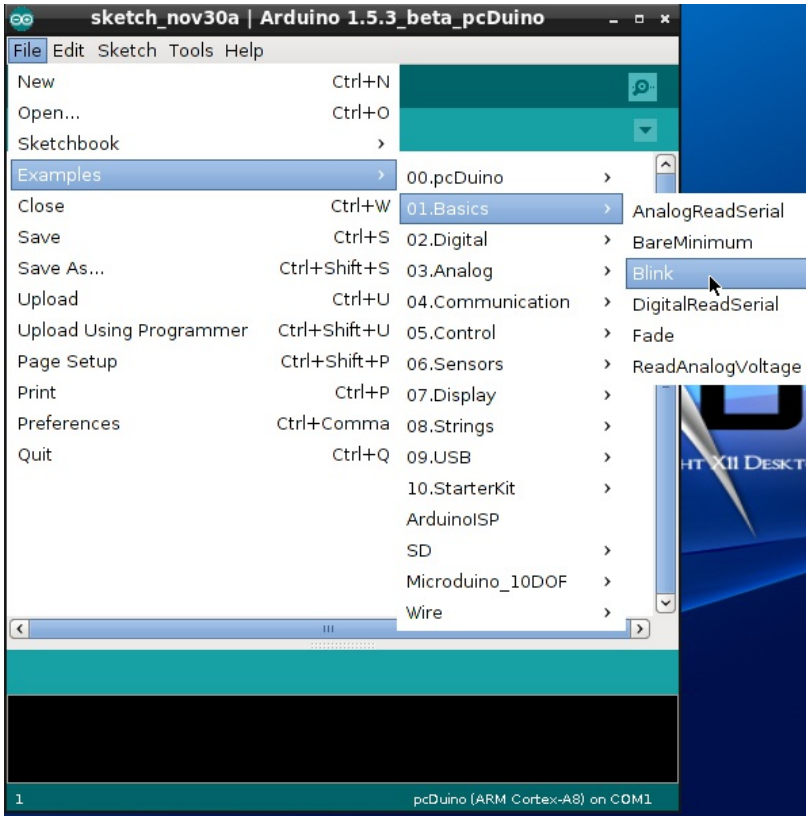


2. Compile and run the example code

When you open the Arduino IDE, you can find a new board type 'pcDuino'.



Take a blink sketch as example. From Menu bar: **File --> Examples --> 01.Basics --> Blink**.



Click this button  to compile and run the source code. Please check the status of LED1, is it blinking?

Note: The compiled binary file is saved at **/home/linaro/Arduino**. So you can manually run this file on Linux terminal.

If you want to create a new design, just enter `Ctrl+n`, and open a window to input source code.

This is the source code, if you have used Arduino before, you will notice that the programming language has no big difference between Arduino and pcDuino.

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Python Programming GPIO Quick Start

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python is pre-installed in the pcDuino Ubuntu image. The sample Python code can be downloaded from: https://github.com/pcduino/python-pcduino

The following steps will tell you how to use Python to control pcDuino8 Uno GPIO.

Steps

1. Install necessary tools

```
$ sudo apt-get -y install git vim
```

2. Download sample python code

```
$ git clone https://github.com/pcduino/python-pcduino.git
```

3. Control LED7

```
$ cd python-pcduino
$ python python-pcduino/Samples/blink_led/blink_led.py
```

Note:**GPIO13 controls LED1, GPIO19 controls LED6 and GPIO18 controls LED7.**

4. Read source code

```
#!/usr/bin/env python
# blink_led.py
# gpio test code for pcduino ( http://www.pcduino.com )
#
import gpio
import time

led_pin = "gpio18"

#define ms delay
def delay(ms):
    time.sleep(1.0*ms/1000)

#set GPIO as output port
def setup():
    gpio.pinMode(led_pin, gpio.OUTPUT)

#loop
def loop():
    while(1):
        gpio.digitalWrite(led_pin, gpio.HIGH)
        delay(200)
        gpio.digitalWrite(led_pin, gpio.LOW)
        delay(100)

def main():
    setup()
    loop()

main()
```

5. Use button to control LED

Connect a button to GPIO0 port and read the button's status as input signal, then control the LED on or off.

Based on the given source code, just add several lines to implement this kind of function.

```
#!/usr/bin/env python
# blink_led.py
# gpio test code for pcduino ( http://www.pcduino.com )
#
import gpio
import time

led_pin = "gpio18"
btn_pin = "gpio0"

#define ms delay
def delay(ms):
    time.sleep(1.0*ms/1000)

#set GPIO18 as OUTPUT port
#set GPIO0 as INPUT port
def setup():
    gpio.pinMode(led_pin, gpio.OUTPUT)
    gpio.pinMode(btn_pin, gpio.INPUT)

#loop
def loop():
    while(1):
        # read GPIO0 status
        if gpio.digitalRead(btn_pin) :
            gpio.digitalWrite(led_pin, gpio.HIGH)    #turn on
        else :
            gpio.digitalWrite(led_pin, gpio.LOW)     #turn off

def main():
    setup()
    loop()

main()
```

How to use Webcam

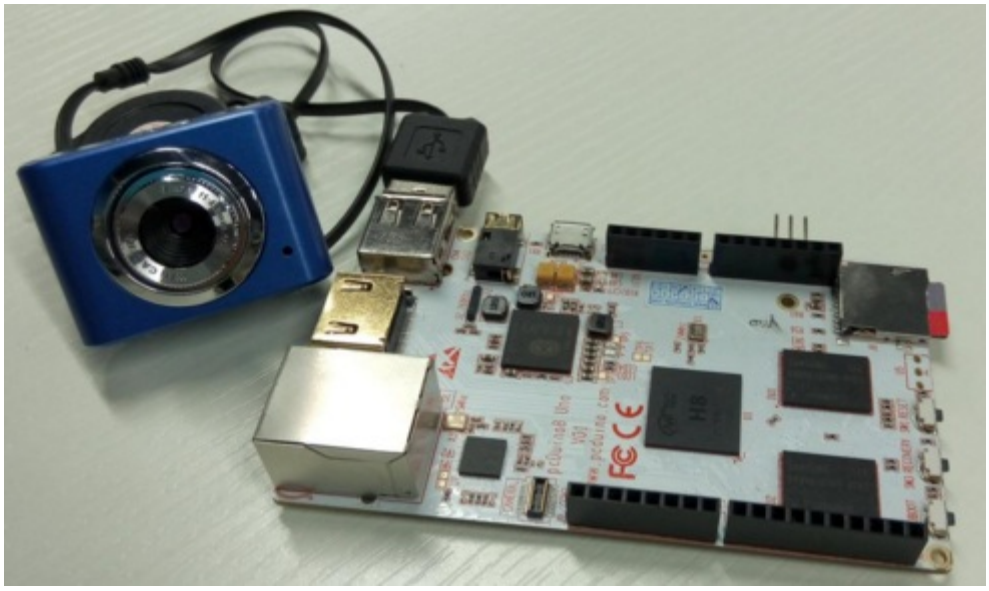
In OpenCV computer vision kit, there is a mini webcam for robot to stream video in real time. It is UVC and can be used with pcDuino8 Uno without installing Linux driver. Many software in Ubuntu can capture the stream video from this webcam, like guvcview, VLC, ffmpeg, and OpenCV.

This tutorial will introduce how to use guvcview to open the webcam.

Steps

Connect webcam

Plug the webcam into pcDuino8 Uno, and power on.



Open a Linux terminal and run command as follows to make sure the webcam could be recognized as /dev/video0.

```
$ ls /dev/video0
```

Open guvcview

The guvcview is a pre-installed software in Ubuntu for pcDuino8 Uno. Run command as follows:

```
$ guvcview
```

If you want to capture the stream video as image or video, you just click the button **Cap. Image** or **Cap. Video**.

Demo Guide

This section will introduce some interesting demos on OpenCV computer vision kit:

1. Camera DIY
2. Face detection
3. Motion detection
4. Network video monitoring

Camera DIY

As mentioned before, we can use guvcview to capture the stream video as a image. So is it possible to DIY a simple camera?

Sure, and it is very easy!

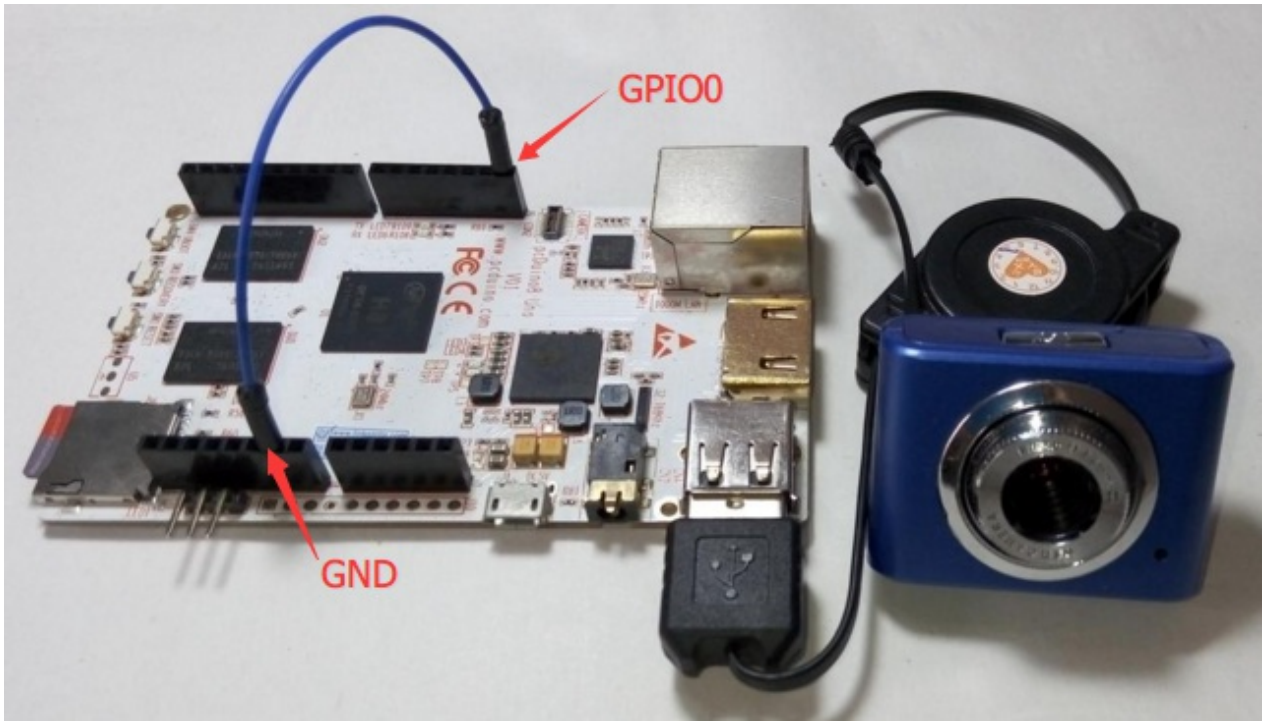
Pre-requisites:

- Hardware
  - OpenCV computer vision kit
- Software
  - Ubuntu 14.04
  - Python 2.7.6
  - OpenCV 2.4.11

Steps

1. Connect

To make it simple, I just connect GPIO0 to GND via a Dupont line. Insert or pull out the line to simulate pressing or releasing a button.



Note: you should know the status that described as the following table.

Connection	Status	Logic
Pull out from GND port, without loading	High level	1
Insert to GND port	Low level	0

2. Run

```
$ git clone https://github.com/pcduino/pcduino8-uno-guide
$ cd pcduino8-uno-guide/demo/1.CameraDIY/
$ python camera.py
```

- The window show the video from webcam in realtime.
- Pull out the Dupont line from GND port and insert into GND port again.
- The photo will be saved to the current directory.
- View the photo.

```
$ gpicview
```

Source code

```
import argparse
import datetime
import time
import cv2
import gpio

btn_pin = "gpio0"
gpio.pinMode(btn_pin, gpio.INPUT)
camera = cv2.VideoCapture(0)
time.sleep(0.25)

while True:
    #read the button status
    current_st = gpio.digitalRead(btn_pin)
    (grabbed, frame) = camera.read()

    if not grabbed:
        break

    cv2.putText(frame, datetime.datetime.now().strftime("%A %d %B %Y %I:%M:%S%p"),
        (10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0, 255), 1)

    cv2.imshow("video",frame)
    next_st = gpio.digitalRead(btn_pin)

    #check the button status has been changed
    # if yes, save as a image to local
    if current_st ^ next_st :
        name = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
        cv2.imwrite(name+".jpg", frame, [int(cv2.IMWRITE_JPEG_QUALITY),100])

    key = cv2.waitKey(1)
    if key == ord("q"):
        break

camera.release()
cv2.destroyAllWindows()
```

Face Detection in Python Using pcDuino8 Uno and Webcam

Introduction

OpenCV is the most popular library for computer vision. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android.

OpenCV uses machine learning algorithms to search for faces within a picture, and break the task of identifying the face into thousands of smaller, bite-sized tasks. These tasks are called classifiers.

OpenCV uses cascades to implement face detection. Though the theory seems complicated, in practice it is very easy to use.

Thanks to Shantnu Tiwari who give us an example that just has several lines of python code to implement face detection using a webcam.

This post , I follow his steps to do face detection on pcDuino8 Uno.

Pre-requisites:

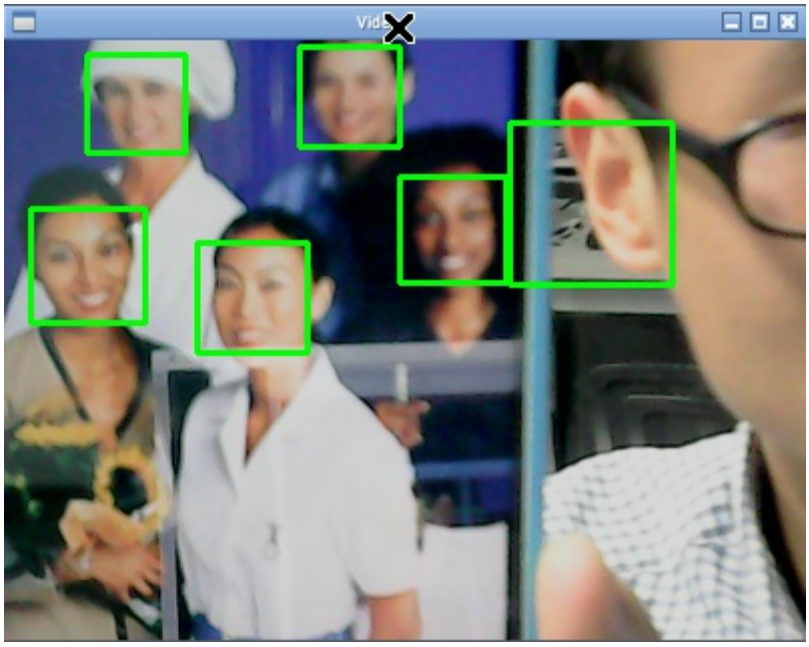
- Hardware
  - OpenCV computer vision kit
- Software
  - Ubuntu 14.04
  - Python 2.7.6
  - OpenCV 2.4.11

Download the source code

```
$ git clone https://github.com/shantnu/Webcam-Face-Detect
$ python webcam.py haarcascade_frontalface_default.xml
```

Note: **Make sure the webcam is recognized as /dev/video0**

I use my mobile phone to show a picture with many faces, the detection result is not bad, although it shows that there are six faces, only five faces are true. Please check the following picture.



Understand the source code

I add some comments in the source code to help understanding the source code.

```
import cv2
import sys

cascPath = sys.argv[1]
faceCascade = cv2.CascadeClassifier(cascPath)

# set the video source to webcam which is /dev/video0
video_capture = cv2.VideoCapture(0)

while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()

   # Convert color to gray, because face detection don't need color
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #Do face detection
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=cv2.cv.CV_HAAR_SCALE_IMAGE
    )

    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    # Display the resulting frame with rectangle
    cv2.imshow('Video', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# When everything is done, release the capture
video_capture.release()
cv2.destroyAllWindows()
```

If you have intesting to try this demo, you can the reference[1].

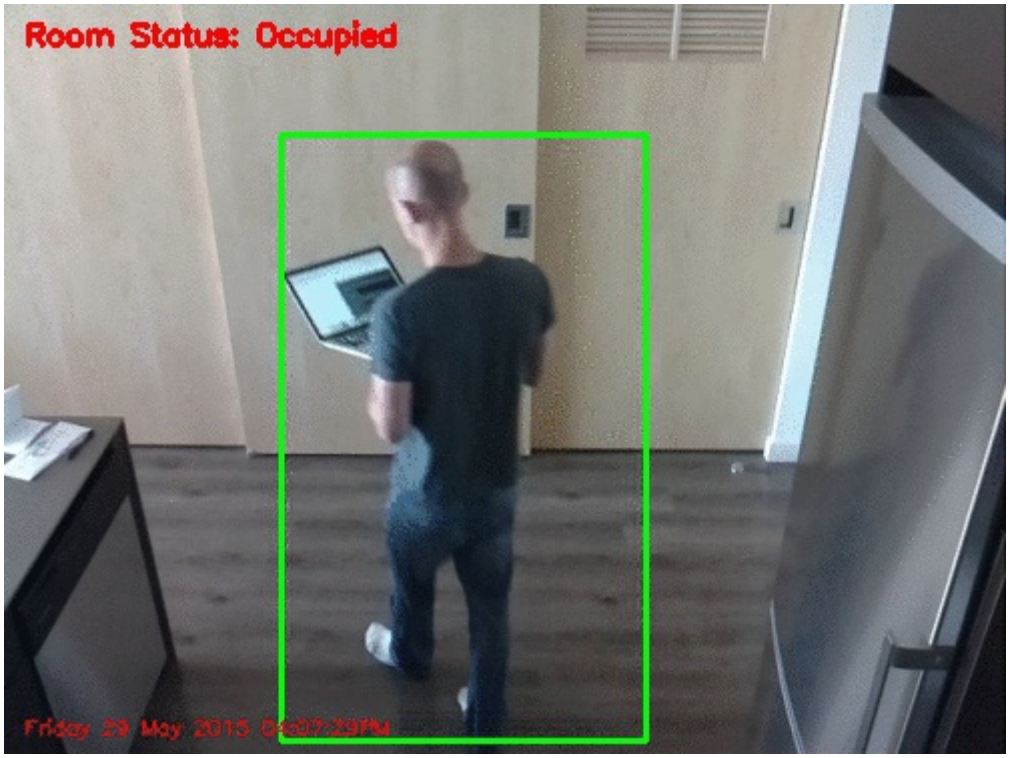
Reference

1. Face Detection in Python Using a Webcam

Motion detection

Introduction

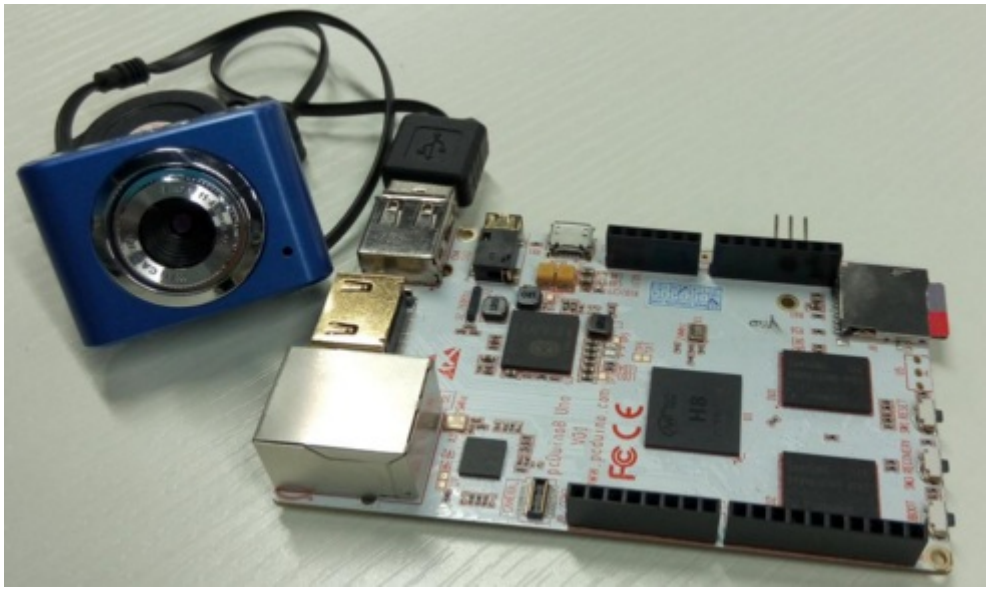
Adrian Rosebrock posted a blog , named " Home surveillance and motion detection with the Raspberry Pi, Python, OpenCV, and Dropbox". This project is very cool, it can take a picture if there are things moving , then upload to Dropbox.



I can't help thinking about how to port this project onto pcDuino8 Uno. Yes, this is not very hard!

Pre-requisites:

- Hardware
  - OpenCV computer vision kit



- Software

- Ubuntu 14.04
- Python 2.7.6
- OpenCV 2.4.11
- imutils

Steps

1. Download source code

Based on the code that Adrian Rosebrock shared, I remove the function that can upload the pictures to Dropbox. Instead, I save these pictures to the local. You can get this source code from github.

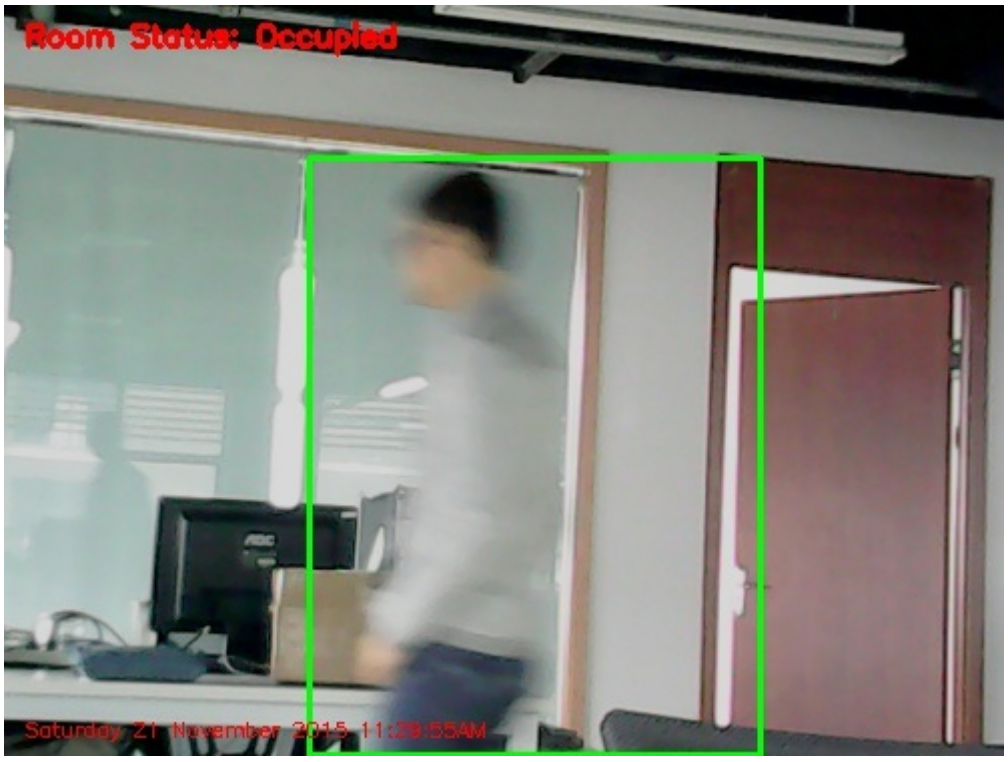
```
$ git clone https://github.com/YaoQ/motion-detection-with-opencv
```

2. Connect webcam

Plug the webcam input pcDuino8 Uno, and power on. the webcam could be recognized as /dev/video0.

3. Test

```
$ python motion-detector.py -c conf.json
```



If you use different types of webcams or want to change the interval time to save a picture, and so on, you should modify these parameters in **conf.json**.

How to use pcDuino8 Uno to implement Network Video Monitoring

If you have pcDuino8 Uno which is connected to Internet, and also, you have a webcam. Then you can build a network video monitoring system, just several minutes.

Let's start.

Prerequisites

1. Hardware

- OpenCV computer vision kit

2. Software

- motion

Steps

1. Install software

```
$ sudo apt-get update
$ sudo apt-get install motion
```

2. Modify configuration

```
$ sudo vim /etc/motion/motion.conf
```

The following parameters need to be confirmed：

> daemon on #default off (This allows the motion to run in the background)
>
> framerate 10 #default 2 (increased framerate)
>
> width 640 #default 320 (changed width to match that of the webcam)
>
> height 480 #default 240 (same as above but for height)
>
> threshold 2000 #default 1500 (*explained in detail below)
>
> pre_capture 2 #default 0 (captures 2 frames before motion was detected and adds that to the videos to make them smoother)
>
> post_capture 5 #default 0 (same as above but captures frames after)
>
> output_normal off #default on (this disables storing images, since we only require video)
>
> ffmpeg_video_codec msmpeg4 #default swf (msmpeg4 is accepted by windows media player, hence easier to play on Windows)
>
> target_dir /mnt/motionvideos #default /tmp/motion (changed the directory where videos will be stored)
>
> webcam_maxrate 5 #default 1 (increase the max framerate on lie stream)
>
> webcam_localhost off #default on (allows you to set up a live stream of the webcam)

Note: you need change some parameters above which are based on what kind of webcam you choose!

3. Create a directory

According to the configuration of motion, we should create a directory where videos will be stored：

```
sudo mkdir /mnt/motionvideos
sudo chown motion /mnt/motionvideos
```

4. Change the default configuration，start motion daemon

```
sudo vi /etc/default/motion
```

change from :

> start_motion_daemon= **no**

to:

> start_motion_daemon= **yes**

5. Plug the USB webcam

To make sure the webcam has been recognized as /dev/video0 file.

```
$ ls /dev/video0
```

6. Start service

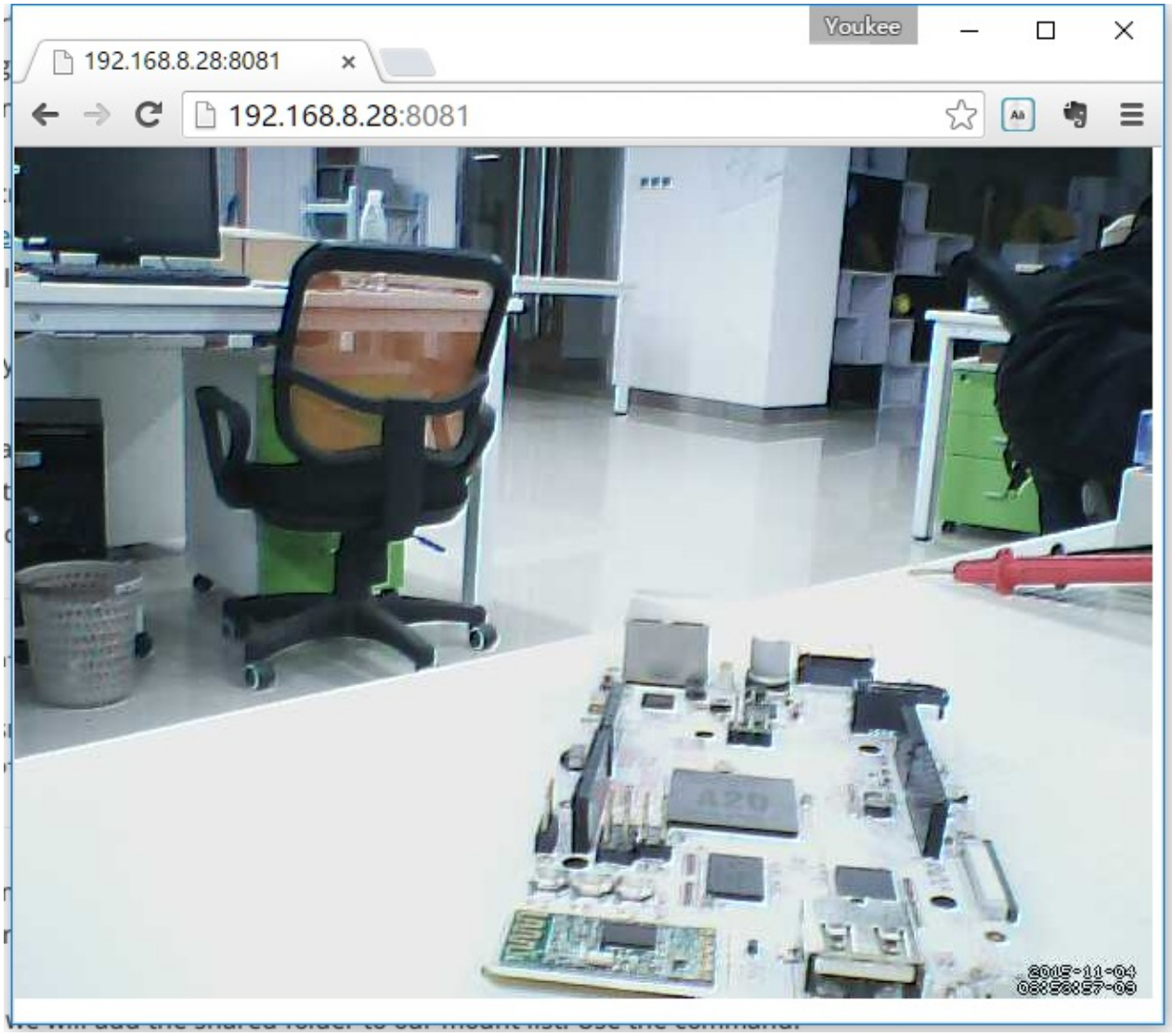
```
linaro@linaro-alip:~$ sudo service motion start
 * Starting motion detection daemon motion       [ OK ]
```

If you want to turn off the motion service, you can run: `sudo service motion stop`

Test

First to get the pcDuino8 Uno IP address, you should open a Linux terminal and run `ifconfig` to read the information of network, especially the IP address. Then, open an browser on your PC or mobile devices(Smart Phone or tablet), which are in the same LAN. Enter the pcDuino8 Uno IP Address, like: **< pcDuino8 Uno IP Address>:8081**

Now, you can watch the video from your webcam via network.



More configuration parameters of motion ,you can check here。

The steps are very easy, isn't it?