Embedded Bluetooth server

Sachin Kharude
Department of Electrical Engineering
V.J.T.I. Mumbai-19, India

R.D.Daruwala
Department of Electrical Engineering
V.J.T.I Mumbai-19, India

Prathamesh Berde
Department of Electrical Engineering
V.JT.I. Mumbai-19, India

ABSTRACT
This paper describes the method and progress of developing BeagleBoard-xM as wireless server. In this paper we are going to use BeagleBoard-xM, which is a new low-cost low-power platform based on the Texas Instruments (TI) DM 3730 processor architecture. The DM3730 processor has dual core architecture consisting of ARM and DSP. The whole process involves compilation and configuration of Android Linux kernel, boot loaders (U-boot and X-loader), construction of root file system, development of cross compiler environment and its implementation on to the target board. To make the BeagleBoard-xM as wireless server it should be enabled for Bluetooth USB dongle and development of application program that makes it wireless server for other clients. This server is small pocket sized portable embedded server when connected in network, can be used to serve the clients in various applications.

General Terms
Kernel compilation using Android Toolchain, Android porting on BeagleBoard-XM.

Keywords
Android, Server, BeagleBoard-xM, kernel, Bluetooth.

1. INTRODUCTION
In a network, the client/server model provides a convenient way to interconnect. Normally the server in network is implemented using one or many computers. These servers are huge and require a lot of space and power. To perform some dedicated and fixed number of tasks in an industry or at home we need servers which are small and portable. When it comes to portability and performing dedicated tasks embedded systems are always a better option.

This paper proposes an idea of pocket size embedded wireless server which will provide service to the multiple clients simultaneously[1].For this work our embedded hardware platform should be the BeagleBoard-xM powered by a DM 3730 mobile processor from TI which runs at 1 GHz and has 512 MB of LPDDR RAM POP memory. It has asymmetric dual-core architecture with an ARM Cortex-A8 and TMS320DMC64X+ Digital Signal Processor (DSP).

For seamless integration of these functional modules an operating system has to be ported on the platform and we need to port Android operating system because of its various features. Android is open-source, viable and robust and supports multi-tasking and multi-process. Android is a Linux-based operating system for mobile devices such as smart phones and tablet computers. It is developed by the Open Handset Alliance led by Google. Android is designed in accordance with the requirement of an embedded mobile operating system like reduced size generally only a few hundred KB, reduced storage space, task specific, specialized and having real time behavior. Android supports TI's BeagleBoard-xM. Any device serving out applications can be called as server. A Bluetooth server is wireless server which serves any Bluetooth enabled device in its vicinity.

2. SYSTEM OVERVIEW

2.1 Embedded Platform
The Beagle Board-XM [2] is a low-power, low-cost Single-board computer produced by Texas Instruments in association with Digi-Key. The Beagle Board was designed with open source development in mind, and as a way of demonstrating the Texas Instruments DM3730 system-on-a-chip. The board was developed by a small team of TI engineers. The Beagle Board is designed specifically to address the Open Source Community. It has been equipped with a minimum set of features to allow the user to experience the power of the processor and is not intended as a full development platform as many of the features and interfaces supplied by the processor are not accessible from the Beagle Board. By utilizing standard interfaces, the Beagle Board is highly extensible to add many features and interfaces. It is not intended for use in end products. All of the design information is freely available and can be used as the basis for a product.

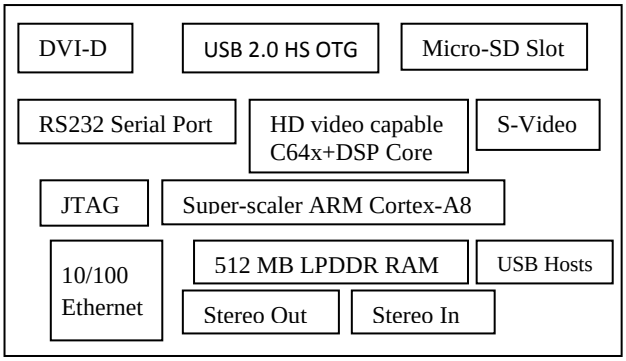

Fig.1. BeagleBoard-xM Block Diagram

2.2 Operating System
Android is an operating system for mobile devices such as mobile phones, tablet computers and net books. Android is developed by Google and is based upon the Linux kernel and GNU software. Android, which was later positioned in the Open Handset Alliance in July 2005, was initially developed by Android Inc. The Android operating system software stack consists of four parts viz. Applications, Application Framework, Libraries and Linux kernel. Applications: basic applications include an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written in Java programming language. Application Framework the developers have full access to the same framework APIs used by applications base. The architecture is designed to simplify the reuse of components, any application can publish its capabilities and any other application can then make use of those capabilities (subject to safety rules

framework). This same mechanism allows components to be replaced by the user.

3. STAGES OF IMPLEMENTATION

In this project we used a kernel source and ARM tool chain by building an open source Android Linux kernel and installing open source tool-chain. We need to use micro SD card as boot device. The serial port interface is used as a console for debugging and testing.

3.1 Host Development Environment

The host development environment for Android is based on Ubuntu, install Ubuntu version 10.04[5] or later. The host installation would need few more Android specific dependencies; these can be installed dynamically over network using below commands.

For Ubuntu on 32 bit machines:

$sudo add-apt-repository "deb http://archive.canonical.com/ Lucid partner"

$ sudo add-apt-repository" deb-src http://archive.canonical .com/Ubuntu lucid partner"

$sudo apt-get update

$sudo apt-get install git-core gnupg sun-java6-jdk flex bison gperf libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl libncurses5-dev zlib1g-dev minicom tftpd uboot-mkimage expect

$sudo update-java-alternatives-s java-6-sun

3.2 Android Source Code

TI provides Android sources [4] for all the supported devices in multiple locations, developers can download the sources from the gitorious.org/rowboat repository or use the pre-packaged repo in the DevKit.

Using pre-packaged repo

$mkdir rowboat-android

$cd rowboat-android

$tar -xvzf TI_Android_GingerBread_2_3_4Sources.tar.gz

$cd TI_Android_GingerBread_2_3_4Sources

$./.repo/repo/repo sync --local-only

This will generate the sources for Android File system, Android Linux Kernel (in kernel directory), u-boot boot loader (in u-boot directory), x-loader (in x-loader directory), Tool chain (prebuilt/linux-x86/tool chain/arm-eabi-4.4.3/bin)

3.3 Compilation Procedure

3.3.1 Toolchain Setup [7]

Setup the tool-chain path to point to arm-eabi-tools in prebuilt/linux-x86/Toolchain/arn-eabi-4.4.3/bin

$exportPATH=<PATH_TO_ROWBOAT>/rowboat-android/prebuilt/linux-6/toolchain/arm-eabi-4.4.3/bin:$PATH

3.3.2 To build x-loader

$cd x-loader

$ make CROSS_COMPILE=arm-eabi- distclean

$ make CROSS_COMPILE=arm-eabi- omap3evm_config

$ make CROSS_COMPILE=arm-eabi-

$. /signGP ./x-load.bin

$mv x-load.bin.ift MLO

3.3.3 To build boot loader (U-boot) [3]

$cd u-boot

$ make CROSS_COMPILE=arm-eabi- distclean

$ make CROSS_COMPILE=arm-eabi- omap3_evm_config

$ make CROSS_COMPILE=arm-eabi-

3.3.4 To build Linux kernel [6]

$cd kernel

$ make ARCH=arm CROSS_COMPILE=arm-eabi- distclean

$ make ARCH=arm CROSS_COMPILE=arm-eabi-omap3 -evm_android_defconfig

$ make ARCH=arm CROSS_COMPILE=arm-eabi- uImage

3.3.5 To build Android file system

$cd <path to android source>

$ make TARGET_PRODUCT=<Beagleboard>OMAPES =<5.x> -j<N>, where <N> should be the twice the no of processors on your host machine. For dual core machine use –j4

3.3.6 Create root file system tar ball

$cd out/target/product/omap3evm

$mkdir android_rootfs

$cp -r root/* android_rootfs

$cp -r system android_rootfs

$sudo../../../../build/tools/mktarball.sh../../../host/linux-x86/bin/fs_get_stats android_rootfs. rootfs.tar.bz2

3.3.7 To generate SD/MMC card to boot Android

$. /mkbootscr

Copy all images to one folder

$cp Tools/mk-mmc/mkmmc-android.sh image_folder

$. /mkmmc-android /dev/[device] MLO u-boot.bin uImage boot.scr rootfs.tar.bz2 Media_Clips

3.3.8 Android VNC Server

To access the board we have compiles Android VNC server application, by using this application we can remotely log into the BeagleBoard-Xm. Following is the procedure for compiling Android VNC Server source code.

$cd <android source path>/external

$git lone git://gitorious.org/rowboat/droid-vnc-server.git

$cd droid-vnc-server/jni

$source <android source path>/build/envsetup.sh

$mm TARGET_PRODUCT=BeagleBoard

Fig.2.VNC Viewer on Ubuntu machine

The fig.2 shows that android VNC server is running on BeagleBoard-Xm. BeagleBoard-Xm is accessed on Ubuntu machine using GtkVNCviewer.

3.3.9 Configure Bluetooth

We need to use TI's WL1271 daughter card to provide bluetooth connectivity to the Beagleboard-XM. TI'sWL1271 is combination of Wi-fi and bluetooth. The WL1271 device is the heart of the WiLink™ 6.0 solution. It is a single-chip combo comprising multiple radio technologies. It encompasses all digital and analog content required for WLAN and Bluetooth® communications. Download WL1271_Linux_SDK 3_00_01_06 package from

http://software.ti.com/dsps/dsps_public_sw/sdo_sb/ecs/WL1271_Linux_SDK/3_00_01_06/index_FDS.html. Install the SDK. $chmod a+x v3.00.01.06-wl6.1.6.0-Linux-x86-Install
$ ./v3.00.01.06-wl6.1.6.0-Linux-x86-install –mode console
Download the Android patch package WL1271Android-patches.tar.gzfrom http://softwaredl.ti.com/dsps/dsps_public_sw/sdo_tii/TI_Android_DevKit/02_02_00/index_FDS.html and install. Then copy Bluetooth firmware. Thus we can add Bluetooth support for Android which is running on the BeagleBoard-Xm.

4. APPLICATION PROGRAMMING

This Bluetooth stack enables to locate Bluetooth devices and establish connection. After configuring the Bluetooth on to the platform, applications are programmed in JAVA language to scan any available Bluetooth device in its vicinity and to be as server. In application program we used APIs to interact with Bluetooth protocol stack layers. Now the Beagleboard-XM is ready to work as independent Bluetooth module. With this approach we can develop embedded server which is magnificent illustration of wireless technology. If it is imbibed into portable kit, it can become very handy tool for data transferring.

5. CONCLUSION

Android is merely a tool, we are the sculptor. This paper describes the process of porting Android on BeagleBoard-Xm which is powered by a DM 3730 mobile processor from TI. Once we port Android on to the BeagleBoard-Xm it's become easy to add Bluetooth support to the board. Bluetooth server is a magnificent illustration of wireless technology that when imbibed into a portable kit became a very handy tool for data transfer. In future we can add various applications by replacing Bluetooth by other wireless technologies such as Wi-Fi.

6. ACKNOWLEDGMENTS

We want to thank Mr. Mahesh U. Patil, CDAC, Hyderabad, India, for his invaluable guidance and support.

7. REFERENCES

[1] Seema, Ashish Phogat. Transplanting Bluetooth Server Using Embedded Linux on OMAP Platform

[2]BeagleBoard-xM Rev c system reference manual

[3]Sun Zhanchen, Zhao Hongwei1, GU Jianhao the U-boot Transplantation Based on S3C2440

[4]Android sourcecode:http://gitorious.org/rowboat/repository Repository

[5] Ubuntu updates http://archive.canonical.com/lucid partner

[3]Sun Zhanchen, Zhao Hongwei1, GU Jianhao the U-boot Transplantation Based on S3C2440

[6] Linux kernel compilation support http://e2e.ti.com/android

[7] Abhyudai Shankar, Somya Lal. Android porting concepts