

Chapter 15

ARM –

Architecture, Programming

and

Development Tools

Lesson 08

Porting Developed Codes and Linux in ARM Based System

Linux Operating System

- RealView Tools and Platform support main Linux kernel
- Linux kernel images, patches and utilities required to run Linux on ARM processors and RealView platforms
- Linux is the version of Linux running on processors with Memory Management Unit (MMU)

Download of the Linux for ARM platform.

- http://www.arm.com/products/os/linux_download.html is used
- Boot loader U-Boot taken from <http://www.denx.de/wiki/U-Boot>
- U-Boot images are ".axf" files
- Linux images are "uImages"
- The configuration files are plain text format files

uClinux

- A modified version of Linux for processors without MMU

Components Required to support the ARM platforms

- Boot loader: U-Boot
- Pre-built Linux kernel, including sources and a binary image
- Configuration files for building the kernel
- Filesystem with pre-built utilities and applications
- Notes to install and setup Linux on your platform

PORTING DEVELOPED CODES IN ARM Based SYSTEM

Porting of image

- With the help of a host
- Host— a personal computer with Windows 2000 and above operating system or Linux RedHat 7 or 8 or 9
- Host Minimum 500 MB free space required
- COM port
- Terminal communication software (Minicom in Linux or Hyper terminal in Windows)

Target

- A board (evaluation board)
- RS232C cable for connection to host
- A cable for Ethernet
- Jumper (J1) for starting and operations later

Configuring the network

- Allocate IP address automatically
- IP of target same as of host
- Host is 192.168.1.56 and object board 192.168.1.55, network mask is 255.255.255.0 and broadcast IP is 192.168.1.1

Booting

- Use UBoot download tool in windows is used or uboot.bin compiled file
- Open the hyperterminal
- Set the baud rate = 115200, parity no and flow control no

Writing kernel and JFFSL file system in Linux

- Use Download command `uboot> tftp 22200000 uImage` to download at address 22200000.
- `uboot> nand erase` to erase the NAND flash memory.
- `uboot>nand write 22200000 200000` to write into the flash.
- `uboot>run ramdisk` to write the files in RAM disk at flash..

Enabling download functions from host to target

- Enables download and debuggung
- Remove the jumper J1

USB

- USB serial bus at COM5
- Connected from target to USB at host

Installing Drivers

- The drivers are installed after prompt ROMBoot> appears on screen

Burning Kernel and File System

- RUN SAM-BA.exe file
- Look in folder Inage
- Select address 0c08000
- Size for receive file = 0x1000 (4 kB)
- Select and open the file
dataflash_sam9263ek.bin
- The files then written in flash automatically

Actions on Open and select uboot.bin

- Reset all counter registers
- chip select 0 selected
- switch to SPI IO mode register
- wait for data flash ready
- Press send button on screen
- The uboot is written into data flash

A download link for ARM compiler and cross development kit for linux

**[http://www.denx.de/en/view/Software/
WebHome#Embedded Linux Development Kit](http://www.denx.de/en/view/Software/WebHome#Embedded_Linux_Development_Kit)**

Source and binary versions of the GNU GCC compiler

<http://www.codesourcery.com/>

- <http://www.codesourcery.com/sgpp/lite/arm/portal/release644>
- **must have Glibc library version 2.3 installed on your system.**

2.6 Linux kernel version

- pre-built uClinux images for **ARM Processors**
- ARM720T*, ARM920T*, ARM922T*,
- ARM926EJ-S, ARM1136J(F)-S
- ARM1176JZ(F)-S, ARM11 MPCore
- Cortex-A8
- Cortex-A9

File System Images

- <http://www.linux-arm.org/git> ael.git
- Include a set of applications tested with ARM RealView platforms

Complete filesystems

- Can be built using the infrastructure provided at <http://linux.onarm.com/>
- Include web browsers, media and other applications

Close and reset

- Close SAM-BA
- Remove USB slave line and press K10 button for reset at the evaluation board
- Start-up information appears at the hyper terminal

Linux Installation and Setup Procedure

- [http://www.linux-arm.org/LinuxReference/LinuxOnARMDev?
redirectedfrom=Main.LinuxOnARMDev](http://www.linux-arm.org/LinuxReference/LinuxOnARMDev?redirectedfrom=Main.LinuxOnARMDev)
- Download of Thumb-2 libraries and applications can be from the [http://www.linux-arm.org/LinuxReference/LinuxOnARMDev?
redirectedfrom= Main.LinuxOnARMDev.](http://www.linux-arm.org/LinuxReference/LinuxOnARMDev?redirectedfrom=Main.LinuxOnARMDev)

Images of Peripherals

- Generated using Linux 2.6 generates images using GCC 4.3.2 and GNU libc 2.8 compilers
- UART (PL010, PL011), Interrupt Controller (VIC, GIC), Ethernet (PL092), LAN9118, LAN91C111, USB, MMC, CompactFlash, Timers, L2C (220), KMI (PL050), RTC (PL030), VGA/LCD (PL110), AACI (PL041) and VFP

Summary

We learnt

- Linux 2.6.19 images
- Host-Target Approach
- Booting by uBoot
- UCLinux
- Porting is at flash in the target (evaluation board)

End of Lesson 09 on
Porting Developed Codes and
Linux in ARM Based System