



eZ80L92 Development Kit

User Manual

PRELIMINARY

UM012906-0103

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

eZ80L92 Development Kit



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.

©2003 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concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded.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 of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

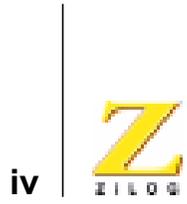


Safeguards

The following precautions must be observed when working with the devices described in this document.



Caution: Always use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).



eZ80L92 Development Kit User Manual



Table of Contents

Safeguards	iii
List of Figures	vii
List of Tables	ix
Introduction	1
Kit Features	1
Hardware Specifications	2
eZ80 [®] Development Platform Overview	2
eZ80 [®] Development Platform	7
Functional Description	7
Physical Dimensions	9
Operational Description	10
eZ80L92 Module Interface	10
Application Module Interface	20
I/O Functionality	23
Embedded Modem Socket Interface	28
eZ80 [®] Development Platform Memory	31
LEDs	34
Push Buttons	34
Jumpers	35
Connectors	40
Console	41
Modem	41
I ² C Devices	41
DC Characteristics	41
eZ80L92 Module	44
Functional Description	44
Physical Dimensions	44
Operational Description	47

Ethernet Media Access Controller	47
eZ80L92 Module Memory	50
Reset Generator	50
IrDA Transceiver	50
DC Characteristics	53
Flash Loader Utility	53
Mounting the Module	54
Changing the Power Supply Plug	54
ZPAKII	56
ZDI Target Interface Module	56
JTAG	56
Application Modules	56
ZDS II	57
Troubleshooting	59
Overview	59
Cannot Download Code	59
No Output on Console Port	59
IrDA Port Not Working	60
Difference Between EMAC and IP Address	60
Media Access Control (MAC)	60
IP Address	61
Contacting ZiLOG Customer Support	62
Schematic Diagrams	63
eZ80 [®] Development Platform	63
eZ80L92 Module	68
Appendix A	77
General Array Logic Equations	77
U10 Address Decoder	77
U15 Address Decoder	79
Customer Feedback Form	83



List of Figures

Figure 1.	eZ80 [®] Development Platform Block Diagram with eZ80L92 Module	4
Figure 2.	The eZ80 [®] Development Platform	5
Figure 3.	The eZ80L92 Module	6
Figure 4.	Basic eZ80 [®] Development Platform Block Diagram	8
Figure 5.	Physical Dimensions of the eZ80 [®] Development Platform	9
Figure 6.	eZ80 [®] Development Platform Peripheral Bus Connector Pin Configuration—JP1	11
Figure 7.	eZ80 [®] Development Platform I/O Connector Pin Configuration—JP2	16
Figure 8.	Trigger Pins J21 and J22	28
Figure 9.	Embedded Modem Socket Interface—J1, J5, and J9	29
Figure 10.	Memory Map of the eZ80 [®] Development Platform and eZ80L92 Module	33
Figure 11.	Physical Dimensions of the eZ80L92 Module	45
Figure 12.	Top Layer	46
Figure 13.	Bottom Layer	47
Figure 14.	IrDA Hardware Connections	52
Figure 15.	9VDC Universal Power Supply Components	54
Figure 16.	Inserting a New Plug Configuration	55
Figure 17.	eZ80 [®] Development Platform Schematic Diagram, #1 of 5	63
Figure 18.	eZ80 [®] Development Platform Schematic Diagram, #2 of 5	64
Figure 19.	eZ80 [®] Development Platform Schematic Diagram, #3 of 5	65
Figure 20.	eZ80 [®] Development Platform Schematic Diagram, #4 of 5	66
Figure 21.	eZ80 [®] Development Platform Schematic Diagram, #5 of 5—RS-485 Cable	67



Figure 22.	eZ80L92 Module Schematic Diagram, #1 of 9—Top Level	68
Figure 23.	eZ80L92 Module Schematic Diagram, #2 of 9—100-Pin QFP eZ80L92 Device	69
Figure 24.	eZ80L92 Module Schematic Diagram, #3 of 9—36-Pin SRAM Device	70
Figure 25.	eZ80L92 Module Schematic Diagram, #4 of 9—Flash Device . . .	71
Figure 26.	eZ80L92 Module Schematic Diagram, #5 of 9—E-NET Module .	72
Figure 27.	eZ80L92 Module Schematic Diagram, #6 of 9—IrDA Reset	73
Figure 28.	eZ80L92 Module Schematic Diagram, #7 of 9—Headers	74
Figure 29.	eZ80L92 Module Schematic Diagram, #8 of 9—Power Supply . .	75
Figure 30.	eZ80L92 Module Schematic Diagram, #9 of 9—Control Logic . .	76



List of Tables

Table 1.	eZ80 [®] Development Platform Hardware Specifications . . .	2
Table 2.	eZ80 [®] Development Platform Peripheral Bus Connector Identification—JP11	12
Table 3.	eZ80 [®] Development Platform I/O Connector Identification—JP21	17
Table 4.	General-Purpose Port Connector J6	21
Table 5.	CPU Bus Connector J8	23
Table 6.	LED and Port Emulation Addresses	24
Table 7.	LED Anode/General-Purpose Port Output Control Register	24
Table 8.	General-Purpose Port Data Register	25
Table 9.	Bit Access to the LED Cathode, Modem, and Triggers . .	26
Table 10.	Connector J5	29
Table 11.	Connector J9	30
Table 12.	Connector J1	30
Table 13.	J2—DIS_IrDA	35
Table 14.	J3—DIS_EM	36
Table 15.	J7—FlashWE	36
Table 16.	J11—DIS_Flash	37
Table 17.	J12—5VDC/3.3VDC for an Embedded Modem	37
Table 18.	J14—RI	38
Table 19.	J15—RS485_1_EN	38
Table 20.	J16—RS485_2_EN	39
Table 21.	J17—RT_1	39
Table 22.	J18—RT_2	40
Table 23.	I ² C Addresses	41



Table 24.	DC Current Characteristics of the eZ80 [®] Development Platform with Different Module Loads	42
Table 25.	Ethernet Connector Pin Assignments	48



Introduction

The eZ80L92 Development Kit provides a general-purpose platform for evaluating the capabilities and operation of ZiLOG's eZ80L92 microprocessor. The eZ80L92 Development Kit features two primary boards: the eZ80[®] Development Platform and the eZ80L92 Module. This arrangement provides a full development platform when using both boards. It can also provide a smaller-sized reference platform with the eZ80L92 Module as a stand-alone development tool.

Kit Features

The key features of the eZ80L92 Development Kit are:

- eZ80[®] Development Platform:
 - Up to 2MB fast SRAM (12ns access time)
 - Embedded Modem Socket with a U.S. Telephone Line Interface
 - I²C EEPROM
 - I²C Configuration Register
 - General-Purpose Port and Memory Headers
 - Supported by ZiLOG Developer Studio II and the eZ80[®] C-Compiler
 - LEDs, including a 7x5 LED matrix
 - Jumpers
 - Two RS232 connectors—Console, Modem
 - RS485 connector
 - ZiLOG Debug Interface (ZDI)
 - JTAG Debug Interface
 - 9VDC Power Connector
 - Telephone Jack



- eZ80L92 Module:
 - eZ80L92 microprocessor operating at 48MHz
 - 1MB Flash Memory
 - 512KB SRAM
 - 10BaseT Ethernet Interface
 - Real-Time Clock with Battery Back-Up
- ZPAKII Debug Interface
- 4-port 10BaseT Ethernet hub
- eZ80L92 Development Kit Software and Documentation CD-ROM

Hardware Specifications

Table 1 lists the specifications of the eZ80[®] Development Platform.

**Table 1. eZ80[®] Development Platform
Hardware Specifications**

Operating Temperature:	20°C ±5°C
Operating Voltage:	9 VDC

eZ80[®] Development Platform Overview

The purpose of the eZ80L92 Development Kit is to provide the developer with a set of tools for evaluating the features of the eZ80[®] family of devices, and to be able to develop a new application before building application hardware. The eZ80[®] Development Platform is designed to accept a number of application-specific modules and Z8- and eZ80[®]-based add-on modules, including the eZ80L92 Module, which features an EMAC, an IrDA transceiver, and the eZ80L92 microprocessor.

The eZ80L92 Development Kit features two primary boards: the eZ80[®] Development Platform and the eZ80L92 Module. This arrangement pro-



vides a full development platform when using both boards. It can also provide a smaller-sized reference platform with the eZ80L92 Module as a stand-alone development tool.

The eZ80[®] Development Platform can operate in stand-alone mode with Flash memory, or interface via the ZPAKII emulator to a host PC running ZiLOG Developer Studio II Integrated Development Environment (ZDS IDE) software. If the user's eZ80[®] application demands Internet connectivity and/or a network connection, the eZ80[®] can serve web pages over a TCP/IP network, allowing easy system monitoring and control, and effortless processor code updates.

The address bus, data bus, and all eZ80L92 Module control signals are buffered on the eZ80[®] Development Platform to provide sufficient drive capability.

A block diagram of the eZ80[®] Development Platform and the eZ80L92 Module is shown in Figure 1.

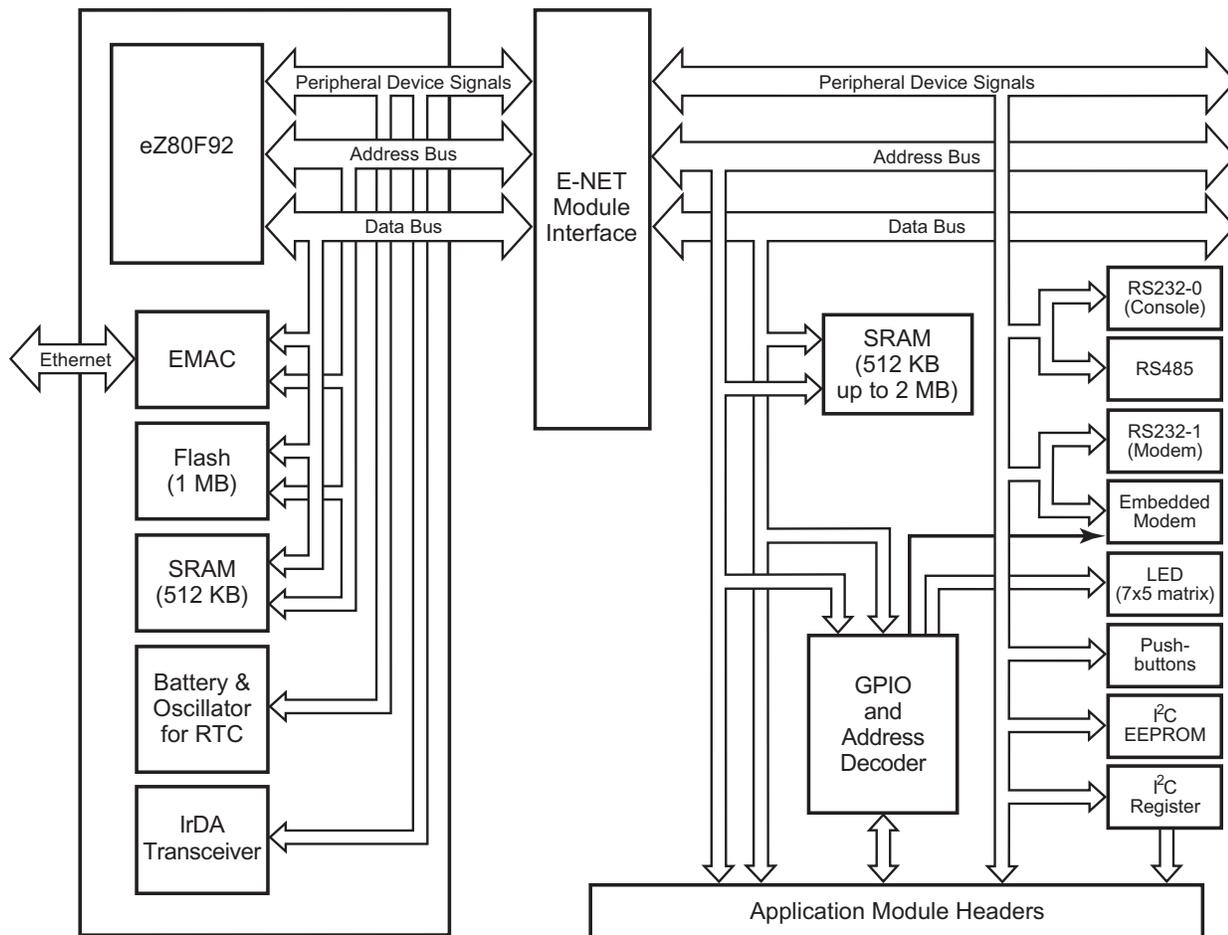
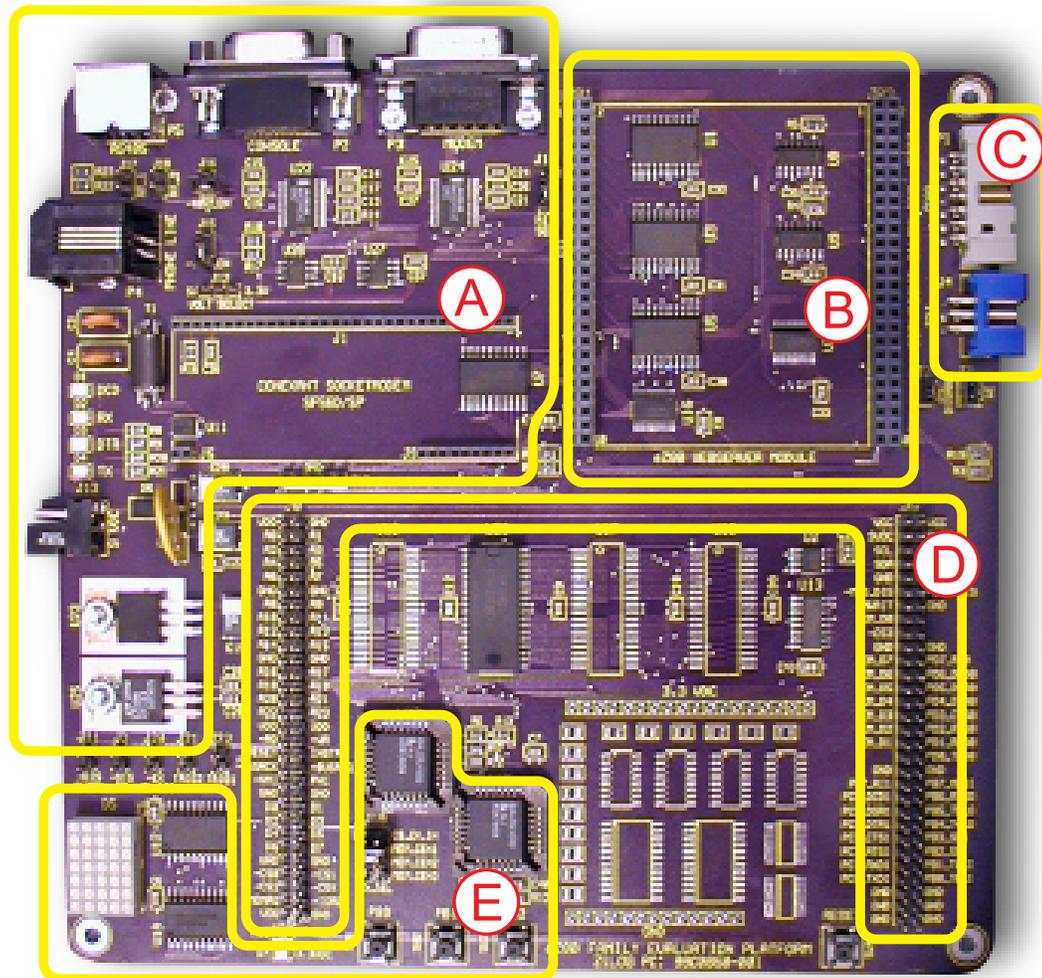


Figure 1. eZ80[®] Development Platform Block Diagram with eZ80L92 Module

Figure 2 is a photographic representation of the eZ80[®] Development Platform segmented into its key blocks, as shown in the legend for the figure.



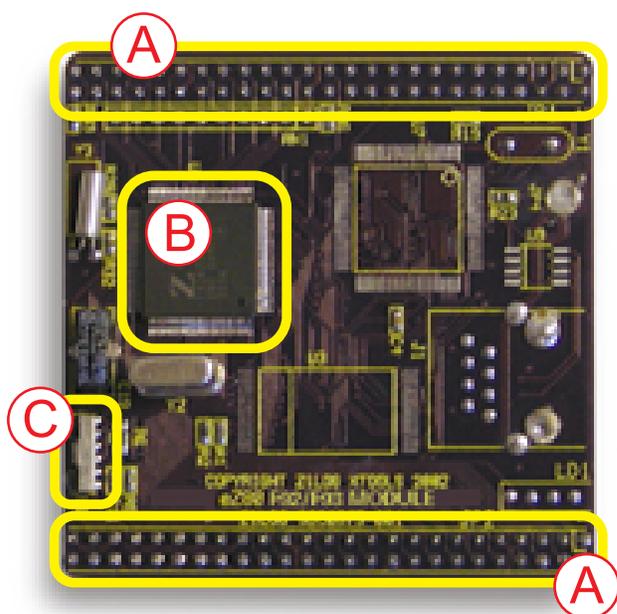
Note: Key to blocks A–E.

- A. Power and serial communications.
- B. eZ80L92 Module interface.
- C. Debug interface.

- D. Application module interfaces.
- E. General-Purpose Port and LED with Address Decoder.

Figure 2. The eZ80[®] Development Platform

Figure 3 is a photographic representation of the eZ80L92 Module segmented into its key blocks, as shown in the legend for the figure.



Note: Key to blocks A–C.

- A. eZ80L92 Module interfaces.
- B. CPU and memory.
- C. Ethernet connection.
- D. IrDA transceiver.

Figure 3. The eZ80L92 Module

The structures of the eZ80[®] Development Platform and the eZ80L92 Module are illustrated in the [Schematic Diagrams](#) starting on page 63.



eZ80[®] Development Platform

This section describes the eZ80[®] Development Platform hardware, its key components and its interfaces, including detailed programmer interface information such as memory maps, register definitions, and interrupt usage.

Functional Description

The eZ80[®] Development Platform consists of seven major hardware blocks. These blocks, listed below, are diagrammed in Figure 4.

- eZ80L92 Module interface (2 female headers)
- Power supply for the eZ80[®] Development Platform, the eZ80L92 Module, and application modules
- Application Module interface (2 male headers)
- General-Purpose Port and LED matrix
- RS232 serial communications ports
- Embedded modem interface
- I²C devices

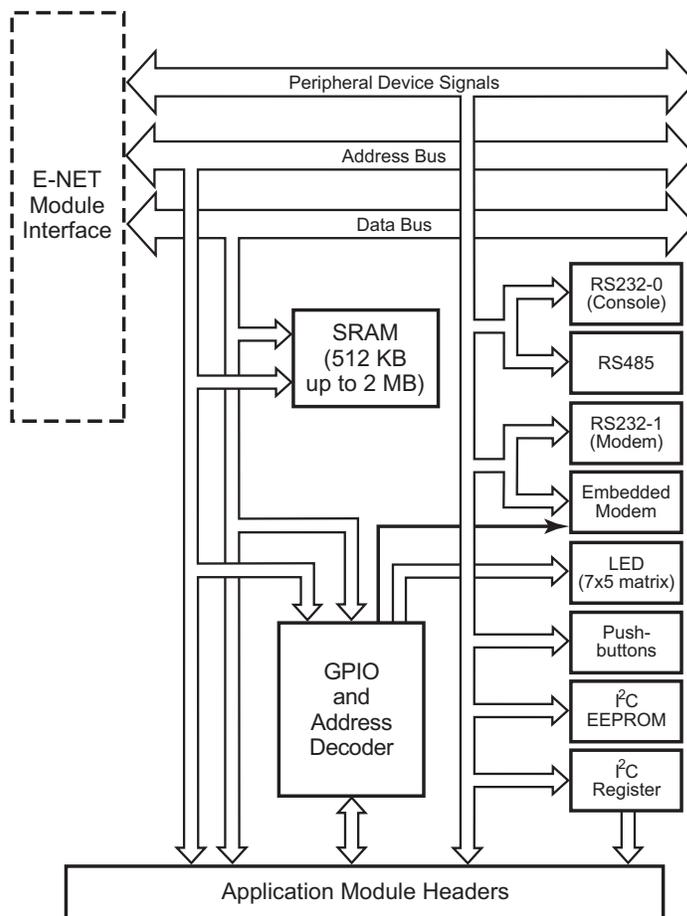


Figure 4. Basic eZ80[®] Development Platform Block Diagram

Physical Dimensions

The dimensions of the eZ80[®] Development Platform PCB is 177.8 mm x 182.9 mm. The overall height is 38.1 mm. See Figure 5.

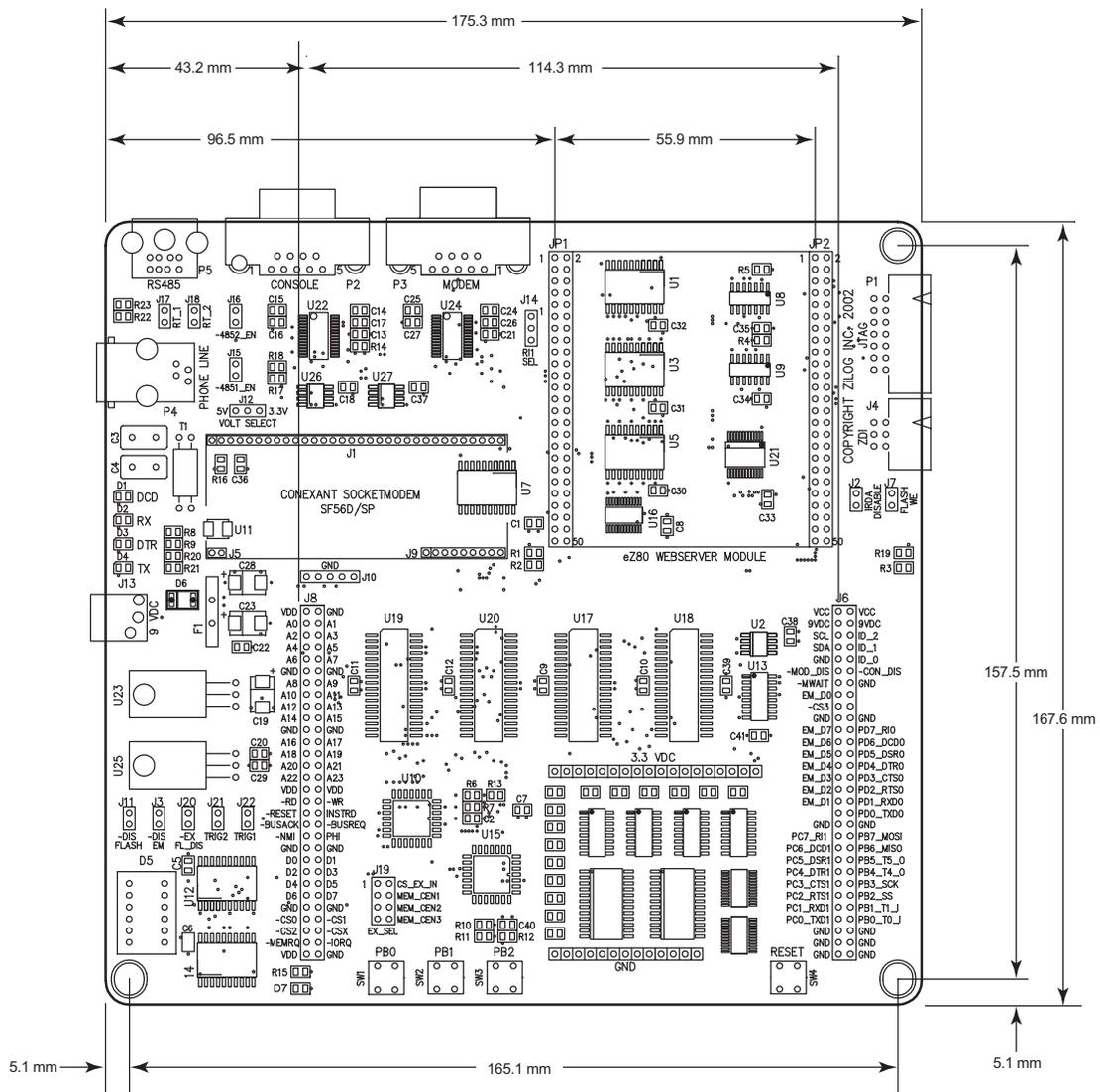


Figure 5. Physical Dimensions of the eZ80[®] Development Platform



Operational Description

The eZ80[®] Development Platform can accept any eZ80[®]-core-based modules, provided that the module interfaces correctly to the eZ80[®] Development Platform. The purpose of the eZ80[®] Development Platform is to provide the application developer with a tool to evaluate the features of the eZ80L92 device and to develop an application without building additional hardware.

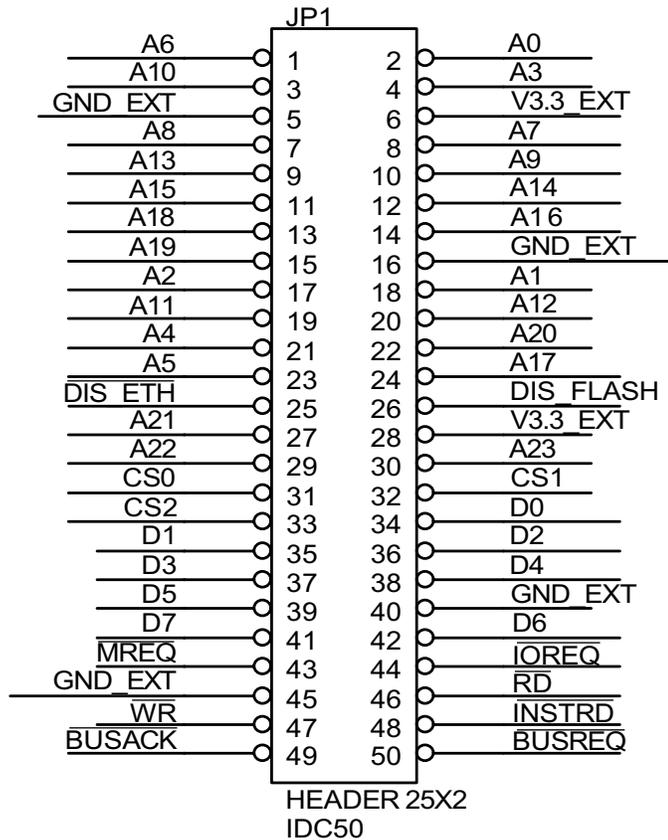
eZ80L92 Module Interface

The eZ80L92 Module interface provides easy connection of the eZ80L92 Module. It also provides easy connection for any eZ80[®]-based module designed to this interface. This includes modules using future eZ80[®] devices, and user-developed modules using current eZ80[®] devices.

The eZ80L92 Module interface consists of two 50-pin receptacles, JP1 and JP2.

Peripheral Bus Connector

Figure 6 illustrates the pin layout of the Peripheral Bus Connector in the 50-pin header, located at position JP1 on the eZ80[®] Development Platform. Table 2 describes the pins and their functions.



**Figure 6. eZ80[®] Development Platform
Peripheral Bus Connector Pin Configuration—JP1**



**Table 2. eZ80[®] Development Platform
Peripheral Bus Connector Identification—JP1¹**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
1	A6	Bidirectional		Yes
2	A0	Bidirectional		Yes
3	A10	Bidirectional		Yes
4	A3	Bidirectional		Yes
5	GND			
6	V _{DD}			
7	A8	Bidirectional		Yes
8	A7	Bidirectional		Yes
9	A13	Bidirectional		Yes
10	A9	Bidirectional		Yes
11	A15	Bidirectional		Yes
12	A14	Bidirectional		Yes
13	A18	Bidirectional		Yes
14	A16	Bidirectional		Yes
15	A19	Bidirectional		Yes

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).

2. The Power and Ground nets are connected directly to the eZ80L92 device.

Additional note: external capacitive loads on RD, WR, IORQ, MREQ, D0–D7 and A0–A23 should be below 10pF to satisfy the timing requirements for the eZ80[®] CPU. All unused inputs should be pulled to either V_{DD} or GND, depending on their inactive levels to reduce power consumption and to reduce noise sensitivity. To prevent EMI, the EZ80CLK output can be deactivated via software in the eZ80F91's Peripheral Power-Down Register.



**Table 2. eZ80[®] Development Platform
Peripheral Bus Connector Identification—JP1¹ (Continued)**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
16	GND			
17	A2	Bidirectional		Yes
18	A1	Bidirectional		Yes
19	A11	Bidirectional		Yes
20	A12	Bidirectional		Yes
21	A4	Bidirectional		Yes
22	A20	Bidirectional		Yes
23	A5	Bidirectional		Yes
24	A17	Bidirectional		Yes
25	$\overline{\text{DIS_ETH}}$	Output	Low	No
26	$\overline{\text{DIS_FLASH}}$	Output	Low	No
27	A21	Bidirectional		Yes
28	V _{DD}			
29	A22	Bidirectional		Yes
30	A23	Bidirectional		Yes

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).
 2. The Power and Ground nets are connected directly to the eZ80L92 device.
- Additional note: external capacitive loads on $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$, $\overline{\text{MREQ}}$, D0–D7 and A0–A23 should be below 10pF to satisfy the timing requirements for the eZ80[®] CPU. All unused inputs should be pulled to either V_{DD} or GND, depending on their inactive levels to reduce power consumption and to reduce noise sensitivity. To prevent EMI, the EZ80CLK output can be deactivated via software in the eZ80F91's Peripheral Power-Down Register.



**Table 2. eZ80[®] Development Platform
Peripheral Bus Connector Identification—JP1¹ (Continued)**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
31	CS0	Input	Low	Yes
32	CS1	Input	Low	Yes
33	CS2	Input	Low	Yes
34	D0	Bidirectional		Yes
35	D1	Bidirectional		Yes
36	D2	Bidirectional		No
37	D3	Bidirectional		Yes
38	D4	Bidirectional		Yes
39	D5	Bidirectional		Yes
40	GND			
41	D7	Bidirectional		Yes
42	D6	Bidirectional		Yes
43	$\overline{\text{MREQ}}$	Bidirectional	Low	Yes
44	$\overline{\text{IORQ}}$	Bidirectional	Low	Yes
45	GND			

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).

2. The Power and Ground nets are connected directly to the eZ80L92 device.

Additional note: external capacitive loads on $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$, $\overline{\text{MREQ}}$, D0–D7 and A0–A23 should be below 10pF to satisfy the timing requirements for the eZ80[®] CPU. All unused inputs should be pulled to either V_{DD} or GND, depending on their inactive levels to reduce power consumption and to reduce noise sensitivity. To prevent EMI, the EZ80CLK output can be deactivated via software in the eZ80F91's Peripheral Power-Down Register.



**Table 2. eZ80[®] Development Platform
Peripheral Bus Connector Identification—JP1¹ (Continued)**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
46	\overline{RD}	Bidirectional	Low	Yes
47	\overline{WR}	Bidirectional	Low	Yes
48	\overline{INSTRD}	Input	Low	Yes
49	\overline{BUSACK}	Input	Pull-Up 10K Ω ; Low	Yes
50	\overline{BUSREQ}	Output	Pull-Up 10K Ω ; Low	Yes

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).
2. The Power and Ground nets are connected directly to the eZ80L92 device.

Additional note: external capacitive loads on RD, WR, IORQ, MREQ, D0–D7 and A0–A23 should be below 10pF to satisfy the timing requirements for the eZ80[®] CPU. All unused inputs should be pulled to either V_{DD} or GND, depending on their inactive levels to reduce power consumption and to reduce noise sensitivity. To prevent EMI, the EZ80CLK output can be deactivated via software in the eZ80F91's Peripheral Power-Down Register.



I/O Connector

Figure 7 illustrates the pin layout of the I/O Connector in the 50-pin header, located at position JP2 of the eZ80[®] Development Platform. Table 3 describes the pins and their functions.

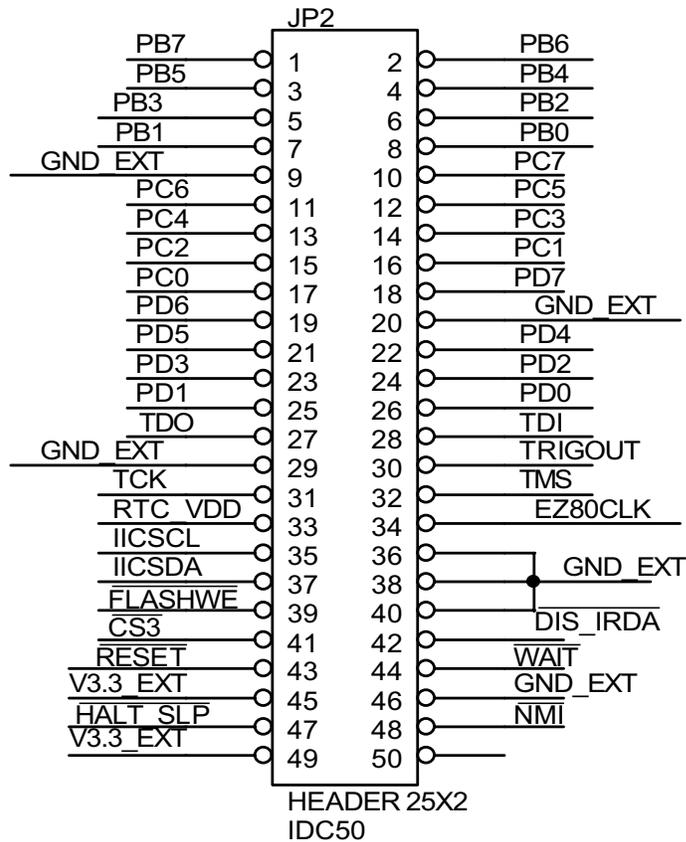


Figure 7. eZ80[®] Development Platform I/O Connector Pin Configuration—JP2



**Table 3. eZ80® Development Platform
I/O Connector Identification—JP2¹**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
1	PB7	Bidirectional		Yes
2	PB6	Bidirectional		Yes
3	PB5	Bidirectional		Yes
4	PB4	Bidirectional		Yes
5	PB3	Bidirectional		Yes
6	PB2	Bidirectional		Yes
7	PB1	Bidirectional		Yes
8	PB0	Bidirectional		Yes
9	GND			
10	PC7	Bidirectional		Yes
11	PC6	Bidirectional		Yes
12	PC5	Bidirectional		Yes
13	PC4	Bidirectional		Yes
14	PC3	Bidirectional		Yes
15	PC2	Bidirectional		Yes
16	PC1	Bidirectional		Yes
17	PC0	Bidirectional		Yes
18	PD7	Bidirectional		Yes

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).
2. The Power and Ground nets are connected directly to the eZ80L92 device.



**Table 3. eZ80[®] Development Platform
I/O Connector Identification—JP2¹ (Continued)**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
19	PD6	Bidirectional		
20	GND			
21	PD5	Bidirectional		Yes
22	PD4	Bidirectional		Yes
23	PD3	Bidirectional		Yes
24	PD2	Bidirectional		Yes
25	PD1	Bidirectional		Yes
26	PD0	Bidirectional		Yes
27	TDO	Input		Yes
28	TDI/ZDA	Output		Yes
29	GND			
30	TRIGOUT	Input	High	
31	TCK/ZCL	Output		Yes
32	TMS	Output	High	Yes
33	RTC_V _{DD}			
34	EZ80CLK	Input		Yes
35	SCL	Bidirectional		Yes
36	GND			

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).
2. The Power and Ground nets are connected directly to the eZ80L92 device.



**Table 3. eZ80[®] Development Platform
I/O Connector Identification—JP2¹ (Continued)**

Pin #	Symbol	Signal Direction	Active Level	eZ80L92 Signal ²
37	SDA	Bidirectional		Yes
38	GND			
39	$\overline{\text{FlashWE}}$	Output	Low	No
40	GND			
41	$\overline{\text{CS3}}$	Input	Low	Yes
42	$\overline{\text{DIS_IrDA}}$	Output	Low	No
43	$\overline{\text{RESET}}$	Bidirectional	Low	Yes
44	$\overline{\text{WAIT}}$	Output	Pull-Up 10K Ω ; Low	Yes
45	V _{DD}			
46	GND			
47	$\overline{\text{HALT_SLP}}$	Input	Low	Yes
48	$\overline{\text{NMI}}$	Output	Low	Yes
49	V _{DD}			
50	Reserved			

Notes:

1. For the sake of simplicity in describing the interface, Power and Ground nets are omitted from this table. The entire interface is represented in the eZ80L92 Module Schematics [on pages 64 through 66](#).
2. The Power and Ground nets are connected directly to the eZ80L92 device.

Almost all of the connectors' signals are received directly from the CPU. Four input signals, in particular, offer options to the application developer by disabling certain functions of the eZ80L92 Module.

These four inputs are:



- Disable Ethernet ($\overline{\text{DIS_ETH}}$)
- Disable Flash ($\overline{\text{DIS_Flash}}$)
- Flash Write Enable ($\overline{\text{FlashWE}}$)
- Disable IrDA ($\overline{\text{DIS_IrDA}}$)

These four signals are described below.

Disable Ethernet

When active Low, the $\overline{\text{DIS_ETH}}$ output signal disables the EMAC from responding to CPU requests. As a result, additional inputs, outputs, or memory devices can be used in the CS3 address space. The logic that disables the Ethernet signal is listed in [Appendix A](#) on page 77.

Disable Flash

When active Low, the $\overline{\text{DIS_Flash}}$ input signal disables the Flash chip on the eZ80L92 Module.

Flash Write Enable

When active Low, the $\overline{\text{FlashWE}}$ input signal enables Write operations on the Flash boot block of the eZ80L92 Module.

Disable IrDA

When the $\overline{\text{DIS_IrDA}}$ input signal is pulled Low, the IrDA transceiver, located on the eZ80L92 Module, is disabled. As a result, UART0 can be used with the RS232 or the RS485 interfaces on the eZ80[®] Development Platform.

Application Module Interface

An Application Module Interface is provided to allow the user to add an application-specific module to the eZ80[®] Development Platform. ZiLOG's Thermostat Application Module (not provided in the kit) is an example application-specific module that demonstrates an HVAC control system. Implementing an application module with the Application Module Inter-



face requires that the eZ80L92 Module also be mounted on the eZ80[®] Development Platform, because it (the eZ80L92 Module) contains the eZ80L92 microprocessor. To mount an application module, use the two male headers J6 and J8.

Jumper J6 carries the general-purpose port, and jumper J8 carries memory and control signals. To design an application module, the user should be familiar with the architecture and features of the eZ80L92 Module currently installed. Tables 4 and 5 list the signals and functions related to each of these jumpers by pin. Power and ground signals are omitted for the sake of simplicity.

Table 4. General-Purpose Port Connector J6*

Signal	Pin #	Function	Direction	Notes
SCL	5	I ² C Clock	Bidirectional	
SDA	7	I ² C Data	Bidirectional	
$\overline{\text{MOD_DIS}}$	9	Modem Disable	Input	If a shunt is installed between pins 6 and 9, the modem function on the eZ80 [®] Development Platform is disabled.
$\overline{\text{MWAIT}}$	13	WAIT signal for the CPU	Input	
EM_D0	15	GPIO, Bit 0	Bidirectional	
$\overline{\text{CS3}}$	17	Chip Select 3 of the CPU	Output	This signal is also present on the J8.
EM_D[7:1]	21,23,25, 27,29,31, 33	GPIO, Bit [7:1]	Bidirectional	
Reserved	35			

Note: *All of the signals are driven directly by the CPU.



Table 4. General-Purpose Port Connector J6* (Continued)

Signal	Pin #	Function	Direction	Notes
PC[7:0]	39,41,43, 45,47,49, 51,53	Port C, Bit [7:0]	Bidirectional	
ID_[2:0]	6,8,10	eZ80 [®] Development Platform ID	Output	
CON_DIS	12	Console Disable	Input	If a shunt is installed between pins 12 and 14, the Console function on the eZ80 [®] Development Platform is disabled.
Reserved	16,18			
PD[7:0]	22,24,26, 28,30,32, 34,36	Port D, Bit[7:0]	Bidirectional	
PB[7:0]	40,42,44, 46,48,50, 52,54	Port B, Bit[7:0]	Bidirectional	

Note: *All of the signals are driven directly by the CPU.



Table 5. CPU Bus Connector J8*

Signal	Pin #	Function	Direction
A[0:7]	3–10	Address Bus, Low Byte	Output
A[8:15]	13–20	Address Bus, High Byte	Output
A[16:23]	23–30	Address Bus, Upper Byte	Output
$\overline{\text{RD}}$	33	READ Signal	Output
$\overline{\text{RESET}}$	35	Push Button Reset	Output
$\overline{\text{BUSACK}}$	37	CPU Bus Acknowledge Signal	Output
$\overline{\text{NMI}}$	39	Nonmaskable Interrupt	Input
$\overline{\text{D}}[0:7]$	43–50	Data Bus	Bidirectional
$\overline{\text{CS}}[0:3]$	53–56	Chip Selects	
$\overline{\text{MEMRQ}}$	57	Memory Request	Output
$\overline{\text{WR}}$	34	WRITE Signal	Output
$\overline{\text{INSTRD}}$	36	Instruction Fetch	Output
$\overline{\text{BUSREQ}}$	38	CPU Bus Request signal	
$\overline{\text{PHY}}$	40	Clock output of the CPU	Output

Note: *All of the signals except BUSACK and INSTRD are driven by low-voltage CMOS technology (LVC) drivers.

I/O Functionality

The eZ80[®] Development Platform provides additional functionality, featuring general-purpose port, an LED matrix, a modem reset, and two user triggers. These functions are memory-mapped with an address decoder based on the Generic Array Logic GAL22IV10D (U15) device manufactured by Lattice Semiconductor, and a bidirectional latch (U16). Additionally, U15 is used to decode addresses for access to the 7x5 LED matrix.



Table 6 lists the memory map addresses to registers that allow access to the above functions. The register at address 800000h controls general-purpose port output control and LED anode register functions. The register at address 800001h controls the register functions for the LED cathode, modem reset, and user triggers. Address 800002h controls general-purpose port data.

Table 6. LED and Port Emulation Addresses

Address	Register Function	Access
800000h	LED Anode/General-Purpose Port Output Control	WR
800001h	LED Cathode/Modem/Trig	WR
800002h	General-Purpose Port Data	RD/WR

General-Purpose Port

The general-purpose port is emulated with the use of the GPIO Output Control Register and the GPIO Data Register. If bit 7 in the GPIO Output Control Register is 1, all of the lines on the general-purpose port are configured as inputs. If this bit is 0, all of the lines on the general-purpose port are configured as outputs. Table 7 lists the multiple functions of the register.

Table 7. LED Anode/General-Purpose Port Output Control Register

Function	Bit #							
	7	6	5	4	3	2	1	0
Anode Col 1								X
Anode Col 2							X	
Anode Col 3						X		
Anode Col 4					X			
Anode Col 5			X					



Table 7. LED Anode/General-Purpose Port Output Control Register (Continued)

Function	Bit #							
	7	6	5	4	3	2	1	0
Anode Col 6			X					
Anode Col 6		X						
GPIO Output	X							

The GPIO Data Register receives inputs or provides outputs for each of the seven general-purpose port lines, depending on the configuration of the port. See Table 8.

Table 8. General-Purpose Port Data Register

Function/Bit #	7	6	5	4	3	2	1	0
GPIO D0								X
GPIO D1							X	
GPIO D2						X		
GPIO D3					X			
GPIO D4				X				
GPIO D5			X					
GPIO D6		X						
GPIO D7	X							



Caution: Reading from the general-purpose port can damage the drivers used for the general-purpose port and memory. The port can, however, be used for writing data.



LED Matrix

The one 7x5 LED matrix device on the eZ80[®] Development Platform is a memory-mapped device that can be used to display information, such as programmed alphanumeric characters. For example, the LED display sample program that is shipped with this kit displays the alphanumeric message:

eZ80

To illuminate any LED in the matrix, its respective anode bit must be set to 1 and its corresponding cathode bit must be set to 0.

Bits 0–6 in Table 7 are LED anode bits. They must be set High (1) and their corresponding cathode bits, bits 0–4 in Table 9, must be set Low (0) to illuminate each of the LED's, respectively.

Bit 7 in Table 7 does not carry any significance within the LED matrix. It is used for the general-purpose port as a control bit.

Table 9 indicates the multiple register functions of the LED cathode, modem, and triggers. This table shows the bit configuration for each cathode bit. Bits 5, 6, and 7 do not carry any significance within the LED matrix. These three bits are control bits for the modem reset, Trig1, and Trig2 functions, respectively.

Table 9. Bit Access to the LED Cathode, Modem, and Triggers

Function	Bit #							
	7	6	5	4	3	2	1	0
Cathode Row 5								X
Cathode Row 4							X	
Cathode Row 3						X		
Cathode Row 2					X			



Table 9. Bit Access to the LED Cathode, Modem, and Triggers (Continued)

Function	Bit #							
	7	6	5	4	3	2	1	0
Cathode Row 1				X				
Modem RST			X					
Trig 1		X						
Trig 2	X							

An LED display sample program is shipped with the eZ80L92 Development Kit. Please refer to the eZ80[®] [Webserver-i Quick Start Guide](#) (QS0015) or to the Tutorial section in the [ZDS II User Manual](#) (UM0123).

Modem Reset

The Modem Reset signal, MRESET, is used to reset an optional socket modem. This signal is controlled by bit 5 in the register shown in Table 9. The MRESET signal is available at the embedded modem socket interface (J9, Pin 1). Setting this bit Low places the optional socket modem into a reset state. The user must pull this bit High again to enable the socket modem. Reference the appropriate documentation for the socket modem to reset timing requirements.

User Triggers

Two general-purpose trigger output pins are provided on the eZ80[®] Development Platform. Labeled J21 (Trig2) and J22 (Trig1), these pins allow the user a way to *trigger* external equipment to aid in the debug of the system. See Figure 8 for trigger pin details.

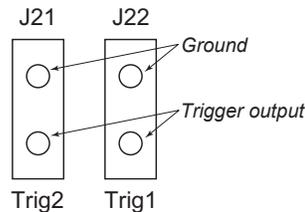


Figure 8. Trigger Pins J21 and J22

Bits 6 and 7 in Table 9 are the control bits for the user triggers. If either bit is a 1, the corresponding Trig1 and Trig2 signals are driven High. If either bit is 0, the corresponding Trig1 and Trig2 signals are driven Low.

Embedded Modem Socket Interface

The eZ80[®] Development Platform features a socket for an optional 56K modem (a modem is not included in the kit).

Connectors J1, J5, and J9 provide connection capability. The modem socket interface provided by these three connectors is shown in Figure 9. Tables 10 through 12 identify the pins for each connector. The embedded modem utilizes UART1, which is available via the Port C pins.

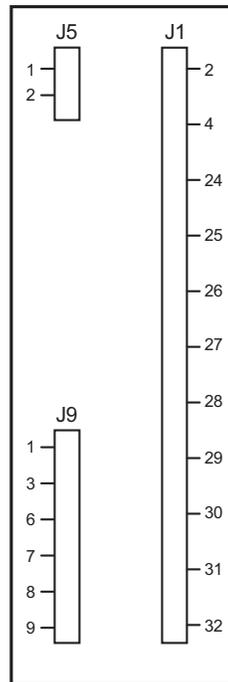


Figure 9. Embedded Modem Socket Interface—J1, J5, and J9

Table 10. Connector J5

Pin	Symbol	Description
1	M-TIP	Telephone Line Interface—TIP.
2	M-RING	Telephone Line Interface—RING.



Table 11. Connector J9

Pin	Symbol	Description
1	MRESET	Reset, active Low, 50–100ms. Closure to GND for reset.
3	GND	Ground.
6	D1	DCD indicator; can drive an LED anode without additional circuitry.
7	D2	RxD indicator; can drive an LED anode without additional circuitry.
8	D3	DTR indicator; can drive an LED anode without additional circuitry.
9	D4	TxD indicator; can drive an LED anode without additional circuitry.

Table 12. Connector J1

Pin	Symbol	Description
2	MOD_DIS	Modem disable, active Low.
4	V _{CC}	+5 VDC or +3.3 VDC input.
24	GND	Ground.
25	PC4_DTR1	DTR interface; TTL levels.
26	PC6_DCD1	DCD interface; TTL levels.
27	PC3_CTS1	CTS interface; TTL levels.
28	PC5_DSR1	DSR interface; TTL levels.
29	PC7_RI1	Ring Indicator interface; TTL levels.
30	PC0_TXD1	TxD interface; TTL levels.
31	PC1_RXD1	RxD interface; TTL levels.
32	PC2_RTS1	RTS interface; TTL levels.



Components P4, T1, C3, C4, and U11 provide the phone line interface to the modem. On the eZ80[®] Development Platform, LEDs D1, D2, D3, and D4 function as status indicators for this optional modem.

The phone line connection for the modem is for the United States only. Connecting the modem outside of the U.S. requires modification.

The tested modem for this eZ80L92 Development Kit is a Conexant socket modem, part number SF56D/SP. Information about this modem and its interface is available in the SmartSCM SocketModem data sheet (Doc. No. 101522D) from www.conexant.com.

eZ80[®] Development Platform Memory

Memory space on the eZ80[®] Development Platform consists of onboard SRAM and additional footprints.

Onboard SRAM

The eZ80[®] Development Platform features 512KB SRAM at U20. This SRAM provides the basic memory requirement for small applications development. This SRAM is in the address range B80000h–BFFFFFFh.

Additional SRAM

The amount of eZ80[®] Development Platform memory can be extended if required by adding SRAM devices. U19, U18, and U17 provide this capability. However, the user should be aware that additional SRAM must be installed in the following order:

1. U19, address range B00000h–B7FFFFh
2. U18, address range A80000h–AFFFFFFh
3. U17, address range A00000h–A7FFFFh

If SRAM memory is installed in a different order than the above sequence, SRAM will not be contiguous unless the user is able to change the address decoder, U10. Memory access decoding is performed by this



address decoder, implemented in the Generic Array Logic device, GAL22LV10D (U10).

Flash Memory

The eZ80L92 Development Kit allows Flash memories between 1 MB and 4 MB. The chips are housed in wide TSOP40 cases. Flash ROM access times are 55–150 ns; typically 90 ns.

When accessing Flash memory, the eZ80L92 device should be configured to operate in Intel bus mode to satisfy setup and hold times and to prevent bus contention with a Write cycle that could possibly follow. For proper CPU operation at 48 MHz, first set the bus mode control register CS0_BMC (I/O address F0h) to 82h, then set the Chip Select Control register CS0_CTL (I/O address AAh) to 08h. These settings select Intel Bus Mode with two system clocks per bus cycle and zero wait states.

Memory Map

A memory map of the eZ80[®] CPU is illustrated in Figure 10. Flash memory and SRAM on the eZ80L92 Module are addressed when CS0 and CS1 are active Low. SRAM on the eZ80[®] Development Platform is addressed when CS2 is active Low.

- **Note:** The Ethernet controller, located on the eZ80L92 Module, is mapped as an I/O device at address 300h. It uses CS3.

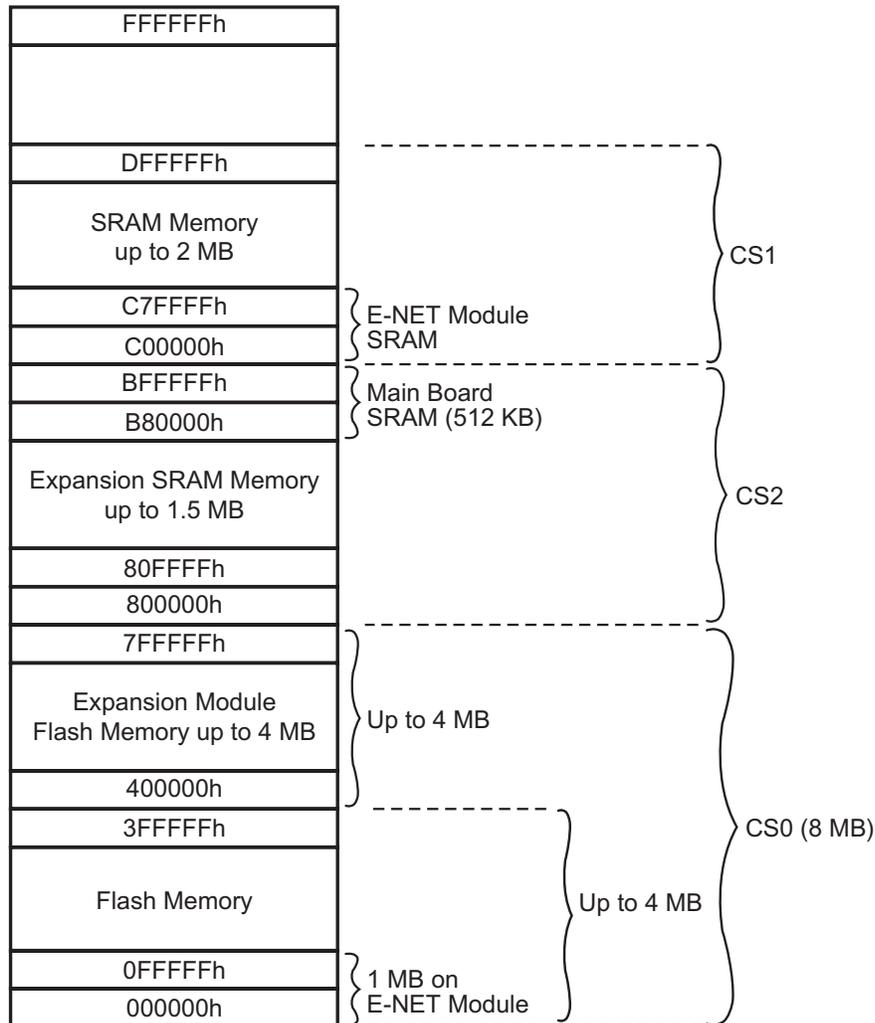


Figure 10. Memory Map of the eZ80[®] Development Platform and eZ80L92 Module

The LED matrix and the general-purpose port circuitry are mapped in the address range 800000h–80ffffh. The CS2 chip select should be driven Low to select the LED matrix or general-purpose port.



LEDs

As stated earlier, LEDs D1, D2, D3, and D4 function as status indicators for an optional modem. This section describes each LED and the LED matrix device.

Data Carrier Detect

The Data Carrier Detect (DCD) signal at D1 indicates that a good carrier signal is being received from the remote modem.

RX

The RX signal at D2 indicates that data is received from the modem.

Data Terminal Ready

The Data Terminal Ready (DTR) signal at D3 informs the modem that the PC is ready.

TX

The TX signal at D4 indicates that data is transmitted to the modem.

Push Buttons

The eZ80[®] Development Platform provides user controls in the form of push buttons. These push buttons serve as input devices to the eZ80L92 microprocessor. The programmer can use them as necessary for application development. All push buttons are connected to the general-purpose port pins.

PB0

The PB0 push button switch, SW1, is connected to bit 0 of the general-purpose port. This switch can be used as the port input if required by the user.



PB1

The PB1 push button switch, SW2, is connected to bit 1 of the general-purpose port. This switch can be used as the port input if required by the user.

PB2

The PB2 push button switch, SW3, is connected to bit 2 of the general-purpose port. This switch can be used as the port input if required by the user.

RESET

The Reset push button switch, SW4, resets the eZ80[®] CPU and the eZ80[®] Development Platform.

Jumpers

The eZ80[®] Development Platform provides a number of jumpers that are used to enable or disable functionality on the platform, enable or disable optional features, or to provide protection from inadvertent use.

Jumper J2

The J2 jumper connection enables/disables IrDA transceiver functionality. When the shunt is placed, IrDA communication is disabled. See Table 13.

Table 13. J2—DIS_IrDA

Shunt Status	Function	Affected Device
IN	IrDA interface disabled	UART0 is configured to work with the RS232 or the RS485 interfaces.
OUT	IrDA interface enabled	The IrDA and UART0 interfaces on the eZ80L92 Module perform their functions.



Jumper J3

The J3 jumper connection controls the mode of the general-purpose port and communication with the 7x5 LED. When the shunt is placed, the general-purpose port is disabled. See Table 14.

Table 14. J3—DIS_EM

Shunt Status	Function	Affected Device
IN	Application Module Hardware Disabled	Communication with 7x5 LED and Port emulation circuit is disabled.
OUT	Application Module Hardware Enabled	Communication with 7x5 LED and the general-purpose port circuit is enabled.

Jumper J7

The J7 jumper connection controls Flash boot loader programming. When the shunt is placed, overwriting of the Flash boot loader program is enabled. See Table 15.

Table 15. J7—FlashWE

Shunt Status	Function	Affected Device
OUT	The Flash boot sector of the eZ80L92 Module is write-protected.	Flash boot sector of the eZ80L92 Module.
IN	The Flash boot sector of the eZ80L92 Module is enabled for writing or overwriting.	Flash boot sector of the eZ80L92 Module.



Jumper J11

The J11 jumper connection controls access to the Flash memory device. When the shunt is placed, access to the Flash device is disabled/prevented. See Table 16.

Table 16. J11—DIS_Flash

Shunt Status	Function	Affected Device
IN	All access to Flash on the eZ80L92 Module is disabled.	Flash on eZ80L92 Module.
OUT	Flash on the eZ80L92 Module is enabled.	Flash on eZ80L92 Module.

Jumper J12

The J12 jumper connection controls the selection of a 5 V or 3 VDC power supply to the embedded modem, if an embedded modem is used. See Table 17.

Table 17. J12—5VDC/3.3VDC for an Embedded Modem

Shunt Status	Function	Affected Device
1–2	5VDC is provided to power the embedded modem.	Embedded modem.
2–3	3.3VDC is provided to power the embedded modem.	Embedded modem.



Jumper J14

The J14 jumper connection controls the polarity of the Ring Indicator. See Table 18.

Table 18. J14—RI

Shunt Status	Function	Affected Device
1–2	The Ring Indicator for UART1 is inverted.	UART1.
2–3	The Ring Indicator for UART1 is not inverted.	UART1.

Jumper J15

The J15 jumper connection controls the selection RS485 circuit along with UART0. When the shunt is placed, the RS485 circuit is enabled. See Table 19. RS485 functionality will be available in future eZ80[®] devices.

Table 19. J15—RS485_1_EN*

Shunt Status	Function	Affected Device
IN	The RS485 circuit is enabled on UART0. The UART0 CONSOLE interface and IrDA are disabled.	IrDA, UART0 CONSOLE interface, RS485 interface.
OUT	The RS485 circuit is disabled on UART0.	IrDA, UART0 CONSOLE interface, RS485 interface.

Note: *To enable the RS485 circuit, the corresponding IrDA/RS232 circuit must be disabled.



Jumper J16

The J16 jumper connection controls the selection of the RS485 circuit. However, UART1 MODEM interface and the socket modem interface are disabled if the RS485 circuit is enabled. When the shunt is placed, the RS485 circuit is enabled. See Table 20.

Table 20. J16—RS485_2_EN

Shunt Status	Function	Affected Device
IN	The RS485 circuit is enabled on UART1. The UART1 MODEM interface and the Socket Modem interface are disabled.	UART1 MODEM interface, Socket Modem Interface, and RS485 interface.
OUT	The RS485 circuit is disabled on UART1.	UART1 MODEM interface, Socket Modem Interface, and RS485 interface.

Jumper J17

The J17 jumper connection controls the selection of the RS485 termination resistor circuit. When the shunt is placed, the RS485 termination resistor circuit is enabled. See Table 21.

Table 21. J17—RT_1*

Shunt Status	Function	Affected Device
IN	The Termination Resistor for RS485_1 is IN.	RS485 interface.
OUT	The Termination Resistor for RS485_1 is OUT.	RS485 interface.

Note: *Before enabling the termination resistor, ensure that the device is located at the end of the interface line.



Jumper J18

The J18 jumper connection controls the selection of the RS485 termination resistor circuit. When the shunt is placed, the RS485 termination resistor circuit is enabled. See Table 22.

Table 22. J18—RT_2*

Shunt Status	Function	Affected Device
IN	The Termination Resistor for RS485_2 is IN.	RS485 interface.
OUT	The Termination Resistor for RS485_2 is OUT.	RS485 interface.

Note: *Before enabling the termination resistor, ensure that the device is located at the end of the interface line.

Connectors

A number of connectors are available for connecting external devices such as the ZPAKII emulator, PC serial ports, external modems, the console, and LAN/telephone lines.

J6 and J8 are the headers, or connectors, that provide pin-outs to connect any external application module, such as ZiLOG's Thermostat Application Module.

Connector J6

The J6 connector provides pin-outs to make use of GPIO functionality.

Connector J8

The J8 connector provides pin-outs to access memory and other control signals.



Console

Connector P2 is the RS232 terminal, which can be used for observing the console output. P2 can be connected to the HyperTerminal if required.

Modem

Connector P3 provides a terminal for connecting an external modem, if used with the eZ80L92 Development Kit. RS485 functionality will be available in future eZ80[®] devices.

I²C Devices

The two I²C devices on the eZ80[®] Development Platform are the U2 EEPROM and the U13 Configuration register. The EEPROM provides 16KB of memory. The Configuration register provides access to control the configuration of an application-specific function at the Application Module Interface. Neither device is utilized by the eZ80L92 Development Kit software. The user is free to develop proprietary software for these two devices. The addresses for accessing these devices are listed in Table 23.

Table 23. I²C Addresses

Device/Bit #	7	6	5	4	3	2	1	0
EEPROM (U10)*	1	0	1	0	0	A1	A0	R/W
Configuration Register (U13)	1	0	0	1	1	1	0	R/W

Note: *EEPROM address bits A0 and A1 are configured for 0s.

DC Characteristics

Understanding proper DC current requirements for the eZ80[®] Development Platform when application modules are plugged into it is very important for developing applications. This section provides an estimate



of the average current requirement when different combinations of these application modules are plugged in to the eZ80[®] Development Platform.

The receiver supply current is 90–150 μ A and the transmitter supply current is 260mA when the LED is active. The measurements of current that are shown in Table 24 are for the user’s reference. These values can vary depending on the type of application that is developed to run with the platform.

Table 24. DC Current Characteristics of the eZ80[®] Development Platform with Different Module Loads

Platform/Modules Configurations	Current Requirement (mA)	Status
eZ80 [®] Development Platform and eZ80L92 Module	173	When connected only to a power supply, and when no program is running.
eZ80 [®] Development Platform, eZ80L92 Module, and Modem Module	174	When connected only to a power supply, and when no program is running.
eZ80 [®] Development Platform, eZ80L92 Module, and Thermostat Application Module	195	When connected only to a power supply, and when no program is running.
eZ80 [®] Development Platform, eZ80L92 Module, Modem Module, and Thermostat Application Module	203	When connected only to a power supply, and when no program is running.
eZ80 [®] Development Platform and eZ80L92 Module	325	When the LED demo is running.
eZ80 [®] Development Platform, eZ80L92 Module, and Modem Module	325	When the LED demo is running.



Table 24. DC Current Characteristics of the eZ80[®] Development Platform with Different Module Loads (Continued)

Platform/Modules Configurations	Current Requirement (mA)	Status
eZ80 [®] Development Platform, eZ80L92 Module, and Thermostat Application Module	350	When the LED demo is running.
eZ80 [®] Development Platform, eZ80L92 Module, Modem Module, and Thermostat Application Module	360	When the LED demo is running.



eZ80L92 Module

This section describes the eZ80L92 Module hardware, its interfaces and key components, including the CPU, Ethernet Media Access Control, and memory.

Functional Description

The eZ80L92 Module is a compact, high-performance Ethernet module specially designed for the rapid development and deployment of embedded systems requiring control and Internet/Intranet connectivity via Ethernet and/or IrDA. Additional devices such as serial ports, LED matrices, general-purpose port, and I²C devices are supported when connected to the eZ80[®] Development Platform. A block diagram representing both of these boards is shown in [Figure 1](#) on page 4.

The eZ80L92 Module is developed to be a plug-in module to the eZ80[®] Development Platform. This small-footprint module provides a CPU, RAM, Flash memory, an IrDA transceiver, and an Ethernet Media Access Controller (EMAC). This low-cost, expandable module is powered by the eZ80L92 microprocessor, a member of ZILOG's new eZ80[®] product family. The module also contains a battery and an oscillator in support of the on-chip Real-Time Clock (RTC). The eZ80L92 Module can also be used as a stand-alone development tool when provided with an external power source.

Physical Dimensions

The dimensions of the eZ80L92 Module PCB is 64x64mm. With an RJ-45 Ethernet connector, the overall height is 25 mm. See Figure 11.

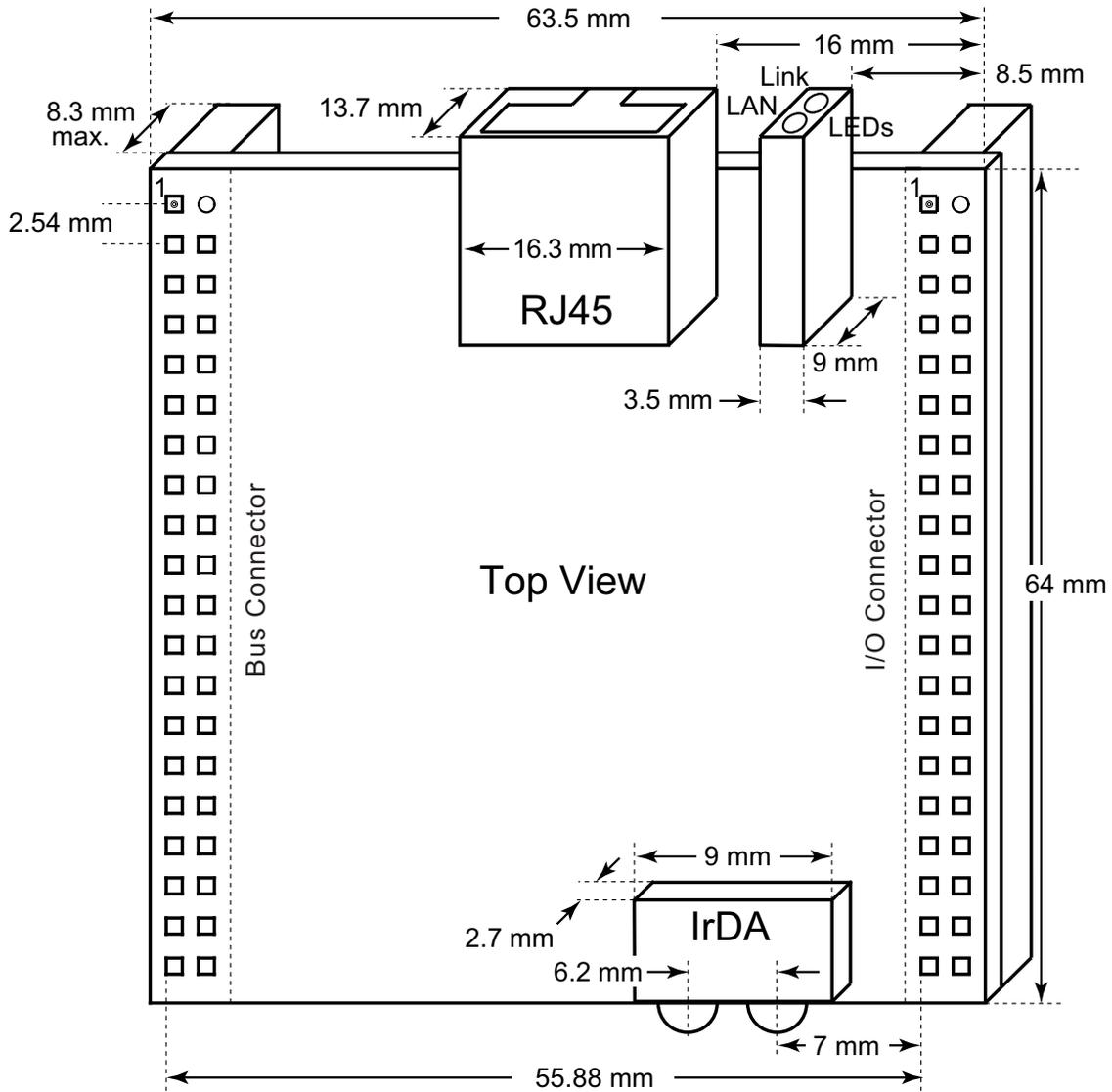


Figure 11. Physical Dimensions of the eZ80L92 Module

Figure 13 illustrates the bottom layer silkscreen of the eZ80L92 Module.

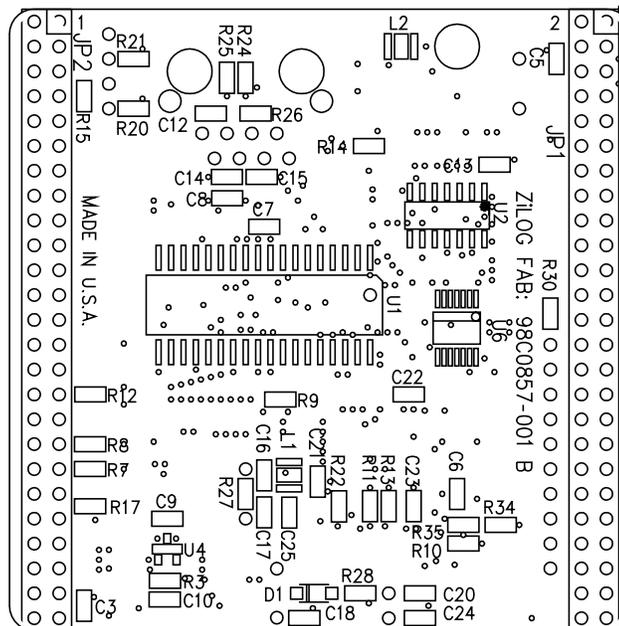


Figure 13. Bottom Layer

Operational Description

The purpose of the eZ80L92 Module as a feature of the eZ80L92 Development Kit is to provide the application developer with a plug-in tool to evaluate the EMAC, memory, IrDA, and other features of the eZ80L92 device.

Ethernet Media Access Controller

The eZ80L92 Development Kit contains a CS8900A Ethernet Media Access Controller (EMAC—controls MAC and PHY functions) which is attached to the data/address bus (A0–A3, D0–D7, RD, and WR) of the processor. This chip is connected to the processor's CS3 Chip Select, and to the PD4 pins for interrupt purposes. Connection of pins PD6 and PD7



for LANACT (wake-up from sleep) and SLEEP is optional and resistor-selectable onboard (see below). Details about the internal registers of the CS8900A EMAC can be found on the Cirrus Logic website at www.cirrus.com.

Ethernet LEDs

There are two green LEDs, a Link LED and a LAN LED, that are located adjacent to each other on the eZ80L92 Module. A steady LAN LED (top) indicates received link pulses from the 10Base-T Ethernet. This LAN LED should be ON if RX+ is connected to TX+ and RX- is connected to TX-. A flashing Link LED (bottom) indicates Traffic (RX or TX) on the LAN.

Ethernet Connector

The eZ80L92 Development Kit is equipped with an RJ-45 connector that features integrated magnetics (transformer, common mode chokes). The remaining pins on the onboard RJ-45 connector are not connected.

An RJ-45 loopback connector can be used to verify the correct operation of the Receiver and the Transmitter. Pin assignments for the RJ-45 Ethernet connector are shown in Table 25.

Table 25. Ethernet Connector Pin Assignments

Pin	Function
1	TX+
2	TX-
3	RX+
6	RX-

To connect the eZ80L92 Development Kit directly to another host (e.g., to a personal computer), a crossover cable must be used.



The EMAC can be additionally protected by placing an ESD protection array on the module at U9. This array can be either of the LCDA15C-6 (Semtech) or ESDA25B1 (ST Microelectronics) devices.

GPIO Pins for Enabling LAN Activity, Sleep, Interrupt

GPIO input bit PD4 serves as an active High interrupt input for the EMAC's INTRQ0 output.

GPIO output bit PD7 can be used to enter the EMAC into SLEEP mode. When pulling SLEEP (PD7) Low after enabling HWStandbyE and HWSleepE modes, the chip draws lower current, because only the receiver is operating. A zero-Ohm resistor at position R14 on the eZ80L92 Development Kit is required for this function.

If LAN activity is detected, the LANACT signal is pulled Low. The LANACT is connected to GPIO input PD6 and can be used in interrupt edge-detection mode to wake up and reinitialize the Ethernet chip. A zero-Ohm resistor at position R15 on the module is required for this function. In this case, the PD6 pin is not available for GPIO on the I/O connector.

EMAC Ports

Chip Select CS3 is used for selecting the EMAC device. The base address is user-selectable. The EMAC is connected as an 8-bit device.

EMAC Wait States

The CS8900A EMAC should be operated in Intel bus mode so that the setup and hold times for the I/O access are met. For 48MHz operation, first set CS3_BMC (I/O address F3h) to 84h (Intel bus mode with four system clock cycles per bus cycle) and then CS3_CTL (I/O Address B3h) to 18h (0 wait states for I/O). For a 20.8ns CPU Clock cycle time, the READ and WRITE access time is:

$$2 \times 4 \times 20.8\text{ns} - 16\text{ns (for capacitive and chip delays)} = 150\text{ns}$$



eZ80L92 Module Memory

The eZ80L92 Module contains 512KB SRAM and 1MB Flash memory. This addressing structure provides 1MB of contiguous SRAM for immediate use.

SRAM Memory

The eZ80L92 Module features 512KB of fast SRAM. Access speed is typically 12ns or faster, allowing zero-wait-state operation at 48MHz. With the CPU at 48MHz, onboard SRAM can be accessed with zero wait states in eZ80 mode. CS1_CTL (chip select CS1) can be set to 08h (no wait states).

Flash Memory

The Flash Boot Loader, application code, and user configuration data are held permanently in Flash memory. As an example, for 128KB onboard SRAM, 1MB of ROM is required.

Reset Generator

The onboard Reset Generator Chip is connected to the eZ80L92 Reset input pin. It performs reliable Power-On Reset functions, generating a reset pulse with a duration of 200ms if the power supply drops below 2.93V. This reset pulse ensures that the board always starts in a defined condition. The RESET pin on the I/O connector reflects the status of the RESET line. It is a bidirectional pin for resetting external peripheral components or for resetting the eZ80L92 Development Kit with a low-impedance output (e.g. a 100-Ohm push button).

IrDA Transceiver

An onboard IrDA transceiver (ZiLOG ZHX1810) is connected to PD0 (TX), PD1 (RX), and PD2 (Shutdown, IR_SD). The IrDA transceiver is of the LED type 870nm Class 1.



The IrDA transceiver is accessible via the IrDA controller attached to UART0 on the eZ80L92 device. The UART0 console and the IrDA transceiver cannot be used simultaneously.

To use the UART0 for console or to save power, the transceiver can be disabled by the software or by an off-board signal when using the proper jumper selection. The transceiver is disabled by setting PD2 (IR_SD) High or by pulling the $\overline{\text{DIS_IRDA}}$ pin on the I/O connector Low. The shutdown feature is used for power savings. To enable the IrDA transceiver, $\overline{\text{DIS_IRDA}}$ is left floating and PD2 is set to Low.

The eZ80L92 Module contains a ZiLOG IrDA transceiver that is connected to the UART0 port. This port can be used as a *wireless* connection into the eZ80L92 Module. The UART0 can connect to a standard RS232 port, or it can be configured to control the IrDA transceiver; however, it cannot do both at the same time. Only a few registers are required to configure the UART0 port to send and receive IrDA data.

The RxD and TxD signals on the transceiver perform the same functions as a standard RS232 port. However, these signals are processed as IrDA 3/16 coding pulses (sometimes called IrDA encoder/decoder pulses). When the IrDA function is enabled, the final output to the RxD and TxD pins are routed through the 3/16 pulse generator.

Another signal that is used in the eZ80L92 Module's IrDA system is **Shut_Down** (SD). The SD pin is connected to PD2 on the eZ80L92 Module. The IrDA control software on the user's wireless device must enable this pin to wake the IrDA transceiver. The SD pin must be set Low to enable the IrDA transceiver. On the eZ80L92 Module, a two-input OR gate is used to allow an external pin to shut down the IrDA transceiver. Both pins must be set Low to enable this function.

Figure 14 highlights the eZ80L92 Module IrDA hardware connections.

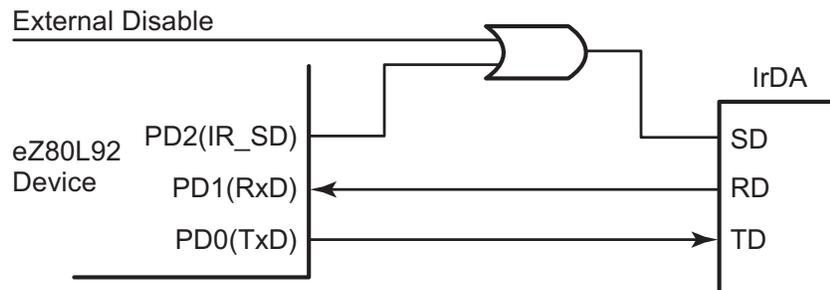


Figure 14. IrDA Hardware Connections

The eZ80L92 Module features an Infrared Encoder/Decoder register that configures the IrDA function. This register is located at address 0BFh in the internal I/O register map.

The Infrared Encoder/Decoder register contains three control bits. Bit 0 enables or disables the IrDA encoder/decoder block. Bit 1, if it is set, enables received data to pass into the UART0 Receive FIFO data buffer. Bit 2 is a test function that provides a loopback sequence from the TxD pin to the RxD input.

Bit 1, the Receive Enable bit, is used to block data from filling up the Receive FIFO when the eZ80L92 Module is transmitting data. Because IrDA data passes through the air as a light source, transmitted data can also be received. This Receive Enable bit prevents this data from being received. After the eZ80L92 Module completes transmitting, this bit is changed to allow for incoming messages.

The code below provides an example of how this function is enabled on the eZ80L92 Module.

```
//Init_IRDA
// Make sure to first set PD2 as a port bit, an output and set it Low.

PD_ALT1 &= 0xFC;           // PD0 = uart0tx, PD1 = uart0_rx
PD_ALT2 |= 0x03;          // Enable alternate function
UART_LCTL0= 0x80;         // Select dlab to access baud rate generator
```



```
BRG_DLRL0=0x2F;           // Baud rate Masterclock/(16*baudrate)
BRG_DLRH0=0x00;           // High byte of baud rate
UART_LCTL0=0x00;          // Disable dlab
UART_FCTL0=0xC7;          // Clear tx fifo, enable fifo
UART_LCTL0=0x03;          // 8bit, N, 1 stop
IR_CTL = 0x03;             // enable IRDA Encode/decode and Receive
                           // enable bit.

//IRDA_Xmit

IR_CTL = 0x01;             //Disable receive
Puchar(0xb0);              //Output a byte to the uart0 port.
```

DC Characteristics

As different combinations of application modules are loaded onto the eZ80[®] Development Platform, current requirements change. Please see [Table 24](#) on page 42 to reference current consumption values for these different module combinations.

A 0.1-Farad capacitor is provided on the eZ80L92 Module as a short-term battery backup for the RTC (see the [eZ80L92 Module Schematics](#)). The part number of the capacitor made by Panasonic is EECS0HDV. The capacitor is connected to RTC_VDD to provide power to the RTC when main power to the chip is removed; it is also connected to the 3.3 V supply to the chip for recharging. The RTC can operate down to 3.0 V; it requires 10 μ A of current. The (keep alive) time this capacitor can supply power to the RTC, from 3.3 V to 3.0 V, is approximately 3000 seconds, or 50 minutes.

Flash Loader Utility

The Flash Loader utility resides in the boot sector of Flash memory, located on the eZ80L92 Module. The Flash Loader utility allows the operation of the Boot Block utility or jumping to user application code.

Please refer to the [Flash Loader Installation Product User Guide](#) (PUG0013) for more details.



Mounting the Module

When mounting the eZ80L92 Module onto the eZ80[®] Development Platform, check its orientation to the platform to ensure a correct fit. Pin 1 of JP1 on the eZ80L92 Module must align with pin 1 of JP1 on the eZ80[®] Development Platform; Pin 1 of JP2 on the eZ80L92 Module must align with pin 1 of JP2 on the eZ80[®] Development Platform, etc.

Changing the Power Supply Plug

The universal 9VDC power supply offers three different plug configurations and a tool that aids in removing one plug configuration to insert another, as shown in Figure 15.



Figure 15. 9VDC Universal Power Supply Components

To exchange one plug configuration for another, perform the following steps:

1. Place the tip of the removal tool into the round hole at the top of the current plug configuration.
2. Press down to disengage the keeper tab and push the plug configuration out of its slot.

3. Select the plug configuration appropriate for your location, and insert it into the slot formerly occupied by the previous plug configuration.
4. Push the new plug configuration down until it snaps into place, as indicated in Figure 16.

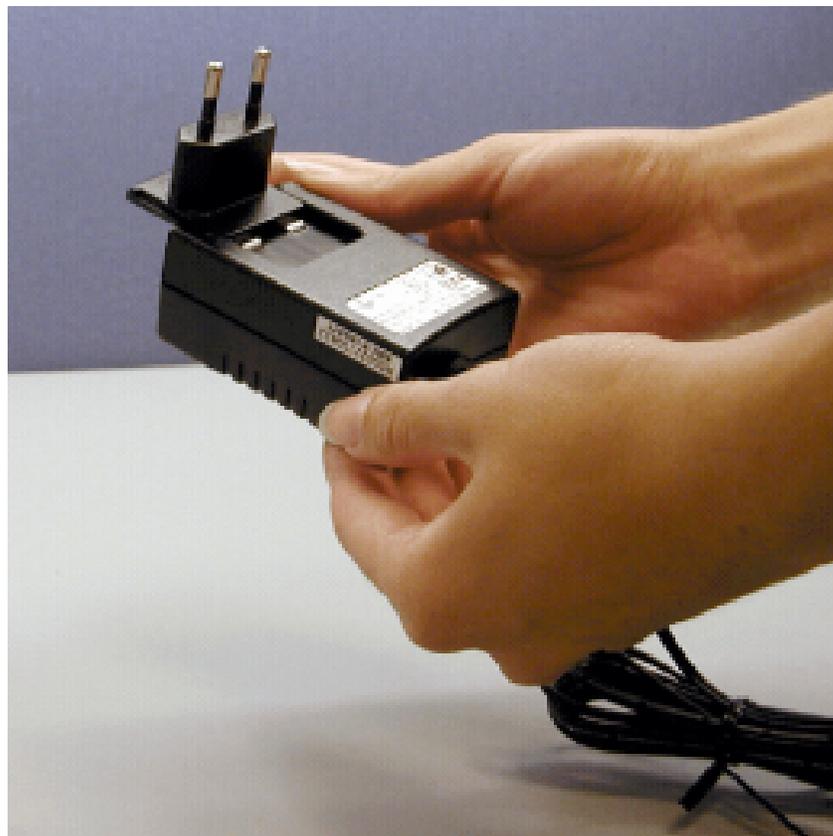


Figure 16. Inserting a New Plug Configuration



ZPAKII

ZPAKII is a debug tool used to develop and debug hardware and software. It is a networked device featuring an Ethernet interface and an RS232 console port. ZPAKII is shipped with a preconfigured IP address that can be changed to suit the user on a local network. For more information about using and configuring ZPAKII, please refer to the [ZPAKII Debug Interface Tool Product User Guide](#) (PUG0015) and the [eZ80L92 Quick Start Guide](#) (QS0015).

ZDI Target Interface Module

The ZDI Target Interface Module provides a physical interface between ZPAKII and the eZ80[®] Development Platform. The TIM module supports ZDI functions. For more information on using the TIM module or ZDI please refer to the [eZ80L92 Quick Start Guide](#) (QS0015) and the [eZ80L92 Module Product Specification](#) (PS0170).

JTAG

Connector P1 is the JTAG connector on the eZ80[®] Development Platform. JTAG will be supported in the next offering of eZ80[®] products.

Application Modules

ZiLOG offers the Thermostat Application module, which can be used for evaluating and developing process control and simple I/O applications. The Thermostat Application module is equipped with an LCD display that can be used to display process control and other physical parameters. For additional reading about the Thermostat application, please see the [Java Thermostat Demo Application Note](#) (AN0104) on zilog.com.



ZDS II

ZiLOG Developer Studio II (ZDS II) Integrated Development Environment is a complete stand-alone system that provides a state-of-the-art development environment. Based on the Windows[®] Win98SE/NT4.0-SP6/Win2000-SP2/WinXP user interfaces, ZDS II integrates a language-sensitive editor, project manager, C-Compiler, assembler, linker, librarian, and source-level symbolic debugger that supports the eZ80[®] Webserver. For more information about using and configuring ZDS II, please refer to the [ZiLOG Developer Studio II—eZ80 User Manual](#) (UM0123).

**eZ80L92 Development Kit
User Manual**



58



Troubleshooting

Overview

Before contacting ZiLOG Customer Support to submit a problem report, please follow these simple steps. If a hardware failure is suspected, contact a local ZiLOG representative for assistance.

Cannot Download Code

If you are unable to download code to RAM using ZDS, make sure to press and release the Reset button on the eZ80[®] Development Platform prior to selecting **Build** → **Debug** → **Reset + Go** in ZDS.

No Output on Console Port

The eZ80L92 Development Kit is shipped with a Flash Loader utility that is loaded in the protected boot sector of Flash memory (U3). Upon power-up of the eZ80[®] Development Platform and the eZ80L92 Module, the eZ80L92 device on the module starts running code from this Flash memory area. This code enables the Console port with settings of 57.6kbps, 8, N, 1.

The Console checks the Receive buffer. If a *space* character is received on the Console port, the Flash Loader utility is enabled and a boot message should be displayed on your connected device. If no message is displayed, check the following:

- Jumper J2 must be ON (IrDA is disabled)
- On Connector J6, the jumper must be removed from pins 6 and 9 (pin names *con_dis* and *GND*).



IrDA Port Not Working

If you plan on using the IrDA transceiver on the eZ80L92 Module, make sure the hardware is set up as follows:

- Jumper J2 must be OFF (to enable the control gate that drives the IrDA device)
- Set port pin PD2 Low. When this port pin and Jumper J2 are turned OFF, the IrDA device is enabled.
- Install a jumper on connector J6 across pin names *con_dis* and *GND* to disable the console serial port driver

Difference Between EMAC and IP Address

Media Access Control (MAC)

Each and every Ethernet device interface to the network media (e.g., network adapter, port on a hub) contains a unique MAC address, which is hard-coded into the hardware when it is manufactured. An Ethernet device addresses a host using a unique 48-bit address called its Ethernet address or Media Access Control (MAC) address.

MAC addresses are usually represented as six colon-separated pairs of hex digits, e.g., 6:0:20:11:ac:85. The first three bytes (e.g., 6-0-20) are the manufacturer's code, which can be used to identify the manufacturer. The last three bytes are the unique station ID or serial number for the interface. This station ID is unique and is associated with a particular Ethernet device. The Data Link layer's protocol-specific header specifies the MAC address of the packet's source and destination. When a packet is sent to all hosts (broadcast), a special MAC address (ff:ff:ff:ff:ff:ff) is used.

MAC addresses uniquely identify each node in a network at the Media Access Control layer, the lowest network layer that directly interfaces with the physical media (e.g., twisted-pair wires).



On a Local Area Network or other network, the MAC address is the computer's unique hardware number. (On an Ethernet LAN, the MAC address is the same as an Ethernet address.) When it is connected to the Internet, a computer (or *host* as the Internet protocol considers it), a correspondence table relates the Internet Protocol (IP) address to the computer's physical (MAC) address on the LAN.

IP Address

An IP address is a 32-bit number that identifies each sender or receiver of information that is sent in packets across the Internet.

An IP address contains two parts: the identifier of a particular network on the Internet, and an identifier of the particular device (which can be a server or a workstation) within that network. On the Internet itself—that is, between the router that moves packets from one point to another along the route—only the network part of the address is examined.

Relationship of the IP Address to the Physical Address

The machine or physical address used within an organization's local area networks can be different than the IP address coming from the Internet. The most typical example is the 48-bit Ethernet address. TCP/IP includes the Address Resolution Protocol (ARP) that lets the administrator create a table that maps IP addresses to physical addresses.

The Ethernet MAC address of the ZPAKII

When connecting the ZPAKII serial port to a PC running HyperTerminal, hold the space bar and reset the ZPAKII.

When HyperTerminal prompts with

```
eZ80>
```

enter `e` to display the MAC address.



Resolving IP Address/Subnet Mask Conflicts

For running demos properly, the ZPAKII IP address and subnet mask must be properly configured. Please follow the instructions provided in the [eZ80L92 Quick Start Guide](#) (QS0015) to set up and run the demos on ZDS II.

Contacting ZiLOG Customer Support

For additional troubleshooting solutions, see ZDS Online Help.

For valuable information about hardware and software development tools, visit [ZiLOG Customer Support](#) online. Download the latest released version of [ZiLOG Developer Studio](#)!

Get the latest [software updates](#) from ZiLOG as soon as they are available!



Schematic Diagrams

eZ80[®] Development Platform

Figures 17 through 21 diagram the layout of the eZ80[®] Development Platform.

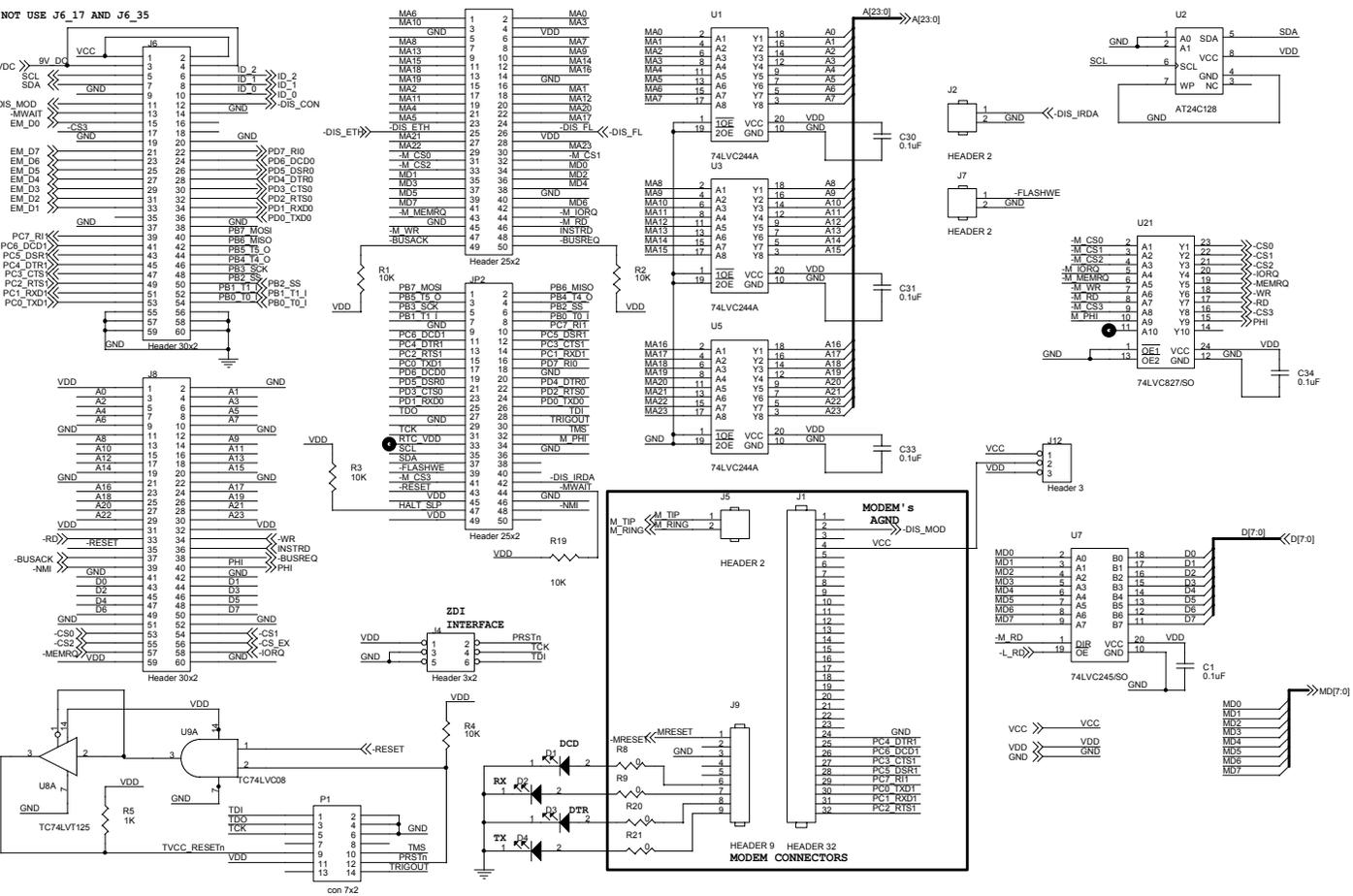


Figure 17. eZ80[®] Development Platform Schematic Diagram, #1 of 5

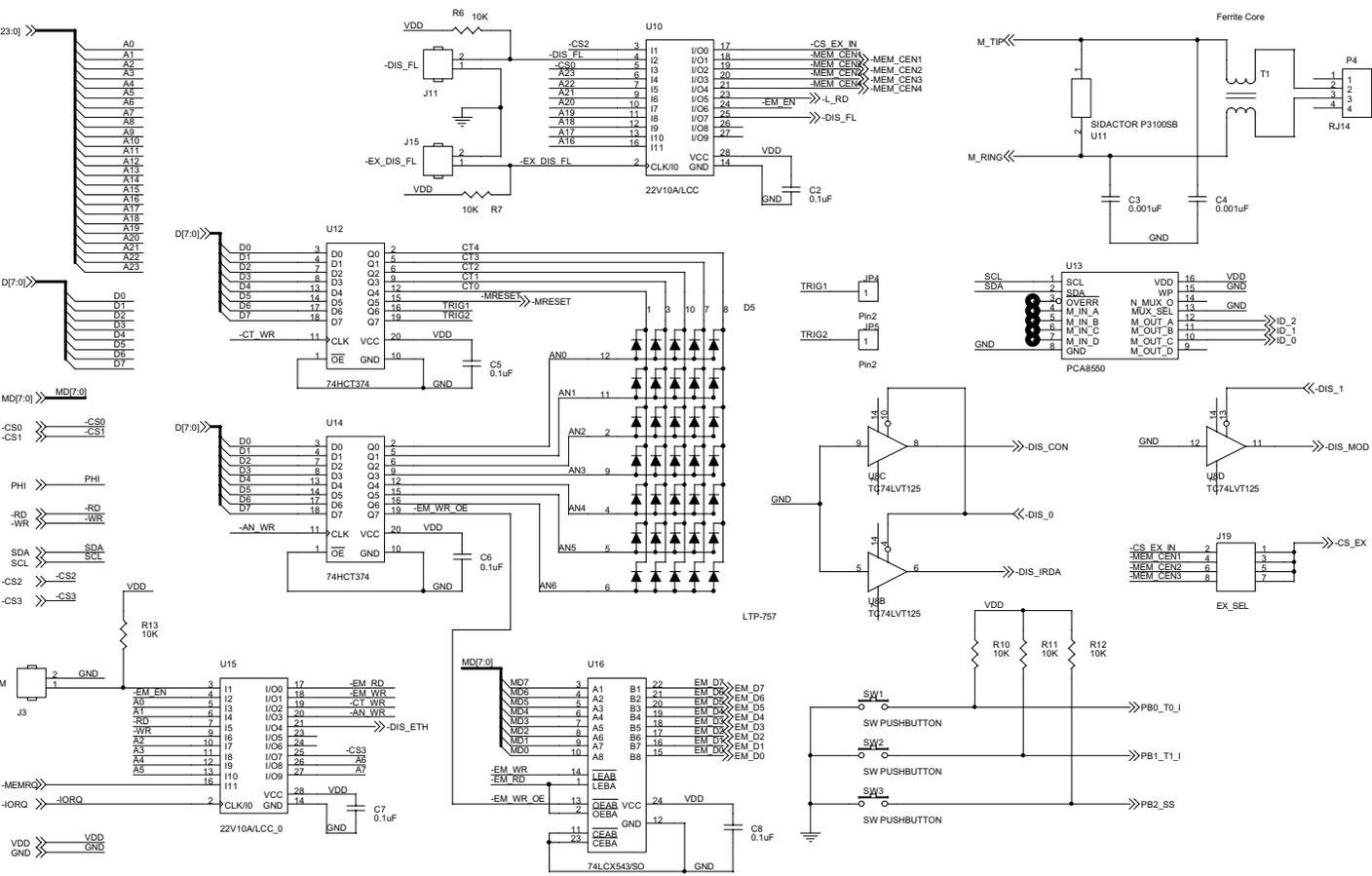


Figure 18. eZ80[®] Development Platform Schematic Diagram, #2 of 5

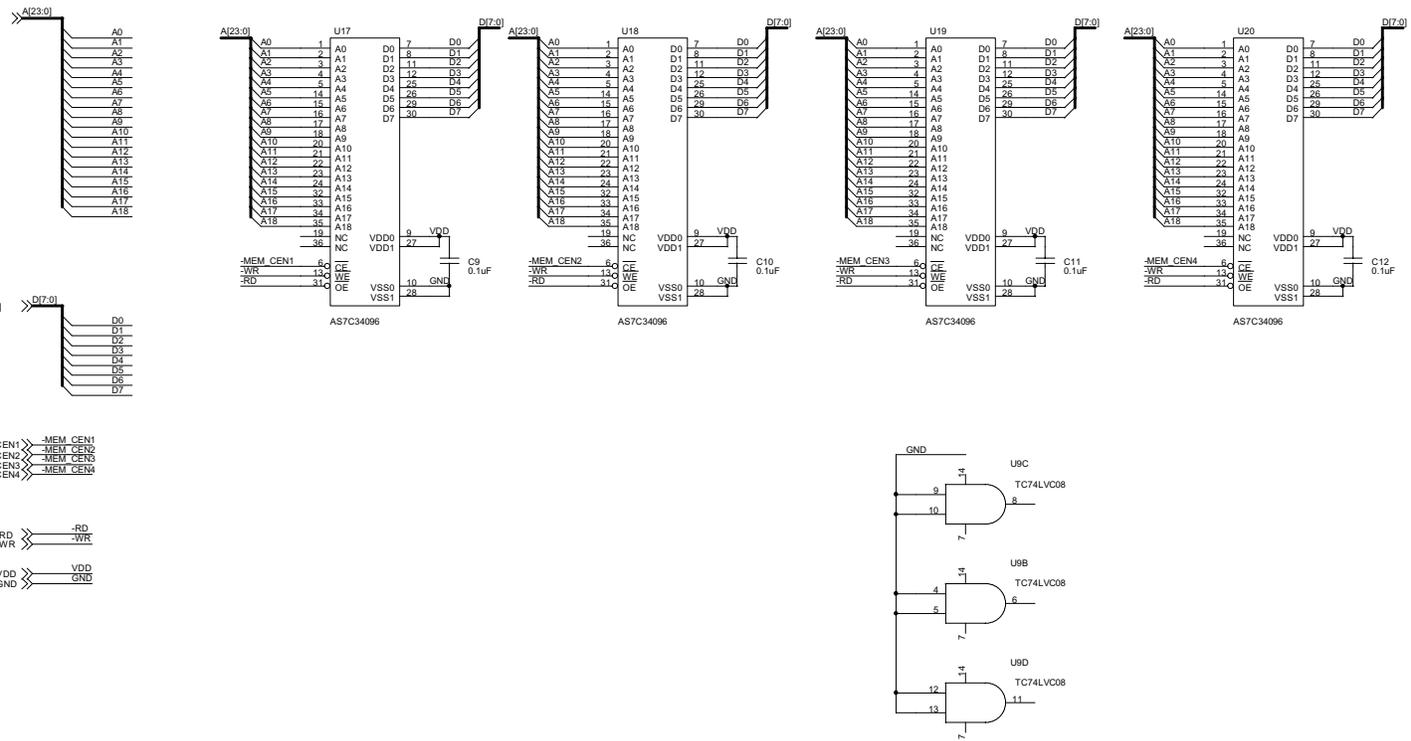


Figure 19. eZ80[®] Development Platform Schematic Diagram, #3 of 5

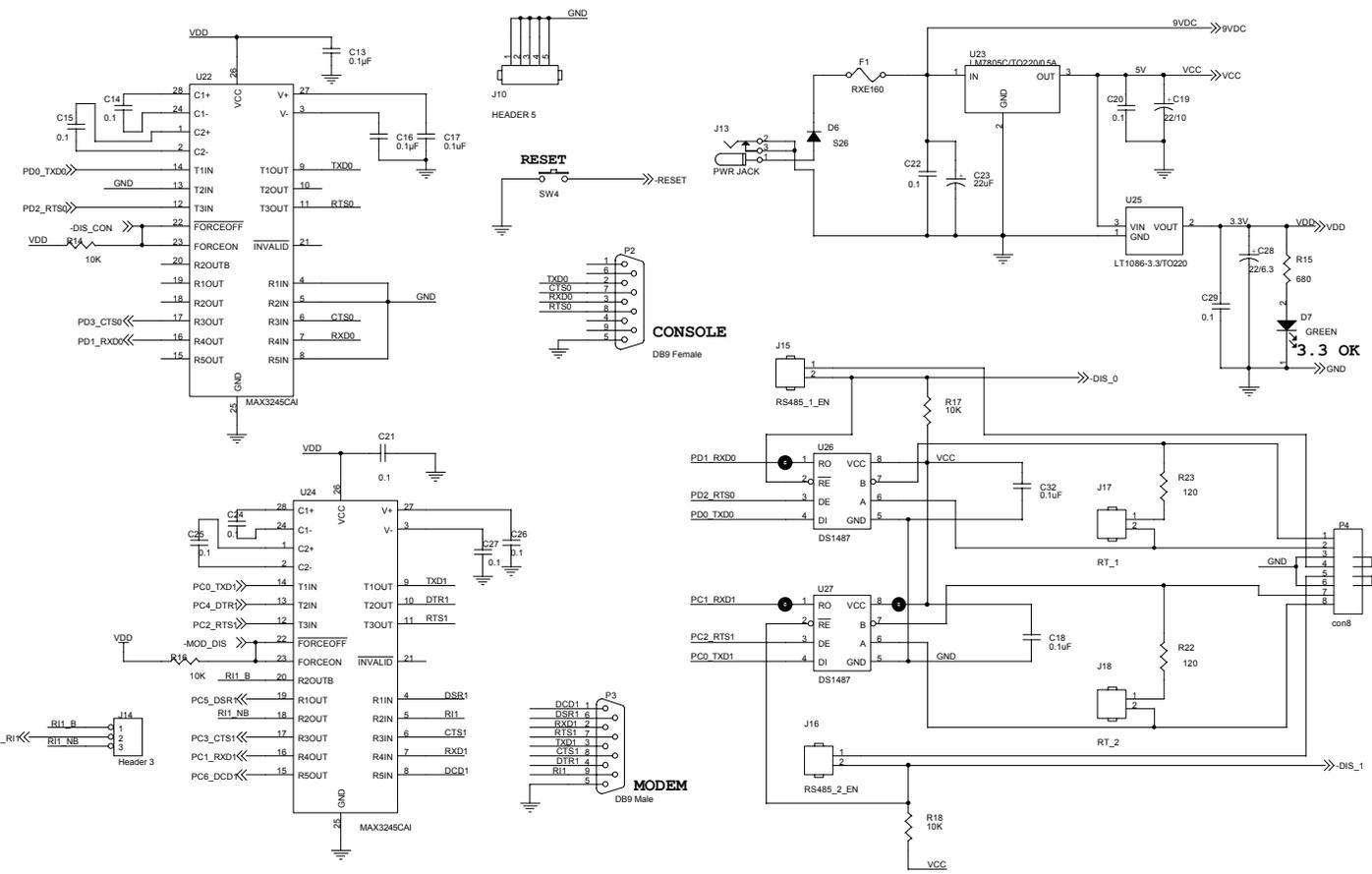


Figure 20. eZ80[®] Development Platform Schematic Diagram, #4 of 5

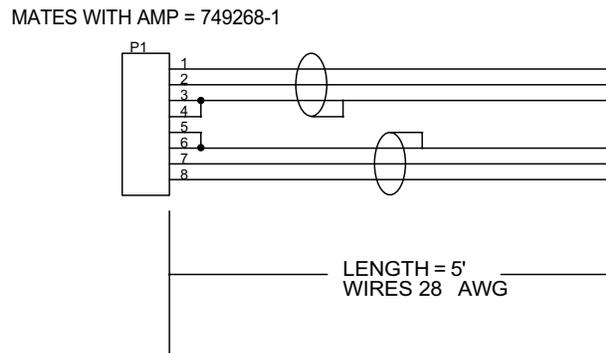


Figure 21. eZ80[®] Development Platform Schematic Diagram, #5 of 5—RS-485 Cable



eZ80L92 Module

Figures 22 through 30 diagram the layout of the eZ80L92 Module.

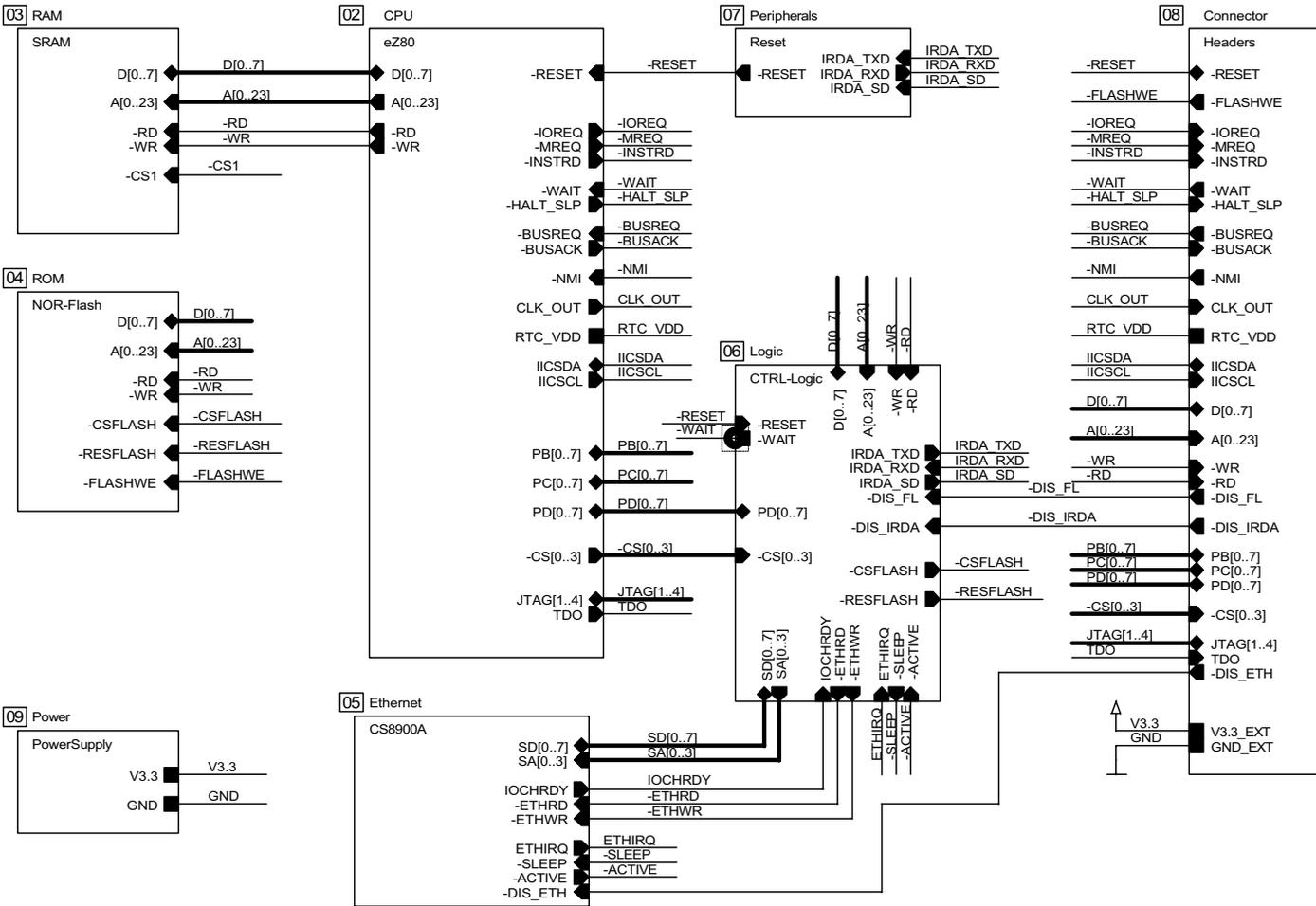
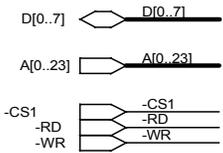


Figure 22. eZ80L92 Module Schematic Diagram, #1 of 9—Top Level



A21/A22/A23
not used
here

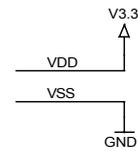
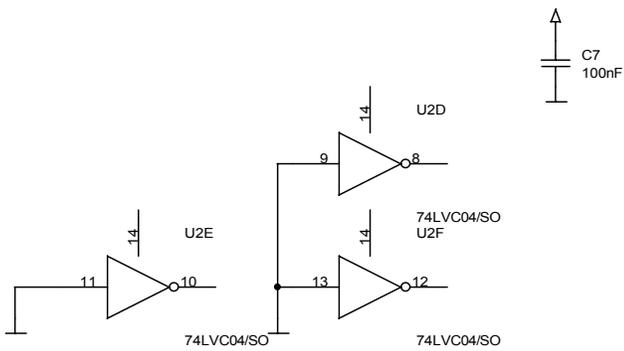
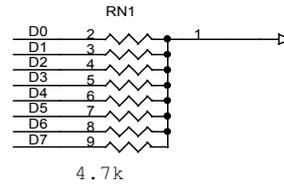
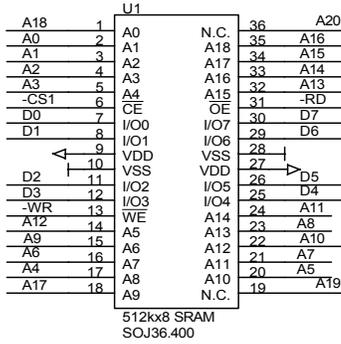


Figure 24. eZ80L92 Module Schematic Diagram, #3 of 9—36-Pin SRAM Device

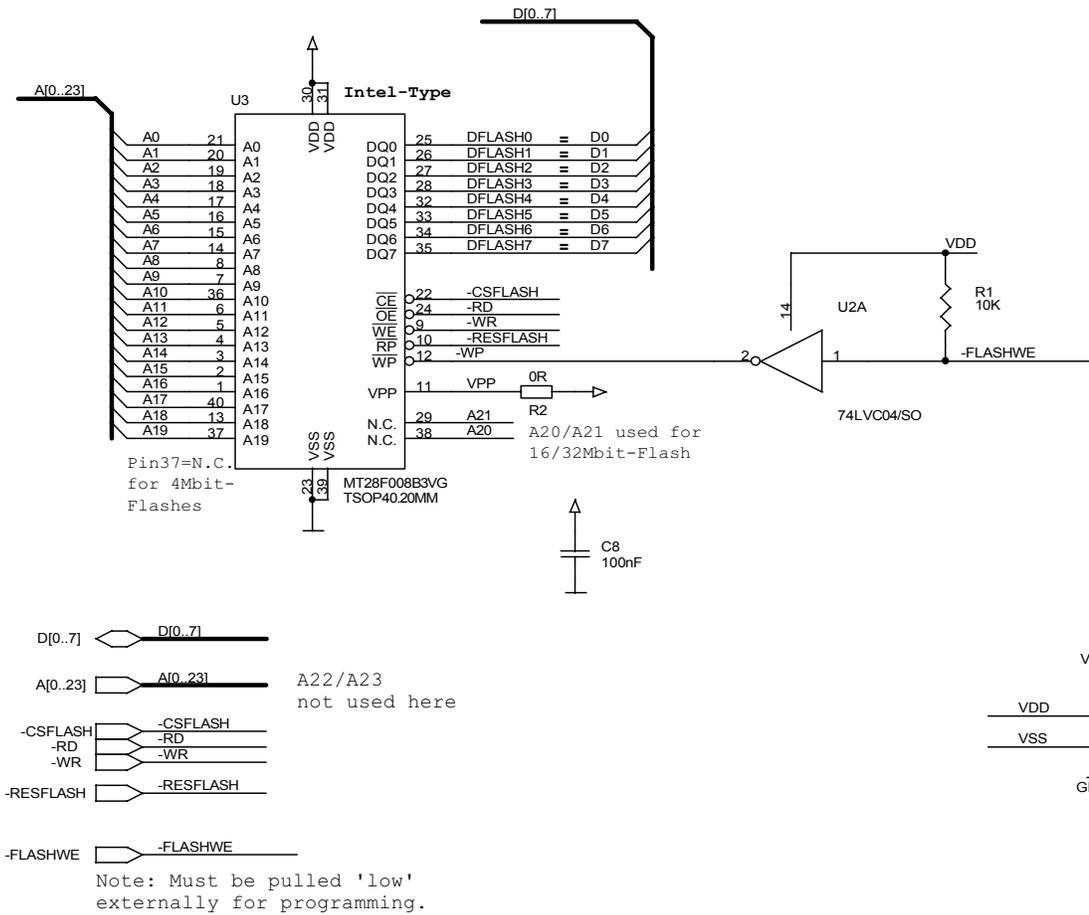


Figure 25. eZ80L92 Module Schematic Diagram, #4 of 9—Flash Device

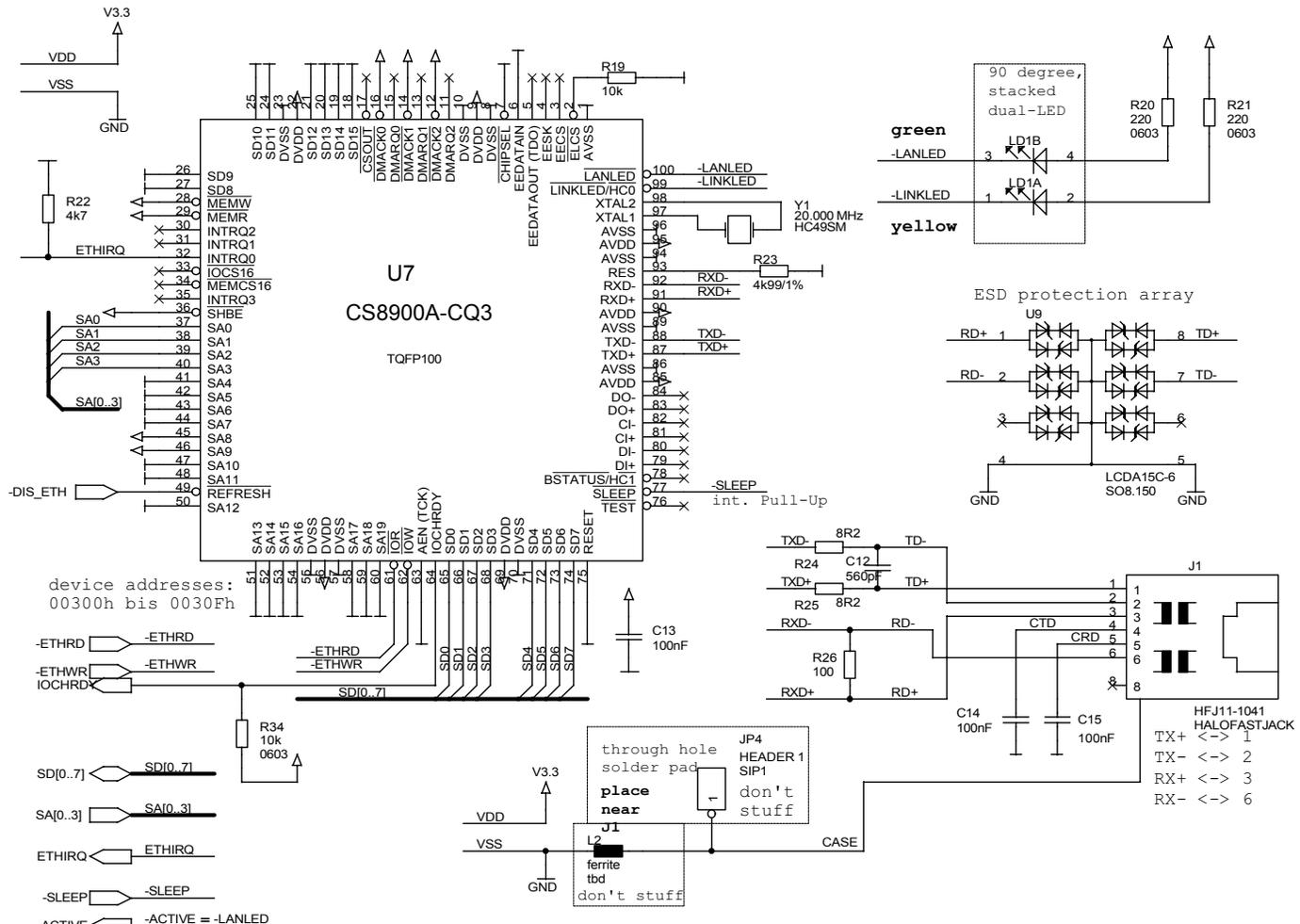
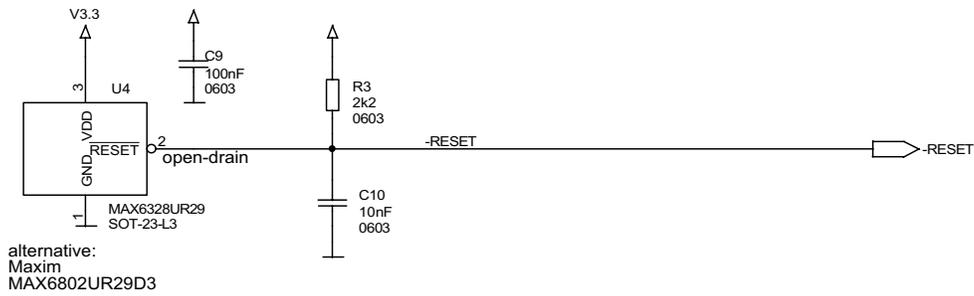


Figure 26. eZ80L92 Module Schematic Diagram, #5 of 9—E-NET Module

power supervisor



IR-transceiver

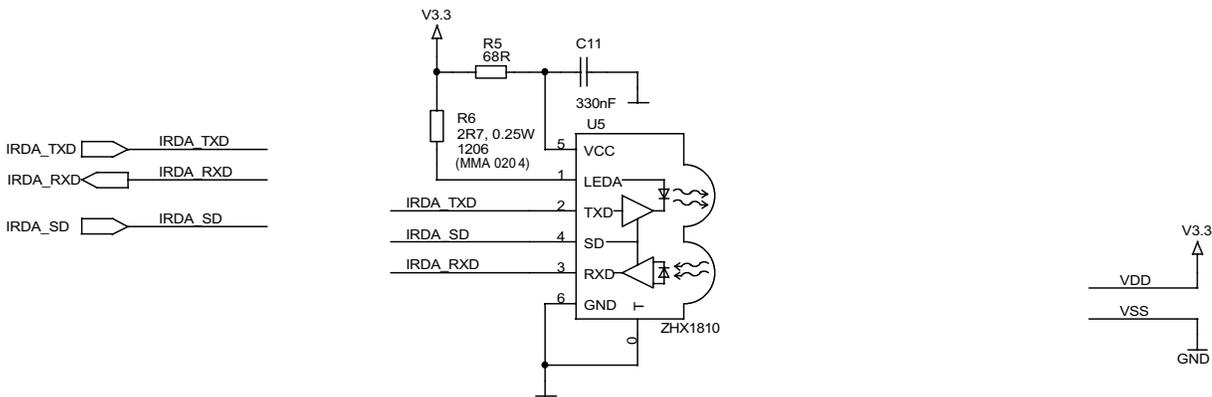


Figure 27. eZ80L92 Module Schematic Diagram, #6 of 9—IrDA Reset

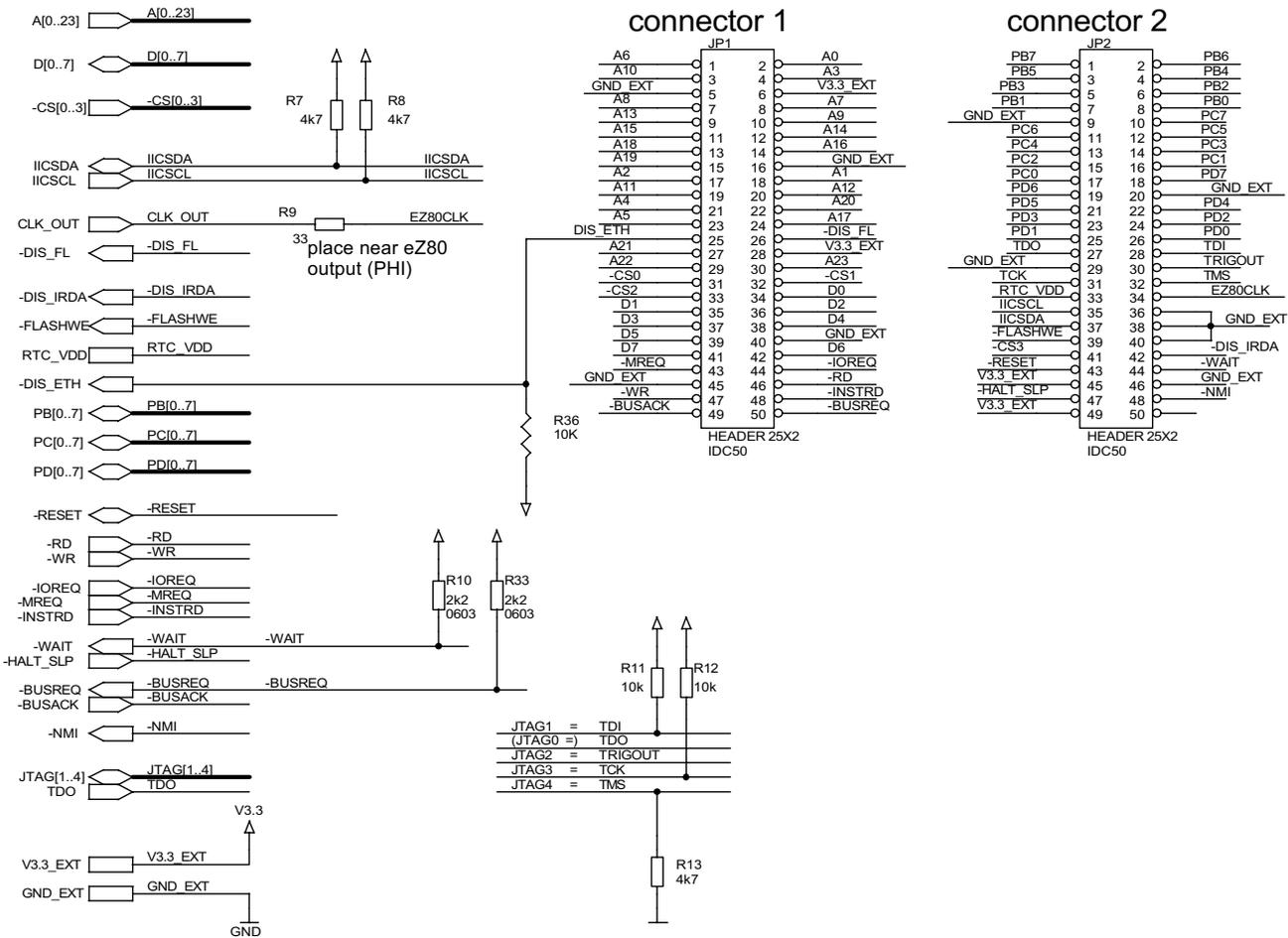
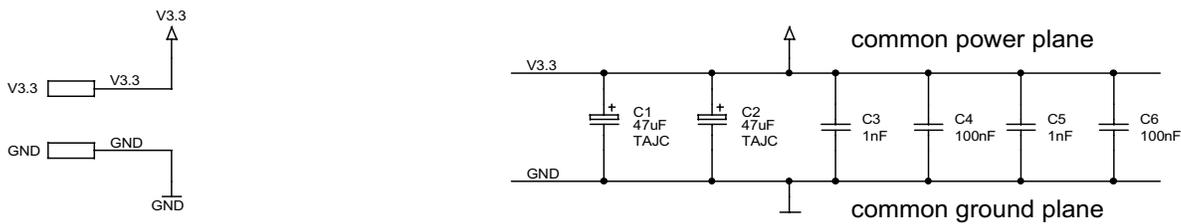


Figure 28. eZ80L92 Module Schematic Diagram, #7 of 9—Headers



no power supply on board!

Input: $VDD(=V3.3) = 3.3V \pm 5\%$

Power: $P_{max} = 1.6W$
 $P_{typ} = 0.4W$

Current: $I_{max} = 200mA$ (IrDA not in use)
 $I_{max} = 460mA$ (IrDA in use)
 $I_{typ} = 100mA$

PCB1
E-NET Module Rev.B
98Cxxxx-xxx

Figure 29. eZ80L92 Module Schematic Diagram, #8 of 9—Power Supply

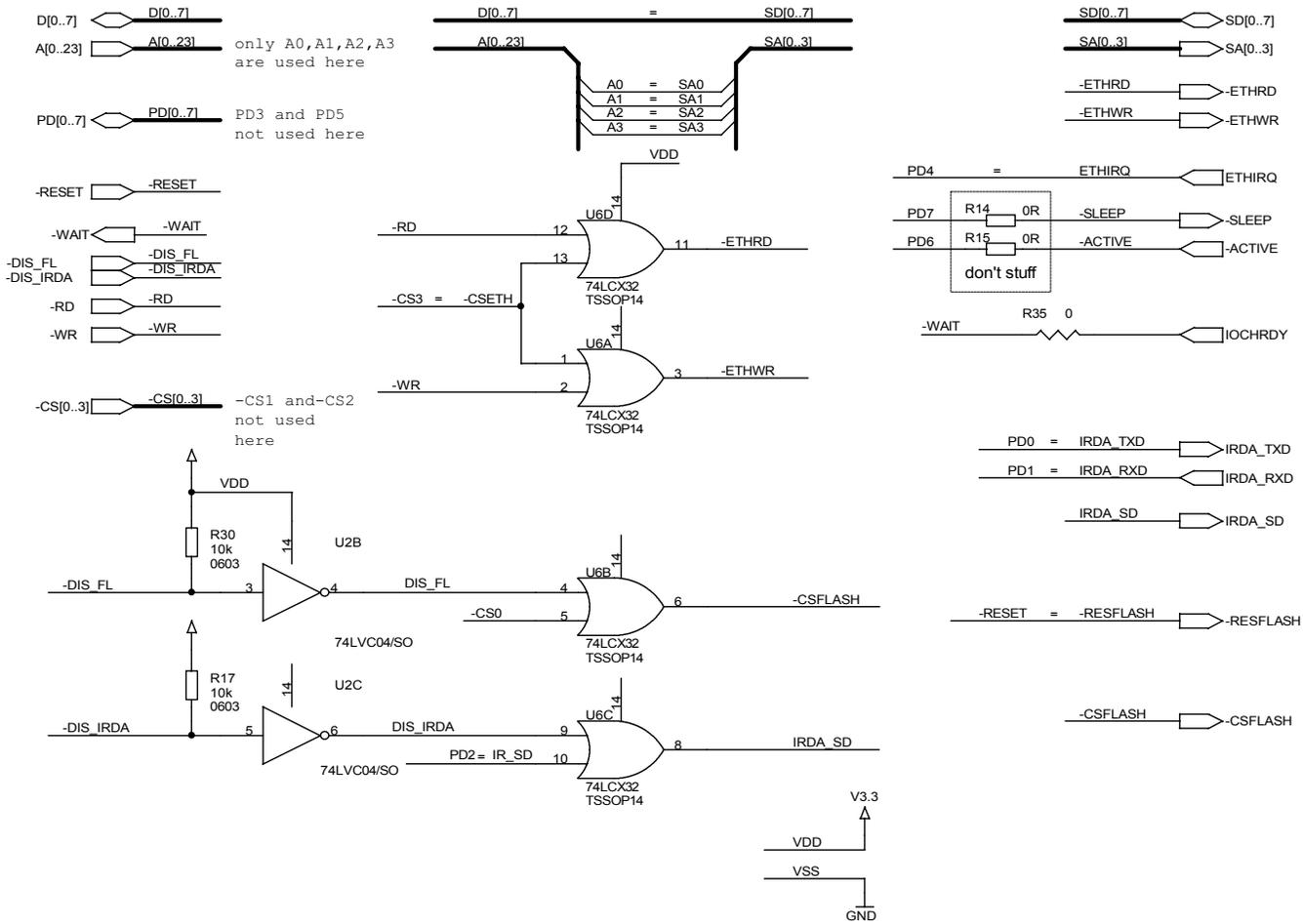


Figure 30. eZ80L92 Module Schematic Diagram, #9 of 9—Control Logic

Appendix A

General Array Logic Equations

This appendix shows the equations for disabling the Ethernet signals provided by the U10 and U15 General Array Logic (GAL) devices.

U10 Address Decoder

```
//`define      idle      2'b00
//`define      state1    2'b01
//`define      state2    2'b11
//`define      state3    2'b10
// FOR eZ80® Development Platform Rev B
// This PAL generates 4 memory chip selects

module l92_decod(
    nCS_EX,      //Enables Extension Module's Memory when Low
    nFL_DIS,     //when Low WEB Module Flash is disabled (nDIS_FL=0),
                //when High nDIS_FL depends upon state of nmemenX
    nCS0,
    A7,          //A23
    A6,          //A22
    A5,          //A21
    A4,          //A20
    A3,          //A19
    A2,          //A18
    A1,          //A17
    A0,          //A16
    nCS2,
    nEX_FL_DIS, //disables Flash on the expansion module, when Low
    nEM_EN,     //enables Development Platform LED and
                //the general-purpose port.
    nDIS_FL,    //disables E-NET Module Flash when Low
    nL_RD,      //enables local data bus to be read by CPU
    nmemen1,
```



```
nmemen2,  
nmemen3,  
nmemen4  
);
```

input

```
nFL_DIS          /* synthesis loc="P4"*/,  
nCS0             /* synthesis loc="P5"*/,  
nCS2            /* synthesis loc="P3"*/, //was 23  
A7              /* synthesis loc="P6"*/,  
A6              /* synthesis loc="P7"*/,  
A5              /* synthesis loc="P9"*/,  
A4              /* synthesis loc="P10"*/,  
A3              /* synthesis loc="P11"*/,  
A2              /* synthesis loc="P12"*/,  
A1              /* synthesis loc="P13"*/,  
A0              /* synthesis loc="P16"*/,  
nEX_FL_DIS      /* synthesis loc="P2"*/;  
//input[7:0]  A;  upper part of Address Bus of L92  
                //A23=A7,A22=A6,A21=A5,A20=A4,A19=A3  
                //A18=A2,A17=A1,A16=A0
```

output

```
nCS_EX          /* synthesis loc="P17"*/, //enables memory on the  
                //Expansion Module  
nmemen1        /* synthesis loc="P18"*/, //enables memory on the  
                //Development Platform  
nmemen2        /* synthesis loc="P19"*/,  
nmemen3        /* synthesis loc="P20"*/,  
nmemen4        /* synthesis loc="P21"*/,  
nEM_EN         /* synthesis loc="P24"*/, //enables LED and the  
                //general-purpose port.  
nDIS_FL        /* synthesis loc="P25"*/,  
nL_RD          /* synthesis loc="P23"*/  
;
```



```

wire nCS_EX,
    nmemen1,
    nmemen2,
    nmemen3,
    nmemen4;

//wire MOD_DIS =
    ((nmemen1==0) | (nmemen2==0) | (nmemen3==0) | (nmemen4==0)); //if any
    //of the signals is Low,
    //Flash on the Module will be
    //disabled if nDIS_FL is High

wire nEXP_EN = ~( (nCS0==0) & (A7==0) & (A6==1) ); //expansion module
    //Flash enabled if this is 0

//wire nDIS_FL = (nFL_DIS) ? ~nEXP_EN : ~(nFL_DIS);

wire nDIS_FL = nFL_DIS & nEXP_EN; //if either of them is 0 Flash is
    //disabled

assign nCS_EX = (nEX_FL_DIS) ? nEXP_EN : ~(nEX_FL_DIS);
assign nL_RD =
    ~( (nmemen1==0) | (nmemen2==0) | (nmemen3==0) | (nmemen4==0) | (nEM_EN==
    0) | (nCS_EX==0) );
assign nmemen4 = ~( (nCS2==0) & ( {A7,A6,A5,A4,A3} ==5'h17) );
assign nmemen3 = ~( (nCS2==0) & ( {A7,A6,A5,A4,A3} ==5'h16) );
assign nmemen2 = ~( (nCS2==0) & ( {A7,A6,A5,A4,A3} ==5'h15) );
assign nmemen1 = ~( (nCS2==0) & ( {A7,A6,A5,A4,A3} ==5'h14) );
assign nEM_EN = ~( (nCS2==0) & ( {A7,A6,A5,A4,A3,A2,A1,A0} ==8'h80) );
endmodule

```

U15 Address Decoder

```

`define      anode      8'h00
`define      cathode   8'h01
`define      latch     8'h02
// FOR eZ80® Development Platform Rev B

```



```
// This PAL generates signals that control Expansion Module
// access, LED and the general-purpose port.
// This device is a GAL22LV10-5JC (5ns tpd) or equivalent with
// Package = 28 pin PLCC
//
//
```

```
module l92_em_pal(
    nDIS_EM,
    nEM_EN,
    A0,
    A1,
    A2,
    A3,
    A4,
    A5,
    A6,
    A7,
    nRD,
    nCS,
    nWR,
    nMEMRQ,
    nIORQ,
    nEM_RD,
    nEM_WR,
    nAN_WR,
    nCT_WR,
    nDIS_ETH
);
```

```
input      nDIS_EM /* synthesis loc="P3"*/,
    nEM_EN    /* synthesis loc="P4"*/,
    A0        /* synthesis loc="P5"*/,
    A1        /* synthesis loc="P6"*/,
    A2        /* synthesis loc="P10"*/,
    A3        /* synthesis loc="P11"*/,
    A4        /* synthesis loc="P12"*/,
```



```
A5          /* synthesis loc="P13"*/,
A6          /* synthesis loc="P27"*/,
A7          /* synthesis loc="P26"*/,
nIORQ       /* synthesis loc="P2"*/,
nRD         /* synthesis loc="P7"*/,
nCS         /* synthesis loc="P25"*/, //CS3 for CS9800
nWR         /* synthesis loc="P9"*/,
nMEMRQ      /* synthesis loc="P16"*/;

output
nEM_RD      /* synthesis loc="P17"*/,
nEM_WR      /* synthesis loc="P18"*/,
nCT_WR      /* synthesis loc="P19"*/,
nAN_WR      /* synthesis loc="P20"*/,
nDIS_ETH    /* synthesis loc="P21"*/;

parameter anode=8'h00;
parameter cathode=8'h01;
parameter latch=8'h02;

wire [7:0] address={A7,A6,A5,A4,A3,A2,A1,A0};

assign nEM_WR =
  ~((nDIS_EM==1)&(nWR==0)&(nEM_EN==0)&(address==latch));
assign nEM_RD =
  ~((nDIS_EM==1)&(nRD==0)&(nEM_EN==0)&(address==latch));

assign nAN_WR =
  ~((nDIS_EM==1)&(nWR==0)&(nEM_EN==0)&(address==anode));
assign nCT_WR =
  ~((nDIS_EM==1)&(nWR==0)&(nEM_EN==0)&(address==cathode));

assign nDIS_ETH = ~(nCS);
endmodule
```





Customer Feedback Form

If you note any inaccuracies while reading this User Manual, please copy and complete this form, then mail or fax it to ZiLOG (see *Return Information*, below). We also welcome your suggestions!

eZ80L92 Development Kit

Serial # or Board Fab #/Rev. #

Software Version

Document Number

Host Computer Description/Type

Customer Information

Name	Country
------	---------

Company	Phone
---------	-------

Address	Fax
---------	-----

City/State/Zip	E-Mail
----------------	--------

Return Information

ZiLOG

System Test/Customer Support

532 Race Street

San Jose, CA 95126

Phone: (408) 558-8500

Fax: (408) 558-8536

Email: zservice@zilog.com

Problem Description or Suggestion

Provide a complete description of the problem or your suggestion. If you are reporting a specific problem, include all steps leading up to the occurrence of the problem. Attach additional pages as necessary.

