

# DIGITAL SIGNAL PROCESSING

Using the ARM<sup>®</sup> Cortex<sup>®</sup>-M4

Donald S. Reay



WILEY

# DIGITAL SIGNAL PROCESSING USING THE ARM<sup>®</sup> CORTEX<sup>®</sup>-M4

---

**DONALD S. REAY**

Heriot-Watt University

**WILEY**

Copyright © 2016 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey  
Published simultaneously in Canada

ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. The MathWorks Publisher Logo identifies books that contain MATLAB® content. Used with Permission. The book's or downloadable software's use of discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by the MathWorks of a particular use of the MATLAB® software or related products.

For MATLAB® product information, or information on other related products, please contact:

The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098 USA, Tel: 508-647-7000,  
Fax: 508-647-7001, E-mail: [info@mathworks.com](mailto:info@mathworks.com), Web: [www.mathworks.com](http://www.mathworks.com), How to buy:  
[www.mathworks.com/store](http://www.mathworks.com/store)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

**Library of Congress Cataloging-in-Publication Data:**

Reay, Donald (Donald S.), author.

Digital signal processing using the ARM Cortex-M4 / Donald Reay.  
pages cm

Includes bibliographical references and index.

ISBN 978-1-118-85904-9 (pbk.)

1. Signal processing—Digital techniques. 2. ARM microprocessors. I. Title.

TK5102.9.R4326 2015

621.382'2—dc23

2015024771

Typeset in 10/12pt TimesLTStd by SPi Global, Chennai, India

Printed in Singapore by Markono Print Media Pte Ltd

10 9 8 7 6 5 4 3 2 1

1 2016

**DIGITAL SIGNAL  
PROCESSING USING THE  
ARM<sup>®</sup> CORTEX<sup>®</sup>-M4**



*To Reiko*



# PREFACE

This book continues the series started in 1990 by Rulph Chassaing and Darrell Horning's *Digital Signal Processing with the TMS320C25*, which tracked the development of successive generations of digital signal processors by Texas Instruments. More specifically, each book in the series up until now has complemented a different inexpensive DSP development kit promoted for teaching purposes by the Texas Instruments University Program. A consistent theme in the books has been the provision of a large number of simple example programs illustrating DSP concepts in real time, in an electrical engineering laboratory setting.

It was Rulph Chassaing's belief, and this author continues to believe, that hands-on teaching of DSP, using hardware development kits and laboratory test equipment to process analog audio frequency signals, is a valuable and effective way of reinforcing the theory taught in lectures.

The contents of the books, insofar as they concern fundamental concepts of digital signal processing such as analog-to-digital and digital-to-analog conversion, finite impulse response (FIR) and infinite impulse response (IIR) filtering, the Fourier transform, and adaptive filtering, have changed little. Every academic year brings another cohort of students wanting to study this material. However, each book has featured a different DSP development kit.

In 2013, Robert Owen suggested to me that hands-on DSP teaching could be implemented using an inexpensive ARM<sup>®</sup> Cortex-M4<sup>®</sup> microcontroller. I pointed out that a Texas Instruments C674x processor was very significantly more computationally powerful than an ARM Cortex-M4. But I also went ahead and purchased a Texas Instruments Stellaris LaunchPad. I constructed an audio interface using a Wolfson WM8731 codec and successfully ported the program examples from my previous book to that hardware platform.



This book is aimed at senior undergraduate and postgraduate electrical engineering students who have some knowledge of C programming and linear systems theory, but it is intended, and hoped, that it may serve as a useful resource for anyone involved in teaching or learning DSP and as a starting point for teaching or learning more.

I am grateful to Robert Owen for first making me aware of the ARM Cortex-M4; to Khaled Benkrid at the ARM University Program and to the Royal Academy of Engineering for making possible a six-month Industrial Secondment to ARM during which teaching materials for the STM32f01 platform were developed; to Gordon McLeod and Scott Hendry at Wolfson Microelectronics for their help in getting the Wolfson Pi audio card to work with the STM32f01 Discovery; to Sean Hong, Karthik Shivashankar, and Robert Iannello at ARM for all their help; to Joan Teixidor Buixeda for helping to debug the program examples; to Cathy Wicks at the TI University Program and Hieu Duong at CircuitCo for developing the audio booster pack; and to Kari Capone and Brett Kurzman at Wiley for their patience. But above all, I thank Rulph Chassaing for inspiring me to get involved in teaching hands-on DSP.

DONALD S. REAY

*Edinburgh*  
2015

# CONTENTS

<b>Preface</b>	<b>xi</b>
<b>1 ARM® CORTEX® - M4 Development Systems</b>	<b>1</b>
1.1 Introduction, 1	
1.1.1 Audio Interfaces, 2	
1.1.2 Texas Instruments TM4C123 LaunchPad and STM32F407 Discovery Development Kits, 2	
1.1.3 Hardware and Software Tools, 6	
Reference, 7	
<b>2 Analog Input and Output</b>	<b>9</b>
2.1 Introduction, 9	
2.1.1 Sampling, Reconstruction, and Aliasing, 9	
2.2 TLV320AIC3104 (AIC3104) Stereo Codec for Audio Input and Output, 10	
2.3 WM5102 Audio Hub Codec for Audio Input and Output, 12	
2.4 Programming Examples, 12	
2.5 Real-Time Input and Output Using Polling, Interrupts, and Direct Memory Access (DMA), 12	
2.5.1 I2S Emulation on the TM4C123, 15	
2.5.2 Program Operation, 15	
2.5.3 Running the Program, 16	
2.5.4 Changing the Input Connection to LINE IN, 16	
2.5.5 Changing the Sampling Frequency, 16	

- 2.5.6 Using the Digital MEMS Microphone on the Wolfson Audio Card, 20
- 2.5.7 Running the Program, 21
- 2.5.8 Running the Program, 23
- 2.5.9 DMA in the TM4C123 Processor, 26
- 2.5.10 Running the Program, 30
- 2.5.11 Monitoring Program Execution, 30
- 2.5.12 Measuring the Delay Introduced by DMA-Based I/O, 30
- 2.5.13 DMA in the STM32F407 Processor, 34
- 2.5.14 Running the Program, 35
- 2.5.15 Measuring the Delay Introduced by DMA-Based I/O, 35
- 2.5.16 Running the Program, 46
- 2.6 Real-Time Waveform Generation, 46
  - 2.6.1 Running the Program, 49
  - 2.6.2 Out-of-Band Noise in the Output of the AIC3104 Codec (`tm4c123_sine48_intr.c`), 49
  - 2.6.3 Running the Program, 53
  - 2.6.4 Running the Program, 62
  - 2.6.5 Running the Program, 69
- 2.7 Identifying the Frequency Response of the DAC Using Pseudorandom Noise, 70
  - 2.7.1 Programmable De-Emphasis in the AIC3104 Codec, 72
  - 2.7.2 Programmable Digital Effects Filters in the AIC3104 Codec, 72
- 2.8 Aliasing, 78
  - 2.8.1 Running the Program, 83
- 2.9 Identifying the Frequency Response of the DAC Using An Adaptive Filter, 83
  - 2.9.1 Running the Program, 84
- 2.10 Analog Output Using the STM32F407'S 12-BIT DAC, 91
  - References, 96

### 3 Finite Impulse Response Filters

97

- 3.1 Introduction to Digital Filters, 97
  - 3.1.1 The FIR Filter, 97
  - 3.1.2 Introduction to the  $z$ -Transform, 99
  - 3.1.3 Definition of the  $z$ -Transform, 100
  - 3.1.4 Properties of the  $z$ -Transform, 108
  - 3.1.5  $z$ -Transfer Functions, 111
  - 3.1.6 Mapping from the  $s$ -Plane to the  $z$ -Plane, 111
  - 3.1.7 Difference Equations, 112
  - 3.1.8 Frequency Response and the  $z$ -Transform, 113
  - 3.1.9 The Inverse  $z$ -Transform, 114
- 3.2 Ideal Filter Response Classifications: LP, HP, BP, BS, 114
  - 3.2.1 Window Method of FIR Filter Design, 114

3.2.2	Window Functions, 116	
3.2.3	Design of Ideal High-Pass, Band-Pass, and Band-Stop FIR Filters Using the Window Method, 120	
3.3	Programming Examples, 123	
3.3.1	Altering the Coefficients of the Moving Average Filter, 132	
3.3.2	Generating FIR Filter Coefficient Header Files Using MATLAB, 137	
<b>4</b>	<b>Infinite Impulse Response Filters</b>	<b>163</b>
4.1	Introduction, 163	
4.2	IIR Filter Structures, 164	
4.2.1	Direct Form I Structure, 164	
4.2.2	Direct Form II Structure, 165	
4.2.3	Direct Form II Transpose, 166	
4.2.4	Cascade Structure, 168	
4.2.5	Parallel Form Structure, 169	
4.3	Impulse Invariance, 171	
4.4	Bilinear Transformation, 171	
4.4.1	Bilinear Transform Design Procedure, 172	
4.5	Programming Examples, 173	
4.5.1	Design of a Simple IIR Low-Pass Filter, 173	
	Reference, 216	
<b>5</b>	<b>Fast Fourier Transform</b>	<b>217</b>
5.1	Introduction, 217	
5.2	Development of the FFT Algorithm with RADIX-2, 218	
5.3	Decimation-in-Frequency FFT Algorithm with RADIX-2, 219	
5.4	Decimation-in-Time FFT Algorithm with RADIX-2, 222	
5.4.1	Reordered Sequences in the Radix-2 FFT and Bit-Reversed Addressing, 224	
5.5	Decimation-in-Frequency FFT Algorithm with RADIX-4, 226	
5.6	Inverse Fast Fourier Transform, 227	
5.7	Programming Examples, 228	
5.7.1	Twiddle Factors, 233	
5.8	Frame- or Block-Based Programming, 239	
5.8.1	Running the Program, 242	
5.8.2	Spectral Leakage, 244	
5.9	Fast Convolution, 252	
5.9.1	Running the Program, 256	
5.9.2	Execution Time of Fast Convolution Method of FIR Filter Implementation, 256	
	Reference, 261	

<b>6 Adaptive Filters</b>	<b>263</b>
6.1 Introduction, 263	
6.2 Adaptive Filter Configurations, 264	
6.2.1 Adaptive Prediction, 264	
6.2.2 System Identification or Direct Modeling, 265	
6.2.3 Noise Cancellation, 265	
6.2.4 Equalization, 266	
6.3 Performance Function, 267	
6.3.1 Visualizing the Performance Function, 269	
6.4 Searching for the Minimum, 270	
6.5 Least Mean Squares Algorithm, 270	
6.5.1 LMS Variants, 272	
6.5.2 Normalized LMS Algorithm, 272	
6.6 Programming Examples, 273	
6.6.1 Using CMSIS DSP Function <code>arm_lms_f32()</code> , 280	
<b>Index</b>	<b>299</b>

---

# 1

---

## ARM<sup>®</sup> CORTEX<sup>®</sup>-M4 DEVELOPMENT SYSTEMS

### 1.1 INTRODUCTION

Traditionally, real-time digital signal processing (DSP) has been implemented using specialized and relatively expensive hardware, for example, digital signal processors or field-programmable gate arrays (FPGAs). The ARM<sup>®</sup> Cortex<sup>®</sup>-M4 processor makes it possible to process audio in real time (for teaching purposes, at least) using significantly less expensive, and simpler, microcontrollers.

The ARM Cortex-M4 is a 32-bit microcontroller. Essentially, it is an ARM Cortex-M3 microcontroller that has been enhanced by the addition of DSP and single instruction multiple data (SIMD) instructions and (optionally) a hardware floating-point unit (FPU). Although its computational power is a fraction of that of a floating-point digital signal processor, for example, the Texas Instruments C674x, it is quite capable of implementing DSP algorithms, for example, FIR and IIR filters and fast Fourier transforms for audio signals in real-time.

A number of semiconductor manufacturers have developed microcontrollers that are based on the ARM Cortex-M4 processor and that incorporate proprietary peripheral interfaces and other IP blocks. Many of these semiconductor manufacturers make available very-low-cost evaluation boards for their ARM Cortex-M4 microcontrollers. Implementing real-time audio frequency example programs on these platforms, rather than on more conventional DSP development kits, constitutes a reduction of an order of magnitude in the hardware cost of implementing hands-on

DSP teaching. For the first time, students might realistically be expected to own a hardware platform that is useful not only for general microcontroller/microprocessor programming and interfacing activities but also for implementation of real-time DSP.

### 1.1.1 Audio Interfaces

At the time that the program examples presented in this book were being developed, there were no commercially available low-cost ARM Cortex-M4 development boards that incorporated high-quality audio input and output. The STMicroelectronics STM32F407 Discovery board features a high-quality audio digital-to-analog converter (DAC) but not a corresponding analog-to-digital converter (ADC). Many ARM Cortex-M4 devices, including both the STMicroelectronics STM32F407 and the Texas Instruments TM4C123, feature multichannel instrumentation-quality ADCs. But without additional external circuitry, these are not suitable for the applications discussed in this book.

The examples in this book require the addition (to an inexpensive ARM Cortex-M4 development board) of an (inexpensive) audio interface.

In the case of the STMicroelectronics STM32F407 Discovery board and of the Texas Instruments TM4C123 LaunchPad, compatible and inexpensive audio interfaces are provided by the Wolfson Pi audio card and the CircuitCo audio booster pack, respectively. The low-level interfacing details and the precise performance characteristics and extra features of the two audio interfaces are subtly different. However, each facilitates the input and output of high-quality audio signals to and from an ARM Cortex-M4 processor on which DSP algorithms may be implemented.

Almost all of the program examples presented in the subsequent chapters of this book are provided, in only very slightly different form, for both the STM32F407 Discovery and the TM4C123 LaunchPad, on the partner website <http://www.wiley.com/go/Reay/ARMcortexM4>.

However, in most cases, program examples are described in detail, and program listings are presented, only for one or other hardware platform. Notable exceptions are that, in Chapter 2, low-level i/o mechanisms (implemented slightly differently in the two devices) are described in detail for both hardware platforms and that a handful of example programs use features unique to one or other processor/audio interface.

This book does not describe the internal architecture or features of the ARM Cortex-M4 processor in detail. An excellent text on that subject, including details of its DSP-related capabilities, is *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors* by Yiu [1].

### 1.1.2 Texas Instruments TM4C123 LaunchPad and STM32F407 Discovery Development Kits

The Texas Instruments and STMicroelectronics ARM Cortex-M4 processor boards used in this book are shown in Figures 1.1 and 1.2. The program examples presented in this book assume the use of the *Keil MDK-ARM* development environment, which is compatible with both development kits. An alternative development environment,

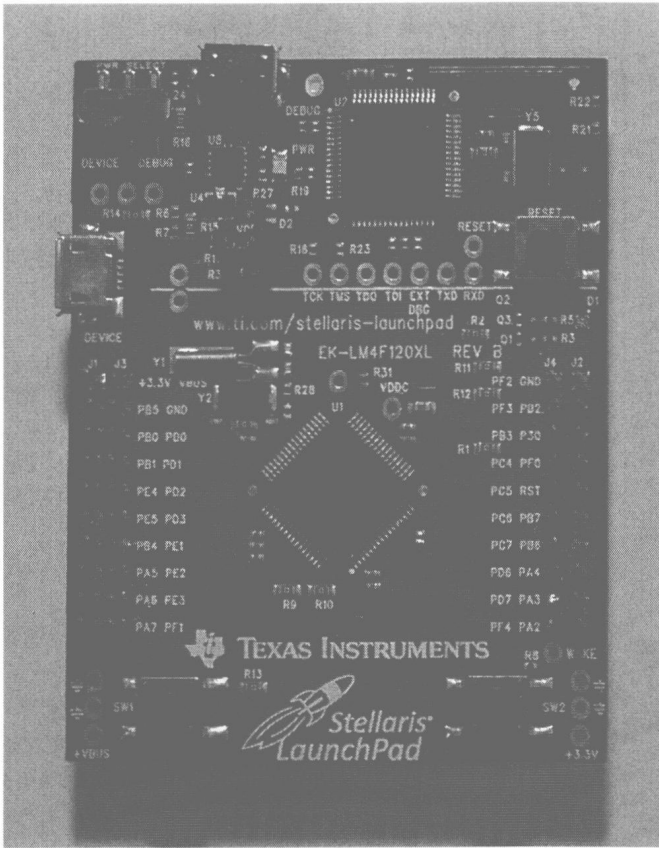


Figure 1.1 Texas Instruments TM4C123 LaunchPad.

Texas Instruments' *Code Composer Studio*, is available for the TM4C123 LaunchPad and the program examples have been tested using this. Versions of the program examples compatible with *Code Composer Studio version 6* are provided on the partner website <http://www.wiley.com/go/Reay/ARMCortexM4>.

The CircuitCo audio booster pack (for the TM4C123 LaunchPad) and the Wolfson Pi audio card (for the STM32F407 Discovery) are shown in Figures 1.3 and 1.4. The audio booster pack and the launchpad plug together, whereas the Wolfson audio card, which was designed for use with a Raspberry Pi computer, must be connected to the Discovery using a custom ribbon cable (available from distributor Farnell).

Rather than presenting detailed instructions here that may be obsolete as soon as the next version of *MDK-ARM* is released, the reader is directed to the “getting started” guide at the partner website <http://www.wiley.com/go/Reay/ARMCortexM4> and before progressing to the next chapter of this book will need to install *MDK-ARM*, including the “packs” appropriate to the hardware platform being used and including the CMSIS DSP library, download the program examples from the website, and become familiar with how to open a project in *MDK-ARM*, add and



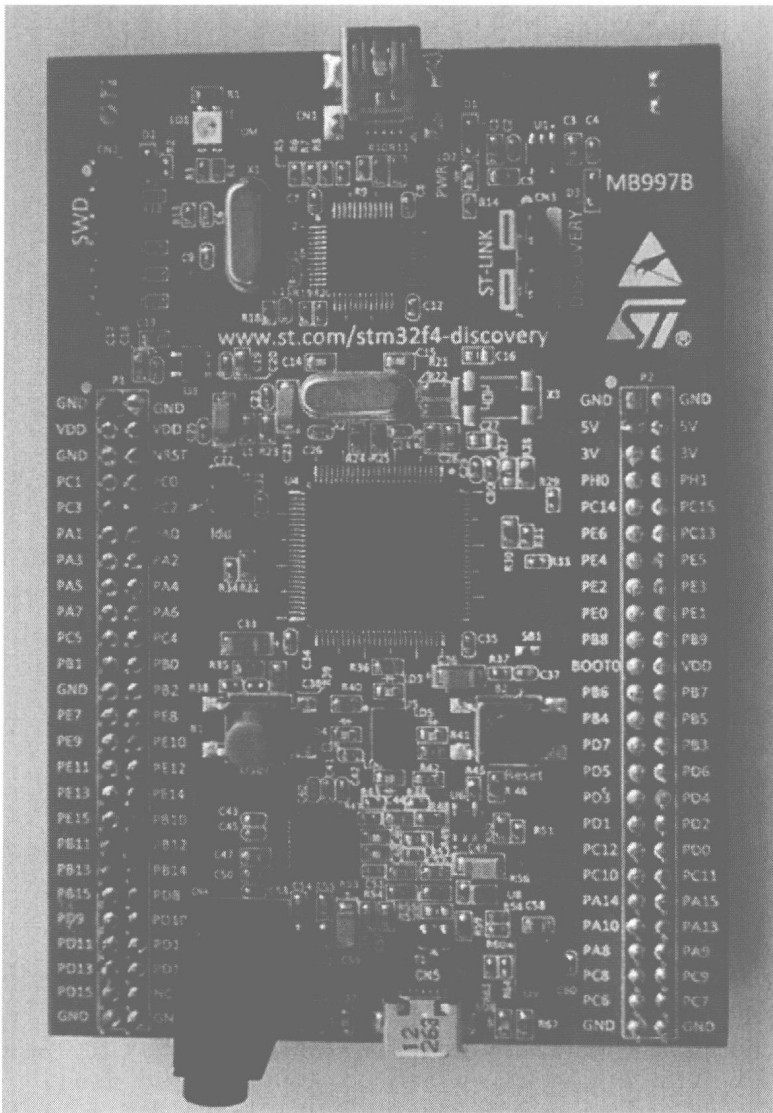


Figure 1.2 STMicroelectronics STM32F407 Discovery.

remove files from a project, build a project, start and stop a debug session, and run and halt a program running on the ARM Cortex-M4 processor.

Some of the example programs implement DSP algorithms straightforwardly, and with a view to transparency and understandability rather than computational efficiency or elegance. In several cases, ARM's CMSIS DSP library functions are used. These are available for both the STMicroelectronics and Texas Instruments processors as part of the *MDK-ARM* development environment. In appropriate circumstances, these library functions are particularly computationally efficient.