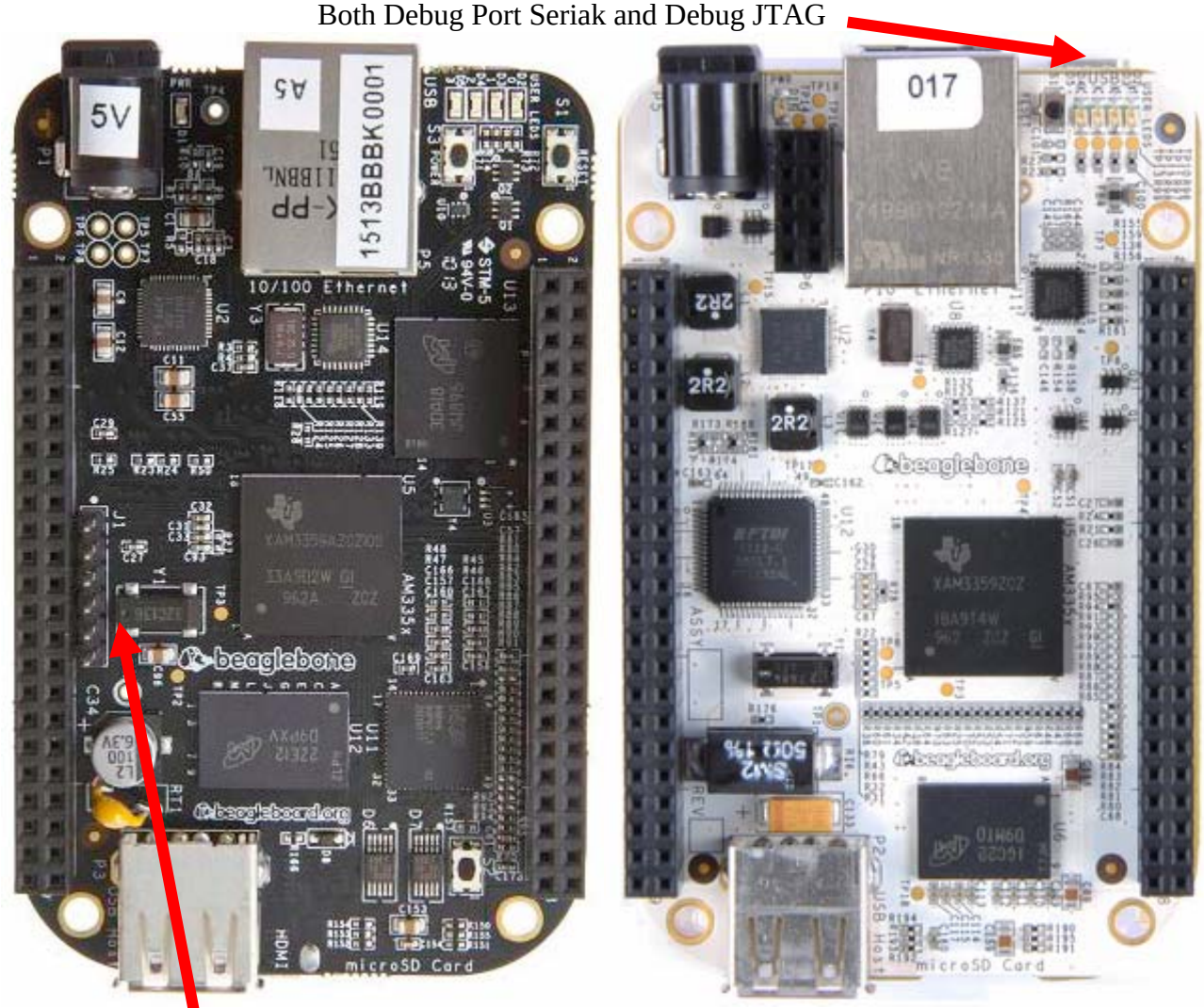
Linux for BeagleBoard – Part 3

Duy-Ky Nguyen **2015-07-20**

Both Debug Port Seriak and Debug JTAG



Debug Port requires USB-Serial adapter

	BeagleBone Black $55	BeagleBone $89
Processor	AM3358BZCZ100, 1GHZ	AM3359ZCZ72, 720MHz
Video Out	HDMI	None
DRAM	512MB DDR3L 800MHZ	256MB DDR2 400MHz
Flash	4GB eMMC, uSD	uSD
Onboard JTAG	Optional	Yes, over USB
Serial	Header	Via USB
PWR Exp Header	**No**	**Yes**
Power	210-460 mA@5V	300-500 mA@5V

1. Introduction

BBB has on-board flash MMC1 as default boot device, and external uSD as MMC0 to boot from IF button BOOT pressed during power-up, and press RESET botton afterward, ie. it remembers the boot device until power-down.

- u-boot-2014.07 in MMC/FAT
- Linux 3.14.1 in MMC/FAT
- BuildRoot-2015.02

Ethernet over USB

BB uses TI Sitara Processor ARM Cortex-A8 AM335x, Original BB White and new BB Black

Boot from SD card with uboot, otherwise from serial port with "C" waiting for the boot image from PC.
SD Card is partitioned into 2 parts (1)Boot_FAT (2)FS_Ext4

Uboot is compiled into 2 files
- (1) stage 1 raw boot name MUST be renamed to "MLO" from "boot_ti.bin"
- (2) staged 2 load boot name must be reanmed to"app" from "app_ti.bin"

	BeagleBone White	**BeagleBone Black**
Power	USB / Ext_DC (Auto-Select if avail)	Ext_DC
Debug Port	USB (Serial & JTAG &Power)	Ext USB Serial required @ J1
Boot	uSD only	On-board flash (default) uSD if BOOT btn pressed during power-up

1.1. Flashing BBB

On-board MMC1 is flashed by operational external uSD

There're 2 ways for flashing

1.1.1. Single Image XZ

Normally download from Beagleboard.org : http://beagleboard.org/latest-images

There're 2 types (1) regular (2) flasher used to flash on-board flash
- Download XZ image
- Unzip it using **7-Zip**
- Flash it using **Win32DiskImager**, it's a GUI version of "**DD**" utility, on Windows
- OR using command line "dd" utility on Linux

NOTE
Its limitation is to create a flash with size just enough for the image, about 4 GB, no matter how big the flash is!

1.1.2. Separate TAR Images :

Normally download from ARMHF : http://www.armhf.com/download/

To flash external uSD

- Untar uboot & root filesystem
- Partition into FAT (type c) and Ext3 (default type) using FDISK on Linux, It's safe on Virtual Machine
- Using Linux Disk Utility to format Ext3 and FAT with **Bootable active**
- Use USB card reader on Linux
    - Copy (1) MLO (2)u-boot.img (3)uImage/zImage (4)device-tree DTB to FAT
    - Copy root files-system to Ext2
- Download & Install 7Zip

1.2. UBoot-2014.07

Download the latest TI BeagleBone Linux
http://software-dl.ti.com/sitara_linux/esd/processor-sdk/PROCESSOR-SDK-LINUX-AM335X/latest/index_FDS.html

Toochain : gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux.tar.bz2

Source uboot-2014.07 & linux-3.14.1 : am335x-evm-sdk-src-01.00.00.03.tar.gz

1.3. TI Linux Kernel

The TI linux-3.14.1 above is already configured for BBB, sp no need to config, but it's broken in using "make xconfig", but OK with "make gconfig"

```
make uImage dtbs LOADADDR=0x80008000
make modules
mkdir RFS
make modules_install INSTALL_MOD_PATH=./RFS
```

NOTE
In the end, it is NOT working for me, it keeps complaining "unable to open /dev/ttyO0" even the inode is there!

1.4. GIT Linux Kernel

For host tools, just in case

```
sudo apt-get ncurses-dev uboot-mkimage build-essential
```

Download linux kernel

```
git clone git://github.com/beagleboard/kernel.git

cd kernel

git checkout 3.14

./patch.sh

cp configs/beaglebone kernel/arch/arm/configs/beagleboneblack_defconfig

wget http://arago-project.org/git/projects/?p=am33x-cm3.git\;a=blob_plain\;f=bin/am335x-pm-firmware.bin\;hb=HEAD -O kernel/firmware/am335x-pm-firmware.bin
```

Compile kernel

```
cd kernel

make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- beagleboneblack_defconfig

make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -j4 uImage dtbs LOADADDR=0x80008000 -j4
```

Compile and Install modules

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -j4 modules

mkdir rootfs

make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- INSTALL_MOD_PATH=./rootfs modules_install
```

1.5. BuildRoot-2015.02

This is the latest version to support Samba-3, the new version Samba-4 is kind of broken!

```
# make xconfig

Toochain
    Type :      External
    Name : Linaro ARM 2014.09
    Origin : To be download

Target packages
    Busybox
        Show packages (to select bash in System config)
    Interpreter languages
        Perl
        PHP * CGI
    Networking
        NTP * NTPDate
        Samba
    Package manager
        Opkg
    System Config
        /bin/sh * bash
```

```
# make busybox-xconfig
Build option
    Static
Linux modules
        modinfo
Simplfied modutils : UNCHECK
    insmod
    rmmod
    lsmod
    modprobe
    depmod
Network
    telnetd
```

httpd
udhcpc
udhcpd

2. Ethernet over USB

- It's NOT possible to use built-in driver. It MUST be loaded as module!
- Make sure under "Driver/USB/USB Gadget/Gadget Drivers" all checked as module

- Use "rcS" to set up network for both regular and Ethernet USB, including Samba server. So remove all S* for those jobs
- On the Linux host, use "modprobe" to load gadget USB module for both Ethernet & Storage USB (modprobe = depmod + insmod). If successfully, "lsusb" lists some thing like "Composite Gadget"
- Use static IP for both regular and USB, but different net_ID : 11.1.1.x for eth0 and **12.1.1.2** for usb0. There's no DHCP server for Ethernet USB dynamic IP

On Linux host, Ethernet USB is config'ed in "/etc/netrwork/interfaces"

```
# /etc/netrwork/interfaces
auto lo
iface lo inet loopback

auto usb0
allow-hotplug usb0
iface usb0 inet static
    address 12.1.1.1
    broadcast 12.1.1.15
    netmask 255.255.255.240
    network 12.1.1.0
    dns-nameservers 11.1.1.1
```

On target, "/etc/init.d/S*networking" is NOT used, the implementation is in "/etc/init.d/rcS" below

```
#!/bin/sh

# /etc/init.d/rcS

# Start all init scripts in /etc/init.d
# executing them in numerical order.
#
for i in /etc/init.d/S??* ;do

     # Ignore dangling symlinks (if any).
     [ ! -f "$i" ] && continue

     case "$i" in
       *.sh)
           # Source shell script for speed.
           (
              trap - INT QUIT TSTP
              set start
              . $i
           )
           ;;
       *)
           # No sh extension, so fork subprocess.
```

```
          $i start
          ;;
    esac
done

#################################################################################

printf "\n\n>>Config g_eth_usb\n"

# Mounted device can be changed, say /dev/mmcblk0p2 for EXT3, instead of FAT bewlow
modprobe g_multi file=/dev/mmcblk0p1 cdrom=0 stall=0 removable=1 nofua=1

# 1st byte nust be EVEN
ifconfig usb0 hw ether 02:33:44:55:66:01
sleep 1
ifconfig usb0 up
sleep 1
ifconfig usb0 12.1.1.2 up

#################################################################################

## printf "\n\n>> rcS IFCONFIG << $(ifconfig) >> \n\n"
if [[ $(ifconfig eth0 up | awk '/not ready/ {print}') ]]; then
       printf "\n\n>>eth0 NOT ready\n"
else
       printf "\n\n>>Config eth0\n"
       ifconfig eth0 up
       sleep 1
       ifconfig eth0 11.1.1.25
       HOSTNAME=bb-25

       printf "\n\n>>Start Telnet Daemon\n"
       telnetd

       printf "\n\n>>Start HTTP Daemon\n"
       httpd -c /etc/httpd.conf -h /home/www

       printf "\n\n>>Start Samba\n"
       smbd -D
       nmbd -D
fi

hostname $HOSTNAME
```

```
# ~/.bashrc: executed by bash(1) for non-login interactive shells.

# /etc/profile

export PATH=\
/bin:\
/sbin:\
/usr/bin:\
/usr/sbin:\
/usr/local/bin

# If running interactively, then:
       alias ll='/bin/ls --color=tty -laFh'
       alias ls='/bin/ls --color=tty -F'
```

```
      export
LS_COLORS='no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;
31;01:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;31:*.z=01;31:*
.Z=01;31:*.gz=01;31:*.bz2=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.jpg=01;35:*.jpeg=01;35:*.png
=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=0
1;35:*.tif=01;35:*.tiff=01;35:*.mpg=01;35:*.mpeg=01;35:*.avi=01;35:*.fli=01;35:*.gl=01;35:*.dl=01;
35:*.xcf=01;35:*.xwd=01;35:';

    export USER=`id -un`
    export LOGNAME=$USER
    export HOSTNAME=`/bin/hostname`
    export HISTSIZE=1000
    export HISTFILESIZE=1000
    export PAGER='/bin/more '
    export EDITOR='/bin/vi'
    export INPUTRC=/etc/inputrc
    export DMALLOC_OPTIONS=debug=0x34f47d83,inter=100,log=logfile

# Source configuration files from /etc/profile.d
for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

################################################################################

HOME=/home

cd /home/

export PS1="${HOSTNAME}:\w % "

################################################################################
################################################################################

alias ssbb='smbd -D; nmbd -D'
alias lll='ls -als'
alias ..='cd ..'
alias ...='cd../..'
```

On the host, “ping 12.1.1.2”
On the target, “ping 12.1.1.1”

3. Using USB_Ether with U-Boot

```
set ethact usb_ether
set ipaddr 12.1.1.2
set serverip 12.1.1.1
```

4. Conclusion

Both Ethernet and Mass Storage are working through USB for both BeagleBone Black & White, same kernel, but different DTB

On Windows, we can see 3 devices (1) Ethernet (2) Serial Port (3) Storage