Preface

The earliest implementation of TINI actually dates back to late 1998 when a handful of engineers at Dallas Semiconductor, working with engineers at Sun Labs, demonstrated a very small, Java programmable device that was capable of controlling household electrical appliances. The prototype modules were crammed into light switch housings, coffee pots, HVAC systems, and fans. The appliances communicated with one another and with a central server, using a crude form of power line networking. The main idea was to provide not only local control of the appliance but also network connectivity to allow for remote control and monitoring. This increased the flexibility as well as the ease of use of the appliance. While none of the engineering work of this ancient version of the technology remains, the concept of a Java programmable runtime environment used to create embedded network applications is still the cornerstone of the TINI platform.

Over the past two years, the power line has given way to Ethernet, and the network programming interface has transitioned from an application specific interface to a standards-based TCP/IP protocol stack. The device I/O capabilities have also been greatly extended. Today, TINI is a broad platform that includes both hardware and software used to create intelligent network devices. These are often devices that require a small footprint, have low power consumption, and are cost sensitive. A few examples include industrial automation equipment, access control, vending machines, remote meters, and environmental sensors.

The TINI development project is a first for Dallas Semiconductor in that its design has been open to public scrutiny. The networking portion of the runtime

environment along with the core Java APIs are of course well defined and well understood by a large development community. However, several new APIs have been created to expose the rich I/O capabilities of the technology. Major contributions to the definition of these new APIs have been made by the TINI SIG (special interest group). The result of this cooperative effort is a feature-rich platform. This work is an attempt at presenting a reasonably complete specification of the platform with plenty of examples to help clarify important topics. The book focuses on the following three areas.

- Platform definition
- Local device I/O APIs
- TCP/IP networking capabilities

Several of the chapters describe the APIs that expose the various forms of device I/O. Some of these may not be required by developers with specific applications in mind. However, the reader is encouraged to read at least the first and last chapters in addition to the chapters that expose capabilities relevant to his or her particular application. The first chapter provides a thorough definition of the platform, while the final chapter focuses on performance improvements and application hardening—two important topics for anyone writing serious applications targeted for the TINI runtime environment. Chapter 7, Building a Remote Data Logger, is also quite useful as it details a large example that brings together several of the concepts presented to that point in the book, including serial communication, 1-Wire networking, and TCP/IP networking over both Ethernet and serial interfaces.

The best way to become familiar with this technology is, of course, to use it. For this reason, every attempt has been made to create examples that are easily run on the most commonly available hardware. Some of the larger examples require additional hardware, but any additional hardware should be relatively inexpensive and easy to attain.

A strong familiarity with the Java programming language and some experience with network programming concepts is assumed. While a comfort level with hardware-related topics is helpful, it is not a requirement for understanding the bulk of the contents of this book. It is my hope that “pure programmers” can start with the code examples and gradually become more comfortable with the hardware-oriented concepts presented here.

ACKNOWLEDGMENTS

I would like to thank the many people who have contributed to the TINI project and this book. First and foremost I would like to thank the talented engineers who contributed so much to this long and intense development effort for their hard work and dedication: Kris Ardis, Bryan Armstrong, Tom Chenot, Chris Fox, Stephen Hess, Nicolas Kral, Yolanda Lei, Jesse Marroquin, Caroline McLean, Jeff Owens, David Smiczek, Lorne Smith, Stephen Umfleet, and Clayton Ware. I would also like to thank my management, Steve Curry and Michael Bolan, for their support and encouragement while I was writing this book. I am grateful for the volunteer efforts of many on the TINI SIG, who not only provide fantastic support to new developers but also contribute to the quality and definition of the platform.

Thorough and insightful technical reviews of early drafts were provided by Tom Cargill, Steve Curry, Peter Haggar, Judy Loomis, Robert Muchsel, and John Wilson. I appreciate all of the excellent feedback.

I am also grateful to Mike Hendrickson and Heather Olszyk at Addison-Wesley, who patiently guided me through the writing process. I would also like to thank the copy editor, Debbie Prato, who did a terrific job.

Finally, many thanks to the folks at Sun Microsystems who allowed me to use their excellent MIF Doclet tool to create the Almanac. The legend page of the Almanac is also the result of blatant thievery from the Java Real-Time specification.