

# Core Toolchain Setup

for MIO32 application development on Linux

## Preparation

This document assumes that you have a relatively modern x86 based linux distribution. While it may work other distributions, it has only been tested on x86 Ubuntu 10.10. Many of the commands, especially “aptitude” are only available on Debian derived distributions like Ubuntu.

## Linux Utilities

### subversion

You will need to install subversion to download the svn repository

- `sudo aptitude install subversion`

### wget

Wget is a useful tool for downloading files from the Internet and can be downloaded with:

- `sudo aptitude install wget`

## Required Stuff

### Download the MIO32 repository

Create a directory to download the toolchain to ( `~/midibox` is my personal favourite). and download the repository:

- `mkdir ~/midibox`
- `cd ~/midibox`
- `svn checkout svnmios.midibox.org/mios32`

(If that doesn't work try:)

```
* svn checkout svn://svnmios.midibox.org/mios32/
```

## Linux build utilities

Terminal commands for installing basic build tools like “make”:

- \* **Debian/Ubuntu/Mint:** *sudo aptitude install build-essential*
- \* **Arch Linux/Manjaro:** *sudo pacman -S base-devel*

## GNU Multiple Precision Library

Terminal Commands:

- \* **Arch Linux/Manjaro:** *sudo pacman -S gmp*
- \* **Debian/Ubuntu/Mint:** *sudo aptitude install libgmp3-dev* (or *lib32gmp3-dev* for AMD64Version)

**Ubuntu, Mint troubleshoot:** if it still doesnt build(make) it might be that you use a 64bit Computer, but the toolchain is for 32bit, so you will need additonal following commands:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libstdc++6:i386 zlib1g:i386
sudo apt-get install libc6:i386
```

## MIOS32 Toolchain

The MIDibox community have created a modified GNU Compiler Collection, ready for use with the ARM Cortex M3 platform. The toolchain contains all of the tools required to build MIOS32 applications including GCC and NEWLIB.

The Toolchain has currently been compiled for Windows (2000+), Mac OS X (Leopard or newer) and Linux x86 (built on Ubuntu 10.10).

The download directory for the toolchain is [http://www.midibox.org/mios32\\_toolchain](http://www.midibox.org/mios32_toolchain)

Download the latest “linux” file and extract into your favorite directory, If you extract into /usr/local the toolchain will create /usr/local/mios32\_toolchain and various sub-directories containing the toolchain files.

- cd /usr/local
- sudo wget [http://www.midibox.org/mios32\\_toolchain/gcc-arm-none-eabi-xxxxxx.tar.bz2](http://www.midibox.org/mios32_toolchain/gcc-arm-none-eabi-xxxxxx.tar.bz2)
- sudo tar xzvf gcc-arm-none-eabi-xxxxxx.tar.bz2

replace xxxxxx with the date of the latest linux toolchain in  
[http://www.midibox.org/mios32\\_toolchain](http://www.midibox.org/mios32_toolchain)

You now need to set the path variable for the toolchain bin directory (/usr/local/mios32\_toolchain/bin) this is covered in the next section

# Environment Variables

If you are using the default bash shell then setting up path variables in the shell is quite easy, in a terminal window just type:

- `export VARIABLE_NAME=something`

and if you want to check to what the variable has been assigned, type:

- `echo $VARIABLE_NAME`

this will print what's inside the variable. For example, type:

- `export MIOS32_PATH=~/.svn/mios32/trunk`

to set up the mios32 folder path, then type

- `echo $MIOS32_PATH`

## Adding to the path

To temporarily add the mios32\_toolchain bin directory to the path, type:

- `export PATH=$PATH:/usr/local/mios32_toolchain/bin`

to check that the path variable has been assigned type:

- `echo $PATH`

You should see the toolchain directory at the end of the path

## Setup MIOS32 Environment

The following commands will setup the build environment for MIOS32, you must change MIOS32\_PATH to be wherever you installed the svn repository:

- `export MIOS32_PATH=~/.midibox/mios32/trunk`
- `export MIOS32_BIN_PATH=$MIOS32_PATH/bin`
- `export MIOS32_GCC_PREFIX=arm-none-eabi`
- `export MIOS32_FAMILY=STM32F10x`
- `export MIOS32_PROCESSOR=STM32F103RE`
- `export MIOS32_BOARD=MBHP_CORE_STM32`
- `export MIOS32_LCD=universal`

You can add these lines to your ~/.profile file to ensure that they are always set, the editor pico (or nano) is a fairly simple text editor but you may have your favourite.

### **If you know what you are Doing in Linux-you can do this systemwide:**

open your filebrowser with sudo rights, for example type in terminal:

\* **Linux Mint:** *sudo nemo*

\* **Linux Arch:** *sudo thunar*

the filebrowser now opens with root rights, navigate to the folder “etc” open the file “environment”

»> **/etc/environment**

Depending on the path where your toolchain is situated, the Example Content off this file is:

- PATH="/home/autark/midibox/gcc-arm-none-eabi/bin"
- MIOS32\_PATH=/home/autark/midibox/mios32
- MIOS32\_BIN\_PATH=/home/autark/midibox/mios32/bin
- MIOS32\_BOARD=MBHP\_CORE\_STM32F4
- MIOS32\_FAMILY=STM32F4xx
- MIOS32\_GCC\_PREFIX=arm-none-eabi
- MIOS32\_PROCESSOR=STM32F407VG
- MIOS32\_LCD=universal

if you cant log after reboot, then mostly you messed up your “PATH” variable, in this case, boot with Linuxbootstick, and remove the PATH - line...

### **you can also add more folders to the PATH, they are separeted with a “:”, for example:**

\* PATH="/home/autark/midibox/gcc-arm-none-eabi/bin:/home/autark/program"

here we also have a folder named “program”, when you put in there “Mios Studio”, then you can write in Terminal “Mios Studio” and mios studio will open

## **Building your first application**

Once you have completed the above steps, you should be able to build your first application.

- cd \$MIOS32\_PATH/apps/tutorials/001\_forwarding\_midi
- make

if this works great! If not, either you made a mistake (or I did!!)

From:

<http://www.midibox.org/dokuwiki/> - MIDIBOX

Permanent link:

[http://www.midibox.org/dokuwiki/doku.php?id=linux\\_mios32\\_toolchain\\_core&rev=1716488944](http://www.midibox.org/dokuwiki/doku.php?id=linux_mios32_toolchain_core&rev=1716488944)

Last update: **2024/05/23 18:29**

