

μC/OS-III™ *The Real-Time Kernel*

Jean J. Labrosse

Micrium
 **Press**

Weston, FL 33326

Micrium Press
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.micrium.com

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where Micrium Press is aware of a trademark claim, the product name appears in initial capital letters, in all capital letters, or in accordance with the vendor's capitalization preference. Readers should contact the appropriate companies for more complete information on trademarks and trademark registrations. All trademarks and registered trademarks in this book are the property of their respective holders.

Copyright © 2010 by Micrium Press except where noted otherwise. Published by Micrium Press. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher; with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

The programs and code examples in this book are presented for instructional value. The programs and examples have been carefully tested, but are not guaranteed to any particular purpose. The publisher does not offer any warranties and does not guarantee the accuracy, adequacy, or completeness of any information herein and is not responsible for any errors and omissions. The publisher assumes no liability for damages resulting from the use of the information in this book or for any infringement of the intellectual property rights of third parties that would result from the use of this information.

Library of Congress Control Number: 2010930936

Library of Congress subject headings:

1. Embedded computer systems
2. Real-time data processing
3. Computer software - Development

For bulk orders, please contact Micrium Press at: +1 954 217 2036

100-uCOS-III-NXP-LPC1768-001

ISBN: 978-0-9823375-5-4

Micrium
 **Press**

Table of Contents

Part I: μ C/OS-III – The Real-Time Kernel

	Foreword to μC/OS-III – by Jack Ganssle	19
	Preface	21
Chapter 1	Introduction	23
1-1	Foreground/Background Systems	24
1-2	Real-Time Kernels.....	25
1-3	RTOS (Real-Time Operating System).....	27
1-4	μ C/OS-III	27
1-5	μ C/OS, μ C/OS-II and μ C/OS-III Features Comparison	32
1-6	How the book is organized.....	34
1-7	μ C/Probe	34
1-8	Conventions	35
1-9	Chapter Contents.....	36
1-10	Licensing	40
1-11	Contacting Micrium.....	40
Chapter 2	Directories and Files	41
2-1	Application Code	44
2-2	CPU.....	45
2-3	Board Support Package (BSP)	46
2-4	μ C/OS-III, CPU Independent Source Code	47
2-5	μ C/OS-III, CPU Specific Source Code	51
2-6	μ C/CPU, CPU Specific Source Code	52
2-7	μ C/LIB, Portable Library Functions.....	54
2-8	Summary	55

Chapter 3	Getting Started with μ C/OS-III	59
3-1	Single Task Application	60
3-2	Multiple Tasks Application with Kernel Objects	68
Chapter 4	Critical Sections	77
4-1	Disabling Interrupts.....	78
4-1-1	Measuring Interrupt Disable Time.....	78
4-2	Locking the Scheduler	79
4-2-1	Measuring Scheduler Lock Time.....	80
4-3	μ C/OS-III Features with Longer Critical Sections.....	81
4-4	Summary	82
Chapter 5	Task Management	83
5-1	Assigning Task Priorities	92
5-2	Determining the Size of a Stack	94
5-3	Detecting Task Stack Overflows	95
5-4	Task Management Services	99
5-5	Task Management Internals	100
5-5-1	Task States.....	100
5-5-2	Task Control Blocks (TCBs).....	105
5-6	Internal Tasks	114
5-6-1	The Idle Task (OS_IdleTask())	115
5-6-2	The Tick Task (OS_TickTask())	117
5-6-3	The Statistic Task (OS_StatTask())	124
5-6-4	The Timer Task (OS_TmrTask())	127
5-6-5	The ISR Handler Task (OS_IntQTask())	128
5-7	Summary	129
Chapter 6	The Ready List	131
6-1	Priority Levels.....	132
6-2	The Ready List	136
6-3	Adding Tasks to the Ready List	139
6-4	Summary	140
Chapter 7	Scheduling.....	141
7-1	Preemptive Scheduling.....	142
7-2	Scheduling Points	144

7-3	Round-Robin Scheduling	146
7-4	Scheduling Internals	149
7-4-1	OSSched()	150
7-4-2	OSIntExit().....	151
7-4-3	OS_SchedRoundRobin()	152
7-5	Summary	154
Chapter 8	Context Switching.....	155
8-1	OSCtxSw()	158
8-2	OSIntCtxSw()	161
8-3	Summary	163
Chapter 9	Interrupt Management	165
9-1	Handling CPU Interrupts.....	166
9-2	Typical μ C/OS-III Interrupt Service Routine (ISR).....	167
9-3	Short Interrupt Service Routine (ISR)	170
9-4	All Interrupts Vector to a Common Location	171
9-5	Every Interrupt Vectors to a Unique Location	173
9-6	Direct and Deferred Post Methods	174
9-6-1	Direct Post Method.....	174
9-6-2	Deferred Post Method	177
9-7	Direct vs. Deferred Post Method.....	180
9-8	The Clock Tick (or System Tick).....	181
9-9	Summary	183
Chapter 10	Pend Lists (or Wait Lists)	185
10-1	Summary	190
Chapter 11	Time Management	191
11-1	OSTimeDly().....	192
11-2	OSTimeDlyHMSM()	197
11-3	OSTimeDlyResume()	199
11-4	OSTimeSet() and OSTimeGet()	200
11-5	OSTimeTick()	200
11-6	Summary	200

Chapter 12	Timer Management.....	201
12-1	One-Shot Timers.....	203
12-2	Periodic (no initial delay).....	204
12-3	Periodic (with initial delay).....	204
12-4	Timer Management Internals.....	205
12-4-1	Timer Management Internals - Timers States	205
12-4-2	Timer Management Internals - OS_TMR.....	206
12-4-3	Timer Management Internals - Timer Task.....	208
12-4-4	Timer Management Internals - Timer List.....	211
12-5	Summary	216
Chapter 13	Resource Management.....	217
13-1	Disable/Enable Interrupts.....	220
13-2	Lock/Unlock	222
13-3	Semaphores	223
13-3-1	Binary Semaphores.....	225
13-3-2	Counting Semaphores	232
13-3-3	Notes on Semaphores	234
13-3-4	Semaphore Internals (for resource sharing)	235
13-3-5	Priority Inversions	240
13-4	Mutual Exclusion Semaphores (Mutex)	242
13-4-1	Mutual Exclusion Semaphore Internals	247
13-5	Should You Use a Semaphore Instead of a Mutex?	253
13-6	Deadlocks (or Deadly Embrace).....	253
13-7	Summary	258
Chapter 14	Synchronization	259
14-1	Semaphores	260
14-1-1	Unilateral Rendezvous.....	262
14-1-2	Credit Tracking.....	265
14-1-3	Multiple Tasks Waiting on a Semaphore	267
14-1-4	Semaphore Internals (for synchronization).....	268
14-2	Task Semaphore	275
14-2-1	Pending (i.e., Waiting) on a Task Semaphore.....	276
14-2-2	Posting (i.e., Signaling) a Task Semaphore	277
14-2-3	Bilateral Rendezvous	279

14-3	Event Flags.....	281
14-3-1	Using Event Flags	283
14-3-2	Event Flags Internals	287
14-4	Synchronizing Multiple Tasks.....	294
14-5	Summary	296
Chapter 15	Message Passing.....	297
15-1	Messages	298
15-2	Message Queues	298
15-3	Task Message Queue	300
15-4	Bilateral Rendezvous	301
15-5	Flow Control	302
15-6	Keeping the Data in Scope.....	304
15-7	Using Message Queues.....	307
15-8	Clients and Servers.....	315
15-9	Message Queues Internals.....	316
15-10	Summary	319
Chapter 16	Pending On Multiple Objects.....	321
16-1	Summary	329
Chapter 17	Memory Management.....	331
17-1	Creating a Memory Partition	332
17-2	Getting a Memory Block from a Partition	336
17-3	Returning a Memory Block to a Partition.....	337
17-4	Using Memory Partitions	338
17-5	Summary	341
Chapter 18	Porting μ C/OS-III.....	343
18-1	μ C/CPU.....	346
18-2	μ C/OS-III Port.....	349
18-3	Board Support Package (BSP).....	351
18-4	Summary	353

Chapter 19	Run-Time Statistics.....	355
19-1	General Statistics – Run-Time.....	356
19-2	Per-Task Statistics – Run-Time.....	360
19-3	Kernel Object – Run-Time.....	363
19-4	OS_DBG.C – Static	366
19-5	OS_CFG_APP.C – Static.....	379
19-6	Summary	381
Appendix A	µC/OS-III API Reference Manual.....	383
A-1	Task Management	384
A-2	Time Management	386
A-3	Mutual Exclusion Semaphores – Resource Management	387
A-4	Event Flags – Synchronization	388
A-5	Semaphores – Synchronization.....	389
A-6	Task Semaphores – Synchronization.....	390
A-7	Message Queues – Message Passing	391
A-8	Task Message Queues – Message Passing	392
A-9	Pending on Multiple Objects	393
A-10	Timers.....	394
A-11	Fixed-Size Memory Partitions – Memory Management	395
	OSCtxSw()	396
	OSFlagCreate().....	398
	OSFlagDel()	400
	OSFlagPend()	402
	OSFlagPendAbort()	406
	OSFlagPendGetFlagsRdy().....	409
	OSFlagPost()	411
	OSIdleTaskHook()	413
	OSInit()	415
	OSInitHook()	417
	OSIntCtxSw()	418
	OSIntEnter()	420
	OSIntExit().....	421
	OSMemCreate().....	422
	OSMemGet()	425
	OSMemPut()	427
	OSMutexCreate().....	429
	OSMutexDel()	431

OSMutexPend()	433
OSMutexPendAbort()	436
OSMutexPost()	438
OSPendMulti()	440
OSQCreate()	444
OSQDel()	446
OSQFlush()	448
OSQPend()	450
OSQPendAbort()	454
OSQPost()	456
OSSafetyCriticalStart()	459
OSSched()	460
OSSchedLock()	462
OSSchedRoundRobinCfg()	464
OSSchedRoundRobinYield()	466
OSSchedUnlock()	468
OSSemCreate()	470
OSSemDel()	472
OSSemPend()	475
OSSemPendAbort()	478
OSSemPost()	480
OSSemSet()	483
OSStart()	485
OSStartHighRdy()	487
OSStatReset()	489
OSStatTaskCPUUsageInit()	490
OSStatTaskHook()	491
OSTaskChangePrio()	493
OSTaskCreate()	495
OSTaskCreateHook()	506
OSTaskDel()	508
OSTaskDelHook()	510
OSTaskQPend()	512
OSTaskQPendAbort()	515
OSTaskQPost()	517
OSTaskRegGet()	520
OSTaskRegSet()	522
OSTaskReturnHook()	524

	OSTaskResume()	526
	OSTaskSemPend()	528
	OSTaskSemPendAbort()	531
	OSTaskSemPost()	533
	OSTaskSemSet()	535
	OSTaskStatHook()	537
	OSTaskStkChk()	539
	OSTaskStkInit()	542
	OSTaskSuspend()	546
	OSTaskSwHook()	548
	OSTaskTimeQuantaSet()	551
	OSTickISR()	553
	OSTimeDly()	555
	OSTimeDlyHMSM()	558
	OSTimeDlyResume()	561
	OSTimeGet()	563
	OSTimeSet()	564
	OSTimeTick()	566
	OSTimeTickHook()	567
	OSTmrCreate()	569
	OSTmrDel()	575
	OSTmrRemainGet()	577
	OSTmrStart()	579
	OSTmrStateGet()	581
	OSTmrStop()	583
	OSVersion()	585
Appendix B	μC/OS-III Configuration Manual	587
B-1	μC/OS-III Features (OS_CFG.H)	590
B-2	Data Types (OS_TYPE.H)	600
B-3	μC/OS-III Stacks, Pools and other (OS_CFG_APP.H)	601
Appendix C	Migrating from μC/OS-II to μC/OS-III	607
C-1	Differences in Source File Names and Contents	610
C-2	Convention Changes	613
C-3	Variable Name Changes	619
C-4	API Changes	620

C-4-1	Event Flags.....	621
C-4-2	Message Mailboxes	623
C-4-3	Memory Management.....	625
C-4-4	Mutual Exclusion Semaphores.....	626
C-4-5	Message Queues	628
C-4-6	Semaphores	630
C-4-7	Task Management	632
C-4-8	Time Management	636
C-4-9	Timer Management.....	637
C-4-10	Miscellaneous	639
C-4-11	Hooks and Port	641
Appendix D	MISRA-C:2004 and μ C/OS-III	645
D-1	MISRA-C:2004, Rule 8.5 (Required)	646
D-2	MISRA-C:2004, Rule 8.12 (Required)	646
D-3	MISRA-C:2004, Rule 14.7 (Required)	647
D-4	MISRA-C:2004, Rule 15.2 (Required)	648
D-5	MISRA-C:2004, Rule 17.4 (Required)	649
Appendix E	Bibliography	651
Appendix F	Licensing Policy	653

Part II: μ C/OS-III and the NXP LPC1768

	Foreword by Geoff Lees and Reinhard Keil.....	657
Chapter 1	Introduction	659
1-1	The KEIL MCB1700.....	660
1-2	Part II Chapter Contents.....	663
1-3	Acknowledgements	665

Chapter 2	The ARM Cortex-M3 and the NXP LPC1768	667
2-1	The Cortex CPU	668
2-1-1	The Program Status Register	670
2-1-2	Stacking and Interrupts	670
2-2	Nested Vector Interrupt Controller (NVIC)	671
2-3	Exception Vector Table.....	672
2-4	SysTick (System Tick).....	673
2-5	Memory Map	674
2-6	Instruction Set.....	675
2-7	Debugging Features	676
2-8	The NXP LPC1768.....	677
Chapter 3	Setup	679
3-1	Downloading Keil μ Vision	680
3-2	Downloading Keil MCB1700 and LPC17xx Documentation	681
3-3	Downloading μ C/Probe	681
3-4	μ C/OS-III Library	682
3-5	μ C/TCP-IP, μ C/DHCPc and μ C/TFTP Library.....	683
3-6	μ C/FS Library	684
3-7	μ C/OS-III Projects for this Book.....	684
3-7-1	Example Projects Folders.....	685
3-8	Connecting the Keil MCB1700 Board to a PC.....	688
3-8-1	Power Supply & Debugging.....	688
3-8-2	Serial Ports.....	688
3-8-3	μ C/Probe Interface.....	689
3-8-4	Ethernet Connection	689
Chapter 4	DMA-Based LED Blinking.....	691
4-1	Example Description.....	691
4-2	Opening and Running the Project.....	693
4-3	Running μ C/Probe	695
4-4	How the Example Project Works.....	700
4-5	Summary	708

Chapter 5	LCD Display.....	709
5-1	Opening and Running the Project.....	710
5-2	How the Example Project Works.....	710
5-2-1	Graphical User Interface (GUI) Implementation.....	711
5-2-2	Font and Image Support.....	716
5-3	Summary	717
Chapter 6	Ringtone Player.....	719
6-1	Opening and Running the Project.....	720
6-2	How the Example Project Works.....	721
6-2-1	Requirements.....	721
6-2-2	Ringtone Player Architecture	724
6-2-3	Audio Manager.....	726
6-2-4	Implementation of the Audio Manager on the LPC17xx.....	729
6-2-5	RingTone Files	733
6-3	Analysis with μ C/Probe.....	733
6-4	Audio Signal Analysis	735
6-4-1	Sample Rate.....	736
6-4-2	Total Harmonic Distortion.....	738
6-4-3	Note Frequency and Note Duration	739
6-5	Summary	740
Chapter 7	Playing Audio Wave Files	741
7-1	Opening and Running the Project.....	742
7-1-1	Application Modes.....	743
7-1-2	Playing the Songs	743
7-1-3	Updating the microSD card.....	743
7-2	How the Example Project Runs.....	746
7-2-1	Wave File Format	747
7-3	μ C/Probe	750
Appendix A	μ C/OS-III Port for the Cortex-M3	753
A-1	OS_CPU.H	754
A-2	OS_CPU_C.C.....	756
A-2-1	OS_CPU_C.C – OSIdleTaskHook()	756
A-2-2	OS_CPU_C.C – OSInitHook().....	757
A-2-3	OS_CPU_C.C – OSStatTaskHook()	758

A-2-4	OS_CPU_C.C – OSTaskCreateHook()	759
A-2-5	OS_CPU_C.C – OSTaskDelHook().....	760
A-2-6	OS_CPU_C.C – OSTaskReturnHook()	761
A-2-7	OS_CPU_C.C – OSTaskStkInit()	762
A-2-8	OS_CPU_C.C – OSTaskSwHook()	765
A-2-9	OS_CPU_C.C – OSTimeTickHook()	768
A-2-10	OS_CPU_C.C – OS_CPU_SysTickHandler().....	769
A-2-11	OS_CPU_C.C – OS_CPU_SysTickInit().....	770
A-3	OS_CPU_A.ASM.....	770
A-3-1	OS_CPU_A.ASM – OSStartHighRdy()	771
A-3-2	OS_CPU_A.ASM – OSCtxSw() and OSIntCtxSw()	772
A-3-3	OS_CPU_A.ASM – OS_CPU_PendSVHandler()	773
Appendix B	µC/CPU Port for the Cortex-M3	775
B-1	CPU_CORE.C	775
B-2	CPU_CORE.H	776
B-3	CPU_DEF.H	776
B-4	CPU_CFG.H.....	776
B-5	µC/CPU Functions in BSP.C	777
B-5-1	µC/CPU Functions in BSP.C, CPU_TS_TmrInit()	778
B-5-2	µC/CPU Functions in BSP.C, CPU_TS_TmrRd().....	778
B-6	CPU.H	779
B-6-1	CPU.H – #defines	779
B-6-2	CPU.H – Data Types	780
B-6-3	CPU.H – Function Prototypes	783
B-7	CPU_A.ASM	783
Appendix C	Micrium’s µC/Probe	785
C-1	µC/Probe is a Windows™-Based Application.....	788
C-2	Assigning a Variable to an Object	790
Appendix D	Debugging with ARM CoreSight Technology on Cortex-M3 Processors.....	793
D-1	Serial Wire Viewer (SWV)	794
D-1-1	DAP: Real-Time Memory Read and Write.....	795
D-1-2	Embedded Trace Macrocell (ETM).....	796

D-1-3	Breakpoints and Watchpoints.....	796
D-2	What Kind of Data can the Serial Wire Viewer Display?	797
D-3	Debug Ports: JTAG, SWD and SWJ.....	797
D-4	Hardware Connections: Connecting to a Target.....	798
D-5	Usefulness of the Trace.....	799
Appendix E	Schematics.....	801
Appendix F	Bibliography	809
Appendix G	Licensing Policy	811
Index	813

Table of Contents

Chapter

1

Introduction

Part II of this book delivers to the reader the experience of $\mu\text{C}/\text{OS-III}$ through the use of world-class tools and step-by-step instruction.

The book is packaged with the ARM/Keil MCB1700 evaluation board. The MCB1700 Evaluation Board is available in two different configurations:

- MCB1750 Evaluation board which contains an NXP LPC1758, Cortex-M3 based Microcontroller.
- MCB1760 Evaluation board which contains an NXP LPC1768, Cortex-M3 based Microcontroller.

There are also two revisions of the board: Rev. 1 and Rev 1.2. In revision 1.2 only minor changes have been made that have no influence on examples and applications. Here are the changes in detail:

- Different USB host connector. Because of a mistake in the layout of revision 1, it was necessary to use an upright version of USB host connector. Revision 1.2 has the intended angular version populated.
- Two 0 Ohm resistors R82 and R83 have been added to be able to disconnect the USB interface from pin P1.22 and P1.19 in case these pins are not used for USB host.
- About half of all holes of the port headers were too small by mistake.

To build the example code provided in Part II of this book, you will need to download the Keil Microcontroller Development Kit (MDK-ARM), which enables you to build applications up to 32 Kbytes in size (see Chapter 3, “Setup” on page 679). You can also download the award-winning $\mu\text{C}/\text{Probe}$ from the Micrium website to monitor and change application variables at run time (see Chapter 3, “Setup” on page 679).

The heart of the LPC1700 family is the ARM Cortex-M3 CPU, one of the most popular CPU cores on the market today. The Cortex-M3 runs the very efficient ARMv7 instruction set.

The LPC1758/68 runs at clock frequencies up to 100 MHz and contains such high-performance peripherals as a 10/100 Ethernet MAC, full-speed USB 2.0 (Host/Device/OTG), timers, UARTs, and more. The LPC1758/68 also features built-in flash memory of 512 Kbytes, and 64 Kbytes of high-speed static RAM.

1-1 THE KEIL MCB1700

Figure 1-1 shows a picture of the Keil MCB1700. Board features include:

- Low cost
- 100 MHz LPC1758/68 Cortex-M3 based microcontroller
- On-chip memory: 512KB flash memory, 64KB RAM
- 10/100 Ethernet connector
- Color QVGA TFT LCD
- SD/MMC card socket
- USB 2.0 full speed host/device/OTG
- Dual CAN Interfaces
- Dual RS-232C serial ports
- 5-position joystick and push-button
- Potentiometer for analog-to-digital converter (ADC) input
- 70 GPIO pins
- LEDs
- Debug interface connectors (20-pin JTAG, 10-pin Cortex debug, 20-pin Cortex debug + ETM trace)

The LPC1758/68 is capable of running a TCP/IP stack such as Micrium's high-quality μ C/TCP-IP. This allows applications to network with other devices, as well as access the Internet. Other application protocols allow applications to run a DHCP client (to obtain an IP address), FTP client/server, HTTP server (i.e., a web server), SMTP client (to send e-mails), POP3 client (to receive e-mails), and more. μ C/TCP-IP can also be used as an interface to μ C/Probe.

The LPC1758/68 is also capable of running USB stacks such as Micrium's μ C/USB-Device, and μ C/USB-Host. The USB controller is a full-speed device, and can transfer data up to 12 Mbps. The MCB1700 can thus act as a USB device or a USB host. As a device, the evaluation board can act as a Human Interface Device (HID) or a Mass Storage Device (MSD), especially with the on-board SD card. In other words, the MCB1700 can appear as a disk drive to a PC. As a USB host, the MCB1700 interfaces to USB memory sticks, allowing for the storage and retrieval of data.

The included microSD connector allows applications to run a file system such as Micrium's μ C/FS to save and retrieve contents onto a Micro Secure Digital (microSD) card. The microSD card can be used to log data to a file, which can be read from a PC (assumes USB Device, MSD, or an FTP client or server). Note that the microSD card is not provided with the MCB1700 evaluation board.

The two RS-232C connectors allow an application to output information to a terminal (or a terminal emulator). One of the two RS-232C interfaces can also be used with μ C/Probe.

Appendix E contains the complete schematics for the MCB1700.

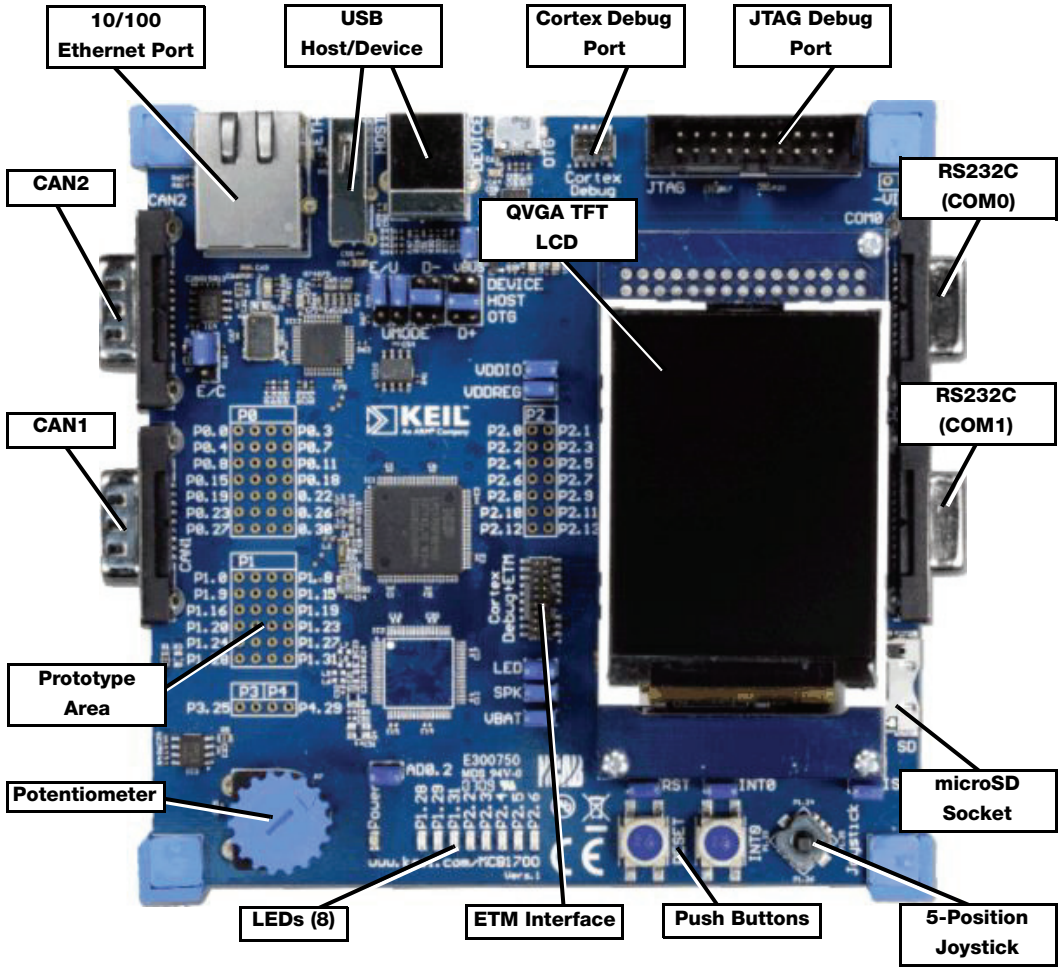


Figure 1-1 Keil MCB1700 Evaluation Board

1-2 PART II CHAPTER CONTENTS

Chapter 1, The ARM Cortex-M3 and the LPC1768. This chapter provides a brief introduction to the ARM Cortex-M3 CPU and its implementation in the LPC1768.

Chapter 2, Setup. The chapter explains how to set up the test environment to run $\mu\text{C}/\text{OS-III}$ examples. The chapter describes how to download the 32K version of the Keil MDK-ARM tool chain, how to obtain example code that accompanies the book, and how to connect the MCB1700 evaluation board to a PC.

Chapter 3, DMA-Based LED Blinking. This chapter explains how to get $\mu\text{C}/\text{OS-III}$ up and running. This first example will perform a modified version of the classical ‘blink a light’ test. After 10 seconds the application sets the CPU in sleep mode. The LPC1768’s sleep mode halts the clock to the CPU’s core and powers down the on-chip flash memory but the peripheral clocks and the SRAM memory module remain powered.

DMA is used to transfer data that is located in SRAM to the LEDs (which are connected to GPIOs on the MCB1700 evaluation board) while the CPU is in sleep mode. Since the CPU is in sleep mode, DMA requests are made by Timer 0 at a rate of 1Hz. An external interrupt (‘INT0’ push button) brings the CPU back to normal mode and the execution is resumed.

Chapter 4, LCD Display. This chapter describes how to create a simple Graphical User Interface (GUI) to be used in a multitask environment.

Chapter 5, Ringtone Player. In this chapter we implement a Ringtone Player using the MCB1700’s speaker and the LPC17xx’s digital-to-analog converter (DAC). The ringtones are stored onto the LPC17xx’s on-chip flash memory in a Ring Tone Text Transfer Language Format (RTTTL). A simple user interface is created to allow the user to select and play a specific ringtone from a playlist.

The example requires a tone generator that creates a sine wave to be output to the DAC. The audio is sampled at 44,100 Hz and, instead of using a timer to generate an interrupt every 22.67 μs , the LPC17xx’s DMA is used to eliminate the overhead of an interrupt.

The example also shows how audio quality is reduced when the application’s task priorities are not properly assigned.

Chapter 6, Playing Audio Wave Files. This chapter describes the implementation of a simple media player. The media player will play files in WaveForm Audio file format (WAVE) from a microSD card. The audio will be played through the LPC17xx DAC and the MCB1700's speaker.

Files can be uploaded into the microSD card using the Trivial File Transfer Protocol (TFTP) through an Ethernet connection. Note that the microSD card is not provided with the MCB1700 evaluation board and you will thus need to acquire such a card from your favorite computer part supplier.

Appendix A, μ C/OS-III Port for the Cortex-M3. This appendix explains how μ C/OS-III was adapted to the Cortex-M3 CPU. The Cortex-M3 contains interesting features specifically designed for real-time kernels, and μ C/OS-III makes good use of these.

Appendix B, μ C/CPU Port to the Cortex-M3. This appendix describes how μ C/CPU was adapted to the Cortex-M3 processors.

Appendix C, μ C/Probe. This appendix provides a brief description of Micrium's award-winning μ C/Probe, which easily allows users to display and change target variables at run time.

Appendix D, Debugging with ARM CoreSight Technology on Cortex-M3 Processors. This appendix describes the debug components in the ARM CoreSight as implemented in the NXP LPC1768.

Appendix E, MCB1700 Schematics. This appendix provides the full schematics for the Keil MCB1700 evaluation board.

Appendix F, Bibliography.

Appendix G, Licensing μ C/OS-III.

1-3 ACKNOWLEDGEMENTS

I would like to thank Mr. Bob Boys from ARM/Keil for his support and help on this project. His knowledge and insight of the Cortex-M3 architecture and the Keil tools have proved to be invaluable.

Thank you to Mr. Kenneth Dwyer and the fine folks at NXP for their help and for working with Micrium to make this book a reality.

Finally, thank you to my great team at Micrium for the help and support they provided for this project.

Chapter 1
