I've created a shell script to easily install the native version of TinyOS on Ubuntu. Most of it is for installing dependencies that are not included by default, and then automatically following the install instructions found here (Links to an external site.)Links to an external site., and finally, tests if the Blink project can be successfully compiled.

The script installs git, make, automake, emacs, gperf, bison, flex, default-jdk, python2.7-dev, python-minimal, and g++ for a total of about a 203MB download decompressed to 809MB disk space.

Occasionally the installs will fail. If this happens, just run the script again.

I have tested it on a VM of Ubuntu (I used Vmware, but any should do) and the Bash on Ubuntu on Windows (Windows Subsystem for Linux), and it should also work on non-virtualized versions such as those found in the lab. Instructions for Vmware can be found here (Links to an external site.)Links to an external site., and Bash on Ubuntu on Windows here (Links to an external site.)Links to an external site.(note, you should run sudo apt-get update for this one because the Ubuntu version is an older one from the windows store).
Here is the script: to run it, follow these instructions (Links to an external site.)Links to an external site. Note: if you create the file on a windows machine, the EOL character will be crlf while linux wants lf, so be sure to use an editor which allows you to choose the EOL character. It will create two new folders "nesC" and "tiny" in the directory it is run from. When it is finished, you should see "*** Successfully built micaz TOSSIM library.". I will be happy to accept any suggested improvements and will answer questions posted here when I can.

```
#!/bin/bash
mkdir tiny
mkdir nesC
cd nesC

sudo dpkg --configure -a

sudo apt-get install git make automake emacs gperf bison flex openjdk-8-jdk-headless
default-jdk python2.7-dev python-minimal g++-4.8 gcc-4.8

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 10
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.8 10

git clone git://github.com/tinyos/nesc.git

cd nesc
./Bootstrap
./configure

sudo make
sudo make install
#nesC is now installed

cd ..
cd ..
cd tiny
```

```
git clone git://github.com/tinyos/tinyos-main.git
cd tinyos-main/tools
./Bootstrap
./configure
sudo make
sudo make install
#tinyos is now installed

cd ..

export TINYOS_ROOT_DIR="$PWD"

cd apps
cd Blink
sudo make micaz sim && exit #if this succeeds, no further action is required

cd ..
cd ..
cd tools
./Bootstrap
./configure
sudo make
sudo make install

cd ..
cd ..
cd apps
cd Blink
sudo make micaz sim && exit #if this succeeds, no further action is required
```

And run export TINYOS_ROOT_DIR=theLocationOfTiny/tinyos-main

for example, if the tiny folder is in /usr/Desktop, run

```
export TINYOS_ROOT_DIR=/usr/Desktop/tiny/tinyos-main
```

Unfortunately, The export command is needed to be run each time you launch terminal, same as the manual install.