Ultra low power IEEE 802.15.4 compliant wireless sensor module
Humidity, Light, and Temperature sensors with USB

Quick Start Guide

This document describes how to establish a development environment for Tmote on a Windows PC using the Tmote Tools CD. It describes how to install the Tmote development tools, how to install the USB driver to access Tmote as a virtual COM port, how to compile and install Moteiv's wireless mesh networking application, and how to visualize and control wireless sensor networks.

This document is not a comprehensive guide for developing for Tmote; instead the basic steps for getting started with TinyOS and Tmote are presented in this guide. Other useful documents for development include the Tmote Datasheet and the Tmote Application Notes found at http://www.moteiv.com
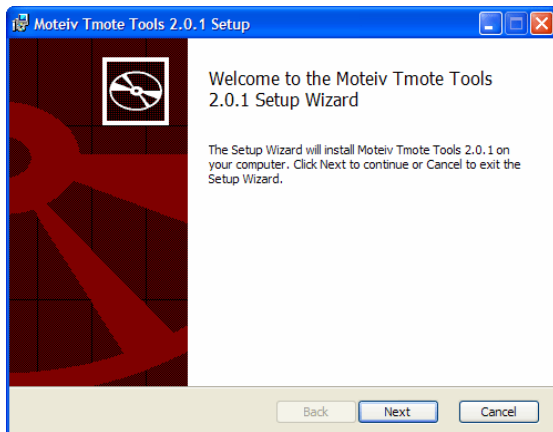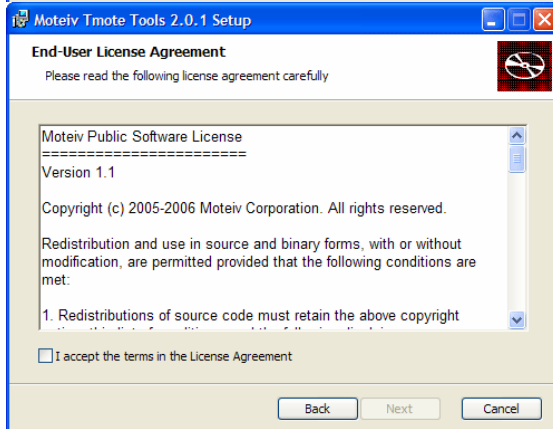
Contents

Tmote Tools Installation

> ⚠ **NOTE:** If you have previously installed TinyOS or Cygwin, either from a previous Moteiv installer, from another vendor's installer, or on your own accord, we *recommend* you *remove* all previous TinyOS and Cygwin installations before proceeding. See page **Error! Bookmark not defined.** for more information.
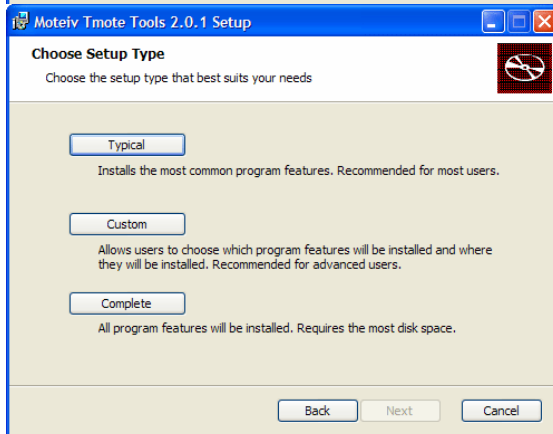
Please place the Tmote Tools CD into your computer's CD-ROM drive. After a few moments, the Moteiv Tmote Tools Setup Wizard will display:
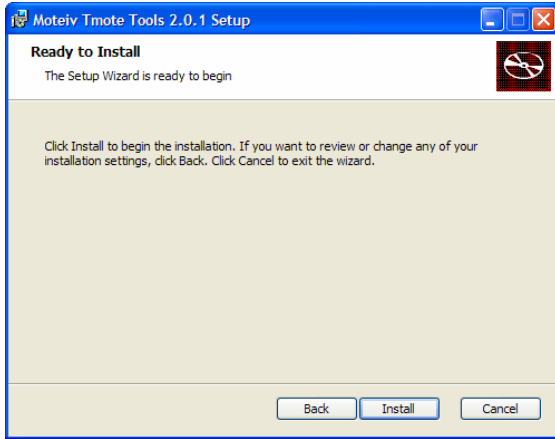
Please click "Next".

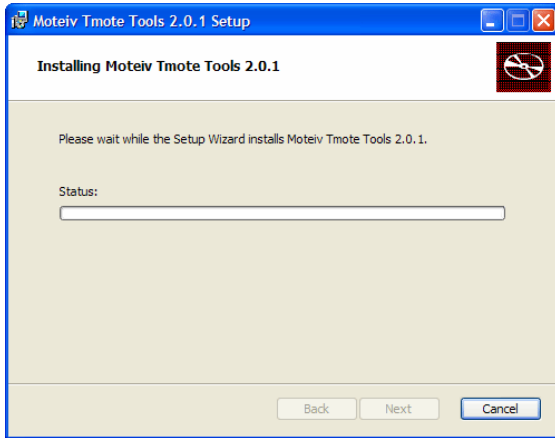Moteiv's software is distributed under the Moteiv Public Software License. Accept the terms of the license agreement and click "Next".

Please select the "Typical" installation option.

.

Simply click "Install".

The Moteiv Tmote Tools installer will begin installation and configuration of your system.

First, the Cygwin Setup window will appear and will take a few minutes to install all of the necessary packages.

After Cygwin Setup is complete, the Java platform is installed. Accept the Sun Microsystems license agreement and click "Next".

Choose additional Java installation options if desired, and then click "Next".

The Java installer will proceed to install and configure Java.

Upon completion of the Java installation, click "Finish".

After Java is complete, the Moteiv Tmote Tools Setup Wizard will install and configure the tools required to develop and run applications using the TinyOS open source operating system. Please be patient as all of the files are installed and server tools are compiled.

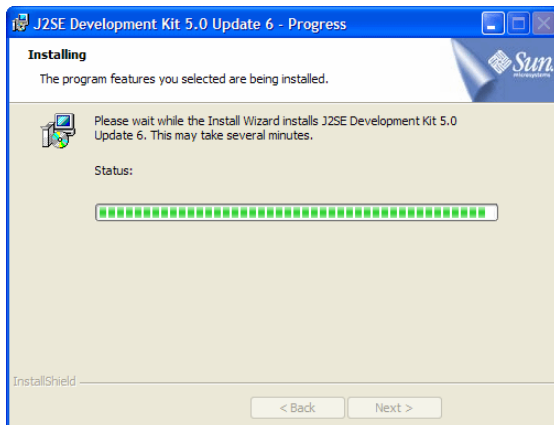Following installation of the TinyOS tools, the Moteiv Tmote Tools Setup Wizard installs and configures Boomerang by Moteiv. This is Moteiv's distribution of the TinyOS operating system that includes extended API support, example applications, and additional development tools.

Congratualations! Installation of Moteiv's Tmote Tools is complete! Please proceed to the next section to start building applications with Moteiv's Tmote devices.

USB Serial COM Driver Install

This section describes the process to install drivers for Tmote if they are not already installed. The first time a Tmote module is plugged into a system, if the drivers are not found, Windows opens the "Found New Hardware Wizard".

Note: If you experience any problems with your drivers, we recommend you uninstall all other "USB Serial" device drivers from your system.

The first step is to connect a Tmote module to your computer's USB port.

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 "USB Serial Driver" folder from the Tmote Tools CD, click "OK", then "Next".

Windows installs the driver files, completing the installation for Tmote as a "USB Serial Port".

After installation, a second New Hardware window may appear for a "USB Serial Port". At this point it is okay to select "Install the software automatically".

Application Compile and Reprogram

Cygwin is used to compile and reprogram applications for Tmote.

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

This an initial a Cygwin bash shell prompt. The installer configures Moteiv's open source system at /opt/moteiv/. Let's compile Delta, Moteiv's wireless sensing and mesh networking application.

Compiling Delta is as easy as changing to its directory and running "make tmote".

cd /opt/moteiv/apps/Delta
make tmote

Use the motelist utility to show which COM ports actively have Tmote modules. Just type "motelist" at the command prompt:

motelist

It may take a few moments for a Tmote 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 Tmote module if only one Tmote is connected, enter the following command:

make tmote reinstall,1

"reinstall,1" means "program the module with the already compiled binary image, set the network address of the module to 1".

To install the application onto a Tmote module if more than one Tmote is connected, you must specify the serial port to install onto using the following command:

make tmote reinstall,1 bsl,3

"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 Tmote!

Install Delta onto a few of your Tmote Sky modules. Make sure that each node has a different network address (in other words, change the number after "reinstall" to give each node a unique address).

Running Trawler

After installing Delta on your modules, we can display the mesh network using Moteiv's Trawler application.

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

MOTECOM=serial@COM4:tmote java com.moteiv.trawler.Trawler

The MOTECOM variable, "serial@COM4:tmote", tells the Java tools to communicate using the serial protocol over COM4 with an attached a Tmote module.

When Trawler starts, it will begin the process of establishing an ad-hoc mesh network and display the network topology on the screen.

Be patient, it may take several minutes for the entire network to come online and establish stable, reliable routes to the Tmote Invent device connected to your PC.

Trawler includes a number of features to assist with standard data collection applications. By clicking on the "Sensor readings" tab, Trawler displays the temperature values coming from the nodes in the network.

Zoom in to the data by pressing the "Zoom In" and "Zoom Out" buttons. You can scroll up, down, left, and right using the arrow buttons on the bottom right of the display. You can also use your mouse to select a region on the graph—Trawler will zoom in to display the data selected.

Now, click on the "links" tab. This display assists with network commissioning—you can determine if the link quality between nodes is extremely low, you may need to move the node or add additional nodes to the network that participate in the mesh.

You can log the data readings to a file by clicking on "Log Packets" in the "Vizualization Controls" sidebar. If you cannot find the "Vizualization Controls", minimize Trawler. Sometimes the controls are hidden by the main Trawler window.

API documentation on the Delta application is provided in the /opt/moteiv/doc/nesdoc directory. Simply open the index.html file and select TinyOS components to learn more about them and how to use them in your own applications.

You have successfully set up a wireless network with Tmote 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 Tmote module, contact Moteiv Support at support@moteiv.com or view the support documentation at http://www.moteiv.com/support.php. You can also find help for programming in TinyOS on the TinyOS Help mailing list at tinyos-help@millennium.berkeley.edu.

Moteiv TinyOS Tools Manual Install

If you have previously installed TinyOS or Cygwin, either from a previous Moteiv installer, from another vendor's installer, or on your own accord, we **recommend** you **remove** all previous TinyOS and Cygwin installations before proceeding.

Uninstalling Cygwin

Cygwin provides no method for convenient uninstall. The following steps are usually sufficient to remove Cygwin from your system:
- Close **all** Cygwin applications and services
- Delete or rename the following keys in the registry by invoking RegEdit or RegEdt32
  - HKEY_CURRENT_USER\Software\Cygnus Solutions
  - HKEY_LOCAL_MACHINE\SOFTWARE\Cygnus Solutions
- Delete or rename your Cygwin install directory, which defaults to c:\cygwin

Uninstalling TinyOS

If you installed TinyOS with an installation utility, just run its associated uninstaller.

If not, you must manually uninstall any "tinyos", "nesc", and "msp430" RPM's. Discover which RPM's are installed by starting a Cygwin shell and running the command "`rpm -qa`". Remove packages with the command "`rpm --erase --nodeps [package1] [package2] […]`". Here is a sample list of RPM's that may be installed, though the particular packages and versions installed on your computer may differ:

tinyos-tools-1.2.1-3
msp430tools-binutils-2.16-20050607
make-3.80tinyos-1
msp430tools-python-tools-1.0-1
tinyos-javacomm-1.0.0-1
tinyos-moteiv-2.0.1-1
nesc-1.2.4-1
msp430tools-base-0.1-20050607
msp430tools-gcc-3.2.3-20050607
msp430tools-libc-20050308cvs-20050608
mspgcc-win32tinyos-20041204-2
tinyos-1.1.15Dec2005cvs-1

Upgrading Cygwin

Moteiv only supports the version of Cygwin installed by Moteiv Tmote Tools. However, any version of Cygwin installed or updated since December 2004 should be compatible, although Moteiv does not support user-installed Cygwin installations. The following Cygwin packages are installed by the Moteiv Tmote Tools CD, and are the minimum required to install TinyOS and its related tools:

```
ash autoconf autoconf-devel autoconf-stable automake automake-devel
automake-stable
base-files base-passwd bash binutils bison bzip2
crypt ctags cvs cygipc cygrunsrv cygutils cygwin
diffutils
editrights emacs expat
file fileutils findutils flex
gawk gcc gcc-core gcc-g++ gcc-mingw gcc-mingw-core gcc-mingw-g++ gdb
gdbm gettext gperf grep groff gzip
less libbz2_1 libcharset1 libdb4.2 libgdbm libgdbm-devel libgdbm3
libgdbm4 libgettextpo0 libiconv libiconv2 libintl1 libintl2 libintl3
libncurses5 libncurses6 libncurses7 libncurses8 libpcre libpcre0
libpopt0 libreadline4 libreadline5 libreadline6 login
m4 make man mingw-runtime minires mktemp more
nano ncurses
openssh openssl
patch patchutils perl perl_manpages postgresql python
rcs readline rpm rpm-build rpm-doc rxvt
sed sh-utils
tar tcltk tcsh termcap terminfo texinfo textutils time
unzip
vim
w32api wget which
zip zlib
```

Upgrading TinyOS

Moteiv Tmote Tools installs new versions of the compilers and tools used by TinyOS. You *may* be able to directly upgrade an existing TinyOS installation using the Moteiv Tmote Tools CD, although you will be on your own if something does not work.

Manual Installation of Moteiv Tools for Windows and Linux

The following RPMs are required to support Moteiv's Tmote Tools. They are provided on the installation CD in the windows/ directory for Windows/Cygwin installations and in the linux/ directory for Linux systems. We recommend these RPMs as they have been tested to work well with TinyOS. In the source/ directory is the TinyOS source and Moteiv's distribution. If all of the above packages are installed on your Windows or Linux machine, you can install the TinyOS and Moteiv tools with the following commands. These commands must be issued in the following order due to dependences. More detailed Linux instructions are available at http://www.moteiv.com/support.php.

First, in the windows/ or linux/ directory (depending on the system that you're using), run the following command:

```
rpm -ivh --force --ignoreos \
 tinyos-tools-1.2.1-3.rpm \
 msp430tools-binutils-2.16-20050607.rpm \
 make-3.80tinyos-1.rpm \
 msp430tools-python-tools-1.0-1.rpm \
 tinyos-javacomm-1.0.0-1.rpm \
 nesc-1.2.4-1.rpm \
 msp430tools-base-0.1-20050607.rpm \
 msp430tools-gcc-3.2.3-20050607.rpm \
 msp430tools-libc-20050308cvs-20050608.rpm
```

Next, in the source/ directory, install the TinyOS and Moteiv source code:

```
rpm -ivh --force --nodeps --ignoreos tinyos-1.1.15Dec2005cvs-1.rpm
rpm -ivh --force --nodeps --ignoreos tinyos-moteiv-2.0.1-1.rpm
```

Tmote Sky Software

Communications

Moteiv's communication system includes three main components: a Multihop mesh networking protocol, a network duty cycling protocol, and the recently proposed "Sensornet Protocol" (SP) abstraction for sending and receiving messages. All of these protocols are used in Moteiv's mesh networking application, Delta. The source code for Delta is located in `/opt/moteiv/apps/Delta`.

Multihop Networking

Moteiv's on-demand ad-hoc networking utilizes spatial and temporal redundancy to reliability deliver messages across a network to their destination. To use the Multihop library in an application, first include Multihop in your configuration:

```
components Multihop;
```

Then wire your application to the appropriate message handlers for your message type. For example, in your configuration:

```
AppM.Send -> MultiHop.Send[APP_ID];
AppM.Receive -> MultiHop.Receive[APP_ID];
```

Where `APP_ID` is a unique 8-bit identifier for your service or application defined in a header file. Please see the documentation for details of using the `Send` interface in Moteiv's API documentation available at `/opt/moteiv/docs/nesdoc`

Messages are submitted to the Multihop service and queued until there is an opportunity to route the message towards the destination. After a message is successfully sent, an event (`Send.sendDone()`) is fired to your service notifying you that it is now safe to use the message buffer for other purposes.

Low Power Operation

Moteiv's software includes a synchronization protocol for low power wireless network. The network duty cycling approaches uses SP (described below) for establishing and maintaining a schedule whereby the entire network wakes up together and then returns to sleep.

Including Moteiv's network duty cycling is as simple as adding a single parameter to the compilation command. Simply add the `lowpower` keyword after the compilation platform. For example:

```
make tmoteinvent lowpower
```

Try the low power networking by using Delta, the mesh networking data collection application, with the `lowpower` option:

```
cd /opt/moteiv/apps/Delta
make tmoteinvent lowpower
```

Be aware that bandwidth is very limited in low power mode (each node is only awake for a few milliseconds every two seconds). The initial synchronization of the network may require up to 15 minutes to stabilize, but will reliably report data after the initial setup phase. *Please be patient!*

Information about Moteiv's network duty cycling is included in the API documentation for the **NetSyncC** and **NetWakeC** components. The source is at **/opt/moteiv/tos/lib/netsync**; however we strongly recommend that only the most advanced users consider modifying this code. Please note that Moteiv does not support any modifications to our source.

Sensornet Protocol (SP)
SP is a unifying link abstraction for running network protocols over a variety of link layer and physical layer technologies without changing network protocol implementation. SP is implemented by the **SPC** component.

SPC and its interfaces are described in detail in the following publication:
A Unifying Link Abstraction for Wireless Sensor Networks
In Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2-4, 2005.
http://www.polastre.com/papers/sensys05-sp.pdf

Messages are transmitted using the **SPSend** interface and message futures are handled through the **SPSendNext** interface. To send a message on a particular AM type, such as AM type 5, wire your network protocol to **SPSend[5]**. The SP message pool will hold on to a message and its corresponding packets until it may be sent over the channel.

Fields of each SP message (**sp_message_t**) should never be directly accessed. Instead, they can be set using the parameters of the **SPSend** interface. Reading parameters should be done through the **SPMessage** interface.

Reception is on a per packet basis (not a per message basis like **SPSend**). Packets are immediately dispatched to higher layer services based on AM type. **SPReceive** provides information about each packet, including a token that identifies which interface a message originated.

The SP Neighbor Table is accessed through the **SPNeighbor** interface. Users must wire to the SP Neighbor Table with the parameter **unique("SPNeighbor")**. Each service has its own identity for controlling the insertions, removals, and changes of entries in the SP Neighbor Table. See the **SPNeighbor** interface in the API documentation for more information.

Various utilities as part of SP's processing are available in the **SPUtil** interface. These utilities include link estimation functions and link post-arbitration time stamps.

Useful TinyOS Components

There are many useful libraries including with Moteiv's distribution of TinyOS. Below, many of these components and their functions are listed. For additional resources, please check Moteiv's support website at http://www.moteiv.com/support.php. It is frequently updated with tips, techniques, and troubleshooting articles.

TinyOS Distribution Organization under the `/opt/moteiv` directory:

Directory	Description
apps	Moteiv applications
apps/invent	Tmote Invent-specific applications
doc/nesdoc	API documentation in HTML format
tos/lib	TinyOS libraries (SP, Multihop, etc)
tos/platform/tmote	Tmote-specific platform components
tos/sensorboards/invent	Tmote Invent driver components
tools/java	Moteiv mote-interface java tools
tinyos-1.x	TinyOS components used by Moteiv

Useful TinyOS components:

TinyOS Component	Function
BitVectorC	Methods to manipulate vectors of bits
LedsC	Turn on or off the LEDs
MainControlC	Start a component on boot with MainControlC
ObjectPoolC	Create and manage a pool of generic objects
TimerMilliC	Create a new millisecond system timer
UartDetectC	Detect if a PC is active & connected
UartPresenceC	Detect the presence of PC's USB port
UserButtonC	Enable input from User Button
UserButtonAdvancedC	Advanced functionality from User Button

The documentation for all of these components is available within the `/opt/moteiv/doc/nesdoc` directory.

In addition to TinyOS components, there are a few useful C libraries available in `/opt/moteiv/tos/lib/util/`:

C Library	Function
circularQueue.h	A generic circular queue object
fft_i8.h	8-bit integer FFT routines

General Information

Document History

Revision	Date	Notes
1.0	2005/03/04	Initial Release
1.1	2006/03/01	Updated for Tmote Tools 2.0

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.

Address Information

Web site: http://www.moteiv.com
E-mail: info@moteiv.com
Technical Support Web site: http://www.moteiv.com/support.php
Technical Support E-mail: support@moteiv.com
Phone Number: +1.415.692.0960
Fax Number: +1.415.358.4872

Headquarters

Moteiv Corporation
55 Hawthorne St, Suite 550
San Francisco, CA 94105