



APEX ZF

PC/104+ Single Board Computer

User Guide

Document Reference: Product User Guide

Document Issue: 1.1

Contents

Section 1 - Introduction	5
Copyright	6
Limitations of Liability	6
Trademarks	6
Regulatory Statements	7
Manual Organisation	8
Section 2 - Quick Start.....	9
APEX-ZF Development Kit	10
Standard Cables Supplied with Development Kit.....	11
Section 3 - General Precautions.....	14
PS/2 Devices.....	14
Electro-Static Discharges.....	14
Off-Board Battery	14
BIOS & CMOS Memory	15
Electromagnetic Compatibility	15
Section 4 – Product Summary	16
System Diagram	16
Hardware Description.....	17
System Control and Reset.....	17
User EEPROM	17
GPIO / Analogue Input.....	17
Watchdog.....	18
RS232/422/485 Control	18
Environment Monitoring	18
On-Board Ethernet.....	18
Mini-PCI.....	19
Compact Flash	19
IDE	19
USB	19
Battery	19
Specification	20
Section 5 - Installation	22
Connector Locations.....	22
Connector Pin Outs.....	23
P4 IDE: Board Straddled 2mm Pin Header	23
P5 Serial: Upright 2.54mm pitch SM pin header	23

P9 USB: Straddled 2.54mm pin header	24
P10 Utilities/Miscellaneous: Straddled 2mm pin header	24
P11 Ethernet: Straddled 2mm pin header	25
P12 Power: 2.54mm pitch latched Molex KK	25
Jumper Locations.....	25
CMOS Clear	25
On-Board LED's.....	26
Software Setup and Drivers	27
Section 6 – BIOS SETUP	28
Firmware Setup	28
BIOS Setup	28
Main Setup page	29
Advanced Setup Page	31
Advanced Chipset Control.....	32
I/O Device Configuration	33
PCI Configuration.....	34
Console Redirection	36
Security Setup Page	37
Power Setup Page	38
Boot Setup Page.....	39
Exit Setup Page.....	40
Supervisory Microprocessor Setup – HSETUP.....	41
Appendix A – BIOS Int 50h Functions:	42
Appendix B – SMBUS Commands	52
SMBUS Slave Address.....	52
Watchdog functions	52
EnableWatchdog.....	52
DisableWatchdog.....	52
RefreshWatchdog	52
SetWatchdogResponse	53
EEPROM Functions	53
SetEepromPointer	53
WriteEepromByte	53
ReadEepromByte	54
EraseEepromByte	54
EraseEeprom.....	54
GPIO / ADC Functions.....	54
SetGpioPinFunction.....	54

SetGpioBitDirection	55
SetGpioPullups	55
WriteGpioPort	55
ReadGpioPort	56
SetGpioBit	56
GetGpioBit	56
GetAnalogueInput	56
System Monitor Functions.....	57
ReadSystemMonitor	57
SetCpuTempLimit	57
Miscellaneous Functions	58
GetFirmwareVersion	58
Reset Functions.....	58
ResetMicroProcessor	58
ResetSystem	58
GetIrqSource.....	59
PowerOffSupplies – Reserved – Do not use.....	59
Appendix C: SMBUS Command Codes.....	60
Maintenance	61
Amendment History	62



APEX-ZF

Section 1 - Introduction

The APEX-ZF single board computer is the latest in a line of successful PC/104+ single board computers designed and manufactured in the United Kingdom by Blue Chip Technology. The APEX-ZF complies with the PC/104+ standard form factor providing PCI, and ISA bus interfaces from a single card.

The PC/104+ is an ideal platform for the increasing low power requirements for embedded applications. The design is based on the ZF Micro ZFx86 system-on-chip running at internal clock rates of 33MHz, 66MHz or 100MHz providing 486 level performance. The memory sub-system is on-board and consists of 64MB SDRAM.

The ZFx86 system-on-chip, from ZF Micro Solutions, provides unique integration with support for 16 bit ISA interface, 32 bit PCI 2.1 interface, ATA-4 compatible devices, PS/2 Keyboard and Mouse, two Serial Ports, one Parallel Port and two USB ports.

Networking is supported via a Davicom DM9102D PCI Ethernet controller providing 10/100baseT networking. The APEX-ZF Single board computer provides support for Type 1 Compact Flash via a socket on the base of the PCB, as well as an optional Type IIIA MiniPCI connector on the top of the PCB. The MiniPCI socket provides further expansion capability via suitable graphical or I/O plug in modules.

Furthermore, the APEX-ZF platform incorporates a Supervisory microprocessor to perform a number of control and monitoring functions. These functions include control of the ZFx86 strapping options, processor reset, RS232/485 control, voltage and temperature monitoring, user EEPROM, GPIO and watchdog.

The APEX-ZF Single board computer can be self powered for standalone operation, or can be powered from a PC/104+ backplane to provide additional I/O capability.

Further expansion can also be achieved via the use of additional 3rd party PC/104+ modules which can make use of the Bottom Entry and optional Pass through PC/104 and PC/104+ connectors.

Copyright

All rights reserved. No part of this publication may be reproduced, stored in any retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopied, recorded or otherwise, without the prior permission, in writing, from the publisher. For permission in the UK please contact Blue Chip Technology.

Information offered in this manual is believed to be correct at the time of printing. Blue Chip Technology accepts no responsibility for any inaccuracies. The information contained herein is subject to change without notice. There are no express or implied licences granted herein to any intellectual property rights of Blue Chip Technology Ltd.

Limitations of Liability

In no event shall Blue Chip Technology be held liable for any loss, expenses or damages of any kind whatsoever, whether direct, indirect, incidental or consequential, arising from the design or use of this product or the support materials supplied with this product. If this product proves to be defective, Blue Chip Technology is only obliged to replace or refund the purchase price at Blue Chip Technology's discretion according to their Terms and Conditions of Sale.

Trademarks

All trademarks and registered names acknowledged.

IBM, PC, AT and PS/2 are trademarks of International Business Machines Corporation (IBM).

MSDOS and WINDOWS are registered trademarks of the Microsoft Corporation.

Regulatory Statements

CE

This product meets the essential protection requirements of the European EMC Directive (89/336/EEC) and its amending Directives, and the Low Voltage Directive 73/23/EEC, and is eligible to bear the CE mark.

Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

FCC

NOTE:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

WARNING:

Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

Manual Organisation

This manual describes in detail the Product range of the APEX-ZF Single Board Computer.

We have tried to include as much information as possible but we have not duplicated information that is provided in the standard IBM Technical References, unless it proved to be necessary to aid in the understanding of the product.

The manual is sectioned as follows:

1. Introduction;
2. Quick Start Guide
3. General Precautions
4. Product Specifications
5. Installation, including Layout, connector details and what software to install
6. Bios Setup
7. Appendix A: Int 50 Routines
8. Appendix B: SMBUS Commands
9. Appendix C: SMBUS Command Codes
10. Maintenance details

We strongly recommend that you study this manual carefully before attempting to interface with the APEX-ZF single board computer (SBC) or change the standard configurations. Whilst all the necessary information is available in this manual we would recommend that unless you are confident, you contact your supplier for guidance. ***IT IS PARTICULARLY IMPORTANT THAT YOU READ THE SECTION 'GENERAL PRECAUTIONS' BEFORE HANDLING ANY COMPONENTS INSIDE THE UNIT.***

If you have any suggestions or find any errors concerning this manual and want to inform us of these, please contact our Technical Services department with the relevant details.

Section 2 - Quick Start

The following sections explain how to install the APEX- ZF Single Board Computer.

First ensure that you are familiar with the contents of the section "[General Precautions](#)". It contains important information to avoid damage to the board.

If Compact Flash or MiniPCI modules are to be fitted, then install these first.

The PC/104+ specification allows for a maximum Stack of 4 PC/104+ items, so if the APEX-ZF is to be used in conjunction with other PC/104+ carrier boards or modules, then check all PC/104 and PC/104+ pins are straight and have not been bent or crushed during transport or storage. Straighten any pins which are off centre before carefully connecting each item together.

Connectors for Utility, IDE, USB and Ethernet straddle the edge of the APEX-ZF PCB. Before connecting any header, ensure that all pins are straight. Care should be taken to ensure the correct orientation of any header, especially if unkeyed connectors are being used. For instance if the IDE connector is misaligned, then it is possible to damage any IDE device attached

The APEX-ZF requires +5V for operation. As soon as power is applied, the APEX-ZF will start. So ensure that all required connections are in place before applying power.

APEX-ZF Development Kit

A Development Kit is available in order to help customers develop their own application based on the APEX-ZF SBC.

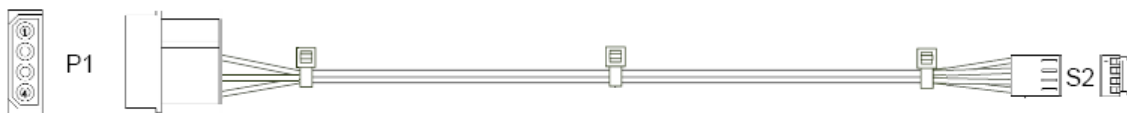
As well as the APEX-ZF SBC, the kit contains cables, peripherals, PSU and OS as well as a Windows CE 6.0 Development kit. The APEX-ZF Development Kit can be tailored to suit most requirements, with the Standard and main optional component listed below

Item	Standard / Option
APEX-ZF	Standard
Power Cable	
Ethernet Cable	
Utilities Cable	
USB Cable	
IDE Cable	
Serial Coms Cable	
Keyboard/Mouse Splitter	
Compact Flash	Optional
Operating System	Optional
DOS	
Windows CE 6.0	
Windows CE 6.0 Development Kit	Optional
Power Supply	Optional
65W power Brick	
300W ATX	
Backplane	Optional
miniPCI Adapter	Optional
Wireless Lan	
Multiple COM ports	
VGA	
LCD (640x480) (including cable)	Optional
Touch screen (640x480) (including cable)	Optional

Standard Cables Supplied with Development Kit

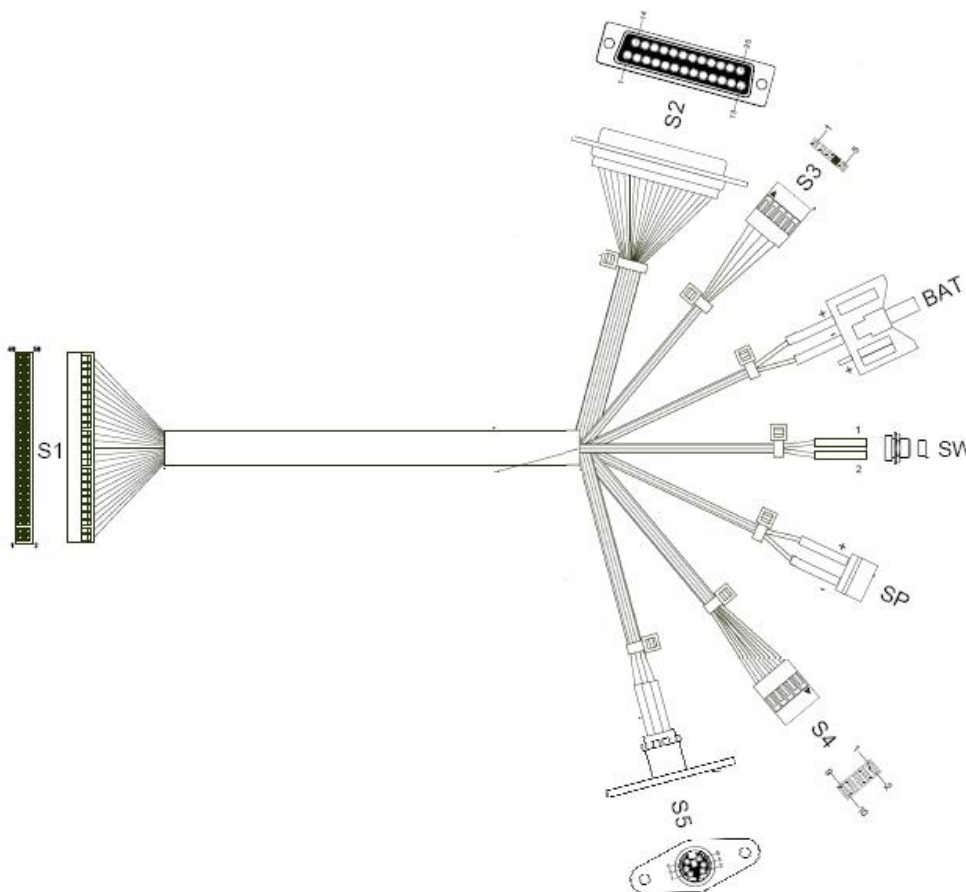
The following cables are supplied with the Development kit

1. Power



This cable allows attachment to a standard Molex Power connector of the type that powers a 5¼ optical device or HDD. The other end attaches to connector P12 on the APEX-ZF SBC

2. Utility



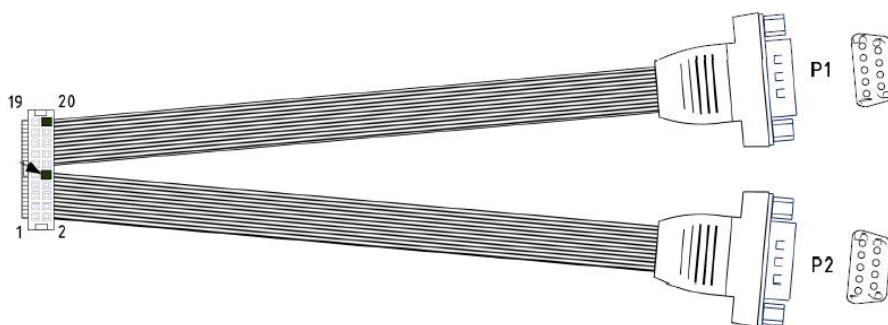
This connector attaches to P10 on the SBC and provides suitable connectors for Parallel Port, Keyboard, Mouse, GPIO and Infra Red. The cable also has a Battery holder, Reset Switch and Speaker.

3. Ethernet



This cable connects to P11 with the other end connected to a standard RJ45 socket

4. Serial



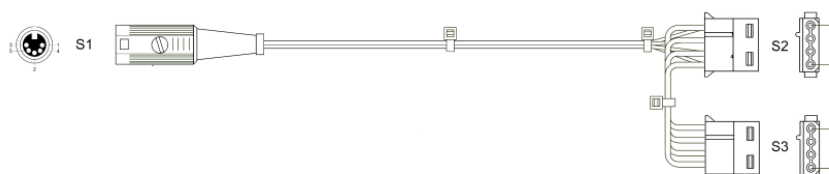
This cable connects to P5 and is terminated in two standard 9way serial connectors

5. IDE

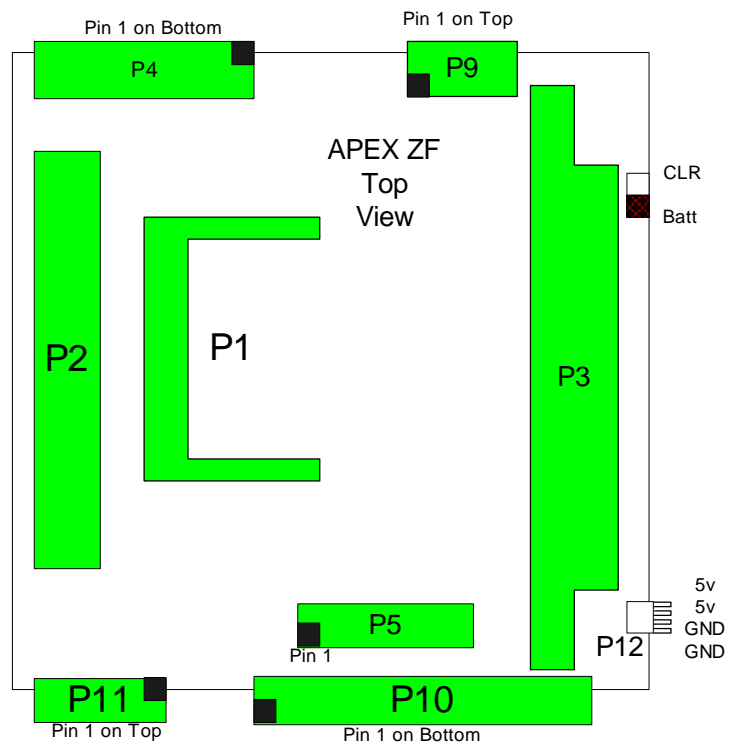


This cable connects to P4 and allows connection to two standard IDE devices

6. Power Connector 2 (optional)



If the optional power brick is selected, then this cable is supplied. One end connects to the Power Supply and is terminated in two standard 5 1/4 Molex power connectors. If more than two power leads are required, these can be achieved by the addition of standard RB-511 power splitters



Section 3 - General Precautions

Your SBC is susceptible to damage by electrostatic discharges. In order to avoid damage, you should work at an anti-static bench and observe normal anti-static precautions. Wear an anti-static wrist strap connected to an earth point *before* opening any packaging.

Where a wrist strap is not available, discharge any static charge you may have built-up by touching an earth point. Avoid any further movement that could build up another static charge. Touch an earth point from time to time to avoid further build-up, and remove the items from their anti-static bags only when required

PS/2 Devices

It is important that PS/2 devices (mouse and keyboard) are not connected or disconnected with the unit powered on. Damage or data corruption may occur if this precaution is not observed.

Electro-Static Discharges

If you are going to open up the unit, it is important to realise that the devices on the cards within this unit can be damaged by static electricity. Bear in mind that the damage caused by static electricity may vary from total destruction to partial damage, which may not be immediately obvious. This could have an effect on the product's reliability and warranty. Before opening the chassis, ensure that you take necessary static precautions. Ideally you should work at an anti-static bench and wear an approved wrist strap or if that is not possible, touch a suitable ground to discharge any static build up before touching the electronics. This should be repeated if the handling continues for any length of time.

If it is necessary to remove a board or electronic assembly, place it into an anti-static bag. This will prevent any static electricity build up damaging the board. Metallised bags are preferred. Do not use black anti-static bags for any item containing a battery because these tend to be conductive and will discharge the battery.

Off-Board Battery

The APEX-ZF supports an off-board battery connected via the Utilities Header P10.

Great care should be taken with any type of battery. Under NO circumstances should:

- the outputs be shorted
- be exposed to temperatures in excess of 100°C
- be burnt
- be immersed in water
- be unsoldered
- be recharged
- be disassembled

If the battery is mistreated in any way there is a very real possibility of fire, explosion, and harm.

BIOS & CMOS Memory

Please be aware that with personal computer products, it is possible to create configurations within the CMOS memory that make booting impossible. If this should happen, clear the CMOS settings; (see the description of the Jumper Settings for details).

Electromagnetic Compatibility

This product has been designed to meet the requirements of the European EMC Directive (89/336/EEC) and is eligible to bear the CE mark.

However, because the board can be installed in a variety of computers, certain conditions have to be applied to ensure that the compatibility is maintained. Subject to those conditions, it meets the requirements for an industrial environment (Class A product).

- The board must be installed in a computer system chassis that provides screening suitable for an industrial environment.
- Any recommendations made by the computer system manufacturer/supplier must be complied with regarding earthing and the installation of boards.
- Any metal back plate must be securely screwed to the chassis of the computer to ensure good metal-to-metal (i.e. earth) contact.
- Connector bodies must be securely connected to the enclosure.
- The external cabling to boards causes most EMC problems. It is imperative that any external cabling to the board is totally screened, and that the screen of the cable connects to the metal end bracket of the board or the enclosure and hence to earth. It is recommended that round, screened cables with a braided wire screen are used in preference to those with a foil screen and drain wire. Use metal connector shells that connect around the full circumference of the cable screen: they are far superior to those that earth the screen by a simple “pig-tail”.
- The keyboard and mouse will play an important part in the compatibility of the processor card since they are ports into the board. Similarly, they will affect the compatibility of the complete system. Fully compatible peripherals must be used otherwise the complete system could be degraded. They may radiate or behave as if keys/buttons are pressed when subject to interference. Under these circumstances it may be beneficial to add a ferrite clamp on the leads as close as possible to the connector. A suitable type is the Chomerics type H8FE-1004-AS.
- USB cables should be high quality screened types.
- Ensure that the screens of any external cables are bonded to a good RF earth at the remote end of the cable.

Failure to observe these recommendations may invalidate the EMC compliance.

Section 4 – Product Summary

System Diagram

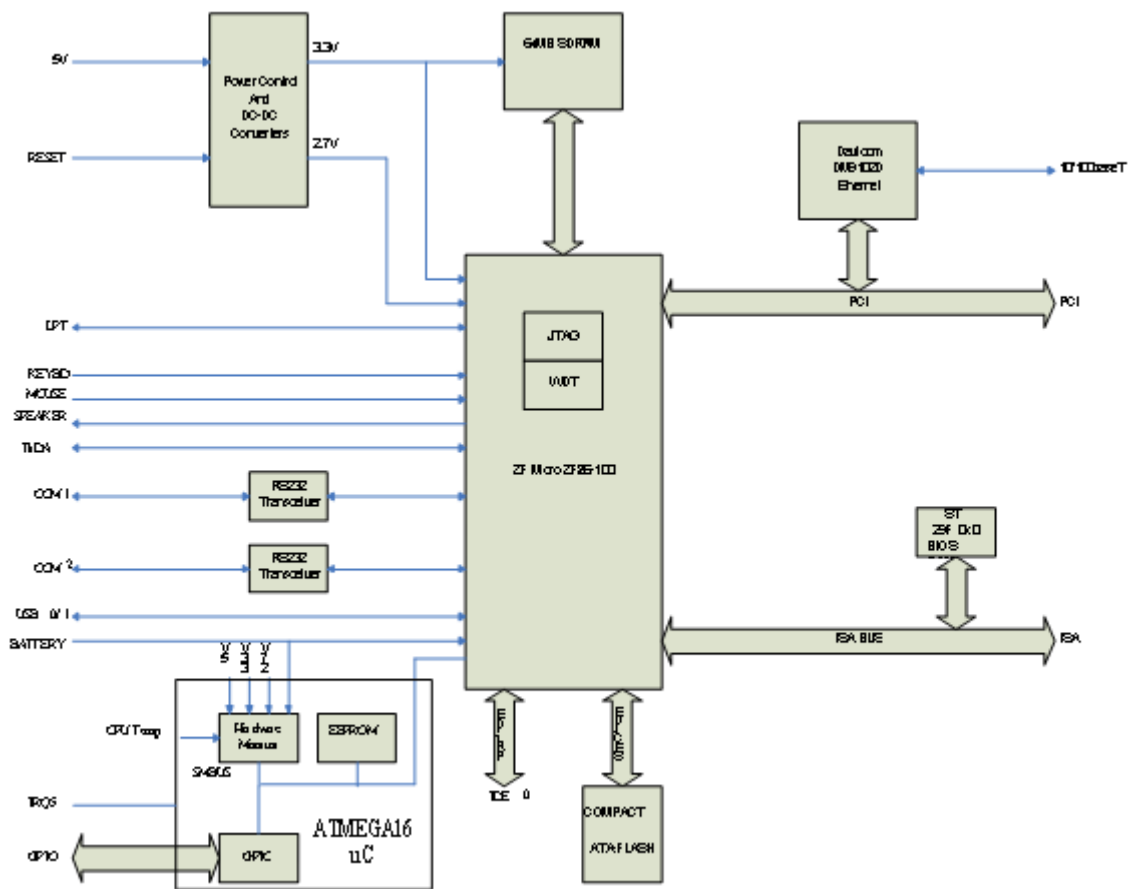


Figure 1: System Block Diagram

Hardware Description

System Control and Reset

The Supervisory microprocessor is responsible for some of the configuration options of the ZFx86 processor. These options include

- Monitoring of the reset button input. This signal is debounced for 250ms before the reset of the ZFx86 processor is triggered.
- User EEPROM
- GPIO
- Watchdog
- RS232/422/485 control
- Processor Speed
- Ethernet Enable / Disable
- Voltage monitoring
- Temperature monitoring

User EEPROM

The Supervisory microprocessor being used contains 512 bytes of EEPROM. Of this 500 bytes is available for use as non-volatile storage. The remaining 12 bytes is used by the microprocessor firmware to store its own internal configuration information.

Access to the User EEPROM in Real Mode is via INT 50h BIOS functions, as outlined in Appendix A and in Protected Mode via SMBUS commands as outlined in Appendices B and C.

GPIO / Analogue Input

The Supervisory microprocessor provides 8 digital general purpose input / output pins (GPIO) (four shared with the analogue input pins) that are configurable under application control as either input or output pins. .

An output pin from the microprocessor is connected to an IRQ on the ZFx86 and can be used for change of state notification.

The Supervisory microprocessor provides 4 analogue inputs capable of reading a voltage between 0 and 2.56V which are shared with GPIO pins 3...0. These voltages are sampled using a 10bit analogue to digital converter. The SMBUS function returns the raw value read by the A/D converter and the API function converts this value to a real world voltage.

Access to the GPIO in Real Mode is via INT 50h BIOS functions, as outlined in Appendix A and in Protected Mode via SMBUS commands as outlines in Appendices B and C. Connection to the GPIO lines is via the [Utilities connector P10](#).

Watchdog

The watchdog countdown timer is programmable to have a timeout from 1 second to 255 seconds with 1 second increments.

If a watchdog refresh command is received then the timeout reloads to the value specified when the watchdog was enabled. If the timer reaches 0 then the Supervisory microprocessor can trigger a reset of the ZF_x86 processor and associated peripherals or alternatively can issue an interrupt request.

Access to the Watchdog in Real Mode is via INT 50h BIOS functions, as outlined in [Appendix A](#) and in Protected Mode via SMBUS commands as outlined in [Appendices B](#) and [C](#).

RS232/422/485 Control

The second serial port COM2 can be configured to be either RS232, RS422/RS485 Full Duplex or RS485 Half Duplex. An SMBUS function allows the user through the extended BIOS set-up to set the port into the required mode. This function is handled via the DOS [Hsetup](#) routine supplied with the board

Note: if RS485 Half Duplex mode is being used with a 2 wire cable, then the TX and RX lines need to be connected together at each end of the cable as follows

Tx+/Rx+ connect to Tx+/Rx+

Tx-/Rx- connect to Tx-/Rx-

Access to both serial ports is via [connector P5](#).

Either port can be set to output text to a remote console. This is setup via the [Console Redirection](#) section under the [Advanced menu in BIOS](#).

Environment Monitoring

Four 10bit A/D channels are dedicated to system monitoring functions. These channels measure the CPU core voltage, 3.3V supply, 5V supply and the system temperature via a thermistor located on the APEX-ZF board.

On-Board Ethernet

The microprocessor provides the ability to enable or disable the on-board Ethernet controller.

Access to the Ethernet is via [connector P11](#)

Mini-PCI

The APEX-ZF board has an optional Mini-PCI connector. When fitted this conforms to Type IIIA and supports 124-pin mini-PCI adapters of dimensions 2.4 x 59.6 x 50.95mm

Compact Flash

The APEX-ZF board has a dedicated Type I Compact Flash socket on the base of the board. This socket supports only Type I Compact Flash media.

The Compact Flash device is fixed as the IDE Secondary Master device.

IDE

As well as supporting the Compact Flash as a Secondary IDE device, the APEX-ZF also supports two standard IDE devices as Primary Master and Primary Slave.

Connection is via [Connector P4](#)

USB

The APEX-ZF supports USB Legacy Devices such as Keyboard and Mouse.

Connection is via [connector P9](#)

Battery

The APEX-ZF does not have an on-board battery to save CMOS settings. If this is required, then an off-board battery can be attached via the [Utilities connector P10](#)

The on-board circuitry has been designed to support a CR2032 button Cell battery. To avoid damage to your APEX-ZF you are strongly advised not to use any other battery without first checking with Technical Support at support@bluechiptechnology.co.uk

Specification

APEX-ZF Power Requirements	+5V \pm 5% +12V \pm 5% -5V \pm 5% -12V \pm 5%	Required for processor operation † These are not required for board operation } The PC 104-plus and PC/104+ voltage rails are ‡ linked on board
Typical System Power Consumption	33Mhz 66Mhz 100Mhz	† 2.25W DOS without graphics support ‡ 3.8 W Widows CE plus miniPCI Graphics † 2.65W DOS without graphics support ‡ 4.15 W Widows CE plus miniPCI Graphics † 2.55W DOS without graphics support ‡ 4.0 W Widows CE plus miniPCI Graphics
GPIO	Source/Sink	20mA max (10mA recommended) per line
Temperature	Non-operating Operating	TBA TBA
Shock	Non-operating	Half sine, 2ms, 1 m drop
Vibration	Non-operating Operating	5 Hz - 500 Hz, 3.1 g RMS random 10 Hz - 55 Hz, at 2mm displacement 55Hz to 150Hz at 2g peak acceleration
EMC	Emissions Susceptibility	EN55022 Class A EN55024
Safety		EN60950 UL60950 Third Edition
MTBF	Estimated	130,000 hrs at 40°C ground-benign environment to Mil-Hdbk-217F notice 2.
Dimensions	Board only	Refer to Figure 2

Dimensions are in inches / (millimeters)

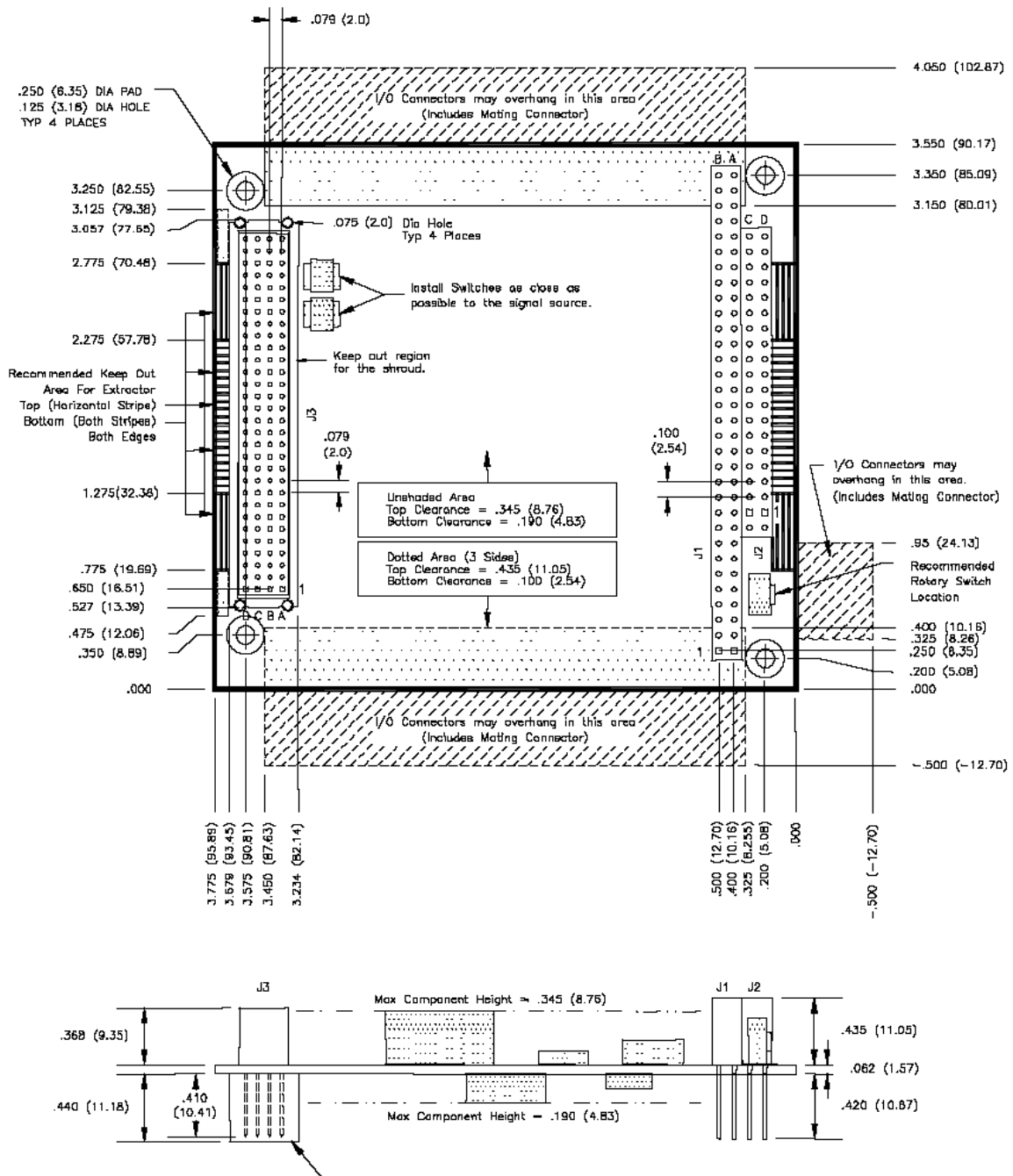


Figure 2: Board Dimensions

Section 5 - Installation

Connector Locations

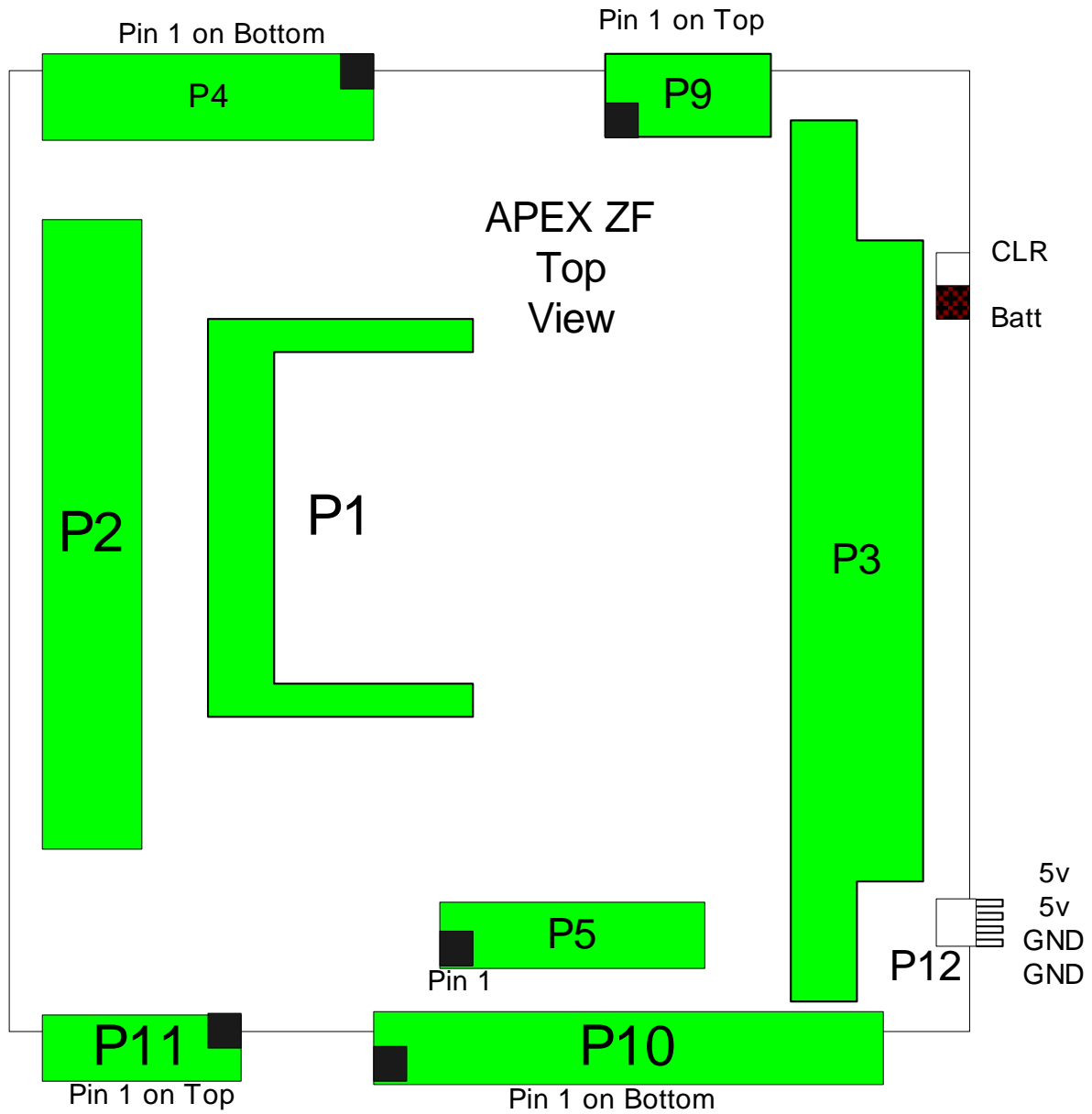


Figure 3: Connector Locations

Connector	Description	Connector	Description
P1	Type IIIA miniPCI	P2	PC/104+
P3	PC/104	P4	IDE: Straddled 2mm pin header
P5	Serial: Upright 2.54mm pin header	P9	USB: Straddled 2.54mm pin header
P10	Utils/Misc: Straddled 2mm pin header	P11	Ethernet: Straddled 2mm pin header
P12	Power: 2.54mm latched Molex KK header	CF1	Type 1 Compact Flash (on bottom)

Connector Pin Outs

P4 IDE: Board Straddled 2mm Pin Header

Description	Pin	Pin	Description
RESET#	1	2	0V
D7	3	4	D8
D6	5	6	D9
D5	7	8	D10
D4	9	10	D11
D3	11	12	D12
D2	13	14	D13
D1	15	16	D14
D0	17	18	D15
0V	19	20	- No Pin
REQ	21	22	0V
IOW#	23	24	0V
IOR#	25	26	0V
RDY	27	28	CSEL#
ACK#	29	30	0V
IRQ	31	32	IO16#
A1	33	34	NC
A0	35	36	A2
CS0#	37	38	CS1#
LED#	39	40	0V
5V	41	42	5V
0V	43	44	0V

P5 Serial: Upright 2.54mm pitch SM pin header

Description	Pin	Pin	Description
DCD1	1	2	DSR1
RX1	3	4	RTS1
TX1	5	6	CTS1
DTR1	7	8	RI1
0V	9	10	- NC
DCD2/RX2+	11	12	DSR2/TERM+
RX2/RX2-	13	14	RTS2
TX2/TRX2+	15	16	CTS2
DTR2/TRX2-	17	18	RI2/TERM-
0V	19	20	- NC

P9 USB: Straddled 2.54mm pin header

Description	Pin	Pin	Description
5V	1	2	5V
D0-	3	4	D1-
D0+	5	6	D1+
0V	7	8	0V
- No Pin	9	10	0V

P10 Utilities/Miscellaneous: Straddled 2mm pin header

This header provides connection for the following: Keyboard, Mouse, Parallel Port, Battery, Power Reset, GPIO and IR

Description	Pin	Pin	Description
P-STROBE#	1	2	P-AFD#
P-D0	3	4	P-ERROR#
P-D1	5	6	P-INIT#
P-D2	7	8	P-SLIN#
P-D3	9	10	0V
P-D4	11	12	0V
P-D5	13	14	0V
P-D6	15	16	0V
P-D7	17	18	0V
P-ACK#	19	20	0V
P-BUSY#	21	22	0V
P-PE	23	24	P-SLCT
+5V	25	26	IR-RX
IR-TX	27	28	0V
BATT+	29	30	0V
RESET#	31	32	0V
SPKR	33	34	SPKR#
0V	35	36	0V
GP0	37	38	GP4
GP1	39	40	GP5
GP2	41	42	GP6
GP3	43	44	GP7
MOUSE-D	45	46	KEYBOARD-D
MOUSE-C	47	48	KEYBOARD-C
MS/KBD 5V	49	50	0V

P11 Ethernet: Straddled 2mm pin header

Description	Pin	Pin	Description
T+	1	2	T-
R+	3	4	CT0
CT0	5	6	R-
CT1	7	8	CT1
NC	9	10	NC

P12 Power: 2.54mm pitch latched Molex KK

Description	Pin
5V	1
5V	2
0V	3
0V	4

Pin Outs for the PC/104 and PC/104+ connectors please refer to the Blue Chip Technology Standard Pin Out Document

Jumper Locations

CMOS Clear

The APEX-ZF single board computer has only one Jumper located on the top side of the PCB to the right of connector P3 as shown in Figure X on the previous page. If a battery has been connected via the Utilities Header, then this jumper allows the CMOS settings to be reset to default

Position	Function
BAT	Normal Operation
CLR	Clears CMOS

On-Board LED's

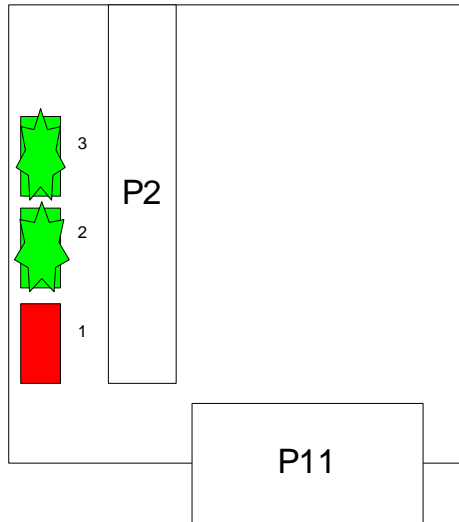


Figure 4: On-board LED's

There are 3 on-board LED's. The following table describes their functions

LED	Colour	Action	OFF	ON	FLASHING
1	Red	Network Speed	10 Mb/S	100 Mb/S	Not Applicable
2	Green	Network Connect	Not connected	Cable Connected	Network Activity
3	Green	IDE Activity	No Activity	IDE Activity	IDE Activity

Software Setup and Drivers

The APEX-ZF Single Board Computer can be supplied with the following Operating Systems pre-installed

- Datalight ROM-DOS
 - One image with TCP/IP sockets
 - One image without TCP/IP sockets
- Windows CE 6.0

Any unique Drivers used in the above images are supplied on the Support CD as well as available for download from the Blue Chip Technology website at www.bluechiptechnology.co.uk.

Also available is the [Hsetup](#) utility for configuring the microcontroller, and System Health Monitor Utilities which monitor the CPU temperature and System Voltages. These Utilities make use of Int50 functions for Real Mode operation, such as used in DOS, and SMBUS commands for Protected Mode operation, such as used in Windows CE.

The Int50 functions and SMBUS commands also allow communication with Watchdog, GPIO and EEPROM.

Int50 functions are described in [Appendix A](#) while SMBUS Commands are described in [Appendix B](#) and [Appendix C](#).

Section 6 – BIOS SETUP

Firmware Setup

The Firmware setup options for the APEX-ZF single board computer are split between the Phoenix [BIOS Setup](#), accessed by pressing <F2> during POST, and the microprocessor setup, accessed through a DOS utility [HSETUP](#) provided with the board.

BIOS Setup

To enter the BIOS setup Pages, press <F2> during POST.

Navigation through the different menus is via the Arrow Keys as explained in the bottom panel. Up/Down arrows navigate through the items in a particular menu, and Left/Right arrows navigate between Menu's.

If entering a sub-menu, <esc> will take you back up to the previous level.

Pressing <F9> at any time sets the BIOS to its Default settings, and pressing <F10> at any time will prompt to Save the settings prior to exit. The APEX-ZF will then reboot and load the desired settings.

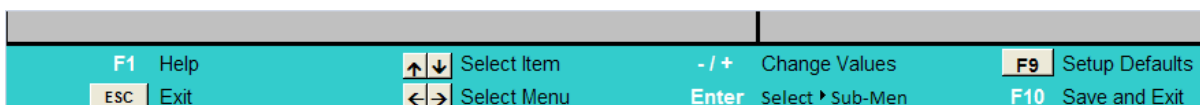


Figure A: Navigation Keys

The following section provides the options available under the different menu's.

The APEX-ZF supports an off-board battery for storing CMOS settings. If a battery is not fitted, then any BIOS settings changed will be lost when power is removed

Main Setup page

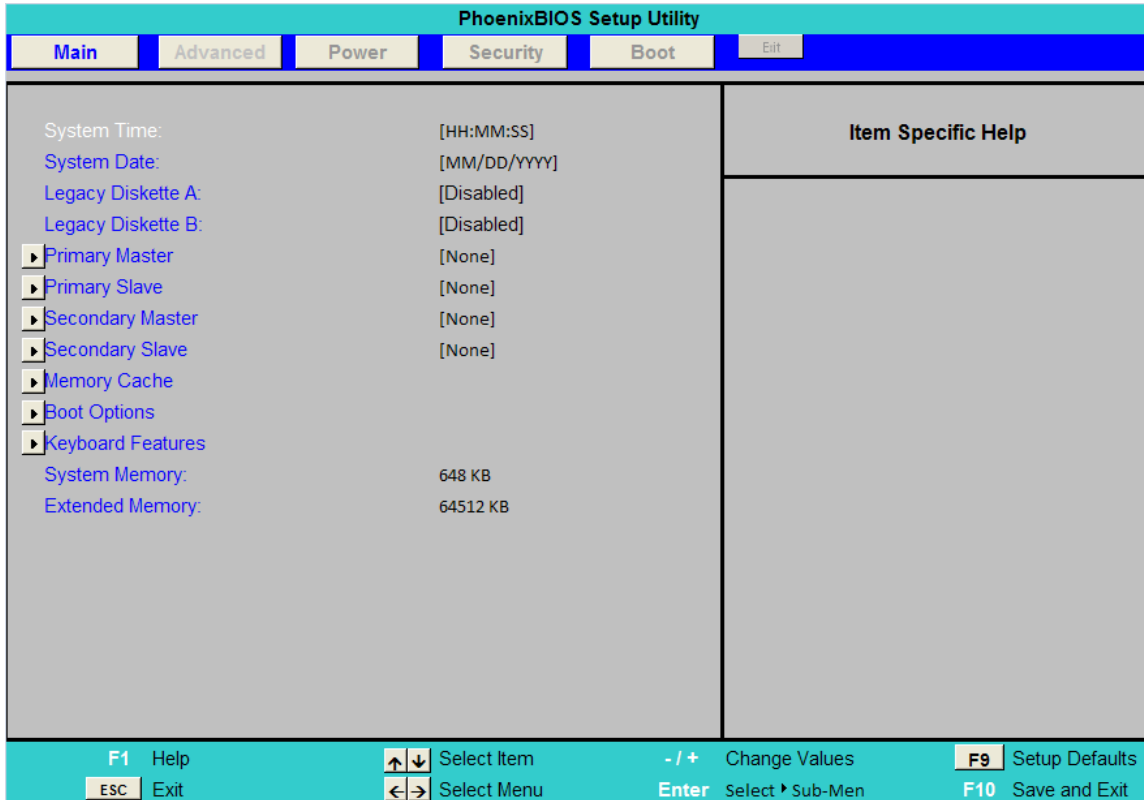


Figure B: Main Setup Page

Figure A shows the options available under the Main section with their default values.

Memory Cache:

Options are Enabled / Disabled.

Boot Options:

On this sub-menu, there are 3 choices

Summary Screen: Options are Enabled/Disabled. If enabled, the system configuration is displayed on boot

Floppy Check: Options are Enabled/Disabled. Enabled verifies floppy type on Boot, Disabled speeds boot

Hard Disk Pre-delay: Options Enabled/Disabled. Some hard disks hang if they are accessed before they initialise themselves. If this happens then setting this option to Enabled, adds a delay before the first access of the hard disk by BIOS

Keyboard Features:

The following four settings can be used

NumLock: selects the power on state of the NumLock key. Default is ON

Key Click: enables Key click. Default is Disabled

Keyboard auto-repeat rate: selects the key repeat rate. Default is 30/sec

Keyboard auto-repeat delay: selects delay before key repeat. Default is ½ sec

Primary Master/Slave and Secondary Master/Slave are set to Auto detect attached devices. This feature can be over changed by navigating to the required device and pressing <Enter> to enter the sub-menu.

The default setting page will be displayed as in Figure C below. Navigate to the Type field, and press the <+> or <-> keys to change the setting.

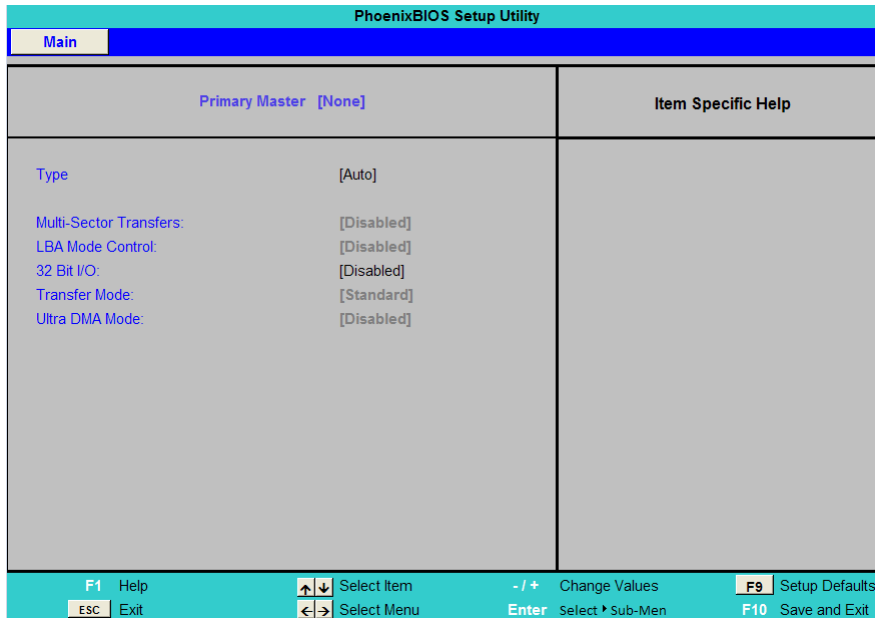


Figure C: IDE Settings Sub-menu

Options available are:

- Auto
- None
- CD-ROM
- User

Advanced Setup Page

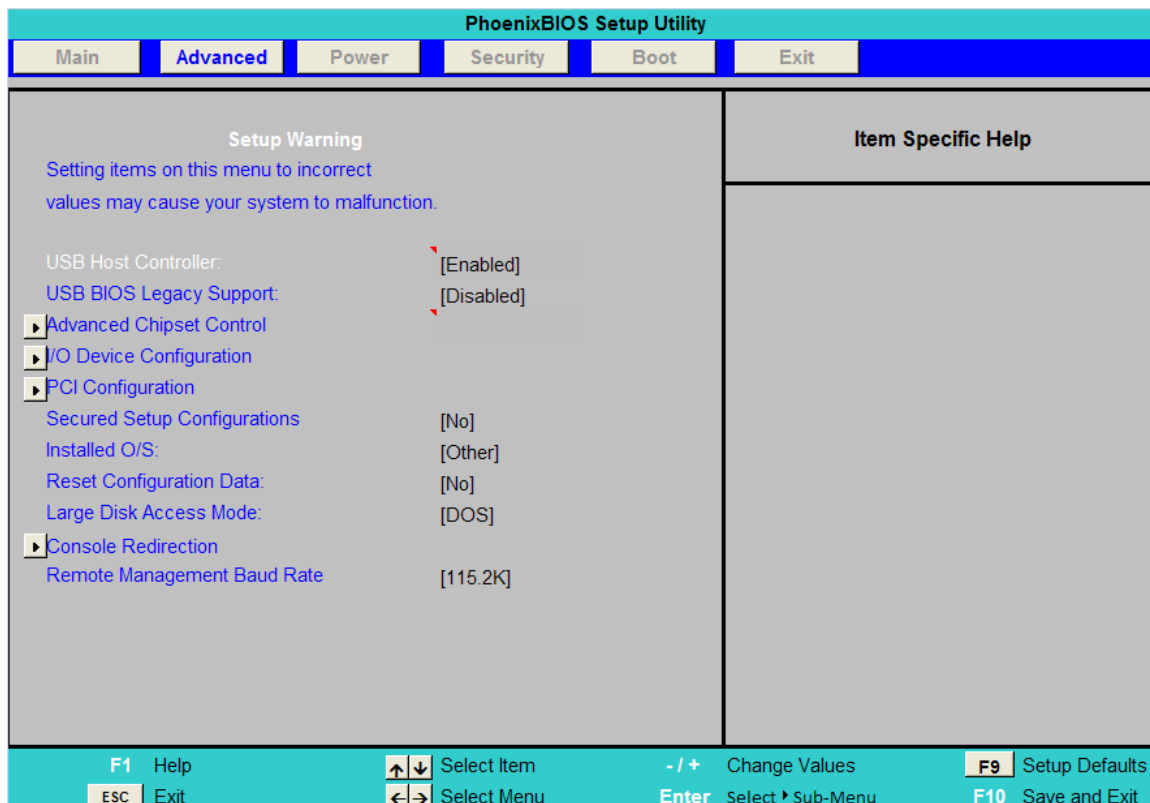


Figure D: Advanced Menu

The following options do not have a sub-menu

- USB Host Controller:** default is Enabled. Set to Disabled if USB is not required, and to free the resources for other use
- USB BIOS Legacy Support:** default is Disabled. Set to enable for support of USB Keyboard and Mice
- Secured Setup Configurations:** default is No. Set to Yes to prevent a Plug’n’Play OS from changing system settings
- Installed OS:** Default is Other. Other option is Win95 if this is being used
- Reset Configuration Data:** Default is No. Set to YES if you want to clear the ESCD area
- Large Disk Access Mode:** Default is DOS. Set to Other if other OS is being used
- Remote Management baud Rate:** selects the baud rate for serial remote configuration mode.

The following options have a sub-menu for setup

- [Advanced Chipset Control](#)
- [I/O Device Configuration](#)
- [PCI Configuration](#)
- [Console Redirection](#)

Advanced Chipset Control

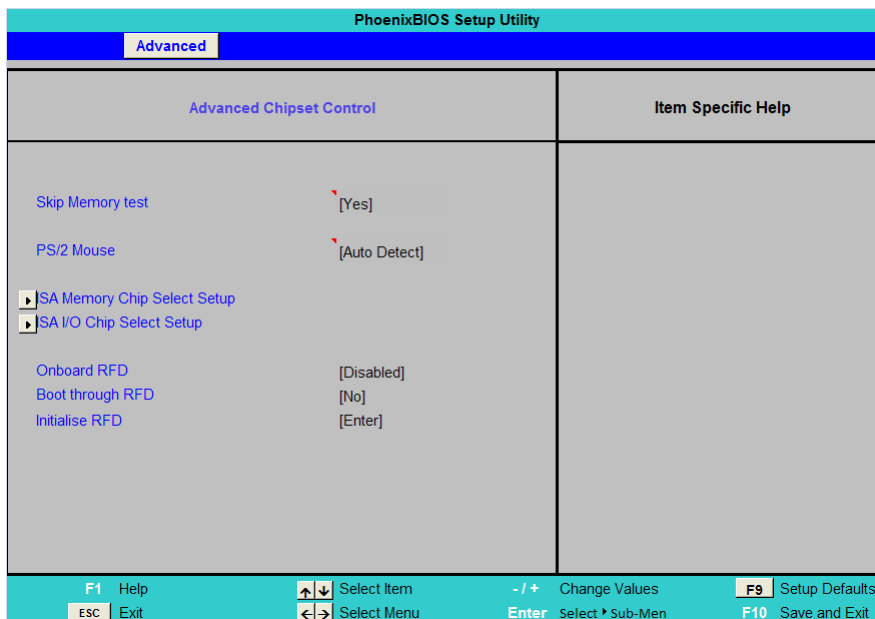


Figure E: Advanced Chipset Control

In this sub-menu, as shown in Figure E, the following options are available

Skip Memory Test: Default is Yes. This allows a faster post. Set to No to allow checking the memory during POST

PS/2 Mouse: Default is Auto Detect. Options are Enabled and Disabled. If Disabled, IRQ 12 is freed up for other use, specifically useful for PCI

ISA Memory Chip Select: Do not Use. These are for BIOS development use only

ISA I/O Chip Select: Do not Use. These are for BIOS development use only

Onboard RFD: No on-board Flash device. Leave as defaults

Boot through RFD: No on-board Flash device. Leave as defaults

Initialise RFD: No on-board Flash device. Leave as defaults

I/O Device Configuration

This sub-menu sets up Serial, parallel and Infrared ports, as well as Floppy and IDE controllers

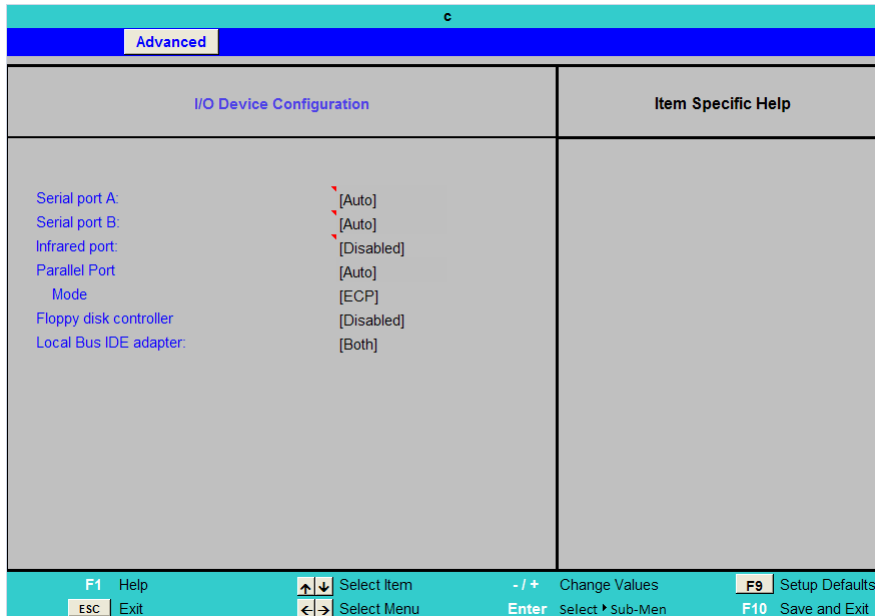


Figure F: I/O Device Configuration

- Serial Port A:** Default is Auto. Options are Auto, Enabled and Disabled
- Serial Port B:** Default is Auto. Options are Auto, Enabled and Disabled. [Refer to Hsetup for Mode selection](#)
- Infrared Port:** Default is Disabled. Options are Auto, Enabled and Disabled
- Parallel Port:** Default is Auto. Options are Auto, Enabled and Disabled
- Mode:** Default is ECP. Options are Output Only, Bi-directional, EPP and ECP
- Floppy Disk Controller:** Default is Disabled. This should be left at this setting as there is no floppy interface
- Local Bus IDE adapter:** Default is Both. Options are Disabled, Primary, Secondary and Both

PCI Configuration

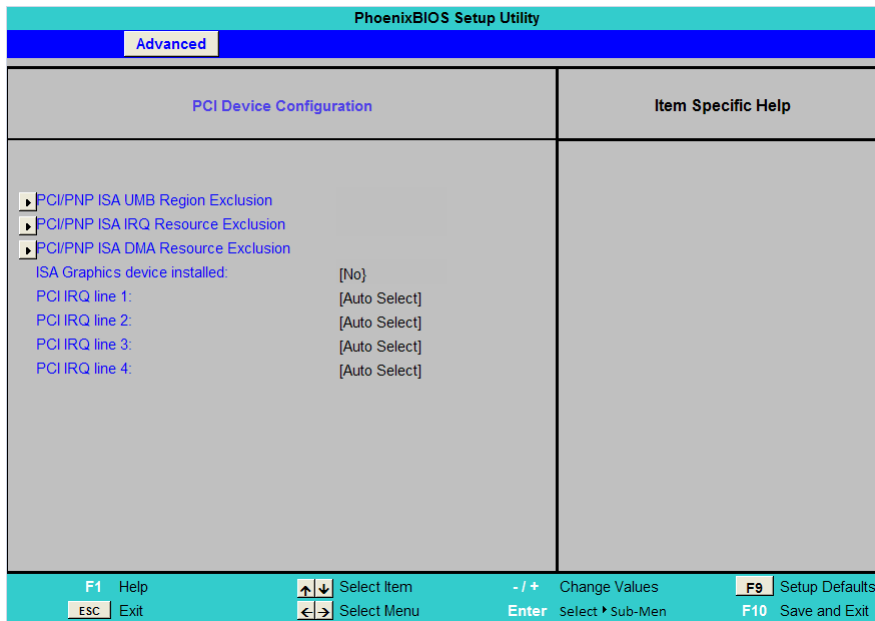


Figure G: I/O Device Configuration

ISA Graphics Device Installed: Default is No. Select YES to access palette data in PCI VGA device

PCI IRQ line 1 -4: Default is Auto Select. If ISA or EISA legacy cards are installed then manually select IRQ's for these lines

There are also 3 sub-menus for reserving ISA resources for use with legacy ISA devices

UMB Region Exclusion

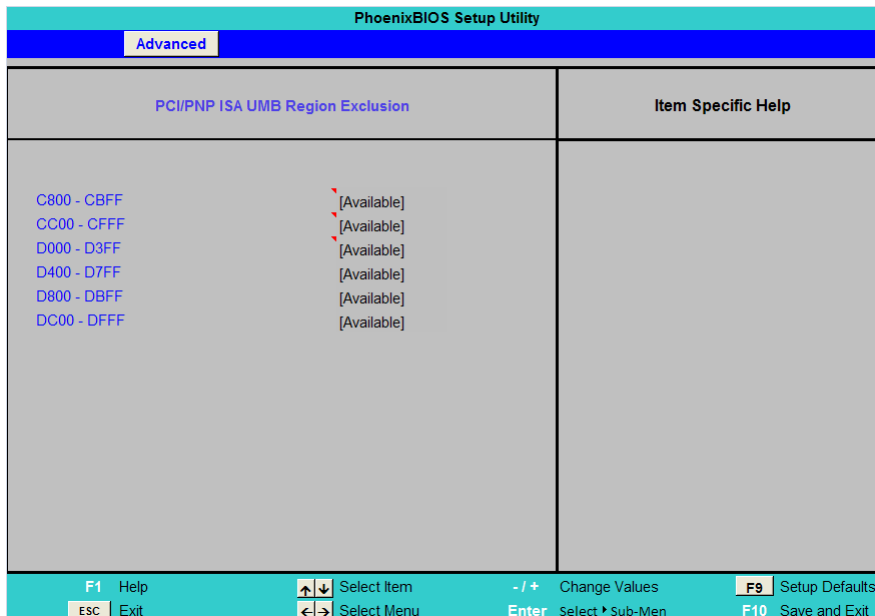


Figure H: ISA UMB region

This sub-menu as shown in Figure H: allows specified blocks of upper memory to be reserved for use by legacy ISA devices only

IRQ Resource Exclusion

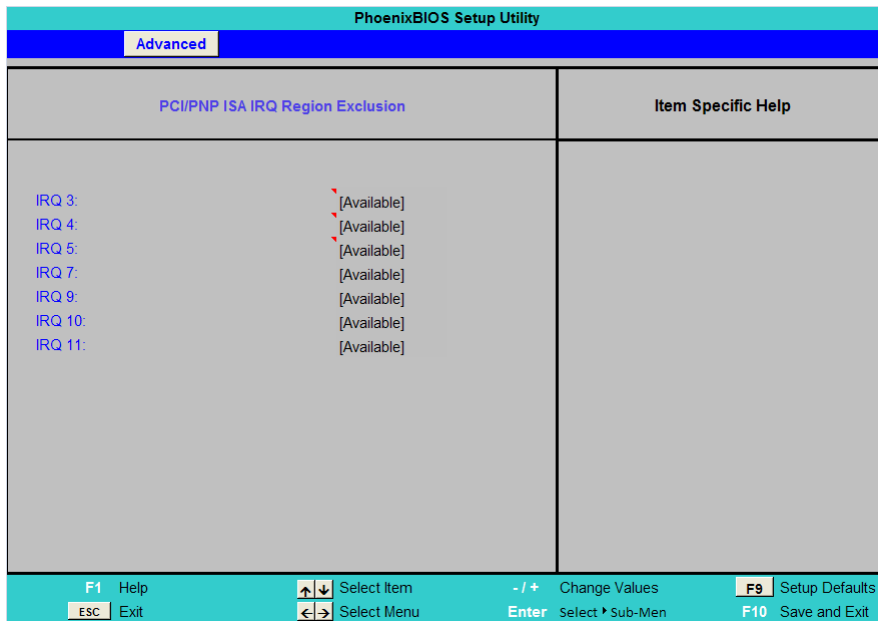


Figure I: IRQ Resources

The Sub-menu shown in Figure I, allows the specified IRQ resources to be reserved for use by legacy ISA devices only

DMA resource Exclusion

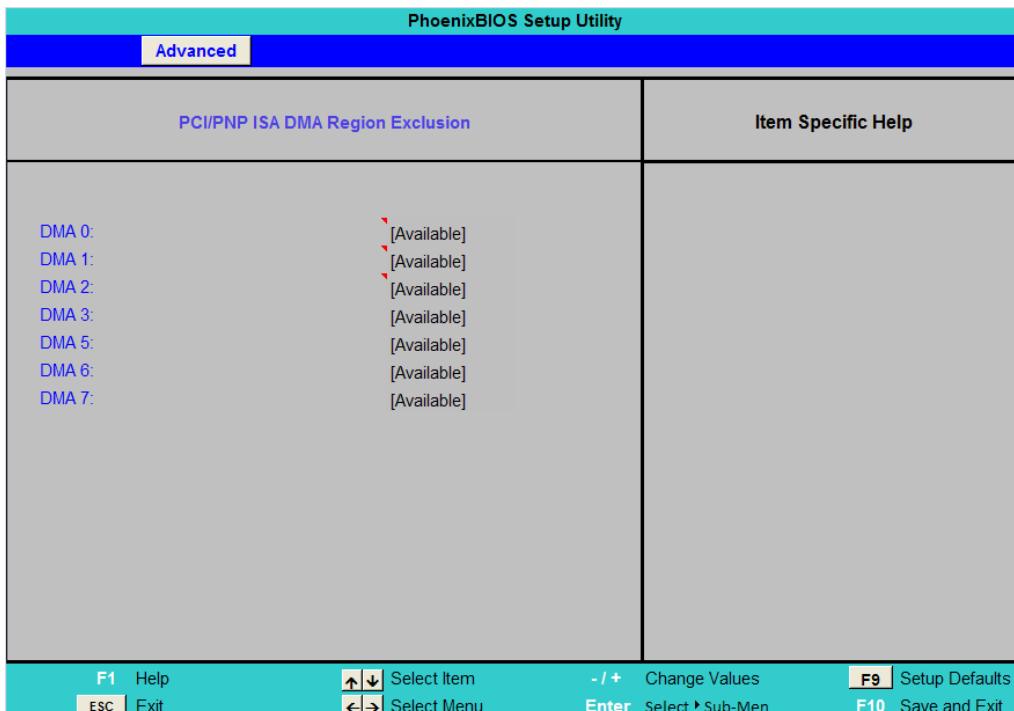


Figure J: DMA resources

The sub-menu shown in Figure J: allows the specified DMA resources to be reserved for use by non Plug-and-play ISA devices

Console Redirection

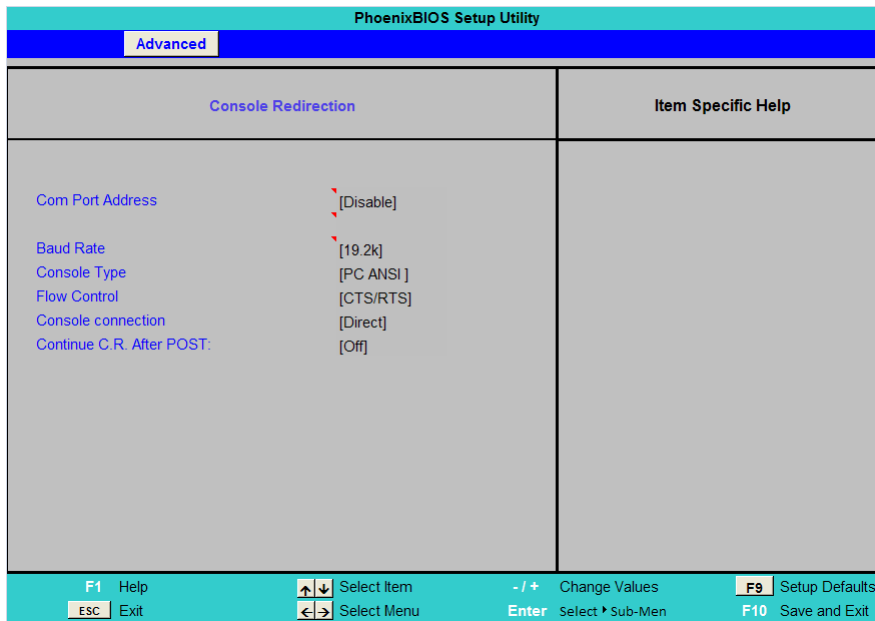


Figure K: Console redirection.

The last sub-menu in the Advanced section, allows the setting up of one of the com ports to act as a text output to a remote console

- Com Port Address:** Default is Disabled. Options are Disabled Com 1 or Com 2
- Console type:** Default is PC ANSI. Options are PC ANSI or VT100
- Flow Control:** Default is CTS/RTS: options are None, XON/XOFF or CTS/RTS
- Console Connection:** Default is Direct. Options are Direct or Via Modem
- Continue After POST:** Default is Off. Options are Off or ON

Security Setup Page

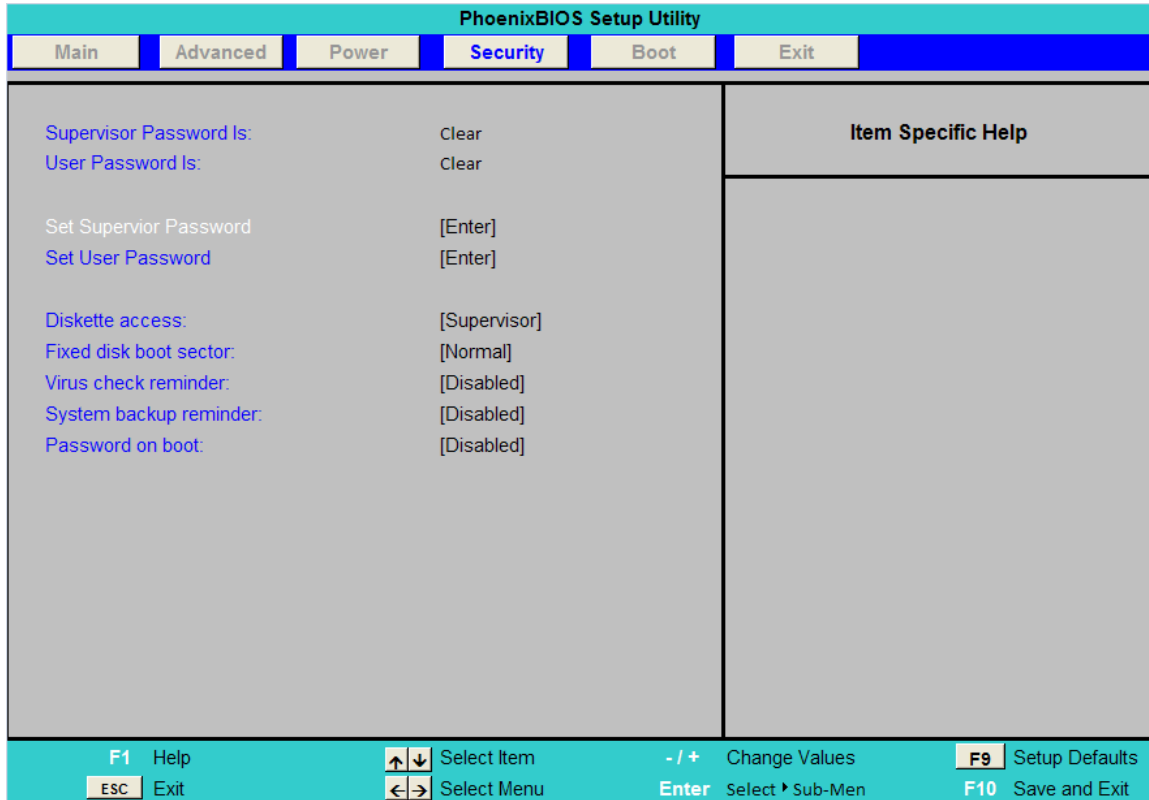


Figure L: Security menu

Use this menu to set Supervisor and Boot passwords.

The fixed disk Boot sector can be left at its default of Normal, or can be set to write protect to protect against viruses

Power Setup Page

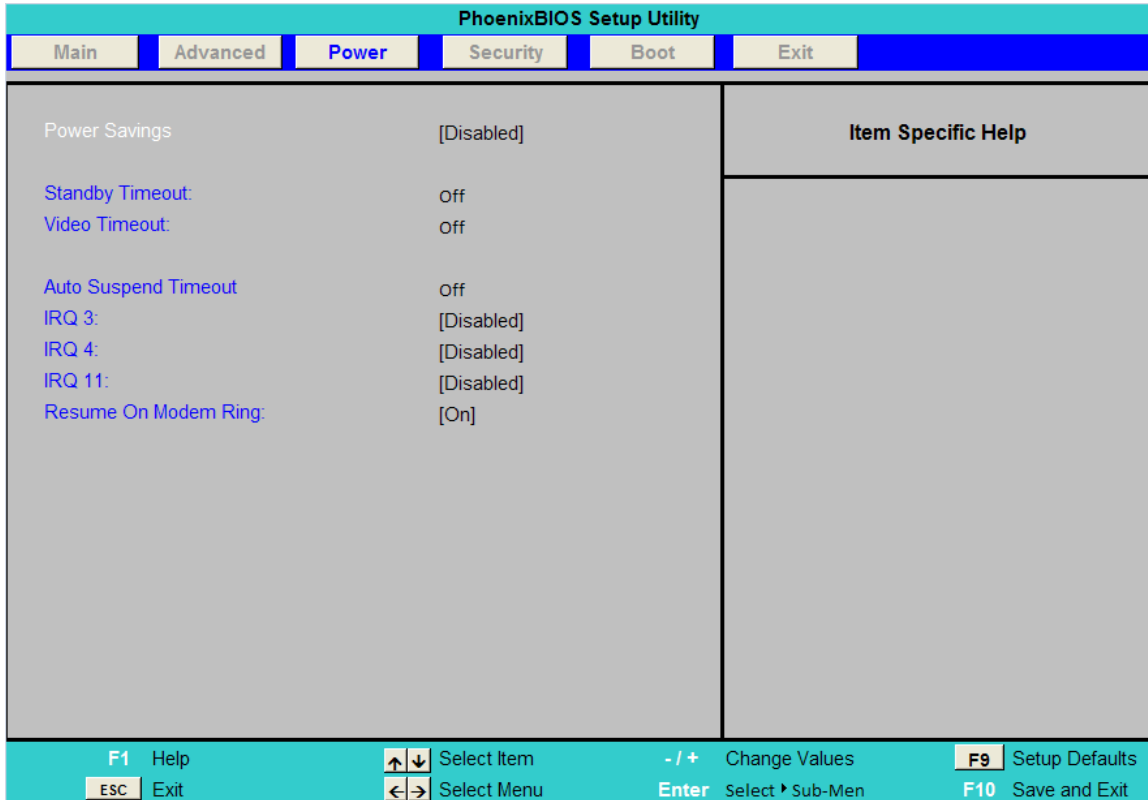


Figure M: Power menu

This Menu section sets up the unit so that it can automatically go into power saving modes,

Options for IRQ 3, 4 and 11, if Enabled, can allow the system to wake from a suspend state if that IRQ is detected

Boot Setup Page

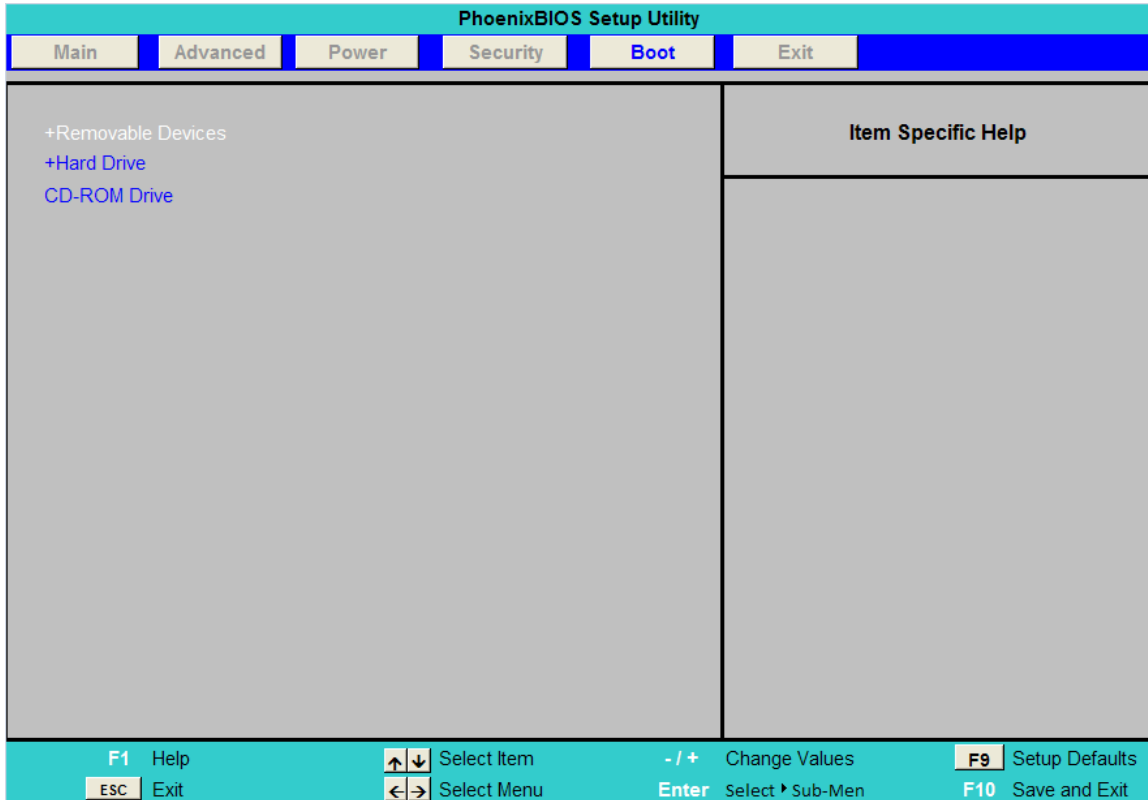


Figure N: Boot Setup

The Boot Setup page allows the setting for Boot Order. Note that if there is more than one Hard drive device is fitted, the BIOS automatically sets the boot order in line with

Primary Master > Primary Slave > Secondary Master.

On the APEX-ZF single board computer, the Compact flash is hard wired as Secondary Master. A Secondary Slave device is not supported.

Exit Setup Page

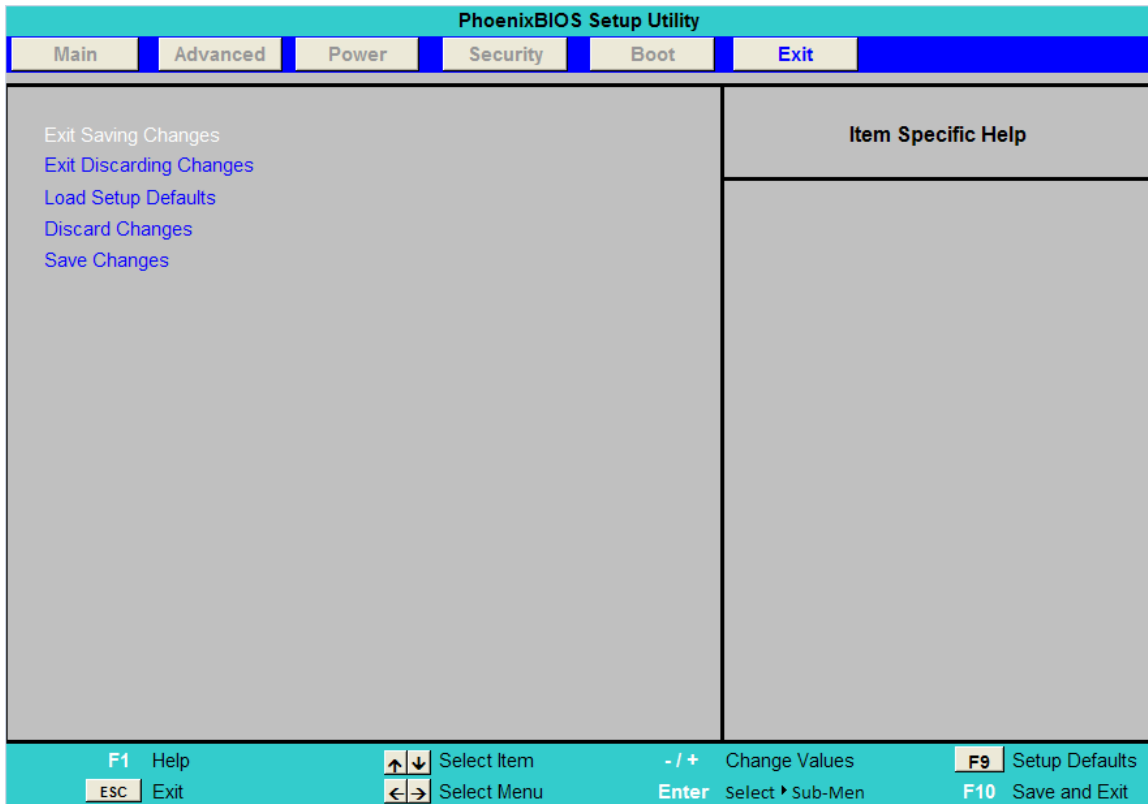


Figure O: Exit Setup menu

The Exit Setup page provides the ability for Loading default Settings as well as being able to exit with and without saving any BIOS setting changes made

Supervisory Microprocessor Setup – HSETUP

The HSETUP utility provides a means of

- Enabling / disabling the on-board Ethernet
- Setting the system frequency
- Setting Com 2 Mode

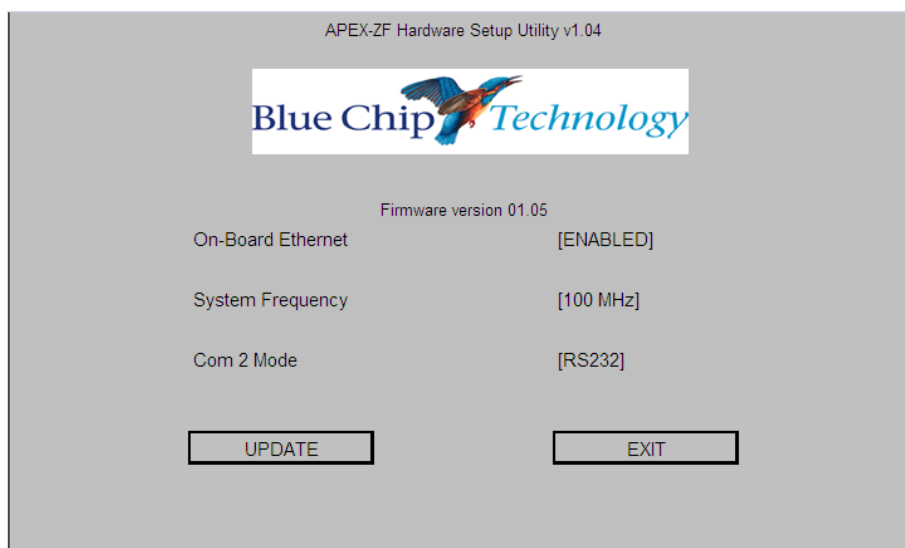


Figure P: Hsetup utility

The settings are adjusted by selecting the appropriate option from the pull down menus. To navigate, use the <TAB> key to move section, the Down Arrow to show the choices available, the up and down arrows to scroll through the options and the Enter key to select the appropriate option.

When selection is complete, press UPDATE. The APEX-ZF will then reboot and the new settings will be applied.

Ethernet: Default is Enabled. Option is Enabled or Disabled. If not being used, then Disable will free resources for other use

System Frequency: Default is 100 Mhz. Options are 33Mhz, 66Mhz or 100 Mhz. Selecting a slower speed will slow down the CPU and system bus

Com 2 Mode: Default is RS232. Options are RS232, RS422/RS485 Full Duplex or RS485 Half Duplex.

Appendix A – BIOS Int 50h Functions:

The following table details the Int 50h functions that are to be supported on the APEX-ZF PC/104 processor board. The input parameters and return values for each function will be detailed later in the chapter.

Function	Name
00h	GetBIOSRevision
01h	E2Read
02h	E2Write
03h	E2EraseByte
04h	E2EraseDevice
05h	WatchdogEnable
06h	WatchdogTick
07h	WatchdogResponse
08h	Shutdown Supplies - Reserved – Do Not Use
09h	ConfigureGPIO
0Ah	ReadGPIOBit
0Bh	WriteGPIOBit
0Ch	ReadGPIO
0Dh	WriteGPIO
0Eh	Unused
0Fh	Unused
10h	SmbusReadByte
11h	SmbusWriteByte
12h	SmbusReadWord
13h	SmbusWriteWord
14h	SmbusProcessCall
15h	Unused
16h	ReadVoltage
17h	ReadTemperature
18h	ReadFanSpeed
19h	SetCpuTempLimit
1Ah	Unused
1Bh	EnableLPTLCD
1Ch	LCDClearScreen
1Dh	LCDOutputCharacter
1Eh	Unused
1Fh	GetIRQSource

Where a function is defined as unused then the registers should not be changed but the carry flag should be set to signify an error

GetBIOSRevision

This function returns the version number of the system BIOS.

Input : AH = 00h

Return : AH = BIOS Version number
AL = BIOS Revision number
BL = BIOS Increment number
Carry flag clear.

i.e. For BIOS version 1.02.00, AH = 01h, AL = 02h, BL = 00h on return from function.

E2Read

This function reads a byte from the user EEPROM.

Input : AH = 01h
BX = EEPROM byte to read (0...499)

Return : DL = Value read from location given in BX.
Carry flag clear for success, set for error.

Note: the default entry for a location is FFh

E2Write

This function writes a byte to the user EEPROM and verifies that it has been written correctly.

Input : AH = 02h
BX = EEPROM byte to write (0...499)
DL = Value to write to location given in BX.

Return : Carry flag clear for success, set for error writing or verifying value

E2EraseByte

This function erases a single byte within the EEPROM.

Input : AH = 03h
BX = EEPROM byte to erase (0...499)

Return : Carry flag clear for success, set if byte not erased.

E2EraseDevice

This function bulk erases the whole of the EEPROM.

Input : AH = 04h

Return : Carry flag clear for success, set if device not erased.

Note: This resets all locations to FFh

WatchdogEnable

This function enables the watchdog timer and specifies the timeout period.

Input : AH = 05h
AL = 01h to enable watchdog, 00h to disable.
DL = Watchdog timeout in seconds. 01h to FFh are valid.

Return : Carry flag clear if watchdog enabled, set if invalid parameter given.

WatchdogTick

This function refreshes the watchdog timer.

Input : AH = 06h

Return : Carry flag clear for success.

WatchdogResponse

This function specifies to the microprocessor what action to take in the event of a watchdog timeout.

Input : AH = 07h
AL = Response to watchdog timeout (note only one bit should be set)
b7...2 are ignored
b1 – generate IRQ
b0 – reset system

Return : Carry flag clear for success, set for error.

ShutdownSupplies – Reserved – Do Not Use (may damageSBC)

This function turns off the 2.2V (CPU core) and 3.3V (IO) power supplies. The system is then restarted by pressing the reset button.

Input : AH = 08h

Return : Carry flag clear for success, set for error.

ConfigureGPIO

This function programs each GPIO bit to be input or an output.

Input : AH = 09h
CL = IO bit function, 0 = GPIO, 1 = analogue input. Only bits 3..0 are used, bits 7..4 are ignored.
DL = IO direction mask, 0 = output bit, 1 = input bit.
DH = Enable input bit pull-up resistors. 0 = disabled, 1 = enabled.

Return : Carry flag clear for success, set for error.

For example; CL=00, DL=00, DH=00 sets up 8Digital Outputs with no pull up resistors

ReadGPIOBit

This function reads the current value of a specific GPIO input bit or the last value programmed into an output bit.

Input : AH = 0Ah
AL=Bit mask position

Return : DL = Value read, bit 0 = GPIO0, bit 1 = GPIO1, etc.
Carry flag clear for success, set for error, Zero Flag set according to Bit value.

WriteGPIOBit

This function writes a new value to a specific GPIO output bit.

Input : AH = 0Bh
AL=Bit mask position
DL bit0 =Value to write.

Return : Carry flag clear for success, set for error.

ReadGPIO

This function reads the current value of the GPIO input bits and the last value programmed into output bits.

Input : AH = 0Ch

Return : DL = Value read, bit 0 = GPIO0, bit 1 = GPIO1, etc.
Carry flag clear for success, set for error.

WriteGPIO

This function writes a new value to the GPIO output bits.

Input : AH = 0Dh
DL = Value to write, bit 0 = GPIO0, bit 1 = GPIO1, etc.

Return : Carry flag clear for success, set for error.

SmbusReadByte

This function reads a byte from a device on the I²C bus.

Input : AH = 10h
BL = I²C device address
DL = I²C command

Return : Carry flag clear for success, set for error.
AL = Value read from I²C device.

For example, with BL = 28h (SMBUS Slave Address) and DL = 24h, this would be the equivalent of the ReadGPIO Int50 routine

SmbusWriteByte

This function writes a byte to a device on the I²C bus.

Input : AH = 11h
BL = I²C device address
CL = Data to write to I²C device
DL = I²C command

Return : Carry flag clear for success, set for error.

For example, with BL = 28h (SMBUS Slave Address), CL = 20h and DL = 00h, this will set the GPIO/Analogue function to GPIO

SmbusReadWord

This function reads a word from a device on the I²C bus.

Input : AH = 12h
BL = I²C device address
DL = I²C command

Return : Carry flag clear for success, set for error.
AX = Value read from I²C device.

SmbusWriteWord

This function writes a word to a device on the I²C bus.

Input : AH = 13h
BL = I²C device address
CX = Data to write to I²C device
DL = I²C command

Return : Carry flag clear for success, set for error.

SmbusProcessCall

This function allows a word to be written and a word to be read in a single transaction to a device on the I²C bus.

Input : AH = 14h
BL = I²C device address
CX = Data to write to I²C device
DL = I²C command

Return : Carry flag clear for success, set for error.
AX = Value read from I²C device

ReadVoltage

This function reads a voltage within the hardware monitor.

Input : AH = 16h
AL = Voltage input to read

Return : AL = Voltage input read
DH = Integer part of voltage (BCD)
DL = Decimal part of voltage (BCD)
Carry flag clear for success, set for error.

For a voltage of 5.03V DH = 05h, DL = 03h.

ReadTemperature

This function reads a temperature within the hardware monitor.

Input : AH = 17h
AL = Temperature input to be read

Return : AL = Temperature input read
DX = Temperature in Celsius (BCD)
Carry flag clear for success, set for error.

ReadFanSpeed

This function reads a fan speed within the hardware monitor. NB – not supported on APEXZF platform.

Input : AH = 18H
AL = Fan speed input to be read

Return : AL = Fan speed input read
DX = Fan speed in RPM
Carry flag clear for success, set for error.

SetCpuTempLimit

This function sets the cpu temperature limit which triggers an IRQ.

Input : AH = 19H
DX = Temperature in Celsius (BCD)

Return : Carry flag clear for success, set for error.

For example to set a temperature of +44C, DX would be 044C

EnableLPTLCD

This function initializes a character mode LCD screen interfaced to the parallel port.

Input : AH = 1Bh

Return : Carry flag clear for success, set for error.

LCDClearScreen

This function clears the character mode LCD screen.

Input : AH = 1Ch

Return : Carry flag clear for success, set for error.

LCDOutputCharacter

This function outputs a specified ASCII character to a given screen location.

Input : AH = 1Dh
 AL = ASCII character code
 DH = Screen X location
 DL = Screen Y location.
Return : Carry flag clear for success, set for error.

GetIRQSource

This function returns the source of the IRQ so that the host system can take the appropriate action.

Input : AH = 1Fh
Return : DL = Value read,
 bit 7-3 unused,
 bit 2 = CPU temp limit exceeded,
 bit 1 = GPIO input pin change of state,
 bit 0 = WD timeout.
 Carry flag clear for success, set for error.

Appendix B – SMBUS Commands

The communication between the ZFx86 processor and the control microprocessor on the APEX-ZF processor board is performed using the SMBUS. The following section details the commands that are supported and the format of the data that is passed into and returned by the functions.

SMBUS Slave Address

The microprocessor responds to SMBUS slave address 0x28. This value should be built into the firmware and should not be configurable by the user.

Watchdog functions

EnableWatchdog

To enable the watchdog timer a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_ENABLE_WATCHDOG (see appendix C).</i>
Data	<i>0x00 invalid value and is ignored. 0x01...0xFF defines the timeout period in seconds.</i>

DisableWatchdog

To disable the watchdog timer a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_DISABLE_WATCHDOG (see appendix C).</i>
Data	<i>This parameter is ignored.</i>

RefreshWatchdog

To refresh the watchdog timer a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_REFRESH_WATCHDOG (see appendix C).</i>
Data	<i>This parameter is ignored.</i>

SetWatchdogResponse

To set the response of the microprocessor in the event of a watchdog timeout a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_WATCHDOG_RESPONSE (see appendix C).</i>
Data	<i>Bit mask to signify what action to take in the event of timeout Bit 0 – Reset system Bit 1 – Generate IRQ to host Note – only one option can be selected at any given time.</i>

EEPROM Functions

SetEepromPointer

In order to read a value from, write a value to or erase a location in the EEPROM then an internal pointer needs to be setup to point to the correct location. This is achieved by a “write word” SMBUS command issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_EEPROM_POINTER (see appendix C).</i>
Data	<i>0x0000...0x01F3 defines the new EEPROM pointer location. 0x01F4...0xFFFF invalid values and are ignored.</i>

WriteEepromByte

To write a byte to the EEPROM a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_WRITE_EEPROM (see appendix C).</i>
Data	<i>The 8 bit value to be written to the EEPROM.</i>

ReadEepromByte

To read a byte from the EEPROM a “read byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_READ_EEPROM (see appendix C).</i>
Data	<i>The 8 bit value read from the EEPROM is returned.</i>

EraseEepromByte

To erase a byte in the EEPROM a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_ERASE_EEPROM_BYTE (see appendix C).</i>
Data	<i>Ignored.</i>

EraseEeprom

To erase the entire contents of the user accessible part of the EEPROM a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_ERASE_EEPROM (see appendix C).</i>
Data	<i>Ignored.</i>

GPIO / ADC Functions

SetGpioPinFunction

To specify the function of the shared GPIO / ADC pins a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_GPIO_FUNCTION (see appendix C).</i>
Data	<i>b7...4 are unused and are ignored. b3...0 specifies the function for GPIO pins 0 to 3. If the bit is set to “1” then the corresponding pin is configured as an analogue input, if the bit is set to “0” then the pin is a GPIO bit.</i>

SetGpioBitDirection

To specify the direction of the GPIO pins a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_GPIO DIRECTIONS (see appendix C).</i>
Data	<i>b7...0 specifies the IO direction for the GPIO pins. If the bit is set to “1” then the corresponding GPIO pin is configured as an input, if the bit is set to “0” then the pin is an output.</i>

SetGpioPullups

To specify whether the internal pull-up resistors of GPIO pins configured as inputs should be enabled a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_GPIO_PULLUPS (see appendix C).</i>
Data	<i>b7...0 specifies whether to enable the internal pull-up resistors on GPIO pins set as inputs. If the bit is set to “1” then the corresponding input pin has its pull-up resistor enabled, if the bit is set to “0” then the pull-up resistor is disabled.</i>

WriteGpioPort

To set the state of the GPIO pins that are configured as an output a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_WRITE_GPIO (see appendix C).</i>
Data	<i>b7...0 contains the bit states to write to GPIO bits 0 to 7</i>

ReadGpioPort

To read the state of the GPIO pins that are configured as an input a “read byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_READ_GPIO (see appendix C).</i>
Data	<i>b7...0 contains the bit states read from GPIO bits 0 to 7. Bits that are configured as inputs are read directly from the input, bits that are configured as outputs return the last value written.</i>

SetGpioBit

To set the state of an individual GPIO bit that is configured as an output then the API layer software, or BIOS extension, should issue a call to ReadGpioPort to get the current state of the bits, perform the appropriate masking operations and then modify the bits using a call to WriteGpioPort.

GetGpioBit

To get the state of an individual GPIO bit that is configured as an input then the API layer software, or BIOS extension, should issue a call to ReadGpioPort to get the current state of all the bits and perform the appropriate masking operations and return the appropriate value to the application program.

GetAnalogueInput

To read the raw A/D value from an analogue input a “process call” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_READ_ANALOGUE_INPUT (see appendix C).</i>
Sent data	<i>b15...4 not used and are ignored b3...0 specifies the A/D channel to read (4 bits are used to support a maximum of 16 A/D channels to provide future proofing using other microprocessors from the AVR family on future hardware designs)</i>
Returned data	<i>b15...12 set to the A/D channel number read b11...10 not used and are ignored b9...0 the raw 10bit A/D value read.</i>

For example; return data of 03FFh would equate to 2.56V on channel 0, 0000h would be 0V

0277h would be 1.579V: 0227h = 631 decimal. Voltage = (631*2.56)/1023 = 1.579V

System Monitor Functions

ReadSystemMonitor

To read the raw A/D value from one of the system hardware monitor inputs a “process call” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_READ_SYSTEM_MONITOR (see appendix C).</i>
Sent data	<i>b15...3 not used and are ignored b2...0 specifies the system hardware monitor channel to read (3 bits are used to support a maximum of 8 hardware monitor inputs to provide future proofing using other microprocessors from the AVR family on future hardware designs) 000 – CPU core voltage 001 – 3.3V 010 – 5V 011 – Reserved for future use 100 – Reserved for future use 101 – System temperature 110 – Reserved for future use 111 – Reserved for future use</i>
Returned data	<i>b15...12 set to the A/D channel number read b11...10 not used and are ignored b9...0 the raw 10bit A/D value read.</i>

SetCpuTempLimit

To provide the cpu temperature upper threshold, a “write word” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SET_CPU_TEMP_LIMIT (see appendix C).</i>
Data	<i>b15...8 ADC value for CPU temperature limit (high byte). b7...0 ADC value for CPU temperature limit (low byte).</i>

Miscellaneous Functions

GetFirmwareVersion

It is envisaged that some customer specific versions of firmware will be produced for the microprocessor and these will be identified using specific version numbers. In order for the BIOS or API layer software to determine if a particular function is supported by the microprocessor a “read word” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_GET_VERSION (see appendix C).</i>
Data	<i>b15...8 gives the major version number b7...0 gives the minor version number.</i>

Reset Functions

ResetMicroProcessor

To reset the Supervisory microprocessor and reload its configuration followed by a reset of the ZFx86 a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_RESET_MICROPROCESSOR (see appendix C).</i>
Data	<i>Ignored</i>

ResetSystem

To reset the ZFx86 processor a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_RESET_SYSTEM (see appendix C).</i>
Data	<i>Ignored</i>

GetIrqSource

To get details of which function has generated the IRQ a “read byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_GET_IRQ_SOURCE (see appendix C).</i>
Returned data	<i>b7...3 not used and are ignored b2 CPU temperature has exceeded specified limit b1 A GPIO input pin has changed state b0 Watchdog timeout has occurred</i>

PowerOffSupplies – Reserved – Do not use

To turn off the 2.2V (CPU core voltage) and 3.3V (IO) supplies a “write byte” SMBUS command should be issued with the following format:

Slave address	<i>As defined in the SMBUS slave address section of this document.</i>
Command code	<i>SMBUS_COMMAND_SHUTDOWN_SUPPLIES (see appendix C).</i>
Data	<i>Ignored</i>

Appendix C: SMBUS Command Codes

The following table lists the full set of SMBUS command codes to be used in communication between software running on the ZFx86 processor and the control microprocessor.

SMBUS Command Name	Value
SMBUS_COMMAND_ENABLE_WATCHDOG	0x00
SMBUS_COMMAND_DISABLE_WATCHDOG	0x01
SMBUS_COMMAND_REFRESH_WATCHDOG	0x02
SMBUS_COMMAND_SET_WATCHDOG_RESPONSE	0x03
SMBUS_COMMAND_SET_EEPROM_POINTER	0x10
SMBUS_COMMAND_WRITE_EEPROM	0x11
SMBUS_COMMAND_READ_EEPROM	0x12
SMBUS_COMMAND_ERASE_EEPROM_BYTE	0x13
SMBUS_COMMAND_ERASE_EEPROM	0x14
SMBUS_COMMAND_SET_GPIO_FUNCTION	0x20
SMBUS_COMMAND_SET_GPIO DIRECTIONS	0x21
SMBUS_COMMAND_SET_GPIO_PULLUPS	0x22
SMBUS_COMMAND_WRITE_GPIO	0x23
SMBUS_COMMAND_READ_GPIO	0x24
SMBUS_COMMAND_READ_ANALOGUE_INPUT	0x25
SMBUS_COMMAND_READ_SYSTEM_MONITOR	0x40
SMBUS_COMMAND_SET_CPU_TEMP_LIMIT	0x41
SMBUS_COMMAND_GET_VERSION	0x50
SMBUS_COMMAND_RESET_MICROPROCESSOR	0x60
SMBUS_COMMAND_RESET_SYSTEM	0x61
SMBUS_COMMAND_GET_IRQ_SOURCE	0x62
SMBUS_COMMAND_SHUTDOWN_SUPPLIES (Reserved – Do Not Use)	0x63

Maintenance

The APEX ZF Single Board Computer does not itself require any regular maintenance.

Amendment History

Issue Level	Issue Date	Author	Amendment Details
1.0	31/08/2007	T Mck	First Release
1.1	27/11/2007	T Mck	Add MTBF and GPIO data

Contact Details

Blue Chip Technology Ltd.
Chowley Oak
Tattenhall
Chester
CH3 9EX
U.K.

Telephone: +44 (0)1829 772000

Facsimile: +44 (0)1829 772001

www.bluechiptechnology.co.uk

Plasma PC Sales

PlasmaPC@bluechiptechnology.co.uk

Single Board Computer Sales

singleboardcomputer@bluechiptechnology.co.uk

Rack mount PC Sales

rackmountpc@bluechiptechnology.co.uk

Data and IO Sales

DataIO@bluechiptechnology.co.uk

Technical Support

Support@bluechiptechnology.co.uk