

# TinyOS Tutorials

From TinyOS Wiki

These brief tutorials are intended to get you started with TinyOS. They show you the basics of writing, compiling, and installing TinyOS applications. They introduce the basic TinyOS abstractions: computation, communication, sensing, and storage. The later tutorials go a little deeper into some of the more advanced areas of TinyOS, such as handling interrupts, power management, and how platforms are organized.

## Contents

- 1 Working Group Tutorials
  - 1.1 **Getting Started with TinyOS**
  - 1.2 **Modules and the TinyOS Execution Model**
  - 1.3 **Mote-mote radio communication**
  - 1.4 **Mote-PC serial communication and SerialForwarder**
  - 1.5 **Sensing**
    - 1.5.1 **ADC**
  - 1.6 **Boot Sequence**
  - 1.7 **Storage**
  - 1.8 **Resource Arbitration and Power Management**
  - 1.9 **Concurrency**
  - 1.10 **Platforms**
  - 1.11 **TOSSIM**
  - 1.12 **Network Protocols**
  - 1.13 **TinyOS Toolchain**
  - 1.14 **Building a simple but full-featured application**
  - 1.15 **The TinyOS printf Library**
  - 1.16 **Writing Low-Power Applications**
  - 1.17 **TOSThreads Tutorial**
  - 1.18 **CC2420 Security Tutorial**
- 2 Other Tutorials
  - 2.1 **Ipsn2009-tutorial**
- 3 User Contributed Tutorials
  - 3.1 **Platform Creation and Testing**
  - 3.2 **Rssi Demo**

## Working Group Tutorials

### ■ Getting Started with TinyOS

Lesson 1 introduces the major concepts of TinyOS: components, modules, configurations and interfaces. It shows you how to compile and install a TinyOS program on a mote.

### ■ Modules and the TinyOS Execution Model

Lesson 2 explains the TinyOS execution model, looking more closely at modules. It explains events, commands and their relationships to interfaces in greater depth, introducing split-phase operations. It explains tasks, the basic mechanism in TinyOS for components to cooperatively share the processor.

## ■ Mote-mote radio communication

Lesson 3 introduces the TinyOS communication model. There is an exercise that illustrates sending and receiving messages.

## ■ Mote-PC serial communication and SerialForwarder

Lesson 4 introduces the the TinyOS toolchain for PCs and laptops to communicate with motes. It describes the concept of a packet source, the mig tool, and SerialForwarder.

## ■ Sensing

Lesson 5 explains how to sample sensors in TinyOS. There is an exercise that periodically samples a sensor and displays the value on the leds.

## ■ ADC

Lesson 5.1 explains how to use lesson 5 to sample any adc in TinyOS. There is an exercise that periodically samples any adc and sends its value via serial port.

## ■ Boot Sequence

Lesson 6 details the boot sequence and, in doing so, answers the question, "But where is main()?".

## ■ Storage

Lesson 7 introduces the TinyOS storage model. Sample applications illustrate the use of the Mount, ConfigStorage, LogRead and LogWrite interfaces.

## ■ Resource Arbitration and Power Management

Lesson 8 introduces the TinyOS resource arbitration and power management model. There are two exercises in this tutorial. The first one illustrates how to gain access to predefined shared resources. The second one illustrates how to create your own shared resource. In both tutorials, the process for controlling the power states of the resource is presented.

## ■ Concurrency

Lesson 9 introduces the TinyOS concurrency model. Tasks are revisited and async code is introduced. This tutorial is currently unfinished.

## ■ Platforms

Lesson 10 provides a better understanding of the difference between "make micaz" and "make telosb," including how these commands map into underlying files, directories, and definitions. It is not necessary for most TinyOS developers, but is included as a starter's guide for people who wish to understand the make system better or wish to design a new platform.

## ■ TOSSIM

Lesson 11 introduces TOSSIM, a TinyOS simulator. TOSSIM allows you to compile your TinyOS applications into a simulation framework, where you can perform reproducible tests and debug your code with standard development tools.

## ■ Network Protocols

### ■ Dissemination and Collection (Old Link)

Lesson 12 introduces two basic multihop protocols, Dissemination and Collection. Dissemination reliably delivers small data items to every node in a network, while collection delivers small data items from every node in a network to designated collection roots.

### ■ Dissemination

This tutorial introduces multihop data dissemination. Dissemination reliably delivers small data items to every node in a network

### ■ Tymo

This lesson introduces another multihop protocol that allows sending data to an addressed node. It is based on the DYMO routing protocol.

### ■ Deluge T2

Deluge is a system for efficiently updating the code image running on a network of motes running TinyOS.

### ■ BLIP Tutorial (2.1.1)

This tutorial explains how to compile, configure, and commission a 6lowpan/IPv6 subnet using blip, the TinyOS IPv6 stack.

### ■ BLIP 2.0 (2.1.2 and svn)

Information on the latest version of the IPv6 stack and associated protocols.

### ■ TinyRPL

This tutorial introduces the IETF RPL IPv6 routing protocol implementation in TinyOS.

## ■ TinyOS Toolchain

Lesson 13 describes the details of the TinyOS toolchain, including the build system, how to create your own Makefile, and how to find out more information on the various tools included with TinyOS.

## ■ Building a simple but full-featured application

Lesson 14 goes through the process of building a simple anti-theft application using many of the features and services of TinyOS 2. Lesson 14 is found in the `tinys-2.x/apps/AntiTheft` directory. The powerpoint slides found there (also available in pdf form) go over the basic principles of TinyOS, and show how to build the accompanying AntiTheft application. Please start by reading the README.txt file in the AntiTheft directory.

To run the AntiTheft demo you will need mica2 or micaz motes, and some mts310 sensor boards (you can also use mts300 boards, though you will lose the movement detection functionality). If you do not have this hardware, the slides and AntiTheft code should still provide a good overview of TinyOS 2.

## ■ The TinyOS printf Library

Lesson 15 describes the details of using the TinyOS printf library to print debug messages to your PC from a TinyOS application running on a mote.

## ■ Writing Low-Power Applications

This lesson demonstrates how to write low power sensing applications in TinyOS. At any given moment, the power consumption of a wireless sensor node is a function of its microcontroller power state, whether the radio, flash, and sensor peripherals are on, and what operations active peripherals are performing. This tutorial shows you how to best utilize the features provided by TinyOS to keep the power consumption of applications that use these devices to a minimum.

## ■ TOSThreads Tutorial

This lesson demonstrates how to use the TOSThreads library to create and manipulate user-level threads in TinyOS. It also walks through the steps of adding TOSThreads support to split-phase services.

## ■ CC2420 Security Tutorial

This lesson introduces and illustrates how the CC2420 in-line security features can be enabled on a step-by-step basis.

# Other Tutorials

## ■ Ipsn2009-tutorial

# User Contributed Tutorials

## ■ Platform Creation and Testing

## ■ Rssi Demo

This lesson describes how to get RSSI readings from incoming packets or for noise floor estimation.

Retrieved from "[http://tinyos.stanford.edu/tinyos-wiki/index.php?title=TinyOS\\_Tutorials&oldid=5643](http://tinyos.stanford.edu/tinyos-wiki/index.php?title=TinyOS_Tutorials&oldid=5643)"

Category: Tutorials

- 
- This page was last modified on 24 October 2011, at 14:00.
  - This page has been accessed 628,511 times.