Linux on the OMAP 1611

Brian A. Carter

April 17, 2008

Connectivity

- Serial port
  - Works with USB-to-serial adapters
  - Connect with ckermit
  - Built-in bootloader, uboot
  - 115200n1
- Ethernet port
  - 10BASE-T
  - Easily changeable MAC address (stored in flash)

Linux

- Kernel 2.6.9
- Heavily modified version released by TI
- Open-source
- Support for most (not all) of OMAP's features
- Compiled image size: just over a megabyte

Boot Sequence Overview

- Flip switch: power on
- Bootloader: uboot
- Network configuration: BOOTP
- Obtain boot image: TFTP
- Boot image: execute kernel image
- Root file system: NFS
- Network configuration: ipconfig, route
- SSH daemon: sshd

uboot

- uboot is a simple bootloader flashed into memory on OMAP boards
- Executed immediately when OMAP gets power
- Supports many common features:
  - Read/erase/write flash (blocks at a time)
  - Persistent environment variables (writeable to flash)
  - BOOTP network configuration
  - TFTP network transfer
  - Execute image
  - Automatic boot over network

BOOTP

- BOOTP servers assign IP addresses to clients that request them
- Client broadcasts IP address request
- If a BOOTP server is on the network, it responds with:
  - Client IP address (lookup based on client MAC address, or automatically selected from a range)
  - Client hostname
  - Server IP address (the one running TFTP)
  - TFTP location to grab kernel image from
  - Kernel image filename
- Client applies information received from server
- Sometimes requires a few tries before it works

TFTP

- **Not** the same as FTP
- “Trivial FTP”
- TFTP servers process requests from clients and transmit requested files to the client over the network
- Client must know the full path and name of the file it wants from the server
- In our case, used to transfer the kernel image from a host to the OMAP
- UDP protocol, so not entirely reliable

NFS

- “Network File System”
- NFS allows hosts to export directories to clients
- Clients mount NFS shares like any other file system
- Client must know full path of the share on the host
- Multiple clients can mount shares simultaneously
- Changes are propagated to all clients immediately
- Allows UNIX permissions to be applied over the network
- Linux kernel can be configured to automatically mount the root file system
  - NFS share path is provided as kernel boot option (in this case, by uboot from an environment variable stored in the OMAP’s flash memory)

The Boot Process

1. Connect serial cable from OMAP to a computer
2. Run ckermit on the computer and open a connection to the serial line
3. Connect an Ethernet cable to the OMAP to a private network with a BOOTP, TFTP, and NFS server
4. Turn OMAP power switch on
5. OMAP executes uboot

The Boot Process (2)

6. uboot restores environment variables from flash memory
7. uboot broadcasts BOOTP request to the network (connected via Ethernet)
8. BOOTP server responds to request and gives OMAP network configuration, which the OMAP applies
9. uboot broadcasts TFTP request for kernel image to the network
10. TFTP server responds to request and gives kernel image, which the OMAP writes to RAM

The Boot Process (3)

11. uboot executes the kernel image from RAM, providing kernel boot arguments from an environment variable
12. Linux boots and eventually needs a root file system
13. Linux sends an NFS request to the network
14. NFS server responds to request, verifies that OMAP has access, and provides access
15. Linux mounts root file system (NFS share) and continues booting

The Boot Process (4)

16. Linux calls init, which configures boot parameters
17. init configures network like a “regular” computer: gives access to local network
18. init sets the time over the network
19. init runs an SSH daemon (for remote connections via Ethernet)
20. init gives a shell prompt over the serial line

Compiling

- On-host compiling with native gcc, as, ld, etc.
  - **Very** slow
  - No special configuration needed to “just work”
- Cross-compiling for OMAP (ARM) architecture
  - Substantially faster, depending on host speed
  - A pain to set up, but works fine once you do
  - Great for packages that use autoconf

Resources

- Ryan's documentation:
  `http://robust.cs.utep.edu/~rcspring/omap/`
- TI's OMAP Linux web page:
  `http://linux.omap.com/`