Trampoline on targets

Release 1.0

Florent PAVIN

April 1, 2010

CONTENTS

CHAPTER

ONE

Getting started on an example

This document describes how to get started a Trampoline [1] application on any target.
The following steps will be overviewed :

- Trampoline installation
- Target Compiler needed
- Installation of drivers and softwares to be able to upload programs on your board.

This document is explained from a UNIX side. If you are on a Windows system, download and install Cygwin as it is explained on NXTOSEK website (http://lejos-osek.sourceforge.net/installation_windows.htm) first.

I.1 Trampoline

To download Trampoline, type in a terminal (login:anonymous - password:anonymous) :

```
$svn checkout https://trampoline.rts-software.org/svn/trunk
```

Download the libpm (http://galgas.rts-software.org/download/) for your achitecture and copy it in trunk/. To compile GOIL, go in goil/makefile_[ARCH] depending on your architecture and type in a terminal (you can add -j2 if you have 2 processors) :

```
$make goil
```

This will generate the GOIL executable you'll need. You can add the path to your GOIL (in ∼/.profile), adding :

[1] Trampoline is an open source RTOS which, once certified, could be compliant with the OSEK/VDX specification. Currently it is not the case, so while Trampoline has the same API as OSEK/VDX, it is not officially compliant. Trampoline is available under the GNU Lesser General Public License V2 and **is maintained by gcc-4.0.1**. Trampoline runs on several platforms like POSIX, C166 with Keil compiler, Star12X (courtesy of Geensys), PowerPC, ARM (NXT2 for example)

```
export PATH=[TRAMPOLINE_REPOSITORY]/goil/makefile_[ARCH]/:$PATH
```

You can also do it typing ”$sudo make install goil” (this will copy the goil executable in /usr/local/bin).

You can now use Trampoline on Unix system by executing the tests as described in Annex II.1.

I.2 ARM

For ARM target, you can use GNUARM as compiler. Thus, the first section will show you how to compile gcc for ARM achitecture (you can find some binaries on GNUARM website if you prefer http://www.gnuarm.com/).
The second section depends on your board. It's the drivers installation to communicate to your board and upload a program.

I.2.1 GNUARM

Unix

To compile gcc for ARM architecure on UNIX system, follow all the steps below, after having download :

- binutils-2.19.1
- gcc-4.4.1
- newlib-1.17.0
- gdb-6.8

You can download other versions of these softwares, don't forget to change the header below.

```
#!/bin/sh
set -x

DIR=`dirname $0` && cd $DIR &&
#---
BINUTILS_VERSION=binutils-2.19.1 &&
GCC_VERSION=gcc-4.4.1 &&
NEWLIB_VERSION=newlib-1.17.0 &&
GDB_VERSION=gdb-6.8 &&
#INSTALL_DIR=$DIR/TEMP &&
INSTALL_DIR=/usr/local/gcc-for-arm/gcc-4-4-1 &&
USE_SUDO="sudo" &&
#---
rm -f *.tar &&

#--- Binutils
echo "----------- CROSS COMPILING BINUTILS"
rm -fr build-binutils $BINUTILS_VERSION &&
bunzip2 -kc $BINUTILS_VERSION.tar.bz2 > $BINUTILS_VERSION.tar &&
tar xf $BINUTILS_VERSION.tar &&
rm -f *.tar &&
mkdir build-binutils &&
cd build-binutils &&
../$BINUTILS_VERSION/configure --help &&
../$BINUTILS_VERSION/configure --prefix=$INSTALL_DIR --target=arm-elf --disable-werror "CC=
    gcc-4.0 -m32" &&
```

```
time make all &&
$USE_SUDO make install &&
cd .. &&
rm -fr build-binutils $BINUTILS_VERSION &&

#--- GCC
echo "------------ CROSS COMPILING GCC"
rm -fr build-gcc $GCC_VERSION $NEWLIB_VERSION &&
bunzip2 -kc $GCC_VERSION.tar.bz2 > $GCC_VERSION.tar &&
tar xf $GCC_VERSION.tar &&
bunzip2 -kc $NEWLIB_VERSION.tar.bz2 > $NEWLIB_VERSION.tar &&
tar xf $NEWLIB_VERSION.tar &&
rm -f *.tar &&
mv $NEWLIB_VERSION $GCC_VERSION/ &&
mkdir build-gcc &&
cd build-gcc &&
../$GCC_VERSION/configure --help &&
../$GCC_VERSION/configure --prefix=$INSTALL_DIR --target=arm-elf --with-newlib --enable-
    languages=c --disable-libssp --disable-werror "CC=gcc-4.0 -m32" &&
time make all &&
$USE_SUDO make install &&
cd .. &&
rm -fr build-gcc $GCC_VERSION &&

#--- GDB
echo "------------ CROSS COMPILING GDB"
rm -fr build-gdb $GDB_VERSION &&
bunzip2 -kc $GDB_VERSION.tar.bz2 > $GDB_VERSION.tar &&
tar xf $GDB_VERSION.tar &&
rm -f *.tar &&
mkdir build-gdb &&
cd build-gdb &&
../$GDB_VERSION/configure --help &&
../$GDB_VERSION/configure --prefix=$INSTALL_DIR --target=arm-elf --disable-werror "CC=gcc-4.0
    -m32" &&
time make all &&
$USE_SUDO make install &&
cd .. &&
rm -fr build-gdb $GDB_VERSION &&
#---
echo "------------ Success!"
```

Windows

- Download a GCC-4.0.2 tool chain (http://www.gnuarm.com/bu-2.16.1_gcc-4.0.2-c-c++_nl-1.14.0_-gi-6.4.exe).
- Install only selected components in the below picture. ARM7(ATMEL AT91SAM7S256) in the NXT and in the Olimex board is Little Endian and does not have FPU.
- Do not check "Install Cygwin DLLs..." because Cygwin was already installed.

- At the end of the installation, you were asked about adding the tool path to Windows Environment Variables, but it would not be needed.

I.2.2 Board installation drivers and upload program

I.2.2.1 Lego Mindstorm NXT2.0

I.2.2.1.1 Nexttool + Lego Drivers

MAC OS

Download and install the Lego Drivers (http://mindstorms.lego.com/en-us/support/files/default.aspx #Driver) and the firmware update (http://mindstorms.lego.com/en-us/support/files/default.aspx#Firmware) for MAC OS.
Download Nexttool (http://bricxcc.sourceforge.net/utilities.html) and a new firmware (http://bricxcc.sourceforge.net/lms_arm_jch.zip) and update the firmware as explained below :

- Reset the NXT : To go into firmware update mode, press the reset button (at the back of the NXT, upper left corner beneath the USB connector) for more than 5 seconds while the NXT is turned on. The NXT will audibly tick when it is in firmware update mode.
- Copy an Enhanced NXT firmware (i.e. lms_arm_nbcnxc_107.rfw) to NeXTTool extracted directory.
- Launch Nexttool, and updload the Enhanced NXT firmware to the NXT (clicking on "Download firmware"), selecting it.

- Remove the battery from the NXT and insert it again, and then press orange rectangle button on the NXT to turn on the Enhanced NXT firmware. The Enhanced NXT firmware has same GUI as the LEGO standard firmware.

Linux

Note: Nextool binary for Linux seems to fail for firmware upload.
Install required packages:

```
$sudo apt-get install scons libusb-dev libusb-0.1-4
```

Download the libnxt (http://libnxt.googlecode.com/files/libnxt-0.3.tar.gz) archive and extract it. Go in the new directory and build the project with scons:

```
$ cd libnxt-0.3/
$ scons
```

A program call fwflash is created.
Download John Hansen's Enhanced NXT firmware (http://bricxcc.sourceforge.net/lms_arm_jch.zip) (any version numbered 106 or later includes the native-invocation feature) and store the Enhanced NXT firmware (i.e lms_arm_nbcnxc_1xx.rfw) in the directory where fwflash is stored.
Connect the NXT brick to usb and turn it on. Then press the reset button for more than 4s to put it in firmware upload mode (nxt display is cleared but it makes a ticking sound).
Flash the firmware with the following command (it takes some dozen of seconds), where 1xx is replaced by the number of the firmware:

```
$ sudo ./fwflash lms_arm_nbcnxc_1xx.rfw
```

Troubleshooting: After completion of the upload, sometimes NXT display is messed and block : reboot it by a quick push on the reset button or remove the battery. If the NXT makes a ticking sound, it is still in firmware upload mode. If troubles, use and see windows firmware update procedure (and LEGO UserGuide).

Windows

Download and install the Lego Drivers (http://mindstorms.lego.com/en-us/support/files/default.aspx#Driver) for PC.
Download Nexttool (http://bricxcc.sourceforge.net/nexttool.zip) and a new firmware (http://bricxcc.sourceforge.net/lms_arm_jch.zip) and update the firm-ware as explained below :

- Reset the NXT : To go into firmware update mode, press the reset button (at the back of the NXT, upper left corner beneath the USB connector) for more than 5 seconds while the NXT is turned on. The NXT will audibly tick when it is in firmware update mode.
- Copy an Enhanced NXT firmware (i.e. lms_arm_nbcnxc_107.rfw) to NeXTTool extracted directory.
- Execute Cygwin and type the following command to change the current directory to the NexTTool extracted directory. (NeXTTool is assumed to be extracted under C:\cygwin\nexttool directory)

```
$cd C:\cygwin\nexttool
```

- Connect PC and the NXT by USB cable.
- Type the following command in Cygwin to upload the Enhanced NXT firmware to the NXT (Program upload may take around half minutes and then, NXT LCD is turned to display some chunk from blank).

```
$./NeXTTool.exe /COM=usb -firmware=lms\_arm\_nbcnxc\_107.rfw
```

- Remove the battery from the NXT and insert it again, and then press orange rectangle button on the NXT to turn on the Enhanced NXT firmware. The Enhanced NXT firmware has same GUI as the LEGO standard firmware.

I.2.2.1.2 Upload a program

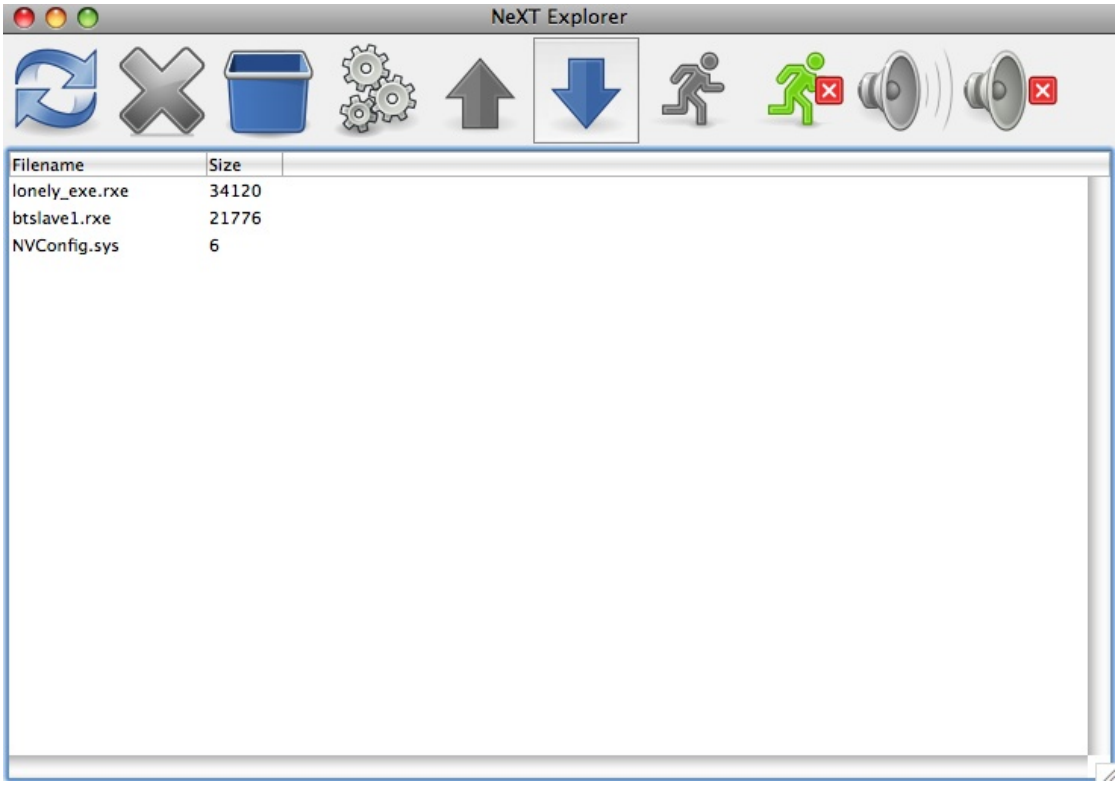
MAC OS

To upload a program in the NXT (the nxt example examples/arm/nxt/simple/nxt_simple_exe.rxe)

- Connect the PC and the NXT by USB cable.
- Launch Nexttool, select ”usb port”.

- Go to ”NXT Explorer”
- Click on the ”Download selected files to the NXT” and select the nxt_simple_exe.rxe file.
- If program upload was succeeded, you can see the nxt_simple_exe.rxe file in the files list as below.

[h]



- To execute a program on the NXT, go in ”My files”/”Software files”.

Linux

Required Packages: libusb-0.1-4
Download this executable of John Hansen’s NeXTTool (http://bricxcc.sourceforge.net/nexttool.zip) (built

from bricxcc svn repository, revision 1)
Check the version of NeXTTool, it should be 1.0.1.0:

```
$ sudo ./[NEXTTOOL_PATH]/NeXTTool
nexttool version 1.0.1.0 (1.0.1.0)
Copyright (c) 2006, John Hansen
Use "NeXTTool -help" for more information.
```

To ubload over usb, turn on the NXT, connect it to USB and run the following command (example: nxt_simple_exe.rxe) :

```
$ sudo ./[NEXTTOOL_PATH]/NeXTTool /COM=usb -download=nxt\_simple\_exe.rxe
```

Troubleshooting: Test the following command to see if Nexttool is working (set execution right for NeXTTool):

```
$ sudo ./[NEXTTOOL_PATH]/NeXTTool /COM=usb -versions
Protocol version = 1.124
Firmware version = 1.xx
```

To ubload over bluetooth, you need to define an alias name in a file 'nxt.dat' as explained in this post: Minsdtorm 2.0 development on linux . Then turn on the NXT and run the following command (example: nxt_simple_exe.rxe) :

```
$ sudo ./[NEXTTOOL_PATH]/NeXTTool /COM=alias_bt -download=nxt\_simple\_exe.rxe
```

To execute a program on the NXT, go in "My files"/"Software files".

Windows

To upload a program in the NXT (the nxt example examples/arm/nxt/lonely_exe.rxe) follow the steps below :

- Connect the PC and the NXT by USB cable.
- Type the following command in Cygwin (from examples/arm/nxt) :

```
$./[NEXTTOOL_PATH]/NeXTTool.exe /COM=usb -download=lonely_exe.rxe
$./[NEXTTOOL_PATH]/NeXTTool.exe /COM=usb -listfiles=lonely_exe.rxe
```

- If program upload was succeeded, program size could be displayed in Cygwin such as the second line in the below command outputs.

```
Executing NeXTTool to upload helloworld.rxe...
helloworld.rxe=15280
NeXTTool is terminated.
```

- To execute a program on the NXT, go in ”My files”/”Software files”.

I.2.2.2 Olimex LPC-E2294

I.2.2.2.1 OPENOCD + drivers

openOCD is a software that communicates via the JTAG (Open On-Chip Debug Solution for Embedded Target Systems based on the ARM7 and ARM9 Family). It’s an open source software by BerliOS (http://openocd.berlios.de/web/).
Download and compile openOCD for your achitecture (it appears revision after 1200 are not able to compile correcty under MacOS X).
As you don’t need just openOCD because it needs the USB driver, you’ve got two choice :

- install libusb et libftdi
- install D2XX

I.2.2.2.2 Upload a program

To upload a program double click on *2-run-openocd.command* to open the connection between your computer and the board.
And double click on *2a-debug-external-ram.command* if you want to upload your program in external ram with debug.
In the debug consol type :

- ”c” to continue
- ”b” to place a breakpoint
- ”si” to step instruction
- ”display/i $pc” to display assembler code on the pc register
- ”x/i $pc” to visualise the i blocs from pc
- ”info registers” to see the registers state

CHAPTER

TWO

Appendix

II.1 Launching Trampoline tests

To launch the tests you have to compile ViPER [1] first. Go in viper/ and type in a terminal :

```
$make
```

To launch the tests, go in check/ and type in a terminal :

```
$./tests.sh
```

At the end of the tests you should see :

```
...
Compare results with the expected ones...
Functional tests Succeed!!
GOIL tests Succeed!!
```

If an error occurs, you can visit Trampoline's forum (http://trampoline.rts-software.org/bb/).

II.2 Cross-Compile an application

To cross-compile a Trampoline application for ARM for example, you need to set :

```
COMPILER = "arm-elf-gcc";
ASSEMBLER = "arm-elf-as";
LINKER = "arm-elf-ld";
```

in your oil file as you can see in examples/arm/nxt/nxt_simple.oil.

[1] Virtual Processor Emulator, ViPER is used on Posix system to send interrupts to Trampoline to emulate the timers. It is launched by Trampoline.

For the Lego Mindstorm NXT2.0 only, you also need to add the path to your libgcc and libc as below (X.X.X is your cross-gcc version) :

```
LDFLAGS = "-L[GNUARM_PATH]/lib/gcc/arm-elf/X.X.X -lgcc";
LDFLAGS = "-L[GNUARM_PATH]/arm-elf/lib -lc";
```

And then, compile your application typing in a terminal (from the example in examples/arm/nxt/simple) :

```
$goil -t=arm/nxt --templates=../../../../goil/templates -g -i nxt_simple.oil
```

This will generate the Makefile needed, thus type :

```
$make
```

Then you need to upload the nxt_simple_exe.rxe file after installing drivers and softwares on your platform.