

Building the GNU Tools for the SH (Cygwin host)

The following shows how to build the GNU tools for the SH for use on the CYGWIN host. Although the binary files for the CYGWIN host are already released from the SH/M32R T-Engine Home Page, the explanation below is given for the case where the GNU tools need to be built as may be needed when you want to update the compiler. The source code of GNU tool can be obtained from the Free Software Foundation (FSF) or the KPIT Cummins Infosystems Limited (KPIT GNUtools) etc. , and it explains the build method by using what downloaded from the SH/M32R T-Engine Home Page as one example here.

1. Downloading the GNU Tool Source Code

(1) Download the source code for the GNU tools. Select GNU tools from the “Tools” menu and then download the files listed below. (See Firure 1)

- binutils-XX.tar.gz (XX denote version numbers)
- gcc-YY.tar.gz (YY denote version numbers)
- newlib-ZZ.tar.gz (ZZ denote version numbers)



Figure 1 GNU Tool Download File Select Screen (Example)

2. How to Build the GNU Tools

The following explains how to build the GNU tools. In the example here, we'll install the tools in the `/usr/local/sh-elf` directory and those hierarchically below it. Note that if GNU tools already exist in `/usr/local/sh-elf`, you need to change the existing file directory name or change the directory name in which you want to build and install.

- (1) Create a directory in which to build, and expand the build script in it. Here, `/home/tkernel/shtools/buildshgcc` is assumed to be the build directory.

```
$ mkdir /home/tkernel/shtools/buildshgcc
$ cd /home/tkernel/shtools/buildshgcc
$ cp /cygdrive/d/tools/buildscript.tar.gz .
$ tar zxvf buildscript.tar.gz
```

- (2) Extract the GNU tool source code you've downloaded. Here, the downloaded source code is assumed to be stored in the `/home/tkernel/shtools/buildshgcc` directory.

```
$ tar zxvf binutils.tar.gz
$ tar zxvf gcc.tar.gz
$ tar zxvf newlib.tar.gz
```

When extracted, source directories `binutils`, `gcc`, and `newlib` will be created.

- (3) Execute the setup file and register the information necessary for the build operation. Correct the `setup.sh` file to make it suitable for the environment used.

```
$ . setup.sh
```

- (4) Go to the build directory, and build `Binutils` in it. Correct `buildbinutils.sh` to make it suitable for the `binutils` source code used.

```
$ cd build_binutils
$ . buildbinutils.sh
```

- (5) Add a **PATH** and check to see if the tools built by `Binutils` start up. In the explanation here, this is performed in `sh-elf-as`.

```
$ cd ..
$ . appendpath.sh
$ sh-elf-as -version
GNU assembler 2.16
Copyright 2005 Free Software Foundation, Inc.
This program is free software; you may redistribute it under the terms of
the GNU General Public License. This program has absolutely no warranty.
This assembler was configured for a target of `sh-elf'.
```

- (6) Build and install only the C compiler from GCC. Modify *buildgccPART1.sh* (modify directory name) to make it suitable for the GCC source code used.

```
$ cd build_gcc
$ . buildgccPART1.sh
```

- (7) Build and install *Newlib*.

```
$ cd ../build_newlib
$ . buildnewlib.sh
```

- (8) Build and install the remaining GCC compiler, language runtime library, and language support library.

```
$ cd ../build_gcc
$ . buildgccPART2.sh
```

- (9) Check to see if the GNU tools have been built and installed.

```
$ cd ..
$ sh-elf-gcc -version
sh-elf-gcc (GCC) 4.0
Copyright (C) 2005 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```