Building TCI6614 U-Boot and Linux Kernel

Introduction

This topic describes how to build the TCI6614 and TCI6638 u-boot and linux kernel. However the content is generic and applies to other platforms as well. A Ubuntu 10.04 Linux Host is required. It is assumed that the reader went through the Linux Appleton Training Introduction and has set up the Ubuntu 10.04 linux host environment.

Initial Git Linux Host Setup

The u-boot and linux code are managed in git repositories. They are located at the following sites tci6614 u-boot, tci6614 kernel

Config apt-get by adding following to /etc/apt/apt.conf (create the file if one does not exist). Root access required to modify/create this file.

```
ACQUIRE {
  http::proxy "http://wwwgate.ti.com:80"
}
```

For proxy set up add following to ~/.bashrc

```
export http_proxy="http://webproxy.ext.ti.com:80"
export ftp_proxy="http://webproxy.ext.ti.com:80"
export https_proxy="http://webproxy.ext.ti.com:80"
export no_proxy="ti.com"
export GIT_PROXY_COMMAND=$HOME/git-proxy.sh
```

Source .bashrc file after updating it:

```
>source ~/.bashrc
```

cut-n-paste following in $HOME/git-proxy.sh (create the file if it does not exist) and save the file

```
#!/bin/sh
if [ $(getent hosts intranet.ti.com|cut -d' ' -f 1)x = "127.0.0.1x" ]
then
                # this machine is inside TI's network
                if echo $1 | grep ti.com > /dev/null
                then
                        # ... and so is the remote machine
                        socat - tcp:$1:$2
                else
                        socat - proxy:wwwgate.ti.com:$1:$2,proxyport=80
                fi
else
        socat - tcp:$1:$2
fi
```

Install socat by typing following in a terminal

```
>sudo apt-get install socat
```

Perform the following:

```
sudo apt-get install -y build-essential git-core expect automake
echo "no" | sudo dpkg-reconfigure -f teletype dash
```

Installing the Tool Chain

The following is performed on the Linux host. Download and install CodeSourcery ARM tool chain version 2009-Q1 from http://www.codesourcery.com/sgpp/lite/arm/portal/package4573/public/arm-none-linux-gnueabi/arm-2009q1-203-arm-none-linux-gnueabi.bin .

```
Select the Recommended Package IA32 GNU/Linux Installer.
```

Execute as follows

If needed follow the recommendations to update

After installing the toolchain, related environment variables must be set.

```
export PATH=<path-to-code-sourcery-bin>:$PATH
export CROSS_COMPILE=arm-none-linux-gnueabi-
export ARCH=arm
```

Example of CodeSourcery toolchain path added to the PATH environment variable:

```
export PATH=/home/user/CodeSourcery/Sourcery_G++_Lite/bin:$PATH
```

The environment variables can be added to the $HOME/.bashrc file. Root access is required to modify this file. A simple text editor, gedit, can be used to perform these changes:

```
sudo gedit $HOME/.bashrc
```

After updating the .bashrc file and check the value of the PATH environment variable to make sure it was updated:

```
>source ~/.bashrc
>echo $PATH
```

You should something similar to:

```
user@ubuntu:~/workdir/u-boot-tci6614$ echo $PATH
/home/user/CodeSourcery/Sourcery_G++_Lite/bin:/home/user/bin:/usr/local/sbin:/usr/local/b
in:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

Building U-Boot

Go to the $HOME/workdir directory

```
$ cd $HOME/workdir
```

Clone u-boot-tci6614 git tree and build u-boot:

```
$ git clone git://arago-project.org/git/projects/u-boot-tci6614.git
$ cd u-boot-tci6614
$ make tci6614_evm_config
$ make
```

u-boot binary will be generated in the same directory.

mkimage Utility

The u-boot building process generates in u-boot-tci6614/tools directory a utility called mkimage which is required to build the kernel Image.

We will create a directory $HOME/bin and we will copy the mkimage utility located in u-boot-tci6614/tools directory to the $HOME/bin. After that we will add this directory to the PATH so that the kernel will find it during kernel the building process

```
$ mkdir -p $HOME/bin
$ cp $HOME/workdir/u-boot-tci6614/tools/mkimage $HOME/bin
```

Update the PATH in $HOME/.bashrc and don't forget to source the .bashrc as previously described.

Building Linux Kernel

Go to the $HOME/workdir directory

```
$ cd $HOME/workdir
```

Clone linux-tci6614 git tree and build the kernel:

```
$ git clone git://arago-project.org/git/projects/linux-tci6614.git
```

If you see the following **warning** the HEAD is not defined in the git repository

```
icesar@icesar-laptop:~/workdir$ git clone git://arago-project.org/git/projects/linux-
tci6614.git
Initialized empty Git repository in /home/icesar/workdir/linux-tci6614/.git/
remote: Counting objects: 2242201, done.
remote: Compressing objects: 100% (351065/351065), done.
remote: Total 2242201 (delta 1869911), reused 2242201 (delta 1869911)
Receiving objects: 100% (2242201/2242201), 531.30 MiB | 441 KiB/s, done.
Resolving deltas: 100% (1869911/1869911), done.
warning: remote HEAD refers to nonexistent ref, unable to checkout.
```

We will create and checkout a branch called my_master to copy the content of the master branch

```
$ cd linux-tci6614
$ git branch my_master origin/master/master
$ git checkout my_master
```

Next we will create a branch called my_scmcsdk_2_4 to copy the content of the release DEV.SC-MCSDK-02.00.00.04

```
$ git branch my_scmcsdk_2_4 tags/DEV.SC-MCSDK-02.00.00.04
```

Let's switch to the my_scmcsdk_2_4 branch as follows

```
$ git checkout my_scmcsdk_2_4
```

You can check that the my_scmcsdk_2_4 branch is active by performing

```
$ git branch
```

You should get something similar to the following. The asterisk is in front of the active branch

```
icesar@icesar-laptop:~/workdir/linux-tci6614$ git branch
  my_master
* my_scmcsdk_2_4
```

Now we can build the SC-MCSDK_02.00.00.04 kernel image

```
$ make tci6614_evm_defconfig
$ make uImage
```

uImage will be generated in the arch/arm/boot directory. **Note** - make sure you don't build Image instead of uImage.