

Embedded System Development and Application Course Series

# **Embedded System Development and Labs for ARM**

*Edited, revised, and updated by **Radu Muresan***

- Compound with Embest ARM Labs System**
- Compound with Multimedia Teaching Demo Modules**

Embedded System Development and Application Course Series

## To Readers

- *Embedded System Application Development and Labs* Textbook is compounded with the Embest ARM Development System that was developed by Embest Inc. at Shenzhen, China. Any reader who is interested in using the Embest development tools for ARM can contact Embest Inc. The following is the Embest contact information:

Room 509, Luohu Science&Technology Building,  
#85 Taining Road, Shenzhen, Guangdong, China  
ZIP: 518020

Tel: +86-755-25635656, 25635626

Fax: +86-755-25616057

Web: <http://www.armkits.com> or <http://www.embedinfo.com>

E-mail: [market@embedinfo.com](mailto:market@embedinfo.com)

*Embedded System Development and Applications* Textbook and *Embedded System Application Development and Labs* Textbook are compounded with teaching demo modules separately. If you are interested in any of these demo modules please contact Embest Inc.

## **An Introduction to This Book**

This book is a Lab manual and is part of the “Embedded System Development and Application” course series. This Lab manual is based on the Embest ARM Labs System development platform hardware, which uses an ARM processor as its core. The Lab manual is a complete teaching and training tool for developing Embedded Systems. The book contains 22 Labs that include: Labs for the embedded software development fundamentals; Labs for basic interfaces; Labs for human-machine interface; Labs for communication and audio interfaces; Labs for uC/OS-II embedded real-time operating system porting and application; etc. This book offers many examples for the embedded system learners. The Labs form an embedded system teaching or training tool and are introduced in a gradual manner from simple to complex applications that are close related to the engineering world. This book is accompanied by a free CD that contains the Embest IDE Pro Education Version software produced by Embest Inc.

This book can be used as a Lab teaching material for embedded and real-time embedded systems at undergraduate or graduate level with majors in Commuter Science, Computer Engineering, Electrical Engineering; or for professional engineers.

### **About the Editor of the English Version of the Embedded System Development and Labs**

Radu Muresan is the editor of the English version of the “Embedded System Development and Labs” book offered first in Chinese by Embest as an accompaniment book to their ARM development platform. Radu Muresan has a PhD in Electrical and Computer Engineering from the University of Waterloo Canada and is currently an assistant professor at the University of Guelph Canada. He is currently teaching the Real-Time Systems Design course in the School of Engineering using the “Embedded System Development and Labs” book and the Embest development platform and tools.

# FOREWORD

## The Evolution of the Embedded Technology

The embedded systems based on 8-bits single-chip microprocessors have already being used in many fields. Even today, most of the embedded applications are still belonging to the early stage of embedded system. A general characteristic of these applications is that they include a MCU (micro-controller unit), sensors, monitoring or testing devices, service systems, display devices, etc. Also, these applications have functions such as testing, displaying, processing and automatic control of information. In some industrial control applications such as automobile electronic systems and intelligent home electronic devices, these MCUs are interconnected in a network through common buses such as CAN, RS-232, RS-485, etc. This kind of network has limited applications. The related communication protocols are relatively simple and excluded to the Internet that is being widely used. Today Internet has become a fundamental communication system that is serving the society and also becomes an important way of exchanging information needed by the people. The embedded systems can be integrated to the Internet, helping to transfer information world wide.

With the process of integration of embedded devices and Internet, complex high-end applications, mobile phones, PDAs, routers, modems, etc will demand high performance from the embedded processors. Although the embedded technologies that are based on the 8-bit single-chip computer still exist, these kinds of technologies can't meet the requirements of the evolution of the future embedded system technology due to its limited functions. The market and technology competition consistently requires a higher function/price ratio. On the other hand, the time development of an embedded system is also being required to be shorter and shorter. From the early 70s when the embedded system concept has been introduced, the embedded systems have evaluated rapidly and high performance and low power consumption systems have been developed. In the early stages, many embedded system had no real time operation system (RTOS) support; those embedded system were merely processing some functions such as simple controls that respond to the outside input through a simple loop control algorithm, etc. With the evolution of technology, the system complexity increased and the application fields of the embedded system expanded. Every time when some new functions were introduced, the system software design needed to be completely changed. So, the lacking of real time operating system support became an important issue. Due to the fact that running RTOS on an 8-bit single-chip processor has some difficulties, the 32-bit microprocessor (as core of high performance embedded systems) became a common trend of technology development.

From the early 90s, the way of embedded system design had gradually been changed from "Integrated Circuit" level to "Integrated System" level. The "Integrated Circuit" is based on embedded microprocessors and DSPs. The "Integrated System" is based on SoC (System on Chip) concept that was introduced at that time. Nowadays, the embedded system has entered a design phase that is based on SoC and the SoC standardization is used more and more. SoC provide complex hardware features for high performance embedded system. SoC also provides the basic hardware support for real time embedded operating system. During 80s, some real time operating systems emerged. The most common RTOS include VxWorks, Windows CE, Palm, ucLinux, pSOS, uC/OS etc. However, the real application on these RTOS happened only in recent last few years.

There are two reasons for this development. One is the increased requirement of the complexity of embedded

software development in the last few years; another is the SoC. RTOS can be run in a dependable, effaceable and affordable way. Most of the RTOS are expensive. As a result, some open sourced RTOS such as uc/OS-II, ucLinux are being chosen by many users. These open sourced RTOS are also suitable as teaching tools. The RTOS functionality and principles are relatively complex. Anyone who is interested in the RTOS research or development, please refer to related books in the field.

Embedded systems based on embedded processor are characterized by small size, lightweight, low cost and high performance. The largely used 32-bit microprocessors are ARM from ARM Ltd, Alpha from Compaq, PA-RISE from HP, Power-PC from IBM, MIPS from MIPS Technologies Inc., SPARC from Sun etc.

ARM processors have merits of high performance, low power consumption, low cost, etc. ARM processors are the most widely used microprocessors amongst the 32 bit and 64 bit microprocessors.

In the early 90s, the semiconductor industry formed a production chain that combined the design industry, manufacture industry, packaging and testing industry. Some real semiconductor companies were greatly developed and some fabless (chipless) companies also emerged. The Advanced RISC Machines (ARM), is the most successful company based on the fabless chipless mode. ARM doesn't produce or sale chips but provides high performance IP cores that are being sold to authorized semiconductor companies.

Let's look back to the development history of ARM technologies. At the time when ARM7 system architecture (system architecture v3) was just been accepted and applied, the embedded microprocessor market was overwhelmingly occupied by 8-bit and 16-bit microprocessors. However these microprocessors can't meet the requirements of developing high-end applications such as mobile phones, modems, etc. These high-end products needed the 32-bit microprocessors processing power and higher programming code density than the 16-bit CISC processors. In order to meet these requirements, a T variety of ARM architecture was developed. This T variety is called 16-bit Thumb Instruction Set. Thumb technology is one of the best characteristics of ARM technology. The ARM7TDMI (system architecture v4T) is the first microprocessor that supports Thumb instruction set. ARM7TDMI's work mode can be switched to the Thumb working state. The 32-bit processor can be run with 16-bit Thumb instruction set. So, thumb is a bridge between the 16-bit older system and the 32-bit new system. ARM architecture provided higher performance processor solutions to the users who were looking for higher performance processors. These features greatly increased the embedded development as well as ARM technology. The 16-bit microprocessors were not developed as people expected. The reason was complicated. Maybe one of the reasons was that the 32-bit ARM processors provided higher performance and lower price than the 16-bit processors and enabled the high-end embedded applications to jump to the new 32-bit generation.

Many semiconductor companies have accepted the ARM processor production development. There are more than 100 IT companies that are currently using the ARM processors. Among them 19 of the 20 largest semiconductor companies are developing chips based on the ARM architecture. These semiconductor companies include TI, Philips, Intel etc. The excellent processor performance and the punctual marketing enabled ARM to get tremendous resources. These resources greatly accelerated many kinds of system chips developed for different applications. ARM has already established its lead position in the embedded technologies and the ARM technologies are being widely used. ARM has gained great success in the field of high performance embedded applications and the number one position in 32-bit embedded applications in the world. In 2002, ARM processors occupied 79.5% of 32-bits and 64-bit microprocessor market in the world. There were 20 billion ARM cores used by 2002. Nowadays, ARM processors are almost in everybody's pocket

because almost all of the mobile phones, PDAs are developed based on ARM cores. As a result, in order to keep up with the modern embedded technologies, people need to study the embedded development technologies that are based on 32-bit ARM processors and also need to study its development environment and platform technologies.

If integrated circuit and related technologies are the drivers of PC development that have increased the IT technologies in the last twenty years, we could say that, besides the PC technologies, the portable, mobile and Internet related embedded Internet information processing devices will be the main drivers that will enable a Post-PC time becomes true in the next few decades. Currently the embedded Internet is merely limited to some applications such as mobile business, intelligent electronic home devices, control and intelligent devices etc. With the development of related technologies, embedded technology will be developed more and more at an unimaginable speed with more complex applications. The area of embedded applications will be expanded and the embedded systems and applications will be more valuable to the society.

Currently the Wintel (Microsoft and Intel federation established at early 90s) has dominated the computer industry. With the development of information technology and network technology, the embedded technology will make this monopoly not exist in the Post-PC time. Embedded System will be the main portion of non-PC devices.

## Current Status of Embedded System Tools for Teaching and Development

Human resource is the key of developing embedded system technology. Enhancing the embedded technology teaching in the universities is to provide the embedded development human resources. On the other hand, the existing engineering staffs in the companies are also needed to be trained by modern embedded technology.

The engineering staffs in companies welcome the embedded system training courses that are based on ARM. To this point, establishing a new embedded system training system that is based on ARM is very necessary and urgent. This kind of university training courses will resolve the problem of lacking technology human resources for developing embedded systems.

Although the ARM processors have higher performance and higher processing power than 8-bit single chip computers such as 51 series microprocessors, the complexity and difficulty of developing embedded system hardware and software based on ARM are greater.

The main purpose of establishing new tools based on ARM embedded technologies, is the need to enhance the traditional embedded system training by adding complex embedded sample program modules, real time operation system, etc to the text book to make the teaching closer to the real world of electrical and computer engineering.

## About the Course Series and Related Labs

In order to establish tools based on 32-bit ARM embedded technologies, the main requirement is to develop the basic knowledge about ARM architectures. The “ARM System on Chip Architecture” by Steve Furber together with the “ARM Architecture Reference Manual” by David Seal can provide the necessary background.

The Course Series consists of the following basic textbooks:

□ ***Embedded System Development and Applications Textbook (Available in Chinese)***

**--Compound with multimedia demo modules**

Major Contents: Basic concepts of embedded system application development, an overview of ARM technology, ARM instruction set, the foundation of embedded program design based on ARM, development samples based on ARM, open sourced real time operating system uC/OS-II and uCLinux, porting and application software development. The readers can completely master the basic concepts and the design flow of developing an embedded system, the embedded software development skills based on ARM, the basic concept of porting and application development of embedded operating systems.

□ ***Embedded System Application Development and Labs Textbook***

**-- Compound with Embest ARM Labs System**

**--Compound with Multimedia Teaching Demo Modules**

Major Contents: The embedded system application development Labs is based on the Embest ARM development system. The Labs are coordinated with the course textbook *Embedded System development and Applications*. The Labs include five parts: basic labs for embedded development, basic device interfacing labs, complex human-machine interfacing labs, communication and voice interface labs, embedded RTOS (Real-Time Operating Systems) porting and application development. These five parts have 22 Labs in total. The labs increase in their difficulty as the book progresses through more material. The labs are very practical and target real world applications. The readers can quickly master the skills that are needed to develop real projects. The purpose of this book is to develop students' creation ability, design ability, real world engineering project development ability.

In order to coordinate with the course teaching and Lab teaching, we developed Multimedia Demo Modules for the *Embedded System Development and Applications* and *Embedded System Application Development and Labs* courses. As a start point, we will continually change or add new course textbooks, lab textbooks or multimedia demo modules based on real practical teaching techniques and the evolution of related technologies.

This set of textbook combined with class teaching and lab teaching, provides a solution for students to master embedded system development technologies based on ARM. The tools used in the ARM embedded application development include the Integrated Development Environment (IDE), the Embedded Real-Time Operating System, the evaluation board, the JTAG emulator, and other auxiliary tools. Generally, an Integrated Development Environment (IDE) with its basic functions is the only needed tool for embedded system development. Others tools are optional.

The major IDEs used in the world include: SDT and ADS from ARM, Multi2000 from GreenHill, Embest IDE for ARM from Embest Inc, etc. The emulators used are Muti-ICE from ARM and ARM JTAG Emulator from Embest Inc.

SDT and ADS is the IDE produced by ARM Ltd in its early state (discontinued). The S3C series chips from SAMSUNG are the most widely used ARM based microprocessors. Embest Inc has developed the Embest ARM Development board based on the S3C44B0 chip. This development board has memory, I/O, digital LCD display, touch screen, keyboard, IIS, Ethernet interface, USB interface IIC interface, advanced extension including IDE hard disk, CF card, flash disk etc. The Embest software and hardware tools are complete, reliable and easy to use. These qualities are most needed in an university environment and made us use these tools for

our embedded based courses.

**NOTE that other microprocessor and interfacing courses and textbooks can provide the basic background for using the Embest development system.**

## The Prerequisites for Studying This Course Series

Before studying this course, students should have studied courses such as Microcomputer Interfacing, C Language Programming, and have some basic knowledge of operating systems, computer architecture and network protocols. The text book series have also presented background knowledge of basic networks protocol, touch panel basics, keyboard interface programming basic etc.

## Thanks (from Radu Muresan)

I am using this book to teach the “Real-Time Systems Design” course at the University of Guelph Canada. I want to thank the Embest engineers {Liuchi, Zhang Guorui, Xu Guangfeng, Baidong} for their full support during editing the English version of this book. They have provided detailed technical support and materials for the assimilation of the existing labs and development of new labs. I also want to thank Oliver Zhihui Liu for providing the first English draft of the Chinese version of the “Embedded Systems Development and Labs” book. I have worked with his translation and generated the English version of the “Embedded Systems Development and Labs”. In this version, I have updated the technical content of the labs based on the “ARM Architecture Reference Manual”, I have verified all the labs and I have added a new real-time lab. Finally I want to thank my master students Zhanrong Yang and Shukla Nupoor for their contributions to the testing of the labs. Zhanrong Yang has work hard to help with the board setup and with providing support with the Chinese documentation. Unfortunately, I was not able yet to edit other books based on the Embedded System Development series. However, this lab manual can be combined with any microcomputer interfacing or real-time system design courses offered in other universities.

Due to the fact that the translation draft was sort of word by word translation there are still English and technical errors throughout the book. I have issued this version so the students can perform the required labs for my Real-Time System Design course. I am still working on editing and updating the book and I hope to produce a better version soon. Also, I am planning to produce an Embedded Real-Time system design text book that can accompany this lab manual. However, I believe that this lab book is an excellent tool for teaching embedded systems based on the ARM architecture. I have used other IDEs and I can say that the Embest engineers have developed an excellent product. I want to congratulate the Embest engineers for putting together this product.

**Radu Muresan, 2005**

## PREFACE

Theory teaching and Lab teaching are two important parts of the modern advanced education. Lab course is an important part in the teaching process. This book is the Lab manual of the Embedded System Development Course Series that provides teachers and students with complete embedded system training tools based on the ARM architectures. In this Lab manual, we focus mainly on developing complete embedded applications using the Embest development system. The applications provide the software and hardware details of the designs. We integrated complex embedded system application sample modules, porting of embedded operating systems, etc. Using this manual the students can learn not only the basics of the embedded system development, but also can learn how to develop complex interface modules that apply to real world applications.

The following outlines the content of the chapters:

**Chapter One:** An overview of embedded system development, embedded system IDE, ARM embedded development system, embedded study, etc.

**Chapter Two:** Embest embedded IDE for ARM, Embest ARM development system and Embest JTAG emulator.

**Chapter Three:** Basic Labs of embedded software development based on ARM including: ARM basic instruction set, Thumb instruction set, assembly programming, ARM processor mode switching, embedded C programming, C and assembly language mix programming, overview of programming. (This chapter provides the basic knowledge of embedded software development, basic programming skills, usage of IDE)

**Chapter Four:** Labs that target basic peripheral interfacing in embedded systems. The chapter includes applications using memory, I/O interface, interrupts, serial communication, real-time clock and simple digital LED interface.

(These Labs teach the student the basic principles of peripheral interfacing in embedded systems)

**Chapter Five:** Complex applications that introduce the human-machine interfacing. This chapter includes a Lab using the LCD display, a Lab using the keyboard control, a Lab using the touch screen control.

(These Labs are more complex, difficult and closer to the real engineering applications. These labs require good skills in using the Embest development system)

**Chapter Six:** Complex labs for developing applications using communication interfacing and IIS voice interfacing. This chapter includes a Lab of IIC serial communication bus, a Lab of Ethernet communication and a Lab of IIS voice bus interface communication.

(Chapters 4, 5, and 6 can prepare the students to develop applications using various interfaces and device development that target real world applications.)

**Chapter Seven:** Introduces the uC/OS-II real-time operating system, porting and real-time application development based on the Embest tools.

(Through the Labs of this chapter, the students will learn how to port uC/OS-II to the ARM processor and how to build simple real-time applications based on the uC/OS-II kernel. They will learn the porting steps of the uC/OS-II kernel to the ARM7 microprocessor, the boot flow of the uc/OS-II, the task management, the inter-task communication, the synchronization and the memory management under uc/OS-II kernel.)

**Appendix A and B:** Instruction Quick Reference Table and Instruction Set Coding Table.

**Appendix C:** An introduction to Embest ARM products.

**Appendix D:** An introduction to the contents of the CD attached to this book.

The CD attached to this book is IDE Pro, a free educational version of the IDE software that Embest Inc provides to the readers of this book. The readers can install this software and edit, compile and debug the sample programs on a software target emulator. After this software is installed, the readers can find the basic Lab sample software of Chapter 3 of this manual in the “\EmbestIDE\Examples\S3CEV40” directory. To run the rest of the sample programs of the manual the readers need to purchase the full version of the Embest IDE, the Embest development board and ICE emulator. The students should also study the embedded Lab development system course that introduces computer interfacing, computer application software development, computer operating systems, applied electronic technology, network communication, etc.

This lab manual can be used as a reference book for embedded system development based on ARM. There are many real-time operating systems (RTOS) for embedded applications based on 32b-bit systems (RTOS such as VxWorks, Windows CE, Palm, uClinux, uC/OS, etc). We have selected the uC/OS since this kernel is fully documented and is an excellent tool for learning to develop real-time embedded applications.

This manual together with the Embest development system can be used in teaching undergraduate and graduate courses in embedded systems design.

<b>Chapter 1: An Overview of Embedded System Application Development</b>	<b>19</b>
1.1 Embedded System Development and Applications .....	19
1.2 An Overview of Embedded Development Environment for ARM .....	19
1.2.1 Cross Development Environment .....	19
1.2.2 Software Emulator .....	20
1.2.3 Evaluation Board .....	20
1.2.4 Embedded Operation System.....	20
1.3 An Overview of ARM Development system .....	20
1.3.1 ARM SDT .....	20
1.3.2 ARM ADS.....	21
1.3.3 Multi 2000.....	22
1.3.4 Embeds IDE for ARM .....	24
1.3.5 OPENice32-A900 Emulator .....	25
1.3.6 Multi-ICE Emulator .....	25
1.4 How to Study Embedded System Application Development Based on ARM .....	26
<b>Chapter 2: Embest ARM Lab Development system</b>	<b>27</b>
2.1 An Overview of the Lab Development system .....	27
2.1.1 The Embest IDE.....	28
2.1.2 Embest Emulator for ARM JTAG.....	30
2.1.3 Flash Programmer .....	31
2.1.4 Embest S3CCEV40 Development Board.....	32
2.1.5 Connection Cables and Power Adapters .....	33
2.2 The Installation of Lab Development system .....	33
2.2.1 The Installation of Embest IDE.....	33
2.2.2 The Installation of Flash Programmer.....	36
2.2.3 The Interconnection of Software and Hardware Platforms.....	37
2.3 Lab Development system Hardware Circuits .....	37
2.3.1 An Overview of Lab Development Hardware .....	37
2.3.2 Hardware Reference for Software Design .....	46
2.3.3 Bus Expansion .....	50
2.4 The Usage of Embest IDE .....	51
2.4.1 Embest IDE Main Window .....	51
2.4.2 Project Management .....	51
2.4.3 Project Basic Settings .....	54
2.4.4 Project Compiling and Linking.....	68
2.4.5 Load Debugging.....	68
2.4.6 Flash Programmer .....	76
<b>Chapter 3 Embedded System Development Basic Labs</b>	<b>78</b>
3.1 ARM Assembly Instructions Lab 1 .....	78
3.1.1 Purpose.....	78

3.1.2 Lab Equipment.....	78
3.1.3 Content of the Lab 1 .....	78
3.1.4 Principles of the Lab 1 .....	78
3.1.5 Lab 1 Operation Steps.....	80
3.1.6 Sample Programs of Lab 1.....	84
3.1.7 Exercise.....	87
3.2 ARM Assembly Instruction Lab 2 .....	87
3.2.1 Purpose.....	87
3.2.2 Lab Equipment.....	87
3.2.3 Content of the Lab 2 .....	88
3.2.4 Principles of the Lab 2 .....	88
3.2.5 Lab Operation Steps.....	90
3.2.6 Sample Programs of Lab 2.....	91
3.2.7 Exercises .....	92
3.3 Thumb Assembly Instruction Lab.....	93
3.3.1 Purpose.....	93
3.3.2 Lab Equipment.....	93
3.3.3 Content of the Lab .....	93
3.3.4 Principles of the Lab .....	93
3.3.5 Operation Steps of Lab 3 .....	95
3.3.6 Sample Programs .....	95
3.3.7 Exercises .....	97
3.4 ARM Work Mode Labs.....	97
3.4.1 Purpose.....	97
3.4.2 Lab Equipment.....	97
3.4.3 Content of the Lab .....	97
3.4.4 Principles of the Lab .....	97
3.4.5 Operation Steps of the Lab.....	100
3.4.6 Sample Programs of the Lab.....	102
3.4.7 Exercises .....	104
3.5 C Language Program Lab 1 .....	104
3.5.1 Purpose.....	104
3.5.2 Lab Equipment.....	104
3.5.3 Content of the Lab .....	104
3.5.4 Principles of the Lab .....	104
3.5.5 Operation Steps.....	107
3.5.6 Sample Programs .....	107
3.5.7 Exercises .....	108
3.6 C Language Program Lab 2 .....	108
3.6.1 Purpose.....	108
3.6.2 Lab Equipment.....	109

3.6.3 Content of the Lab .....	109
3.6.4 Principles of the Lab .....	109
3.6.5 Operation Steps .....	111
3.6.6 Sample Programs .....	115
3.6.7 Exercises .....	118
3.7 Assembly and C Language Mutual Call .....	118
3.6.1 Purpose.....	118
3.6.2 Lab Equipment.....	118
3.6.3 Content of the Lab .....	118
3.6.4 Principles of the Lab .....	118
3.7.5 Operation Steps .....	120
3.7.6 Sample Programs .....	121
3.7.7 Exercises .....	124
3.8 Sum Up Programming .....	124
3.8.1 Purpose.....	124
3.8.2 Lab Equipment.....	124
3.8.3 Content of the Lab .....	124
3.8.4 Principles of the Lab .....	124
3.8.5 Operation Steps.....	126
3.8.6 Sample Programs .....	129
3.8.7 Exercises .....	132
<b>Chapter 4 Basic Interface Labs 133</b>	
4.1 Memory Lab .....	133
4.4.1 Purpose.....	133
4.4.2 Lab Equipment.....	133
4.1.3 Content of the Lab .....	133
4.1.4 Principles of the Lab .....	133
4.1.5 Operation Steps .....	140
4.4.6 Sample Programs .....	141
4.1.7 Exercises .....	145
4.2 I/O Interface Lab.....	145
4.2.1 Purpose.....	145
4.2.2 Lab Equipment.....	145
4.2.3 Content of the Lab .....	145
4.2.4 Principles of the Lab .....	145
4.2.5 Operation Steps .....	148
4.2.6 Sample Programs .....	149
4.2.7 Exercises .....	153
4.3 Interrupt Lab .....	153
4.3.1 Purpose.....	153
4.3.2 Lab Equipment.....	153

4.3.3 Content of the Lab .....	153
4.3.4 Principles of the Lab .....	154
4.3.5 Operation Steps .....	161
4.3.7 Exercises .....	165
4.4 Serial Port Communication Lab.....	165
4.4.1 Purpose.....	165
4.4.2 Lab Equipment.....	165
4.4.3 Content of the Lab .....	165
4.4.4 Principles of the Lab .....	165
4.4.5 Operation Steps.....	171
4.5.6 Sample Programs .....	171
Exercises .....	176
4.5 Real-Time Timer Lab.....	176
4.5.1 Purpose.....	176
4.5.2 Lab Equipment.....	176
4.5.3 Content of the Lab .....	176
4.5.4 Principles of the Lab .....	176
4.5.5 Lab Design.....	178
4.5.6 Operation Steps.....	180
4.5.7 Sample Programs .....	180
4.5.8 Exercises .....	185
4.6 8-SEG LED Display Lab .....	185
4.6.1 Purpose.....	185
4.6.2 Lab Equipment.....	185
4.6.3 Content of the Lab .....	185
4.6.4 Principles of the Lab .....	185
4.6.5 Operation Steps.....	188
4.6.6 Sample Programs .....	188
4.6.7 Exercises .....	189
<b>Chapter 5 Human Interface Labs    190</b>	
5.1 LCD Display Lab.....	190
5.1.1 Purpose.....	190
5.1.2 Lab Equipment.....	190
5.1.3 Content of the Lab .....	190
5.1.4 Principles of the Lab .....	190
5.1.5 Lab Design.....	199
5.1.6 Operation Steps.....	202
5.1.7 Sample Programs .....	203
5.1.8 Exercises .....	209
5.2 4 x 4 Keyboard Control Lab .....	209
5.2.1 Purpose.....	209

5.2.2 Lab Equipment.....	209
5.2.3 Content of the Lab .....	209
5.2.4 Principles of the Lab .....	209
5.2.5 Lab Design.....	210
5.2.6 Operation Steps.....	214
5.2.7 Sample Programs .....	214
5.2.8 Exercises .....	218
5.3 Touch Panel Control Lab .....	218
5.2.1 Purpose.....	218
5.2.2 Lab Equipment.....	218
5.2.3 Content of the Lab .....	218
5.2.4 Principles of the Lab .....	218
5.3.5 Lab Design.....	221
5.3.6 Operation Steps.....	223
5.3.7 Sample Programs .....	224
5.3.8 Exercises .....	229
<b>Chapter 6 Communication and Voice Interface Labs</b>	<b>230</b>
6.1 IIC Serial Communication Lab .....	230
6.1.1 Purpose.....	230
6.1.2 Lab Equipment.....	230
6.1.3 Content of the Lab .....	230
6.1.4 Principles of the Lab .....	230
6.1.5 Lab Design.....	237
6.1.6 Operation Steps.....	238
6.1.7 Sample Programs .....	239
6.1.8 Exercises .....	239
6.2 Ethernet Communication Lab .....	242
6.2.1 Purpose.....	242
6.2.2 Lab Equipment.....	242
6.2.3 Content of the Lab .....	242
6.2.4 Principles of the Lab .....	242
6.2.5 Operation Steps.....	256
Sample Programs .....	256
Exercises .....	259
6.3 IIS Voice Interface Lab .....	259
6.3.1 Purpose.....	259
6.3.2 Lab Equipment.....	259
6.3.3 Content of the Lab .....	259
6.3.4 Principles of the Lab .....	259
6.3.5 Sample Programs .....	263
6.3.6 Exercises .....	268

<b>Chapter7 Real Time Operation System Labs</b>	<b>269</b>
7.1 uC/OS Immigration Lab .....	269
6.3.1 Purpose.....	269
7.1.2 Lab Equipment.....	269
7.1.3 Content of the Lab .....	269
7.1.4 Principles of the Lab .....	269
7.1.5 Sample Programs .....	272
Exercises .....	275
7.2 uC/OS Application Lab.....	275
7.2.1 Purpose.....	275
7.1.2 Lab Equipment.....	275
7.1.3 Content of the Lab .....	275
7.1.4 Principles of the Lab .....	275
7.2.5 Sample Programs .....	277
7.2.6 Exercises .....	279
<b>Appendix A ARM Instruction, ARM Addressing and Thumb Instruction Quick Reference</b>	<b>287</b>
<b>Appendix B ARM and Thumb Instruction Code</b>	<b>288</b>
<b>Appendix C Embest ARM Related Products</b>	<b>289</b>
<b>Appendix D Content of CD-ROM</b>	<b>290</b>
<b>Reference Documentations</b>	<b>291</b>

# Chapter 1: An Overview of Embedded System Application Development

## 1.1 Embedded System Development and Applications

Embedded system based on embedded microprocessor is a new technology direction in IT technology.

ARM series processors are products from Advanced RISC Machine. The current ARM core includes ARM7TDMI, ARM720T, ARM9TDMI, ARM920T, ARM940T, ARM946T, ARM966T and Xscale, etc. Recently ARM Ltd. renounced to 4 ARM11 microprocessors (ARM1156T2-S, ARM1156T2F-S, ARM1176JZ and ARM11JZF-S). ARM chips are supported by many real time operating systems providers such as WindowsCE, uCLinux, VxWorks, Nucleus, EPOC, uc/OS, BeOS, Palm and QNX, etc.

## 1.2 An Overview of Embedded Development Environment for ARM

### 1.2.1 Cross Development Environment

Cross development means editing and compiling software on a general-purpose computer, and then downloading the software to the embedded device and debugging it on both, host and target. The general-purpose computer is called host. The embedded device is called target. Cross Development Environment consists of cross development software running on the host PC and the debug channel from host to target. There are three types of debug channels from host to target:

#### 1. The JTAG Based ICD

JTAG based ICD (In-Circuit Debugger) is also called JTAG Emulator. The JTAG Emulator connects to the target through the JTAG interface of the ARM processor and connects to the host through the serial port, the network port, or the USB port. JTAG Emulator has the following functions:

- Read/write CPU registers, visit and control ARM processor core.
- Read/write memory.
- Visit ASIC system.
- Visit I/O system.
- Single step execute program and real time execute program.
- Set break points.

JTAG Emulator is the most widely used debug method.

#### 2. Angel Debugging Software

Angel debug monitor software is a group of software programs running at the target board. It receives debug commands from host to set break points, single step execute programs, read/write memory, etc. Angel software is cheap and it doesn't need any other hardware debugging emulators. The inconvenient of this software is that it can be used only after the hardware is in a stable state.

#### 3. The In Circuit Emulator (ICE) is a CPU emulation device.

The ICE can completely emulate a target CPU such as the ARM processor and provides deeper debug functions.

Serial port, network port and USB port are also the communication channels of ICE. ICE can emulate high speed ARM processor. ICE is expensive and normally used in hardware development. It is seldom used in software development.

### 1.2.2 Software Emulator

Software Emulator can partly emulate the target hardware. It is normally an instruction set emulator. Software emulator can only be used as a primary debug tool because its function is limited and can't completely emulate the real hardware.

### 1.2.3 Evaluation Board

The evaluation board is also called a development board. It is useful for the developers. Experienced engineers can also make their own development board. A good development board has complete documentation, hardware and software implementations, schematic, sample programs, source code, etc for development references.

### 1.2.4 Embedded Operation System

Embedded real time operation system (RTOS) provides memory management, task management and resource management, etc. RTOS can save a lot of troubles in complicated applications. But if the application is not complicated, embedded systems can run without real time operation systems.

## 1.3 An Overview of ARM Development system

### 1.3.1 ARM SDT

ARM SDK is called ARM Software Development Kit. It is made by ARM Ltd. The latest version is 2.5.2. The highest ARM processors it can support are ARM9 series. ARM Ltd will not continue the ARM SDK in the future. The user interface of ARN SDK is shown in Figure 1-1 and Figure 1-2.

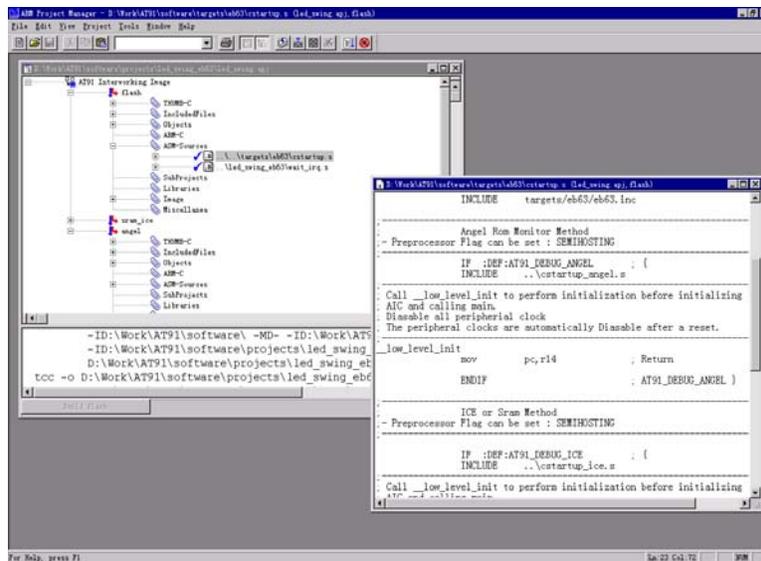


Figure 1-1 ARM Project Manager