Lab Activity #1: Exploring TinyOS

Cleveland State University
Electrical and Computer Engineering
Distributed: September 12, 2006

Introduction

Welcome to your first hands-on experience with TinyOS. In this lab you will be browsing the TinyOS installation directory, observing the contents, understanding how they relate to the needs of application developers for wireless sensor networks and finally answering questions based on what you did in this lab session.

On the Windows system the installation directory is \( c:/\text{tinyos/cygwin} \). This is normally referred to as the root directory where all files related to the \textit{cygwin} installation are present.

Cygwin is a Linux-like environment for Windows. It consists of two parts (that are of consequence to us, at least):

1. A DLL (\textit{cygwin1.dll}) which acts as a Linux API emulation layer providing substantial Linux API functionality, and
2. A collection of tools, which provide Linux look and feel.

Exploring the TinyOS Directory Structure

On the Desktop, double click the icon Cygwin which would open a linux like console.

You will see a \$ prompt which is a bash shell. Now you can browse through the directory structure of the TinyOS which contains files and folders for different purposes.

Note :- You can also browse through the directory structure in Windows however you are encouraged to do so in Cygwin since it will provide you practice to type in commands which you would require to compile, run, test and debug TinyOS applications.

TinyOS is installed in \( /\text{opt/tinyos-1.x} \). To move into that directory type in

\$ cd /opt/tinyos-1.x

This directory is referred to as the tinyos root. You can use the environment variable \$TOSROOT to access it easily.

To view the contents of the directory use the \texttt{ls -l} command

\$ ls -l

You should be able to see a bunch of directories and files. We will go into all those directories one by one and you should observe the contents of the directory.

\texttt{apps} Directory

\texttt{cd} into the \texttt{apps} directory and type the \texttt{ls} command.

It contains sample TinyOS applications that have been written in NesC. This folder should be the starting point for all beginners learning to write Wireless Sensor Network applications.
written in NesC. These applications are for you to understand and learn by example. It is the
directory where you should normally place the implementation files of your own applications
as you begin to implement them.

**tos** Directory
To move up in the directory and into the tos directory type in `cd ../tos`. You should be in
the path

`/opt/tinyos-1.x/tos`

This directory contains the shared system components. Type in the `ls` command to see the
list of directories in it.

**interfaces** Directory

`/opt/tinyos-1.x/tos/interfaces`

It contains interface files that components (modules and configurations) provide and use.
The interface file contains commands and events which are implemented by the component
providing the interface.

**lib** Directory

`/opt/tinyos-1.x/tos/lib`

This folder contains library services for applications developers. Using the existing library
services, application developers build better programs quickly and easily.

**platform** Directory

`/opt/tinyos-1.x/tos/platform`

This folder contains the driver files for different hardware platforms that are supported by
TinyOS.

**sensorboards** Directory

`/opt/tinyos-1.x/tos/sensorboards`

This folder contains files required to use with different type of sensors attached to various
sensor boards. These sensor boards are platform dependent.

**system** Directory

`/opt/tinyos-1.x/tos/system`

This folder contains all the system components which TinyOS uses. You can use this folder
to find system components which are required to you to build applications.

Remember you should NOT modify any system components, if you really need to modify
them first copy that file into your application directory and then modify them.

**types** Directory

`/opt/tinyos-1.x/tos/types`

This folder contains the header files which are used by the system components of TinyOS.
tools Directory
/opt/tinyos-1.x/tools
Move up the directory structure and moving into the tools directory by typing $ cd $TOSROOT/tools. This folder contains files which can be used as tools for building, testing and debugging TinyOS applications.

doc Directory
/opt/tinyos-1.x/doc
Move up the directory structure and moving into the doc directory by typing $ cd $TOSROOT/doc. This folder contains the documentation for applications built in wireless sensor networks and system components like hardware components.

contrib Directory
/opt/tinyos-1.x/contrib
Move up the directory structure and moving into the contrib directory by typing the $ cd $TOSROOT/contrib. This directory contains implementations of protocols, applications for sensor networks etc which have been successfully tested and which are contributions of the TinyOS community to help other application developers to use them to build applications which might use them.

Verification of TinyOS Installation
To verify the software installation of TinyOS, use the command $ toscheck in any directory. You should normally see output fly by and at the end toscheck will report whether or not there were errors in the installation.

Answer the following questions
1. Draw the TinyOS directory tree.
2. Write the full path name of the TinyOS installation.
3. If you had to write your own TinyOS application where you would normally put your files? What files are required to compile an application? (Hint: look up sample applications).
4. If you wrote an application which would become part of the TinyOS system, where would you find those files? Should any one modify them? If no/yes, give reasons why?
5. There is an routing protocol or an application which you successfully implemented and tested and now you want to contribute it to the TinyOS community. Where would your contributions appear? Name at least two contributions you saw.
6. Name two tools that you can use to build TinyOS applications and describe how you would use them.
7. Give the full path name where you would find the source code for the `toscheck` command.

8. Where would you look to find the list of different hardware platforms that are supported currently by TinyOS? Name the different platforms that TinyOS supports that you saw.