Quick Start Guide

This document describes how to establish a development environment for Telos on a Windows PC using the Telos Tools CD. It describes how to install the Telos development tools, how to install the USB driver to access Telos as a virtual COM port, how to compile and install a TinyOS application, and how to run basic Java tools such as SerialForwarder to read and write data to and from the wireless network.

This document is not a comprehensive guide for developing for Telos; instead the basic steps for getting started with TinyOS with Telos are presented in this guide.
Windows Quick-Start
This section guides you through the process of installing the Telos tools and COM drivers, how to compile and install applications, and how to run Java tools like SerialForwarder.

Tools Install
A complete install includes Cygwin, Java, the MSPGCC MSP430 compiler, TinyOS sources, and other supporting programs.

Run tinyos-msp430-setup.exe or setup.exe from the Telos Tools CD.

The installer may spend some time unpacking data in an “InstallShield Wizard” window.

Once the main InstallShield Wizard window opens, if this is a clean system, select “Complete” install, which installs Cygwin, Java, the MSPGCC MSP430 compiler, TinyOS sources, and other programs. If you already have some of these programs installed, do a “Custom” install for only the new things you need.

The installer then goes through the formalities such as asking you to read the Java licenses before it installs those packages and verifying your install options before it continues.
After completing the initial install phase, the installer continues on to install Cygwin and its related packages.

Cygwin continues the installation in a “Cygwin Setup” window.

Then, Cygwin performs some additional installation operations in a series of “Cygwin Setup Post-Install Script” windows.

Everything is done installing when this “Information” window appears.

You are now ready to start programming for Telos!
COM Driver Install

This section describes the process to install COM drivers for Telos. The first time a Telos module is plugged into a system, Windows opens the “Found New Hardware Wizard”.

Select “Install from a list or specific location (Advanced)” then “Next”.

When prompted, select only “Include this location in the search” then click “Browse”.

Select the “FTDI Driver” folder from the Telos Tools CD, click “OK”, then “Next”. You will be informed that the driver is not a Windows Certified Driver. Click “Continue Anyway”.

Windows installs the driver files, completing the installation for Telos as a “USB Serial Port”.

Because the XP drivers are uncertified, this process occurs for each new Telos plugged into the system. Windows reserves a unique serial port per Telos. Since the drivers are stored in the cache, this manual process only takes a few moments for each new Telos plugged into the system.

**Application Compile and Reprogram**

Cygwin is used to compile and reprogram applications for Telos.

To open a shell, invoke the Cygwin shortcut that the installer placed on your desktop.

This an initial a Cygwin bash shell prompt. The install shield installs the TinyOS sources into /opt/tinyos-1.x/. Now we’ll compile CntToLedsAndRfm, a program that increments a counter, displaying it to the Telos LED’s while sending a message of the count over the radio.

Compiling CntToLedsAndRfm is as easy as changing to its directory and running “make telos”.

```
cd /opt/tinyos-1.x/apps/CntToLedsAndRfm
make telos
```
Use the motelist utility to show which COM ports actively have Telos modules. Just type "motelist" at the command prompt:

```
motelist
```

It may take a few moments for a Telos module to appear in the list the first time it is plugged into a system, because Windows must automatically install the drivers and allocate a COM port.

To install the application onto the Telos module, enter the following command:

```
make telos reinstall,1 bsl,3
```

“reinstall,1” means “program the module with the already compiled binary image, set the network address of the module to 1”. “bsl,3” means “send the program using the Boot Strap Loader on COM4”. Note that the bsl programmer uses Linux-style indexing for the COM ports, so COM4 is specified as “3” for bsl.

You’ve now successfully compiled and installed an application onto Telos!

**Running Java Tools**

In addition to reprogramming, USB is used to receive data from Telos into a PC. When a module is plugged into a PC, it can essentially act as a “Telos module adapter”, bridging the PC and the Telos networks. SerialForwarder is a Java tool that listens for TinyOS packets on a serial port and forwards them over a local TCP network socket. This allows more than one application to send and receive packets to the attached module. First plug in a Telos module to your PC’s USB port.

Run SerialForwarder by entering the following command on a single line:

```
java net.tinyos.sf.SerialForwarder -comm serial@COM4:telos
```
The full Java package specification for SerialForwarder is “net.tinyos.sf.SerialForwarder”. “-comm serial@COM4:telos” tells SerialForwarder to communicate using the serial protocol over COM4 with an attached a Telos module.

This is the Java application window created by SerialForwarder. It allows you to stop and start forwarding packets, specify a new communication device, and displays useful statistics.

Bringing It All Together

You now have the knowledge and experience to use all these tools together. To gather data from the Telos network, connect a second Telos module to a USB port. Change to the TOSBase directory in TinyOS (in the directory /opt/tinyos-1.x/apps/TOSBase). Compile with “make telos” and install a TOSBase application on the second Telos module (“make telos reinstall”). The TOSBase application acts as a gateway, relaying Telos packets to the PC and vice versa. Keep your first Telos module running CntToLedsAndRfm, attach the Telos module running TOSBase to a USB port, and run SerialForwarder on that port. Now run “java net.tinyos.tools.Listen”, configure SerialForwarder for the virtual COM port assigned to the TOSBase Telos, and see the raw data from the first mote arrive through the TOSBase mote and onto your PC. If you don’t know the port Telos is using, run motelist from any Cygwin command window. The “Packets Read” counter should now be increasing each time the first Telos module increments its count on the LEDs. You have successfully set up a wireless network with Telos and can move on to the TinyOS tutorial in /opt/tinyos-1.x/doc/tutorial/ or start developing your own applications.

If you need help using the Telos module, contact Moteiv Support at support@moteiv.com. You can also find help for programming in TinyOS on the TinyOS Help mailing list at tinyos-help@millennium.berkeley.edu.
Telos/TinyOS Manual Install for Microsoft Windows

This document describes how to download and install the software and tools necessary to develop for Telos using TinyOS. This document is also available on the web at http://moteiv.com/support/install-windows.html.

Last revised on August 12, 2004.

Cygwin

Cygwin [1] is a Linux-like environment in Windows. It provides the basic development environment for TinyOS and Telos. Cygwin's setup program [2] downloads and installs necessary packages. It doesn't hurt to install every package, it just takes a little more space and time to download.


Initial Environment

This install guide uses certain environment variables which are listed here with their defaults. Double check their values before using them in this guide. They must be defined using unix-style paths and not as windows paths like "c:/" or "c:".

```
export JDKROOT=/cygdrive/c/j2sdk1.4.2_05
export MSPGCCROOT=/cygdrive/c/mspgcc
export TOSROOT=$HOME/tinyos-1.x
```

Java

Java [1] is used for the tools that interface between a mote and your PC. The TinyOS tools have been tested on the Java 2 Platform, Standard Edition v 1.4.2 (J2SE) [2].

After installing the JDK, ensure that JDKROOT is set for your installation, such as:

```
export JDKROOT=/cygdrive/c/j2sdk1.4.2_05
```


JavaComm

JavaComm [1] is an additional package for Java that provides access to serial ports, required for tools like SerialForwarder. JavaComm for Windows [2] is available directly from Sun; JavaComm for Linux [3] requires more effort. Both require some command-line effort to correctly install, feel free to read the instructions, though these commands work nicely to get JavaComm installed for Windows:

```bash
```
unzip javacomm20-win32.zip
cp commapi/win32com.dll $JDKROOT/jre/bin/
cp commapi/comm.jar $JDKROOT/jre/lib/ext/
cp commapi/javax.comm.properties $JDKROOT/jre/lib/


FTDI VirtualComm USB Drivers

FTDI [1] provides Windows drivers [2] to communicate with their USB-to-RS232 chips as a standard COM port; Linux drivers have been available as part of the kernel since 2.4.0. Use the drivers with enhanced BM series support [3], currently dated March 2004.


TinyOS CVS Repository

The TinyOS [1] CVS repository is hosted by the TinyOS SourceForge project [2]. To get the latest, bleeding edge version of the code, enter the following commands at a shell prompt; just press Enter when prompted for a password:

    cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/tinyos login
    cvs -z3 -d:pserver:anonymous@cvs.sf.net:/cvsroot/tinyos co tinyos-1.x

After downloading the TinyOS 1.x source code, ensure that TOSROOT is set for your installation, such as:

    export TOSROOT=$HOME/tinyos-1.x


MSPGCC

MSPGCC [1] is the compiler used by TinyOS for the TI MSP430 platform. Packages [2] for both Windows and Linux are provided.

In Windows, download the mspgcc-win32 package [3]; during the install shield dialog "Choose Compoments", uncheck the "Make utilities & libs" (top-most) and "GivelIO" (bottom-most) options.

In Linux, the Cross Development Kit [4] for MSPGCC is a popular pre-compiled set of RPMs for Linux systems.
After installing MSPGCC, ensure that MSPGCCROOT is set for your installation, such as:

```bash
export MSPGCCROOT=/cygpath/c/mspgcc
```


**Patched MSP430 Assembler**

nesC requires a slightly modified assembler to recognize `$` as a valid identifier symbol. GNU binutils [1] provides the assembler, which you must download [2], patch, compile, and install over the default MSPGCC assembler binaries, using the following commands:

```bash
tar xfvz binutils-2.15.tar.gz
cd binutils-2.15/gas/
perl -i.orig -pe 's/define (LEX_DOLLAR) 0/undef $1/\' gas/config/tc-msp430.h
./configure --target=msp430
make
cp gas/as-new.exe $MSPGCCROOT/bin/msp430-as.exe
cp gas/as-new.exe $MSPGCCROOT/msp430/bin/as.exe
```

As an alternative to manually compiling a patched msp430-as.exe, Moteiv provides a precompiled version [3] that you may download and install:

```bash
wget http://moteiv.com/support/msp430-as.zip
unzip msp430-as.zip
cp msp430-as/msp430-as.exe $MSPGCCROOT/bin/msp430-as.exe
cp msp430-as/msp430-as.exe $MSPGCCROOT/msp430/bin/as.exe
```


**MSP430-BSL for Telos**

msp430-bsl is the programmer, or "Boot Strap Loader", for the MSP430 developed by the MSPGCC project. Telos requires a special version distributed with TinyOS to support its unique hardware design. Go to $TOSROOT/tools/src/mspgcc-pybsl/ and follow the instructions detailed in README.msp430-bsl-telos.

As an alternative to manually compiling msp430-bsl.exe, Moteiv provides a precompiled version [1] of msp430-bsl that you may download and install:

```bash
wget http://moteiv.com/support/msp430-bsl.zip
unzip -o -d $MSPGCCROOT/bin msp430-bsl.zip msp430-bsl.exe python23.dll \ lib/`
```

MoteList

Motelist is a command-line utility that displays which serial ports have Telos motes attached. Run it with "motelist"; try "motelist -h" for additional options. To compile and install, enter the following command:

```
cd $TOSROOT/tools/src/motelist/ && make && make install
```

As an alternative to manually compiling motelist.exe, Moteiv provides a precompiled version [1] that you may download and install:

```
wget http://moteiv.com/support/motelist.zip
unzip -o -d /usr/local/bin motelist.zip motelist.exe
```


TinyOS Scripts

Some TinyOS scripts, specifically set-mote-id, are necessary to the build process. Install these scripts by entering the following command:

```
cd $TOSROOT/tools/scripts/ && make install prefix=/usr/local
```

Environment

Now is a good time to fully prepare your environment for Telos and TinyOS. Add these commands like these to your ~/.bashrc script. Note that the xxxROOT variables must match your installation and must be defined using unix-style paths.

```
# /usr/local/bin
export PATH="/usr/local/bin:$PATH"

# Java
export JDKROOT=/cygdrive/c/j2sdk1.4.2_05
export PATH="$JDKROOT/bin:$PATH"
# MSPGCC
export MSPGCCROOT=/cygdrive/c/mspgcc
export PATH="$MSPGCCROOT/bin:$PATH"

# TinyOS
export TOSROOT=$HOME/tinyos-1.x
export TOSDIR=$TOSROOT/tos
export CLASSPATH="`$TOSROOT/tools/java/javapath`"
export MAKERULES=$TOSROOT/tools/make/Makerules
```

Before continuing to the final steps of this guide, ensure that these commands have been executed, possibly by logging back into your bash shell. Compiling in TinyOS requires TOSDIR
be set, using the TinyOS Java tools requires that CLASSPATH be properly set, and building for Telos requires that MAKERULES be set.

**nesC**

**nesC** [1] is the programming language used for TinyOS, and it requires its own front-end compiler. Check out the source code for nesC version 1.1.2 by entering the following commands, just press Enter when prompted for a password:

```
cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/nescc login
cvs -z3 -d:pserver:anonymous@cvs.sf.net:/cvsroot/nescc co -r v1_1_2 nesc
cd nesc
./Bootstrap
```

nesC by default compiles a new assembler for the AVR platform, which is not necessary for the Telos/MSP430 platform. If you want to avoid this stage, cut-and-paste the following command into the shell prompt:

```
perl -i.orig -pe 's{\S+avr-as[^s"]+}{\}g if /^\s*ac_config_f/;
$_="" if /avr-as/;' Makefile.in configure{,.in} tools/Make*
```

Then, to compile and install nesC, enter the follow command at the shell prompt:

```
./configure && make && make install
```


**TinyOS Java Tools**

The TinyOS Java tools require the MIG utility provided by nesC to generate certain source files for processing messages. Compile the Java sources with the following commands:

```
cd $TOSROOT/tools/java; make; make
```
General Information

Document History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2004/06/02</td>
<td>Initial Release</td>
</tr>
<tr>
<td>1.01</td>
<td>2004/06/09</td>
<td>Fixes for TinyOS paths and inclusion of CDK4MSP for Linux Added Java Tools</td>
</tr>
<tr>
<td>1.02</td>
<td>2004/08/14</td>
<td>Overhauled the Manual Install for Microsoft Windows section.</td>
</tr>
</tbody>
</table>

Disclaimer

Moteiv Corporation believes the information contained herein is correct and accurate at the time of this printing. However, Moteiv Corporation reserves the right to make changes to this product without notice. Moteiv Corporation does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. This product is not designed for use in life support devices or any other system where malfunction can reasonably be expected to result in significant personal injury to the user. This product is not designed for critical systems where failure of the product to perform affects safety or effectiveness. Moteiv Corporation customers using or selling products for use in such applications do so at their own risk and agree to fully indemnify Moteiv Corporation for any damages resulting from improper use or sale.

As far as possible, major changes of product specifications and functionality, will be stated in product specific errata notes published at the Moteiv website. The latest updates are available from the Moteiv website at www.moteiv.com or by contacting Moteiv directly.

© 2004 Moteiv Corporation