

# Webcam and servo motor installation guide

By Group MK2J, Fall 2014 CMPT 433

---

## Environment pre-conditions

- BeagleBone Black running version 3.8.13-bone68
- BeagleBone Black must have connected internet access to download the open source libraries
  - ie) Via Ethernet to Router

## Webcam how-to-setup

### 1. Building x264:

Install the following libraries for webcam:

```
# git clone git://git.videolan.org/x264.git
# cd x264
# ./configure --enable-shared --prefix=/usr
# make
# make install
```

### 2. Building ffmpeg:

```
# git clone git://git.videolan.org/ffmpeg.git
# ./configure --enable-shared --enable-libx264 --enable-gpl
# git remote set-url origin git://source.ffmpeg.org/ffmpeg
# make
# make install
```

### 3. Fixing library problem:

```
# vi /etc/ld.so.conf
```

add "/usr/local/lib" to the file and then type:

```
# ldconfig
```

### 4. v4l2-ctl libraries setup:

```
# sudo apt-get install v4l-utils
```

## 5. Install the imagemagick

```
# sudo apt-get install imagemagick
```

## 6.Run the script

After the library installed and set-up,

In /Webcam\_part/ folder, run the shell script webcam.sh and it will automatically start taking frames in the background.

```
# cd Webcam_part/  
# ./webcam.sh
```

The record.sh script is available for saving the frames for recording purpose.

The build script is to make the compiled c code.

## 7.Run the server

After the webcam setup is finished,

Go to /webServer\_part/, and run “nodejs server.js” to start the server. After this, you can go to url of (ip\_addr):3001 to view the webpage of our security system.

```
# cd webServer_part/  
# nodejs server.js
```

## **Controlling Servo with BeagleBone Black**

### 1.Equipment:

1. BeagleBone Black
2. Pan/Tilt bracket
3. 2x Micro servo
4. Male to Male jumper wires

### 2.Connecting the Servos

1. Connect the ground wire to P9 pin 1
2. Connect the positive wire to P9 pin 3
3. Connect the PWM wire to P9 pin 14
  - Refer to the “Cape Expansion Headers” image below if more than one servo is used
  - Note that the ground wire is usually black or brown and the PWM wire is usually orange or yellow

### 3.Setting up the PWM pin

1. Backup the current Linux bootfile

```
# cd /boot/u-boot
# cp uEnv.txt uEnv.bak
```

## 2. Edit uEnv.txt

```
# nano uEnv.txt
```

- add the following line into the uEnv.txt

```
optargs=quiet drm.debug=7 capemgr.enable_partno=am33xx_pwm,bone_pwm_P9_14
```

- we could also manually do this by using the following commands (Will need to do it every reboot):

```
# echo am33xx_pwm > /sys/devices/bone_capemgr.*/slots
# echo bone_pwm_P9_14 > /sys/devices/bone_capemgr.*/slots
```

- Note that the value for the @ sign is different for each user

## 4. Controlling the Servos

### 1. Initialize servo

```
# cd /sys/devices/ocp.3/pwm_test_P9_14.*
# echo 0 > run
# echo 0 > polarity
# echo 20000000 > period
# echo 1000000 > duty
# echo 1 > run
```

- Note that the value for the @ sign is different for each user

### 3. Move servo

```
# echo 2000000 > duty
```

### 4. Turn off servo

```
# echo 0 > run
```

## Troubleshooting

1. When trying to manually control the servo motor via CLI command in the user space, it does not move!
  - There is a minimum and maximum limit for the `duty` at which the servo motor can move to.

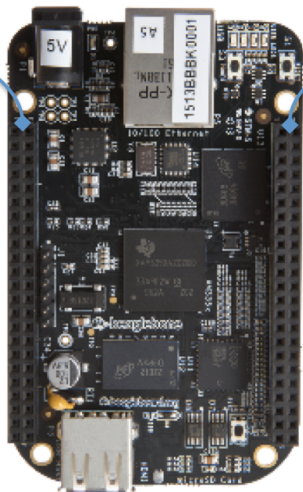
If the number for the `duty` is currently set higher than the maximum limit (2400000), or lower than (600000), then it does not move. (This applies to both left/right, up/down)

- Try setting the duty to be within the range of 600000 and 2400000
2. When we have the frames being taken in “.ppm” format, we spend a lot of time converting the pictures formats so that we can pass it by socket.io and display them on webpage.

# Cape Expansion Headers

P9

|           |    |    |            |
|-----------|----|----|------------|
| DGND      | 1  | 2  | DGND       |
| VDD_3V3   | 3  | 4  | VDD_3V3    |
| VDD_5V    | 5  | 6  | VDD_5V     |
| SYS_5V    | 7  | 8  | SYS_5V     |
| PWR_BTN   | 9  | 10 | SYS_RESETN |
| UART4_RXD | 11 | 12 | GPIO_60    |
| UART4_TXD | 13 | 14 | EHRPWM1A   |
| GPIO_48   | 15 | 16 | EHRPWM1B   |
| SPI0_CS0  | 17 | 18 | SPI0_D1    |
| I2C2_SCL  | 19 | 20 | I2C2_SDA   |
| SPI0_D0   | 21 | 22 | SPI0_SCLK  |
| GPIO_49   | 23 | 24 | UART1_TXD  |
| GPIO_117  | 25 | 26 | UART1_RXD  |
| GPIO_115  | 27 | 28 | SPI1_CS0   |
| SPI1_D0   | 29 | 30 | GPIO_122   |
| SPI1_SCLK | 31 | 32 | VDD_ADC    |
| AIN4      | 33 | 34 | GNDA_ADC   |
| AIN6      | 35 | 36 | AIN5       |
| AIN2      | 37 | 38 | AIN3       |
| AIN0      | 39 | 40 | AIN1       |
| GPIO_20   | 41 | 42 | ECAPPWM0   |
| DGND      | 43 | 44 | DGND       |
| DGND      | 45 | 46 | DGND       |



P8

|            |    |    |             |
|------------|----|----|-------------|
| DGND       | 1  | 2  | DGND        |
| MMC1_DAT6  | 3  | 4  | MMC1_DAT7   |
| MMC1_DAT2  | 5  | 6  | MMC1_DAT3   |
| GPIO_66    | 7  | 8  | GPIO_67     |
| GPIO_69    | 9  | 10 | GPIO_68     |
| GPIO_45    | 11 | 12 | GPIO_44     |
| EHRPWM2B   | 13 | 14 | GPIO_26     |
| GPIO_47    | 15 | 16 | GPIO_46     |
| GPIO_27    | 17 | 18 | GPIO_65     |
| EHRPWM2A   | 19 | 20 | MMC1_CMD    |
| MMC1_CLK   | 21 | 22 | MMC1_DAT5   |
| MMC1_DAT4  | 23 | 24 | MMC1_DAT1   |
| MMC1_DAT0  | 25 | 26 | GPIO_61     |
| LCD_VSYNC  | 27 | 28 | LCD_PCLK    |
| LCD_HSYNC  | 29 | 30 | LCD_AC_BIAS |
| LCD_DATA14 | 31 | 32 | LCD_DATA15  |
| LCD_DATA13 | 33 | 34 | LCD_DATA11  |
| LCD_DATA12 | 35 | 36 | LCD_DATA10  |
| LCD_DATA8  | 37 | 38 | LCD_DATA9   |
| LCD_DATA6  | 39 | 40 | LCD_DATA7   |
| LCD_DATA4  | 41 | 42 | LCD_DATA5   |
| LCD_DATA2  | 43 | 44 | LCD_DATA3   |
| LCD_DATA0  | 45 | 46 | LCD_DATA1   |

**LEGEND**

|                        |
|------------------------|
| POWER/GROUND/RESET     |
| AVAILABLE DIGITAL      |
| AVAILABLE PWM          |
| SHARED I2C BUS         |
| RECONFIGURABLE DIGITAL |
| ANALOG INPUTS (1.8V)   |

image from: <http://rabbit-note.com/2014/08/23/beaglebone-black-power-meter-hard/>