GCC for ARM, AVR and BlackFin

Author : Andras Tantos

What's included

Using the following links you can download the GNU compiler toolset, compiled for the various processor targets. The toolset runs under Windows, and was compiled using the MinGW environment. The following components are included:

GCC 4.2.0 for AVR processors

- GCC C compiler front-end version 4.2.0
- C++ compiler front-end version 4.2.0
- AVR-elf target code generators version 4.2.0
- Bin-utils for ARM-elf target version 2.17.50
- AVR-libc runtime libraries version 1.4.6

GCC 4.2.0 for BlackFin processors

- GCC C compiler front-end version 4.2.0
- C++ compiler front-end version 4.2.0
- BFin-elf target code generators version 4.2.0
- Bin-utils for ARM-elf target version 2.17.50
- Newlib runtime libraries version 1.15.0
- elf2ldr utility to convert ELF output files from the linker to LDR files, downloadable to the processor

GCC 4.0.1 for ARM processors

- GCC C compiler front-end version 4.0.1
- C++ compiler front-end version 4.0.1
- Objective C compiler front-end version 4.0.1
- Java compiler front-end version 4.0.1
- Fortran 95 compiler front-end version 4.0.1
- ADA compiler front-end version 4.0.1
- ARM-elf target code generators version 4.0.1
- Bin-utils for ARM-elf target version 2.16.1
- Newlib runtime libraries version 1.13.0

GCC 4.0 for ARM processors

- GCC C compiler front-end version 4.0
- C++ compiler front-end version 4.0
- Objective C compiler front-end version 4.0

- Java compiler front-end version 4.0
- Fortran 95 compiler front-end version 4.0
- ADA compiler front-end version 4.0
- ARM-elf target code generators version 4.0
- Bin-utils for ARM-elf target version 2.15.94
- Newlib runtime libraries version 1.13.0

GCC 3.4.0 for ARM processors

- GCC C compiler front-end version 3.4.0
- C++ compiler front-end version 3.4.0
- Objective C compiler front-end version 3.4.0
- Java compiler front-end version 3.4.0
- Fortran compiler front-end version 3.4.0
- ADA compiler front-end version 3.4.0
- ARM-elf target code generators version 3.4.0
- Bin-utils for ARM-elf target version 2.15.90
- Newlib runtime libraries from their CVS repository as of 05/12/04

GCC 3.3.1 for ARM processors

- GCC C compiler front-end version 3.3.1
- C++ compiler front-end version 3.3.1
- Objective C compiler front-end version 3.3.1
- Java compiler front-end version 3.3.1
- Fortran compiler front-end version 3.3.1
- ADA compiler front-end version 3.3.1
- Pascal compiler front-end version 3.3.1
- ARM-elf target code generators version 3.3.1
- Bin-utils for ARM-elf target version 2.14.90
- Newlib runtime libraries from their CVS repository as of 10/21/03

A version of GDB and Insight is also available for download. It's pre-compiled for the ARM-elf target and was compiled under the Cygwin. I've tried to pack all the required support-DLLs in the package but if you have problems of missing DLLs, you can get them by installing the Cygwin enviroment from here. The debugger can be used with the USB programmer and CPLD-based programmer JTAG ICEs to debug embedded targets but contains a simulator too so you can test your code without the actual HW.

I've separated the Java compiler and the libraries out from the main compiler set to reduce the file size, so you can save yourself 53MBytes worth of download time if you are not planning on using the Java language.

Download

AVR GCC 4.2.0

- GCC 4.2.0 for AVR-elf targets (27MBytes) DOWNLOAD

BlackFin GCC 4.2.0

- GCC 4.2.0 for BFin-elf targets (74MBytes) DOWNLOAD
- elf2ldr with sources (54kBytes) DOWNLOAD

ARM GCC 4.0.1

- GCC 4.0.1 for ARM-elf targets (86MBytes) DOWNLOAD
- Java libraries 4.0.1 for ARM-elf targets (76MBytes) DOWNLOAD

ARM GCC 4.0

- GCC 4.0 for ARM-elf targets (41MBytes) DOWNLOAD
- Java libraries 4.0 for ARM-elf targets (76MBytes) DOWNLOAD

ARM GCC 3.4.0

- GCC 3.4.0 for ARM-elf targets (38MBytes) DOWNLOAD
- GJC Java compiler and libraries 3.4.0 for ARM-elf targets (58MBytes) DOWNLOAD

ARM GCC 3.3.1

- GCC 3.3.1 for ARM-elf targets (43MBytes) DOWNLOAD
- GJC Java compiler and libraries 3.3.1 for ARM-elf targets (53MBytes) DOWNLOAD

ARM GDB and Insight 6.0

- GDB/Insight 6.0 for ARM-elf targets (13MBytes) DOWNLOAD

Install notes

First of all, I had no time to test all to tools, so use them at your own risk. If you find a problem, you are welcome to report it to me.

I've used the following configure options to compile the tools:

AVR GCC 4.2.0

```
--prefix=c:/gcc-avr-4.2.0  --target=avr  --enable-threads  --disable-n
ls  --disable-win32-registry  --disable-shared  --enable-
sjlj  --enable-languages=c,c++
```

BlackFin GCC 4.2.0

```
--prefix=c:/gcc-bfin-elf-4.2.0 --target=bfin-elf --enable-threads --
-disable-nls --disable-win32-registry --disable-shared --enable-
sjlj --enable-languages=c,c++ --with-newlib
```

ARM GCC 4.0.1

```
--prefix=c:/gcc-arm-elf-4.0.1 --target=arm-elf --enable-threads --d
isable-nls --disable-win32-registry --disable-shared --enable-sjlj-
exceptions --with-newlib --enable-languages=c,c++,java,objc,f95,ada
```

ARM GCC 4.0

```
--prefix=c:/gcc-arm-elf-4.0.0 --target=arm-elf --enable-threads --d
isable-nls --disable-win32-registry --disable-shared --enable-sjlj-
exceptions --with-newlib
```

ARM GCC 3.4.0

```
--prefix=c:/gcc-arm-elf-3.4.0 --target=arm-elf --enable-threads --d
isable-nls --disable-win32-registry --disable-shared --enable-sjlj-
exceptions
```

ARM GCC 3.3.1

```
--prefix=c:/gcc-arm-elf-3.3.1 --target=arm-elf --enable-threads --d
isable-nls --disable-win32-registry --disable-shared --enable-sjlj-
exceptions
```

From this you see that the install directory should be **'c:/gcc-avr-4.2.0**
', **'c:/gcc-bfin-elf-4.2.0'**, **'c:/gcc-arm-elf-4.0.1'**, **'c:/gcc-arm-elf-4.0.0'**, **'c:/gcc-arm-elf-3.3.1'** or **'c:/gcc-arm-elf-3.4.0'** respectively. Though my experience is that the compiler isn't that sensitive for the install path, it appeared to me that the ADA compiler actually requires the files to be in that location. All in all, you can try put the tools into whatever location you whish but if you experience problems, that's one thing to look at.

Anything newer than the 3.3.1 verision is too new to have GPC support so the Pascal front-end is missing from those toolsets. Also note that I previously had some problems generating the ADA runtime library and ADA tools so those are missing from the distribution as well, except for the latest ARM compiler version.

To install the toolset, do the following:

- Download the files, you need
- Create the above mentioned directory(s)
- Uncompress the content of the file(s) to the newly created directory
- Add \bin to your path

How to use

Here's a simple example to use the ARM toolset. We want to compile a simple application, from two files STARTUP.S and PROGRAM.C:

```
arm-elf-as STARTUP.S -o STARTUP.O  arm-elf-gcc -c PROGRAM.C -o PROGRAM
.O  arm-elf-ld -Ttext 0 -e 0 -Map APPLICATION.MAP -o APPLICATION.ELF S
TARTUP.O PROGRAM.O  arm-elf-
objcopy -O binary APPLICATION.ELF APPLICATION.BIN
```

Here we created two object files, one from the asm, one from the C source, linked them together (with 0-based executable segment) and extracted the binary image from the output of the linker. This last file, APPLICATION.BIN is the binary image, ready to be downloaded to the targets' memory.