HOWTO Watch Mjpg-stream by Remote Device

2012.02.15 Gavin@mobilogics

Subject

- Environment
  - Hardware
  - Software & tool
- Toolchains of Sourcery G++ Lite
- Bootable SD Card
  - Setup SD in Ubuntu10.10
  - Build a customize uboot
    - Expansion header
- Control GPIO high/low
- Setup Mjpg-Streamer Tool
- Setup Ad-Hoc wifi with Mjpg-Streamer
- Result

Environment

- Hardware
  - beagleboard_xM_C
  - ZyDas zd1211 wireless dongle
  - USB-RS232 cable
  - Mini-HDMI_to_Dvi cable
  - webcam

Environment

- Software & tool
  - ubuntu-10.10-r7-minimal-armel.tar.xz
  - u-boot-2011.12.tar.bz2
  - mjpg-streamer-mjpg-streamer.tar.gz
  - arm-2009q1-203-arm-none-linux-gnueabi.src.tar.bz2

Toolchains of Sourcery G++ Lite

- Download
  - arm-2009q1-203-arm-none-linux-gnueabi.src.tar.bz2
- extract
  - Remember the path in order to set env params
    - e.g ~/toolchains/CodeSourcery/Sourcery_G++_Lite/bin/

Bootable SD Card

-Setup SD in Ubuntu10.10

- Download and extract
  - ubuntu-10.10-r7-minimal-armel.tar.xz
    - Linux omap 2.6.38.4-x3
- Step by step
  - Plug your SD card(2G at least)
  - $cd <path>
  - $dmesg | tail
    - take down your mount point,e.g sdX
  - $sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot beagle
  - $sudo sync
- now,plug your sd card into beagleboard,then boot it

Bootable SD Card

-Setup SD in Ubuntu10.10

```
U-Boot SPL 2011.12-00005-g0a44c98 (Feb 13 2012 - 18:49:45)
Texas Instruments Revision detection unimplemented
OMAP SD/MMC: 0
timed out in wait_for_bb: I2C_STAT=1000
reading u-boot.img
reading u-boot.img


U-Boot 2011.12 (Feb 15 2012 - 14:18:14)

OMAP36XX/37XX-GP ES1.2, CPU-OPP2, L3-165MHz, Max CPU Clock 1 Ghz
OMAP3 Beagle board + LPDDR/NAND
I2C:   ready
DRAM:  512 MiB
NAND:  0 MiB
MMC:   OMAP SD/MMC: 0
*** Warning - readenv() failed, using default environment

In:    serial
Out:   serial
Err:   serial
Beagle xM Rev C
No EEPROM on expansion board
Die ID #721400029ff80000016830c40301a014
Net:   Net Initialization Skipped
No ethernet found.
Hit any key to stop autoboot:  0
OMAP3 beagleboard.org #
```

Bootable SD Card

-Build a customize uboot

- Download uboot source code
  - u-boot-2011.12.tar.bz2
    - For ubuntu10.10
- Step by step
  - $cd <extract path>
  - $export ARCH=arm;
    export CROSS_COMPILE=arm-none-linux-gnueabi-;
    export PATH=$PATH:<your toolchain path>
    - Setup compile env params
  - $make mrproper
  - $make omap3_beagle_config
    - Patch omap3 kernel config
  - $make

Bootable SD Card

-Build a customize uboot

- Expansion header configure
  - $pico ./board/ti/beagle/beagle.h
  - Modify gpio136~139 to PTD
    - Make sure low status what we wanna using them after booting.

```
/*Wireless LAN */\
        MUX_VAL(CP(MMC2_CLK),          (IEN  | PTU | EN  | M4)) /*GPIO_130*/\
        MUX_VAL(CP(MMC2_CMD),          (IEN  | PTU | EN  | M4)) /*GPIO_131*/\
        MUX_VAL(CP(MMC2_DAT0),         (IEN  | PTU | EN  | M4)) /*GPIO_132*/\
        MUX_VAL(CP(MMC2_DAT1),         (IEN  | PTU | EN  | M4)) /*GPIO_133*/\
        MUX_VAL(CP(MMC2_DAT2),         (IEN  | PTU | EN  | M4)) /*GPIO_134*/\
        MUX_VAL(CP(MMC2_DAT3),         (IEN  | PTU | EN  | M4)) /*GPIO_135*/\
        MUX_VAL(CP(MMC2_DAT4),         (IEN  | PTU | EN  | M4)) /*GPIO_136*/\
        MUX_VAL(CP(MMC2_DAT5),         (IEN  | PTU | EN  | M4)) /*GPIO_137*/\
        MUX_VAL(CP(MMC2_DAT6),         (IEN  | PTU | EN  | M4)) /*GPIO_138*/\
        MUX_VAL(CP(MMC2_DAT7),         (IEN  | PTU | EN  | M4)) /*GPIO_139*/\
```

→

```
/*Wireless LAN */\
        MUX_VAL(CP(MMC2_CLK),          (IEN  | PTU | EN  | M4)) /*GPIO_130*/\
        MUX_VAL(CP(MMC2_CMD),          (IEN  | PTU | EN  | M4)) /*GPIO_131*/\
        MUX_VAL(CP(MMC2_DAT0),         (IEN  | PTU | EN  | M4)) /*GPIO_132*/\
        MUX_VAL(CP(MMC2_DAT1),         (IEN  | PTU | EN  | M4)) /*GPIO_133*/\
        MUX_VAL(CP(MMC2_DAT2),         (IEN  | PTU | EN  | M4)) /*GPIO_134*/\
        MUX_VAL(CP(MMC2_DAT3),         (IEN  | PTU | EN  | M4)) /*GPIO_135*/\
        MUX_VAL(CP(MMC2_DAT4),         (IEN  | PTD | EN  | M4)) /*GPIO_136*/\
        MUX_VAL(CP(MMC2_DAT5),         (IEN  | PTD | EN  | M4)) /*GPIO_137*/\
        MUX_VAL(CP(MMC2_DAT6),         (IEN  | PTD | EN  | M4)) /*GPIO_138*/\
        MUX_VAL(CP(MMC2_DAT7),         (IEN  | PTD | EN  | M4)) /*GPIO_139*/\
```

- Rebuild uboot as previous page,result will be a ~320k sized u-boot.img in main directory
- Replace your u-boot.img on sd card(sdX1)

Control GPIO high/low

- After booting into filesystem
  - omap login:ubuntu
  - password :temppwd
- $cd /sys/class/gpio ; ls

```
ubuntu@omap:/sys/class/gpio$ cd /sys/class/gpio/;ls
export     gpiochip128  gpiochip192  gpiochip64  unexport

gpiochip0  gpiochip160  gpiochip32   gpiochip96
```

- $sudo -s
- $echo 136 > export
  - Enable gpio136 what we wanna config it
- $cd gpio136 ; echo high > direction
  - Set gpio136 status to output high
  - Same as above command,you can $echo low > direction to output low

```
root@omap:/sys/class/gpio# ls
export   gpiochip0    gpiochip160  gpiochip32  gpiochip96
gpio136  gpiochip128  gpiochip192  gpiochip64  unexport
```

Setup Mjpg-Streamer Tool

- Download from sourceforge
  - mjpg-streamer-mjpg-streamer.tar.gz
- Step of compiling mjpg-stream tool
  - $export ARCH=arm;
    export CROSS_COMPILE=arm-none-linux-gnueabi-;
    export PATH=$PATH:<your toolchain path>
  - $cd mjpg-streamer
  - $pico Makefile(all Makefiles)
    - Modify DISTDIR=`pwd`
    - Modify CC = arm-none-linux-gnueabi-gcc
  - $make clean all

Setup Mjpg-Streamer Tool(cont.)

- Solve some issues between compileing
  - Lose <jpeglib.h>,”jmorecfg.h”,”jconfig.h” in jpeg_utils.c
  - Lose jpeg library(cannot find -ljpeg)
    - Download jpeg-6b and compile
      - --prefix=<your toolchain/>
    - jpeg-6b will produce 2 folders which named 'include' & 'lib'
    - Copy include/* to <toolchains/libc/usr/include>
    - Copy lib/* to <toolchains/lib/gcc/4.3.3/>

Setup Mjpg-Streamer Tool(cont.)

- Copy compiling directory “mjpg-streamer” to SD card
  - $cp -R mjpg-streamer /media/rootfs/home/ubuntu/
- After booting your ubuntu10.10,install relation package
  - $sudo apt-get update
  - $sudo apt-get upgrade
  - $sudo apt-get install build-essential libv4l-dev imagemagick
- Check your ethernet ip address
- $cd mjpg-streamer
- $pico start.sh
  - Add <path> at top line of script : cd /home/ubuntu/mjpg-streamer/
- $./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -y" -o "./output_http.so -w ./www"
  - -y : YUV (default is Mjpeg)
- Open web-browser,fill in beagleboard ip_addr,then you can receive mjpg stream by ethernet
  - Port 8080

Setup Ad-Hoc wifi with Mjpg-Streamer

- Create user 『ggs』 to run stream server
  - $sudo adduser ggs
- Give ggs has permission to run shell script
  - $sudo pico /etc/sudoers
  - Add:
    - ggs ALL=(ALL:ALL) ALL
- Configure mjpg-streamer for ggs
  - $sudo -s
  - $cp mjpg-streamer /home/ggs/streamer -R
  - $cd /home/ggs ; chown -R ggs:ggs streamer

Setup Ad-Hoc wifi with Mjpg-Streamer(cont.)

- Edit start.sh
  - $cd /home/ggs/streamer ; pico start.sh
  - Modify 1 cmd at top of script:
    - cd /home/ggs/streamer
    - Make sure permission of start.sh for exec
- Create start up script
  - $cd
  - $pico ggsinit.sh
    - Contents as next page

Setup Ad-Hoc wifi with Mjpg-Streamer(cont.)

```
root@omap:~# cat /root/ggsinit.sh
#!/bin/bash


#wait for camera device
while [ ! -e /dev/video0 ];do
        sleep 1
done

#wait for wifi

WLAN=`iwconfig | awk '{print $1}'| grep ^wlan[[:digit:]]`
WLAN=`expr $WLAN:'\(wlan)\'`


while [ -z $WLAN ];do
        sleep 1
        WLAN=`iwconfig | awk '{print $1}'| grep ^wlan[[:digit:]]`

        WLAN=`expr $WLAN:'\(wlan)\'`
done

#give all users access to video devices
chmod o+rw /dev/video*

#enable wifi ad-hoc mode for incoming connections
iwconfig wlan0 essid GGScope
iwconfig wlan0 mode ad-hoc
iwconfig wlan0 channel 7
ifconfig wlan0 192.168.88.166 netmask 255.255.255.0 broadcast 192.168.88.255
iwconfig wlan0 essid GGScope

#start the streamer
su `nohup /home/ubuntu/mjpg-streamer/start.sh > /home/ubuntu/mjpg-streamer/out.l
og 2>&1 &` ubuntu
#enable dhcp server
/etc/init.d/dhcp3-server start
root@omap:~#
```

Setup Ad-Hoc wifi with Mjpg-Streamer(cont.)

- Make sure permission of ggsinit.sh
  - $chmod +x ggsinit.sh
- Add script to /etc/rc.local startup script
  - $pico /etc/rc.local
    - Add as below
      - nohup /root/ggsinit.sh > /dev/null 2>&1 &
      - nohup /home/ggs/streamer/user_led.sh &
- Install dhcpd and modify
  - $apt-get install dhcp3-server
  - $pico /etc/default/dhcp3-server
    - Modify INTERFACE=''wlan0''
  - $pico /etc/dhcp3/dhcpd.conf
    - Modify as below
      - option domain-name ''mobilogics.com.tw'';
      - option domain-name-server 168.95.1.1;
      - default-lease-time 600;
      - Max-lease-time 7200;
      - subnet 192.168.88.0 netmask 255.255.255.0 {
        range 192.168.88.66 192.168.88.88;
        }

Result

- Reboot beagleboard
- Connect ad-hoc with essid:GGScope with your remote device
- Open web-browser,fill in wireless IP of beagleboard,then you can receive mjpg stream by wifi on your mobile-device
  - Port 8080