Embedded Linux Build Procedures

Buildroot Procedure:

1. Download the buildroot source code from www.buildroot.org
2. Create one folder e.g. buildroot and copy the source code to that folder
3. Untar buildroot source

   veda@emb:buildroot$ tar xvf buildroot-2011.02.tar.bz2

4. After this we get a folder with name buildroot-2011.02
5. Enter into that folder using following command

   veda@emb:buildroot$ cd buildroot-2011.02

6. Now we have to configure the buildroot source code according to our board using menuconfig.

   veda@emb:buildroot$ make menuconfig

   After this command you'll get a one graphical window

7. The first option is for target architecture in our case it is **arm** hence change it **arm**

   Target Architecture (arm) --->

8. Second option is about target variant in our case it is **arm920t.**

   Target Architecture Variant (arm920t) --->

9. Third option is about target **ABI** select **EABI**

   Target ABI (EABI) --->

10. After this go into Toolchain option.

Toolchain --->

11. Select toolchain type as **Buildroot toolchain**

Toolchain --->
Toolchain type (Buildroot toolchain) --->

12. Select the kernel headers

Toolchain --->
Kernel Headers (Linux 2.6.37.x kernel headers) --->

13. Select the **uClibc** version

Toolchain --->
uClibc C library Version (uClibc 0.9.30.x) --->

14. Select the **Binutils** version

Toolchain --->
Binutils Version (binutils 2.21) --->

15. Select the **GCC** compiler version

Toolchain --->
GCC compiler Version (gcc 4.3.x) --->

Just save the configuration and come out.

- If you have internet connection the compilation process of buildroot is automated.
- If you don't have it then we haver to create the dl folder in the parent directory of buildroot source code and copy the source codes of uClibcm, binutils etc into that folder and compile
- Make sure that you have to copy the same versions of source codes that you selected in the configuration options
- After this give **make** command

 # make

On success, cross-tool-chain gets installed under

$<buildroot_src>/output/host/usr/bin/ directory with a **"arm-linux-"** prefix

Procedure to build U-boot image:

1. Download the u-boot source code from http://www.denx.de/wiki/U-Boot
2. Untar the source code

   tar xvf u-boot-2009.11.tar.bz2
3. Enter into the **u-boot** source code
4. Give the following command in case of **kb9202** board

   make kb9202_config
5. Set the cross compiler path. In my case it is located in **/home/veda/emb/buildroot/buildroot-2011.02**

/output/host/usr/bin hence the command goes like this

$ **PATH=$PATH:/home/veda/emb/buildroot/buildroot-2011.02 /output/host/usr/bin**

6. Now compile the u-boot with the following command

   $ make CROSS_COMPILE=arm-linux-
7. After successful compilation you'll get the **u-boot.bin**

Procedure to build the Kernel Image:

1. Download the latest kernel source from www.kernel.org
2. Untar the source code and enter into the source code directory

$**tar xvf linux-2.6.37.2.tar.bz2**

$cd linux-2.6.37.2

3. Check for the **kb9202_defconfig** file in **arch/arm/configs** folder if your latest kernel don't have this default configuration file copy it from the older kernel version and copy to the same directory that **arch/arm/configs**

4. Set the cross compiler path

5. Give the following command

$**make ARCH=arm kb9202_defconfig**

6. Now compile the kernel using **make**

$make CROSS_COMPILE=arm-linux- uImage

7. After successful compilation the image will be created in **arch/arm/boot** folder

Procedure to build busybox:

1. Download the busybox source code from www.busybox.net

2. Untar the source code and enter into the source code directory

3. Give the following command to configure the busybox

$make menuconfig

4. Select **Build BusyBox as a static binary** which is located in the following path

Busybox Settings --->
Build Options --->
[*] Build BusyBox as a static binary (no shared libs)

5. Select **mdev** option which is located in the following path

Linux System Utilities --->
[*] mdev (NEW)

6. Select the remaining binaries according to your needs

7. Set the cross compiler path as explained above.

8. Compile the busy box

   $make CROSS_COMPILE=arm-linux-

9. Give make install command

   $make install

10. After successful compilation **_install** folder will be create in the parent directory of busybox source code. In that directory all the binary will be there.