

A13-OLinuXino-WIFI

How to configure your own kernel and install Debian

Marcin Twardak
twardakm@gmail.com

October 15, 2013

Contents

1	Introduction	2
2	Desktop configuring	2
2.1	Installing new software	2
2.1.1	Adding sources	2
2.1.2	Installing new software	3
3	Configuring U-Boot-Sunxi	3
4	How to compile the Linux Kernel	3
4.1	Downloading	4
4.2	Kernel configuration	4
4.3	Building	4
5	Make the SD-card	5
6	Copying Debian Distribution, U-Boot and Kernel	5
6.1	Installing Debian	5
6.2	Installing U-Boot and Kernel image	6
6.3	Choosing script.bin	6
7	Basic Debian configuration	6
7.1	Configuring Wi-Fi device	7
7.2	Locales error	7
7.3	Adding user	8
7.4	Configuring sudo	8
7.5	Installing ssh	8
7.6	Exiting chroot environment	9
8	References	9

Abstract

I describe how to build your own kernel and install Debian distro, on an embedded device called "A13-OLinUXino-WIFI". I will begin with instruction how to configure your desktop machine with Debian (in my case - testing version, but it should work on any Linux distribution with small changes). Then I show you, how to build your own kernel, it's easy, believe me. In the end how to configure SD card and put it all together. The whole text about installing Debian based on: <http://olimex.wordpress.com/2012/10/12/building-bootable-sd-card-with-debian-linux-image-for-a13-olinuxino/> with my changes and updates.

1 Introduction

First of all, I want to say, that my English is vary bad, and I know that, it's not my native language and I never had enough time to learn grammar. If you see bugs you can mail to me. I will appreciate that.

I hope that this tutorial is good for newbies people. I try to explain everything what I can. I know that not everything I can, because I am not a Linux expert, so if you want to understand more than I could tell you, you need to search in Web.

Enjoy :)

2 Desktop configuring

I work on Debian testing, I think it's a good choice for new Linux users, because it's quite easy (not so easy as Ubuntu of course, but I think, if you want to do something cool on OLinUXino you need to learn many things about Linux and Debian is excellent choice). I will not describe how to install and configure this distro on your desktop, there is lots of manuals in Internet. Look at: <http://www.debian.org/>.

2.1 Installing new software

You will need to install some software, like compilers, to build own kernel.

2.1.1 Adding sources

Open terminal and type: (instead of *nano* you could use other editors like *gedit*)

Command 1 `sudo nano /etc/apt/sources.list`

Add in the end of file new sources:

```
1 # A13 tools
2 deb http://www.emdebian.org/debian/ wheezy main
3 deb http://www.emdebian.org/debian/ sid main
```

Listing 1: New sources

Then type in the command line:

Command 2 `sudo apt-get update && sudo apt-get upgrade`

This will up to date your system.

2.1.2 Installing new software

Official tutorial (https://www.olimex.com/wiki/Build_Bootable_SD_Card_with_Debian) says that instead of my choice `gcc-4.5-arm-linux-gnueabi` you should install `gcc-4.6-arm-linux-gnueabi`. It is ok on other distro, but in Debian there is a dependency hell problem, I couldn't solve it, so I changed C compiler, but really it doesn't matter. To install everything you need type:

Command 3 `sudo apt-get install gcc-4.5-arm-linux-gnueabi ncurses-dev uboot-mkimage build-essential git`

Now you need to configure your compiler. But really you only need to change the name of one file, because makefiles in the next steps are configured to use other C compiler. Type now:

Command 4 `sudo mv /usr/bin/arm-linux-gnueabi-gcc-4.5 /usr/bin/arm-linux-gnueabi-gcc`

3 Configuring U-Boot-Sunxi

The first question is: what's that U-Boot-Sunxi? This is second level bootloader. It loads image of Linux kernel, which will build on the next step. If you want to know more, read: <http://www.stlinux.com/u-boot>.

Ok, now you should create a directory to do everything in it. I did it in my home directory (`/home/marcin/`) and I named it `Debian-olinuxino`, so if any command has words like that you should change it for your own.

Command 5 `cd ~ && mkdir Debian-olinuxino && cd Debian-olinuxino`

Now download u-boot-sunxi: (about 65 MB)

Command 6 `git clone -b sunxi https://github.com/linux-sunxi/u-boot-sunxi.git`

And change directory to the new one:

Command 7 `cd u-boot-sunxi`

Let's go to compile. In command 8 you need to use `CROSS_COMPILE` because probably you have a desktop computer, so architecture of your processor is x86 or amd64, but you need to compile this program to work with arm.

Command 8 `sudo make A13-OLinUxino CROSS_COMPILE=arm-linux-gnueabi-`

I hope you have any bug. To check everything went well, find files `u-boot.img` and `spl/sunxi-spl.bin`

Command 9 `ls u-boot.img spl/sunxi-spl.bin`

If you execute command 9 and you get any error, it means everything is fine.

4 How to compile the Linux Kernel

If you want to know more, visit: <http://www.wikihow.com/Compile-the-Linux-Kernel>, and many other sites.

I try to explain you in a few simple words, what is Kernel? It is something magic, and it communicates directly with computer devices (in desktops, tablets, embedded computers). If you are using for example USB drive, your system tells to Kernel, that he needs to send data to USB port and Kernel reads data from the program and sends to USB. Maybe it is not for professionals but I hope new Linux user will understand :)

4.1 Downloading

Go back to directory `/home/marcin/Debian-olinuxino` or whatever have you named that. And type command:

Command 10 `git clone https://github.com/linux-sunxi/linux-sunxi.git`

Sunxi represents the family of ARM SoC (System on Chip) designed for embedded systems, and made by Allwinner Tech. in Zhuhai (Guangdong, China). The most popular sunxi SoC model is the Allwinner A10 and the Allwinner A13. Their predecessor was an ARM9 named Boxchip F20 (sun3i) and their successors are A20 and Allwinner A31.

That definition of Linux Sunxi says everything. I recommend to visit: http://linux-sunxi.org/Main_Page and just try to read something.

Command 11 `cd linux-sunxi`

4.2 Kernel configuration

A13 processor has built in default configuration:

Command 12 `sudo make ARCH=arm a13_defconfig`

You will get information: `sudo make ARCH=arm a13_defconfig`. And exactly in this file (`.config`) is whole configuration of Linux Kernel. You can open that file and look at it. But as you can see, it wouldn't be nice to edit everything manually. And here we have tool `menuconfig`.

Command 13 `sudo make ARCH=arm menuconfig`

Have fun! But be carefully. If you change something wrong system will not work. But easy, you can't broke your device.

Defaults Input/Outputs pin are disabled. To enable them, you need to add (after saving all changes) to the end of `.config` `SUN4I_GPIO_UGLY = y`

4.3 Building

It's time to begin compilation of your own Linux Kernel!

Command 14 `make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- uImage`

And that's it :) You made it!.

To work properly kernel need modules. They are something like drivers:

Command 15 `sudo make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- INSTALL_MOD_PATH=out modules`

Command 16 `sudo make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- INSTALL_MOD_PATH=out modules_install`

5 Make the SD-card

To have good working Debian on your machine, you need Micro SD Card class 10. I personally tried with class 4 and it just didn't work. No error, no fireworks, it just didn't work, so be careful.

Connect your card to your computer. Now you need to know, how your card is represented by your system. The best way to do that is open program *GParted* (*sudo gparted*), if you don't have: *sudo apt-get install gparted*. Now on the right you have menu list with devices: */dev/sda*, */dev/sdb*... Now find your card and remember */dev/sdX*, where X is a, b, c... Go back to console application and execute:

Command 17 *sudo fdisk /dev/sdX*

Then delete all partitions: key *d* and \leftrightarrow as many times as you have partitions. Now you need to create 2 partitions. First to boot (fat32) and second to keep Linux Distribution (ext3). To create first partition press *n* then \leftrightarrow and \leftrightarrow , beginning sector 2048, last sector 34815. This is 17 MB, but you don't need more. Now again create new partition for distribution (*n*, \leftrightarrow , \leftrightarrow), first sector 34816, last should you leave defaults (last sector, so it will be all memory used). And write partition to your device by pressing *w*.

Now you need to create filesystems in your partitions, first you need to unmount, sometimes even restart your computer (table of partition will be updated in your system only after reboot, so it is good to restart now your computer). After reboot unmount partitions - maybe you get errors because they aren't mounted but better do that.

Command 18 *sudo umount /dev/sdX1* $\&\&$ *sudo umount /dev/sdX2*

And create filesystems:

Command 19 *sudo mkfs.vfat /dev/sdX1* $\&\&$ *sudo mkfs.ext3 /dev/sdX2*

6 Copying Debian Distribution, U-Boot and Kernel

6.1 Installing Debian

When you prepared your SD Card, it's time to copy files to that. First you need to download distro. I suppose to use ready to use distro, because building last very long. But you can read about BusyBox if you want to know more. Ok, go to your working directory: *cd /home/marcin/Debian-olinuxino* and begin downloading:

Command 20 *wget http://hands.com/~lkcl/mele_debian_armhf_minimal.cpio.gz*

Now mount your device:

Command 21 *sudo mount /dev/sdX2 /mnt* $\&\&$ *cd /mnt*

In command 21 is *sdX2* because second partition is used to have distribution, first partition is only to boot and have kernel image. Now unzip debian:

Command 22 *sudo gunzip -c /home/user/olinuxino/mele_debian_armhf_minimal.cpio.gz | cpio -i*
 $\&\&$ *sudo sync*

Sync is a function (program?) which checked, that the file is properly saved. If not you need to wait to the end of copying. Now you need to copy modules:

Command 23 `sudo cp -a linux-sunxi/out/lib/modules/3.4.61+/ /mnt/lib/modules/ && sudo sync`

You have now everything needed on the second partition so unmount it.

Command 24 `cd && sudo umount /mnt`

6.2 Installing U-Boot and Kernel image

To run your distro you need to copy now U-Boot and Kernel image to your first partition on SD Card, so mount it.

Command 25 `sudo mount /dev/sdX1 /mnt`

And copy:

Command 26 `sudo cp /home/marcin/linux-sunxi/arch/arm/boot/uImage /mnt && sudo sync`

Now go to the directory with U-Boot: `cd /home/marcin/u-boot-sunxi`. This steps are very important. If you read the old instruction there is a mistake (not mistake exactly, but it's not up to date) and it didn't work with that. I used instruction from http://linux-sunxi.org/Bootable_SD_card and everything was fine.

Command 27 `sudo dd if=spl/sunxi-spl.bin of=/dev/sdX bs=1024 seek=8 && sudo dd if=u-boot.img of=/dev/sdX bs=1024 seek=40 && sudo sync`

It is important to write `/dev/sdX` NOT `/dev/sdX1`.

6.3 Choosing script.bin

Olimex prepared some scripts, which you can use to change setting OLinuXino without recompiling kernel. That script is needed to boot your device. To download, visit:

<https://docs.google.com/open?id=0B-bAEPML8fw1NE1ERXRUZURTTUU>.

If you use VGA output choose script

`A13_script_files / script_GPIO_VGA / script.bin` otherwise,

`A13_script_files / script_GPIO_LCD_800x480 / script.bin` and copy it:

Command 28 `sudo cp script.bin /mnt && sudo sync`

Now unmount your device:

Command 29 `sudo umount /mnt`

And that is all! You can use your SD Card in OLinuXino and it should work. In the next section I will show you how to configure your Debian.

7 Basic Debian configuration

Now you have a working system, but probably you want to configure it some. I think the most important thing is to get wireless to work.

7.1 Configuring Wi-Fi device

To install WPA2 support you will need to chroot to your device. That means you will be working on your desktop PC but really you will work on your SD Card. If you want to know more visit: https://wiki.archlinux.org/index.php/Change_Root. First of all you need to install qemu module, which make your computer able to work on another architecture (computer → i386/x64, SD Card → arm).

Command 30 *sudo apt-get install qemu-user-static*

And mount your SD Card

Command 31 *sudo mount /dev/sdX2 /mnt && cd /mnt*

And do the following:

Command 32 *sudo mount -t proc proc proc/
sudo mount -t sysfs sys sys/
sudo mount -o bind /dev dev/
sudo mount -t devpts pts dev/pts/*

Now you can chroot to your new environment

Command 33 *sudo chroot /mnt /bin/bash*

Now you're in your new Debian! :) Install WPA2 support:

Command 34 *apt-get update && apt-get install wpasupplicant*

To edit your wireless setting I suggest to install very user-friendly text-editor: nano.

Command 35 *apt-get install nano*

Now edit your settings:

Command 36 *nano /etc/network/interfaces*

This is my example settings: (sometimes it isn't wlan0, instead of is wlan1, to check type: *ifconfig -a*)

```
1 auto wlan0
  iface wlan0 inet dhcp
3 wpa-driver wext
  wpa-ssid TP-Link
5 wpa-psk "My Password"
```

Listing 2: Interfaces settings

As you can see, if you have two or more words ssid or password you need to use quotations marks.

7.2 Locales error

As you could see if you use apt-get you will get locales errors. You can live with that. But if you don't get that type of errors you can just install package:

Command 37 *apt-get install locales-all*

7.3 Adding user

You should know, that you shouldn't work on root account! It's just dangerous. Of course nothing wrong will happen to your devices but system is open to potential attack or just your mistake. To add new user you should:

Command 38 `useradd -d /home/username -c "Full Name" -s /bin/bash username`

`/home/username` will be now you home directory. But first you need to create it. And of course user should be an owner.

Command 39 `mkdir /home/username && chown username /home/username`

And to protect your account you should make a password

Command 40 `passwd username`

By the way you should change your root password

Command 41 `passwd`

You should now always login into user account and use sudo.

7.4 Configuring sudo

Sudo is a very powerfull tool. I will show you only a basic configuration. First of all you need to install it.

Command 42 `apt-get install sudo`

And add your new user to sudo group (in other case your user will be not able to use sudo)

Command 43 `useradd -G sudo username`

The most important thing is I think to let user shutdown and reboot computer.

Command 44 `visudo`

And add one line:

```
user hostname =NOPASSWD: /usr/bin/shutdown -h now,/usr/bin/halt ,/usr/bin/poweroff ,/usr/bin/reboot
```

Listing 3: Sudo settings

Now your user will shutdown the computer with command: `sudo shutdown -h now` without password. You will can make scripts or something like that to make it easier.

7.5 Installing ssh

Ssh is very easy tool to connect to your device and you will able to do everything from your PC. Everything you need to know is IP address (`ifconfig -a`).

Command 45 `apt-get install openssh-server`

And open new tab in Terminal (in this tab you are logged to your desktop computer)

Command 46 `sudo apt-get install openssh-client`

To connect with your device you will need one command (executing on desktop):

Command 47 `ssh username@ip_adress`

7.6 Exiting chroot environment

To do that type:

```
Command 48 exit  
cd /  
sudo umount /dev/pts  
sudo umount /dev  
sudo umount mnt/sys  
sudo umount mnt/proc  
sudo umount mnt
```

That's it and end of my tutorial :) I hope that it helped you to begin exciting adventure with OLinuXino.

8 References

- [1] Bootable SD card - linux-sunxi.org, http://linux-sunxi.org/Bootable_SD_card.
- [2] Building bootable SD-card with debian linux image for a13-OLinuXino, <http://olimex.wordpress.com/2012/10/12/building-bootable-sd-card-with-debian-linux-image-for-a13-olinuxino/>.
- [3] Change root - ArchWiki, https://wiki.archlinux.org/index.php/Change_Root.
- [4] How to compile the linux kernel, <http://www.wikihow.com/Compile-the-Linux-Kernel>.
- [5] LC_CTYPE, LC_ALL problems on a fresh install of debian squeeze by mrilhan - ilhan demirok, http://mrilhan.com/247/lc_ctype-lc_all-problems-on-a-fresh-install-of-debian-squeeze/.
- [6] SD card - fixing shutdown bug + updated steps to building your own card, <https://www.olimex.com/forum/index.php?topic=518.0>.
- [7] U-boot | STLinux, <http://www.stlinux.com/u-boot>.