ZirDA™ Software Development Kit

ZiLOG IrDA Protocol Stack Software for eZ80Acclaim!™ MCUs

Product Brief

PB014401-0304 PRELIMINARY

Introduction

ZirDA™, ZiLOG's highly optimized IrDATM-compliant protocol stack software, is a layered set of embedded protocols aimed at providing a total data transfer solution over the infrared medium. IrDA is an international standard for point-to-point communication of typical mobile devices using wireless infrared technology. ZirDA™ is specifically designed for use with ZiLOG's award-winning eZ80Acclaim!™ microcontrollers, ZiLOG's ZHX-series SIR IrDA transceivers, and targets applications in portable office equipment, wireless serial adapters, cell phones, PDAs, pagers, digital cameras, and other emerging markets such as fixed IrDA online access beacons.

ZiLOG's IrDA expertise and extensive involvement in the IrDA standards organization ensures quality code compliance with IrDA specifications, and offers complete IrDA functionality. ZirDA™ is currently available on the eZ80F91 and eZ80F92 microcontrollers—members of ZiLOG's family of eZ80Acclaim!™ Flash MCUs—as well as the ROMless eZ80L92 microprocessor. ZirDA™ is configurable, scalable, and modular in design; it provides a rich set of features via easy-to-use and well-documented APIs. ZirDA™ is available as a standard release with an object/library package, as well as a source release package with full source code.

Architecture

The ZirDA™ protocol stack is a layered set of protocols that is particularly aimed at point-to-point infrared communications. ZirDA™, based on ZiLOG's Real-Time Kernel, RZK, is meant to be integrated with the main user application software required to complete a wireless IrDA communication system.

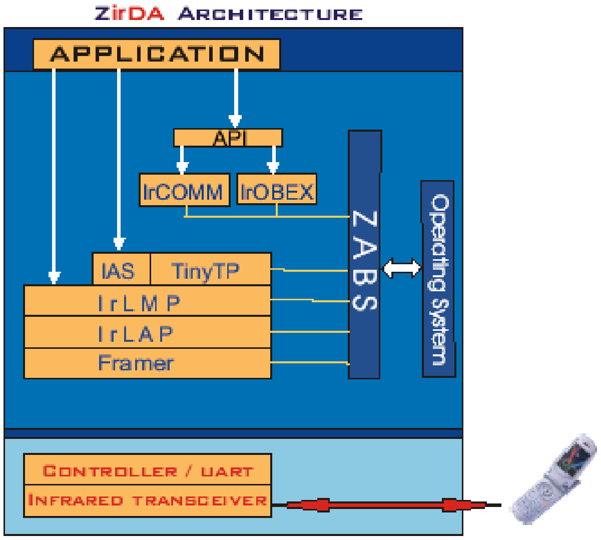

Figure 1. ZirDA™ Architecture and Features

ZirDA™ implements all of the mandatory IrDA protocols and additional applications protocols. The mandatory protocols are IAS, TinyTP, IrLMP, IrLAP, and Framer. Application protocols such as IrCOMM for emulating a serial port and the IrOBEX for exchanging data objects between two IrDA devices are included in ZirDA™. Each of the ZirDA™ protocol layers provides easy-to-use APIs that implement the features of the IrDA stack. The protocol layers are independent of each other, offering the user the flexibility to select only the layers that are required. Each layer communicates to another by means of an abstraction interface termed *ZABS*. This ZiLOG abstraction interface is a proprietary technology that makes the ZirDA™ stack easily portable to any other OS kernel. ZABS provides services such as state machine, memory management, message management, and entity management. ZABS allows each layer to execute in a separate task context for speedier responses or in a single-task context where resource constraints are an issue.

ZirDA™ represents a complete IrDA solution when combined with ZiLOG's eZ80Acclaim!™ Flash MCUs and the ZHX Series of IrDA transceivers.

Features

The integrated features of ZirDA™ are offered especially with the requirements of typical 8-bit applications in mind. A description of each of the salient features of ZirDA™ follows.

Complete IrDA Stack Functionality

ZirDA™ protocol software is a complete and modular IrDA solution. ZirDA™ includes the SIR framer, IrLAP, IrLMP, IAS, and TinyTP protocol implementations. For applications requiring an extension of infrared functionality, the optional add-in services of IrCOMM (3-wire raw, IrLPT) and IrOBEX are available. The stack supports both primary (initiate) and secondary (respond) connections.

Simple Data Object Exchange

The ZirDA™ IrOBEX API complies with the IrDA Object Exchange specification and handles discovery, connection setup, and maintenance of the infrared connection—it easily transmits such objects as files, graphics, vCards, vCalendars, and vMessages). IrOBEX is embraced by many devices that must transmit or receive documents, phone messages, digital messages, electronic business cards, database records, and programming. IrOBEX also allows the application developer to abort any object transfer that is in progress by simply calling the appropriate API.

ZirDA™ supports the CONNECT, PUT, GET, ABORT, and DISCONNECT operations of IrOBEX.

Rapid Application Development

The ZirDA™ IrOBEX API provides the ability to PUT and GET data objects simply and flexibly, thereby enabling rapid application development and interaction with a broad class of devices including PCs, PDAs, data collectors, cellular phones, handheld scanners, cameras, and more. The application developer does not have to worry about low-level IrDA functions such as IAS connection establishment, but rather can focus on higher-level application development. The ZirDA™ IrOBEX API supports most of the optional headers that the standard IrOBEX specification allows, such as object Description, object Name, and object Type. This support allows the application developer to provide as much information as possible during data transfer. The IrOBEX PUT request included with ZirDA™ is not only used to make the file transfer, but also allows the developer to create and delete an object on the receiving end. The ZirDA™ IrOBEX API enables developers to create a wide range of IrOBEX applications with maximum flexibility.

Emulating a Serial Port with IrCOMM

The ZirDA™ IrCOMM module allows legacy applications that use a serial port to operate over IrDA without modification. The motivation to use IrCOMM comes from many printing and communication applications that use standard communication APIs to talk to other devices through serial and parallel ports. By making IrDA protocols accessible through these APIs, many existing applications, including printing, can operate over an IrDA link without modifications. Thus, IrCOMM is included to support these so-called legacy applications. The current implementation of the ZirDA™ IrCOMM API supports the IrCOMM 3-wire RAW format. It also supports all of the mandatory services of standard IrCOMM specifications.

Information Access Service

The ZirDA™ IAS API supports all of the mandatory services of IAS. IAS allows devices to discover which services are available on the host device and provides the configuration information necessary to access those services. The application developer has the option of setting the size of the IAS class names and attribute names in the configuration file, according to requirements, to reduce the RAM size.

High Level Flow Control using TinyTP

The ZirDA™ TinyTP API supports all of the mandatory services of TTP. It adds per-channel flow control to keep things moving smoothly. ZirDA™ TinyTP also supports Segmentation And Reassembly (SAR) functionality. The ZirDA™ IrOBEX API uses the services of TinyTP to make object transfer.

Multiple Logical Channels on a single Physical Link using IrLMP

The ZirDA™ IrLMP API allows services of one device to communicate with identical services in a peer device, without any interference from other services, using the same IrLAP connection. It supports Exclusive Access Mode, which allows the developer to exclusively own the Physical link connection for the entire data transfer. The number of LSAP connections can be configured based on RAM requirements.

Reliable Connection through IrLAP

The ZirDA™ IrLAP API provides a point-to-point reliable connection that supports services such as discovery, address conflict resolution, connection establishment, data transfer, and flow control. Based on application requirements, the application developer can also set the QOS parameters to be negotiated by IrLAP by calling a simple API. The same QOS parameters are used during IrLAP negotiation.

SIR Framer

The ZirDA™ Framer API supports all of the SIR baud rates up to 115.2kbps.

Support for Security and Power Savings

ZirDA™ provides APIs that allow all of the managed peripherals to operate in power-save mode when unused. The peripherals can be activated and deactivated by calling the IrDAActivate and IrDADeactivate APIs.

ZirDA™ Configurability

The Transmit and Receive buffers used in IrLAP and IrLMP can be specified at compile time. ZirDA™ allocates memory for these buffers based on a size specified by the user. The user can also specify the window size, the Maximum Transmit Data size, and Maximum Receive Data size in the configuration file.

All of these compile-time configuration options make ZirDA™ fully configurable, and thus enable the application developer to fine-tune it for the end application.

Complete Solution at Lowest Cost

With ZirDA™ protocol software, ZiLOG's eZ80Acclaim!™ Flash MCUs, and IrDA transceivers, ZiLOG provides the embedded industry's first complete solution for Flash-based wireless IrDA connectivity. As a result, upgradability is maintained, and development cycle time and cost of ownership are reduced.

Demonstration Example

The standard ZirDA™ SDK package ships with a sample IrDA Monitor application that:

- Is a simple command-line-based, user-interactive application to showcase the features of ZirDA™.
- Provides a facility to:
  - Discover devices present in the vicinity
  - Test the features of IrOBEX and IrCOMM
  - Send/receive objects such as vCard to/ from a remote device
  - Send a file to a printer to print the file's contents using IrCOMM
  - Chat with a remote device using IrCOMM.

The chat functionality works only when both the sender and receiver are eZ80Acclaim!™ devices (eZ80L92, eZ80F91, eZ80F92), as this operation and the commands used in the implementation of this operation are more specific to the ZirDA™ Demo Application.

Development Tools

ZirDA™ is built on ZiLOG's Integrated Development Environment, ZDS II, that provides compiling, debugging, and project-building facilities for the quick and efficient development of embedded applications. ZDS II, together with a full-featured ANSI C-compiler and debugger, is included in all eZ80Acclaim!™ development kits. Please visit zilog.com for more details.

Packaging

The standard ZirDA™ package for eZ80Acclaim!™ is supplied as C object library module that includes the necessary source files and application examples. A package with full source code is also available.

Documentation

The documents in Table 1 are available with the release. They comprehensively describe all of the features, functions, and usage of ZirDA™.

Table 1. ZirDA™ Documentation Set

ZirDA™ SDK Quick Start Guide	This document enables the user to install and get started with the ZirDA™ SDK. It guides the user through the sample application.
ZirDA™ User Manual	This document explains the different IrDA configurations and features and offers a procedure to run the IrDA sample demo application.
ZirDA™ IrOBEX API Reference Manual	This document details all of the APIs that are provided in the IrOBEX module, and provides an overview of IrOBEX functionality and the architecture of the ZirDA™ IrOBEX implementation.
ZirDA™ IrCOMM API Reference Manual	This document details all of the APIs that are provided in the IrCOMM module.
ZirDA™ Common API Reference Manual	This document explains the common APIs that are used by both IrOBEX and IrCOMM modules. These general-purpose APIs are used to activate and deactivate the stack.
ZirDA™ Demo Commands Reference Manual	This document describes the commands used in the ZirDA™ Demo Application to communicate with another device.

Ordering Information

ZirDA™ Package	Part Number	Description	Support
Standard Release	eZ800000100SIO	Object code/library package	Free support at zilog.com
Full release	eZ800000100SIS	Source code package	Free support at zilog.com

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.

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.