Excellent Integrated System Limited

Stocking Distributor

Click to view price, real time Inventory, Delivery & Lifecycle Information:

Zilog
EZ800000100KRS

For any questions, you can email us directly:
sales@integrated-circuit.com

Distributor of Zilog: Excellent Integrated System Limited
Datasheet of EZ800000100KRS - SOFTWARE REAL TIME KERNEL
Contact us: sales@integrated-circuit.com Website: www.integrated-circuit.com

eZ80Acclaim™ Family of Microcontrollers

ZiLOG Real-Time Kernel

Version 1.2.0

Product Brief

PB015503-0305 PRELIMINARY

Introduction

The ZiLOG Real-Time Kernel (RZK) is a real-time, preemptive, multitasking kernel designed for time-critical embedded applications. It is currently available for use with ZiLOG's eZ80Acclaim™ product line. RZK is supplied as a C library module.

Features

This version of RZK includes the following features, which are described on the following pages:

- Fast context switching
- Small memory footprint
- Modular design
- Configurable compile options
- Portable design
- Interrupt handling
- Threads with 32 priority levels
- Support for priority inheritance
- Interprocess communication, including semaphore, message queue, and event groups
- Software timer
- Memory management
- Device Driver Framework, including several drivers
- RZK file system

These features are tailored to the requirements of typical 8-bit applications. A description of these features follows.

Fast Context Switching

RZK can context-switch between threads in approximately 12 µs (measured on an eZ80F91 MCU operating with 1 wait state at 50 MHz). This interval includes the time required to identify the next ready-to-run thread as well as the time required to save and restore the context of the two threads, resulting in very low system overhead. Context-switching time exhibits negligible variation as the number of threads increases.

Small Memory Footprint

RZK offers a very low memory footprint. The basic kernel and thread occupies only 9 KB of ROM. Other objects can be optionally linked in. Additional objects are optimized for minimum memory requirements—the full version of RZK occupies about 21 KB of ROM. The minimum RAM required is only about 1 KB.

Modular Design

The modular design of RZK allows minimal footprint increase when using the object library. The RZK binary image file is prepared in such a way so that only objects required by the application are linked. Interdependency between modules is very minimal. Therefore, only the objects that are used in the application are included in the final downloadable image.

Configurable Compilation

System parameters, such as the number of objects required (threads, semaphores, etc.) can be specified at compile time. RZK allocates memory for object control blocks based on the number of objects specified in the configuration file. Features such as debug support and priority inheritance support can be enabled or disabled at compile time. Additionally, the duration of the RZK system timer

Distributor of Zilog: Excellent Integrated System Limited
Datasheet of EZ800000100KRS - SOFTWARE REAL TIME KERNEL
Contact us: sales@integrated-circuit.com Website: www.integrated-circuit.com

tick is configurable. These compile-time configuration options make RZK fully configurable, thus enabling the developer to fine-tune RZK to application requirements.

Portable Design

Most of RZK is written in C. To boost RZK performance, a few of the critical routines are written in Assembly. Only these files must be changed when RZK is ported to a new platform. As a result, porting operations are minimized.

Interrupt Handling

RZK manages all interrupt-related tasks. It provides APIs to install an interrupt service routine (ISR) for a particular peripheral device, and to conditionally enable and disable interrupts. The interrupt-handling mechanism allows ISR execution in a separate thread so as to minimize worst-case system interrupt latency. Worst-case system interrupt latency for RZK is approximately 30 microseconds.

Threads with 32 Priority Levels

RZK provides preemptive and nonpreemptive threads. The threads can either preempt other threads based on priority, or share the CPU cooperatively among equal-priority threads using round-robin scheduling. Thread priority can be changed at run-time.

Support for Priority Inheritance

RZK supports a priority inheritance protocol to solve a priority inversion problem common in binary semaphores. The low-priority blocked thread inherits the priority of the highest-priority thread blocked on that semaphore. RZK provides solutions for both unbounded and bounded priority inversion.

Interprocess Communication

RZK offers various mechanisms for communication and synchronization between threads, as described below.

Semaphore

RZK provides both binary and counting semaphore support. A priority inheritance protocol is used to emerge from deadlock scenarios.

Message Queue

RZK provides message queues for asynchronous communication between threads. It supports buffered message copying. High-priority messages can be placed at the front of the queue. Threads can wait on the queue based on their priority, or FIFO.

Event Groups

Event Groups are provided for control synchronization and do not contain any information. An event group can accommodate a maximum of 24 events. This mechanism is very useful for synchronizing multiple threads based on specific events.

Software Timer

RZK provides software timers to invoke tasks at periodic intervals. RZK maintains a system tick with the help of a hardware timer device that enables RZK threads to run the software timer, perform timed blocks on resources, and support clock functions.

Memory Management

Fixed-size memory pools are provided for deterministic memory allocation, which is very useful for time-critical applications. RZK also supports variable-size memory allocation with features such as finite and infinite blocking.

RZK Device Driver Kit

A device driver framework is provided to support the development of drivers for most of the peripherals on the eZ80Acclaim™ product line. This support includes a board support package consisting of drivers for UART, EMAC, SPI, I^2C, and RTC in all eZ80Acclaim™ and eZ80L92 platforms. It also provides the drivers for different NOR Flash devices that provide simple-to-use APIs for interacting with devices equipped with Flash memory.

Distributor of Zilog: Excellent Integrated System Limited
Datasheet of EZ800000100KRS - SOFTWARE REAL TIME KERNEL
Contact us: sales@integrated-circuit.com Website: www.integrated-circuit.com

ZiLOG File System

This release includes a file system to manage both RAM and Flash media on eZ80Acclaim™ platforms. Using this file system, the application developer can store files in RAM or Flash media, then perform any file-related operation such as read, write, or append to the stored files. The ZiLOG File System also provides full-fledged directory operations, along with power-failure mode support.

Network Services

A full fledged TCP/IP stack has been developed using RZK, which offers basic as well as advanced networking features for low-cost embedded networking applications.

Architecture

The ZiLOG Real-Time Kernel architecture follows a layered approach. The lowest layer is the RZK Core, which provides basic kernel services. Above the core layer are the RZK objects, which are used by the application via an application programming interface (API).

RZK objects use the kernel services of the RZK Core for resource management. These kernel services provide a set of easy-to-use APIs that allow the end-user application to manage different activities, as shown in Figure 1.

The Resource Queue Manager keeps track of threads waiting for resources, and the Time Queue Manager provides facilities such as the Timed Wait and Sleep functions. The Scheduler schedules and dispatches threads based on a scheduling policy. The RZK Core also features hardware-dependent files that are specific to particular architectures.

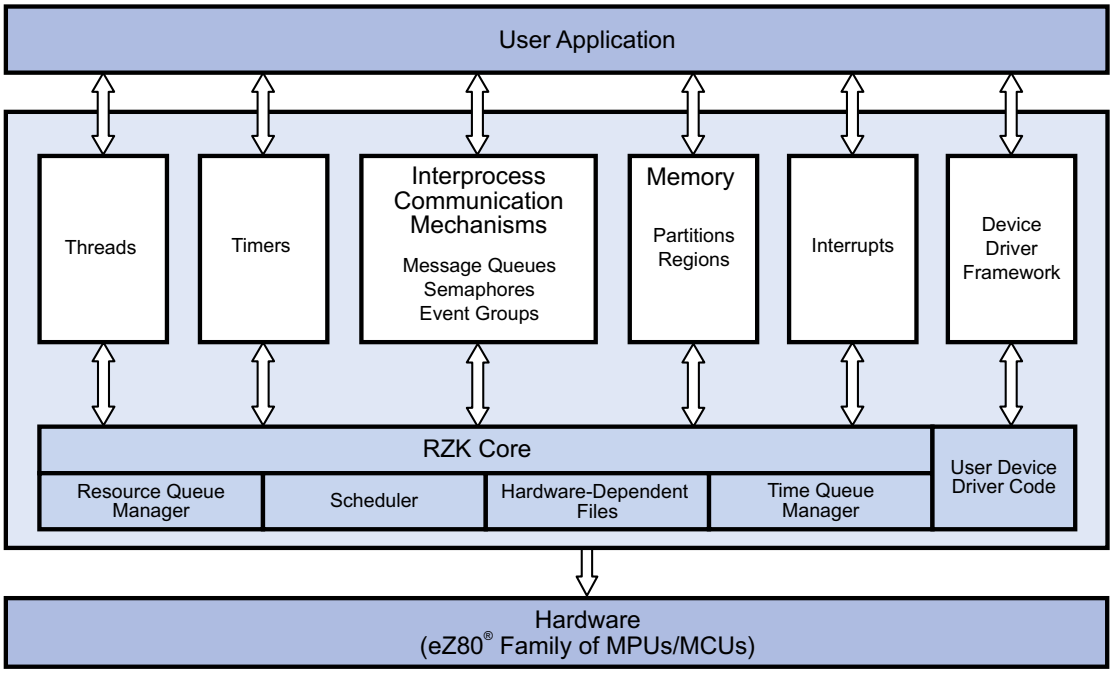

Figure 1. ZiLOG Real-Time Kernel Architecture

Distributor of Zilog: Excellent Integrated System Limited
Datasheet of EZ800000100KRS - SOFTWARE REAL TIME KERNEL
Contact us: sales@integrated-circuit.com Website: www.integrated-circuit.com

The RZK Object Layer library comprises a number of optional RZK objects, such as threads, semaphores, and memory management. The design of individual modules inside the library is extremely modular. Unless and until an application specifically uses an RZK object, it is not included in the final binary image. This modular approach reduces the final footprint of your application. RZK services can be accessed by an application via a set of easy-to-use APIs. The layered architecture of RZK makes it scalable, and new features can be easily plugged in.

Version 1.2.0 of RZK provides improved performance through a new interrupt model, as well as a Device Driver Framework (DDF) that includes drivers for a Flash file system and several board interfaces (UART, EMAC, SPI, I^2C, and RTC).

Examples Provided

The standard RZK package provides several sample applications that demonstrate various features of the RZK for easy reference.

Router Example

This application simulates a data router using relevant RZK objects to demonstrate the routing of packets and messages. The following RZK features are shown:

- Thread communication with message queues
- Thread synchronization with semaphores and event groups
- Use of timers
- Use of memory partitions

Interrupt Example

This simple application demonstrates the use of an interrupt used within the RZK environment to synchronize and cause the display of a message by an idle thread.

This package also contains examples that demonstrate the use of the UART and RTC drivers. It also contains the interactive ZiLOG File System shell that implements file and directory operations that are similar to DOS commands, such as md, cd, dir, del, deldir, deltree, and more.

Development Tools

RZK is supported by the ZiLOG Developer Studio Integrated Development Environment (ZDS II), which provides compiling, debugging, and project-building tools for the quick and efficient development of embedded applications. ZDS II is included in all eZ80Acclaim™ development kits. RZK is also compatible with the IAR Embedded Workbench, a third party C/C++ tool chain.

Packaging

The standard RZK package for eZ80Acclaim™ microcontrollers is supplied as a C object library module plus application examples.

Documentation

The following documents describe the features, functions, and usage of the ZiLOG Real-Time Kernel and are available in each RZK kit.

ZiLOG Real-Time Kernel v1.2 Quick Start Guide (QS0048). Enables the user to install the RZK software and quickly get up and running. It guides the user through a sample application.

ZiLOG Real-Time Kernel v1.2 User Manual (UM0075). Offers a detailed explanation of the different RZK configurations and sample applications. It also contains a section that analyzes RZK performance numbers and FAQs.

ZiLOG Real-Time Kernel v1.2 Reference Manual (RM0006). Provides a comprehensive explanation of the APIs and data structures provided by RZK.

Distributor of Zilog: Excellent Integrated System Limited
Datasheet of EZ800000100KRS - SOFTWARE REAL TIME KERNEL
Contact us: sales@integrated-circuit.com Website: www.integrated-circuit.com

Ordering Information

RZK Package	Part Number	Description	Support
Standard Release	eZ800000100KRO	Object code/library package	Free support via zilog.com
Source Release	eZ800000100KRS	Full RZK source is provided	Full support with 6-month product updates

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Document Disclaimer

ZiLOG is a registered trademark of ZiLOG Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

ZiLOG is a registered trademark of ZiLOG Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

©2005 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses or other rights are conveyed, implicitly or otherwise, by this document under any intellectual property rights.