

# SIXNET Embedded Linux Development Kit (ELDK)

## Installation

The SIXNET Embedded Linux Development Kit (ELDK) is a complete Linux based cross development environment. It contains an i386 cross GCC tool chain for compiling code for ARM processors. This lets you conveniently build programs for SIXNET ARM-based devices such as managed switches (ET-5MS, newer models of the ET-9MS, etc.) on a faster Linux desktop. Contact SIXNET for the latest information on which devices use an ARM processor.

The ELDK has been successfully installed on the following Linux distributions:

- Fedora Core 1,2,3,4
- Redhat 7.3, 8.0, 9.0
- SuSE 8.x, 9.1, 9.2, 9.3
- Debian 3.0, 3.1(sarge)

## Obtaining the ELDK for ARM

The DENX ELDK can be downloaded directly from <http://www.denx.de> or several mirror sites listed there.

## Installing the ELDK for ARM

This installation guide assumes you are using bash. You need your root password (or ask your system administrator to create /opt/sixnet and give you read/write/execute permission).

### Step 1:

Make the directory /opt/sixnet (and give ownership to the desired user and group):

```
> su -- switch to root
# mkdir -p /opt/sixnet/eldk -- make the directory and its parent, /opt
# chown user:group /opt/sixnet/eldk -- give 'user' and 'group' ownership of the directory
# exit -- give up root control
```

### Step 2:

Download the ELDK 3.1.1 ARM iso image, arm-2005-03-06.iso from the <http://www.denx.de> or one of the mirror sites. Either make a CD from this image and mount the CD or mount the image as a loopback file system. You must be root or have root privileges to do the either of these steps.

Mount the CD or image containing the ELDK.

#### a. Option A – mounting the CD

```
# mount /dev/cdrom /mnt/cdrom
```

#### b. Option B – mounting image as a loopback file system

```
# mount -o loop -t iso9660 arm-2005-03-06.iso /mnt/cdrom
```

# SIXNET Embedded Linux Development Kit (ELDK)

## Step 3:

Install the ELDK to the directory created in step 1.

```
> cd /mnt/cdrom
> ./install -d /opt/sixnet/eldk arm
```

**Note:** you should now have an /opt/sixnet/eldk subdirectory with the cross-compiler, libraries and sample programs inside.

## Step 4:

Add /opt/sixnet/eldk/bin and /opt/sixnet/eldk/usr/bin to PATH:

In BASH this means adding the following line to ~/.bash\_profile along with all the other PATH statements:

```
PATH=/opt/sixnet/eldk/bin:/opt/sixnet/eldk/usr/bin:$PATH
```

In csh:

```
Setenv PATH /opt/sixnet/denx/bin:/opt/sixnet/denx/usr/bin:$(PATH)
```

Use your favorite text editor (vi, emacs, pico). Alternately, as root you could edit /etc/profile to make this path global for all users. But it is probably better to do it for each user individually.

## Using the ELDK

### Step 1:

If '/opt/sixnet/bin' is missing from the path, set it now.

```
using bash: export PATH=/opt/sixnet/denx/bin:/opt/sixnet/denx/usr/bin:${PATH}
```

```
using csh: setenv PATH /opt/sixnet/denx/bin:/opt/sixnet/denx/usr/bin:${PATH}
```

### Step 2:

Compile the program:

To compile a program, just use 'arm-Linux-' pre-pended to the normal gcc commands.

Example:

```
arm-linux-gcc file.c -o my_program -- turns 'file.c' source code into 'my_program'
arm-linux-objdump -d my_program -- dumps to the screen the code for 'my_program'
```

You can also use make files in this fashion. You can find tutorials for make files on the Internet.

### Step 3:

Load the program (my\_program) to the SIXNET device (such as a Managed Switch) using ftp.

#### Using FTP and Telnet:

**Note:** FTP and Telnet must first be enabled in the SIXNET device. To enable ftp and telnet you must install the package msdev.tgz. See the "Linux Development in Managed Switches" tech note to find instructions on installing this package.

# SIXNET Embedded Linux Development Kit (ELDK)

Now run your favorite FTP client (or just type "ftp" at the command prompt) and open a connection to the SIXNET device (see example below). Login is "anonymous". No password is necessary. (Hit <enter>.)

Example: if your SIXNET device's IP address is 10.2.0.1, you would type:

```
ftp 10.1.0.1 [enter]
anonymous      [enter]
[enter]
```

Next, 'cd' to the existing location you want to place 'my\_program' (typically /usr/local/bin) and use 'put' in binary mode to upload your file:

```
bin [enter]
cd /usr/local/bin [enter]
put my_program [enter]
quit [enter]
```

Optionally, you can use the mkdir command and create a folder to upload your file into:

```
mkdir my_folder [enter]
bin [enter]
cd /my_folder [enter]
put my_program [enter]
quit [enter]
```

## Step 4:

Start your program (my\_program) running in the SIXNET device.

To start the program running, make a Telnet connection to the SIXNET device. (Note: Telnet must be enabled in the unit. See "Using FTP and Telnet" above.) The default login/password is 'root'. There is no password. Then 'cd' to the directory where you placed the program:

```
telnet 10.1.0.1 [enter]
root [enter]
cd /usr/local/bin [enter]
```

Note: You may need to set the execute flag for the file:

```
chmod +x my_program
```

To run the program, type:

```
./my_program
```

# SIXNET Embedded Linux Development Kit (ELDK)

## Configuring the SIXNET device to start your program automatically:

Your program, and the appropriate path to it, will need to be listed in a startup script file (.sh file) located in the /etc/user.d folder in the SIXNET device. Any startup script files found in the /etc/user.d folder will be launched automatically by the SIXNET device after powerup. For more information on the behavior of startup scripts, refer to the User Script Management technical note (user\_script\_management.pdf). The latest IPm and other Linux-related technical notes are available for download from:

[http://www.sixnetio.com/html\\_files/ipm\\_tech\\_notes/ipm\\_technotes.htm](http://www.sixnetio.com/html_files/ipm_tech_notes/ipm_technotes.htm)

Here is an example of a startup script file:

```
#!/bin/sh
/usr/local/bin/my_program
```

**Note:** Your startup script file must be saved or resaved in such a manner that each line does not end with a carriage return. You can assure that the file is in the correct format by running Telnet and issuing the following commands on your file (called "myfile.sh" in this example):

```
cd /etc/user.d
dos2unix -u myfile.sh
chmod 755 myfile.sh
```

If you experience problems such as:

```
bash: ./my_program: Permission denied
Use 'chmod +x my_program' to make the file executable.
```

```
bash: ./my_program: cannot execute binary file
The file is probably compiled for a desktop, not the IPm. Make sure you didn't use 'gcc' instead of
'arm-linux-gcc'.
```