

HOWTO Build a Bootable USB Thumbdrive from a New BSP

Introduction

The Linux and Open Source Software world changes fast. Renesas is working hard to maintain highly active participation level in both the Linux Kernel and Linux User spaces. With highly active development, code changes daily. To facilitate adoption of the latest developments within the Open Source communities actively working on SuperH development boards and other hardware IP, Renesas frequently updates and makes available on the SuperH Linux and Open Source Portal Board Support Packages, drivers, codecs, etc.

This HOWTO describes the process of building a new, boot-able USB key from a new Board Support Package downloaded directly from the SuperH Linux and Open Source Portal.

Prerequisites

Although steps below will satisfy some of the prerequisites listed here, they are provided to ensure that you have a complete understanding of this process. This HOWTO was developed using the following components:

- Linux desktop system (Ubuntu 9.1)
- USB Thumbdrive (2GB)
- MS7724 Development Board
- Board Support Package (v.1.0.3 Feb.25, 2010)
 - packed-bsp-1.0.3-r0p7724lc.tar.bz2

Step 1: Obtain the latest BSP

The fastest way to the latest BSP Updates is from the portal's Frontpage. Within the Recent BSP Updates section, you may click on the applicable Target link to learn about the changes within each version of the BSP. At the bottom of the latter page, you can click on the latest version of the BSP in the Release column to move onto the Download page. Alternatively, from the Frontpage, click on the Release column link directly to go to the Download page.

The Download page offers the option to Download either a complete packed archive of the BSP, or each component of the BSP individually. This HOWTO utilizes the packed archive. Click on the large red Download button under Packed Archive heading.

At this point you will be presented with a small forum to complete (legally required). Assuming you've completed the and can continue with downloading clicking the Downloading button at the bottom of the forum will begin the download. Depending on how you've configured your system to handle downloads, the file will either be saved to your hard drive automatically or you'll be asked to select a location to save.

Step 2: Preparing a USB Thumbdrive

This HOWTO recommends using a 2GB USB Thumbdrive. We will format the drive into 2 partitions as follows:

Drive Size	2GB
Partitions	Partition 1: 512MB VFAT stores zImage and romImage Partition 2: 1.5GB EXT3 store Debian rootfs

- Insert the USB Thumbdrive into a USB slot on your desktop computer. Ubuntu will (in most cases) recognize the new device and assign it a device node. Note that this assumes the thumbdrive is unformatted. If it is formatted, Ubuntu will automount the drive and pop up a File Browser window (Nautilus on my system). If this occurs, select File -> Unmount to unmount the drive.
- Determine the assigned device node by looking at dmesg output:

```
$dmesg
```

At the bottom of the dmesg output you should see something similar (though likely not identical to) this:

```
[1468349.520436] usb 1-2.1: new high speed USB device using ehci_hcd and address 21
[1468349.631924] usb 1-2.1: configuration #1 chosen from 1 choice
[1468349.634122] scsi7 : SCSI emulation for USB Mass Storage devices
[1468349.634294] usb-storage: device found at 21
[1468349.634300] usb-storage: waiting for device to settle before scanning
[1468354.630950] usb-storage: device scan complete
[1468356.844457] scsi 7:0:0:0: Direct-Access I-O DATA USB Flash Disk 3A PQ: 0 ANSI:
2
[1468356.845202] sd 7:0:0:0: Attached scsi generic sg8 type 0
[1468356.852499] sd 7:0:0:0: [sdh] 4014080 512-byte logical blocks: (2.05 GB/1.91 GiB)
[1468356.853092] sd 7:0:0:0: [sdh] Write Protect is off
[1468356.853102] sd 7:0:0:0: [sdh] Mode Sense: 03 00 00 00
[1468356.853108] sd 7:0:0:0: [sdh] Assuming drive cache: write through
[1468356.856704] sd 7:0:0:0: [sdh] Assuming drive cache: write through
[1468356.856716] sdh: sdh1 sdh2
[1468356.861580] sd 7:0:0:0: [sdh] Assuming drive cache: write through
[1468356.861592] sd 7:0:0:0: [sdh] Attached SCSI removable disk
```

This tells me that Ubuntu has detected my USB Thumbdrive and assigned it device node: /dev/sdh Specifically, my drive already has 2 partitions on it, therefore, Ubuntu assigned subnodes: /dev/sdh1 and /dev/sdh2. The rest of the HOWTO will use /dev/sdh as the device node, please substitute your device node as applicable.

- Partition the Thumbdrive From the Linux command line, using fdisk enter the following:

```
$ sudo fdisk /dev/sdb
Command (m for help): d
Selected partition 1
Command (m for help): d
No partition is defined yet!
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-245, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-245, default 245): 63
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (64-245, default 64):
Using default value 64
Last cylinder, +cylinders or +size{K,M,G} (64-245, default 245): 245
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

- Format the new partitions as follows:

```
$ sudo mkfs.vfat /dev/sdb1
mkfs.vfat 3.0.1 (23 Nov 2008)

$ sudo mkfs.ext3 /dev/sdb2
mke2fs 1.41.4 (27-Jan-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
93888 inodes, 375519 blocks
18775 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=385875968
12 block groups
32768 blocks per group, 32768 fragments per group
7824 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Great, now we have a 2GB Thumbdrive with 2 partitions formatted for our kernel image and roots.

Step 3:Unpack BSP

Now we need something to copy to the thumbdrive. From a Linux command line change to the directory where you downloaded the BSP: packed-bsp-1.0.3-r0p77241c.tar.bz2 Make sure you have about 2GB of space and untar the package. I like to create a bsp temporary directory in my Downloads directory as follows:

```
$cd ~Downloads
$mkdir bsp
$cd bsp
$tar jxf ~Downloads/packed-bsp-1.0.3-r0p77241c.tar.bz2
```

At this point, the archive directory tree should look like the following: (note that you may need to apt-get install the tree program if you don't have it installed already)

```

|-- bootloader_kernel
|   |-- bin
|   |   |-- romImage
|   |   |-- src -> ../kernel/src/
|-- directfb
|   |-- bin
|   |   |-- directfb-bin.tar.bz2
|   |   |-- src
|   |       |-- directfb-src.tar.bz2
|-- host_environment
|   |-- bin
|   |   |-- chroot4cross-20100223.tar.bz2
|   |   |-- src -> ../rootfs/src/
|-- kernel
|   |-- bin
|   |   |-- linux-kernel-bin-20100225.tar.bz2
|   |   |-- src
|   |       |-- linux-kernel-src-20100225.tar.bz2
|-- multimedia
|   |-- bin
|   |   |-- shcodecs100-sighttpd090-uiomux100.tar.bz2
|   |   |-- src
|   |       |-- libshcodecs-20100217.tar.gz
|-- rootfs
|   |-- bin
|   |   |-- debian-rootfs-20100223.tar.bz2
|   |   |-- src
|   |       |-- src-pkg-20091117.tar.bz2
|-- rootfs_tiny
|   |-- bin
|   |   |-- initramfs-20100222.tar.bz2
|   |   |-- src
|   |       |-- tiny-rootfs-20100222.tar.bz2
|-- shjpeg
|   |-- bin
|   |   |-- shjpeg-bin.tar.bz2
|   |   |-- src
|   |       |-- libshjpeg-20091116.tar.bz2

```

Step 4: Copy content to the USB Thumbdrive

From the Linux command line enter the following to copy and extract the necessary software onto the USB thumbdrive. Note that I'm doing the following from my temporary ~/Downloads/bsp/ directory and remember to substitute your applicable device node where you see /dev/sdh.

Populating Partition 1:

```

$ sudo mount /dev/sdh1 /mnt/usbdisk/
$ sudo cp bootloader_kernel/bin/romImage /mnt/usbdisk/
$ sudo cp kernel/bin/zImage_usbboot /mnt/usbdisk/
$ ls /mnt/usbdisk/
romImage zImage_usbboot
$ sudo cp zImage_usbboot zImage
$ sudo umount /mnt/usbdisk/

```

Populating Partition 2:

```

$ sudo mount /dev/sdh2 /mnt/usbdisk/
$ cd /mnt/usbdisk/
$ sudo tar jxf ~/Downloads/bsp/rootfs/bin/debian-rootfs-20100223.tar.bz2
$ ls /mnt/usbdisk/
debian-rootfs
$ sudo mv debian_rootfs/* ./
$ sudo rm -rf debian_rootfs
$ ls
bin  demo  etc  lib  mnt  proc  sbin  srv  tmp  var
boot dev  home media opt  root  selinux sys  usr

```

Install the software that takes advantage of the MS7724 hardware:

```
$sudo tar jxf ~/Downloads/bsp/multimedia/bin/shcodecs100-sighttpd090-uiomux100.tar.bz2
$sudo tar jxf ~/Downloads/bsp/shjpeg/bin/shjpeg-bin.tar.bz2
$sudo tar jxf ~/Downloads/bsp/directfb/bin/directfb-1.4.2-bin-20091118.tar.bz2
$cd
$sudo umount /mnt/usbdisk/
```

Step 5: Boot USB Thumbdrive in MS7724

- Insert USB Thumbdrive into MS7724 USB slot and power on. Assuming your minicom setup is configured as described in MS7724 Beginners Guide you should see the normal boot up messages displayed.