

Embedded Systems Design for High-Speed Data Acquisition and Control

Maurizio Di Paolo Emilio

Embedded Systems Design for High-Speed Data Acquisition and Control

 Springer

Maurizio Di Paolo Emilio
Pescara
Italy

ISBN 978-3-319-06864-0 ISBN 978-3-319-06865-7 (eBook)
DOI 10.1007/978-3-319-06865-7

Library of Congress Control Number: 2014946192

Springer Cham Heidelberg New York Dordrecht London

© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Imagination is more important than knowledge.

A. Einstein

When wireless is perfectly applied, the whole earth will be converted into a huge brain, capable of response in every one of its parts.

N. Tesla

To Julia, Elisa and Federico

Forewords

It's a great pleasure for me to write the Preface of a book which aims to help to implement one of the oldest technologies (data acquisition) using one of the concepts (embedded system) that will reshape the future of the world. The first known acquisition system used was by Egyptian pharaohs in 2500 BC to record the water flow of the Nile to predict flood. From this ancestor engraving down values on stone, with an imaginary time travel, we arrive in 1963: in that year was invented the first data acquisition systems. In the past, data acquisition systems were largely mechanical, using smoked drums or chart recorders. Today, powerful microprocessor performs data acquisition faster, more accurately, more flexibly, with more sensors, more complex data processing and elaborate presentation of the final information. Data acquisition systems can now be used as application demand, wired or wireless. Mobile devices is another area which has grown in adoption and most of them have many kinds of data acquisition systems built in. Now the data acquisition industry may be at the beginning of another revolutionary data acquisition change with the advancements of the cloud computing on the Internet. Benefits of cloud computing include real-time access of information, scalability of processing power, less risk of downtime and direct interaction with other web services. The embedded systems industry, on the other hand, is very recent: it was born with the invention of microcontrollers and since then it has evolved into various forms, from primarily being designed for machine control applications to various other new verticals with the convergence of communications. The importance of embedded systems is growing continuously. Exponentially increasing computing power (Moore's law), ubiquitous connectivity and convergence of technology have resulted in hardware/software systems being embedded within everyday products and places. In fact, 98 % of all computing devices manufactured today go into embedded systems. By definition (I like the definition given by Edward Lee—Berkeley—of Cps systems, integration of computation with physical processes) embedded systems are an optimized mix of hardware and software. So another key word is codesign. In synthesis, the reach of system-level objectives by exploiting the trade-offs between hardware and software in a system through concurrent design. Embedded developers must have deep insight into both the

hardware and software sides of the design. This is the goal, and the usefulness, of Maurizio's book: provide a solid groundwork for all embedded developers—and for all the people approaching this exciting world—involved in the “codesign” of data acquisition systems in sync with the existing and future technology, in both hardware and software.

Milan, June 2014

Filippo Fossati

Technology marches on, almost relentlessly, as every aspect of our lives demands newer, smaller, more feature-rich and comfort-and convenience-driven electronics at an unabated pace. At the heart of all these devices, however small, are embedded systems. It has been said that one of the very first recognizably modern embedded systems was the Apollo Guidance Computer, developed by the MIT's Labs. But, however technically impressive, this guidance computer might have been used and its use of the then newly developed integrated circuits (ICs) to reduce the system's size and weight had sent shivers down many a space scientists' spines not to mention those footing the bill for the mission; the guidance computer was considered the riskiest item in the Apollo project. This granddaddy of embedded systems did humankind proud: not only did it go to space, but also it duly verified the use of ICs into electronics systems, however mission critical they may be. Since then, the use of embedded systems has skyrocketed too. Such systems are found in just about every electronics system we can think of. But, to keep embedded systems in tune with all the other developments that go hand in hand, we—the editors, authors, engineers, lecturers, etc.—should continue keeping all involved parties up-to-date. Maurizio is an author who has dedicated his efforts to learning and working with embedded systems, as well as data acquisition and control systems, and has translated that knowledge and experience into information and education for the practicing and learning engineers alike. His work can be regularly read about on the pages of *Electronics World* magazine to which he has been a regular technical contributor—with his own column—for many years. Topics he regularly covers in the magazine range from basic electronics through data acquisition and control systems to embedded systems design. As embedded systems permeate every sphere of our lives, nowadays they are also found in data acquisitions systems. Data acquisition has its own historical record too—its early beginnings can be traced back to 1963, when IBM produced computers that specialized in data acquisition. Among the earliest models were the IBM 7700 and IBM 1800 Data Acquisition and Control Systems, which were rather expensive. Later, these systems were superseded by the general-purpose computer and specialist data acquisitions cards. Today data acquisition systems are much smaller and less expensive than their first versions, and also rely on embedded systems for their successful operation. As such, Maurizio is well placed to prepare this book and present his knowledge to an

even wider audience. So, electronics can continue on its inventive path, launching new technologies and rather revolutionary systems and devices at a high speed, as long as we have dedicated individuals such as Maurizio teaching us about them, we are in good hands to learn more and create more. Electronics is a fabulously creative discipline, and all of us involved in this field feel truly blessed to be a part of it.

London, June 2014

Svetlana Josifovska

Preface

Embedded systems are a combination of hardware and software, which facilitates mass production and a host of applications, benefiting from economies of scale. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose. Some applications of embedded system are in the automotive field as control system and in the data acquisition for scientific and industrial fields (control system).

Modern industrial plants utilize robots for obtaining temperature controls, pressure controls, speed controls and position controls, and so on. Feedback control system is to be found in almost every aspect of our daily environment.

Data Acquisition Systems (DAQ) are now the main instruments for testing, measuring in automation field and so on, which is used widely in the research laboratory by scientists and engineers.

Data acquisition is a necessity, which is why data acquisition systems and software applications are essential tools in a variety of fields. For instance, research scientists rely on data acquisition tools for testing and measuring their laboratory-based projects. Therefore, as a data acquisition system designer, you must have in-depth understanding of each part of the systems and programmes you create.

The complexity of new physics experiments and industrial processes require more complex DAQ with the following characteristics:

- Capable of managing large amounts of data.
- High-speed connection.
- Digital recording.
- Full reconfigure possibility.

Signals that are hard to characterize and analyse with real-time display are evaluated in terms of the following parameters:

- High frequency.
- Large dynamic range.
- Gradual changes.

Data acquisition software is typically available in a text-based user interface (TUI) that comprises an ASCII configuration file and a graphic user interface (GUI), which are generally available with any web browser. Both interfaces enable data acquisition system management and customization, do not need to recompile the sources. This means even inexperienced programmers can have full acquisition control.

Well-designed data acquisition and control software should be able to quickly recover from instrumentation failures and power outages without losing any data. Data acquisition software must provide a high-level language for algorithm design. Moreover, it requires data archiving capability for verifying data integrity.

You have many data acquisition software options. An example is programmable software that uses a language such as C. Other software and data acquisition software packages enable you to design the custom instrumentation suited for specific applications (e.g., National Instruments's LabVIEW and MathWorks's MATLAB).

A data acquisition system's complexity tends to increase with the number of physical properties it must measure. Resolution and accuracy requirements also affect a system's complexity. To eliminate cabling and provide for more modularity, you can combine data acquisition capabilities and signal conditioning in one device.

Recent developments in the field of fiber-optic communications have shown longer data acquisition transmission distances can cause errors. Electrical isolation is also an important topic. The goal is to eliminate ground loops (common problems with single-ended measurements) in terms of accuracy and protection from voltage spikes.

Recently, some new technological developments have proven to be beneficial to the overall efficacy of data acquisition applications. For instance, in USB flash drive successfully makes the data acquisition and storage simpler and more efficient than ever (think "plug and play") and wireless improves the speed of data transmission and security.

In future, consumers' demand for mobile computing systems will only increase, and this will require tablet computers to feature improved data acquisition and storage capabilities. Having the ability to transmit, receive and store larger amounts of data with tablets will become increasingly important to consumers as time goes on. There are three main things to consider when creating a data acquisition-related application for a tablet. Hardware connectivity: Tablets have few control options (e.g., Wi-Fi and Bluetooth). Program language support: Many tablets support Android apps created in Java. Device driver availability: Device drivers permit a high-level mode to easily and reliably execute a data acquisition board's functionality. C and LabVIEW are not supported by Android or Apple's iOS. USB, a common DAQ bus, is available in a set of tablets. In the other case, an adapter is required. Among these examples, moving a possible data acquisition system to a tablet requires extra attention.

For all of the aforementioned reasons that embedded system will figure prominently in the evolution of acquisition system technology makes them ideal for custom data acquisition systems and control system.

The limited function required of embedded systems allows them to be designed for the most efficient performance.

Such embedded computing and information technologies have become, at the same time, an enabler for future manufacturing enterprises as well as a transformer of organizations and markets.

Digital embedded security is no more an option but a necessity as it is critical for more transactions happening over embedded devices as front ends. Due to constrained resources on systems, embedded systems have challenges in implementation on full-fledged security systems; therefore, the concept of “embedded security” offers a new differentiator for embedded product marketing.

The main idea of this book is to describe the theory of the embedded system with the realization of a versatile project (hardware and software) for application as high-speed data acquisition and programmable control system.

Starting from the review of analogue and digital electronics, the book aims to provide the reader into a competent and independent practitioner in the field of embedded systems by providing several skills, in both hardware and software development.

On the hardware side, the book will focus on among microcontroller design, techniques of embedded design, high-speed data acquisition (DAQ) and control system. This culminates in the study and application of a Real-Time Operating System (Open Source), representing the most important way that an embedded system can be programmed. It is presented as a useful tool for embedded designers. Every concept has been made to present the many complex concepts in a way that is easy to understand and which makes them readily usable. Embedded Linux (both the free and licensed versions) remains an attractive choice for a range of development teams and its use is poised to see a manifold increase.

In the chapter of microcontroller design, techniques of design of FPGA will also be presented. FPGA designs combine multiple components into a single package that reduces component count, board size and manufacturing complexity. Processors, memory, custom logic and many of the peripherals in a typical embedded project can be found in the FPGA. Today’s FPGA architecture has grown into billions of logic blocks (equivalent to gates), and with programmable interconnection flexibility designers can easily create hardware functions that exactly match the needs of a specific embedded application.

Moreover, embedded development system and PCB techniques will be presented. An embedded system is identified as the electronic device designed for a particular function.

The design of embedded system makes use of compilers, assembler, debugger and a whole range of suites for the development of both software and hardware.

PCB layout is one of the last steps but the most critical in the design process. High-speed circuit performance is heavily dependent on layout. A high-performance design can be rendered useless due to a poor or sloppy layout.

For a long time, embedded devices were mostly operating as stand-alone systems. However, with the advent of wireless connectivity, like Bluetooth, Zigbee, RFID, the scenario has changed. The recent trends in wireless for use in embedded systems are in the areas of system-on-chip (SoC) architecture, reduced power consumption and application of short-range protocols.

In future, security in the embedded devices will be a critical issue. The security requirements of the connected embedded devices are distinct according to their limited memory, constrained middleware and low computing power. Today, power consumption is still a key issue in the design of the embedded systems that directly affects the battery life, which the technology has not been able to match the advancements in the hardware that drives these systems in recent years.

Pescara, Italy, June 2014

Maurizio Di Paolo Emilio

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this book. In particular, I would like to thank Charles B. Glaser Executive Editor, Applied Sciences USA, for the publication of the present book. Many thanks to Filippo Fossati, Editor-in-Chief of Fiera Milano Media Electronic Division and Svetlana Josivofska, Editor of Electronics World, for the contribution of the Foreword. I want to thank my family for their patience, and also for encouraging and inspiring me to follow my dreams. I am especially grateful to my wife, Julia, for her support in terms of linguistic form and also to Albert Einstein and Nikolas Tesla, because without them today's innovative world would not exist.

Contents

1	Review of Microelectronics	1
1.1	Introduction	1
1.2	Basic of Semiconductor's Physics	1
1.2.1	PN Junction	3
1.3	Diode	5
1.4	Bipolar Transistor: Emitter Follower	7
1.5	MOS Transistor	11
1.6	Differential Amplifiers	14
1.7	Feedback	16
1.7.1	Effects of Feedback	17
1.7.2	PID Controller	18
1.8	Digital CMOS Circuits	18
1.8.1	CMOS Inverter	19
1.8.2	Example of Circuits	20
1.9	Current Mirror	22
1.9.1	Ideal Current Mirror	22
1.9.2	Current Mirror BJT/MOS	23
	References	24
2	Features of Embedded System	25
2.1	The Components of Embedded System	25
2.1.1	Processor	25
2.1.2	Memory	25
2.1.3	System Clock	26
2.1.4	Peripherals	26
2.2	Characteristics and Example of Embedded System	26
2.3	Hardware and Software Design	29
	References	31

- 3 Microcontroller Design** 33
 - 3.1 Introduction 33
 - 3.2 CPU 35
 - 3.3 Memory 37
 - 3.4 Devices 37
 - 3.4.1 I/O Devices 38
 - 3.5 Power Saving 39
 - 3.6 Instructions 39
 - 3.6.1 Enforcement of Instructions 40
 - 3.7 ARM Architecture 41
 - 3.8 DSP Microprocessor 43
 - 3.8.1 Evaluation Parameters for a DSP 44
 - 3.8.2 Commercial DSP 45
 - 3.9 Microcontroller as Embedded System 46
 - 3.10 FPGA 46
 - References 48

- 4 Design Techniques of Embedded System** 49
 - 4.1 Design 49
 - 4.2 The Waterfall Model 51
 - 4.3 Model V 53
 - 4.4 Architecture 55
 - 4.4.1 ASIC-ASSP 56
 - 4.5 Software 60
 - 4.6 Embedded Linux, Windows, Android 63
 - 4.6.1 Windows Embedded Compact 64
 - 4.6.2 Embedded Linux 64
 - 4.6.3 Embedded Android 65
 - 4.7 Power Management 67
 - 4.7.1 Dynamic Power Management 68
 - 4.7.2 Dynamic Voltage Scaling 69
 - 4.7.3 Latencies 70
 - 4.8 Bus Interface 70
 - 4.8.1 USB and FireWire 72
 - 4.8.2 Standarization and Technical Details of USB Bus 72
 - 4.8.3 Serial Communications 76
 - 4.8.4 Wireless, Ethernet and Bluetooth 77
 - 4.8.5 GSM for Embedded System 79
 - 4.8.6 PCI and PCI Express 80
 - 4.8.7 Compact PCI 81
 - 4.8.8 Zigbee and RFID 85
 - 4.9 Memory 89
 - 4.9.1 Memory Flash 90
 - References 92

- 5 Embedded Development System and C Programming** 93
 - 5.1 Development System 93
 - 5.2 C Programming 94
 - 5.2.1 From Assembly Language to C Language 95
 - 5.2.2 Choose the Right C Compiler 96
 - 5.2.3 ANSI C or C++ 96
 - 5.3 Code Warrior IDE 97
 - 5.4 Commercial Software 99
 - 5.4.1 Labview Embedded 99
 - 5.4.2 Intel Studio 99
 - 5.4.3 Altera 99
 - 5.4.4 IAR Embedded Workbench 99
 - References 100

- 6 Real Time Operating System (RTOS)** 101
 - 6.1 Operating System 101
 - 6.1.1 Classification of Operating Systems 103
 - 6.2 Real Time Software 104
 - 6.3 Examples of Real-Time Embedded Systems 106
 - 6.4 Scheduling 108
 - 6.5 Scheduler Based on Deadline 109
 - 6.6 RTOS for Multicore 109
 - 6.7 RTOS and Application Specific Processors 110
 - 6.8 An RTOS for Complex Systems 110
 - 6.9 An RTOS Customizable 111
 - 6.10 Interrupt 111
 - 6.10.1 Classification 112
 - 6.10.2 Management of Interrupt 114
 - 6.11 Linux 114
 - 6.11.1 Problems 115
 - 6.11.2 Real Time Linux 116
 - References 117

- 7 Design PCB for Embedded System** 119
 - 7.1 Materials for Printed Circuits 119
 - 7.2 Electrical Insulation on PCB 123
 - 7.3 Routing PCB 125
 - 7.4 PCB Embedded 126
 - 7.4.1 Design Guidelines 128
 - References 129

- 8 Features of High Speed Data Acquisition and Control System . . .** 131
 - 8.1 Data Acquisition System 131
 - 8.1.1 Data Acquisition Hardware 133
 - 8.1.2 Data Acquisition Software 135
 - 8.2 High Speed PCB Layout 137
 - 8.2.1 Power Supply Bypassing 141
 - 8.2.2 Stray Capacitance 142
 - 8.3 Feedback Control System 142
 - References 145

- 9 Embedded Board for High-Speed Data Acquisition and Control System** 147
 - 9.1 General Layout 147
 - 9.2 Hardware 147
 - 9.3 Software and GUI 150
 - 9.4 Real Time Software 150
 - 9.5 Future and Improvement 152
 - References 152

- Index** 153