Chapter 1

Introduction to Microcontrollers and This Book

As the development of Very Large Scale Integrated Circuits (VLSI) in recent years, more and more advanced semiconductor devices and equipments have been built with very high intensity and density. Millions of MOSFETs can be integrated in a very small semiconductor chip to generate multifunction processors, called `Microprocessors`. Microprocessors can be considered as a very large scale integrated circuit device that can be programmed to perform specific functions or tasks.

One of the most popular and important microprocessors is the Center Processing Unit, or CPU, which is the center of a computer and used to process and coordinate all operations on a computer. Some other popular microprocessors can be categorized into the different groups based on their functions. Some popular microprocessors are:

1. CPU—Including the Intel family such as 8080/8085/8086/80286/80386/80486/80586 and Pentium, the Motorola family such as M6800, M68000, M68HC11, and M68HC12, and the Apple family such as 6502.
2. Parallel 8-bit I/O ports—Including the Intel family such as 8255 and the Motorola family such as M68230 Parallel Interface and Timer (PIT).
3. Parallel-to-Serial Converter—Including the Intel family such as 8251 and the Motorola family such as M68681 dual UART.
4. Timer and Counter—Including the Intel family such as 8253 and the Motorola family such as MC1555U timer.
5. Interrupt Control Unit—Including the Intel family such as 8259 and the Motorola family such as MC6828 Priority Interrupt Controller.
6. Random Access Memory (RAM) Chips—Including the Intel family 28C256 (32 K × 8), 62512 (64 K × 8), and 62158 (1024 K × 8).
7. Erasable Programmable Read-Only Memory (EPROM) Chips—Including the Intel family such as 27128 (128 K×8), 27256 (256 K × 8), and 27512 (512 K × 8).

By combining microprocessors with memory units and I/O ports, a `Microcontroller` system can be built. Sometimes a microcontroller is also called a microcomputer. In fact, a microcontroller is made by embedding processors, a memory unit, and I/O ports into a single semiconductor chip, and this is the current module of a modern

Practical Microcontroller Engineering with ARM® Technology, First Edition. Ying Bai.

Companion Website: www.wiley.com/go/armbai

microcontroller unit (MCU) used in all aspects in our present-day society. The latest MCU module is the ARM® Cortex®-M4 family.

Let's have a closer look at the structure and configuration of a microcontroller or a microcomputer system.

1.1 MICROCONTROLLER CONFIGURATION AND STRUCTURE

As we mentioned, by combining some microprocessors with memory units and I/O ports, a microcontroller can be built. In fact, a microcontroller can be built by combining three basic components with three system buses as shown in Figure 1.1.

Three components are CPU, Memory, and I/O Ports. These three components are connected with three system buses, Address Bus (`A.B.`), Data Bus (`D.B.`) and Control Bus (`C.B.`), to provide the following functions:

- The CPU works as headquarters for the microcontroller to provide all controls to other components and coordinate them to fulfill the desired tasks assigned to the microcontroller.
- The memory unit works as a storage unit to store the user's program, including the user's instructions and data. Some system programs and data are also stored in special memory units such as PROM, EPROM, EEPROM, or flash memory.
- The I/O Ports work as an interface and provide the communications between the CPU and the peripheral devices.

The communications between these three components are performed via three system buses. The Address Bus provides a valid address to the memory to enable the CPU to select and pick up the desired instruction or data from the selected memory space. The Data Bus is used to transfer a valid data item between components. The Control Bus provides valid operational signals to coordinate the information transfer between components. Some popular control signals are Read/Write (`R/W`) signal used to read from or write into the memory, Chip Select (`CS`) signal used to decode the address to select the desired microprocessor chip, and Enable signal `E` that is similar to the `CS` signal.

Three components are connected together via three buses in tri-state mode, which means that the connection between any two components is disconnected or high

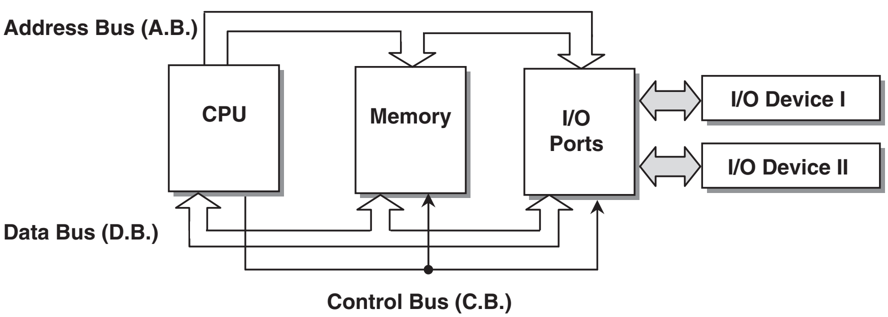

Figure 1.1. The basic structure and configuration of a microcontroller system.

impedance until a valid address is applied and decoded to enable the selected tri-state gates to turn on.

Regularly, a CPU contains three components: (1) a group of registers made of flip-flops, (2) Arithmetic and Logic Unit (ALU), and (3) control signal generator. The registers are used to assist the instruction's decoding and data operations since most operations between the CPU and memory are performed inside registers in the CPU because of the high execution speed of registers. The ALU is used to perform arithmetic and logic operations, and the control unit provides all timing and control signals required to perform all related operations of the CPU.

Generally the memory space is divided into two separate areas: One is the system memory space that is used to store instructions and data related to the normal operations of the microcontroller, and the other one is the users' memory space that is used to store the users' instructions and data.

The memory spaces also can be divided into the catch and heap areas based on the materials used to build the memory; the former is made of high-speed static RAM (SRAM), and the latter is made of dynamic RAM (DRAM) with relatively slower accessing speeds. The advantage of using the SRAM is that a higher memory accessing speed can be obtained, but much more MOSFETs are utilized for each SRAM unit and therefore makes the memory structure complicated with higher cost. The advantage of using DRAM is that higher memory densities or integration intensities can be obtained with mush simpler MOSFET structure and lower cost for each DRAM unit, but the working speed is relatively slower because of an additional refresh circuit applied on the DRAM. Because of the cost issue, usually the size of SRAM or the catch memory is small but the size of the heap or DRAM is huge.

The memory can also be categorized to the Random Access Memory (RAM), the Read Only Memory (ROM), the Erasable Programmable ROM (EPROM), or Electrically Erased Programmable ROM (EEPROM). Generally, the system instructions and data are stored in the ROM, EPROM, or EEPROM spaces. The users' instructions and data are stored in the RAM space. Based on the functions, the memory can be divided into either volatile memory or nonvolatile memory. The RAM belongs to the volatile memory since all information stored in this kind of memory would be gone when the power is off. However, the ROM, EPROM, and EEPROM belong to the nonvolatile memory since the information stored in this kind of memory would be still in there even the power is off.

Based on the structure, the RAM can be categorized to static RAM (SRAM) or dynamic RAM (DRAM). We will provide more detailed discussion about the static and dynamic RAM in Chapter 6.

The I/O Ports can be divided into two categories: the parallel and the serial I/O ports. Each I/O port can be mapped to a memory address, or each of them can have special I/O address that is different with a normal memory address. The former is called the I/O memory mapping addressing and the latter is called the direct I/O addressing.

1.2 THE ARM® CORTEX®M4 MICROCONTROLLER SYSTEM

Different embedded systems or MCUs have been developed and built by different vendors in recent years. One of the popular MCUs is the ARM® Cortex®-M MCU family. This kind of MCU provides multifunctions and control abilities, low-power consumptions, high-efficiency signal processing functionality, and low-cost and easy to use advantages. The latest product of the ARM® Cortex®-M family is Cortex®-M4 MCU.

The ARM® Cortex-M is a group of 32-bit Reduced Instruction Set Computing (RISC) ARM® processor cores licensed by ARM® Holdings. The cores are intended for microcontroller use and consist of the Cortex®-M0, Cortex®-M0+, Cortex®-M1, Cortex®-M3, and Cortex®-M4.

The ARM® Cortex®-M4 processor is the latest embedded processor by ARM® specifically developed to address digital signal control markets that demand an efficient, easy-to-use blend of control and signal processing capabilities. The combination of high-efficiency signal processing functionality with the low-power, low-cost, and ease-of-use benefits of the Cortex-M family of processors is designed to satisfy the emerging category of flexible solutions specifically targeting the motor control, automotive, power management, embedded audio, and industrial automation markets.

The ARM® Cortex®-M4 MCU provides the following specific functions:

- Although the Cortex-M4 processor is a 32-bit MCU, it can also handle 8-bit, 16-bit, and 32-bit data efficiently.
- The Cortex®-M4 MCU itself does not include any memory, but it provides different memory interfaces to the external Flash ROMs and SRAMs.
- Due to its 32-bit data length, the maximum searchable memory space is 4 GB.
- In order to effectively manage and access this huge memory space, different regions are created to store system instructions and data, users' instructions, data, and mapped peripheral device registers and related interfaces.
- The internal bus system used in Cortex-M4 MCU is 32-bit, and it is based on the so-called Advanced Microcontroller Bus Architecture (AMBA) standard. The AMBA standard provides efficient operations and low power cost on the hardware.
- The main bus interface between the MCU and external components is the Advanced High-performance Bus (AHB), which provides interfaces for memory and system bus, as well as for peripheral devices.
- A Nested Vectored Interrupt Controller (NVIC) is used to provide all supports and managements to the interrupt responses and processing to all components in the system.
- The Cortex-M4 MCU also provides standard and extensive debug features and supports to enable users to easily check and trace their program with breakpoints and steps.

To assist users to build professional microcontroller application projects, some useful development tools and kits are involved in this book to enable users to develop specific implementations easier and faster.

1.3 THE TM4C123GH6PM MICROCONTROLLER DEVELOPMENT TOOLS AND KITS

In this book, we concentrate on a typical and popular ARM® Cortex®-M4 MCU system built by Texas Instruments™ called TM4C123GXL, in which two ARM® Cortex®-M4 MCUs, TM4C123GH6PM, are utilized. The related development tools and kits can be categorized into two parts: the hardware part and the software part.

The hardware part includes:

- Tiva™ LaunchPad TM4C123GXL Evaluation Board (EVB).
- EduBASE ARM® Trainer (contains most popular peripherals and interfaces).
- Some other related peripherals, such as DC Motors, CAN interfaces, and D/A converters.

The software part includes:

- Integrated Development Environment Keil® ARM®-MDK μVersion®5 (IDE).
- TivaWare™ SW-EK-TM4C123GXL Software Driver Package.
- Stellaris In-Circuit Debug Interface (ICDI).

Appendices A~D, available from the website www.wiley.com, provide detailed information and directions for downloading and installing these software tools in your host computers.

1.4 OUTSTANDING FEATURES ABOUT THIS BOOK

1. Both ARM® assembly and C codes are provided in this book to assist users to develop professional projects with any language easily and faster.
2. More than 70 real example projects are provided in this book with detailed and line-by-line explanations and illustrations to enable users to understand and learn the programming skills easily and faster. These example projects covered most popular peripherals, such as Flash Memory and EEPROM, ADC, 4×4 Keypad, 7-Segment LEDs, LCD, DAC, I2C, UART, PWM, USB, Timers, Watchdog Timers, QEI, Analog Comparator, PID Controller, Fuzzy Logic Controller, FPU, and MPU, in different chapters.
3. Both the Direct Register Access (DRA) model and the Software Driver (SD) model programming techniques are introduced and discussed with a set of complete real example projects to cover all peripherals in the book.
4. A complete set of home works, including the true/false, multi-choice questions, comprehensive questions, and lab projects, is attached after each chapter. This enables students to understand what they learned better by doing something themselves.
5. A complete set of answers to all home works is provided for the instructors.
6. A complete set of MS Power Point teaching slides are provided for the instructors to make them teaching this book easily and conveniently.
7. Appendices A~E, available from the website www.wiley.com, provide a complete set of instructions and directions to enable users to download and install development tools and kits easily and faster.
8. Good textbook for college students and a good reference book for programmers, software engineers, and academic researchers.

1.5 WHO THIS BOOK IS FOR

This book is designed for college students and software programmers who want to develop practical and commercial control programming with ARM® Cortex®-M4 MCU and related development tools. Fundamental knowledge and understanding on C language programming is assumed.

1.6 WHAT THIS BOOK COVERS

This book is composed of 12 chapters with an easy study way to enable students to learn the ARM® Cortex®-M4 microcontroller technology and interface implementations easily. Each chapter contains homework and exercises as well as lab projects to enable students to perform necessary exercises to improve their learning and understanding for the related materials and technologies.

`Chapter 1`: Provides an overview and introduction about the microcontrollers and a global review for the book with highlights on outstanding features and organizations of the book.

`Chapter 2`: Provides detailed discussion and analysis about the ARM® Cortex®-M4 MCU hardware, which includes the architecture of the Cortex®-M4 Core and processor, memory (flash memory, SRAM, and EEPROM), GPIO Parallel and serial ports, Nested Vectored Interrupt Controller (NVIC), Private Peripheral Bus (PPB) and Advanced High-performance Bus (AHB), System Timer SysTick, system control block (SCB), floating point unit (FPU) and memory protection unit (MPU). The detailed discussions and introductions to the special MCU used in this book, TM4C123GH6PM, are also given in this chapter.

`Chapter 3`: Provides detailed discussion and analysis about the development tools and kits for the Cortex-M4 MCU. These tools and kits are discussed separately based on the hardware and software sections. The TM4C123GXL EVB and EduBASE ARM® Trainer are discussed as the hardware kits and tools. The Keil® ARM®-MDK µVersion®5 that works as an IDE and the TivaWare™ SW-EK-TM4C123GXL Software Driver Package that works as a software suite and driver library are introduced as software tools and kits. The Stellaris® ICDI driver that works as a debugger for the TM4C123GXL EVB is also discussed in this chapter.

`Chapter 4`: Provides detailed and complete discussion about the ARM® Cortex®-M4 microcontroller software and instruction set. These discussions include the ARM® Cortex®-M4 software development structure, a complete Cortex®-M4 assembly instruction set, the Keil® CMSIS Core specific intrinsic functions, inline assembler, C programming procedure for the Cortex®-M4 MCU, and two programming models applied by the TivaWare™ Peripheral Driver Library. The detailed procedure of building and developing an example ARM® Cortex®-M4 MCU project with C codes is discussed step by step at the end of this chapter.

`Chapter 5`: Provides detailed discussions about the ARM® Cortex®-M4 interrupts and exceptions. These discussions covered the interrupt and exception sources, interrupt handlers, and interrupt and exception vector tables. Most popular control registers involved in the Nested Vectored Interrupt Controller (NVIC) are introduced in details with related examples. The NVIC macros and NVIC API functions supported by the CMSIS Core software package are also introduced in this chapter. The GPIO interrupts handled by some API functions provided by the TivaWare™ Peripheral Driver Library are discussed at the end of this chapter.

`Chapter 6`: Provides detailed discussion about ARM® Cortex®-M4 memory system. In particular, the memory system used in the TM4C123GH6PM MCU is discussed in extensive detail. These discussions include the memory architecture, the entire memory map with accurate addresses for each component, SRAM, Flash Memory, internal ROM, and EEPROM. Most popular control registers applied on these memory models are introduced and discussed in detail with related example projects. The API functions used for flash memory and EEPROM are also discussed in this chapter. Some special memory implementation techniques, such as bit-band alias, flash memory programming, and EEPROM programming, are introduced with actual example projects.

`Chapter 7`: Provides detailed discussions about the ARM® Cortex®-M4 Parallel I/O Ports programming. The major parallel peripherals discussed in this chapter include the on-board keypads, analog-to-digital converter, and PWM. All peripherals in the Cortex-M4 MCU system, including the parallel and serial, are interfaced to the processor via a group of General-Purpose Input–Output ports, also called GPIO Ports. All GPIO Ports and control registers related to these parallel peripherals are introduced with some real example projects in details. Those peripheral-related API functions provided by the TivaWare™ Peripheral Driver Library are also discussed in this chapter with actual example projects.

`Chapter 8`: Provides detailed discussions about the ARM® Cortex®-M4 Serial I/O Ports programming. The major serial peripherals discussed in this chapter include the Synchronous Serial Interface (SSI), Inter-Integrated Circuit (I2C) Interface, and Universal Asynchronous Receivers/Transmitters (UARTs). Several real example projects related to these serial peripherals include: on-board LCD Interface project, on-board 7-segment LED project, digital-to-analog converter project, I2C interfacing project, and UART project. The related API functions provided by the TivaWare™ Peripheral Driver Library are also discussed with some real example projects in this chapter.

`Chapter 9`: Provides detailed discussions about the ARM® Cortex®-M4 Timer system and the USB system as well as their applications. These discussions include the General-Purpose Timers Module (GPTM), Watchdog Timers Module (WDTM), and USB Controllers. All timer-related control registers, including the GPTM, WDTM, and USB, are introduced and discussed in detail with several real example projects. The 64-bit Wide-Purpose Timers Module (WGPTM) is also discussed. Different implementations of using GPTM, WGPTM, and WDTM are analyzed and discussed with some real example projects. These implementations include the one-shot/periodic timer, input edge-count mode, input edge-time mode, real-time clock (RTC) mode, and PWM mode. The related API functions supporting those modes and provided by the TivaWare™ Peripheral Driver Library are also introduced with example projects.

`Chapter 10`: Provides detailed discussion about the ARM® Cortex®-M4 other peripheral programming. These peripherals include the Controller Area Network (CAN), Quadrature Encoder Interface (QEI), and Analog Comparators (ACMP). The fundamental architectures of the CAN, QEI, and ACMP are introduced first. Then all control registers related to these peripherals are introduced with several example projects. Some advanced and updated control strategies, such as PID control and fuzzy logic control, are applied in those projects to attract the students' interests with regard to learning these peripherals. Some professional techniques, such as motor model identification and PID controller design with MATLAB® simulation, are also involved in these projects. The related API functions for these peripherals are introduced in this chapter.

`Chapter 11`: Provides detailed discussion about the ARM® Cortex®-M4 Floating Point Unit (FPU). First the single-precision and double-precision floating point numbers and protocols are introduced. Then the FTP architecture applied in the Cortex-M4 MCU is discussed in details. All control registers used for the FPU are discussed and illustrated in detail. The FPU-related API functions provided by the TivaWare™ Peripheral Driver Library are also discussed. Finally a real example project is provided to illustrate how to use the FPU to perform some sophisticated floating point data operations.

`Chapter 12`: Provides detailed discussion about the ARM® Cortex®-M4 Memory Protection Unit (MPU). An overview about the MPU is provided first in this chapter. All control registers used for the MPU are discussed and illustrated in details. The MPU-related API functions provided by the TivaWare™ Peripheral Driver Library are also discussed. Finally a real example project is provided to illustrate how to use the MPU to perform desired protection functions for the selected memory regions.

- `Appendix A`—Provides instructions about downloading and installing the Keil® MDK-ARM® 5.1 IDE.
- `Appendix B`—Provides instructions about downloading and installing the TivaWare™ SW-EK-TM4C123GXL Software Package.
- `Appendix C`—Provides instructions about downloading and installing the Stellaris ICDI and Virtual COM Port.
- `Appendix D`—Provides the Tiva™ C Series TM4C123G Based EVB Hardware Setup.
- `Appendix E`—Provides a set of CMSIS Core-Specific Intrinsic Functions.

1.7 HOW THIS BOOK IS ORGANIZED AND HOW TO USE THIS BOOK

This book is designed for both college students who are new to ARM® Cortex®-M4 microcontroller system and professional application programmers who have professional experience on this topic.

Chapters 2~9 provide fundamental and professional introductions and discussions about the most popular ARM® Cortex®-M4 MCU applications with the most widely used peripherals, such as flash memory and EEPROM, ADC, DAC, PWM, UART, USB, I2C, SSI, LCD, and GPTM. Some other peripherals, including the CAN, ACMP, and QEI, are discussed in Chapter 10. Two optional components, FPU and MPU, are discussed in Chapters 11 and 12.

Based on the organization of this book we described above, this book can be used as two categories such as Level I and Level II, which is shown in Figure 1.2.

For undergraduate or graduate college students or beginning software programmers, it is highly recommended to learn and understand the contents of Chapters 2~9 since those are fundamental knowledge and techniques used in the ARM® Cortex®-M4 microcontroller programming and implementations (Level I). For the material in Chapters 10~12, they are only related to the additional and optional peripherals and components used in the ARM® Cortex®-M4 microcontroller programming, and it is optional to instructors and it depends on the time and schedule (Level II).

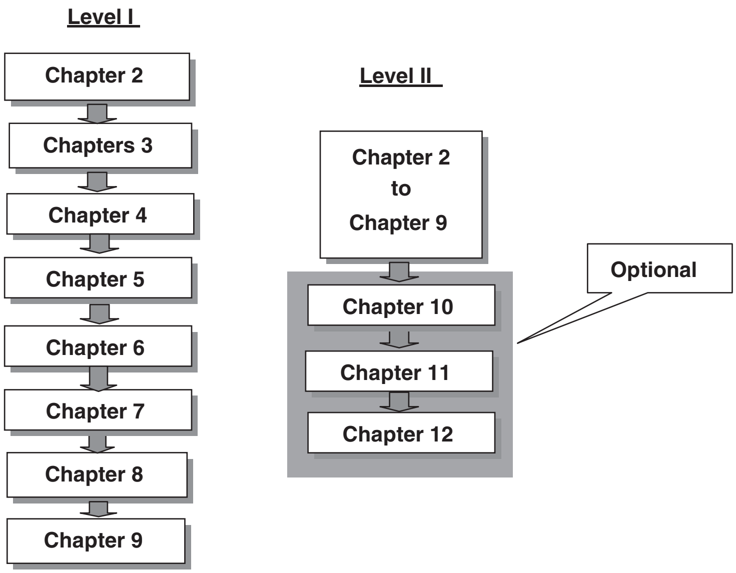

Figure 1.2. Two study levels in the book.

1.8 HOW TO USE THE SOURCE CODE AND SAMPLE PROJECTS

All projects in the book can be divided into two parts: the class projects and the lab projects. All source codes for those projects are available in the book. However, all class projects are available to both instructors and students, but the lab projects are only available to instructors since students need to build these lab projects themselves. All sources codes for these projects have been debugged and tested, and they are ready to be executed in any TM4C123GXL EVB.

All class projects are categorized into the associated chapters that are located at the folder ARM `Class Projects` that is located at the site http://booksupport.wiley.com. You need to use either the book `ISBN`, the book `Title`, or the `Author` name to access and download these projects into your computer and run each project as you like. To successfully run those projects on your computer, the following conditions must be met:

1. The Keil™ MDK-ARM® 5.1 and above IDE must be installed in your computer.
2. The TivaWare™ SW-EK-TM4C123GXL Software Package should be installed in your computer, and this package must be installed if you want to use any API function provided by TivaWare™ Peripheral Driver Library.
3. The Stellaris ICDI and Virtual COM Port driver must be installed in your computer.
4. The TM4C123GXL EVB and EduBASE ARM® Trainer must be installed and connected to your host computer.

Refer to Appendices A~C to complete steps 1~3, and refer to Appendix D for step 4.

All book-related teaching and learning materials, including the class projects, lab projects, appendices, faculty teaching slides, and home work solutions, can be found from the associated folders located at the Wiley Book Support site, as shown in Figure 1.3.

These materials are categorized and stored at different folders in two different sites based on the teaching purpose (for instructors) and learning purpose (for students).

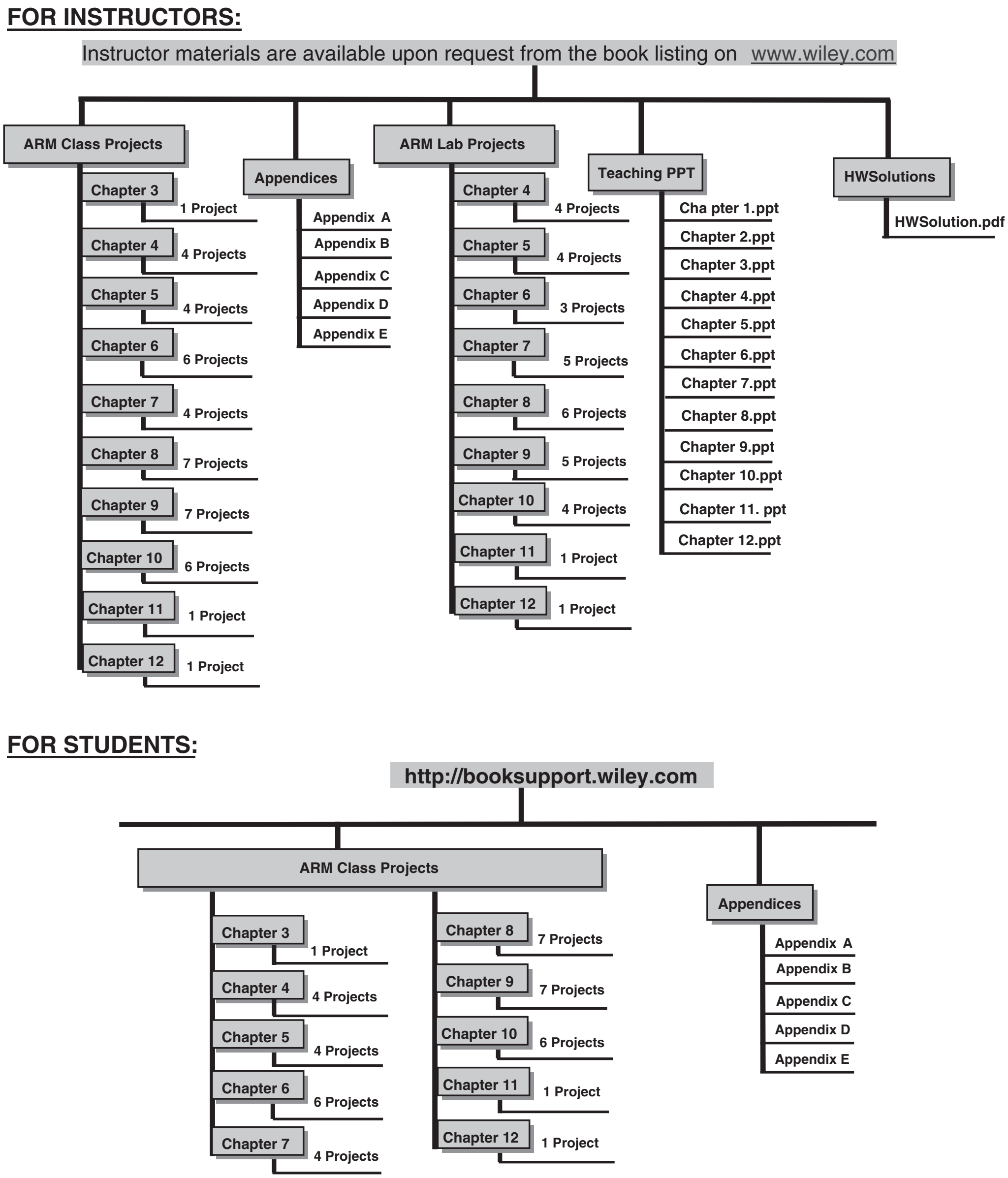

Figure 1.3. Book-related materials on the website www.wiley.com.

FOR INSTRUCTORS

1. ARM `Class Projects` Folder: Contains all class projects for different chapters.
2. ARM `Lab Projects` Folder: Contains all lab projects included in the home work sections in different chapters. Students need to follow the directions provided in each homework lab section to build and develop these lab projects themselves.
3. `Appendices` Folder: Contains all Appendices (`Appendices A~E`) that provide useful references and practical instructions to download and install the ARM® Cortex®-M4-related development tools and kits.
4. `TeachingPPT` Folder: Contains all MS-PPT teaching slides for each chapter.
5. `HWSolutions` Folder: Contains a set of complete solutions for the homework developed and used in the book. The solutions are categorized and stored at the different chapter subfolder based on the book chapter sequence.

FOR STUDENTS

1. ARM `Class Projects` Folder: Contains all class projects in different chapters. Students can download and run these class projects in their host computers after a suitable environment has been set up (refer to conditions listed above).
2. `Appendices` Folder: Contains all Appendices (`Appendices A~E`) that provide useful references and practical instructions to download and install the ARM® Cortex®-M4 related development tools and kits.

1.9 INSTRUCTORS AND CUSTOMERS SUPPORTS

The teaching materials for all chapters have been extracted and represented by a sequence of Microsoft Power Point files, each file for one chapter. The interested instructors can find those teaching materials from the folder `TeachingPPT` that is located at the site www.wiley.com, and those instructor materials are available upon request from the book listing on www.wiley.com.

A set of complete homework solution is also available upon request from the book listing on www.wiley.com.

E-mail support is available to readers of this book. When you send e-mail to us, please provide the following information:

- The detailed description about your problems, including the error message and debug message as well as the error or debug number if it is provided.
- Your name, job title, and company name.

Please send all questions to the e-mail address: ybai@ieee.org.

Detailed structure and distribution of all book-related material in the Wiley site, including the teaching materials for instructors and learning materials for students, are shown in Figure 1.3.