

# Manual installation using RPM packages

---

## Contents

[\[hide\]](#)

### 1 Manual installation on your host OS with RPMs

- 1.1 Step 1: Install Java JDK
- 1.2 Step 2: Install Cygwin
- 1.3 Step 3: Install native compilers
- 1.4 Step 4: Install TinyOS toolchain
- 1.5 Step 5: Install the TinyOS 2.x source tree
- 1.6 Step 5b (optional): Post-install fixes in Windows/Cygwin environment
- 1.7 Step 6: Installing Graphviz

## Manual installation on your host OS with RPMs

---

Currently, the TinyOS Core Working Group supports TinyOS on two platforms: Cygwin (Windows) and Linux. There have been some successful efforts to getting TinyOS environments working on Mac OSX, but OSX is not supported by the Core WG.

Installing a TinyOS environment has five basic steps; Windows requires an extra step, installing Cygwin, which is a UNIX-like environment. The steps are:

1. **Installing a Java JDK.** Java is the most common way of interacting with mote base stations or gateways that are plugged into a PC or laptop.
2. **Windows only. Install Cygwin.** This gives you a shell and many UNIX tools which the TinyOS environment uses, such as perl and shell scripts.
3. **Installing native compilers.** As you're compiling code for low-power microcontrollers, you need compilers that can generate the proper assembly code. If you using mica-family motes, you need the AVR toolchain; if you're using telos-family motes, you need the MSP430 toolchain.
4. **Installing the nesC compiler.** TinyOS is written in nesC, a dialect of C with support for the TinyOS concurrency model and component-based programming. The nesC compiler is platform-independent: it passes its output to the native compilers, so that it can take advantage of all of the effort put into their optimizations.
5. **Installing the TinyOS source tree.** If you want to compile and install TinyOS programs, you need the code.
6. **Installing the Graphviz visualization tool.** The TinyOS environment includes `nesdoc`, a tool that automatically generates HTML documentation from source code. Part of this process involves drawing diagrams that show the relationships between different TinyOS components. [Graphviz](#) is an open source tool that nesdoc uses to draw the diagrams.

### Step 1: Install Java JDK

---

Download and install Sun's JDK from <http://java.sun.com>.

### Step 2: Install Cygwin

---

This step is required for Windows installations only. If you are installing on Linux, skip to step 3.

1. Download [setup.exe](#) and run it to install Cygwin.
2. When the window pops up to allow you to select what packages to install, be sure to select:
  - rpm
  - make
  - perl
  - python

## Step 3: Install native compilers

Install the appropriate compiler based on your hardware and operating system.

### Atmel AVR Tools

Tool	Windows/Cygwin	Linux	Command
avr-binutils†	<a href="#">avr-binutils-2.17tinyos-3.cygwin.i386.rpm</a>	<a href="#">avr-binutils-2.17tinyos-3.i386.rpm</a>	
avr-gcc	<a href="#">avr-gcc-4.1.2-1.cygwin.i386.rpm</a>	<a href="#">avr-gcc-4.1.2-1.i386.rpm</a>	
avr-libc	<a href="#">avr-libc-1.4.7-1.cygwin.i386.rpm</a>	<a href="#">avr-libc-1.4.7-1.i386.rpm</a>	<code>rpm -ivh</code>
avarice	<a href="#">avarice-2.4-1.cygwin.i386.rpm</a>	<a href="#">avarice-2.4-1.i386.rpm</a>	<code>rpm -ivh --ignoreos</code> (if Cygwin complains)
insight (avr-gdb)	<a href="#">avr-insight-6.3-1.cygwin.i386.rpm</a>	<a href="#">avr-insight-6.3-1.i386.rpm</a>	
avrdude††	<a href="#">avrdude-tinyos-5.6cvs-1.cygwin.i386.rpm</a>	<a href="#">avrdude-tinyos-5.6cvs-1.i386.rpm</a>	

†If you receive an rpm error that indicates that you have a newer version already installed, try `rpm -Uvh --force`

††If you receive an rpm error that indicates that you are missing /bin/sh, try `rpm -Uvh --force --nodeps`

### PXA27x Tools (iMote2)†

Tool	Windows/Cygwin	Linux	Command
xscale-elf-binutils	<a href="#">xscale-elf-binutils-2.15tinyos-1.cygwin.i386.rpm</a>	<a href="#">xscale-elf-binutils-2.15-2.i386.deb</a>	
xscale-elf-gcc	<a href="#">xscale-elf-gcc-3.4.3-1.cygwin.i386.rpm</a>	<a href="#">xscale-elf-gcc-3.4.3-2.i386.deb</a>	<code>rpm -ivh</code> <code>rpm -ivh --ignoreos</code> (if Cygwin complains)
xscale-elf-newlib	<a href="#">xscale-elf-newlib-1.11.0tinyos-1.cygwin.i386.rpm</a>	<a href="#">xscale-elf-newlib-1.13.0-1.i386.deb</a>	
jtag	<a href="#">jflashmm.tgz</a> ††	<a href="#">OpenOCD toolchain</a>	

† Newer versions of GCC can be installed in certain cases. For further info see section [iMote2 Compiler](#) from this wiki.

†† Copy `jflashmm.tgz` to `/opt` directory, untar it (`tar xzf jflashmm.tgz`), and make sure it's on your path (see Step 5). If the download link above doesn't work, try the [cvs](#).

## TI MSP430 Tools

...wow this is outdated. Don't use any of this. Use the debian packages or install from source. See the packaging folder in tinyos-main.

Tool	Windows/Cygwin	Linux	Command
base	<a href="#">msp430tools-base-0.1-20050607.cygwin.i386.rpm</a>	<a href="#">msp430tools-base-0.1-20050607.i386.rpm</a>	
python tools	<a href="#">msp430tools-python-tools-1.0-1.cygwin.noarch.rpm</a>	<a href="#">msp430tools-python-tools-1.0-1.noarch.rpm</a>	
binutils	<a href="#">msp430tools-binutils-2.16-20050607.cygwin.i386.rpm</a>	<a href="#">msp430tools-binutils-2.16-20050607.i386.rpm</a>	<code>rpm -ivh</code>
gcc	<a href="#">msp430tools-gcc-3.2.3-20050607.cygwin.i386.rpm</a>	<a href="#">msp430tools-gcc-3.2.3-20050607.i386.rpm</a>	<code>rpm -ivh --ignoreos</code> (if Cygwin complains)
libc	<a href="#">msp430tools-libc-20080808-1.cygwin.i386.rpm</a>	<a href="#">msp430tools-libc-20080808-1.i386.rpm</a>	
jtag	[1]Not yet available	<a href="#">msp430tools-jtag-lib-20031101cvs-20050610.i386.rpm</a>	
gdb	[2]Not yet available	<a href="#">msp430tools-gdb-6.0-20050609.i386.rpm</a>	

## Renesas M16C (m32c-elf-gcc)

[http://www.eistec.se/docs/wiki/index.php?title=Mulle\\_software\\_with\\_GCC](http://www.eistec.se/docs/wiki/index.php?title=Mulle_software_with_GCC)

## Step 4: Install TinyOS toolchain

The TinyOS-specific tools are the NesC compiler and a set of tools developed in the `tinyos-2.x/tools` source code repository. They are also installed using `rpms`.

There are two Linux versions of `tinyos-tools`, depending on whether you have a 32-bit or 64-bit machine. The first is the `i386` RPM and the second is the `i686` RPM. If you have a 64-bit Java VM, it is important that you install the `i686` RPM or otherwise the Java support may not work properly.

### TinyOS-specific Tools

Tool	Windows/Cygwin	Linux	Command
NesC	<a href="#">nesc-1.3.1-1.cygwin.i386.rpm</a>	<a href="#">nesc-1.3.1-1.fc9.i386.rpm</a>	<code>rpm -Uvh</code> <code>rpm -Uvh --ignoreos</code> (if Cygwin complains)
Deputy	<a href="#">tinyos-deputy-1.1-1.cygwin.i386.rpm</a>	<a href="#">tinyos-deputy-1.1-1.fc9.i386.rpm</a>	<code>rpm -i</code> <code>rpm -i --ignoreos</code> (if Cygwin complains)
tinyos-tools	<a href="#">tinyos-tools-1.4.0-3.cygwin.i386.rpm</a>	<a href="#">tinyos-tools-1.4.0-3.ubuntu.i386.rpm</a> <a href="#">tinyos-tools-1.4.2-20120807.x86_64.rpm</a>	<code>rpm -Uvh</code>

## Step 5: Install the TinyOS 2.x source tree

---

Now that the tools are installed, you need only install the tinys 2.x source tree and then set your environment variables.

1. Get the code from the TinyOS release repository:

```
2. wget http://github.com/tinyos/tinyos-release/archive/tinyos-2_1_2.tar.gz
3. tar xf tinys-2_1_2.tar.gz
```

This will extract the actual TinyOS code in a folder named `tinys-release-tinys-2_1_2` inside the directory the command was issued. Feel free to rename this folder to `tinys-main`.

4. You will need to add some environment variables to your shell. The following file includes the necessary ones. Substitute the placeholder `<local-tinys-path>` with the path where you chose to place the code in the previous section (full path recommended).

```
5. # Here we setup the environment
6. # variables needed by the tinys
7. # make system
8.
9. export TOSROOT="<local-tinys-path>"
10. export TOSDIR="$TOSROOT/tos"
11. export CLASSPATH=$CLASSPATH:$TOSROOT/support/sdk/java
12. export MAKERULES="$TOSROOT/support/make/Makerules"
13. export PYTHONPATH=$PYTHONPATH:$TOSROOT/support/sdk/python
14.
15. echo "setting up TinyOS on source path $TOSROOT"
```

Suppose you named this file `tinys.env`. There are now at least two possibilities to have these variables accessible in your shell:

1. Place it as root user in `/etc/profile.d/`
2. Place it in `<local-tinys-path>` and add the following line to your `.bashrc`

```
source <local-tinys-path>/tinys.env
```

You can also add the contents of the file directly to the `.bashrc`.

To make the changes work, the easiest way is to open a new terminal window.

16. In order to program motes you will need to access the serial ports. This can be done by joining the group that grants this privilege. `sudo gpasswd -a <your-user> dialout` This change only takes effect when log out and log in again.

## Step 5b (optional): Post-install fixes in Windows/Cygwin environment

---

If you later experience problems when building some tutorials, running Java tools, ... see [Geoffrey Lo's excellent blog post](#), especially step 2.

## Step 6: Installing Graphviz

---

Go to [download page](#) of the Graphviz project and download the appropriate RPM. You only need the basic graphviz RPM (graphviz-); you don't need all of the add-ons, such as -devel, -doc, -perl, etc. If you're not sure what version of Linux you're running,

```
uname -a
```

might give you some useful information. Install the rpm with `rpm -i rpm-name`. In the case of Windows, there is a simple install program, so you don't need to deal with RPMs.

you may need to download graphviz 1.1.0 [here](#)