



CITY UNIVERSITY OF HONG KONG

DEPARTMENT OF ELECTRONICS ENGINEERING

---

## EE5414 Course Mini Project Report

---

*Author:*

Wangchen DAI (53623708)

Jingwei HU (53656463)

*Instructor:*

Dr. L.L.CHENG

October 31, 2014

## CONTENTS

<b>I</b>	<b>Introduction</b>	2
<b>II</b>	<b>Hardware Description and Implementation</b>	2
<b>III</b>	<b>Software Description and Implementation</b>	4
III-A	Operating System Installation and Configuration . . . . .	4
III-B	Wireless Adapter Configuration . . . . .	6
III-C	Web Server Setup . . . . .	8
III-D	LED Test . . . . .	9
III-E	Web Camera Configuration and Functioning . . . . .	11
III-F	SQL Server Setup and Application . . . . .	12
<b>IV</b>	<b>Conclusions</b>	15
<b>Acknowledgment</b>		15
<b>References</b>		15
<b>Appendix</b>		16
A	An example of LED_0 control shell script file . . . . .	16
B	An example of how to insert a file into a SQL database . . . . .	16

## 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. Ubuntu Trusty 14.04, which is known as a version of open source Linux system, is loaded into the ARM 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 device for embedded application development. BBB board mainly contains an ARM Cortex A8 series processor, a 512MB DDR3 RAM memory, an onboard 2GB MMC chip, with some other necessary peripherals. In this project, the BBB board 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 5V and a minimum of 1A current output;
- A USB Hub to expand the onboard USB Host port from one to four;
- A 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 a

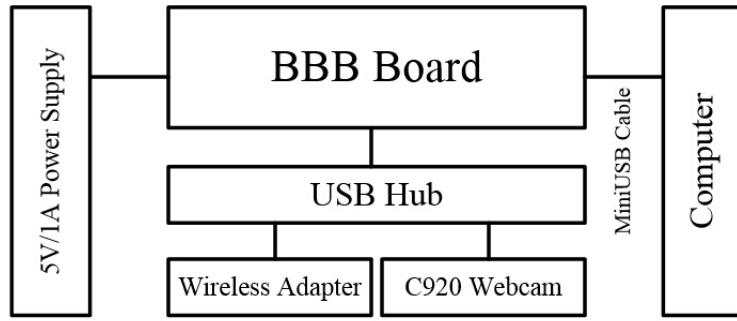


Fig. 1. Block Diagram of Hardware Description.

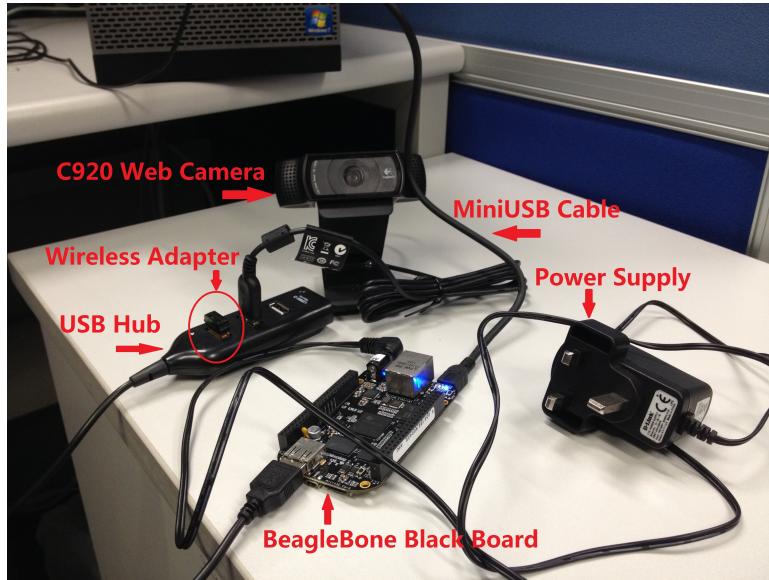


Fig. 2. Real View of Hardware Description.

5V/500mA power supply provided by PC's USB port, but also be accessible from a host PC and share the Internet connection by setting up Windows ICS [1]; 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, a 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, system could have two different IP addresses for Ethernet and wireless connection, respectively, also the BBB board could have an alternative way to access the Internet: getting access to a WiFi or Hotspot connection via the wireless adapter.

### III. SOFTWARE DESCRIPTION AND IMPLEMENTATION

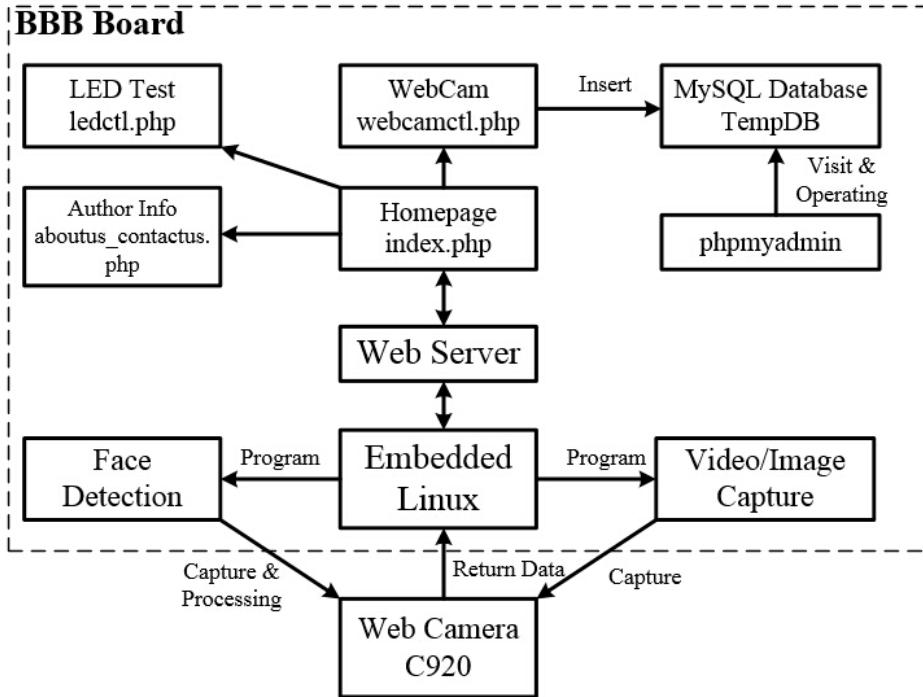


Fig. 3. Block Diagram of Software Description.

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 and thus results in 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 [7] to find more details on the OS installation.

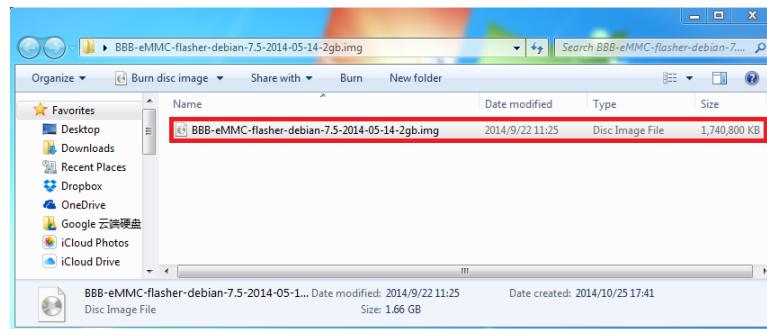
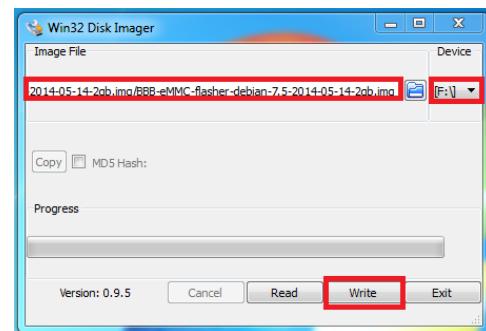


Fig. 4. Ubuntu Trusty 14.04 Image.



(a) MicroSD Card.



(b) Win32DiskImager.

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 then 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 the BBB [9]. One important thing worth mentioning is that after finishing this step, we do not need microSD card for 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 set up 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 web camera 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 the Secure Shell Client available on <http://mepopedia.com/forum/file.php?793,file=1103,filename=SSHSecureShellClient-3.2.9.exe,download=1>. Because of the integration of *scp* command, this tool allows us to transfer files between the host PC and the BBB 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



(a) Wireless Network Adapter.

(b) Usb Hub.

(c) MiniUSB Cable.

Fig. 6. Essential components for WiFi connection.

to BBB as shown in Fig.7. Remember the default Ethernet IP address for BBB is 192.168.7.2 and the default user account is ‘ubuntu’ with password ”temppwd”.

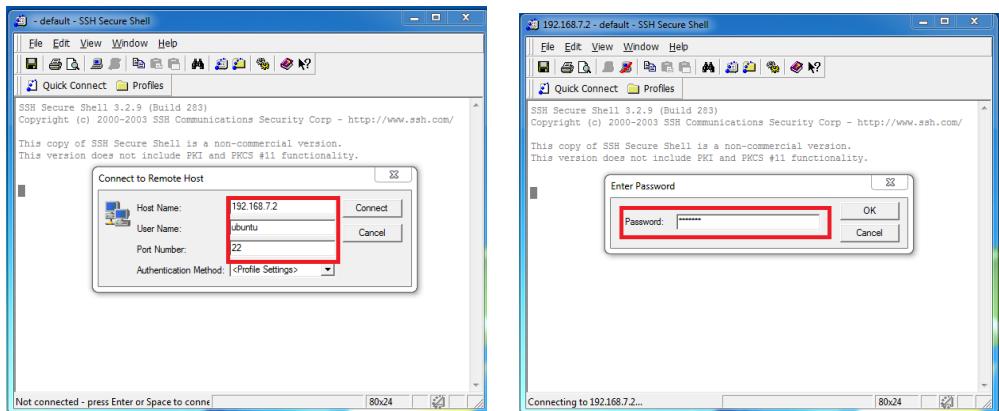


Fig. 7. SSH log on via Secure Shell Client for Windows.

Finally, we configure the WiFi setup by adding the following commands in the network configuration file located at /etc/network/interfaces:

```
#WiFi Example
auth wlan0
iface wlan0 inet dhcp
#WPA Enterprise setup
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 *yourEID*@my.cityu.edu.hk
wpa-password *yourPASSWORD*
```

The idea of this setup is to connect to CityU Campus WiFi network — "Universities WiFi" with the student EID and password.

To make the modifications into effect, we type in the following command from Secure Shell Client to restart the wireless adapter:

```
ifconfig wlan0 down && ifconfig wlan0 up
```

By typing following command, the function of WiFi connection can be verified:

```
ping www.google.com.hk
```

Fig.8 shows that the network setup is correct with all 4 transmitted packets getting received.

```
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 time=5.69 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=2 ttl=56 time=5.87 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=3 ttl=56 time=5.44 ms
64 bytes from hkg03s12-in-f15.1e100.net (173.194.127.111): icmp_seq=4 ttl=56 time=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:~#
```

Fig. 8. A Successful Ping to Google.

### C. Web Server Setup

Apache together with PHP5 is used to establish the web server. Apache can be installed on Ubuntu by typing the following command in the terminal:

```
apt-get install apache2
```

Then, direct the browser to “<http://192.168.7.2>” or “[http://\\*wireless IP address\\*](http://*wireless IP address*)”, and the Apache placeholder page should appears: 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
```

PHP is configured and can be used for web server developments. Please refer to [6] 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 homepage, it can be accessed by typing the IP address of the board, see Fig.9. The template source code is provided by CityU of Hong Kong. All the subpages’ source code is stored under the path */var/www/php/* on BBB, which indicates that visiting any sub websites, you should direct the browser to “[http://192.168.7.2/php/\\*source file name.php\\*](http://192.168.7.2/php/*source file name.php*)”.

#### D. LED Test

The LED Test program could test 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, for example:

```
echo none > /sys/class/leds/beaglebone:green:usr0/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:

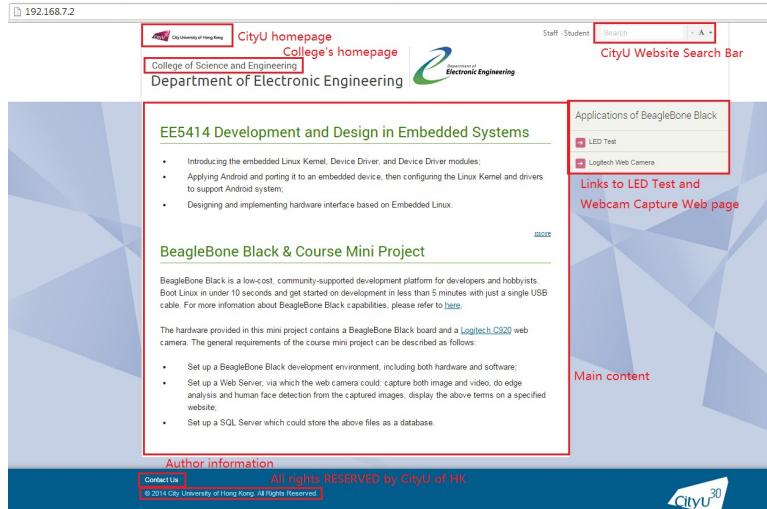


Fig. 9. Project Homepage.

```
echo 1 > /sys/class/leds/beaglebone:green:usr0/brightness
```

1 for light on, and 0 for light off. By default, some of the LEDs are set to a kind of trigger mode which could display information to users about what is going on. Such trigger mode must be set to passive mode before the state of light changes. As for blink mode, only the first command affects. Note that “usr0” in the above paths can be changed to usr1, 2, and 3. Appendix A provides an example of LED\_0 control shell script. By modifying Appendix A, a shell script to control all four LEDs can be obtained. Such file is named as *ledctl.sh* and an example of the all-four LEDs control command is given:

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

By calling this shell script in the *ledctl.php* file:

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

the website based LED test can be implemented. For example, Fig.10 indicates LED\_0 and 2 are switched on, LED\_1 is switched off, and LED\_3 is blink mode.

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.

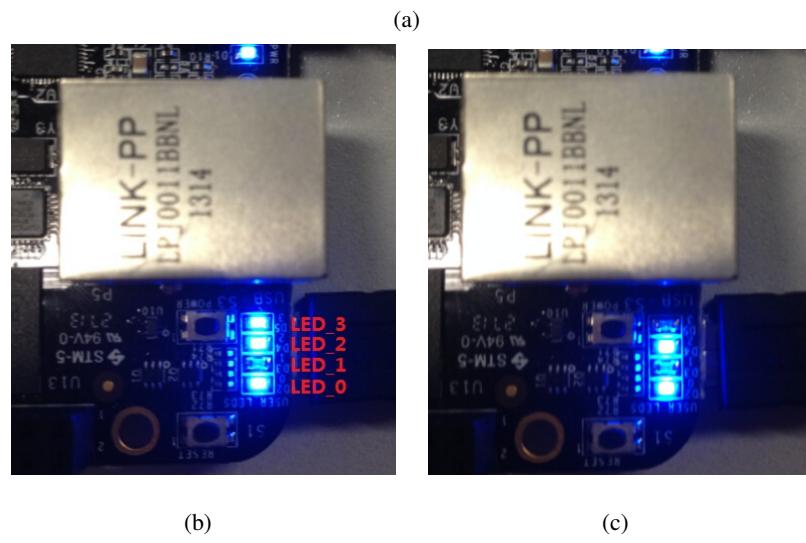


Fig. 10. Web Based LED Test.

### E. Web Camera Configuration and Functioning

A Logitech HD Pro Webcam C920 [2], which is capable of delivering high-definition 1080p resolution and supports H.264 encoded compression to provide high quality video at low bit rates, is applied in this project, See Fig.11(a). As previously stated in Section III-B, the camera is connected to BBB via the USB hub due to the limited number of the onboard USB ports.

In order to have a friendly interactive web camera operation, a controller is implemented and embedded on the web page “<http://192.168.7.2/php/webcamctl.php>”. The *webcamctl.php* file is coded for basic functions like photographing, video capturing and face detection. The

corresponding web page can be accessed by simply copy the above address into the web browser's navigation bar when BBB is connected to the host PC. A snapshot of its web page is shown in Fig.11(b).

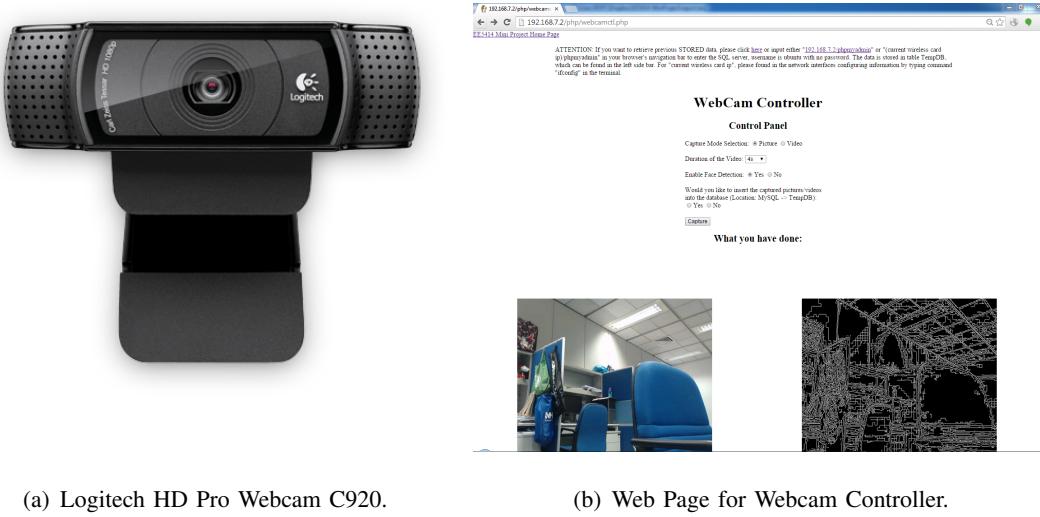


Fig. 11. C920 Web Camera and Web Page.

It is suggested to use OpenCV library to achieve these functionalities due to its cross-platform library, which mainly focuses on real-time image processing and is free for use under the open source BSD license. On top of this, OpenCV is pre-installed by default on our Ubuntu Trusty 14.04 embedded system, so that OpenCV can be used directly without considering the porting issues. For photographing, we offer an option to enable face detection function or not. Besides, it could automatically detects the edge of this picture by using image edge detection method. For video capturing, the program provides 4 different capture duration options: 4s, 8s, 12s and 16s, where s denotes to second. Fig.12 presents the demo of this work by detecting the author himself.

The details about the implementation of photographing and video capturing are introduced by Derek Molloy in his personal website [8]. Please refer to [3], [4] for more information.

#### *F. SQL Server Setup and Application*

First installing the MySQL packages, and getting MySQL support in PHP:

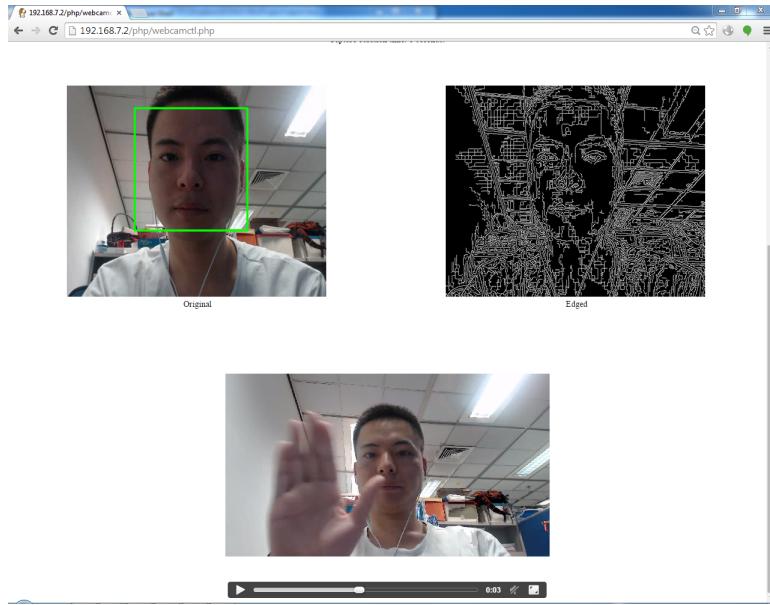


Fig. 12. Demonstrate for the webcam controller.

```
apt-get install mysql-server mysql-client php5-mysql
```

Then restart Apache service:

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

phpMyAdmin provides a web interface through which the MySQL databases can be easily managed:

```
apt-get install phpmyadmin
```

A link must be established between phpMyAdmin and Apache:

```
ln -s /usr/share/phpmyadmin /var/www
```

Finally, by tying “<http://192.168.7.2/phpmyadmin>” in the browser, and login with the username and password, the MySQL databases can be accessed with a graphical interface (See Fig.13). Through which we could add, delete, set and insert data into databases.

After the configuration procedure, the SQL onboard is ready for use. In this project, the main idea we use to insert captured video and image files into the database (TempDB) is: first, three .cpp files are generated, in which some MySQL functions are included, such as

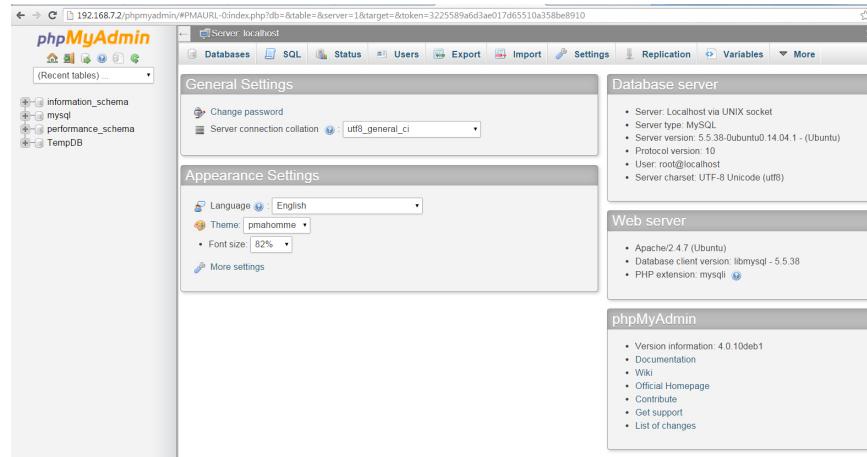


Fig. 13. phpMyAdmin interface.

The screenshot shows the phpMyAdmin browse interface for the TempDB database. The left sidebar lists databases (information\_schema, mysql, performance\_schema, TempDB) and tables (New, Capture\_Edges, Capture\_Image, Capture\_Video, TestTable). The main area shows the Capture\_Image table with the following data:

	id	TimeStamp	data
<input type="checkbox"/>	1	2014-10-22 21:00:27	[BLOB - 331.7 KiB]
<input type="checkbox"/>	2	2014-10-22 21:12:32	[BLOB - 279.2 KiB]

Below the table are buttons for Edit, Copy, Delete, Change, and Export. The bottom navigation bar includes Show (start row 0, number of rows 30, headers every 100), Sort by key (None), and Query results operations (Print view, Print view (with full texts), Export, Display chart).

Fig. 14. Inserted data in the tables of TempDB, MySQL.

*mysql\_real\_connect()* and *mysql\_query()*, in order to login to MySQL [5] and insert specified file (video, image, edge) into the right table of database TempDB (see Appendix B as an example); then compile this files using g++ compiler, taking *video.cpp* as an example:

```
g++ video.cpp -o video -I/usr/include/mysql mysqlclient
```

Finally, calling all these three compiled files in *webcamctl.php*, so that the captured videos and images can be stored into the specified tables in TempDB with a corresponding time stamp through webpage. (See Fig.14)

#### IV. CONCLUSIONS

In this project, we use BBB and embedded ARM-Linux as the hardware and software development tools, respectively. Implementing a series of functions including: Web server and SQL server setup and configuration; webpage based LED Test, Image/Video Capture and Face Detection; and MySQL based data storage functionality.

#### ACKNOWLEDGMENT

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

#### REFERENCES

- [1] Getting Started USB Network Adapter on the Beaglebone,  
<http://derekmolloy.ie/beaglebone/getting-started-usb-network-adapter-on-the-beaglebone/>.
- [2] Logitech HD Pro Webcam C920, <http://www.logitech.com/en-hk/product/hd-pro-webcam-c920>.
- [3] Beaglebone: Video Capture and Image Processing on Embedded Linux using OpenCV, <http://derekmolloy.ie/beaglebone/beaglebone-video-capture-and-image-processing-on-embedded-linux-using-opencv/>.
- [4] opencv and C++:detect face in image, <http://stackoverflow.com/questions/20757147/opencv-and-cdetect-face-in-image>.
- [5] MySQL C API programming tutorial, <http://zetcode.com/db/mysqlc/>.
- [6] Feb, 2012. Falko Timme, [installing-apache2-with-php5-and-mysql-support-on-ubuntu-12.04-lts-lamp](http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-ubuntu-12.04-lts-lamp),  
<http://www.howtoforge.com>.
- [7] June, 2014. BeagleBone getting started, <http://www.beagleboard.org/getting-started>.
- [8] June, 2014. derekmolloy.ie, <http://derekmolloy.ie/tag/beaglebone-black>.
- [9] May, 2013. Flashing the BeagleBone Black, <http://www.synopsys.com>.

## APPENDIX

### A. An 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
```

### B. An example of how to insert a file into a SQL database

```
#include <mysql.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void error_exit(MYSQL *conn) {
    fprintf(stderr, "%s\n", mysql_error(conn));
    if (conn != NULL) {
        mysql_close(conn);
    }
    exit(1);
}

int main() {
    MYSQL *conn = mysql_init(NULL);
    if(conn == NULL) {
        error_exit(conn);
    }

    if (mysql_real_connect(conn, "localhost", "root", "temppwd", "TempDB", 0, NULL, 0) ==
        NULL) {
        error_exit(conn);
    }
}
```

```
}

if (mysql_query(conn, "CREATE TABLE IF NOT EXISTS Capture_Video(id INT PRIMARY KEY
    AUTO_INCREMENT, TimeStamp DATETIME, movie LONGBLOB)")){
    error_exit(conn);
}

mysql_query(conn, "INSERT INTO Capture_Video(TimeStamp, movie) VALUES(NOW(), LOAD_FILE('/
    var/www/WebCam/output.mp4'))");
mysql_close(conn);
return 0;
}
```