



An  IXYS Company

eZ80Acclaim!™ Product Family

Zilog Full-Featured TCP/IP Software Suite

Product Brief

PB015412-1211

Introduction

The Zilog TCP/IP Software Suite solution, ZTP, is an integrated, preemptive multitasking operating system and set of TCP/IP protocols that has been optimized for embedded systems. ZTP works in conjunction with the award-winning eZ80Acclaim! family of Flash microcontrollers. It also provides standard network connectivity in a wide range of applications, such as industrial control, automation, facility management, IP appliances, and remote systems communication.

ZTP is optimized to allow quick development of these low-cost systems. Powered by the Zilog Real-Time Kernel (RZK), ZTP offers fully-featured operating system services, in addition to network services, while occupying minimal program memory. Your applications can be integrated with ZTP via its easy-to-use and well-documented application programming interface (API).

► **Note:** The ZTP standard release is integrated with and installed by the *ZDSII_eZ80Acclaim!* install package.

ZTP supports the following TCP/IP protocols:

- IPv4, TCP, UDP, ARP and RARP
- ICMP, IGMP and DHCP/BOOTP

ZTP supports three physical interfaces:

- Ethernet 10/100
- PPP
 - On serial link with HDLC
 - On Ethernet with PPPoE
- 802.11 b/g Wireless LAN (wireless interface supports WEP, WPA and WPS2)

Building upon these field-proven core protocols, ZTP offers a rich set of higher-level TCP/IP services to

accelerate application development, and includes the following elements:

- HTTP/HTTPS server, TFTP client, SNMP agent, TELNET server and client, SMTP client, DNS client, TimeP client and SNTP client
- FTP server and client, plus an embedded Flash file system that works within the Flash and static RAM memories
- Local or remote run-time debugging operating system command shell

An industry-standard socket interface and native RZK APIs are the primary interfaces available to your applications for interfacing with the TCP/IP stack. These interfaces accelerate development time for network programmers who are already familiar with their socket interface. For new TCP/IP programmers, the native RZK APIs shorten the learning process. In addition, an internal mapping layer maintains a set of backward-compatible APIs to previous ZTP releases.

ZTP is royalty-free, configurable, scalable, and modular in design; it provides a rich set of features via easy-to-use and well-documented APIs. In addition, its features are highly optimized to the stringent memory and performance requirements typical of 8-bit embedded applications.

ZTP is available in a standard release package with an object library. It is also available in a source release package complete with full source code.

ZTP is currently available for Zilog's eZ80Acclaim! family of devices, including the eZ80F91, eZ80F92 and eZ80F93 microcontrollers, plus the eZ80L92 microprocessor.

Architecture

The ZTP software suite is designed to be highly modular and scalable. It is broadly classified into three major subsystems:

ZTP Drivers. The ZTP driver subsystem is comprised of device drivers such as Ethernet, UART, I²C, SPI,

RTC, WLAN, plus protocol drivers such as PPP and HDLC.

ZTP Core. The ZTP core subsystem is comprised of TCP/IP protocols and Zilog’s real-time operating system. ZTP supports stream connections (TCP), datagram connections (UDP) and secure connections (SSL).

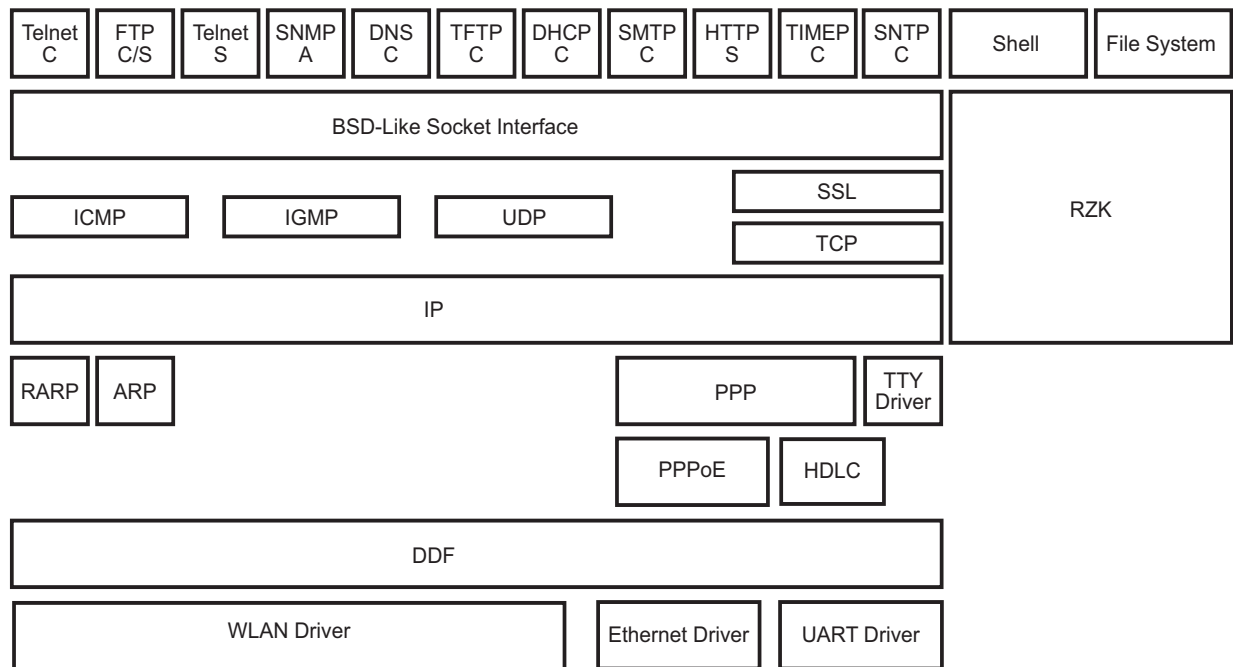
ZTP Applications. Most ZTP applications are comprised of high-level application protocols that use core components to perform standard network functions. An optional run-time shell interface is provided so that you can initiate shell commands using either a local RS-232 terminal or a remote terminal via Telnet.

The shell implements a rich set of informative debugging commands, and custom shell commands can be added. Many ZTP functions are initiated or monitored via the shell during run-time. For example, if a remote host initiates a Telnet client request, the ZTP shell provides a network virtual terminal to the remote host.

Another component of the ZTP applications subsystem is an embedded Flash file system, which supports multiple-disk volumes and operates in the Flash and RAM memory spaces. The Flash file system facilitates the easy handling of files over a network connection such as FTP and HTTP/HTTPS, and can be employed by the application to manage data files and directories. The Flash file system supports standard C APIs, such as `fopen`, `fclose`, etc.

ZTP application developers can enjoy flexibility toward using either the native RZK programming interface or a BSD socket-like interface to communicate with the network. The highly-configurable nature of ZTP thereby enables you to include only those components that are appropriate to the application, and exclude components that are not required to build your final binary image.

Figure 1 displays the ZTP architecture.



Legend: S = Server, C = Client, A = Agent

Figure 1. ZTP Architecture Block Diagram

Features

ZTP has been optimized with the requirements of typical 8-bit applications in mind. In addition to its industry standard RFC-compliant protocols, the key features of ZTP include:

HTTP Server. An application protocol to retrieve hypertext documents from the World Wide Web. Java applets and CGI calls are supported, as are many standard website objects/file types. The HTTP1.1 Keep-Alive mechanism and HTTP1.1 basic and message digest (MD5-based) scheme authentication are also supported.

HTTPS Server. An application protocol that adds data privacy, data integrity and authentication features to HTTP using SSL technology (optional: included in a separate SSL plug-in package).

Telnet Client. An application protocol that makes it possible for you to connect and run programs on remote hosts. The protocol implements options such as *echo* and *suppress go ahead*.

Telnet Server. An application protocol that allows remote users to Telnet into a system.

FTP Client/Server. An application protocol that allows you to transfer files from one system to another system over a TCP/IP network. This protocol operates seamlessly with the RZK-embedded Flash file system.

SNMP Agent. An application protocol that makes network management information available to SNMP managers. ZTP implements SNMPv1, SNMPv2 and SNMPv3. SNMPv2 includes an additional `GetBulk` command, and SNMPv3 implements authentication and encryption.

TFTP Client. The Trivial file transfer protocol (TFTP) is a simple form of FTP. This protocol is implemented on top of the User Datagram Protocol (UDP). Because it is small, it lacks most of the features of FTP.

DHCP Client. An application protocol that dynamically requests an IP address from an available DHCP server on a TCP/IP network.

SMTP Client. An application protocol that is used to transfer mail reliably and efficiently; it supports *challenge/response* (CRAM MD5-based) digest scheme authentication.

TimeP. An application protocol that retrieves a time reference from a remote TimeP server on a TCP/IP network.

SNTP Client. An application protocol that gets time from the time server.

PPPoE Client. PPPoE uses the features of Ethernet and PPP standards. If PPPoE is used, *dial* from one machine to another over an Ethernet network using the DSL line to establish a point-to-point connection between the host and target.

See the [Documentation](#) section of this product brief for a listing of the ZTP manuals that fully describe these features.

High Performance and Fast Context Switching

ZTP is powered by RZK, which can perform a context switch between threads in less than 12.5 μ s (measured on a 50 MHz eZ80F91 MCU running with 1 wait state in eZ80 bus mode). On this host platform, ZTP's high-performance core can transmit data on a UDP socket with a speed of 6Mbps. In the current ZTP release, this performance is reflected in the performance of the higher-layer network. Refer to the latest ZTP Release Notes for updated information.

Small Memory Footprint

The memory footprint for the basic ZTP core is less than 60 KB. The memory footprint of ZTP when running application protocols varies based on the components being included. RAM requirements for ZTP depend on, but are not limited to, actual configuration, such as the number of application threads and the number of TCP/UDP connections supported at run time.

Modular, Scalable and Configurable

ZTP is highly configurable and scalable; you can derive a light- to full-featured version per your application requirements. The modular design of ZTP enables you to include only those library objects that are required by your application, thereby saving memory space. The standard ZTP package is released as a set of libraries.

TCP/IP parameters, such as the number of UDP/TCP connections and the size of the UDP/TCP/ buffer, are configurable. System parameters, such as the number of objects required (threads, semaphores, etc) can be specified at compile time. ZTP allocates memory for the object control blocks based on the number of

objects specified in the configuration file. All of these compile-time configuration options make ZTP fully configurable, thereby enabling you to fine-tune it to your requirements.

For complete information about configuring ZTP, refer to the [Zilog TCP/IP Software Suite Programmer's Guide \(RM0041\)](#).

Portability

ZTP applications can be easily migrated/ported to a custom eZ80[®] CPU-based hardware platform. Extensive user documentation is available for the standard ZTP release; see the [Documentation](#) section of this product brief for a listing of the ZTP manuals that fully describe these features.

Powerful Kernel

Zilog's latest RZK supports both preemptive and cooperative multitasking. The RZK scheduler runs tasks of equal priority in round-robin sequence. Thread priority can be changed at run-time. RZK offers the following mechanisms for inter-thread communication and synchronization:

Semaphores. RZK provides both binary as well as counting semaphores, and implements a priority inheritance protocol to solve priority inversion issues.

Message Queues. RZK provides message queues for asynchronous communication between threads; high-priority messages can be placed in front of the queue.

Event Groups. Event groups are provided for control synchronization, and do not carry any information; an event group can accommodate a maximum of 24 events.

Memory Management. Fixed-size memory pools are provided for deterministic memory allocation, which are very useful for time-critical applications. RZK also supports variable-size memory allocation.

For complete information about the RZK kernel, refer to the following documents, which are available free for download from the Zilog website:

- [Zilog Real-Time Kernel Product Brief \(PB0155\)](#)
- [Zilog Real-Time Kernel User Manual \(UM0075\)](#)

- [Zilog Real-Time Kernel Reference Manual \(RM0006\)](#)

Demonstration Examples

The standard ZTP package includes sample applications that demonstrate a number of ZTP features for easy reference. These demo applications can be used as a reference to start a new project with a new configuration. These samples are simple interactive applications that are used to demonstrate:

- Socket programming for network data transfer over Ethernet and PPP interfaces
- Raw data transfer ability of ZTP over serial peripherals such as UART, I²C and SPI using RZK's DDF
- A simple webserver application
- An FTP client demo
- A minimal configuration demo
- A *ping* API utility demo
- A TCP and UDP socket programming demo

These examples are available with ZDSII tool chains for the entire family of eZ80Acclaim![®] microcontrollers.

Development Tools

ZTP is built upon the Zilog Developer Studio Integrated Development Environment (ZDSII), which provides code editing, assembler/C Compiler/linker features, source-level debugging and project managing facilities for the quick and efficient development of embedded applications. ZDSII is included with all eZ80Acclaim! development kits.

Packaging

The standard ZTP package for eZ80Acclaim! MCUs is supplied as C object library modules, complete with source files and application examples. A package with full source code is also available. For part number references, refer to the [Zilog TCP/IP Stack API Reference Manual \(RM0040\)](#).