

NewAE

LoonBoard Unified Bootloader Manual

## 1.0 Getting the Files

The files can be downloaded from the Sourceforge project at <http://sourceforge.net/projects/lubloader/> where you will find three types of download. The first is the source code for the project, the second is the Windows executable, and the final one is the AVR hex files.

See the installation section to understand which files you need.

## 2.0 Installing

### 2.1 Linux

For Linux you will need to compile your own binary. Luckily this is very easy. First, download the source package. Extract it to a location of your choosing, as long as you remember where you extracted it to.

Then:

```
cd /where/you/have/sources
cd PC
make clean
make
```

You can run lubloader from that location by just typing

```
./lubloader
```

However you should either copy or link the binary to somewhere in the path. The directory /usr/bin would be a good choice. If you wish to copy the program into the path run

```
su
[enter root password]
cp lubloader /usr/bin/lubloader
exit
```

Test it works by running from any directory

```
lubloader
```

If you will be changing the executable a lot, such as for development work, you may wish to just make a symbolic link to the file. If you link the file it will still be in your path, but now when it is executed the file in the source directory is executed. This means two things: (1) you cannot delete the sources directory, and

(2) it is possible to replace the lubloader file by something simply named lubloader. This second point is a security risk as lubloader is supposed to be a trusted command, but now could be anything. On a personal box, where you won't be running lubloader from root anyway, the risk may be worth it. To do this:

```
su
[enter root password]
ln -s /where/you/have/sources/PC/lubloader
/usr/bin/lubloader
exit
```

## 2.2 Windows

The Windows binary is prebuilt, although you could build your own if you want. To use it you simply have to either copy it to somewhere in the PATH, or put it in its own directory and setup that directory in the PATH.

It is generally safer and easier to make your directory to install the binary into. To do this copy the lubloader.exe to somewhere such as C:\newae\lubloader and you may also need to copy the cygwin1.dll as in Figure 1. Then follow either section 2.2.0 or 2.2.1 for your windows versions.

### 2.2.0 Windows XP

Go to Start – Control Panel. Find the “System” icon – if you don't have one then try clicking a link that says “Switch to Classic View” on the left-hand side.



Figure 1: Copy to a known directory.

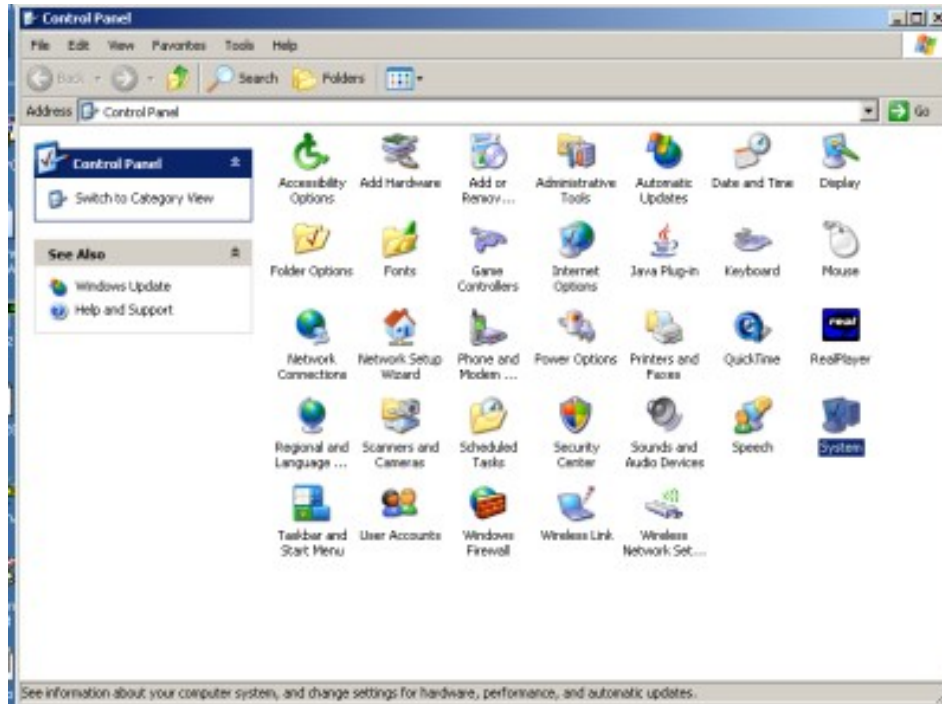


Figure 2: Control Panel

Go to the “Advanced” tab and click the “Environmental Variables” button. You now have the option of either setting the “System Variable” Path or making a local one. If you set the “System Variables” add on to the existing Path variable. Say you copied the .exe to C:\newae\lubloader you would add “;C:\newae\lubloader” onto the end of the Path statement.

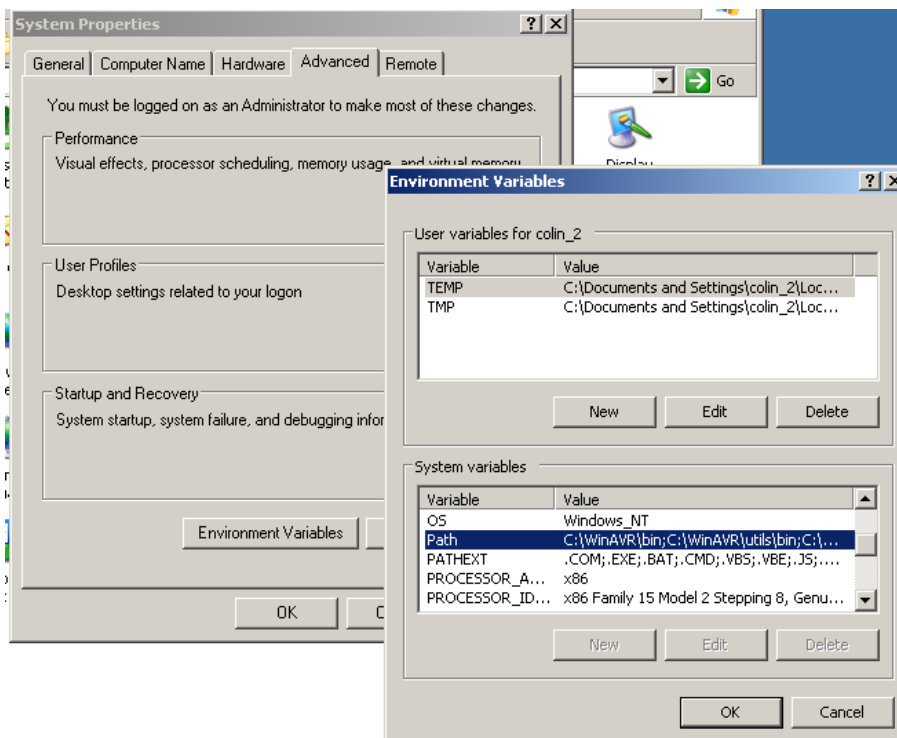


Figure 3: Adding to the PATH

If you don't have access to the system variables because you aren't an administrator, then you can make a new user variable. Click the “New” button and fill in the name as “PATH” and the value as the directory the .exe is stored in.

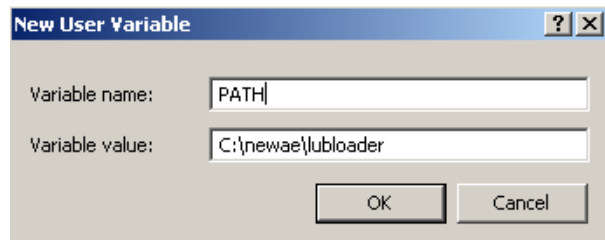


Figure 4: Create a new local PATH variable

Hit enough “OK” buttons that the Windows disappear now, and let's give it a test. Go to Start-Run and type “cmd” then hit OK. You should see a Command window open. Type “lubloader” and hit enter, the program should print a quick message indicating it ran OK.

## 2.2.1 Windows 98

Go to Start-Run and type “msconfig” and hit OK.

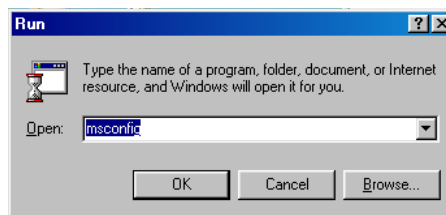


Figure 5: msconfig

Then go to the “Autoexec.bat” tab. Hit the New button and make the line read “SET PATH=C:\newae\lubloader;%PATH%” (with no “” marks) and hit OK. You will need to restart your computer.

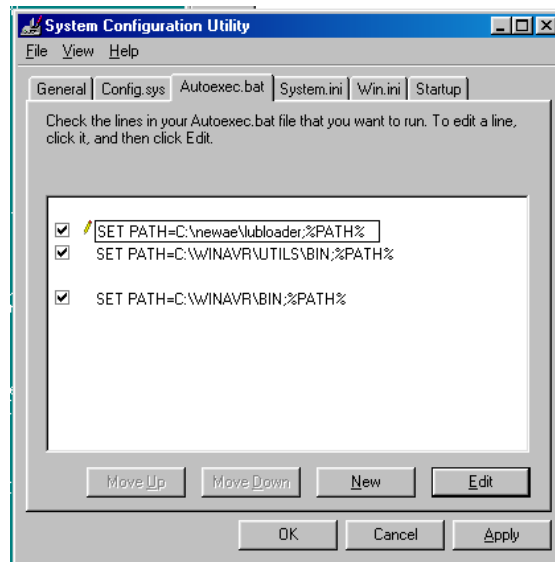


Figure 6: msconfig - edit autoexec.bat

Now after it starts up go back to Start-Run and enter command.com and hit OK. The command line should pop up – again just enter the command “lubloader” and hit enter. The banner message should print indicating the program ran.

### 3.0 Running

Running the lub is easy! Open a terminal in your operating system.

In Windows do this by going Start-->Run and entering either “command.com” for Win95/98/ME or “cmd” for 2000/XP.

In Linux this is accomplished in a variety of manners depending on your distribution... normally it is either a right click on the desktop to bring up the option or in a menu somewhere.

Now – hook up your target hardware. If you are using the lb-cable see the manual for that device, if using your own hardware then hook that up.

To test the connection first figure out what port your LoonBoard is on. If you are using lb-cable this is likely “com3” on Windows and “/dev/ttyUSB0” on Linux. However it will depend on your system completely. Then run the command for Linux:

```
lubloader -P /dev/ttyUSB0 -v
```

or for Windows:

```
lubloader -P com3 -v
```

Note that for the rest of the guide there will be only Linux examples. To use them in Windows it is only the com part that changes: com3 instead of /dev/ttyUSB0.

There is two main operations you will need to do – download a new AVR file , and download a new FPGA file. To download a new AVR file compile it into an Intel HEX format. Then type

```
lubloader -a main.hex -P /dev/ttyUSB0
```

And you should see it download and verify. To send a new FPGA file to the board create the .bit file in Xilinx ISE. This is normally created by running the “Generate Programming File” process. Then run the command:

```
lubloader -f fpgafile.bit -P /dev/ttyUSB0
```

And you can program both files at once to save a small amount of time:

```
Lubloader -f fpgafile.bit -a main.hex -P /dev/ttyUSB0
```

## 4.0 Upgrading the Target's Bootloader

As NewAE releases newer versions of the bootloader, or you want to try your own modifications, you will need to update the loader stored on the target. Remember there is always two versions of the bootloader stored on the target: the normal and safemode. To update the normal bootloader run the command

```
lubloader -q main.hex -P /dev/ttyUSB0
```

Then if that works, you can update the safemode bootloader using:

```
lubloader -w main.hex -P /dev/ttyUSB0
```

If the normal bootloader didn't work, you can get an old version running using the safemode bootloader. Force the safemode bootloader by running:

```
lubloader -q main.hex -P /dev/ttyUSB0 -s
```

And now you should have a working system again. The lub system's power shows through here, as you rarely have to worry about a bad upgrade rendering your system unusable.

## 5.0 Making Changes

Since the entire system is open source, you are free to make changes to it. You are also free to use in on a commercial product, provided you stay within the GPL. It's not that hard to do, and if you really don't want to contact NewAE for alternate licensing.

Compiling the AVR target is easy and has already been explained. If you compile for targets other than the Mega168 or Mega88 you will have to change the tinyloader.S file at least. You will find a section saying the page size in bytes, and you will need to add more possibilities if your target is not 128 or 64 byte page sizes. As well in the Makefile you will find a define saying where the bootloader space starts in **byte** addresses.

Compiling the PC target is fairly easy, but you will have to change some settings. In the PC/Makefile file, you will see the following lines of interest:

```
#Select either POSIX or Windows 32 interface
#SER=ser_win32.o
SER=ser_posix.o

#Uncomment this if building for Windows
#CFLAGS = -DWIN32NATIVE

#Uncomment this if you don't have the gettimeofday
function
#CFLAGS += -DGETTIMEOFDAY
```

This works for most Linux systems. To build on Windows you will have to change that section to:

```
#Select either POSIX or Windows 32 interface
SER=ser_win32.o
#SER=ser_posix.o

#Uncomment this if building for Windows
CFLAGS = -DWIN32NATIVE

#Uncomment this if you don't have the gettimeofday
function
CFLAGS += -DGETTIMEOFDAY
```

This should work for the gcc as provided in Cygwin or MinGW. You may or may not need to have the -DGETTIMEOFDAY, if you get errors about it being redefined then comment that out.

If you make useful changes, why not submit them? Or even if you just have ideas or find bugs, you can report all this through the Sourceforge.net interface.

## ***Document Revision History***

December 6, 2005:

- Initial document release

**Disclaimer:** NewAE assumes no liability whatsoever and disclaims any express, implied, or statutory warranty related to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or non-infringement. In no event shall NewAE be liable for any direct, indirect, consequential, punitive, special or incidental damages (including, without limitation, damages for loss of profits, business, interruption, or loss of information) arising out of the use or inability to use this document, even if NewAE has been advised of the possibility of such damages. NewAE makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to the specifications and product descriptions at any time without notice. NewAE does not make any commitment to update the information contained herein. NewAE's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© NewAE 2005. All rights reserved.