

EE5414 Course Mini Project Report

(BeagleBone Black)

Wangchen DAI (53623708)

Jingwei HU (53656463)

Instructor: Dr. L L CHENG

Department of Electronics Engineering

City University of Hong Kong

October 28, 2014

CONTENTS

I	Introduction	2
II	Hardware Description and Implementation	2
III	Software Description and Implementation	4
III-A	Operating System Installation and Configuration	4
III-B	Wireless Adapter Configuration	5
III-C	Web Server Setup	7
III-D	LED Test	7
III-E	Web Camera Configuration and Functioning	7
III-F	SQL Server Setup and Application	9
IV	Conclusions	9
References		10

I. INTRODUCTION

An embedded system is a computer system with a specific function and embedded in a mechanical or electrical system. Due to the limitation of processing resources, lower power consumption, smaller size, lower cost, and simpler operating function are the main properties of embedded computers when compared with the general ones. Embedded systems are usually based on microcontrollers or microprocessors, such as MCU, ARM, DSP and FPGA. In this project, an ARM based circuit board, BeagleBone Black (BBB), is applied as the hardware development tool. An Ubuntu14.04 terminal system, which is known as a kind of open source Linux system, is loaded into the ARM7 chip as the software development platform. Using this specified embedded system, a series of functions are developed including: LED Test, Image/Video Capture, Web Server and SQL Server Establishment.

II. HARDWARE DESCRIPTION AND IMPLEMENTATION

BBB is a low-cost, community-supported hardware platform for embedded application development. BBB board mainly contains an ARM Cortex A8 series processor, a 512MB DDR3 RAM memory, an onboard 2GB MMC chip, and some other necessary peripherals. In this project, the BBB is developed with following cables and devices:

- A MiniUSB Cable, which can be used to connect the board with PC and served as power source (limited to 500mA), network, and serial port for data transmission;
- An external DC power supply with a minimum of 1A current output;
- An USB Hub to expand the onboard USB Host port from one to four;
- An USB portable 802.11N wireless adapter;
- A Logitech HD Pro C920 Web Camera, which is a USB webcam and provides full HD 1080p video recording in wide-screen at a maximum rate of 30 frames per second;
- A mini SD card and its corresponding USB portable card reader.

The detailed and connection information of all hardware cables and devices is provided in both Fig.1, for block diagram view, and Fig.2 for real view.

Both of the figures indicate the hardware setup details. First, the BBB board is connected to a PC using the MiniUSB Cable. Via the cable, the BBB board could not only receives

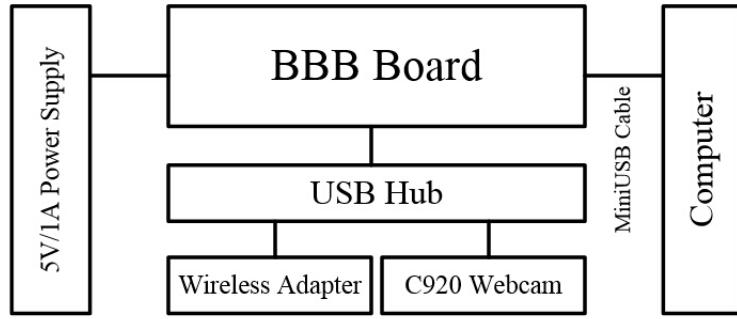


Fig. 1. Block diagram of hardware description.

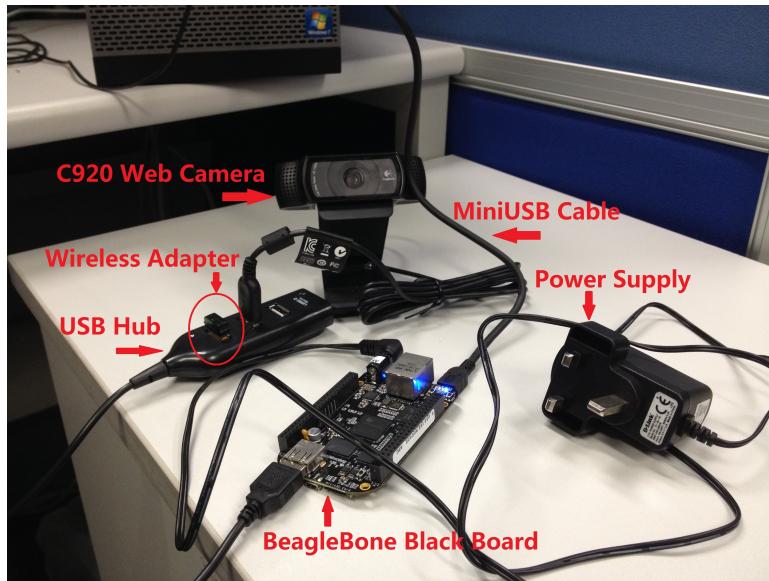


Fig. 2. Real view of hardware description.

a 5V/500mA power supply provided by PC's USB port, but also accesses to the Internet by setting up Windows ICS, in addition, users could operate Linux instructions through specific Windows-based software clients. Second, in case of unexpected system shutdown issue caused by current exceeding (current supply using MiniUSB Cable is set at a maximum of 500mA), an external power supply is used. Third, an USB Hub is connected to the onboard host USB port for expanding the port from one to four. Finally, the Logitech C920 webcam is connected to the USB Hub with its attached USB cable, together with a USB portable wireless adapter. Note that in this project, the BBB board could have an alternative way to access the Internet: getting

access to a WiFi or a Hotspot connection via the wireless adapter; and the system could have two different IP address for Ethernet and wireless connection, respectively.

III. SOFTWARE DESCRIPTION AND IMPLEMENTATION

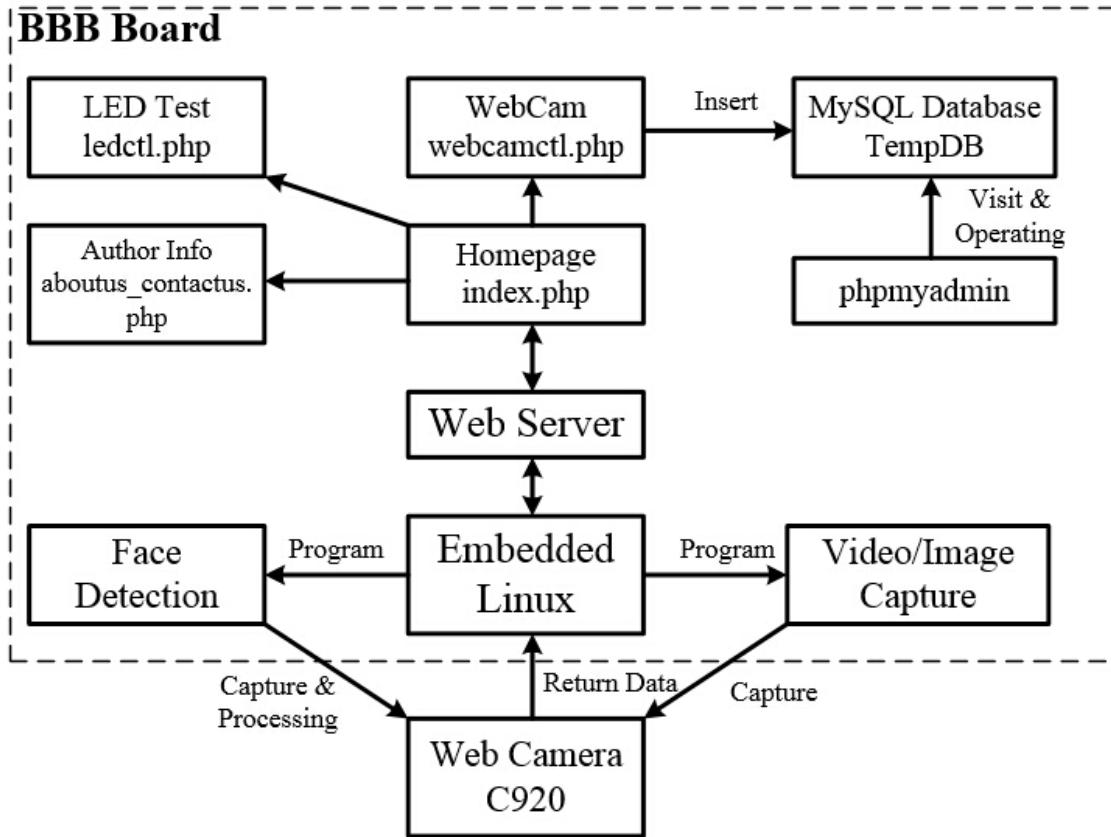


Fig. 3. Block diagram of software description.

In this project, the software implementation includes installation, configuration as well as application of embedded OS, Web server, PHP, and MySQL; setup and website-based control of onboard LEDs and the C920 Web Camera. Fig.3 provides a block diagram of software description.

A. Operating System Installation and Configuration

In our implementation, we install Ubuntu Trusty 14.04 on BBB. This image uses the Ubuntu 14.04 core filesystem from Ubuntu with the minimal meta package applied. The kernel is

compiled from the mainline Linux kernel git repository. The result is an easy-to-install and stable Linux image that works well with the BBB boards.

First, OS binaries “ubuntu-trusty-14.04-rootfs-3.14.4.1-bone-armhf.com.tar.xz” can be downloaded from <http://www.armhf.com/boards/beaglebone-black/>; After uncompressed this file, we can get the Ubuntu Trusty 14.04 image as shown in Fig.4. Next, this image should be burn into a microSD card such that we can install Ubuntu 14.04 onto BBB. One way of doing this task is by utilising Win32DiskImager, which is a powerful tool automatically creating a OS boot disk and available at <http://sourceforge.net/projects/win32diskimager/files/latest/download>. In Fig.5(b), Win32DiskImager is starting to write Ubuntu.img to the microSD card, as presented by ‘device’ F:. Please refer to the getting started page on the BeagleBone official website [2] to find more details on the OS installation

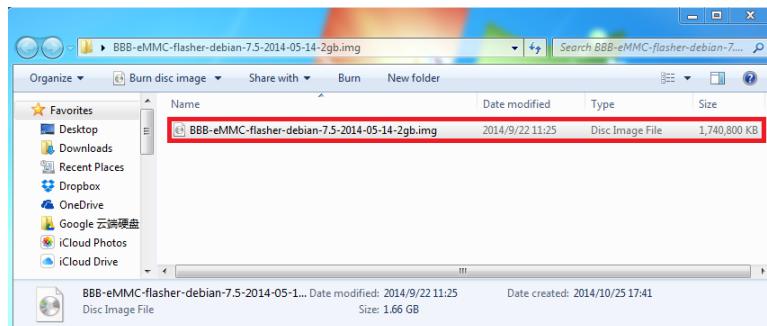


Fig. 4. Ubuntu Trusty 14.04 Image.



Fig. 5. Creating A Ubuntu bootable image on microSD card.

When the microSD Card is now ready to boot, we insert it into BBB board, hold down the USER/BOOT button (in our board, it is S3 button.) and apply power, either by the USB cable or 5V adapter. However, it is strongly recommended to plug the 5V adapter in all the time because power adapter can provide more stable voltage than USB cable and thus it is less likely to have unexpected problems caused by power cutoff or insufficient.

Keep holding down the button until the bank of 4 LED's light up for a few seconds and now release the button. It will take anywhere from 30-45 minutes to flash the image onto the on-board chip. Once it's done, the bank of 4 LED's to the right of the Ethernet will all stay lit up at the same time. We can then power down your BBB [4]. One important thing worth mentioning is that after finishing this step, we do not need microSD card installation anymore — you can plug it out and whenever you power the board again, Ubuntu is able to automatically boot from MMC, the inner storage component on chip.

B. Wireless Adapter Configuration

Once the OS is successfully set up on board, it becomes critical to have Internet access — We need to install extra system utilities or download third-party libraries to support this project. So in this section, we detail how to setup a WiFi connection for BBB.

To begin with, we must connect the components needed into BBB: wireless network adapter, USB hub and miniUSB cable (See Fig.6). The wireless network adapter is the key for Internet connectivity. Besides this adapter, a webcam is also required by this project but we only have one USB port for them. We therefore use a USB hub to install both components. We also need a miniUSB cable to connect BBB to the host PC such that we can remote control BBB through host PC's keyboard. Or to be more precise, we start a SSH connection to BBB through this cable.

To establish SSH connection, we must log on BBB via SSH protocol and this could be done with the help of SSH tool. In fact, there are a variety of SSH GUI tools available for Windows. In this report, we take Secure Shell Client available on <http://mepopedia.com/forum/file.php?793,file=1103,filename=SSHSecureShellClient-3.2.9.exe,download=1> as an example because this tool integrates scp command that allows us to transfer files between the host PC and the BBB



(a) Wireless Network Adapter

(b) Usb Hub

(c) MiniUSB Cable

Fig. 6. Essential components for WiFi connection

freely.

When the wireless network adapter is plug into BBB and BBB itself is also connected to one USB port on our host PC, we open the Secure Shell Client and start a SSH connection from PC to BBB as shown in Fig.7. Remember the default IP address for BBB is 192.168.7.2 and the default user account is ‘ubuntu’ with password ”temppwd”.

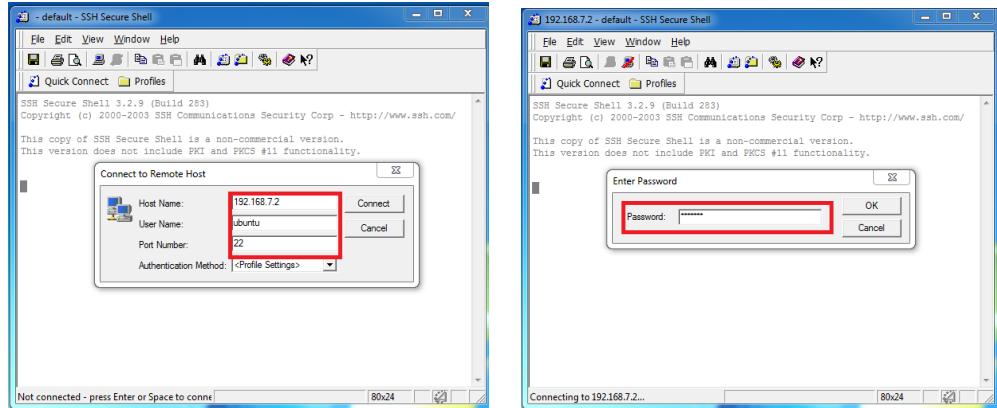


Fig. 7. SSH log on via Secure Shell Client for Windows.

Finally, we configure the WiFi setup by modifying the network configuration file located at /etc/network/interfaces. In figure 8 we present the changes that have been made marked in red box. The idea of the is to connect to CityU Campus WiFi network — ”Universities WiFi” with the student EID and password. For safety reason, the EID and password are wiped out and readers can use their own account to fill in.

```

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
# Example to keep MAC address between reboots
#hwaddress ether DE:AD:BE:EF:CA:FE

# The secondary network interface
#auto eth1
#iface eth1 inet dhcp

# WiFi Example
auto wlan0
iface wlan0 inet dhcp
#WPA2-PSK setting begins
#wpa-driver wext
#wpa-ssid ITISWIFI
#wpa-ap-scan 1
#wpa-proto WPA RSN
#wpa-pairwise CCMP TKIP
#wpa-group CCMP TKIP
#wpa-key-mgmt WPA-PSK
#wpa-psk ecadcaf0cfffa84b0a7a3bdede7cc371fe11ea226c9ee54e04718f305037a30
#WPA2_PSK setting ends

#WPA Enterprise setting begins
wpa-driver wext
wpa-ssid Universities WiFi
wpa-ap-scan 1
wpa-proto RSN
wpa-pairwise CCMP TKIP
wpa-group CCMP TKIP
wpa-eap PEAP
wpa-key-mgmt WPA-EAP
wpa-identity [REDACTED]
wpa-password [REDACTED]
# ... or on host side, usbn0 and random hwaddr
# Note on some boards, usbn0 is automatically setup with an init script
iface usbn0 inet static
    address 192.168.7.2
    netmask 255.255.255.0
    network 192.168.7.0
    gateway 192.168.7.1
~.
~.
~.

```

Fig. 8. Modify /etc/network/interfaces to establish a WiFi connection to access the Internet.

To make the modifications into effect, we type in "ifconfig wlan0 down && ifconfig wlan0 up" from Secure Shell Client to restart the wireless adapter. Then we type in *pingwww.google.com.hk* to test whether the WiFi connection is functioning. Fig.9 shows that the network setup is correct with all 4 transmitted packets getting received.

One last reminder is that we suggest to log in as 'root' rather than 'ubuntu'. This is primarily because 'ubuntu' does have full access to all system resources and in many cases, you would get errors like 'permission denied'. Just switch to 'root' and all these annoying errors disappear.

```

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.

Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.8.13-bone63 armv7l)

 * Documentation: https://help.ubuntu.com/
Last login: Wed Oct 22 20:29:57 2014 from 192.168.7.1
root@arm:~# ping www.google.com.hk
PING www.google.com.hk (173.194.127.111) 56(84) bytes of data.
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=1 ttl=56 tim
e=5.69 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=2 ttl=56 tim
e=5.87 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=3 ttl=56 tim
e=5.44 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=4 ttl=56 tim
e=7.93 ms
^C
--- www.google.com.hk ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 5.445/6.236/7.938/0.998 ms
root@arm:~#

```

Connected to 192.168.7.2 SSH2 - aes128-cbc - hmac-md5 - nc | 80x24

Fig. 9. A successful ping to Google.

C. Web Server Setup

Apache together with PHP5 are used to establish the Web server. Apache can be installed on Ubuntu.

```
apt-get install apache2
```

Then, direct the browser to “<http://192.168.7.2>” or “http://*wireless IP address*”, and the Apache placeholder page should appear: It Works! Note that the default document of placeholder page, or it can be called as homepage, can be found and edited at:

```
vi /var/www/index.php
```

In order to let Ubuntu understand PHP code, PHP5 and the Apache PHP5 modules must be installed. By installing both packages

```
apt-get install php5 libapache2-mod-php5
```

and restarting the Apache service

```
/etc/init.d/apache2 restart
```

PHP5 is configured and can be used for Web server developments. Please refer to [1] for detailed Web server configuration and PHP testing examples.

After setting up the Web server, we could build our own main webpage by replacing the default *index.php* file mentioned previously, with a modified one. Here, a CityU style webpage is designed and replaces the original one as the project main webpage, it can be accessed by typing either IP address of the board, see Fig.10. The template source code is provided by CityU of Hong Kong. All the sub pages' source code is stored under the path */var/www/php/*, this indicates that visiting any sub websites, you should direct the browser to “http://192.168.7.2/php/*source file name*”.

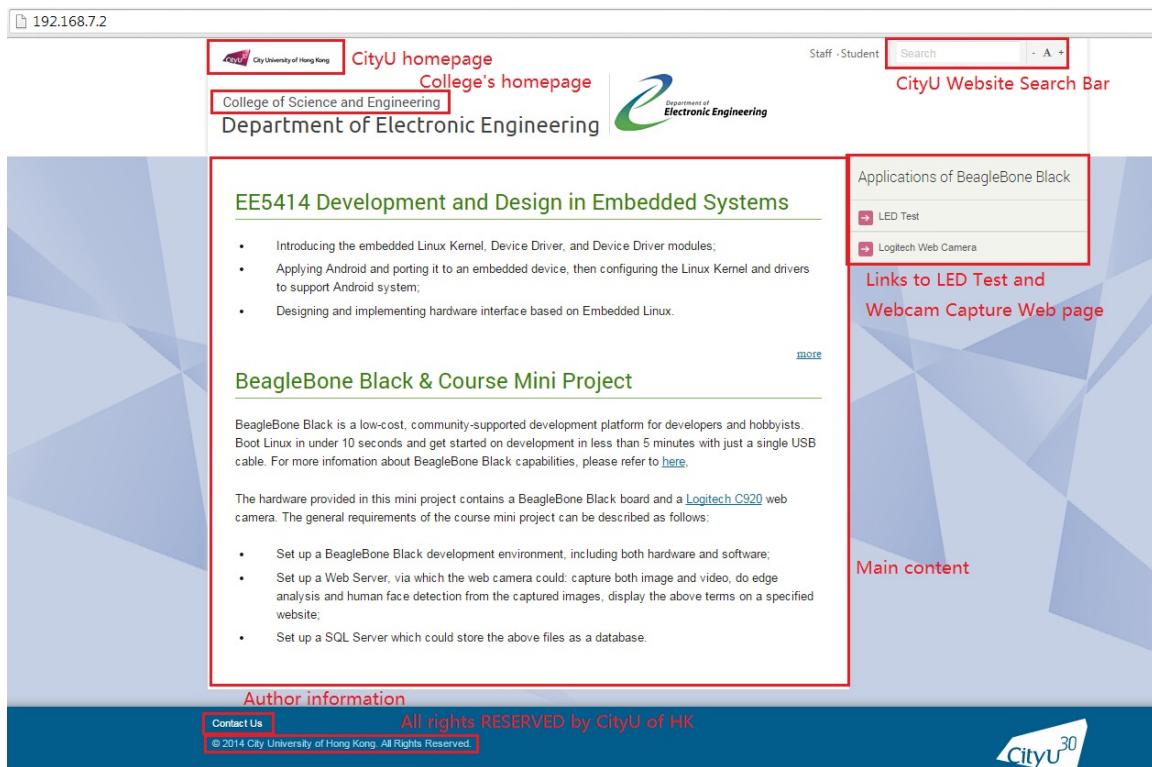


Fig. 10. Project main webpage.

D. LED Test

The LED Test program is testing the four LED lights located near the Ethernet port and the Reset Button. In general, each LED light has two configuration files: one controls the trigger mode:

```
echo none > /sys/class/leds/beaglebone:green:usr/?/trigger
```

“none” for passive mode and it can be changed to “heartbeat” for blink mode; the other file controls the On/Off switch of the light:

```
echo 1 > /sys/class/leds/beaglebone:green:usr/?/brightness
```

1 for light on, and 0 for light off. By default, some of the LEDs are used to display information to users about what is going on. So if the states those LEDs changes, they will be quickly overwritten. That is the reason why the trigger mode must be set before the stage of light changes in passive mode. And for blink mode, only the first command affects. Note that character “?” in the above paths can be 0, 1, 2, and 3. Here gives the example of LED_0 control shell script file.

```
#!/bin/bash
case "$1" in
'on')
echo "LED_0 is turned on"
sudo bash -c "echo none > /sys/class/leds/beaglebone:green:usr0/
trigger"
sudo bash -c "echo 1 > /sys/class/leds/beaglebone:green:usr0/
brightness"
;;
'off')
echo "LED_0 is turned off"
sudo bash -c "echo none > /sys/class/leds/beaglebone:green:usr0/
trigger"
sudo bash -c "echo 0 > /sys/class/leds/beaglebone:green:usr0/
brightness"
```

```

;;
'heartbeat')
echo "LED_0_is_blinking"
sudo bash -c "echo heartbeat > /sys/class/leds/beaglebone:green:
usr0/trigger"
;;
esac

```

Similarly, the shell script can be written to control all four LEDs. An example of the control command:

```
/var/www/LED/ledctl.sh on off on heartbeat
```

By calling this shell script in the *.php* file,

```

<?php
...
$cmd = '/bin/bash ../LED/ledctl.sh $status0 $status1 $status2
$status3';
...
?>

```

the website based LED test can be implemented. See Fig.11: Fig.11(a) indicates LED_0 and 2 are switched on, LED_1 is switched off, and LED_3 is blink mode.

E. Web Camera Configuration and Functioning

In this subsection, a webcam controller is implemented on our Apache web server. A *webcam-ctl.php* file is coded for basic functions like photographing, video capturing and face detection. The corresponding web page can be accessed by typing "192.168.7.2/webcamctl.php" in the address bar of your web browser if BBB is connected to the host PC. A snapshot of this *.php* file and its web page is shown in Fig.12.

It is suggested to use OpenCV library to achieve these functionalities because this library is cross-platform, which focuses mainly on real-time image processing and is free for use under

192.168.7.2/php/ledctl.php
EE5414 Mini Project Home Page

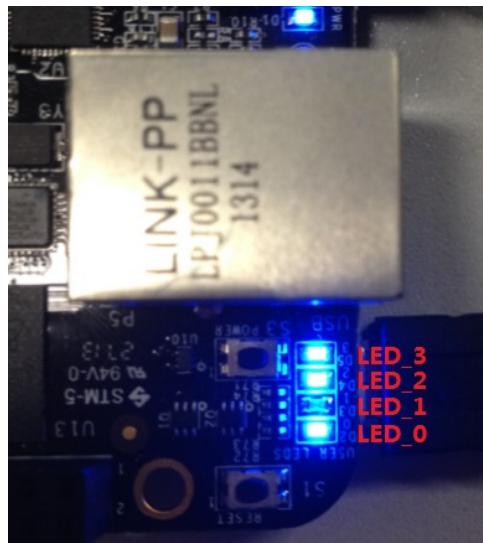
LED Controller

LED_0 HeartBeat ON OFF
 LED_1 HeartBeat ON OFF
 LED_2 HeartBeat ON OFF
 LED_3 HeartBeat ON OFF

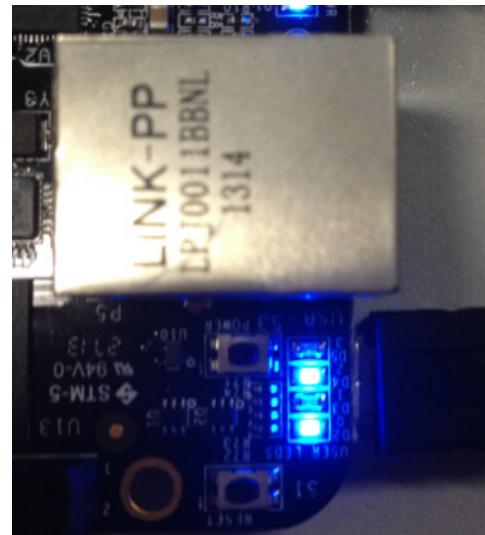
What you have done:

LED_0 is turned on.
 LED_1 is turned off.
 LED_2 is turned on.
 LED_3 is blinking.

(a)



(b)



(c)

Fig. 11. Web based LED test.

the open source BSD license. On top of this, OpenCV is pre-installed by default on our Ubuntu Trusty 14.04 operating system and thus it is not necessary to take pains learning how to immigrate OpenCV into BBB board. For photographing, it is optional to enable face detection function or not. Besides, it automatically detects the edge of this picture by using image edge detection method; For video capturing, we also provide with 4 options for the capture duration: 4 seconds, 8 seconds, 12 seconds and 16 seconds. We present below in Fig.13 the demo of this work by

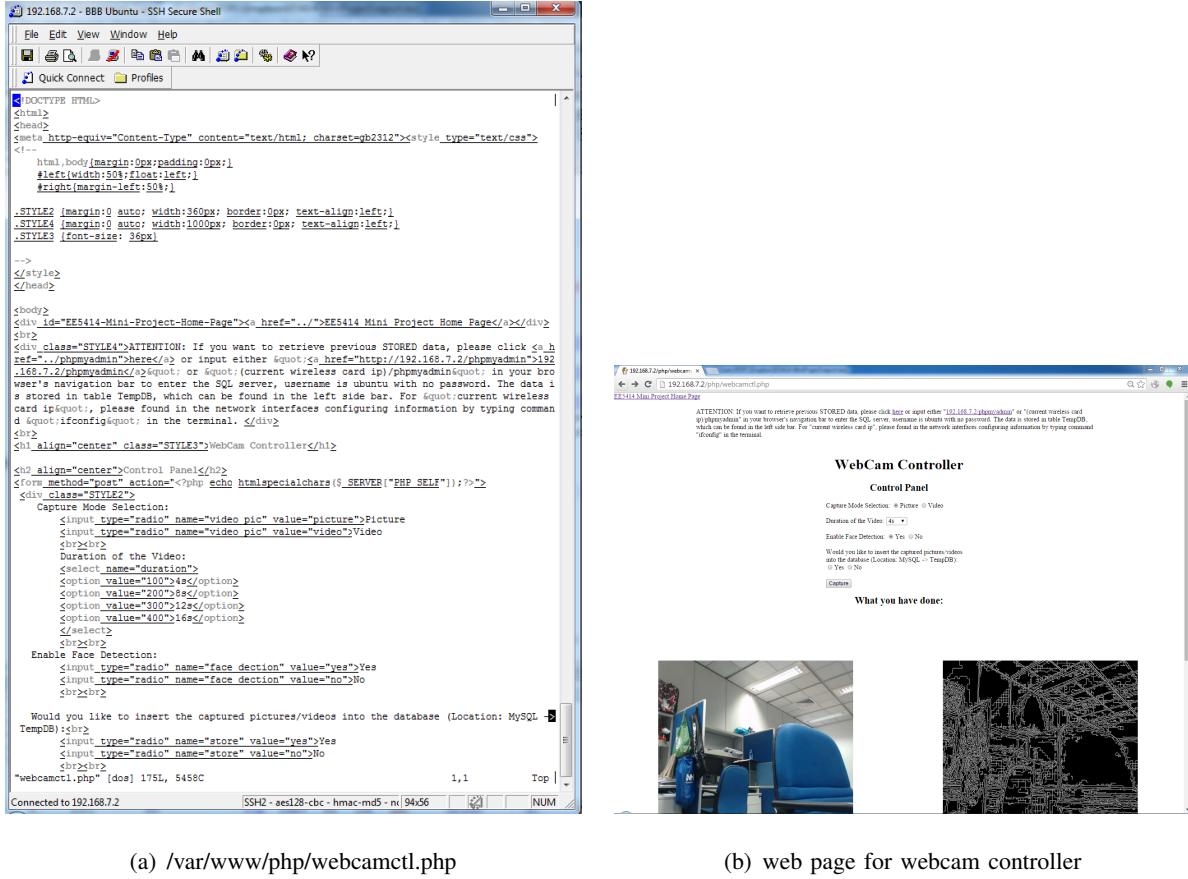


Fig. 12. PHP File Configuration and Web Page

detecting the author himself.

The details about the implementation of photographing and video capturing is introduced by Derek Molloy in his personal website [3]. Please refer to <http://derekmolloy.ie/beaglebone/beaglebone-video-capture-and-image-processing-on-embedded-linux-using-opencv/> and <http://stackoverflow.com/questions/20757147/opencv-and-cdetect-face-in-image> for more details.

F. SQL Server Setup and Application

IV. CONCLUSIONS

ACKNOWLEDGMENT

We would like to thank our colleague Sean Yang for his kindly help with the WiFi setup on BBB board.

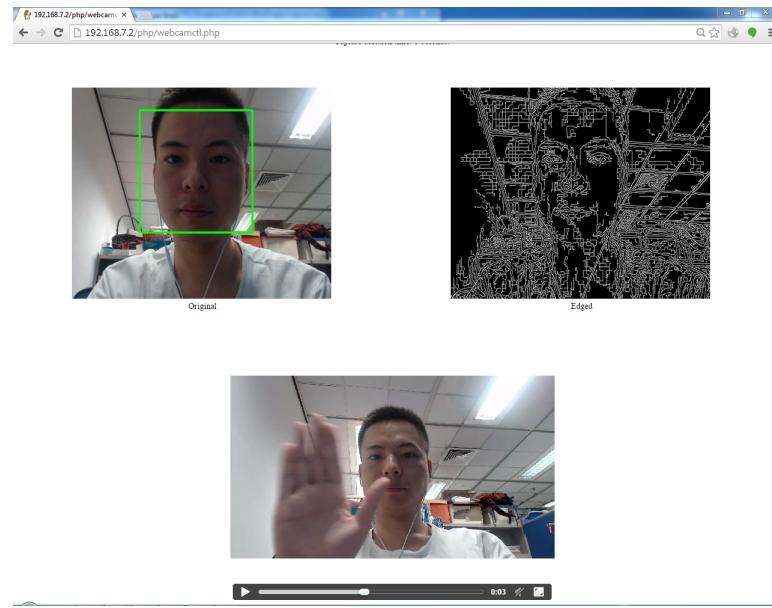


Fig. 13. Demonstrate for the webcam controller

REFERENCES

- [1] Feb, 2012. Falko Timme, <http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-ubuntu-12.04-lts-lamp>.
- [2] June, 2014. BeagleBone getting started, <http://www.beagleboard.org/getting-started>.
- [3] June, 2014. derekmolloy.ie, <http://derekmolloy.ie/tag/beaglebone-black/>.
- [4] May, 2013. Flashing the BeagleBone Black, <http://www.synopsys.com>.