How to build a TonidoPlug2 kernel

The content of this page is contributed by tonido forum member pmcallihan.

This is a step by step procedure to build the original kernel that comes with your tp2. All responsibility is yours. The software built here is to replace the kernel and modules on an existing operational sata or usb disk. The purpose is to assure yourself that you can build a functional kernel. I highly recommend that you build a stable kernel before creating a new one with different capabilities.

The cross compiler used in this procedure is only functional on a linux desktop. This was done with a debian/ubuntu style os called Zonin. It is my current choice. Linux is linux, the procedure should be the same except for installing dependencies.

1) Open a command line terminal.

2) Download a cross compiler.
wget http://www.plugcomputer.org/405/us/gplugd/tool-chain/arm-marvell-linux-gnueabi.tar.bz2

3) Install the cross compiler in the /opt directory.
sudo tar xvjpf arm-marvell-linux-gnueabi.tar.bz2 -C /opt/

4) Delete the download.
rm arm-marvell-linux-gnueabi.tar.bz2

5) Install software dependencies.
sudo apt-get install build-essential automake autoconf libtool pkg-config intltool
sudo apt-get install libncurses5-dev uboot-mkimage

6) Make a working directory and go to it. Mines is called tonido
mkdir tonido
cd tonido

7) Download tonido kernel source code.
wget http://www.tonidoplug.com/partners/topkick/kernel/source/kernel-2.6.31-topkick1281p2-codelathe-20111122.tar.bz2

8) Install kernal source.
tar xvjpf kernel-2.6.31-topkick1281p2-codelathe-20111122.tar.bz2

9) Delete the download.
rm kernel-2.6.31-topkick1281p2-codelathe-20111122.tar.bz2

10) Simplify the package name and move to it.
mv kernel-2.6.31-topkick1281p2-codelathe-20111122 kernel-2.6.31
cd kernel-2.6.31

11) Set the environment for the build. The declarations are only valid for the life of the command shell. If you exit the shell and open a new shell to continue, you will need to set the enviroment again. Highly recommend cut and paste on these commands as the statement need to be exact.
export PATH=/opt/arm-marvell-linux-gnueabi/bin:$PATH
export CROSS_COMPILE="arm-marvell-linux-gnueabi-"
export ARCH=arm

12) Clean up build for start fresh.
make mrproper

13) Create a default configuration file for the build. This is the shipped kernel.
make mv88f6281_gw_defconfig

14) If were going to change the kernel or modules. This is where the configuration is modified. The README file in this directory tells you about different program that will make the changes. I will write another howto telling how to add audio to the tp2. That will change the configuration file. I suggest just building a clean kernel for now using the default configuration.

15) Build the kernel.
make uImage

16) Build the modules
make modules

17) Make a directories to store the new software.
mkdir ../current_kernel
mkdir ../current_kernel/boot

18) Store the modules.
make INSTALL_MOD_PATH=../current_kernel modules_install

19) Move the kernel into the new directory and change is access.
cp arch/arm/boot/uImage ../current_kernel/boot/
chmod 744 ../current_kernel/boot/uImage

The new kernel is at ../current_kernel/boot/uImage
The new modules are at ../current_kernel/lib/modules

I assuming your boot directly from the tp2 and that you have a bootable disk that is not booted but mounted at /media/disk1part1/.

Using the shell
ssh tp2 -l root
cd /media/disk1part1/boot
mv uImage uImage_save
cd /media/disk1part1/lib
mv modules modules_save
exit

Now back at command shell at ~/tonido/kernel-2.6.31
scp ../current_kernel/boot/uImage root@tp2:/media/disk1part1/boot/
scp ../current_kernel/lib/modules/ root@tp2:/media/disk1part1/lib/

reboot and I hope it works.
