RT Real time operating system for ARM/7

© 2006 Enter your company name

Note:
To change the product logo for your own print manual or PDF, click "Tools > Manual Designer" and modify the print manual template.

Title page 1

Use this page to introduce the product

by Enter your company name

This is "Title Page 1" - you may use this page to introduce your product, show title, author, copyright, company logos, etc.

This page intentionally starts on an odd page, so that it is on the right half of an open book from the readers point of view. This is the reason why the previous page was blank (the previous page is the back side of the cover)

RT Real time operating system for ARM/7

© 2006 Enter your company name

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: dicembre 2006 in (whereever you are located)

Publisher

...enter name...

Managing Editor

...enter name...

Technical Editors

...enter name...
...enter name...

Cover Designer

...enter name...

Team Coordinator

...enter name...

Production

...enter name...

Special thanks to:

All the people who contributed to this document, to mum and dad and grandpa, to my sisters and brothers and mothers in law, to our secretary Kathrin, to the graphic artist who created this great product logo on the cover page (sorry, don't remember your name at the moment but you did a great work), to the pizza service down the street (your daily Capricciosas saved our lives), to the copy shop where this document will be duplicated, and and and...

Last not least, we want to thank EC Software who wrote this great help tool called HELP & MANUAL which printed this document.

Table of Contents

Foreword 1

Part I Introduction 3

1 Welcome topic 3

Feature 3

ARM7 LPC2000 Porting 3

8051 Porting 3

Z80 Porting 4

2 Configuration 4

Parameter 4

3 Installation 4

Directories structure 4

Source 5

Project 5

Part II System initialization 7

1 LPC2000Init 7

2 MCS51Init 7

Part III Thread 9

1 Priorities 9

2 Function 9

RTThreadCreate 9

RTThreadSleep 9

Index 0

© 2006 Enter your company name

Foreword

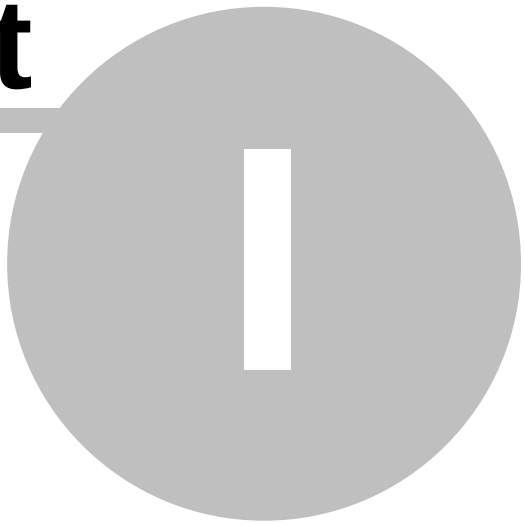
This is just another title page
placed between table of contents
and topics

© 2006 Enter your company name

Top Level Intro

This page is printed before a new
top-level chapter starts

Part



1 Introduction

The documentation is not completed but for now I'm working to implement new functionality or to correct bug i hope in the future to have more time to complete this manual.
for now the better documentation are the source files ....

1.1 Welcome topic

RT is a free realtime operating system specially designed for embedded system. The code is written in ansi c and the hardware depending part is small and well isolated in this way it is simple port the operating system for new micro controller or architecture.

1.1.1 Feature

Principal feature for the RT operating system.

- Preemptive scheduler
- Adjustable priorities for any thread.
- Support for queue and messages.
- Support for semaphore
- Thread pause/resume.
- Virtual console support
- Console support for serial port
- Simple monitor over any console.

1.1.1.1 ARM7 LPC2000 Porting

- GNU ARM Compiler
- Serial support interrupt driver
- Interrupt controller support (VIC)
- Timer support
- PLL configuration
- Tested on KEIL MCB2140 board

1.1.1.2 8051 Porting

The 8051 porting is based upon SDCC compiler version 2.6 and is tested using the J51 emulator. The model used for the compiler is large siwth parameter passing in the stack, this version of library is not present in the standard distribution but in the directories sdcc is present a simple make file to compile the library .

Supported peripheral :
- Timer 0 used for periodic interrupt
- Serial port used for the console

The main function very the main is defined must declare the following symbol CONFIG_RT_8051_MAIN because with the current version of SDCC compiler the interrupt function cannot be defined in the library but only in the main.

© 2006 Enter your company name

Special care must be used for stack usage due to the small availability of internal ram.

1.1.1.3 Z80 Porting

Not yet available.

1.2 Configuration

The configuration of RTOS is made thru one set of parameter defined in the source file. All parameter have a default value.

1.2.1 Parameter

CONFIG_RT_PROFILE			
CONFIG_RT_PROFILE	If defined thread execution and interrupt service are compiled with a simple profile system.	1	1
CONFIG_RT_MAX_INTERRUPT	Maximum number of interrupt service routine.	16	16
CONFIG_RT_MAX_MEMORY_REGION	Maximum number of memory region.	4	1
CONFIG_RT_THREAD_TICKLEN	Periodic interrupt time in ms. Define the tick len for the system.	1	10
CONFIG_RT_THREAD_SLICE	Default thread slice in number of tick.	10	10
CONFIG_RT_UART_SIZETX	Uart default tx buffer size.	32	32
CONFIG_RT_UART_SIZERX	Uart default rx buffer size.	100	100
CONFIG_RT_UART_BAUD	Uart default baud rate.	115200	115200
CONFIG_RT_FORMAT_MSX	If defined the RTFormat function will be compiled using static variable. This can be necessary for the 8051.	0	1
CONFIG_RT_CLOCK	System clock.	12000000	12000000

1.3 Installation

Installation for RT is very simple it is sufficient extract all the file in one directories and set a variable called RT with the root where the file are installed.

1.3.1 Directories structure

After the installation the directories structure reflect this model. In this example the RT is instyalled in one windows machine in the directories **C:\RT.**

General	
Directories	Description
C:\RT	RT Home installation
C:\RT\EFSL--0.2.8	Embedded file system library in this example version -0.2.8
C:\RT\RT	RT source
C:\RT\RT\SRC	Source files

© 2006 Enter your company name

C:\RT\RT\INC	Include files
C:\RT\OBJ	Home for obj

1.3.1.1 Source

General	
Directories	Description
C:\RT	RT Home installation
C:\RT\EFSL--0.2.8	Embedded file system library in this example version -0.2.8
C:\RT\RT	RT source
C:\RT\EXAMPLES	RTOS Examples
C:\RT\OBJ	Home for obj

1.3.1.2 Project

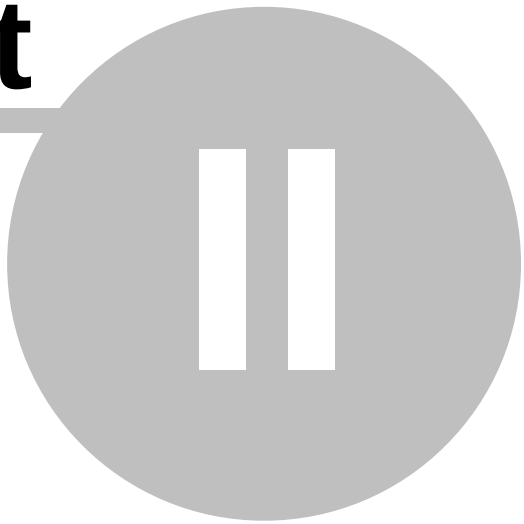
All project have one source directory EX. C:\RT\EXAMPLES\MCB2148 is the source directories for the KEIL MCB2148 board examples and one object directories under C:\RT\OBJ.

© 2006 Enter your company name

Top Level Intro

This page is printed before a new
top-level chapter starts

Part



2 System initialization

System initialization are function depending on the porting.

2.1 LPC2000Init

```
void LPC2000Init(int mainPriority,int stackMain,int stackIrq);
```

Initialization for the LPC2000 ARM7 porting. The system will be initialized and the main thread will be created with the specified priority. The stackMain specify the stack size in byte for the main thread and stackIrq specify the stack size for the interrupt service routine.

2.2 MCS51Init

```
void MCS51Init(unsigned char mainPriority)
```

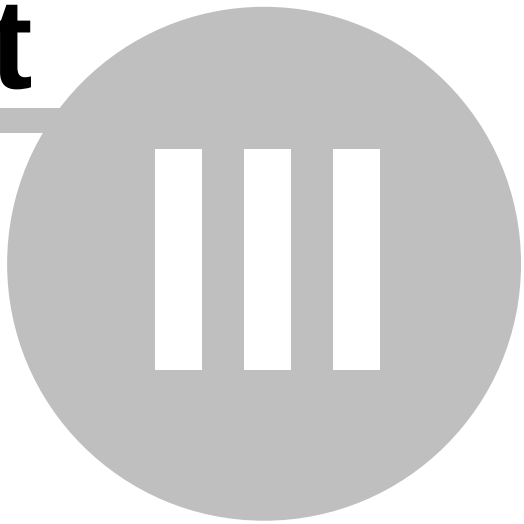
System initialization for 8051 porting.

© 2006 Enter your company name

Top Level Intro

This page is printed before a new
top-level chapter starts

Part



3 Thread

The thread are the core of the RT operating system during the initialization two thread will be created the main (the ca

3.1 Priorities

Any thread can have different priorities and the system run the thread with the high priority. Priority varies from 1 (Maximum priority) to 254 (Minimum priority) the priority 0 and 254 are reserved for internal usage. Constants are provided for thread priority control

Constant name	Value	Description
RT_MAX_PRIORITY	1	Maximum priority.
RT_ABOVE_NORMAL	96	'Above' normal priority.
RT_NORMAL_PRIORITY	128	Normal thread priority.
RT_BELOW_PRIORITY	160	'Below' normal priority.
RT_MIN_PRIORITY	254	Minimum priority.

For fine control of thread priority the constants can be incremented or decremented for example if the thread A have the priority RT_NORMAL_PRIORITY the thread B can be created with priority RT_NORMAL_PRIORITY - 1.

3.2 Function

```
void RTThreadCreate(RT_TCB_T *t,unsigned char priority,char *name,void (* main)(void *),void *arg,int ssize
```

Create a new thread. The thread will be created and started immediately with the specified priority.

```
void RTThreadSleep(unsigned long delay);
```

The current thread sleep the specified number of ms.

3.2.1 RTThreadCreate

```
void RTThreadCreate(RT_TCB_T *t,unsigned char priority,char *name,void (* main)(void *),void *arg,int ssize
```

Create a new thread. The thread will be created and started immediately with the specified priority.

3.2.2 RTThreadSleep

```
void RTThreadSleep(unsigned long delay);
```

The current thread sleep the specified number of ms.

© 2006 Enter your company name

Endnotes 2... (after index)

© 2006 Enter your company name

Back Cover