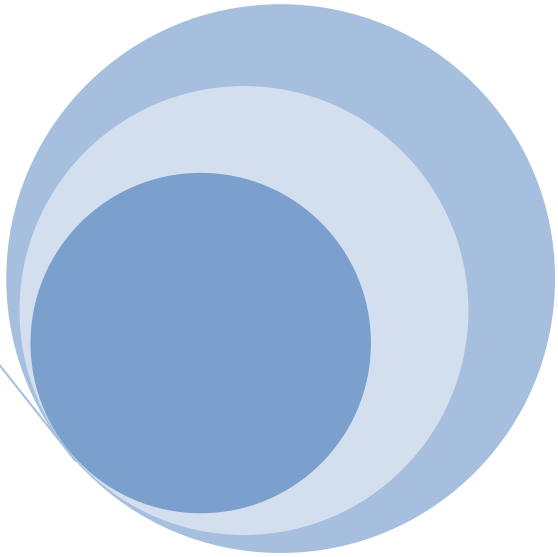
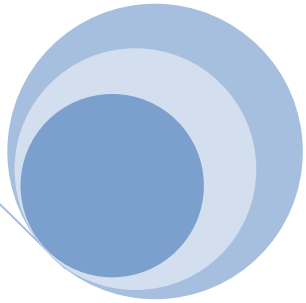


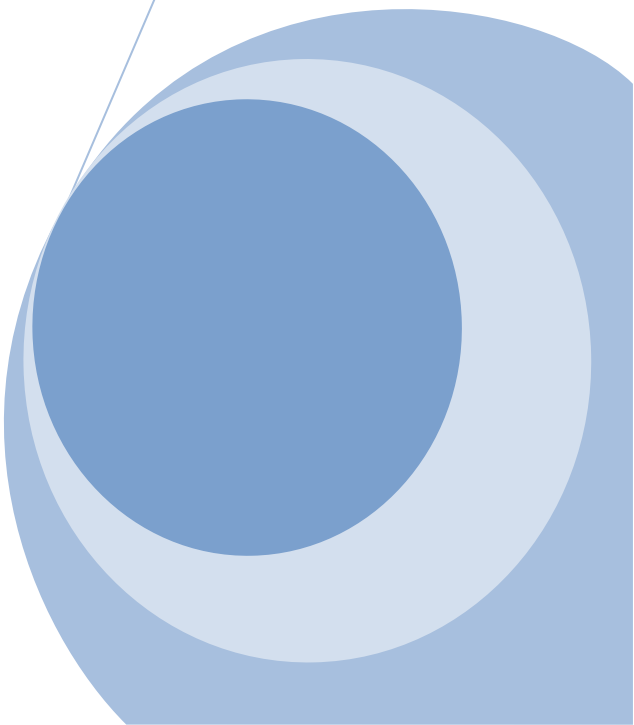


LiteOS
User's Guide

Version 1.0
Last updated: Oct 5 2008

This guide illustrates how to get started with the LiteOS operating system

Download location: www.liteos.net



Contents

PREFACE **1**

MAILING LISTS: **1**

ADDITIONAL REFERENCES **1**

INSTALLATION OF LITEOS **2**

INSTALLING LITEOS ON WINDOWS MACHINE 2
Installing JDK 2
Installing Cygwin 3
Check that Cygwin and Java have been correctly configured. 4
Install the core LiteOS components 4
Hardware Connection 9
INSTALLING LITEOS ON LINUX MACHINE 13
Installing JDK 13
Installing the Java Communication API 14
Installing the GNU avr-gcc toolchain 15
Compiling and Installing uisp 16
Setting privileges for serial ports 17
INSTALLATION LITEOS OPERATING SYSTEM ON THE MOTES 17
Install the base 19
Install the kernel 21
Installer Express Mode 24
Using LiteShell 24

USE LITESHELL COMMANDS **26**

LIST CURRENT DIRECTORY INFORMATION 26
CREATE AND DELETE FILES AND DIRECTORIES 28
COPY DATA BETWEEN THE PC AND A NODE 29
OTHER COMMAND EXAMPLES 31
INTERACTING WITH THE SENSORS 33
Using the Light Sensor and the Temperature Sensor 33
Using the LED 35
INSTALLATION OF SAMPLE APPLICATIONS 36

APPENDIX **38**

APPENDIX I: A LIST OF LITESHELL COMMANDS SUPPORTED 38
APPENDIX II: MAN PAGES 39

TROUBLESHOOTING TIPS **55**

INDEX **56**

Preface

LiteOS provides a UNIX-like environment for sensor networks, networked embedded devices, and cyber physical systems. It provides a thread-based run-time execution environment for applications. The goal of this User's Guide is to get you familiarized with its environment. For programming in LiteOS, see Programmer's guide.

For updates, check:

www.liteos.net

Mailing lists:

liteos-users@cs.uiuc.edu

http://lists.cs.uiuc.edu/mailman/listinfo/liteos-users

liteos-developers@cs.uiuc.edu

http://lists.cs.uiuc.edu/mailman/listinfo/liteos-developers

Additional References

The following are additional documentation, available on www.liteos.net

- LiteOS Programmer's Guide
- LiteOS Application Notes

Installation of LiteOS

This section provides a step-by-step guide on how to install LiteOS on your computer.

Following are the requirements for constructing the working environment of LiteOS:

- **MicaZ or IRIS nodes with Atmega128 processor, at least two nodes are required, one as base station, the other as the experiment node, three or more suggested.**
- **MIB510 Programming Board or MIB520 Programming board (at least one of either of them)**
- **PC installed with Microsoft Windows XP or Linux**
- **Serial or USB port (at least one port on the PC)**

Installing LiteOS on Windows machine

The first step to configure your PC is to install required third-party software. The following software is needed:

- **Java JDK 1.6 or later**
- **Cygwin (must installed with Perl, Python, Makefile)**

You may skip this section if your PC is already installed with JDK and Cygwin.

We next explain how to install various third party software and configure them.

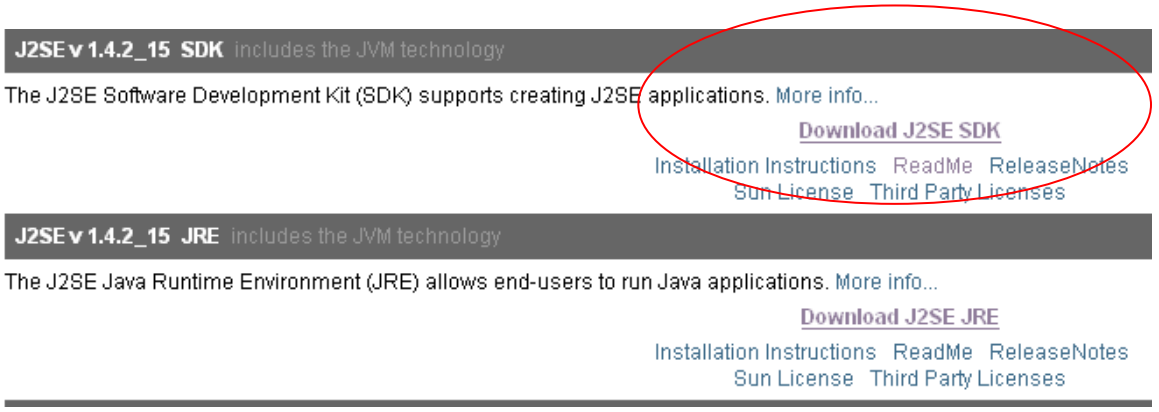
Installing JDK

If you have installed JDK 1.6 or later on your computer, you may skip this section.

Otherwise, click
http://java.sun.com/javase/downloads/index.jsp

to download the latest version of JDK. As of the time this document is being written, this version is JDK 6 update 7.

Click to download Java SDK (the actual download page may be different from shown here)



Then install J2SE SDK on your computer, following its instructions.

Installing Cygwin

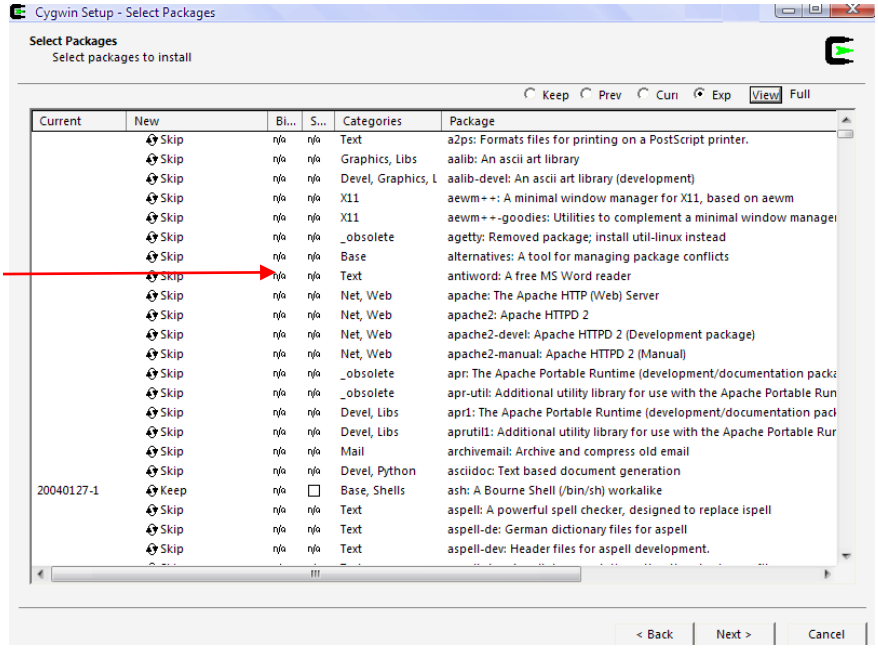
Download Cygwin setup from

http://www.cygwin.com/

Install Cygwin following its instructions. Note that it may take a while to download all Cygwin packages from the Internet, so choose an Internet mirror for Cygwin that is near your location.

When installing Cygwin, you must select **Python, Perl, GCC, and Make** during its installation. More specifically, select such packages in the following screenshot during the installation of Cygwin:

Select cygwin packages here



Check that Cygwin and Java have been correctly configured.

Open the Cygwin window, and type the following:

$ java –version

java version "1.6.0_07"

Java(TM) SE Runtime Environment (build 1.6.0_07-b06)

Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode, sharing)

If you do not see a Java version that matches your configuration, then you probably have not installed JDK on your machine correctly, or have not set up the environment variables correctly.

Install the core LiteOS components

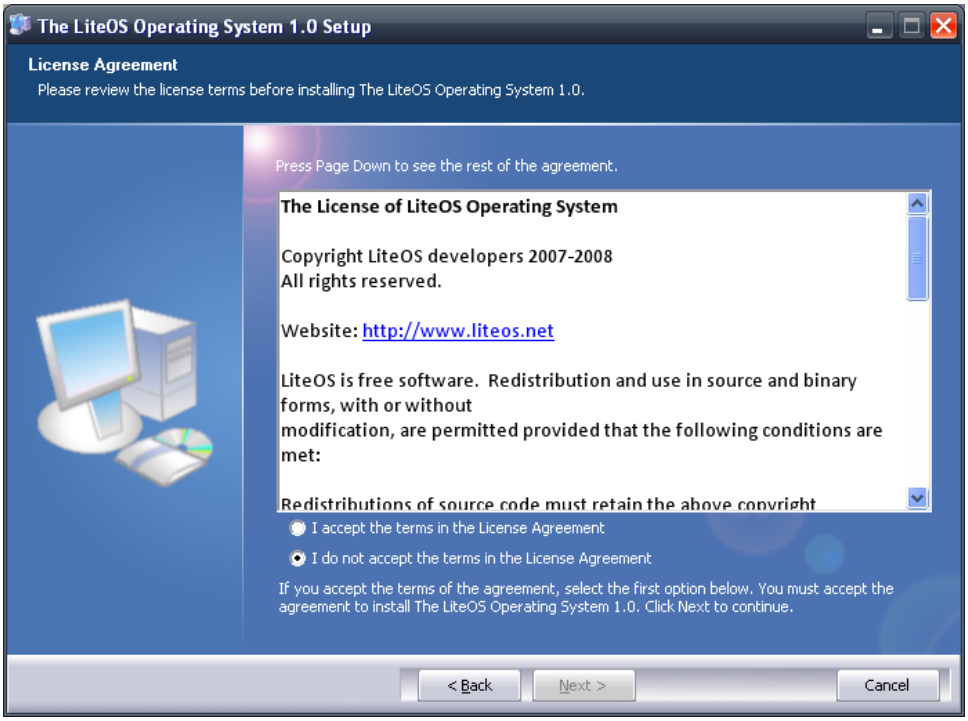
Next, download the LiteOS setup software from the LiteOS website. The updated version is 1.0. Follow the installation instructions to install LiteOS on your computer. Note that this setup program also includes WinAVR 20070122 version. If you have already installed this software before, you may choose not to select this component in the installation process.

 On Windows Vista, the LiteOS setup installer will need administrator's privileges to proceed.

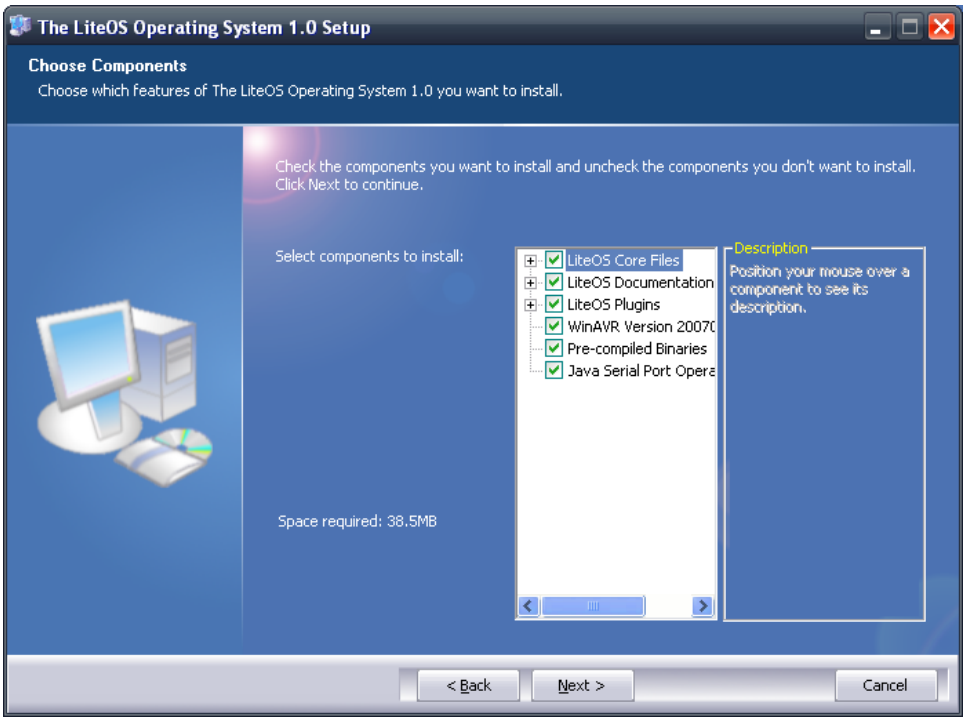
The following screenshots show the basic installation steps of LiteOS on your PC.

The welcome screen, click Next to proceed.

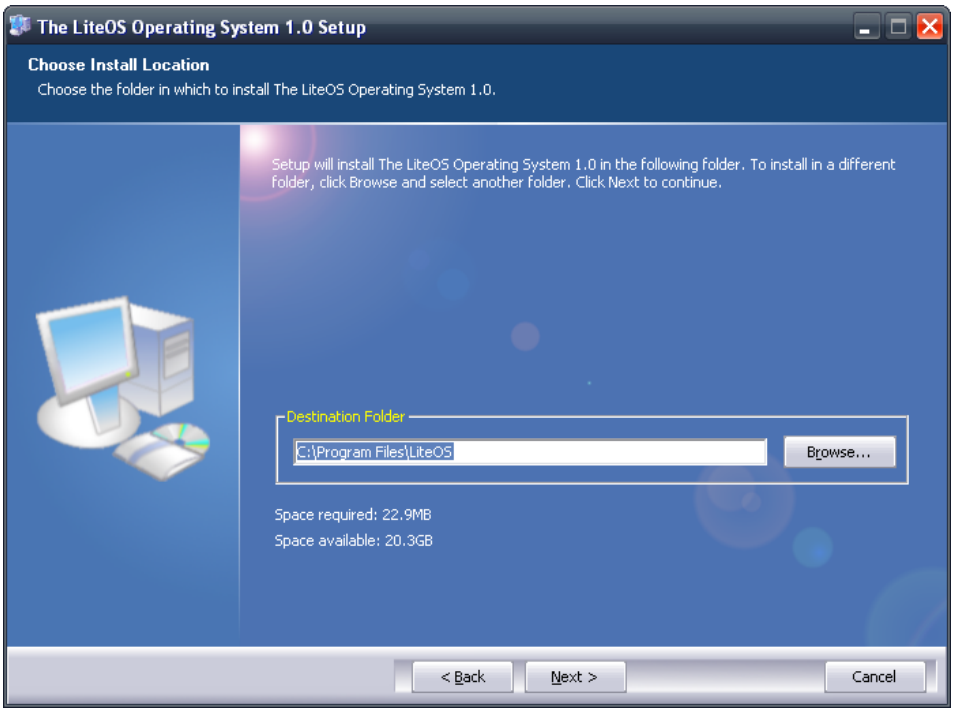
The license screen. If you accept the license (FreeBSD license), click Next to proceed.



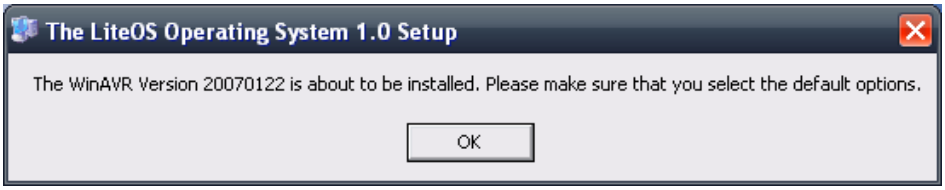
The component selection screen. You may choose the components that you want to install in this dialog. Note that you may view the details and descriptions of different components by moving your mouse on top of them.



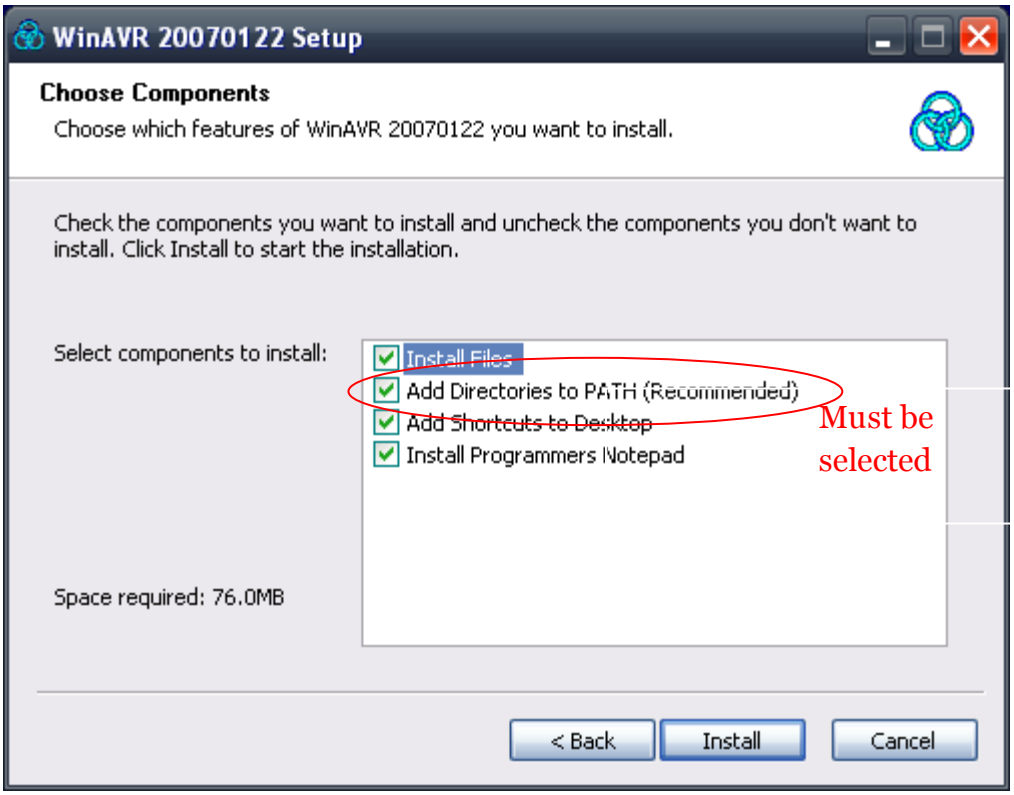
Select the file directory that you wish to install LiteOS to. Note that all directories are accessible from Cygwin as follows. By default, Cygwin has a root directory of /cygdirve, and your local directories are accessible from here. You may also directly type "cd c:" to enter your local directories directly.



Next click install to proceed. If you selected WinAVR as a component in the previous step, the following screen will appear:

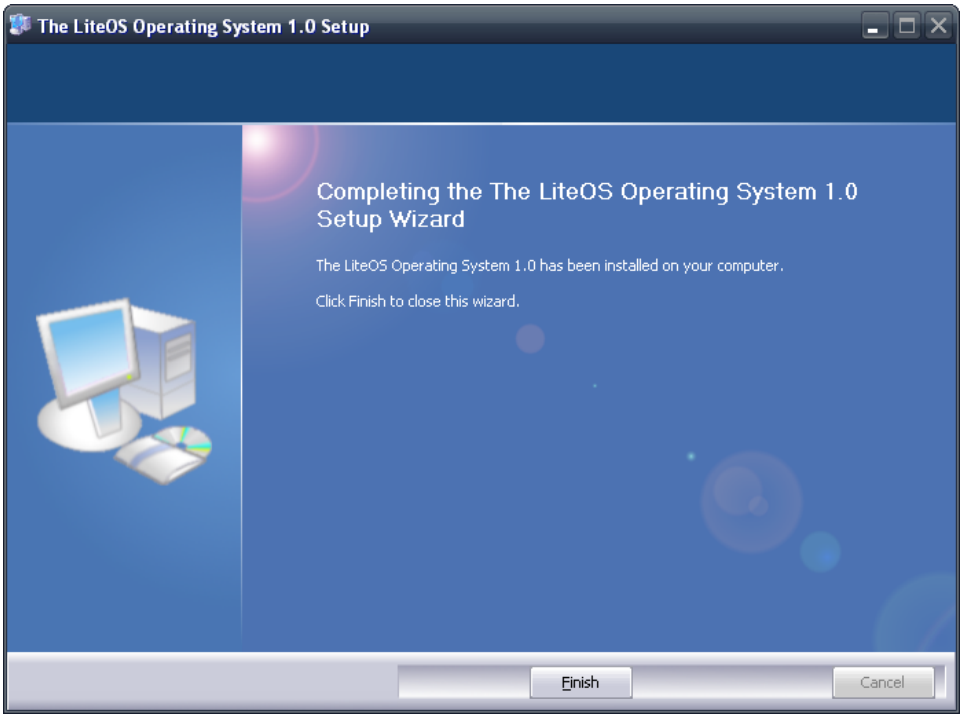


In the next a few dialogs, you need to follow the default selections of WinAVR to ensure that it is properly installed. In particular, if you meet the following screen:



Make sure that you select the option "Add Directories to PATH". The programmers notepad here is a notepad-like program that displays your code. You may choose to de-select it if you have already installed other text editors on your computer.

Finally, you will get the following dialog, showing that LiteOS has been properly installed.



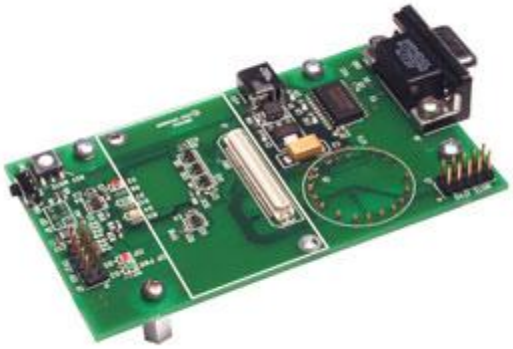
After you install LiteOS on your PC, an entry item in the start menu will be created, containing shortcut links to various components of LiteOS. As the final step of the installation, you may log into the Cygwin environment, and type "verifyLiteOS" to verify that LiteOS has been correctly installed on your computer.

Hardware Connection

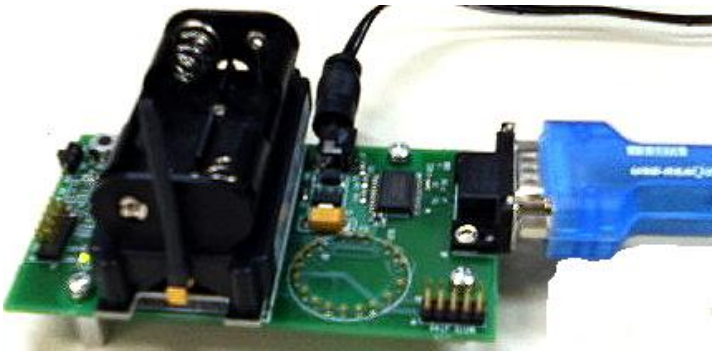
To set up the hardware for evaluating the LiteOS, follow the steps below.

For PCs with SERIAL PORT

Connect the MIB510 programming board to the PC serial port using the cable provided with the MIB510 hardware. An image of MIB510 is as follows.



MIB510 with a MicaZ plugged:



You may also use MIB520, which connects directly to the USB port of the PC. An image of MIB520 is displayed as follows.

The MIB520 programming board need a separate driver, available here:

http://www.ftdichip.com/Drivers/VCP.htm

This driver is also available in the LiteOS software, available at the YOUR_INSTALL_DIRECTORY/Drivers/MIB520_Driver. This driver need to be installed manually just like any other Windows driver.

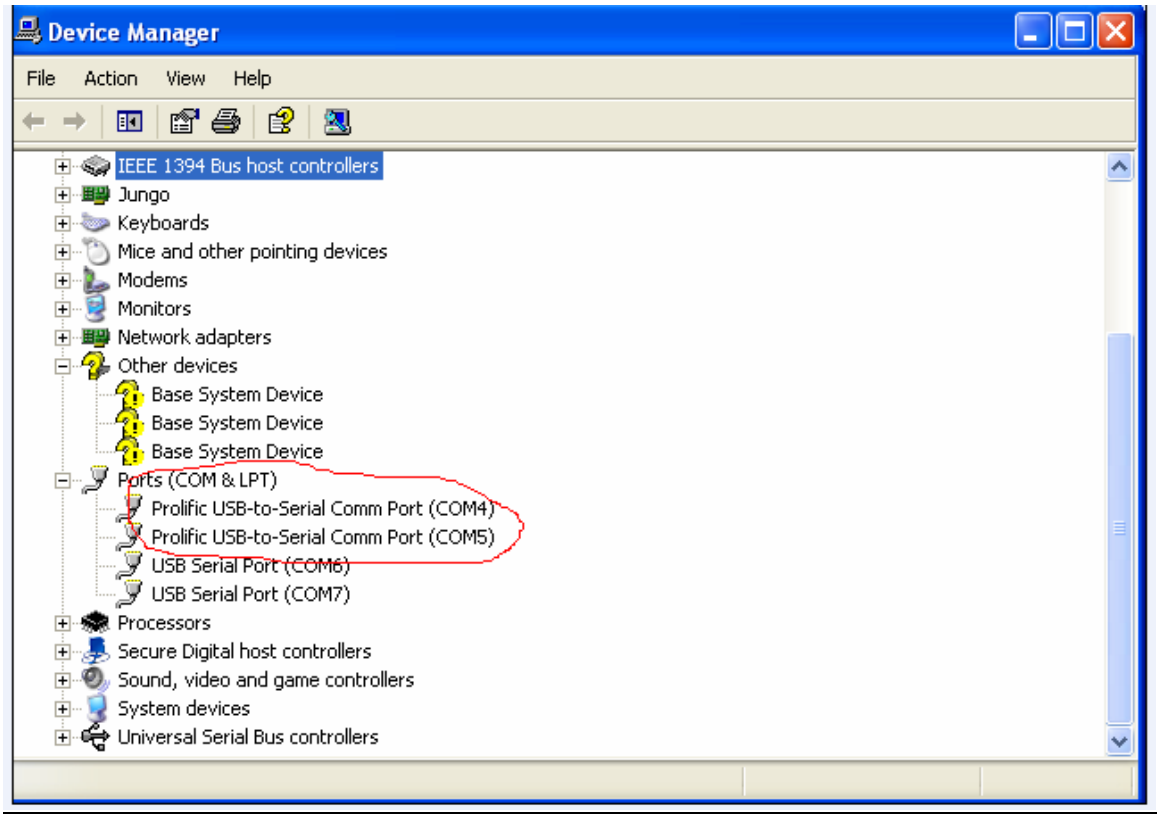
In most cases, you will be using COM1 of your PC to communicate with MIB510. MIB520, on the other hand, uses a pair of COM parts. Finding out exactly which port you are using to communicate to the programming board is very important. You have to later use such port information in various commands. To find such information, open the device manager and find out which ports are being displayed and used.

For PCs with a USB PORT and using MIB510

Some modern PCs, and especially laptops, do not have a serial port. Instead, they supply more than one USB ports. In case you have a MIB510 programming board, it is a little trickier to connect hardware. This section illustrates how to connect MIB510 with USB ports.

To get started, you need to obtain a USB-to-Serial adaptor. The model we tested and found working is Prolific CP-US-03 (a driver is available at http://www.cornerproducts.com/support.html). You may also try out similar models for this purpose.

Once you obtain the USB-to-Serial adaptor, install its driver, and you should see the following in your device manager:



In our settings, we are using two adaptors for two programming boards. Hence, we see two USB-to-Serial ports in our device manager as illustrated above. Note that your COM number will probably be different. Remember the particular port number of your PC. Later, you are going to use this port to communicate with MIB510.

Caution:

> When you connect the programming board and hardware, you may be confused by the on/off buttons on the board and the node. The short answer to the on/off button setup is: turn them all on, but usually this does not matter. That is, even if you forget to turn on the mote or the programming board, this usually does not have any effect on some boards. So don't be confused if you make similar observations.
>
> We also observe that, however, some programming boards do tend to be affected by the on/off buttons. If you meet

problems later such as the LiteOS shell fails to communicate, one debugging technique is to switch the buttons on the programming board. The mote, of course, must be turned on later.

Installing LiteOS on Linux machine

The installation of LiteOS on Linux is somewhat more complicated than on Windows machine since the WinAVR package is not officially incorporated into GNU gcc toolchain. However, the patches provided by WinAVR still work well on Linux some environments. LiteOS has been successfully tested and run on Fedora Core 7. The following software is needed to run LiteOS on Linux machine:

- **Linux with latest version of Perl and Python installed**
- **Java JDK 1.4.2 or later**
- **Java Communication API package**
- **GNU avr-gcc toolchain**
- **uisp utility**

We next explain how to install those software and configure them.

Installing JDK

We recommend that you use Sun Java JDK instead of GNU JDK provided with Fedora Core. If you have installed JDK 1.4.2 or later on your computer, you may skip this section.

Otherwise, click
http://java.sun.com/javase/downloads/index.jsp

to download Java JDK. We recommend that you download the latest version of Java JDK.

Select the latest version of JDK and click Download. Then choose Linux Platform and the Linux RPM in self-extracting file to download, for example jdk-6u4-linux-i586-rpm.bin

Next, change to the directory where you have downloaded the file and execute the following command:

- To become *root* :
  - **su**
- Make JDK installer file executable :

    - **chmod a+x jdk-6u4-linux-i586-rpm.bin**
- Finally, execute JDK install
    - **./ jdk-6u4-linux-i586-rpm.bin**

After installation you can delete installer and the temporary files it created on your directory.

Now, there can be multiple copies of JDK in your machine, you have to select the Sun JDK as the default Java Engine. In the terminal window type in the following command:

- Become *root*
    - **su**
- Add newly installed Sun JDK to the list of Java alternatives
    - **/usr/sbin/alternatives --install /usr/bin/java java /usr/java/latest/bin/java 2**
- Finally, select the Sun JDK as your default Java Engine
    - **/usr/sbin/alternatives --config java**

To check which Java Engine is currently in use, issue the command

- **java --version**
- **javac --version**

If the display looks similar to *Java (TM) SE Runtime Environment* then congratulations, you have successfully installed Sun Java JDK.

Installing the Java Communication API

Java Communication API is a Java extension providing access to RS-232 serial ports and IEEE-1284 parallel ports developed by Sun. Go to the following webpage to download Java Communication API

http://www.sun.com/download/products.xml?id=43208d3d

Follow the instruction on the webpage to download the compressed package. After downloading the package, unzip it, you should find the **doc** subdirectory inside the package which contains the html documents on how to install the API to your system. We recommend that you unzip the file into /usr/local/ directory. Follow the instruction in the Linux section and make sure that you *set all the environment variables properly* so that the system could be able to find all the files.

Installing the GNU avr-gcc toolchain

Currently the GNU gcc does not support compiling ATMEL AVR microprocessors which is currently used in MicaZ motes. But WinAVR did supply patches which allows gcc generate code for AVR. You will have to download GNU binutils, GNU gcc and avr-libc and also their patches at WinAVR website. Make sure that you have administrative privileges for the following commands.

First, install the prerequisite compilation tools:

- **yum install flex bison byacc gcc gcc-c++**

Second, go to the following webpage to download binutils 2.18 to your computer

ftp://sources.redhat.com/pub/binutils/snapshots/

You have to choose exactly the tarball package of version 2.18 to download because patches for binutils is now released for version 2.18. Untar the file and go to the following webpage to download the patches for binutils

http://winavr.cvs.sourceforge.net/winavr/patches/binutils/

put all the patches into the same directory as the GNU and apply the command for each patch file (e.g. *patchfile.patch*) :

- **patch –p0 < patchfile.patch**

Now you are ready to compile binutils:

- **./configure --target=avr --program-prefix="avr-"**
- **make**
- **make install**

Third, you have to download and compile gcc wich is similar to the second step. Go to the following webpage to download GNU gcc

http://gcc.gnu.org/mirrors.html

Select working mirror and download GNU GCC version **4.22** then download patch for AVR chips at

http://winavr.cvs.sourceforge.net/winavr/patches/gcc/

Apply all patches as you did in the second step and execute the following command to compile and install avr-gcc into your system:

- **mkdir avr-gcc**
- **cd avr-gcc**
- **../configure --target=avr --program_prefix="avr-" --enable_languages=c**
- **make**
- **make install**

Finally, download the latest version of avr-libc at

http://savannah.nongnu.org/projects/avr-libc/

We suggest that you unzip this file into your /local/usr/ directory and compile this using

- **./configure**
- **make**
- **make install**

Compiling and Installing uisp

Assuming you have downloaded the LiteOS current release, you will find a subdirectory called **Tools**. Navigate to the directory **Tools/uisp** and please type in the following:

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

Those scripts will compile and install **uisp** utility into your system at /usr/local/bin.

The following sections on installing LiteOS to the motes and using Tools distributed with LiteOS are the same in Linux and Cygwin environment.

Setting privileges for serial ports

In Linux, the I/O devices are treated as a file stored in /dev directory. The serial ports files are usually /dev/ttySn where in is the number of the serial port, for example serial port 0 is /dev/ttyS0. Normally, only root has the privileges to access to those serial ports. You have to set the privilege for the user using *chmod* command. For example, the following command set the privilege for the owner to read and write on serial port 1

- **chmod o+rw /dev/ttyS1**

You can also set which user have the right to access to the serial ports by editing the file /etc/group file and add the username to both tty and uucp lines. Note that only root has the right to change the content of this file.

Installation LiteOS Operating System on the Motes

So far, you have successfully installed LiteOS on your PC. Congratulations! Next you want to install the kernel to the motes. You may already have several MicaZ or Iris motes at hand, and you have connected the programming board to your PC. The guide that follows illustrates how to install the LiteOS components on the motes.

The model followed by LiteOS is simple: a base station node receives your commands from the PC and translates them into message formats that can be understood by other nodes. Therefore, you need to install two images: the base station image, and the kernel image.

Now start the Cygwin you have installed earlier. Note that if you are using Windows Vista, you have to start Cygwin as follows: right click it, and select "Run as Administrator". This is because by default, the UAC mechanism of Windows Vista prevents Cygwin to perform certain actions such as creating new files and directories.

Now enter the /cgydrive/YOUR_LITEOS_PATH/Tools directory. This directory contains the files you have previously installed. By default, this directory is "cygdrive/c/Program\ Files/LiteOS".

Now first enter the Tools directory. Type:

"./verifyLiteOS.sh"

If this script ends with:

The files have been generated in the bin directory
Copying the Kernel images...

Then the script is completed successfully. If you see something else, then probably some error has occurred. While we have extensively tested the code to make sure no error occurs, one common problem is that you may encounter the following error:

cp: cannot stat
`../../../SourceCode/LiteOS_Kernel/bin/micaz/LiteOS.hex': No such file or directory

This is caused by the compilation errors of the kernel. This may be caused by the different version of WinAVR. Type the following command to verify the avr-gcc version:

avr-gcc --version

You should get 4.1.1 with WinAVR 20070122. If you find your version is incorrect, this may be caused by 1) you installed a different version of WinAVR or 2) you have multiple copies of WinAVR on your PC. To find out the location of avr-gcc on your PC, type:

which avr-gcc

This command will tell you where the Cygwin environment locates the avr-gcc binaries. You may modify the PATH environment variable to make sure that the correct version of avr-gcc is located.

If everything goes fine, you may start the interactive installer we have developed, located at /YOUR_LITEOS_PATH/Tools/JavaTools/classes/installer directory. We now describe how to install the base and the kernel using the installer.

Install the base

Before getting started, find out which port you are using on your PC. Open the device manager, under port (COMs and LPT) tab, and find out which port your PC is using. Note that this setup is machine-specific. Normally, you will find that MIB510 connects to COM1 of your PC for the serial port. In case you use a serial-to-USB adaptor, you may find that another port is being used. In case you use MIB520, you will find a pair of ports, say, COM5 and COM6, are used. In our illustration below, we shall assume that the serial port used is COM5 and COM6, given that we are using a MIB520 to do the installation.

Note: Optional setup

Both the interactive installer and the LiteOS shell later supports colorized output. We use terminal characters for colorized output. This is turned on and off by using a –color switch when invoking the installer and the shell. Unfortunately, native Cygwin environment does not support colorized output. Therefore, we suggest you turn it off when using Cygwin. On the other hand, you can use other terminal emulators, such as rxvt, that supports terminal characters. In such cases, you may turn on the colorized output as follows.

We give the following example using rxvt. You may start the rxvt by invoking:

```
$ rxvt -sr -fn "Lucida Console-12" -g 80x40 -bg black -fg cyan -e /bin/bash --login –i
```

> There is no difference between using rxvt and cygwin other than that rxvt supports colorized output. In rxvt, we typically use red color to display warning messages, and use green color to display correct messages. Other than the –color switch, all the other commands are identical when you use rxvt and when you use cygwin.

Now cd into the Tools/JavaTools/classes directory, and type the following to invoke the installer:

```
java tools.installer.installer (-color)
```

Then you can use the –color option to turn on the colorized output.

Note:

You may meet the ClassNotFound error when you use java. In this case, the reason is that you have not invoked the class file in the correct directory. Make sure that you have cd into the /classes directory and use the java command to start the installer.

Then, follow the interactive installer to install the base. The installer asks the following questions:

The port you are using: usually, you are using COM1 if you are using MIB510. Otherwise, input your COM number in case you use USB port or MIB520. You may view your port configurations in the device manager.

Are your using MIB510 or MIB520: input your programming board model

The node ID: input a number between 1 and 255. It is suggested that you reserve node ID 0 for the base station. **Very important: if you install multiple nodes with the kernel image, make sure that each node has a unique ID.** Otherwise, the shell will not work correctly.

The binary image name: input the binary image name. There are four possible cases by default: the LiteOS_micaz.hex file for installing the LiteOS kernel on MicaZ nodes, the LiteOS_iris.hex file for installing the LiteOS kernel on Iris nodes, the

Base_micaz.hex file for installing the LiteOS base on MicaZ nodes, and finally, the Base_iris.hex file for installing the LiteOS base on Iris nodes.

Then the installer will install the image into the mote. If successful, it will ask whether you want to proceed with the kernel configuration. For installing the base station, select **n**.

If you are installing the kernel image instead of the base image, proceed to the next section.

Install the kernel

After following the similar previous procedure to install the kernel, you need to answer **y** when asked whether you want to configure the kernel. The installer will ask the following questions:

The network name of this node: In LiteOS, each node is assumed to be part of a sub-network, where each such network must have a name. From the shell's perspective, such sub-network appears as a directory. Therefore, choose a name that you think is easy to manage, for instance, **net1**.

The node name: Similarly, LiteOS assigns a name to each node, which is also used as a directory name. Choose a name that you think is appropriate, for instance, **node1**.

Another approach for naming the network and the node is that you may use IP addresses. For instance, you may name a node as 192.168.0.1. The important thing to keep in mind is that you should not use more than 16 letter/digits in the naming selection, otherwise the node will not be correctly named.

The communication channel: You also need to select the communication channel for later communication between the base node and the remote node. You may select any channel between 11 and 26. Once the channel is selected, you may change the channel used by the base station using the **setchannel** command. For example, if you have selected 11 for a remote node, later, you want to use the following command:

setchannel 11

To manually change the channel used by the base to channel 11.

At the end of the installation, the mote will reboot. The next time you start the node, its file system is formatted. During reboot, you will find that all three LED on the mote (or on the programming board, if the mote is still plugged into the programming board) will be turned on during the file system formatting procedure. If the three LED fail to turn on, then the image must have not been successfully burnt into the program flash of the mote. In this case, please repeat the procedure above to re-install the image.

Note:

> If your sensor node is faulty, or the programming board is faulty, you will encounter error messages. Checking your connections and power or changing the hardware, if needed, may solve this problem.
>
> Some errors in serial communication can be solved by remove and re-connect the programming board cable. Such errors may have different warning messages printed.
>
> A typical installation (successful) and a failed installation (because the node is not plugged in) look like the following:

```
/* A typical installing procedure for the kernel image. */
$ java tools.installer.installer
Starting the LiteOS installer...
Please set up the following configuration parameters:
Please input the port number you intend to use: (e.g. COM1)
COM7
Are you using the MIB510 or MIB520 programmer? (1 (or MIB510) and 2 (or
MIB520)
2
Please input the node ID (0 for the base station and 1-255 for other nodes).
1
Please input the binary image name (ending with hex). Leave blank or CTRL-C to
exit
LiteOS.hex
Now installing the image into the mote...
First step Installation complete.
Do you want to proceed with kernel configuration? (y/n, this step is not needed
if you are installing the base station or generic hex applications.)
```

```
y
Please input the network name that this node belongs to. (no more than 16
letters/digits.)
sensornet1
Input the name of this node: (1-16 letters/digits. You may use IP addresses, such
as 192.168.0.1., or more intuitive names, such as MyNodeA, etc.)
192.168.0.1

Now reboot. The mote should turn on all three LEDs during reboot, and turn them
off when reboot finishes. If the mote fails to turn on all three LEDs, reboot it
manually, or reinstall the mote.
Installation complete. Bye.
```

```
/*A failed installing procedure for the kernel image. In this scenario, the mote is not
connected to the programming board. */
$ java tools.installer.installer
Starting the LiteOS installer...
Please set up the following configuration parameters:
Please input the port number you intend to use: (e.g. COM1)
COM5
Are you using the MIB510 or MIB520 programmer? (1 (or MIB510) and 2 (or
MIB520))
2
Please input the node ID (0 for the base station and 1-255 for other nodes).
1
Please input the channel for communication. (11-26, only used by the kernel
installation)
15

Please input the binary image name (ending with hex). Leave blank or CTRL-C to
exit
LiteOS.hex
Now installing the image into the mote...
Installation fails. Here is the diagnosis information:
<Diagnosis info starts>
Firmware Version: 1.8
Probably the AVR MCU is not in the RESET state.
Check it out and run me again.
<Diagnosis info ends>
```

Please note that if you are using the Iris motes, the channel selection currently is not yet fully supported, and hence you don't need to select the channel to communicate with nodes.

Installer Express Mode

Currently, the installer also supports an express mode, in addition to the interactive mode as described earlier. The way to use the express mode is simply adding parameters to the command line. For example, a simple example can be written as follows:

```
java tools.installer.installer COM7 MIB520 1 15 LiteOS_micaz.hex sensornet1 node1
```

where the node is assumed to be node1 in a network called sensornet1

For the base station installation, simply use:

```
java tools.installer.installer COM7 MIB520 1 15 Base_micaz.hex
```

Using LiteShell

Similar to the installer, you may invoke LiteShell under Cygwin or rxvt. Again, type the following to start rxvt.

```
$ rxvt -sr -fn "Lucida Console-12" -g 80x40 -bg black -fg cyan -e /bin/bash --login –i
```

Now navigate to the JavaTools/classes subdirectory. Start the shell as follows.

```
$java tools.sf.SerialForwarder -comm serial@COM9:57600
```

This will start serial forwarder. In this case we assume we are using COM9. **Note that, when you use MIB520, which shows up as a pair of communication ports, like COM5 and COM6, you must use the higher number of port in this command.**

Then you input:

```
$java tools.terminal.terminal –color
```

Note that, in this case, the terminal program will connect to the serial forwarder to communicate. If the serial forwarder is not started, you will encounter an error like following:

```
$Error on sf@localhost:9001: java.net.ConnectException: Connection refused:
connnect
```

Note:

You may encouter a Java exception that says it cannot find the main class. If so, make sure that your current working directory is under the classes directory. You do not need to set any extra environment variables to invoke the terminal.

Use LiteShell Commands

Now that you have installed the LiteOS kernel and the base station on a bunch of nodes, you may wonder how to operate a sensor network using LiteShell. The command details of LiteShell are explained in this section. You will also see how you can upload sensor network applications to nodes and execute them.

Suppose we here installed LiteOS on several nodes, we can then use LiteShell commands to operate them.

List current directory information

LiteOS offers a logical view of the sensor network in which they appear to be "mounted" to the root of the PC's directory tree. Hence, the user should expect to see a set of directories, one for each sensor sub-network. Within each, the user should expect to see a set of subdirectories, one for each node of the given sensor network. Finally, within a node, the user should expect to see the local file system. Currently, the file system contains a subdirectory /dev, which contains the sensor device drivers.

The example below illustrates the use of common Unix commands to list directory information on the nodes. In most cases, we use the ls command. This command has the option of –l, which lists detailed information. Ls always displays the current up-to-date information.

Caution:

Every time a mote is turned on, it takes about 1-2 minutes to finish the boot-up process when it formats the file system, among other tasks. Please be patient during this phase. Typing ls during this time will get no results, but this is normal.

```
$ java tools.terminal.terminal
LiteOS [Version]
$ls -l
The returned has 3 packets.

Name    Type    Size    Protection
sensornet1  network  --    rwxrwxrwx
Time elapes 547

$cd sensornet1
cd command successful
Time elapes 0

$ls -l
The returned has 3 packets.

Name    Type    Size    Protection
nodeC  noderoot  --    rwxrwxrwx
nodeB  noderoot  --    rwxrwxrwx
nodeA  noderoot  --    rwxrwxrwx

Time elapes 500

$cd nodeA
cd command successful
Time elapes 0

$ls -l
The returned has 1 packets.
Name    Type    Size    Protection
dev  directory  --    rwxrwxrwx
Time elapes 516

$cd de
No such subdirectory exists. Probably you have not used the ls command to list
uch directory. Currently cd only works for directories that are listed using ls
Time elapes 0

$cd dev
cd command successful
Time elapes 0

$ls
The returned has 6 packets.

led
light
```

```
temp
magnet
accel
radio
```

Create and delete files and directories

The next example illustrates creating and deleting files and directories. The commands involved are ls, cd, mkdir, touch, and rm.

```
$ls
The returned has 3 packets.
nodeC
nodeB
nodeA
Time elapes 500

$cd nodeA
cd command successful
Time elapes 0

$ls
The returned has 1 packets.
dev
Time elapes 500

$mkdir test1
Make dir complete
Time elapes 265

$touch test2
creating file complete
Time elapes 266

$ls -l
The returned has 3 packets.
Name  Type  Size  Protection
dev  directory  --  rwxrwxrwx
test1  directory  --  rwxrwxrwx
test2  file  0  rwxrwxrwx
Time elapes 532

$rm test1
Time elapes 32
```

```
$ls -l
The returned has 2 packets.
Name   Type   Size   Protection
dev  directory  --    rwxrwxrwx
test2  file  0    rwxrwxrwx
Time elapes 500
```

Copy data between the PC and a node

By using the cp command, you may also copy data from and to the PC. The current copy status is displayed in real time to help locate network problems.

Remember that a wireless sensor network is assumed to be organized into one or more sub-networks, each of which has a unique network name. Each sub-network, in turn, has multiple nodes, each again with a different name. The local file system on each node is organized hierarchically. To address a file in a sensor network, therefore, the user simply provides a complete location, or a relative location to the current working directory.

To address a file on your PC, you are required to start the file location with **/.** For example, if a file is located at c:/Data/mydata.txt, then it can be written as /c/Data/mydata.txt in LiteShell.

Caution:

> LiteShell currently does not support navigating in your PC directories. Nor does it support copy file between nodes. To do the former, use Windows Explorer. To do the later, please use the PC as the relay.

In the following example, we copy a file called testcp.zip from the location c:/Temp/testcp.zip to the current working directory of /sn01/nodeC (the root directory of nodeC). To achieve this, do the following:

```
cp /c/Temp/testcp.zip testcp.zip
```

You can again copy it back to your PC location c:/Temp/testcpback.zip by:

```
cp testcp.zip /c/Temp/testcpback.zip
```

The copy operation in LiteShell is reliable. Hence, you can be certain that when you copy large files, not one byte of it will be corrupted by the underlying unreliable radio communication.

```
$ls -l
The returned has 2 packets.
Name    Type    Size    Protection
dev  directory  --     rwxrwxrwx
test2  file   0     rwxrwxrwx
Time elapes 515

$cp /c/temp/test.zip test.zip
The reply has 1 packets.
Now trying to send 0 30
Now trying to send sync
Now reply is good on sync
Now trying to send 30 60
Now trying to send sync
Now reply is good on sync
Now trying to send 60 90
Now trying to send sync
Now reply is good on sync
Now trying to send 90 106
Now trying to send sync
Now reply is good on sync
cp succeed
Copy finished
Time elapes 10375

$ls -l
The returned has 3 packets.

Name    Type    Size    Protection
dev  directory  --     rwxrwxrwx
test.zip  file   5254     rwxrwxrwx
```

```
test2  file   0     rwxrwxrwx
Time elapes 500

$rm test.zip
Time elapes 47

$ls -l
The returned has 2 packets.
Name    Type    Size    Protection
dev  directory   --     rwxrwxrwx
test2  file   0     rwxrwxrwx
Time elapes 500

$cp /c/Temp/test.zip .
The reply has 1 packets.
Now trying to send 0 30
Now trying to send sync
Sync reply shows has got 29
Now trying to send 29 59
Now trying to send sync
No sync reply
Now trying to send 29 59
Now trying to send sync
Now reply is good on sync
Now trying to send 59 89
Now trying to send sync
Now reply is good on sync
Now trying to send 89 106
Now trying to send sync
Now reply is good on sync
cp succeed
Copy finished
Time elapes 13625

$ls -l
The returned has 3 packets.
Name    Type    Size    Protection
dev  directory   --     rwxrwxrwx
test.zip  file   5254     rwxrwxrwx
test2  file   0     rwxrwxrwx
Time elapes 516
```

Other Command Examples

This part illustrates several other command examples for controlling a sensor network. Commands cd, ls, search, ps, pwd, and du are illustrated. Please refer to Appendix I for a list of commands that are supported.

```
$ls
The returned has 4 packets.
dev
test.zip
test2
test3
Time elapes 500

$search test
/sensornet1/nodeA/test.zip
/sensornet1/nodeA/test2
/sensornet1/nodeA/test3
/sensornet1/nodeA/test3/test4
/sensornet1/nodeA/test3/test4/testfile
Time elapes 516

$cd test3
cd command successful
Time elapes 0

$ls
The returned has 1 packets.
test4
Time elapes 500

$cd test4
cd command successful
Time elapes 16

$ls
The returned has 1 packets.
testfile
Time elapes 500

$pwd
/sensornet1/nodeA/test3/test4
Time elapes 0

$du
The reply has 1 packets.
Node nodeA remains 87% of EEPROM and 98% of Flash
```

```
Time elapes 47

$ps
The reply has 1 packets.
sysshell   Active
Time elapes 15
```

Interacting with the sensors

LiteShell also allows the user to directly get sampling readings from sensors. To do this. LiteOS maps sensors that are installed on the nodes into special files. Once one node is formatted, it contains an initial directory called **dev**, which contains such mapped devices. In the current version, four devices, light sensor, temperature sensor, radio, and led, are mapped. We now illustrate how to use them in the following examples.

Using the Light Sensor and the Temperature Sensor

The general syntax for handling the light sensor is as follows:

./light $Number_of_Samplings$ $Sampling_cycle$

For example, the following command reads the light sensor 60 times at a frequency of 500 milliseconds.

./light 60 500

The screenshot for this command is in the next page.

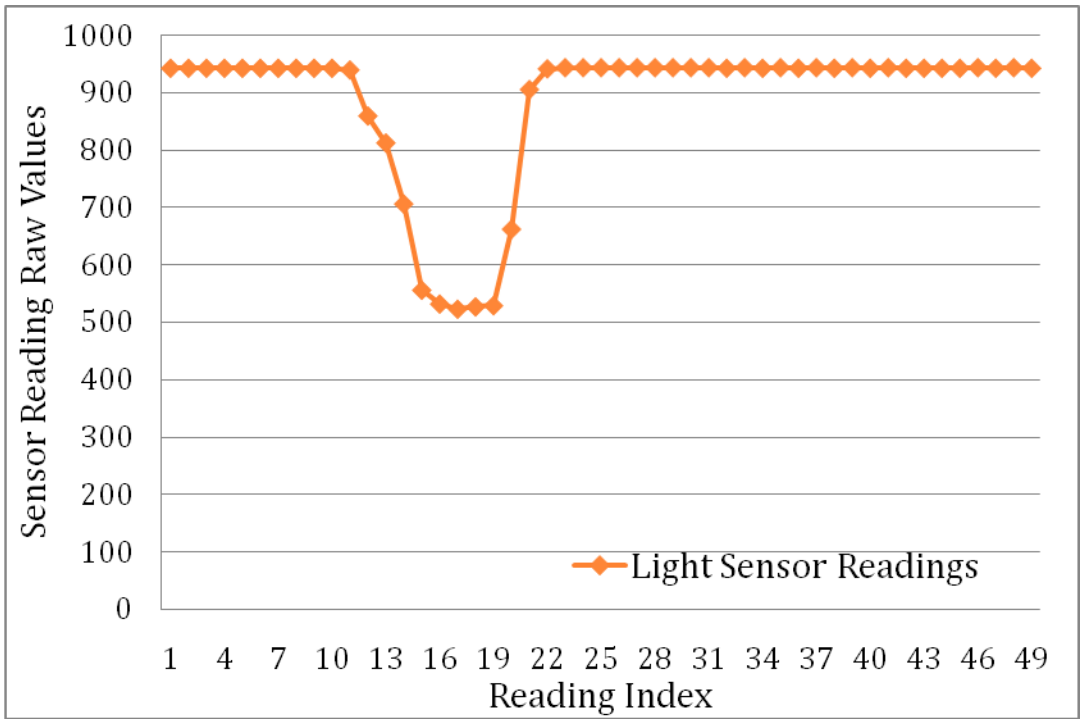
The temperature sensor is similarly used, by replacing ./light with ./temperature.

```
$ls
The returned has 6 packets.
led
light
temp
magnet
accel
radio

Time elapes 515
```

```
$pwd
/sensornet1/nodeB/dev
Time elapes 0

$./light 50 100
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 939
Get reading of 859
Get reading of 812
Get reading of 706
Get reading of 556
Get reading of 532
Get reading of 523
Get reading of 527
Get reading of 529
Get reading of 662
Get reading of 905
Get reading of 941
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 943
Get reading of 942
Get reading of 943
Get reading of 942
Get reading of 943
Get reading of 942
Get reading of 943
Get reading of 942
Get reading of 943
Get reading of 942
Get reading of 943
```

```
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 942
Get reading of 943
Get reading of 942
Get reading of 943
Get reading of 942
Time elapes 5953
```

We can even display the sensor readings easily in graphs. In this example, we manually put a shadow on the sensor twice. Hence, there are clearly fluctuations in the readings over the sixty samplings.



Using the LED

For the Led, use the following command:

./led REDLEDCONTROL, GREENLEDCONTROL, YELLOWLEDCONTROL

Here, LEDCONTROL must be 0, 1, or 2, where:

0 means OFF

1 means ON

2 means TOGGLE

For example, if you invoke

./led 2 2 2

This will toggle all three LED on the node.

Installation of Sample Applications

The following example explains the installation procedure of sample applications, like **Blink**. For simplicity of illustration, we first copy **Blink.lhex** and all the other binaries located under the **Apps** directory to a temporary directory, such as **C:/Temp**. Then, use the **exec (or install)** command to run an application. You may use the **ps** command to verify that the process is indeed running. Finally, you may use the **kill (or uninstall)** command to kill this process.

```
$cp /c/Temp/helloworld.lhex helloworld
The reply has 1 packets.
Now trying to send 0 23
Now trying to send sync
Now reply is good on sync
cp succeed
Copy finished
Time elapes 2547

$ls -l
The returned has 2 packets.

Name    Type    Size    Protection
dev  directory  --     rwxrwxrwx
helloworld  file  1120     rwxrwxrwx
Time elapes 500

$exec helloworld
Process execution succeeds.
Time elapes 828
```

```
$ps
The reply has 2 packets.
sysshell   Active
helloworld   Sleep
Time elapes 313
```

Appendix

Appendix I: A List of LiteShell Commands supported

*The commands with stars are related to debugging and are described in the programmer's guide.

Command Name	Command Semantics
ls	List directory content
Cd	Enter a directory
cp	Copy files
Rm	Delete files or directories
mkdir	Create a new directory
touch	Create a new file
pwd	Display current directory location
du	Display remaining file system space
ps	Display running threads
exec	Execute an application as one or more threads
kill	Kill a thread
man	Show the help of commands
search	Search particular files using names
format	Format the file system
debug*	Set up debugging environment
breakpoint*	Inserts a breakpoint
continue*	Continue with the previous breakpoint
snapshot*	Snapshots a thread state information
restore*	Restore a thread to a previous snapshot state
print*	Print variable values in a running thread
set*	Set variable values in a running thread
setbasemode	Set the base mode to be promiscuous mode or not
rbb	Reset the base station
rbn	Reset the entire sensor network within one hop
history	Print the list of previous commands
setchannel	Change the channel used by the base station
memory	Display memory allocation information of current threads

Appendix II: Man Pages

LS User Commands

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]...

DESCRIPTION

List information about the files (the current directory by default).

-l

List detailed information about files.

CD User Commands

NAME

cd - enter a file directory

SYNOPSIS

cd [FILE DIRECTORY]

DESCRIPTION

Enter a file directory.

CP User Commands

NAME

cp - copy files and directories

SYNOPSIS

cp SOURCE DEST

DESCRIPTION

Copy file between directories. The directories may be located on PC or mounted sensor nodes. To access directories on PC, start the address with PC drive name, such as /c/data.

EXAMPLE

cp /c/Temp/code.lhex /sn01/nodeA/apps/code.lhex

cp /sn01/nodeA/usr/data /c/Temp/data

RM User Commands

NAME

rm - remove a file or a directory

SYNOPSIS

rm FILE/DIRECTORY NAME

DESCRIPTION

Deletes a file or a directory in the file system.

MKDIR User Commands

NAME

mkdir - create a directory

SYNOPSIS

mkdir DIRECTORY NAME

DESCRIPTION

Create a new directory in the current position.

TOUCH User Commands

NAME

touch - creates a new file

SYNOPSIS

touch FILENAME

DESCRIPTION

This command will create a file at the current position and the newly created file is assumed to be empty.

DU User Commands

NAME

du - estimate file system usage

SYNOPSIS

du

DESCRIPTION

Return the current remaining space in EEPROM and Flash

PS User Commands

NAME

ps - view current running processes

SYNOPSIS

ps

DESCRIPTION

Display the current processes and their status.

SEARCH User Commands

NAME

search - finds and locates a file or a directory according to a particular regular expression name

SYNOPSIS

search FILE/DIRECTORY NAME

DESCRIPTION

This command will search across the network by file name and returns a list of files/directories whose names contains the particular file names

MAN User Commands

NAME

Manual, displays the manual pages for particular commands

SYNOPSIS

man COMMAND NAME

DESCRIPTION

This command displays the manual pages of particular commands.

If the manual pages are not available, it will be added in future versions.

EXIT User Commands

NAME

Exit the shell system

SYNOPSIS

exit

DESCRIPTION

This command exits the shell system

PWD User Commands

NAME

Pwd, displays the current working directory

SYNOPSIS

pwd

DESCRIPTION

This command displays the current working directory.

EXEC User Commands

NAME

Exec, executes a user application

SYNOPSIS

exec BINARYNAME.LHEX

DESCRIPTION

This command executes a user application by invoking its binary name.

KILL User Commands

NAME

Kill, terminates a running application

SYNOPSIS

kill BINARYNAME.LHEX

DESCRIPTION

This command terminates a user application by invoking its binary name.

HISTORY User Commands

NAME

History, displays previous commands

SYNOPSIS

history

DESCRIPTION

After you type history, you can use the beginning numbers of each line to re-invoke a

previous command. Such shortcut can be used only once in the current version.

SETCHANNEL User Commands

NAME

setchannel

SYNOPSIS

setchannel Channel Number

DESCRIPTION

Set the channel used by the base station.

Troubleshooting Tips

- Check FAQ on the website for updated answers to common problems.
- If things went wrong, reboot the node.
- The communication between base and node could be severely degraded by all kinds of interference. If one command does not get a reply, try it again.
- Because of unreliable radio, multiple ls –l commands may return different results.
- Connect all cables and battery securely.
- Report bugs in mailing lists so that they can be solved in future releases.
- It takes 1-2 minutes to boot up a node and format itself. Please be patient during this procedure.
- Some environments have strange problems with file permissions when extracting from the tar.gz tar ball. Change file permission to 777 if you meet such problems, i.e., type the following command: chmod 777 *.

Index

CD..... 40
Commands 38
CP 41
Cygwin 3
DU 45
EXEC 51
EXIT 49
HISTORY 53
Installation 2
Installer Express Mode 24
JDK 2, 13
KILL..... 52
Linux Installation 13
LiteShell Commands 26
LS..... 39
MAN..... 48
MIB510 9
MIB520..... 10
MKDIR..... 43
PS..... 46
PWD 50
RM..... 42
SEARCH 47
SETCHANNEL..... 54
TOUCH 44
Troubleshooting Tips 55
uisp 16
USB-to-Serial adaptor..... 11
Windows Installation 2