PhyCORE-AM335X Jelly Bean Android Development

1 About this Quickstart

This document describes how to install and work with the Jelly Bean Android Board Support Package (BSP) for the phyCORE-AM335X platform. This BSP provides a fundamental software platform for development, deployment and execution on the phyCORE-AM335X.

The Quickstart contains instructions for:

```
Host Setup
Building a BSP (Platform, Kernel, Android Filesystem)
Installing the images on the target (Barebox, Kernel, Android Filesystem)
```

Note:

- For performance reasons, the images should be installed into the target's NAND as described below.
- Be sure to use the sudo commands indicated in the following steps rather than running them as a regular, non-root user on the development machine. As a non-root user, some target files will not have root permissions and the target will not operate properly.

2 Host Setup

The following notes assume a Ubuntu 10.04 64 bit build machine.

The host must be a 64 bit machine

2.1 TFTP

TFTP allows files to be downloaded from one machine to another. With most embedded Linux devices, TFTP is an efficient way to boot the kernel during development so that the user does not have to flash a new kernel every time it is modified. It is also helpful when updating images in flash from Barebox. First, start by installing the TFTP server.

```
sudo apt-get install tftpd-hpa
```

Next, files can be accessed from another machine on the same network by simply using the IP address of the host. You must specify a folder where the files will reside on the host by replacing the folder path for TFTP_DIRECTORY with whatever folder you wish to use as your TFTP file storage location, or leave the folder as the default.

sudo gedit /etc/default/tftpd-hpa

```
# /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

If you made any changes to the settings of the TFTP server, you need to restart it for them to take effect.

```
sudo restart tftpd-hpa
```

Lastly, if you would like to grant every user on the system permission to place files in the TFTP directory, use the following command, replacing <TFTP_DIRECTORY> with your chosen location.

```
sudo chmod ugo+rwx <TFTP_DIRECTORY>
```

2.2 Packages

Android development requires certain packages to be installed. Run the following commands to ensure that you have the packages installed:

```
sudo apt-get install git-core flex bison gperf libesd0-dev zip gawk ant libwxgtk2.6-dev
sudo apt-get install zlib1g-dev build-essential tofrodos x-dev libx11-dev libncurses5-dev
sudo apt-get install lib32readline5-dev libstdc++6 lib32z1 lib32z1-dev ia32-libs g++-multilib
sudo apt-get install libx11-dev libncurses5-dev uboot-mkimage libxml2-utils xsltproc
```

If you wish to store images in the target's NAND, also:

```
sudo apt-get install mtd-utils
```

Install Java SE 6 JDK from Oracle:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Add the bin directory of the Oracle JDK package to your PATH.

Android does not build using openjdk. If you have openjdk installed on your development machine, you will need to ensure that the PATH variable includes the path to the Oracle JDK bin directory first. That is, "*type java*" should indicate the Oracle binary.

Also install the repo tool:

```
mkdir ~/bin -p
sudo apt-get install curl
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
export PATH=~/bin:$PATH
```

2.3 Board Setup - phyCORE-AM335X

After the device is out of the box and setup, make sure that there is a DB9/DB25 cable attached between the host machine and UART0 on the AM335X carrier board to enable RS-232 serial communication. In a terminal on the host, access minicom and set it up, allowing console access over the serial port.

```
/* If minicom is not installed */
sudo apt-get install minicom

minicom -c on -s
```

Navigate to "Serial port setup" in minicom and modify line "A - Serial Device : " to read /dev/ttyS0.

Note: The serial device is dependent on what COM port you are connected to on your system, so /dev/ttyS0 is merely an example.

Next, modify "E - Bps/Par/Bits : " to have a speed of 115200 and 8-N-1 (8N1) for the stop bits. Return to the main menu of minicom and select "Save setup as dfl" to make this the default setup anytime minicom is loaded, meaning minicom -c on is all that needs to be done in the future for this machine to be able to communicate with the kit.

3 Building a BSP

3.1 Downloading Source

Important:
Please contact PHYTEC at support@phytec.com to obtain a pre-packaged source code archive. This BSP was based on source code hosted on Gitorious.org, which has been shutdown as of May 2015.

Use the following instructions to obtain the Android source code.

- Create a directory which will house your BSP development:

```
mkdir mydroid
cd mydroid
export MYDROID=`pwd`
```

- Extract PHYTEC provided source code archive:

```
tar -xvzf phyCORE-AM335X-JellyBean-PD14.1.0_extracted_sources.tgz -C $MYDROID
```

3.2 Build Environment Setup

The toolchain used to build all components is included in Android. To add the toolchain to your search path:

```
cd $MYDROID
export PATH=`pwd`/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin:$PATH
```

```
sudo cp build/core/root.mk Makefile
```

3.3 Building The Bootloaders, The Android Filesystem And The Kernel

To build the Android Filesystem and kernel:

```
cd $MYDROID
make TARGET_PRODUCT=pcm051 rowboat_clean
make TARGET_PRODUCT=pcm051 OMAPES=4.x -j<N>
```

where N is the number of CPUs in your development system.

3.4 Creating The Root Filesystem

To assemble the components into a root filesystem

```
cd $MYDROID
make TARGET_PRODUCT=pcm051 fs_tarball
```

The root filesystem will be contained in *out/target/product/pcm051/rootfs.tar.bz2*

If you wish to boot from the target's NAND, run:

```
./mkrootfs
```

3.5 Preparing The SD Card

An SD Card is used either as a boot device or to install files to the target for NAND boot.

Configure the SD Card with the necessary partitions using the following script, mkcard. This need only be done once for each SD card.

```
#!/bin/bash
# Usage: sudo mkcard /dev/sdN   where sdN is the device associated with your SD card
if [ ! "$1" = "/dev/sda" ] ; then
        unset LANG
        DRIVE=$1
        if [ -b "$DRIVE" ] ; then
                dd if=/dev/zero of=$DRIVE bs=1024 count=1024
                SIZE=`fdisk -l $DRIVE | grep Disk | awk '{print $5}'`
                echo DISK SIZE - $SIZE bytes
                CYLINDERS=`echo $SIZE/255/63/512 | bc`
                echo CYLINDERS - $CYLINDERS
                {
                echo ,9,0x0C,*
                echo ,,,-
                } | sfdisk -D -H 255 -S 63 -C $CYLINDERS $DRIVE
                mkfs.vfat -F 32 -n "boot" ${DRIVE}1
                mke2fs -j -L "rootfs" ${DRIVE}2
        fi
fi
```

Note the Usage line in the script.

If you are using an SD card which already had partitions, Ubuntu will automatically mount them when the card is inserted. You will need to unmount them before using the script (*su umount /dev/sdN1*, *su umount /dev/sdN2*, ...).

After using the script, the partitions will be mounted as /media/boot and /media/rootfs when the SD card is inserted into a Ubuntu system

Copy Files to the SD card:

```
cp $MYDROID/barebox_mlo-2013.07.0/MLO /media/boot
cp $MYDROID/barebox-2013.07.0/barebox.bin /media/boot/
```

If you intend to run Android from NAND, the above two files are all that you need on the SD card. If you intend to run Android from the SD card, additional files are necessary:

```
cp $MYDROID/kernel/arch/arm/boot/uImage /media/boot/uImage
sudo rm -r /media/rootfs/*
sudo tar xjf out/target/product/pcm051/rootfs.tar.bz2 -C /media/rootfs
```

3.6 Installing Images Into NAND

The following files should be placed in the directory used by the tftp server (eg /var/lib/tftpboot):

- u-boot-2011.09/MLO

- barebox-2012.11.0/barebox.uimage
- barebox-2012.11.0/common/barebox_default_env
- kernel/arch/arm/boot/uImage (renamed uImage-pcm051)
- rootfs.ubi

Configure boot-mode for SD: S5 switches 1-4 and 6 ON, 5, 7 and 8 OFF. The SD card should be inserted into the SD/MMC slot. After the board is powered on, stop the autoboot. Install the images as outlined in the following table:

Description	Command
Update MLO into NAND via TFTP	`update -t xload -d nand -m tftp -f MLO`
Update barebox into NAND via TFTP	`update -t barebox -d nand –m tftp -f barebox.uimage`
Update bareboxenv into NAND via TFTP	`update -t bareboxenv -d nand –m tftp -f barebox_default_env`
Update kernel into NAND via TFTP	`update -t kernel -d nand –m tftp -f uImage-pcm051`
Update rootfs into NAND via TFTP	`update -t rootfs -d nand –m tftp -f rootfs.ubi`

NOTE : *TFTP is the default mode, as set in /env/config in barebox, therefore, "-m tftp" can be omitted from update commands.*

Power off the board and configure boot-mode for NAND: all S5 switches off

3.7 Keeping Your Kernel On The TFTP Server

For kernel development, it is handy to have the target load the kernel from the tftp server rather than replacing the kernel in NAND or on the SD card. To facilitate this, stop the board from autobooting and:

edit env/config

Change the line:

```
kernel_loc=nand
```

to

```
kernel_loc=tftp
```

Ensure that eth0.ipaddr and eth0.serverip and the other eth0 settings are appropriate. Save the file with <ctrl>d, then:

saveenv

The kernel image file, uImage-pcm051, will then be loaded from /var/lib/tftpboot on your server. The Android filesystem will continue to be loaded from NAND or the SD card.

Note that the power must be cycled on the target for the new environment to become active.

4 Usage

4.1 Jumpers

Ensure that the following jumpers are installed on your carrier board:

JP11 to the right of the purple battery (required for USB Host operation)

JP24 near the LCD connector. Insert a jumper over the top two pins to enable display output.

4.2 Ethernet

The Ethernet interface is not enabled by default. To activate it, start the Settings application and change Ethernet from OFF to ON. Select Ethernet configuration (touch the Ethernet line on the screen) followed by eth0. Select either DHCP or a Static IP address and Apply.

4.3 First Boot

The first boot following a clean rebuild of the target will be slow, taking several minutes to complete. This is due to Android's building of caches in /data.

4.4 adb

Although you have a console with the phyCORE-AM335X, this is quite often not the case for Android devices. Debugging is therefore normally performed using adb, the Android debug bridge.

First, ensure that jumpers JP10 and JP12, located behind the OTG port on the carrier board, are not installed. Next, connect a USB cable between the OTG port on the carrier board and your build machine.

adb can be found in out/host/linux-x86/bin of your Android build tree.

Common adb use includes *adb shell* to login to the device and *adb logcat* to see logging performed by Android services and applications.

Note that the shell user at the console is not root and therefore does not have permission to do many things. adb root/adb shell resolves this problem.

4.5 Buttons

Two switches on the carrier board are defined as navigation buttons:

BTN1: Back

BTN2: Home

5 Media

The phyCORE-AM335X contains a number of devices which can be used for data storage. The Android application vold uses the file system/etc/vold.fstab to control which devices are mounted (when inserted) and where they are mounted. The source for the target vold.fstab file is devices/phytec/pcm051/vold.fstab.

For example, the entry:

```
dev_mount sdcard /storage/sdcard0 3 /devices/platform/omap_hsmmc.0/mmc_host/mmc0
```

mounts the 3rd partition on the first SD card (SDMMC1 or X11) as /storage/sdcard0 when it is inserted.

The entry:

```
dev_mount usb /storage/usb1 auto /devices/platform/omap/musb-ti81xx/musb-hdrc.1/usb1
```

mounts a partition on a USB drive when installed in the USB Host port

For the above mount commands, the mount point must be created in the "on init" section of the init.pcm051.rc boot script:

```
mkdir /storage/sdcard0 0000 system system
mkdir /storage/usb1 0666 system system
```

Depending on the intended use of the media, the mount points may be adjusted accordingly.

A note of caution: By default, the init.pcm051.rc uses /data/media for the storage of media files such as music. That is, these files are stored on the root filesystem and no SD card is required. To accomplish this, the following lines are included in the init.pcm051.rc script:

```
service sdcard /system/bin/sdcard /data/media 1023 1023
    class late_start
```

These lines mount /dev/media as /storage/sdcard0. If instead you wish to use a real SD card for media storage, comment out the above two lines.

6 CANbus

The ip command can be used on the console to configure the can interface. For example:

```
ip link set can0 up type can bitrate 1000000
```

CANbus utilities, such as

```
git clone git://gitorious.org/linux-can/can-utils.git external/canutils
```

In order to add theses utilities to the root filesystem, edit the file device/phytec/pcm051/device.mk and look for the lines

```
# CANbus tools
# uncomment as appropriate
#PRODUCT_PACKAGES += \
#       candump \
```

Uncomment (remove the leading #) from the PRODUCT_PACKAGES as well as any utilities you wish to add to the target. Rebuild Android as indicated above.