
Juphoon Protocol Framework

Juphoon Phone (JPhone)

Published: Dec 2008

Please visit our website at <http://www.juphoon.com>

Setup for JPhone Develop Environment

Juphoon System Software Co., Ltd.
<http://www.juphoon.com>
Tel: +86-574-87287820
Fax: +86-574-87304379
Document Number: 100-000-03-02

Copyright © 2008, Juphoon System Software Corporation.
All rights reserved.

Revisions History

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Bob Liu	Translate this document from Chinese to English.	2008-12-3

Contents

SETUP FOR JPHONE DEVELOP ENVIRONMENT	1
1. INTRODUCTION	4
2. PREPARATION.....	5
2.1 CONSOLE SOFTWARE	5
2.2 SETUP THE X	6
2.3 SETUP THE HOST PC	6
2.3.1 OS.....	6
2.3.2 TFTP Server.....	7
2.3.3 Cross-Compiler.....	7
2.3.4 Environment Variable	7
2.4 UCLINUX PACKAGE.....	8
3. UPDATE APPLICATION PROGRAM.....	8
3.1 COMPILE	8
3.2 UPDATE.....	8
3.3 DEBUG	8
4. UPDATE KERNEL AND FILE SYSTEM	10
4.1 START X SERVER	10
4.2 CONFIGURE KERNEL AND FILE SYSTEM	10
4.3 CUSTOMIZE THE KERNEL	10
4.4 CUSTOMIZE THE FILE SYSTEM	10
4.5 BUILD.....	10
4.6 UPDATE.....	10
4.6.1 U-boot Mode	10
4.6.2 Set Parameters	11
4.6.3 Update Kernel.....	11
4.6.4 Update File System	11

1. Introduction

Hardware for JPhone development environment includes:

- A hardware phone or development board, we use the hardware phone for example;
- A host PC installed with Linux for cross-compile. We used the Redhat 9;
- A PC with RS232 port used as console of hardware phone or development board.

Note: The host PC can be installed on the console PC through virtual software.

The software includes:

- The console software, such as Hyper Terminal or Putty;
- The cross-compile tools, we used blackfin-toolchain-07r1-9.i386.rpm;
- uClinux package, we used uClinux-dist.07R1.1-RC3.tar.bz2;
- TFTP server, X etc.

It includes following steps for developing and debugging software:

- Cross-compile the program on the host PC;
- Download the program to the hardware phone;
- Use console software to run or debug the program through RS232 cable.

And for update kernel or file system:

- Configure the kernel and file system by graphic tools under X;
- Cross-compile the kernel or file system on the host PC;
- Reboot the hardware phone to the u-boot mode;
- Use console software to update the kernel or file system on the phone.

2. Preparation

2.1 Console Software

Console software was installed on the console PC which connected to the hardware phone by a RS232 cable. It used to update or debug the hardware phone. Common console software includes Hyper Terminal, Putty, minicom etc. Following description is based on Hyper Terminal.

Note: The Telnet service was enabled on the hardware phone by default. So we can also update or debug by Telnet. But when you want to update the kernel or file system, the console software must used.

Setup the Hyper Terminal:

1. Connect the hardware phone and console PC with RS232 cable;
2. Open the Hyper Terminal in Start->Programs->Accessories->Communications;
3. New connection, select the proper COM. And set the properties as following:

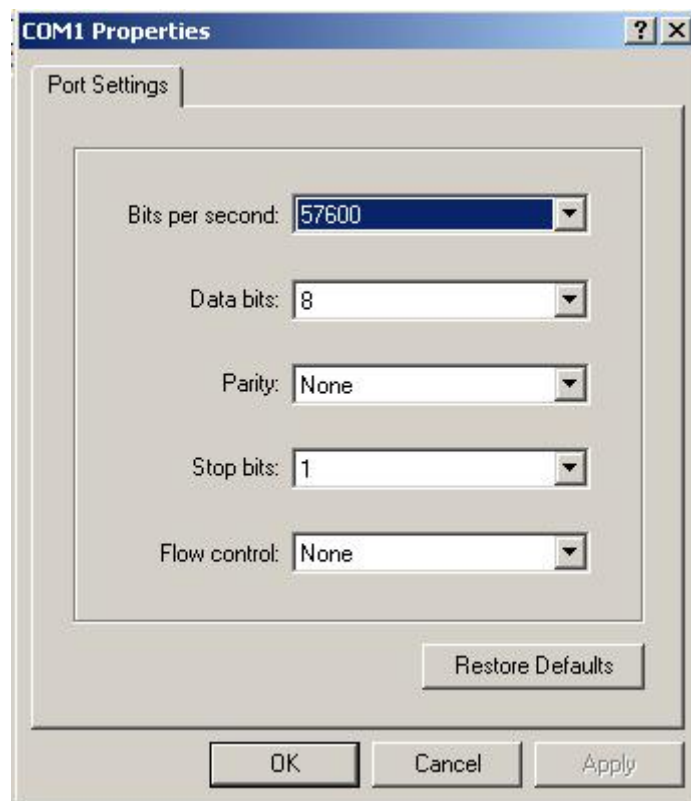


Figure 1: The properties of Hyper Terminal connection

4. Turn on the power of hardware phone. It will display following text if it works OK.

```
CPU: ADSP BF531 Rev.: 0.5
Board: Juphoon BF531 IP Phone board
      Support: http://www.juphoon.com/
Clock: VCO: 400 MHz, Core: 400 MHz, System: 100 MHz
SDRAM: 16 MB
FLASH: 4 MB
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 1
```

2.2 Setup the X

To configure the kernel and file system more easily, we use the graphic tools under X.

Note: If you only want to modify or debug the application program in hardware phone, you can just skip this section.

There 2 way to setup the X:

- Install X package in the host PC;
- Use the 3rd party X server, like Cygwin etc.

Following is based on X server in Cygwin which resource has downloaded.

1. Download the setup program from the website of Cygwin;
2. Run the setup program, and select “Install from Local Directory”;
3. Select install directory, like C:\Cygwin;
4. Press “Browse” on the right of “Local Package Directory”, and select the directory which contain the resource;
5. In package selection page, press “Default”, and change it state to “Install”;
6. Wait until setup is completed.

2.3 Setup the host PC

The host PC has 2 functions:

- The platform for cross-compile with which the hardware phone’s kernel, file system and application was generated;
- The TFTP server for downloading the kernel, file system and application to the hardware phone.

2.3.1 OS

We use the Redhat 9 as the OS of host PC. Because JPhone development environment depends on many software, so we suggest to select the complete mode when installing.

2.3.2 TFTP Server

The TFTP service was disabled in host PC by default. Modify `/etc/xinetd.d/tftp` as following to enable TFTP service:

```
service TFTP
{
socket_type = dgram
protocol = udp
wait = yes
user = root
server = /usr/sbin/in.tftpd
server_args = -s /tftpboot
disable = no
per_source = 11
cps = 100 2
flags = IPv4
}
```

Create the root of TFTP:

```
mkdir /tftpboot
```

Change the read/write properties:

```
chmod o+w /tftpboot
```

Restart the network services:

```
service xinetd restart
```

2.3.3 Cross-Compiler

The software can download from <http://blackfin.uClinux.org/gf/project/toolchain>. Then install the package:

```
rpm -ivh blackfin-toolchain-07r1-9.i386.rpm
```

It will install to the `/opt/uClinux/bfin-uClinux` by default.

2.3.4 Environment Variable

Add following to `~/.bash_profile`:

```
export CROSS_COMPILE=bfin- uClinux
export PATH=$PATH:/opt/uClinux/bfin- uClinux /bin
export LC_ALL=POSIX
export DISPLAY=192.168.0.23:0.0
```

192.168.0.23 is the host IP address the X server runs on. Then run

```
. ~/.bash_profile
```

2.4 uClinux Package

Un-package the uClinux package:

```
tar -xjvf uClinux-dist.tar.bz2
```

Then clean the intermediary files:

```
make clean
```

3. Update Application Program

3.1 Compile

On the host PC:

Change directory to sip_phone/project/uclinux_bfin_ft. To generate the debug version of the program:

```
make
```

To generate the release version of the program:

```
make target=release
```

The program file is bin/jpda by default. Then copy it to the root of TFTP, /tftpboot.

3.2 Update

In the console software:

First login the hardware phone. Then change directory to /home. To download the application program:

```
tftp -r jpda -l jpda -g 192.168.0.228
```

192.168.0.228 is the IP address of host PC.

By default, there already is a jpda file in the hardware phone, the preceding operation will overwrite the old one. If there isn't a jpda file before update, you should add executable properties to the jpda file:

```
chmod +x jpda
```

3.3 Debug

The jpda will start automatically after reboot the hardware phone by default. If you want to debug the new program, you should first modify /etc/rc to prevent the jpda from auto-start.

Note: The program in hardware phone should be the debug version. And keep the source file and program un-modified after compiled.

Do as following to debug the program:

1. Login the hardware phone. Change directory to /home. Start the GDB server:

```
gdbserver 192.168.0.69:1111 jpda
```

192.168.0.69 is the IP address of hardware phone. 1111 is the GDB port.

2. In the host PC, change directory to sip_phone/project/uclinux-bfin-flt/bin. Start the GDB client:

```
bfin-uClinux-gdb jpda.gdb
```

3. Then connect to the GDB server:

```
target remote 192.168.0.69:1111
```

4. Use “b” to add breakpoint, like:

```
b jpda.c:73
```

Add a breakpoint at 73 line in jpda.c;

5. Use “c” to start the program. Refer GDB help file for more.

4. Update Kernel and File System

4.1 Start X Server

We use the graphic tools to configure the kernel and file system. So we must start X server first. The following is based on the X server in Cygwin.

Start X server by open Start->Programs->Singular CAS->Start X - Server. Then start xterm by open Start->Programs->Cygwin-X->xterm, enter as following in xterm window to enable the capability of accepting the connection from other host:

```
$ xhost +
```

4.2 Configure Kernel and File System

In host PC, change directory to uClinux root. Start the graphic tools:

```
make xconfig
```

Then in X server host, it will display a window with “Analog Devices Blackfin Embedded Linux Configuration”

In Vendor/Product Selection, select AnalogDevices for Vendor, and BF533-STAMP for CPU.

In Kernel/Library/Defaults Selection, select uC-libc for Libc Version.

Set Customize Kernel Setting and Customize Vendor/User Setting to Yes, the rest remains No.

Back to Main Menu, press Save and Exit.

4.3 Customize the Kernel

Please refer the uClinux documents.

4.4 Customize the File System

Please refer the uClinux documents.

4.5 Build

In host PC, change directory to uClinux root. Build the kernel and file system:

```
make
```

If succeed, the kernel file is images/vmImage, and the file system file is images/rootfs.jffs2. Then copy them to the TFTP root directory /tftpboot.

4.6 Update

The hardware phone must in u-boot mode before update kernel or file system.

4.6.1 U-boot Mode

Connect to the hardware phone with console software. Press enter to interrupt the boot progress.

Now the hardware phone is in u-boot mode.

4.6.2 Set Parameters

Display current parameters:

```
printenv
```

Please set serverip,gatewayip,ipaddr to the proper value:

- serverip is the IP address of the host PC;
- gatewayip is the IP address of gateway;
- ipaddr is the IP address of the hardware phone.

For example, set the gateway IP to 192.168.0.2

```
set gatewayip 192.168.0.2
```

To save the parameters, run

```
saveenv
```

4.6.3 Update Kernel

Update the kernel to the hardware phone, enter as following in u-boot mode:

```
run updatekernel
```

Note: Enter boot to boot up the hardware phone after updated.

4.6.4 Update File System

Update the file system to the hardware phone, enter as following in u-boot mode:

```
run updatrootfs
```

Note: Enter boot to boot up the hardware phone after updated.