kontron

EB425 / EB420

E^2Brain™ High Performance XScale™ Processor Module

Manual ID: 27854, Rev. Index 02
31 January, 2006

User Guide

The product described in this manual is in compliance with all applied CE standards.

27854.02.VC.060131/114633

Revision History

Publication Title:	EB425 / EB420 E²Brain™ High Performance XScale™ Processor Module	
ID Number:	27854	
Rev. Index	Brief Description of Changes	Date of Issue
01	Initial Issue	6 Apr, 2004
02	Manual revised to include EB420 information and Errata 1	31 Jan, 2006

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH
Sudetenstraße 7
D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON
sales@kontron.com

For further information about other Kontron Modular Computers products, please visit our Internet web site: www.kontron.com

Copyright

Disclaimer:

27854.02.VC.060131/114633

Table of Contents

Chapter 2

27854.02.VC.060131/114634

Chapter 3

Chapter 4

Chapter 5

Annex A

Annex B

27854.02.VC.060131/114635

This page has been intentionally left blank.

List of Tables

List of Figures

27854.02.VC.060131/114636

This page has been intentionally left blank.

27854.02.VC.060131/114636

Proprietary Note

This document contains information proprietary to Kontron Modular Computers GmbH. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of Kontron Modular Computers GmbH or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron Modular Computers GmbH cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

Kontron Modular Computers GmbH reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron Modular Computers GmbH without further notice.

Trademarks

Kontron Modular Computers GmbH, the *PEP* logo and, if occurring in this manual, “CXM” are trademarks owned by Kontron Modular Computers GmbH, Kaufbeuren (Germany). In addition, this document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

Environmental Protection Statement

This product has been manufactured to satisfy environmental protection requirements where possible. Many of the components used (structural parts, printed circuit boards, connectors, batteries, etc.) are capable of being recycled.

Final disposition of this product after its service life must be accomplished in accordance with applicable country, state, or local laws or regulations.

Explanation of Symbols

CE Conformity

This symbol indicates that the product described in this manual is in compliance with all applied CE standards. Please refer also to the section "Applied Standards" in this manual.

Caution, Electric Shock!

This symbol and title warn of hazards due to electrical shocks (> 60V) when touching products or parts of them. Failure to observe the precautions indicated and/or prescribed by the law may endanger your life/health and/or result in damage to your material.

Please refer also to the section "High Voltage Safety Instructions" on the following page.

Warning, ESD Sensitive Device!

This symbol and title inform that electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times.

Please read also the section "Special Handling and Unpacking Instructions" on the following page.

Warning!

This symbol and title emphasize points which, if not fully understood and taken into consideration by the reader, may endanger your health and/or result in damage to your material.

Note ...

This symbol and title emphasize aspects the reader should read through carefully for his or her own advantage.

For Your Safety

Your new Kontron product was developed and tested carefully to provide all features necessary to ensure its compliance with electrical safety requirements. It was also designed for a long fault-free life. However, the life expectancy of your product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of your own safety and of the correct operation of your new Kontron product, you are requested to conform with the following guidelines.

High Voltage Safety Instructions

Warning!

All operations on this device must be carried out by sufficiently skilled personnel only.

Caution, Electric Shock!

Before installing your new Kontron product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks.

Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.

Special Handling and Unpacking Instructions

ESD Sensitive Device!

Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times.

Do not handle this product out of its protective enclosure while it is not used for operational purposes unless it is otherwise protected.

Whenever possible, unpack or pack this product only at EOS/ESD safe work stations. Where a safe work station is not guaranteed, it is important for the user to be electrically discharged before touching the product with his/her hands or tools. This is most easily done by touching a metal part of your system housing.

It is particularly important to observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings etc. If the product contains batteries for RTC or memory backup, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. They can cause short circuits and damage the batteries or conductive circuits on the board.

27854.02.VC.060131/114637

General Instructions on Usage

In order to maintain Kontron's product warranty, this product must not be altered or modified in any way. Changes or modifications to the device, which are not explicitly approved by Kontron Modular Computers GmbH and described in this manual or received from Kontron's Technical Support as a special handling instruction, will void your warranty.

This device should only be installed in or connected to systems that fulfill all necessary technical and specific environmental requirements. This applies also to the operational temperature range of the specific board version, which must not be exceeded. If batteries are present, their temperature restrictions must be taken into account.

In performing all necessary installation and application operations, please follow only the instructions supplied by the present manual.

Keep all the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board, please re-pack it as nearly as possible in the manner in which it was delivered.

Special care is necessary when handling or unpacking the product. Please consult the special handling and unpacking instruction on the previous page of this manual.

Two Year Warranty

Kontron Modular Computers GmbH grants the original purchaser of Kontron's products a ***TWO YEAR LIMITED HARDWARE WARRANTY*** as described in the following. However, no other warranties that may be granted or implied by anyone on behalf of Kontron are valid unless the consumer has the express written consent of Kontron Modular Computers GmbH.

Kontron Modular Computers GmbH warrants their own products, excluding software, to be free from manufacturing and material defects for a period of 24 consecutive months from the date of purchase. This warranty is not transferable nor extendible to cover any other users or long-term storage of the product. It does not cover products which have been modified, altered or repaired by any other party than Kontron Modular Computers GmbH or their authorized agents. Furthermore, any product which has been, or is suspected of being damaged as a result of negligence, improper use, incorrect handling, servicing or maintenance, or which has been damaged as a result of excessive current/voltage or temperature, or which has had its serial number(s), any other markings or parts thereof altered, defaced or removed will also be excluded from this warranty.

If the customer's eligibility for warranty has not been voided, in the event of any claim, he may return the product at the earliest possible convenience to the original place of purchase, together with a copy of the original document of purchase, a full description of the application the product is used on and a description of the defect. Pack the product in such a way as to ensure safe transportation (see our safety instructions).

Kontron provides for repair or replacement of any part, assembly or sub-assembly at their own discretion, or to refund the original cost of purchase, if appropriate. In the event of repair, refunding or replacement of any part, the ownership of the removed or replaced parts reverts to Kontron Modular Computers GmbH, and the remaining part of the original guarantee, or any new guarantee to cover the repaired or replaced items, will be transferred to cover the new or repaired items. Any extensions to the original guarantee are considered gestures of goodwill, and will be defined in the "Repair Report" issued by Kontron with the repaired or replaced item.

Kontron Modular Computers GmbH will not accept liability for any further claims resulting directly or indirectly from any warranty claim, other than the above specified repair, replacement or refunding. In particular, all claims for damage to any system or process in which the product was employed, or any loss incurred as a result of the product not functioning at any given time, are excluded. The extent of Kontron Modular Computers GmbH liability to the customer shall not exceed the original purchase price of the item for which the claim exists.

Kontron Modular Computers GmbH issues no warranty or representation, either explicit or implicit, with respect to its products' reliability, fitness, quality, marketability or ability to fulfil any particular application or purpose. As a result, the products are sold "as is," and the responsibility to ensure their suitability for any given task remains that of the purchaser. In no event will Kontron be liable for direct, indirect or consequential damages resulting from the use of our hardware or software products, or documentation, even if Kontron were advised of the possibility of such claims prior to the purchase of the product or during any period since the date of its purchase.

Please remember that no Kontron Modular Computers GmbH employee, dealer or agent is authorized to make any modification or addition to the above specified terms, either verbally or in any other form, written or electronically transmitted, without the company's consent.

27854.02.VC.060131/114638

This page has been intentionally left blank.

Chapter 1

Introduction

This page has been intentionally left blank.

1. Introduction

The E²Brain™ modules described in this manual are designed to Kontron Modular Computers' “E²Brain™ Module Specification”, revision 01. Section 1.1 provides general information regarding this specification and the system environment as envisioned for E²Brain™ modules. For more detailed information regarding the “E²Brain™ Module Specification”, please contact Kontron Modular Computers. For simplification, in the remaining User Guide text the EB425 / EB420 module is refered to as EB42x.

The remaining sub-sections of the Introduction provide more specific details of the EB42x E²Brain™ modules to familiarize the user with this product as a whole.

1.1 The E²Brain™ Concept

The E²Brain™ concept is a highly flexible approach to providing application developers with the ability to concentrate on the definition of application requirements without having to continuously factor in potential restrictions concerning available data processing and communications functionality.

More specific, data processing and communications requirements become a function of the application and not vice versa. This is possible through the implementation of the E²Brain™ concept. Unlike other approaches to providing application solutions, the E²Brain™ concept concentrates on the most essential aspects of providing data processing and communications without attempting to provide in one entity a complete, self-contained computer system.

The E²Brain™ specification first of all defines a PCB module with a form factor of 115 x 75 x 11.6 millimeters. For interfacing with applications, the specification calls for up to four connectors which provide not only interfacing capability for current industry standards but also for future standards or application specific requirements. The type, location, and usage of these connectors is also defined in the specification so as to guarantee standardized compatibility. The specification is open as to the data processing and communications functionality to be implemented which is by definition a function of the application requirements. In addition, the specification envisions considerations for thermal energy dissipation through the implementation of what are to be known as BrainCAP™s (E²Brain™ **C**ooling **A**ssembly, **P**rotector) which can range from heat spreaders to highly sophisticated heat sink cooling designs.

The key features of the E²Brain™ concept are:

- Very compact and robust form factor
- Independent of CPU architecture
- Scalable, flexible, and open system interface
- PCI Master and Agent Mode
- PCI-64 and PCI-X capability
- Versatile and very powerful communication interfaces
- Complete thermal design concept

1.1.1 E²Brain™ Functionality

E²Brain™ (Embedded Electronic Brain) is a new platform architecture for advanced computer modules. The E²Brain™ specification defines a very compact mechanical form factor and a comprehensive set of functional interfaces which can be adapted to a wide variety of applications. E²Brain™ modules provide complete computer cores integrating a high performance CPU, system memory and - typical for E²Brain™ - advanced communication controllers.

27854.02.VC.060131/114639

E²Brain™ modules are plugged into customized backplanes or standardized carrier boards which themselves provide the physical interfaces (PHYs) and connectors, power, and additional IO controllers. Through the use of E²Brain™ modules the system developer is relieved of the task of designing computers, and, instead, they permit him to concentrate on the specific product development.

E²Brain™ is a computer platform dedicated not just to one architecture like the PC and Windows architecture, but it is open for all architectures including PowerPC, ARM, SH, x86, and others. E²Brain™ modules are very suitable for “deeply” embedded applications requiring flexible computing power combined with versatile and high performance communication power.

Although typical E²Brain™ modules are designed to be low power consumption devices, they are part of a well thought out thermal design concept which considers the thermal aspects right from the beginning. Where higher power consumption is unavoidable, E²Brain™ modules are fitted with appropriate BrainCAP™s. By utilizing BrainCAP™s, cooling, mechanical stabilization, and EMI protection are combined in a single concept to satisfy almost any application requirement.

1.1.2 Basic Architecture

The following figure illustrates the basic functional architecture of E²Brain™ modules. Common to all E²Brain™ modules are the data processing and communications core and the system and communication interfaces.

The application requirements determine the functionality required of the E²Brain™ module core which in turn mandates the functionality to be provided by the system and communication interfaces. Both of these interfaces are comprised of a base set and an extended set of functional features.

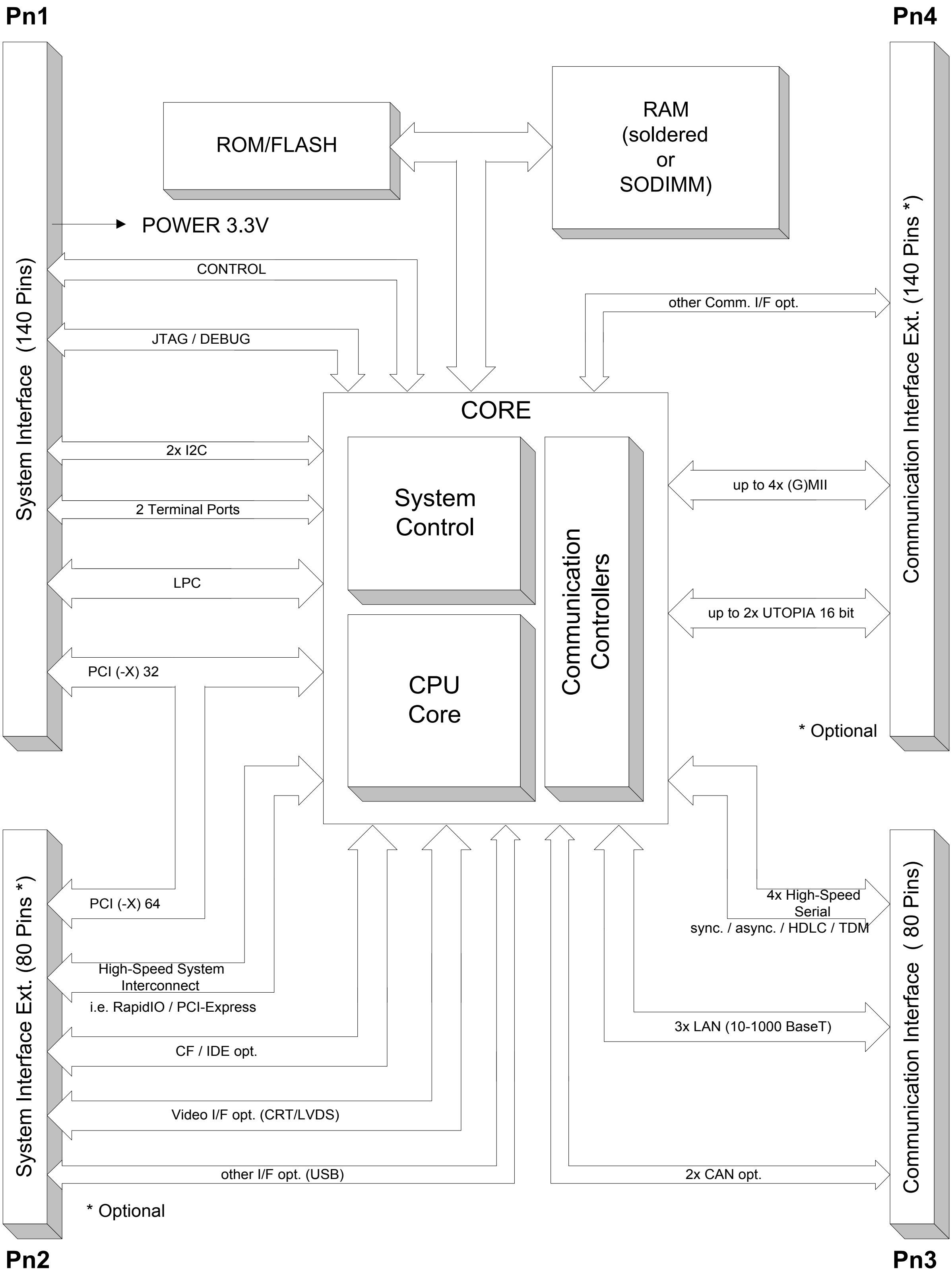
The system interface to the application is accomplished through connectors Pn1 and Pn2. Connector Pn1 provides the base set of system interfacing and Pn2 the extended set. If the application does not require extended system interfacing, it is not necessary to populate connector Pn2. The same analogy applies to the communication interfacing whereby connector Pn3 provides the base set of communication interfacing and Pn4 the extended set. Pn4 is not required to be populated if there is not an application requirement for it. This concept together with a corresponding core provides a maximum of scalability and flexibility to satisfy the most demanding of applications.

1.1.3 E²Brain™ System Environment

The E²Brain™ module form factor and mechanical and electrical interfacing are so conceived as to allow the use of E²Brain™ modules in practically any kind of system environment. These mezzanine modules can easily be integrated on most standardized carrier boards (VME, CompactPCI, PC PCI, etc.) as well as any other conceivable type of carrier board capable of providing the required mechanical and electrical infrastructure.

In addition to this infrastructure, thermal energy dissipation requirements must be taken into consideration when implementing applications using E²Brain™ modules. The E²Brain™ concept basically calls for modules to provide their own thermal energy dissipation. It may, however, be necessary to add additional thermal energy dissipation capability depending on the overall system environment. To satisfy such requirements, E²Brain™ modules may be equipped with specially designed cooling devices that are adapted to the specific system environment.

Figure 1-1: E²Brain™ Basic Architecture



1.2 EB425 / EB420 System Overview

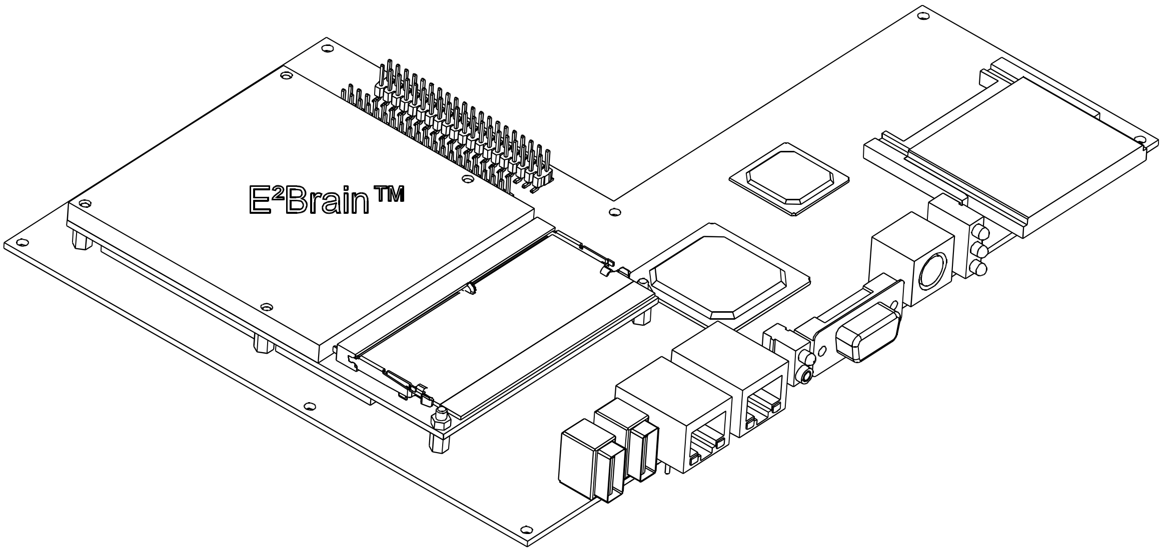
This E²Brain™ module is designed for applications requiring cost effective, high performance data processing capability where compact size, low power consumption, and interfacing flexibility are key factors for achieving a successful system design.

Specifically, system integrators are provided with a comprehensive set of industry standard interfaces from a range of low, medium, and high-speed system level data exchange and monitor and control I/Os, and communication I/Os such as Ethernet, CAN, and high-speed serial data I/Os.

System integration of an EB42x E²Brain™ module requires a carrier board (industry standard or custom) which interfaces the application with the EB42x module. All signal conditioning, mechanical, and electrical interfacing with the application must be accomplished by the carrier board.

In addition, physical packaging and thermal energy dissipation must be provided by the system. As the EB42x is designed as a low power consumption device, it does not pose any special requirements for cooling. Integration on a carrier board is done via the four connectors and standoffs which allow for secure mounting to the carrier board. The following figure illustrates a typical integration of an E²Brain™ module on a custom carrier board.

Figure 1-2: E²Brain™ Module on a Custom Carrier Board



27854.02.VC.060131/114639

1.3 Product Overview

The EB42x E²Brain™ High Performance XScale™ Processor Module is a part of an innovative concept to provide system integrators with a complete range of off-the-shelf as well as custom embedded computer cores for the most demanding of applications.

This module, designed around the Intel® IXP42x Integrated Processor, provides a comprehensive set of standard computer functionality coupled with industry standard system and communication I/O capability. Realized on a compact, standardized, mezzanine board the EB42x provides a complete embedded computer core which can be readily integrated into most any application.

The basic functions of this board are to provide high performance data processing capability as well as flexible and comprehensive system and communication I/O. The major elements involved in these processes are: the IXP42x Integrated Processor, the Logic controller which is realized in a field-programmable gate array (FPGA), four high-speed serial data UARTs, an Ethernet controller, a CAN controller, two system and one communication interfacing connectors. The EB42x provides four system I/Os: I²C, LPC, a PCI-bus, and a serial terminal and console. In addition, there are various onboard memory elements available: SDRAM, SRAM, FLASH, and EEPROM as well as a provision for accessing off-board CompactFlash. For test and programming purposes there is also a JTAG/BDM interface available.

The following table provides a quick overview of the EB42x board.

Table 1-1: EB42x Product Overview

EB42x FEATURES	DESCRIPTION
Product Type	High performance ARM instruction set embedded computer core: • Processor: Intel XScale™ IXP425 / IXP420 • Memory: SDRAM, SRAM (optional), FLASH, EEPROM • Multiple system and communication I/Os • Form factor: E²Brain™ standard: 115 x 75 x 11.6 mm • Complies with the E²Brain™ specification
I/Os	System: • I²C • LPC • PCI bus • Serial: terminal and console • CompactFlash Communication: • Two high-speed serial UARTs (EB425 only) • Two high-speed synchronous serial interfaces (EB425 only) • CAN (optional, EB425 only) • Two Fast Ethernet channels
Other	Test and Programming: • JTAG/BDM (Background Debug Mode) Monitor and Control: • Reset • GPIOs (General Purpose IOs) • Switches • Temperature sensing

1.4 Board Overview

1.4.1 Board Introduction

The EB42x is a high performance mezzanine computer module compliant to the Kontron Modular Computers' E²Brain™ Module Specification, revision 01.

This E²Brain™ module is comprised of a computer core, the Intel XScale™ IXP42x Integrated Processor, and a standard set of E²Brain™ system and communication interfaces as well as onboard memory.

The computer core provides direct interfacing for the system interfaces: I²C, PCI, terminal and console, and test and programming. The remaining memory, system, and communication interfaces are realized using the core's memory controller interfaces such as SDRAM, ROM/ Flash, extension bus, MII and high-speed serial interface) in conjunction with corresponding interfacing and BPCC (Board Process/Communications Controller) devices (CAN, UARTs, Ethernet-PHY, FPGA).

1.4.2 Board Specific Information

Major board components of the EB42x E²Brain™ module are:

- Computer core: Intel XScale™ IXP425 / IXP420 Integrated Processor (ARM instruction set compatible)
- Board Process/Communications Controller (BPCC, realized in an FPGA)
- Soldered SDRAM devices
- Soldered FLASH device
- Buffered SRAM device (optional)
- Serial EEPROM device
- Two Fast Ethernet channels
- Two high-speed serial UARTs (not available on the EB420)
- Two high-speed synchronous serial interfaces (not available on the EB420)
- CAN controller (optional, not available on the EB420)
- Reset controller
- RTC
- Temperature sensor: LM75 (optional)
- GoldCap (SRAM and RTC backup)
- Two System Interface connectors
- Two Communication Interface connectors

27854.02.VC.060131/114641

1.5 System Relevant Information

The following system relevant information is general in nature but should still be considered when developing applications using the EB42x.

Table 1-2: System Relevant Information

SUBJECT	INFORMATION
System Configuration	The EB42x operates with a PCI system clock frequency of 33/66 MHz. The number of EB42xs which can be installed in any one system depends solely on the number of carrier interfaces available.
Master/Slave Functionality	The EB42x can function either as a PCI Master or PCI Agent.
System Controller	The EB42x can function as a system controller.
Application Interfacing	The application interfacing to the EB42x must comply with the specifications set forth in this manual.

1.5.1 System Configuration

System configuration is solely a function of the application, however, when implementing applications, precautions must be taken to ensure that the signals of the EB42x are properly terminated in accordance with the specifications set forth in this manual. For this reason it will be necessary for system integrators to ensure proper signal conditioning for their applications before interfacing with the EB42x. In addition, it is imperative that signal interference be kept to a minimum. Refer to chapters 4 and 5 for further information.

1.5.2 Operating Software

The EB42x is supplied with appropriate operating system and board support software for board operation.

1.6 Board Diagrams

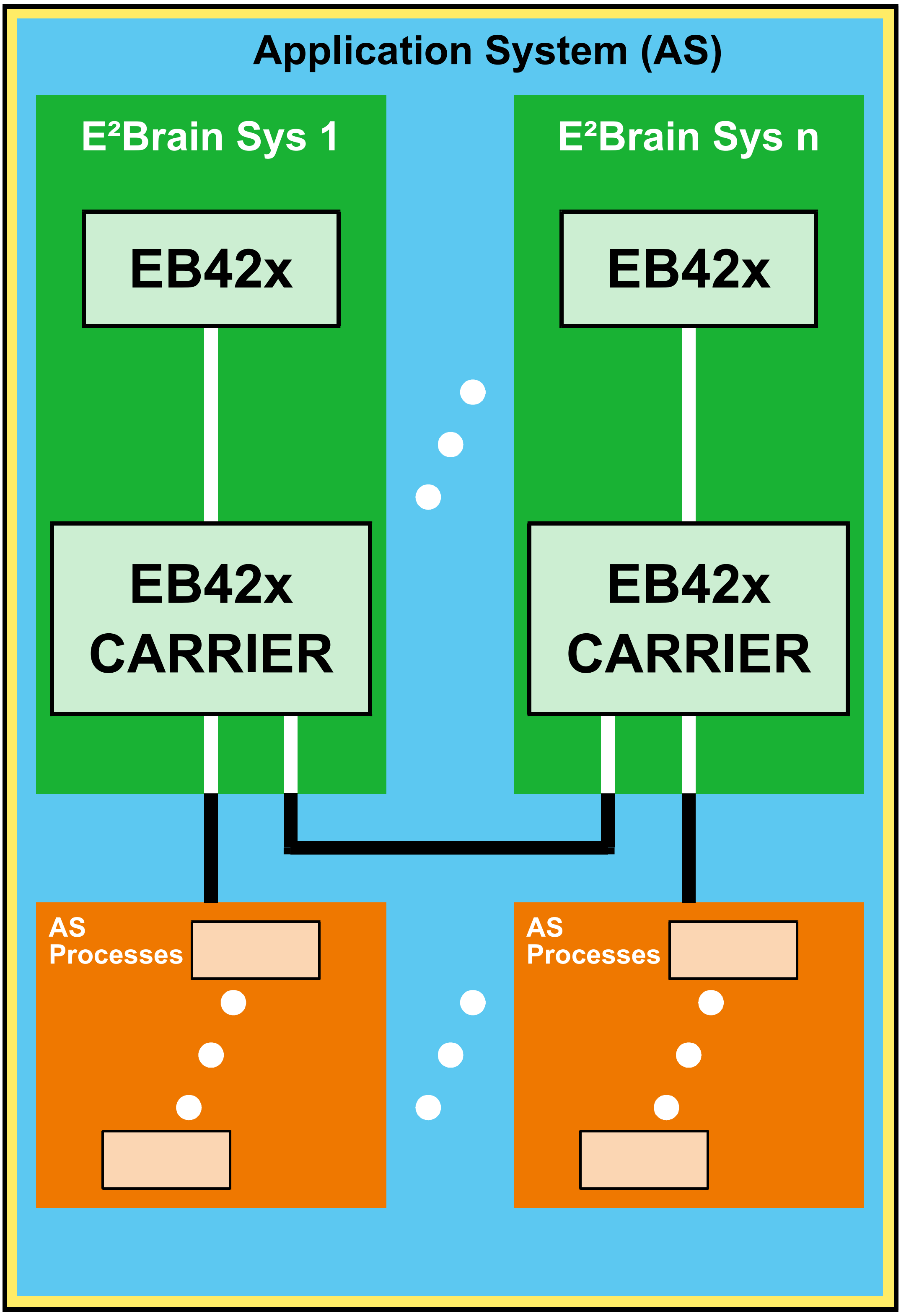
The following diagrams provide additional information concerning board functionality and component layout.

LEGEND FOR FIGURE 1-4:

- **CAN** Communications Area Network
- **CF** CompactFlash
- **FE** Fast Ethernet
- **HSS** High-Speed Serial
- **I²C** Inter-Integrated Circuit
- **LPC** Low Pin Count
- **M/C** Monitor and Control
- **PCI** Peripheral Component Interface
- **T/C** Terminal/Console (Serial Interface)
- **T/P** Test/Programming

1.6.1 Application System Interfacing

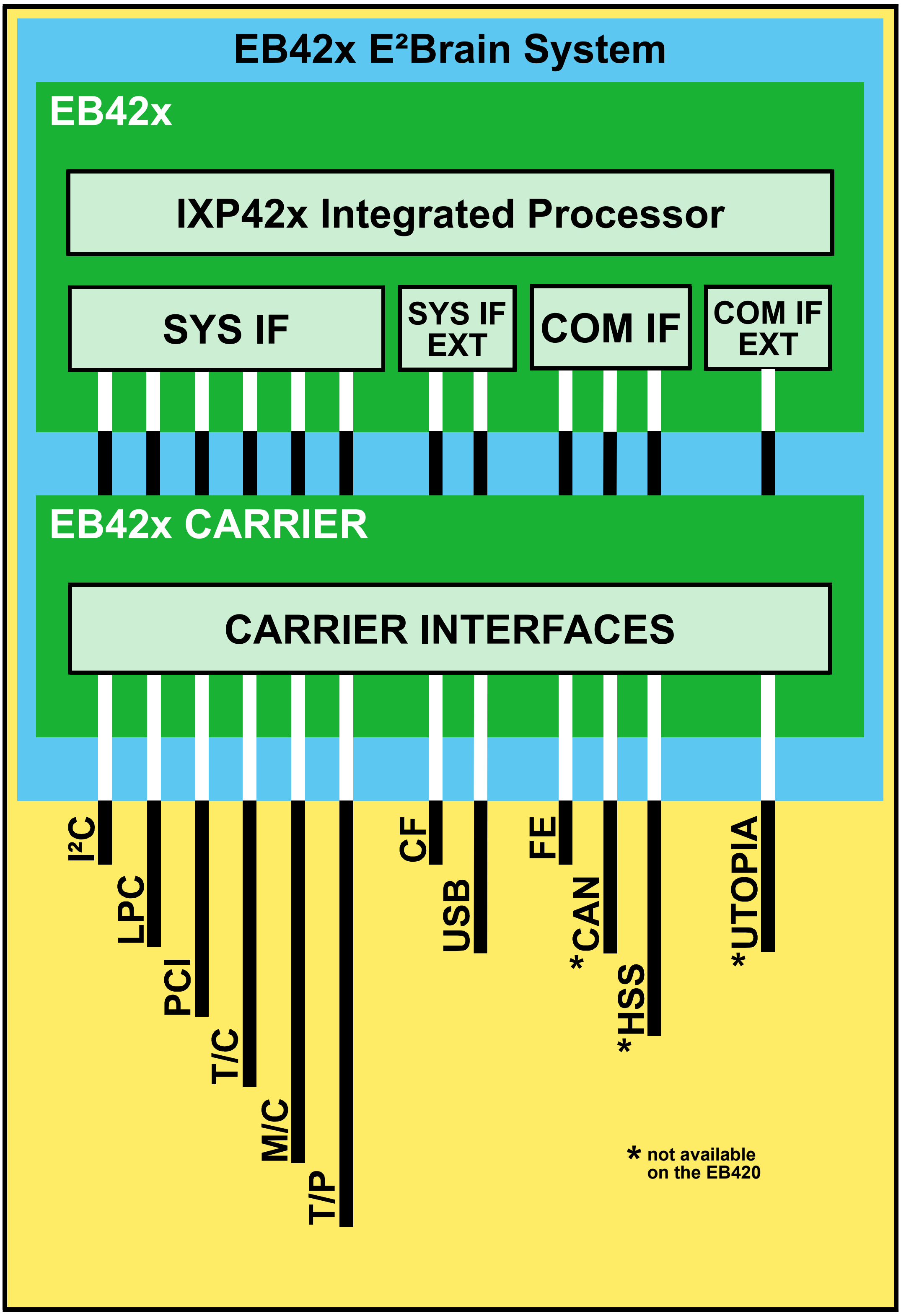
Figure 1-3: EB42x Application System Interfacing Diagram



27854.02.VC.060131/114641

1.6.2 System Level Interfacing

Figure 1-4: EB42x System Level Interfacing Diagram



1.6.3 Board Layout

Figure 1-5: EB42x Board (Top View)

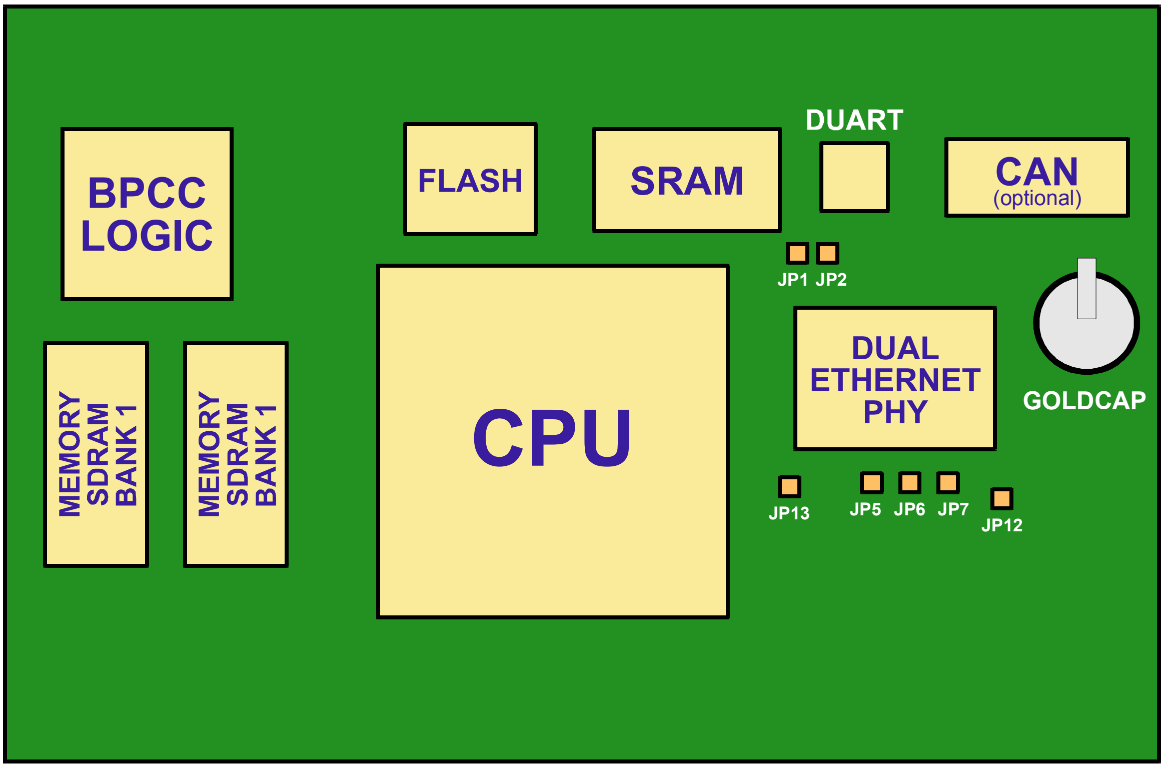
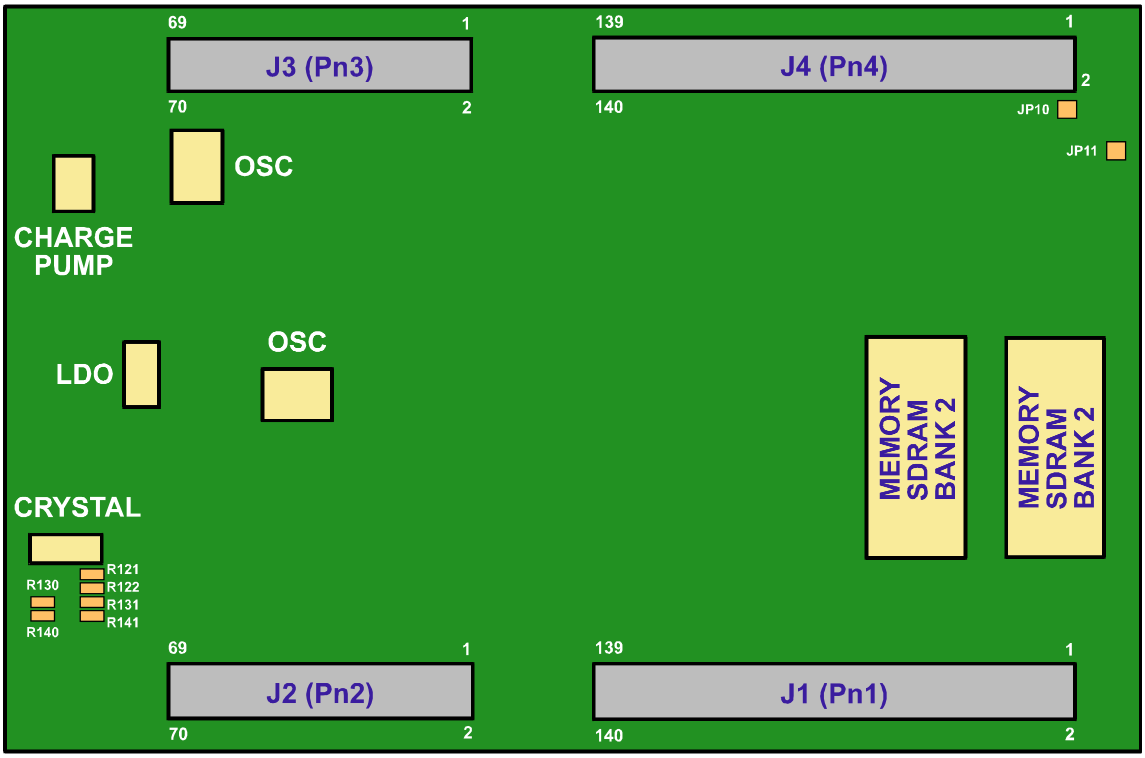


Figure 1-6: EB42x Board (Bottom View)



27854.02.VC.060131/114641

1.7 Technical Specifications

Table 1-3: EB42x Main Specifications

	EB42x	SPECIFICATIONS
Processor and Related	Processor	Intel XScale™ Integrated Processor IXP42x (ARM instruction set compatible)
	Cache Structure	32 kB instruction cache and 32 kB data cache
	Memory Controller	On-chip: SDRAM, expansion bus
	Main Memory	SDRAM soldered: up to 256 MB, 32-bit, 133 MHz
	Watchdog	Watchdog generates exception condition: system reset, interrupt, or cascading
	RTC	Real-time clock, optionally backed up using GoldCap with the data retention being typically about 6 to 7 days or via auxiliary power
	Temperature Sensor	LM75 (optional)
	BPCC	Board Process/Communication Controller: controls expansion bus interfacing, LPC interface, monitor and control functions
Peripheral Memory	SRAM (optional)	1 MB soldered, 512k x 16, optionally backed up using GoldCap with the data retention being typically about 6 to 7 days or via auxiliary power
	EEPROM (I²C)	64 kbit soldered, serial access
	Flash	Minimum of 4 MB and up to 32 MB soldered, paged, 16 MB pages
External/Internal Interfaces	EXTERNAL INTERFACES	
	PCI	On-chip controller 32-bit, 33/66 MHz, PCI System Master and PCI Agent modes
	LPC	Low Pin Count sub-set, 8-bit IO and memory space access, realized in BPCC
	I²C	On-chip, message interface, full master/slave
	CF	CompactFlash, True IDE mode
	Fast Ethernet	10baseT, 100baseTX integrated in the IXP42x processor
	CAN (optional)	Controller Area Network interface (Philips SJA1000) (not available on the EB420)
	Serial Ports	Terminal and Console: Two serial ports: TTL: TERM (Terminal); TxD1/RxD1 TTL: SER0 (Console); TxD2/RxD2 High-Speed Serial: 2 UART ports, full modem : SER1 (Support for RS485 direction control) : SER2 (not available on the EB420) 2 synchronous ports: : HSS0 (not available on the EB420) : HSS1
T/P Interfaces	JTAG/Debug	JTAG/Emulator interface for programming and testing purposes

27854.02.VC.060131/114642

Table 1-3: EB42x Main Specifications (Continued)

	EB42x	SPECIFICATIONS
M/C Interfaces	Inputs	Seven system/application control inputs available Three inputs available for general purpose (with interrupt capability)
	Outputs	Five system/application monitor outputs available Four outputs available for application use
General	Mechanical	Conforms with Kontron Modular Computers' "E²Brain™ Module Specification", revision 01
	Power Consumption	Source: 3.3 V: consumption: 2.8 W
	Temperature Range	Operational: 0°C to +70°C Standard -40°C to +85°C E2 Storage: -55°C to +125°C
	Climatic Humidity	93% RH at 40°C, non-condensing (acc. to IEC 60068-2-78)
	Dimensions	115 mm L x 75 mm W x nn.n mm H (where H is function of application)
	Board Weight	58 g

1.8 Applied Standards

The Kontron Modular Computers' E²Brain™ modules comply with the requirements of the following standards:

Table 1-4: Applied Standards

	TYPE	STANDARD
CE	Emission	EN55022, EN61000-6-3
	Immunity, Industrial Environment	EN61000-6-2
	Immunity, IT Equipment	EN55024, EN61000-6-2
	Electrical Safety	EN60950-1 Note: The EB42x is specified I/O only for: SELV and EVL. It is **NOT SPECIFIED** for "Hazardous"
MECHANICAL	Mechanical Dimensions	E²Brain™ Module Specification, Rev. 01
ENVIRONMENTAL TESTS	Vibration, Sinusoidal	IEC 60068-2-6
	Random Vibration, Broadband	IEC 60068-2-64
	Bump	IEC 60068-2-29
	Single Shock	IEC 60068-2-27
	Temperature Tests A: Cold	IEC 60068-2-1
	Temperature Tests B: Dry Heat	IEC 60068-2-2
	Climatic Humidity	IEC 60068-2-78

1.9 Related Publications

Table 1-5: Related Publications

	ISSUED BY	DOCUMENT
PCI	PCI-SIG	PCI Local Bus Specification, R.2.2
LPC	Intel®	Intel® Low Pin Count (LPC) Interface Specification, Rev. 1.1
I²C	Philips	I2C-BUS SPECIFICATION, Rev. 2.1
E²Brain	Kontron Modular Computers	E²Brain™ Module Specification, Rev. 01

This page has been intentionally left blank.

27854.02.VC.060131/114642

Chapter 2

Functional Description

This page has been intentionally left blank.

2. Functional Description

The following chapters present more detailed, board level information about the EB42x E²Brain™ High Performance XScale™ Processor Module whereby the board components and their basic functionality are discussed in general.

2.1 General Information

The EB42x is comprised basically of the following:

- Intel IXP42x Integrated Processor
  - XScale™ processor core (compliant with StrongARM version 5TE)
  - Peripheral logic
    - Peripheral logic bus
    - Memory Interface for SDRAM, ROM/FLASH, and expansion bus
    - PCI interface
    - DMA controller
    - Dual UART serial interface
    - Debug and JTAG/COP interfaces
- Board Process/Communication Controller (BPCC)
  - Realized in an FPGA device
  - Controls expansion bus interfacing (FLASH, CAN (optional), high-speed serial, CompactFlash)
  - Provides LPC interface
  - Provides I²C interface
  - Provides system monitor and control functions
- System interfaces for:
  - I²C bus
  - LPC bus
  - PCI bus
  - CompactFlash (CF) interface
  - Terminal and Console (T/C) serial interfacing
  - Monitor and Control (M/C) interfacing
  - Test and Programming (T/P) interfacing
  - USB device port
- Communication interfaces for:
  - High-Speed Serial (HSS) communications (not available on the EB420)
  - Communications Area Network (CAN) bus (optional, not available on the EB420)
  - Fast Ethernet (FE)
- Memory
  - Main memory interfaced through soldered SDRAM
  - Soldered SRAM (optional), backed-up using a GoldCap or auxiliary power line
  - Soldered FLASH
  - Serial EEPROM
- Monitor and Control
  - Interfacing for LEDs, operator switches, system monitor/control signals (Reset, FAL, etc.)

  - RTC
  - Temperature sensing
  - Watchdog timer
  - Clock generation
  - Reset generation
  - Board control registers
- Test and Programming
  - Debugging interface
  - JTAG/COP interface
- Software
  - Operating system
  - Board support package
  - Bootstrap loader (NetBootLoader)

2.1.1 Intel IXP42x Integrated Processor

The EB42x is based on Intel's XScale™ processor IXP42x which integrates a large number of peripherals, such as a PCI interface, PCI arbiter, interrupt controller, memory and DMA controller, multiple timers, and a DUART.

Important features of the IXP42x are as follows:

- XScale™ core (compliant with Strong ARM architecture):
  - High performance processor based on Intel XScale™ Microarchitecture
  - Seven/eight-stage Intel Super-Pipelined RISC Technology
  - Management unit
    - 32-entry, data memory management unit
    - 32-entry, instruction memory management unit
    - 32-kB, 32-way, set associative instruction cache
    - 32-kB, 32-way, set associative data cache
    - 2-kB, two-way, set associative mini-data cache
    - 128-entry, branch target buffer
    - Eight-entry write buffer
    - Four-entry fill and pend buffers
  - Clock speeds: up to 533 MHz
  - StrongARM Version 5TE Compliant
  - Intel Media Processing Technology: multiply-accumulate coprocessor
  - Debug unit (accessible through JTAG port)
- Three network processor engines (NPEs), which used to off load typical Layer-2 networking functions such as:
  - Ethernet filtering
  - ATM SARing
  - HDLC
- PCI interface
  - 32-bit interface
  - PCI Local Bus Specification, revision 2.2 compatible
  - Host/option capable
  - Master/target capable

27854.02.VC.060131/114643

- 2-MII/RMII interfaces
- UTOPIA-2 interface (not available on the EB420)
  - Eight-bit interface
  - Up to 33 MHz clock speed
- USB v 1.1 device controller
- Two high-speed serial interfaces (not available on the EB420)
  - Six-wire
  - Supports speeds up to 8.192 MHz
  - Supports connection to T1/E1 framers
  - Supports connection to CODEC/SLICs
  - Eight HDLC channels
- SDRAM interface
  - 32-bit data
  - 13-bit address
  - 133 MHz
- Expansion interface
  - 24-bit address
  - 16-bit data
  - Eight programmable chip selects
- Encryption/Authentication (not available on the EB420)
  - DES
  - DES 3
  - AES 128-bit and 256-bit
- Terminal UART
  - 1,200 Baud to 921 kBaud
  - 16550 compliant
  - 64-byte Tx and Rx FIFOs
- Console UART
  - 1,200 Baud to 921 kBaud
  - 16550 compliant
  - 64-byte Tx and Rx FIFOs
- Internal bus performance monitoring unit
  - Seven 27-bit event counters
  - Monitoring of internal bus occurrences and duration events
- Four internal timers

2.1.2 Board Process/Communication Controller (BPCC)

The BPCC provides extensive interfacing and monitor/control functionality for the EB42x. It provides control and addressing for the expansion bus devices, control of the CPU configuration and board, and an onboard register set. The LPC interface, the I²C interface, and the watchdog timer are also realized in the BPCC.

2.1.3 System Interfacing

The EB42x E²Brain™ module is supplied with a comprehensive set of system interfacing capabilities. The standard set of system interfaces is routed through the System Interface connector J1 (Pn1). An extended set of system interfacing is routed through the System Interface Extension connector J2 (Pn2).

The System Interface connector J1 provides interfacing for the following:

- I²C bus
- LPC bus
- PCI bus
- Terminal and Console (T/C) serial interfacing (TERM and SER0)
- Monitor and Control (M/C) interfacing
- Test and Programming (T/P) interfacing

The System Interface Extension connector J2 provides interfacing for the following:

- CompactFlash (CF) interface
- USB v 1.1 device interfacing

2.1.4 Communication Interfacing

The EB42x E²Brain™ module is also supplied with a comprehensive set of communication interfacing capabilities which are routed through the Communication Interface connectors J3 (Pn3) and J4 (Pn4).

The Communication Interface connector J3 provides interfacing for the following:

- Asynchronous High-Speed Serial (HSS) interfacing (TTL level) (SER1, SER2) *
- Synchronous High-Speed Serial (HSS) interfacing (TTL level) (HSS0, HSS1) *
- CAN bus (TTL level, optional) *
- Fast Ethernet (MDI interface)

* not available on the EB420

The Communication Interface extension connector J4 provides interfacing for the UTOPIA-2 interface. This interface is not available on the EB420.

2.1.5 Memory

Main memory for the EB42x is provided by SDRAM up to 256 MB. One MB of GoldCap backed-up optional SRAM is provided for more permanent storage of application and system data. Up to 32 MB of soldered FLASH is available for ROMable operating systems and boot strap loaders. Finally, there is a 64 kbit (8 x 8 kbit) serial EEPROM connected to the I²C bus for system use.

2.1.6 Monitor and Control (M/C)

Various monitor and control functions are available for use with the EB42x E²Brain™ module. Twelve M/C signals are available on the System Interface for application usage. In addition, the EB42x provides a RTC, a watchdog timer, a digital temperature sensor, clock generators, a reset controller, and variety of board control registers.

2.1.7 Test and Programming

The EB42x supports the comprehensive set of XScale™ debugging via JTAG functionality. Interfacing for this functionality is available on the System Interface.

27854.02.VC.060131/114643

2.1.8 Software

The EB42x is supported by various operating systems. In addition, board support packages are available as well as the “NetBootLoader” bootstrap loader.

2.2 Board-Level Interfacing Diagram

The following figure demonstrates the interfacing structure between the internal processing modules of the EB42x and other major EB42x module components. Where EB42x system elements have common interfacing they are grouped into a block. Interfacing common to only one element of a block is indicated with a direct connecting line. The interfacing lines are shown in white where they are on board and in black for board external interfacing.

LEGEND FOR FIGURE 2-1

BPCC Board Process/Communication Controller

CAN Communications Area Network

CF CompactFlash

CI Communication Interface (J3)

CIE Communication Interface Extension (J4)

FE Fast Ethernet

FEC Fast Ethernet Controller

AHSS Asynchronous High-Speed Serial

SHSS Synchronous High-Speed Serial

I²C Inter-Integrated Circuit

LPC Low Pin Count

MEM IF Memory Interface

PCI Peripheral Component Interface

RTC Real-Time Clock

SI System Interface (J1)

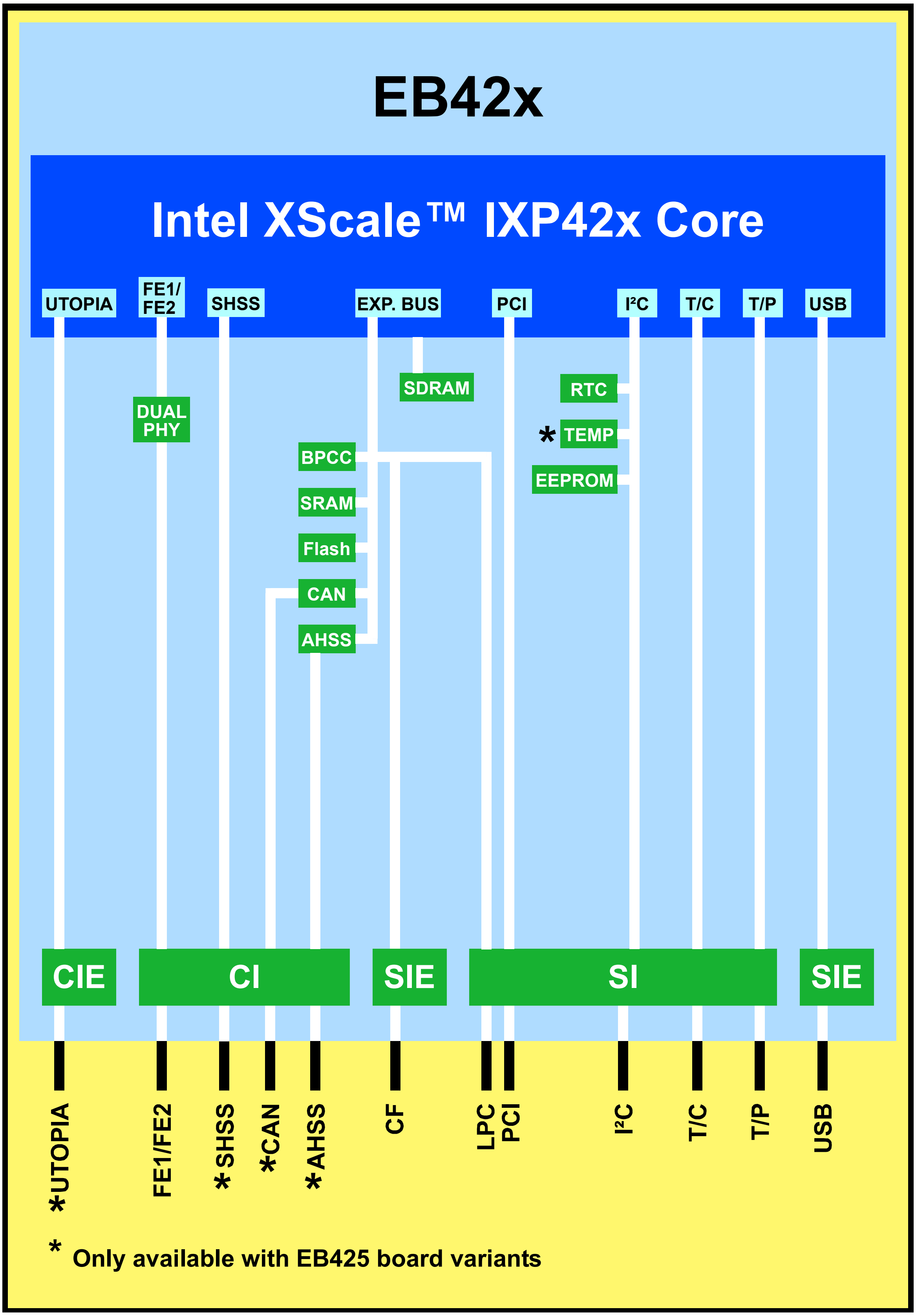
SIE System Interface Extension (J2)

T/C Terminal/Console (Serial Interface)

T/P Test/Programming

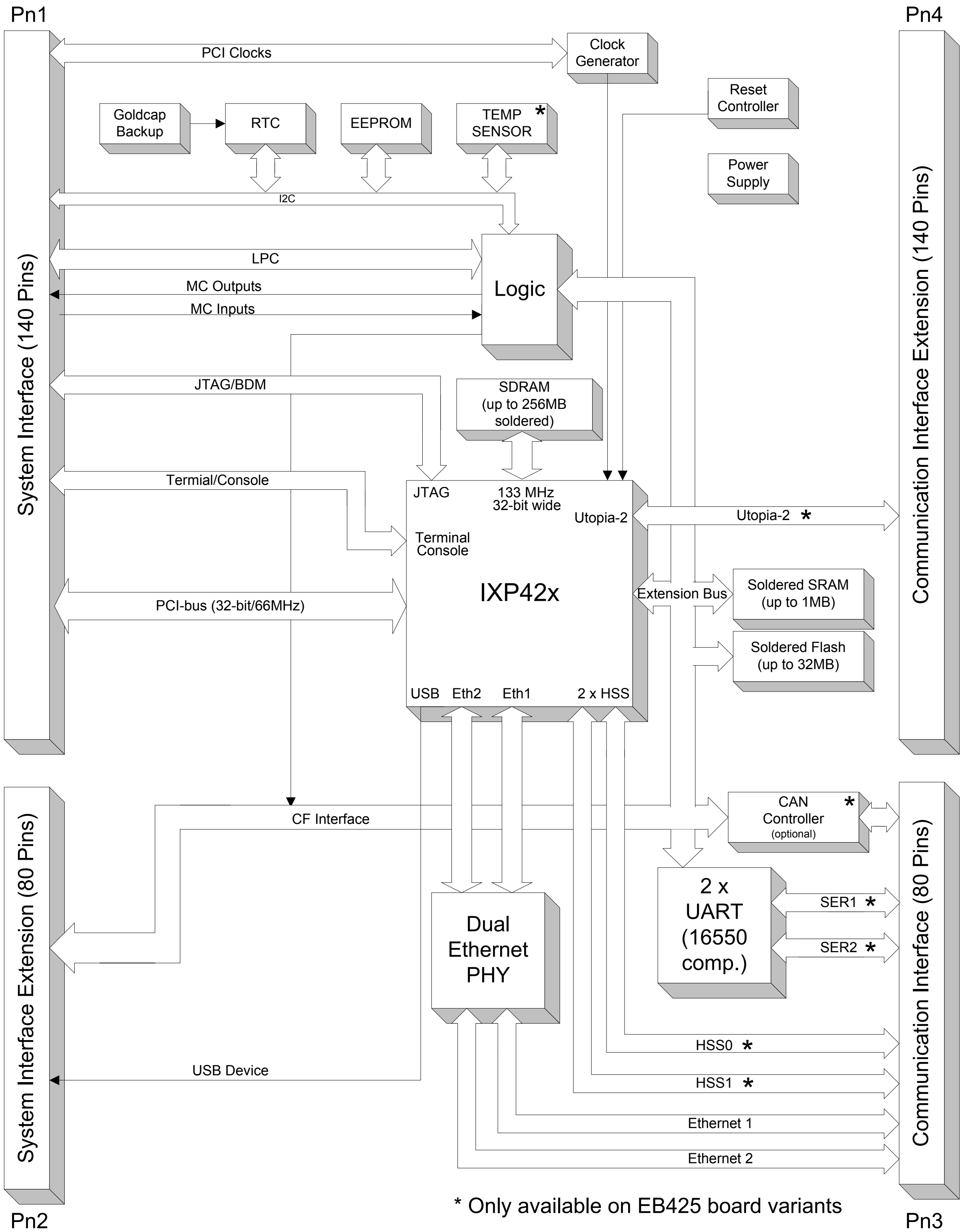
TEMP Digital Temperature Sensor

Figure 2-1: EB42x Board Level Interfacing



27854.02.VC.060131/114643

Figure 2-2: EB42x Functional Block Diagram



27854.02.VC.060131/114644

2.3 System Interfaces

As the name implies, this interface provides the basic application connection functionality required to integrate the EB42x E²Brain™ module either as a high performance core or as a dedicated, special purpose subsystem within comprehensive data processing and handling system.

The System Interface is realized using a 140-pin, HIROSE FX8C-140P-SV connector designated as J1 (Pn1) which is designed to mate with a 140-pin, HIROSE FX8C-140S-SV connector on the EB42x carrier board. The following table provides an overview of the signal types and a brief description of the interfacing realized on this connector. The ensuing sections provide more detailed information concerning the signal specification for this interface.

Table 2-1: System Interface J1 (Pn1) Signal Description

SIGNAL TYPE	DESCRIPTION
POWER	EB42x E²Brain™ module input power, grounds, battery backup power, PCI signaling voltage V(I/O)
MONITOR AND CONTROL (M/C)	Control signals for E²Brain™ module operation, configuration, and additional GPIO interfacing
TEST AND PROGRAMMING (T/P)	JTAG/DEBUG signals and emulator interfacing
TERMINAL AND CONSOLE (T/C)	Two serial interfaces: one for connecting a terminal and one for use as a console interface
I²C	One I²C standard interface for low speed, serial, inter-chip communications
LPC	One LPC standard interface for (GP) IOs and simple memory interfacing
PCI	One PCI standard interface for PCI bus interfacing

Because the E²Brain™ specification defines signal interfacing at the physical component level, the actual electrical characteristics of signals are for the most part different from those specified using accepted industry standards which apply more to unit-to-unit level signals. Only in those cases where the industry standard for such signals is at the physical component level are the characteristics of the signals specified compliant with the standard indicated, for example: PCI compliant signals.

The following table provides signal pinouts along with information concerning the signal characteristics for connector J1 (Pn1) of EB42x E²Brain™ module.

27854.02.VC.060131/114644

Table 2-2: Pinout of J1 (Pn1) Connector

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
1	GND	I	POWER	
2	AUX-Power	I	POWER	Used to supply power to RTC and SRAM
3	SDA	I/O	I²C	Internal pull-up: 1kΩ
4	SCL	I/O	I²C (n.av.)	Internal pull-up: 1kΩ
5	MC6	O	M/C	High, if Control Register 0 = 0; low, if Control Register 0 = 1; can be used to drive a LED; this signal toggles during bootloader startup
6	+3.3V	I	POWER	
7	MC0#	I	M/C	Debounced Reset input, active low
8	MC7	O	M/C	Used for Watchdog activity LED, low if WD enabled
9	MC2#	I	M/C	Internal pull-up: 8.2kΩ
10	MC1#	I	M/C	Internal pull-up: 8.2kΩ; latched in Event-Register, ABORT functionality during bootloader startup
11	+3.3V	I	POWER	
12	GND	I	POWER	
13	MC3#	I	M/C	Internal pull-up: 8.2kΩ; interrupt capable
14	MC11	O	M/C	High, if Control Register 1 = 0; low, if Control Register 1 = 1; can be used to drive a LED
15	MC4	I	M/C	Internal pull-up, this pin must be left unconnected
16	MC5#	I	M/C	This pin requires an external pull-up: 10kΩ; interrupt capable
17	MC9	O	M/C	Open collector: driven low if Control Register 3 = 0; high impedance if Control Register 3 = 1
18	MC8	I	M/C	Internal pull-up: 8.2kΩ; leave this signal open if used in Standalone/ Master configuration; connect to GND to force Slave mode
19	GND	I	POWER	
20	MC10	O	M/C	Open collector: driven low if Control Register 5 = 0; high impedance if Control Register 5 = 1
21	LPCCLK	O	LPC	
22	RESERVED			This pin must be left unconnected
23	LAD0	I/O	LPC	Internal pull-up: 8.2kΩ
24	LAD1	I/O	LPC	Internal pull-up: 8.2kΩ
25	LAD2	I/O	LPC	Internal pull-up: 8.2kΩ
26	+3.3V	I	POWER	
27	LFRAME#	O	LPC	
28	GND	I	POWER	
29	SERIRQ	I/O	LPC	Internal pull-up: 10kΩ
30	LAD3	I/O	LPC	Internal pull-up: 8.2kΩ

Table 2-2: Pinout of J1 (Pn1) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
31	TXD1	O	T/C	TTL level (signal level operational range: min. -0.3V, max. +3.6V) Note: Signal levels outside the above range may cause equipment damage.
32	NC		LPC	This pin is not connected internally
33	+3.3V	I	POWER	
34	TXD2	O	T/C	TTL level (signal level operational range: min. -0.3V, max. +3.6V) Note: Signal levels outside the above range may cause equipment damage.
35	GND	I	POWER	
36	RXD2	I	T/C	Internal pull-up: 10kΩ, TTL level (signal level operational range: min. -0.3V, max. +3.6V) Note: Signal levels outside the above range may cause equipment damage.
37	RXD1	I	T/C	Internal pull-up: 10kΩ, TTL level (signal level operational range: min. -0.3V, max. +3.6V) Note: Signal levels outside the above range may cause equipment damage.
38	NC		T/P	This pin is not connected internally
39	EMU_VCC	O	T/P	Provides the suitable IO voltage to an Emulation probe (3.3V)
40	EMU_SRST	I	T/P	Internal pull-up: 1kΩ
41	NC		T/P	This pin is not connected internally
42	GND	I	POWER	
43	RESERVED			This pin must be left unconnected
44	TMS	I	T/P	Internal pull-up: 10kΩ
45	TRST	I	T/P	Internal pull-down: 10kΩ
46	+3.3V	I	POWER	
47	TCK	I	T/P	Internal pull-up: 10kΩ
48	TDO	O	T/P	
49	RESERVED			This pin must be left unconnected
50	TDI	I	T/P	Internal pull-up: 10kΩ
51	GND	I	POWER	
52	RESERVED			This pin must be left unconnected
53	PCI-CLK-OUT-0	O	PCI	
54	V(I/O)	I	POWER	This pin defines the PCI Signaling Voltage. This pin must be connected to 3.3V
55	+3.3V	I	POWER	
56	PCI-CLK-IN	I	PCI	Clock input used in Agent mode; leave unconnected in Master mode

27854.02.VC.060131/114644

Table 2-2: Pinout of J1 (Pn1) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
57	PCI-CLK-OUT-1	O	PCI	
58	GND	I	I	
59	PCI-RST#	O	PCI	Can also be used to reset other carrier devices
60	PCI-CLK-OUT-2	O	PCI	
61	INTA#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
62	INTC#	I	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
63	INTB#	I	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
64	GNT#0	I/O	PCI	
65	INTD#	I	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
66	+3.3V	I	POWER	
67	GND	I		
68	GNT#1	O	PCI	
69	AD31	I/O	PCI	
70	GNT#2	O	PCI	
71	AD29	I/O	PCI	
72	REQ#0	I/O	PCI	This pin has an internal pull-up when used in Master mode. The value of the internal pull-up resistor is not guaranteed, but is sufficient to prevent unused inputs from floating. Agent mode: no pull-up
73	AD27	I/O	PCI	
74	REQ#1	I	PCI	This pin has an internal pull-up which is enabled at all times. The value of the internal pull-up resistor is not guaranteed, but is sufficient to prevent unused inputs from floating.
75	+3.3V	I	POWER	
76	GND	I	POWER	
77	AD25	I/O	PCI	
78	REQ#2	I	PCI	This pin has an internal pull-up which is enabled at all times. The value of the internal pull-up resistor is not guaranteed, but is sufficient to prevent unused inputs from floating.
79	C/BE3#	I/O	PCI	
80	AD30	I/O	PCI	
81	AD23	I/O	PCI	
82	AD28	I/O	PCI	
83	GND	I	POWER	
84	AD26	I/O	PCI	
85	AD21	I/O	PCI	
86	AD24	I/O	PCI	

27854.02.VC.060131/114645

Table 2-2: Pinout of J1 (Pn1) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
87	V(I/O)	I	POWER	This pin defines the PCI Signaling Voltage. This pin must be connected to 3.3V.
88	+3.3V	I	POWER	
89	AD19	I/O	PCI	
90	IDSEL	I/O	PCI	Leave unconnected in Master mode; IDSEL input in Agent mode
91	AD17	I/O	PCI	
92	GND	I	POWER	
93	C/BE2#	I/O	PCI	
94	AD22	I/O	PCI	
95	+3.3V	I	POWER	
96	AD20	I/O	PCI	
97	IRDY#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
98	AD18	I/O	PCI	
99	DEVSEL#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
100	AD16	I/O	PCI	
101	GND	I	POWER	
102	FRAME#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
103	PCI-X-CAP	I/O	PCI	Place a pull-down resistor of 100 Ω or less on the PCI-X-CAP pin
104	GND	I	POWER	
105	LOCK#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
106	TRDY#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
107	PERR#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
108	STOP#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
109	SERR#	I/O	PCI	Master mode: internal pull-up: 8.2kΩ; Agent mode: no pull-up
110	+3.3V	I	POWER	
111	GND	I	POWER	
112	PAR	I/O	PCI	
113	C/BE1#	I/O	PCI	
114	AD15	I/O	PCI	
115	AD14	I/O	PCI	
116	V(I/O)	I	POWER	This pin defines the PCI Signaling Voltage. This pin must be connected to 3.3V.
117	AD12	I/O	PCI	
118	AD13	I/O	PCI	
119	AD10	I/O	PCI	

27854.02.VC.060131/114645

Table 2-2: Pinout of J1 (Pn1) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
120	AD11	I/O	PCI	
121	+3.3V	I	POWER	
122	AD9	I/O	PCI	
123	M66EN	I/O	PCI	Place a pull-down resistor of 100 Ω or less on the M66EN pin
124	GND	I	POWER	
125	AD8	I/O	PCI	
126	C/BE0#	I/O	PCI	
127	AD7	I/O	PCI	
128	AD6	I/O	PCI	
129	AD5	I/O	PCI	
130	+3.3V	I	POWER	
131	GND	I	POWER	
132	AD4	I/O	PCI	
133	AD3	I/O	PCI	
134	AD2	I/O	PCI	
135	AD1	I/O	PCI	
136	AD0	I/O	PCI	
137	ACK64#	I/O	PCI	Master mode: internal pull-up 10kΩ; Agent mode: no pull-up, signal not used on this board
138	REQ64#	I/O	PCI	Master mode: internal pull-up: 10kΩ; Agent mode: no pull-up, signal not used on this board
139	+3.3V	I	POWER	
140	GND	I	POWER	

2.3.1 Power Interface

For the EB42x E²Brain™ module, a single power supply voltage of 3.3 VDC is specified. The following table summarizes the power specifications.

Table 2-3: EB42x Power Interface Requirements

VOLTAGE	DESCRIPTION
+ 3.3 VDC	Input voltage tolerance: +5% to -3% Supply voltage ripple: 100 mV peak-to-peak; 0 to 20 MHz
GND	Ground voltage reference input
AUX-Power	Optional auxiliary power input for battery backup of CMOS memory devices; input voltage must be =< 3.3V, + 5% tolerance
V(I/O)	PCI signalling voltage selection. Only 3.3 V are allowed according to the IXP42x Hardware Manual.

2.3.2 Monitor and Control Interface

This interface is comprised of a set of twelve IO signals which can be used to facilitate system integration.

The following table provides a listing of Monitor and Control signals along with a brief description.

Table 2-4: Monitor and Control Interface Signal Description

SIGNAL	DESCRIPTION
MC0#	Reset input, debounced (e.g. for push button switch)
MC1#	Input, latched into Event Register (e.g. for ABORT switch)
MC2#	Input, suitable for maskable interrupt operation
MC3#	Input, suitable for maskable interrupt operation
MC4	Input, factory mode, used as boot control signal on the EB42x; low = boot from LPC device; high or open = boot from onboard FLASH
MC5#	Input, suitable for maskable interrupt operation
MC6	Output, LED1, RUN or general purpose LED; can be also used as a general purpose output after the bootloader startup
MC7	Output, WD_LED, low when watchdog timer enabled
MC8	Input, AGENT, low = Agent mode, high = Master mode
MC9	Output, open collector
MC10	Output, open collector
MC11	Output, LED2, general purpose LED; can also be used as a general purpose output

2.3.3 Test and Programming Interface

The Test and Programming interface supports JTAG/DEBUG and ISP operations. This interface can be used for connecting hardware emulators and debuggers (e.g. BDM, COP, etc.) and for “in-system programming” (ISP) of programmable hardware as well as in-system testing (JTAG). It is comprised of a set of six signals whereby some are common to all three interfaces and some are dedicated to only one.

The following table provides a listing of the Test and Programming interface signals and a brief description.

Table 2-5: Test and Programming Interface Signal Description

SIGNAL	DESCRIPTION
TCK	Test Clock in for JTAG/ISP and emulator/debugger
TDI	Test Data In for JTAG/ISP and emulator/debugger
TDO	Test Data Out JTAG/ISP and emulator/debugger
TMS	Test Mode Select, input for JTAG/ISP and emulator/debugger
TRST	Test Reset, input for JTAG/ISP and emulator/debugger
EMU_SRST	Hard Reset, emulator/debugger hard reset In/Out
EMU_VCC	Reference Voltage of the JTAG/DEBUG core

2.3.4 Terminal and Console Interface

The EB42x provides two serial interfaces for supporting a terminal port and a low speed communication interface (console) for firmware updating. These interfaces are realized using the IXP42x on-chip dual UART, and as such provide only a two-wire interface without hardware handshake signals.

The following table provides a listing of the Terminal and Console interface signals and a brief description.

Table 2-6: Terminal and Console Interface Signal Description

SIGNAL	DESCRIPTION
TXD1/2	Serial Transmit Data outputs
RXD1/2	Serial Receive Data signal inputs

Note ...

The corresponding serial signals on the EB42x are TTL logic level signals. Therefore, the transceivers for RS232 must be provided by the carrier board.

Warning!

The signal level on the receive lines must not exceed -0.3V to 3.6V. Voltage undershoot below -0.3V and voltage overshoot over 3.6 may damage the board.

2.3.5 I²C Interface

The EB42x E²Brain™ module provides one I²C serial interface for supporting direct interfacing EB42x and carrier board devices. This interface is two signals wide and fully supports the I²C specification.

The following table provides a listing of the I²C interface signals and a brief description.

Table 2-7: I²C Interface Signal Description

SIGNAL	DESCRIPTION
SCL	Serial Clock Line
SDA	Serial Data line

The onboard I²C devices of the EB42x are:

- RTC (refer to chapter 4.5)
- Temperature sensor (refer to chapter 4.8) (not available on the EB420)
- User EEPROM (refer to chapter 4.7)

Note ...

The I²C interface provided by the EB42x is a “3.3V” interface. If 5V components must be connected to the bus, the level shifters must be provided. For further information refer to the “I²C Bus Specification, Version 2.0, December 1998” (Philips semiconductors).

2.3.6 LPC Interface

One Low Pin Count (LPC) interface for supporting simple IOs, simple static memory devices, and IO controllers is available with the EB42x.

The controller is completely integrated in the BPCC and offers an 8-bit data access port to devices which use LPC IO or memory access protocols. I/O and memory area are selected using different address spaces.

The I/O address space is 64 kByte in size, whereas the memory area offers 16 MByte address space. DMA, however, is not supported by this interface.

In addition, a serial IRQ controller is also implemented in the BPCC, controlling and collecting the serial LPC IRQs and converting and processing them to IRQs for the CPU.

The serial IRQ controller is realized according to the “Serialized IRQ Support for PCI Systems” Specification, Rev. 6.0, Sept. 1, 1995.

The following table provides a listing of the LPC interface signals and a brief description.

Table 2-8: LPC Interface Signal Description

SIGNAL	DESCRIPTION
LAD[0:3]	Multiplexed Command, Address, and Data lines
LFRAME#	Indicates start of a new cycle, termination of broken cycle
LPCCLK	33 MHz clock
SERIRQ	Serialized IRQ, optional for peripherals that need interrupt

2.3.7 PCI Interface

In contrast to all other PCI based modules, the EB42x is capable of both PCI Master mode and PCI Agent mode. PCI Agent mode can be selected by setting the AGENT signal on the System Interface to low.

In PCI Master mode, the EB42x operates as the host, initializing and controlling up to three PCI devices on the carrier, whereas, in the Agent mode, the EB42x itself operates as a PCI target.

The following table identifies the EB42x PCI bus signals and provides a short description of each signal.

Table 2-9: PCI Interface Signal Description

SIGNAL	DESCRIPTION
AD [0:31]	PCI multiplexed address and data bus
INT [A, B, C, D]#	PCI interrupt requests
C/BE [0:3]#	PCI multiplexed bus command and byte enable
IRDY#	Initiator Ready indicates the current bus master is ready to complete the current data phase.
TRDY#	Target Ready indicates the selected device is ready to complete the current data phase.
PCI-RST#	PCI Reset signal, is also used for LPC devices and other devices on the carrier board
PCI-CLK-OUT- [1:3]	PCI clock Outputs for up to 3 external bus mastering PCI devices. All PCI signals except PCI_RST#, and INT [A, B, C, D] # are sampled on the rising edge.
FRAME#	Indicate the beginning and duration of a PCI access.
STOP#	Indicates the target is requesting the master to stop the current transaction
DEVSEL#	Device select generated by the target when cycle refers to its own address.
REQ [0:2]#	PCI Arbiter requests
GNT [0:2]#	PCI Arbiter grants
PAR	Calculated/Checked Parity
PERR#	Parity Error
LOCK#	PCI Lock resource signal
SERR#	System Error
REQ64#	PCI request for a 64-bit access
ACK64#	PCI grant for a 64-bit access
PCI-CLK-IN	PCI clock input, used in agent mode
AGENT	PCI-agent mode logic input, 0 -> PCI agent mode, 1 -> system controller

2.4 System Interface Extension

The System Interface Extension is realized using an 80-pin, HIROSE FX8C-80P-SV connector designated as J2 (Pn2) which is designed to mate with a 80-pin, HIROSE FX8C-80S-SV connector on the EB42x carrier board, and it is used to provide CPU architecture specific system interfaces.

In the case of the EB42x, a CompactFlash interface is made available on the System Interface Extension. The following sections provide further information concerning the CompactFlash and USB interfacing.

2.4.1 CompactFlash Interface

The table provides summary of the CompactFlash signal implemented on the System Interface Extension connector.

Table 2-10: CompactFlash Interface Signal Description

EB42x SIGNAL	IDE SIGNAL	DESCRIPTION
CF_D[0:15]	D0 to D15	CompactFlash data bus – 16-bit wide
CF_CS[0:1]	-CS0 and -CS01	CompactFlash chip select
CF_RD	-IORD	CompactFlash IO read strobe
CF_WR	-IOWR	CompactFlash IO write strobe
CF_RST	-RESET	CompactFlash reset, active low
CF_INTRQ	INTRQ	CompactFlash interrupt request, active high
CF_IORDY	IORDY	CompactFlash IO ready
CF_A[0:2]	A0 to A02	CompactFlash address lines

Note ...

The CompactFlash interface is realized as a true IDE interface (PIO mode). The PC CARD Memory Mode and the PC Card I/O Mode of the Compact Flash Specification are not supported.

In addition, the pinout for the Compact Flash signals have been optimized for routing to a CompactFlash socket.

2.4.2 USB Interface

The table provides summary of the USB signals implemented on the System Interface Extension connector.

Table 2-11: USB Interface Signal Description

SIGNAL	DESCRIPTION
USB_D+	Positive data line of the differential data path
USB_D-	Negative data line of the differential data path

27854.02.VC.060131/114648

The following table provides pinout information for J2 (Pn2) for the CompactFlash and USB interfaces.

Table 2-12: Pinout of J2 (Pn2) Connector

PIN	EB42x SIGNAL	IDE SIGNAL	REMARKS	PIN	EB42x SIGNAL	IDE SIGNAL	REMARKS
1	GND	GND		41	CF_D9	D09	Internal serial resistor 33 Ω
2	NC	NC		42	GND	GND	
3	CF_D3	D03	Internal serial resistor 33 Ω	43	V(I/O)	V(I/O)	Additional V(I/O) pin for PCI signaling; must be connected to 3.3 V
4	CF_D11	D11	Internal serial resistor 33 Ω	44	NC	NC	
5	CF_D4	D04	Internal serial resistor 33 Ω	45	CF_D10	D10	Internal serial resistor 33 Ω
6	+3.3V	+3.3V		46	NC	NC	
7	+3.3V	+3.3V		47	NC	NC	
8	CF_D12	D12	Internal serial resistor 33 Ω	48	GND	GND	
9	CF_D5	D05	Internal serial resistor 33 Ω	49	+3.3V	+3.3V	
10	CF_D13	D13	Internal serial resistor 33 Ω	50	NC	NC	
11	CF_D6	D06	Internal serial resistor 33 Ω	51	NC	NC	
12	GND	GND		52	NC	NC	
13	GND	GND		53	NC	NC	
14	CF_D14	D14	Internal serial resistor 33 Ω	54	GND	GND	
15	CF_D7	D07	Internal serial resistor 33 Ω	55	GND	GND	
16	CF_D15	D15	Internal serial resistor 33 Ω	56	NC	NC	
17	CF_CS0	-CSO	Internal serial resistor 33 Ω	57	NC	NC	
18	+3.3V	+3.3V		58	NC	NC	
19	GND	GND		59	NC	NC	
20	CF_CS1	-CS1	Internal serial resistor 33 Ω	60	+3.3V	+3.3V	
21	CF_RD	-IORD	Internal serial resistor 33 Ω	61	GND	GND	
22	CF_WR	-IOWR	Internal serial resistor 33 Ω	62	NC	NC	
23	CF_INTRQ	INTRQ	Internal serial resistor 33 Ω	63	NC	NC	
24	GND	GND		64	NC	NC	
25	GND	GND		65	NC	NC	
26	CF_RST	-RESET		66	GND	GND	

Table 2-12: Pinout of J2 (Pn2) Connector (Continued)

PIN	EB42x SIGNAL	IDE SIGNAL	REMARKS	PIN	EB42x SIGNAL	IDE SIGNAL	REMARKS
27	CF_IORDY	IORDY	Internal serial resistor 33 Ω	67	NC	NC	
28	CF_A2	A02	Internal serial resistor 33 Ω	68	NC	NC	
29	CF_A1	A01	Internal serial resistor 33 Ω	69	NC	NC	
30	GND	GND		70	NC	NC	
31	+3.3V	+3.3V		71	+3.3V	+3.3V	
32	CF_A0	A00	Internal serial resistor 33 Ω	72	NC	NC	
33	NC	NC		73	NC	NC	
34	CF_D0	D00	Internal serial resistor 33 Ω	74	NC	NC	
35	CF_PDIAG	-PDIAG	Internal pull-up 4.7 kΩ	75	NC	NC	
36	+3.3V	+3.3V		76	GND	GND	
37	GND	GND		77	USB_D+	USB_D+	USB device port signal line
38	CF_D1	D01	Internal serial resistor 33 Ω	78	NC	NC	
39	CF_D8	D08	Internal serial resistor 33 Ω	79	USB_D-	USB_D-	USB device port signal line
40	CF_D2	D02	Internal serial resistor 33 Ω	80	NC	NC	

27854.02.VC.060131/114649

2.5 Communication Interface

The Communication Interface Connector J3 (Pn3), is used to provide a set of standard communication interfaces. In the case of the EB42x, there are four types of interfaces provided: two asynchronous high-speed serial interfaces, two synchronous high-speed serial interfaces, one CAN interface, and two Fast Ethernet interfaces.

All of these interfaces are provided on the Communication Interface J3 (Pn3) connector (an 80-pin, HIROSE FX8C-80P-SV connector) which is designed to mate with a 80-pin, HIROSE FX8C-80S-SV connector on the EB42x carrier board.

The following table provides pinout information for J3 (Pn3).

Table 2-13: Pinout of J3 (Pn3) Connector

PIN	SIGNAL	REMARKS
1	GND	
2	GND	
3	SER1_DSR *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
4	SER1_RI *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
5	SER1_RTS *	
6	SER1_CTS *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
7	SER1_TXD *	
8	SER1_RXD *	Internal pull-up: 100kΩ; input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
9	SER1_DTR *	
10	SER1_CD *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
11	GND	
12	SER2_RI *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
13	SER2_DSR *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
14	SER2_CTS *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
15	SER2_RTS *	
16	SER2_RXD *	Internal pull-up 100kΩ; input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
17	SER2_TXD *	
18	SER2_CD *	Input signal levels must not exceed: – 0.3V (low) or 3.3V (high)
19	SER2_DTR *	
20	HSS0_RXCLK *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
21	HSS0_TXCLK *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
22	GND	
23	HSS0_TXFRAME *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)

27854.02.VC.060131/114649

Table 2-13: Pinout of J3 (Pn3) Connector (Continued)

PIN	SIGNAL	REMARKS
24	HSS0_RXFRAME *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
25	HSS0_TXDATA *	Internal pull-up 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
26	HSS0_RXDATA *	Internal pull-up 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
27	NC	
28	NC	
29	HSS1_TXCLK *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
30	HSS1_RXCLK *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
31	HSS1_TXFRAME *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
32	HSS1_RXFRAME *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
33	HSS1_TXDATA *	Internal pull-up 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
34	HSS1_RXDATA *	Internal pull-down 10kΩ; input signal levels must not exceed: – 0.3V (low) or 3.6V (high)
35	NC	
36	NC	
37	NC	
38	NC	
39	NC	
40	NC	
41	CAN_TXD0 *	
42	CAN_RXD0 *	Internal pull-up 10kΩ; input signal levels must not exceed -0.5V (low) or 5.5V (high)
43	CAN_TXD1 *	
44	CAN_RXD1 *	Internal pull-up 10kΩ; input signal levels must not exceed -0.5V (low) or 5.5V (high)
45	GND	
46	GND	
47	NC	
48	NC	
49	ETH2_ACT/LINK LED	Active low, drives LED from source to sink, drive capacity 10mA
50	NC	

27854.02.VC.060131/114650

Table 2-13: Pinout of J3 (Pn3) Connector (Continued)

PIN	SIGNAL	REMARKS
51	DUP/COL LED	Active low, drives LED from source to sink, drive capacity 10mA
52	ETH2_SPEED LED	Active low, drives LED from source to sink, drive capacity 10mA
53	SD2	Signal detect in fiber mode
54	GND	
55	NC	
56	NC	
57	NC	
58	NC	
59	ETH2_T-	Internal termination, MDI interface, only magnetics required
60	ETH2_R-	Internal termination, MDI interface, only magnetics required
61	ETH2_T+	Internal termination, MDI interface, only magnetics required
62	ETH2_R+	Internal termination, MDI interface, only magnetics required
63	2.5V	
64	2.5V	
65	NC	
66	NC	
67	NC	
68	NC	
69	ETH1_T-	Internal termination, MDI interface, only magnetics required
70	ETH1_R-	Internal termination, MDI interface, only magnetics required
71	ETH1_T+	Internal termination, MDI interface, only magnetics required
72	ETH1_R+	Internal termination, MDI interface, only magnetics required
73	GND	
74	SD1	Signal detect in fiber mode
75	ETH1_ACT/LINK LED	Active low, drives LED from source to sink, drive capacity 10mA
76	NC	
77	DUP/COL LED	Active low, drives LED from source to sink, drive capacity 10mA
78	ETH1_SPEED LED	Active low, drives LED from source to sink, drive capacity 10mA
79	GND	
80	GND	

* All signals marked with an * (asterix) are not available on the EB420.

2.5.1 Asynchronous High-Speed Serial Interfaces

Two, full modem serial ports (SER1 and SER2) are available on some of the EB425 variants but not on the EB420 board variants. Eight signals per port are provided to realize asynchronous high-speed serial links interfaced using dedicated controlling/handshaking. The EB42x uses DUARTs (EXAR XR 16C2850 or XR16L2750) which are 16550 compatible and provide hardware handshaking support for RS485 operation.

Table 2-14: Asynchronous High-Speed Serial Interface Signal Description

SIGNAL	DESCRIPTION
SERx_TXD	Transmit data output
SERx_RXD	Receive data input
SERx_RTS	Request to send output
SERx_CTS	Clear to send input
SERx_DTR	Data terminal ready output
SERx_DSR	Transmit clock for synchronous transmissions/Data set ready input
SERx_CD	Carrier detect input
SERx_RI	Modem ring indicator

Note ...

All signals are available and supplied at 3.3V TTL levels. Further signal conditioning via appropriate transceivers on the carrier board is required to support the respective communication standards.

2.5.2 Synchronous High-Speed Serial Interfaces

Two synchronous high-speed serial ports (HSS0 and HSS1) are available on the EB425 E²Brain™ module but not on the EB420. Six signals per port are provided to realize synchronous high-speed serial links interfaced using dedicated controlling. The EB425 uses the two high-speed synchronous interfaces provided by the IXP425.

Table 2-15: Synchronous High-Speed Serial Interface Signal Description

SIGNAL	DESCRIPTION
HSSx_TXCLK	Synchronous interface transmit clock signals
HSSx_TXFRAME	Synchronous interface transmit frame signals
HSSx_TXD	Synchronous interface transmit data signals
HSSx_RXFRAME	Synchronous interface receive clock signals
HSSx_RXFRAME	Synchronous interface receive frame signals
HSSx_RXD	Synchronous interface receive data signal

Note ...

All signals are available and supplied at 3.3V TTL levels. Further signal conditioning via appropriate transceivers on the carrier board is required to support the respective communication standards.

2.5.3 CAN Interface

To provide field bus support, there are pins available to implement one CAN bus interface (Philips SJA 1000) on the J3 (Pn3) connector. The signals provided are at 5V TTL voltage levels and must be adapted to the CAN bus levels through the use of appropriate CAN transceivers on the carrier board. This interface is optional and only available on the EB425.

Table 2-16: CAN Interface Signal Description

SIGNAL	DESCRIPTION
CAN_TXDx	Transmit data output driver
CAN_RXDx	Receive data input channel

Note ...

For more information concerning the interfacing of the SJA 1000 to the CAN Phy, refer to the Philips Application Note: AN97076 or the data sheet of the PCA 82C250 CAN controller interface.

2.5.4 Ethernet Interfaces

The EB42x modules provide two Fast Ethernet interfaces (Intel IXP42x's internal NPEs) whose signals are converted by using the Intel LXT973 “Dual-Ethernet Phy” interface at copper Ethernet transmission voltage levels. So the carrier board needs to add only the galvanic isolation (magnetics) function and the appropriate transmission connector type.

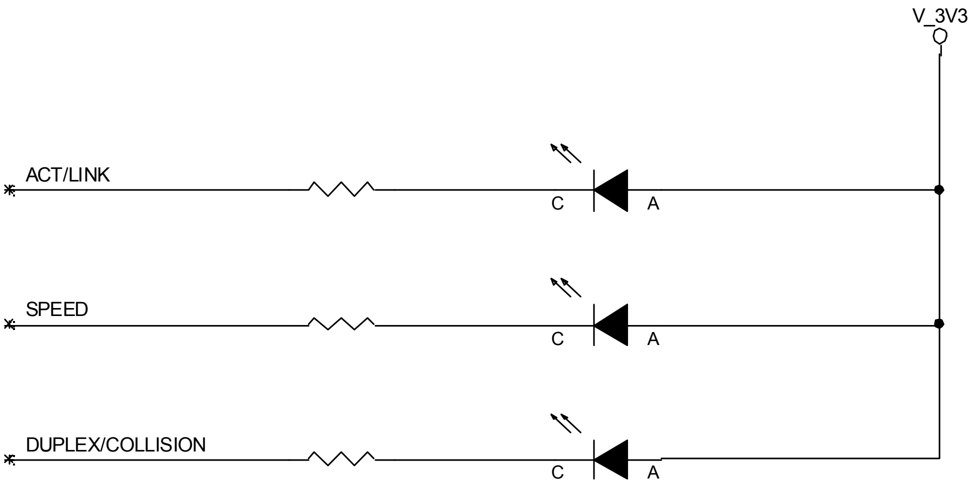
Additionally, for monitoring and control purposes, LED functionality is provided to indicate activity/link, and speed and duplex/collision status information for the respective ports.

Table 2-17: Ethernet Port Signal Description

SIGNAL	DESCRIPTION
ETH1_T+/ETH2_T+ ETH1_T-/ETH2_T-	Transmit pair in 10BaseT/100BaseTX configuration
ETH1_R+/ETH2_R+ ETH1_R-/ETH2_R-	Receive pair in 10BaseT/100BaseTX configuration
ETH1_ACT/LINK_LED/ ETH2_ACT/LINK_LED	Indicates that link is present/Network traffic takes place Steady on: link is present; blinking: traffic on link
ETH1_DUP/COL_LED/ ETH2_DUP/COL_LED	Indicates that transaction mode is DUPLEX/package collision occurs
ETH1_SPEED_LED/ ETH2_SPEED_LED	Indicates link speed Out: 10 Mbit On: 100 Mbit

The following figure illustrates the Ethernet LED configuration.

Figure 2-3: Ethernet LED Configuration



27854.02.VC.060131/114651

2.6 Communication Interface Extension

The Communication Interface Extension Connector J4 (Pn4) provides the signals for an 8-bit Utopia-2 interface and data and address lines for an Utopia PHY control port (UTOPIA is not available with the EB420 board).

The following table provides signal pinouts along with information concerning the signal characteristics for connector J4 (Pn4) of EB42x E²Brain™ module.

Table 2-18: Pinout of J4 (Pn4) Connector

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
1	GND	-	POWER	
2	GND		POWER	
3	RESERVED	I/O	Reserved	
4	RESERVED	I/O	Reserved	
5	RESERVED	I/O	Reserved	
6	RESERVED	I/O	Reserved	
7	X-BUS-RD	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
8	X-BUS-WR	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
9	UTP_CS	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
10	RESERVED	I/O	Reserved	
11	GND	-	POWER	
12	GND	-	POWER	
13	X-BUS-DATA(7)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
14	X-BUS-DATA(6)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
15	X-BUS-DATA(5)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
16	X-BUS-DATA(4)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
17	X-BUS-DATA(3)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
18	X-BUS-DATA(2)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
19	X-BUS-DATA(1)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
20	X-BUS-DATA(0)	I/O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
21	X-BUS-ADDR(9)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
22	GND	-	POWER	
23	X-BUS-ADDR(7)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
24	X-BUS-ADDR(8)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
25	X-BUS-ADDR(5)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
26	X-BUS-ADDR(6)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
27	X-BUS-ADDR(3)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
28	X-BUS-ADDR(4)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
29	X-BUS-ADDR(1)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.

Table 2-18: Pinout of J4 (Pn4) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
30	X-BUS-ADDR(2)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
31	NC	-	-	
32	X-BUS-ADDR(0)	O	X-BUS	This signal can be used, e.g. to configure the ATM PHY.
33	NC	-	-	
34	GND	-	POWER	
35	GND	-	POWER	
36	NC	-	-	
37	NC	-	-	
38	NC	-	-	
39	NC	-	-	
40	NC	-	-	
41	NC	-	-	
42	NC	-	-	
43	NC	-	-	
44	NC	-	-	
45	GND	-	POWER	
46	GND	-	POWER	
47	NC	-	-	
48	NC	-	-	
49	NC	-	-	
50	NC	-	-	
51	NC	-	-	
52	NC	-	-	
53	NC	-	-	
54	NC	-	-	
55	NC	-	-	
56	GND	-	POWER	
57	NC	-	-	
58	NC		POWER	
59	NC	-	-	
60	NC	-	-	
61	NC	-	-	
62	NC	-	-	
63	NC	-	-	
64	NC	-	-	

27854.02.VC.060131/114652

Table 2-18: Pinout of J4 (Pn4) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
65	NC	-	-	
66	GND	-	POWER	
67	NC	-	-	
68	NC	-	-	
69	GND	-	POWER	
70	NC	-	-	
71	NC	-	-	
72	NC	-	-	
73	NC	-	-	
74	NC	-	-	
75	NC	-	-	
76	UTP_OP_DATA0 *	O	UTOPIA	
77	NC	-	-	
78	UTP_OP_DATA2 *	O	UTOPIA	
79	UTP_OP_DATA1 *	O	UTOPIA	
80	GND	-	POWER	
81	GND	-	POWER	
82	UTP_OP_DATA4 *	OI	UTOPIA	
83	UTP_OP_DATA3 *	O	UTOPIA	
84	UTP_OP_DATA6 *	O	UTOPIA	
85	UTP_OP_DATA5 *	O	UTOPIA	
86	UTP_IP_DATA6 *	I	UTOPIA	Internal pull-down 10kΩ
87	UTP_OP_DATA7 *	O	UTOPIA	
88	UTP_IP_DATA4 *	I	UTOPIA	Internal pull-down 10kΩ
89	UTP_IP_DATA7 *	I	UTOPIA	Internal pull-down 10kΩ
90	GND	-	POWER	
91	UTP_IP_DATA5 *	I	UTOPIA	Internal pull-down 10kΩ
92	UTP_IP_DATA2 *	I	UTOPIA	Internal pull-down 10kΩ
93	UTP_IP_DATA3 *	I	UTOPIA	Internal pull-down 10kΩ
94	UTP_IP_DATA0 *	I	UTOPIA	Internal pull-down 10kΩ
95	UTP_IP_DATA1 *	I	UTOPIA	Internal pull-down 10kΩ
96	UTP_IP_ADDR4 *	I	UTOPIA	
97	UTP_OP_ADDR4 *	O	UTOPIA	
98	UTP_IP_ADDR3 *	I	UTOPIA	
99	UTP_OP_ADDR3 *	O	UTOPIA	

Table 2-18: Pinout of J4 (Pn4) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
100	UTP_IP_ADDR2 *	I	UTOPIA	
101	UTP_OP_ADDR2 *	O	UTOPIA	
102	GND	-	POWER	
103	UTP_OP_ADDR1 *	O	UTOPIA	
104	UTP_IP_ADDR1 *	I	UTOPIA	
105	GND	-	POWER	
106	UTP_IP_ADDR0 *	I	UTOPIA	
107	UTP_OP_ADDR0 *	O	UTOPIA	
108	NC	-	-	
109	NC	-	-	
110	NC	-	-	
111	GND	-	POWER	
112	NC	-	-	
113	NC	-	-	
114	GND	-	POWER	
115	GND	-	POWER	
116	NC	-	-	
117	NC	-	-	
118	NC	-	-	
119	NC	-	-	
120	NC	-	-	
121	NC	-	-	
122	NC	-	-	
123	NC	-	-	
124	GND	-	POWER	
125	NC	-	-	
126	NC	-	-	
127	NC	-	-	
128	UTP_OP_FCI *	O	UTOPIA	Internal pull-down 10kΩ
129	UTP_OP_FCO *	O	UTOPIA	
130	UTP_IP_FCO *	I	UTOPIA	
131	UTP_OP_SOC *	O	UTOPIA	
132	UTP_IP_FCI *	I	UTOPIA	Internal pull-down 10kΩ
133	UTP_IP_SOC *	I	UTOPIA	Internal pull-down 10kΩ
134	UTP_IP_CLK *	I	UTOPIA	Internal pull-down 10kΩ

27854.02.VC.060131/114653

Table 2-18: Pinout of J4 (Pn4) Connector (Continued)

PIN	SIGNAL	TYPE	SIGNAL GROUP	REMARKS
135	NC	-	-	
136	GND	-	POWER	
137	NC	-	-	
138	UTP_OP_CLK *	O	UTOPIA	Internal pull-down 10kΩ
139	GND	-	POWER	
140	GND	-	POWER	

* All signals marked with an * (asterix) must be left unconnected on the carrier board for EB420 board variants.

2.7 Monitor and Control (M/C)

Monitor and Control functions are divided essentially into Pre-operation and Operation. Pre-operation M/C deals with board configuration and system requirements. Operation M/C covers direct operational interfaces. For further information regarding Monitor and Control functions refer to chapters 2.3.3 and 4.

2.7.1 Pre-Operation M/C

Pre-operation M/C is a direct function of the application and the system requirements. These requirements dictate the EB42x configuration as well as the overall system integration. Overall system integration and compliance with its requirements is beyond the scope of this manual.

2.7.2 Operation M/C

Operation M/C is primarily a function of the EB42x driver software and the application. M/C signals are available, and, if implemented as part of the application, the operator as well as application software has access to these functions.

This page has been intentionally left blank.

27854.02.VC.060131/114653

Chapter 3

Installation

27854.02.VC.060131/114653

This page has been intentionally left blank.

3. Installation

The EB42x has been designed for easy installation. However, the following standard precautions, installation procedures, and general information must be observed to ensure proper installation and to preclude damage to the board or injury to personnel.

3.1 Hardware Installation

The product described in this manual may only be mounted on an appropriate E²Brain™ carrier board which is specifically designed for this E²Brain™ module.

3.1.1 Safety Requirements

The module must be securely fastened to the carrier board using the mounting standoffs and screws provided with the module.

In addition, the following electrical hazard precautions must be observed.

Caution, Electric Shock Hazard!

Ensure that the system main power is removed prior to installing or removing this board. Ensure that there are no other external voltages or signals being applied to this board or other boards within the system. Failure to comply with the above could endanger your life or health and may cause damage to this board or other system components including process-side signal conditioning equipment.

ESD Equipment!

This Kontron board contains electrostatically sensitive devices. Please observe the following precautions to avoid damage to your board:

Discharge your clothing before touching the assembly. Tools must be discharged before use.

Do not touch any on board components, connector pins, or board conductive circuits.

If working at an anti-static workbench with professional discharging equipment, ensure compliance with its usage when handling this product.

3.1.2 Installation Procedures

To install this E²Brain™ module proceed as follows:

1. Ensure that the safety requirements indicated in chapter 3.1.1 are observed.

Warning!

Failure to comply with the instruction below may cause damage to the board or result in improper system operation. Please refer to chapters 4 and 5 for configuration information.

2. Ensure that the board is properly configured for operation before installing.

Note ...

Care must be taken when applying the procedures below to ensure that when the board is inserted it is not damaged through contact with other boards in the system.

3. To install the E²Brain™ module perform the following:
   1. Orient the E²Brain™ module as appropriate to the carrier board and engage it with the carrier board.
   2. Fasten all mounting screws provided with the E²Brain™ module ensuring that the standoffs are also properly fastened.
   3. As appropriate, install the carrier board in the application system.

3.1.3 Removal Procedures

To remove this module proceed as follows:

1. Ensure that the safety requirements indicated in chapter 3.1.1 are observed.

Warning!

Care must be taken when applying the procedures below to ensure that when the board is removed it is not damaged through contact with other boards in the system.

2. Disconnect any interfacing cable(s) that may be connected to the module.
3. Remove all module mounting screws.
4. Disengage the module from the carrier board.
5. Reinstall the module mounting screws in the module standoffs.
6. Dispose of the module as required observing applicable environmental regulations governing the handling and disposition of this type of product.

3.2 Software Installation

Installation of the EB42x driver software is a function of the application operating system. For further information refer to the appropriate software documentation.

Chapter 4

Configuration

This page has been intentionally left blank.

4. Configuration

The following sections provide system integrators with detailed information for configuring the EB42x module for operation

4.1 Board Address Map

The following table illustrates the address mapping of the EB42x.

Table 4-1: Board Address Map

START ADDRESS	SIZE BYTE	DESCRIPTION
0x1000 0000	756 M	SDRAM Data
0x3000 0000	256 M	Reserved
0x4000 0000	128 M	Reserved
0x4800 0000	128 M	PCI Data
0x5000 0000	256 M	Expansion Bus Data
0x6000 0000	64 M	Queue Manager
0x6400 0000		Reserved
0xC000 0000	64 M	PCI Contr. Conf./Stat Reg
0xC400 0000	64 M	Exp. Bus Config. Reg
0xC800 0000	1 k	High-Speed UART
0xC800 1000	1 k	Console UART
0xC800 2000	1 k	Intl. Bus Perf. Mon. Unit
0xC800 3000	1 k	Interrupt Controller
0xC800 4000	1 k	GPIO Controller
0xC800 5000	1 k	Timers
0xC800 6000	1 k	WAN/HSS NPE
0xC800 7000	1 k	Ethernet NPE A
0xC800 8000	1 k	Ethernet NPE B
0xC800 9000	1 k	Ethernet MAC A
0xC800 A000	1 k	Ethernet MAC B
0xC800 B000	1 k	USB Controller
0xC800 C000		Reserved
0xC801 0000		Reserved
0xCC00 0000	256	SDRAM Config. Reg
0xCC00 0100		Reserved
0xD000 0000		Reserved

START ADDRESS	SIZE BYTE	DESCRIPTION
0x5000 0000	16M	Addr. for Soldered Flash
0x5100 0000	1	Board ID
0x5100 0001	1	Software Compatibility ID
0x5100 0002	1	Memory Configuration
0x5100 0003	1	Flash Bank Select
0x5100 0004	1	Control Register
0x5100 0005	1	Event Register
0x5100 0006	1	Interrupt Config.
0x5100 0007	1	Device Interrupt Pending
0x5100 0009	1	Watchdog Control
0x5100 000a	1	Board/Logic Revision
0x5100 000b	1	External I²C Command
0x5100 000c	1	External I²C Data
0x5100 000d	1	External I²C Divider
0x5100 0010	1	Serial Interrupt Pending1
0x5100 0011	1	Serial Interrupt Pending 2
0x5100 0012	1	Serial Interrupt Mask 1
0x5100 0013	1	Serial Interrupt Mask 2
0x5100 0014	1	Serial Interrupt Polarity 1
0x5100 0015	1	Serial Interrupt Polarity 2
0x5200 0000	8	DUART Channel A
0x5200 0008	8	DUART Channel B
0x5300 0000	1M	SRAM
0x5400 0000	32	CompactFlash
0x5500 0000	16M	LPC Memory
0x5600 0000	64k	LPC IO
0x5700 0000	256	CAN

Access to the following address areas is not permitted for the access types indicated.

Table 4-2: Restricted Access Types

ACCESS TYPE	START ADDRESS	END ADDRESS
32-bit write	0x5000 0000	0x57FF FFFF
16-bit write	0x5000 0000	0x52FF FFFF
	0x5500 0000	0x57FF FFFF
16-bit read	0x5000 0000	0x52FF FFFF
	0x5500 0000	0x57FF FFFF
8-bit write	0x5300 0000	0x54FF FFFF
	0x5300 0000	0x54FF FFFF

For further information, refer to the Intel IXP4XX Product Line and IXC1100 Control Plane Processors Developer’s Manual, Chapter 8.3 Address and Data Byte Steering.

4.2 Board Control Registers

The Board Control registers may be accessed through byte-wide read and write operations.

Table 4-3: Board Control Registers

REGISTER	ADDRESS	ACCESS	
		READ	WRITE
Board ID	0x5100 0000	X	
Software Compatibility ID	0x5100 0001	X	
Memory Configuration	0x5100 0002	X	
Flash Bank Select	0x5100 0003	X	X
Control	0x5100 0004	X	X
Event	0x5100 0005	X	X
Interrupt Configuration	0x5100 0006	X	X
Device Interrupt Pending	0x5100 0007	X	
Watchdog Control	0x5100 0009	X	X
Board/Logic Revision	0x5100 000a	X	
External I²C Command	0x5100 000b	X	X
External I²C Data	0x5100 000c	X	X
External I²C Divider	0x5100 000d	X	X
Serial Interrupt Pending 1	0x5100 0010	X	
Serial Interrupt Pending 2	0x5100 0011	X	
Serial Interrupt Mask 1	0x5100 0012	X	X
Serial Interrupt Mask 2	0x5100 0013	X	X

Table 4-3: Board Control Registers (Continued)

REGISTER	ADDRESS	ACCESS	
		READ	WRITE
Serial Interrupt Polarity 1	0x5100 0014	X	X
Serial Interrupt Polarity 2	0x5100 0015	X	X

4.2.1 Board ID Register

The Board ID is used to identify the EB42x in a E²Brain™ system. The value for the EB42x is 0x42 which is factory set and cannot be changed.

Table 4-4: Board ID Register

REGISTER NAME	BOARD ID						ACCESS	
ADDRESS	0x5100 0000						R	
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	BID7	BID6	BID5	BID4	BID3	BID2	BID1	BID0
DEFAULT	0	1	0	0	0	0	1	0

4.2.2 Software Compatibility ID Register

The Software Compatibility ID register will signal to the software when differences in hardware require different handling by the software. It starts with the value 0x00 and will be incremented with each change in hardware (software sensitive only). This register is set at the factory and is for use only by the boot strap loader “NetBootLoader” and the BSP software, and, as such, is not user relevant.

Table 4-5: Software Compatibility ID Register

REGISTER NAME	SOFTWARE COMPATIBILITY ID						ACCESS	
ADDRESS	0x5100 0001						R	
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

4.2.3 Memory Configuration Register

The Memory Configuration register provides basic information concerning the amount of installed main memory.

Table 4-6: Memory Configuration Register

REGISTER NAME	MEMORY CONFIGURATION						ACCESS	
ADDRESS	0x5100 0002						R	
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	res.	res.	res.	res.	res.	SZ1	SZ0	BK
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

BIT	NAME	VAL	DESCRIPTION
0	BK	0	One Memory Bank equipped (2 chips)
		1	Two Memory Banks equipped (4 chips)
1	SZ0	0	Settings: **SZ1 SZ0** 0 0 64-Mbit SDRAM chips 0 1 128-Mbit SDRAM chips 1 0 256-Mbit SDRAM chips 1 1 512-Mbit SDRAM chips
		1	
2	SZ1	0	
		1	
3		0	reserved
		1	
4		0	reserved
		1	
5		0	reserved
		1	
6		0	reserved
		1	
7		0	reserved
		1	

4.2.4 Flash Bank Select Register

The Flash bank select register is used to select the appropriate soldered Flash bank. As Flash memory may only be accessed through a 16 MB window, this is the only way to address a larger size Flash memory. Using bit 0 (FB0), 2 Flash banks can be selected. The default value on startup of the EB42x is 0x00.

Table 4-7: Flash Bank Select Register

REGISTER NAME	FLASH BANK SELECT						ACCESS	
ADDRESS	0x5100 0003						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	res.	res.	res.	res.	res.	res.	res.	FB0
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	n/a	0

27854.02.VC.060131/114705

4.2.5 Control Register

The Control register provides access to the Monitor and Control output signals (MC6, MC9, MC10, MC11) and allows for the generation of a system reset.

During startup, the state of bit 0 is controlled by the Bootstrap Loader software. After the startup is completed, the Bootstrap Loader sets bit 0 to '1'. Then it can be used as general purpose I/O.

Table 4-8: Control Register

REGISTER NAME	CONTROL						ACCESS	
ADDRESS	0x5100 0004						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	res.	res.	MC10	RST	MC9.	res.	MC11	MC6
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	0	0

BIT	NAME	VAL	DESCRIPTION
0	MC6	0	The result is a high logic level on the MC6 output signal
		1	The result is a low logic level on the MC6 output signal
1	MC11	0	The result is a high logic level on the MC11 output signal
		1	The result is a low logic level on the MC11 output signal
2		0	Reserved
		1	
3	MC9	0	The result is a low logic level on the MC9 output signal
		1	The result is a “High-Z” state on the MC9 output signal
4	RST	0	No operation
		1	Causes a reset to be initiated
5	MC10	0	The result is a low logic level on the MC10 output signal
		1	The result is a “High-Z” state on MC10 output signal
6		0	Reserved
		1	
7		0	Reserved
		1	

4.2.6 Event Register

The Event register is used to indicate the origin of the generation of the non-maskable interrupts caused either by a Watchdog timeout or the pressing of the Abort switch.

Table 4-9: Event Register

REGISTER NAME	EVENT						ACCESS	
ADDRESS	0x5100 0005						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	MC5	MC3	MC4	MC2	MC8	MC1	ALARM	WD
DEFAULT	n/a	n/a	n/a	n/a	n/a	0	0	0

BIT	NAME	VAL	DESCRIPTION
0	WD	0	Indicates that no Watchdog timeout has occurred
		1	Indicates that a Watchdog timeout has occurred (this bit may be cleared by writing a '1')
1	ALARM	0	Indicates that ALARM signal on RTC has not been asserted
		1	Indicates that ALARM signal on RTC has been asserted
2	MC1	0	Indicates that the MC1 signal has not been asserted (active low)
		1	Indicates that the MC1 signal has been asserted (This bit can be cleared by writing a '1'.)
3	MC8	0	Indicates the status of the MC8 pin
		1	
4	MC2	0	If MC2_INT_EN = 0, then it indicates the status of the MC2 input
		1	If MC2_INT_EN = 1, then the falling edge of the signal on MC2 pin sets this bit to '1' and generates the MC2Int on the CPU provided it is enabled there (this may be cleared by writing a '1')
5	MC4	0	Reserved for manufacturer purposes/read only
		1	
6	MC3	0	If MC3_INT_EN = 0, then it indicates the status of the MC3 input
		1	If MC2_INT_EN = 1, then the falling edge of the signal on MC3 pin sets this bit to '1' and generates the MC3Int on the CPU provided it is enabled there (this may be cleared by writing a '1')
7	MC5	0	If MC5_INT_EN = 0, then it indicates the status of the MC5 input
		1	If MC5_INT_EN = 1, then the falling edge of the signal on MC5 pin sets this bit to '1' and generates the MC5Int on the CPU provided it is enabled there (this bit may be cleared by writing a '1')

27854.02.VC.060131/114706

4.2.7 Interrupt Configuration Register

The interrupt configuration register acts as an interrupt enable register for the MC2, MC3 and MC5 signals.

Table 4-10: Interrupt Configuration Register

REGISTER NAME	INTERRUPT CONFIGURATION						ACCESS	
ADDRESS	0x5100 0006						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	MC5_INT_EN	MC3_INT_EN	res.	MC2_INT_EN	res.	res.	res.	res.
DEFAULT	0	0	0	0	0	0	0	0

0		0	Reserved
		1	
1		0	Reserved
		1	
2		0	Reserved
		1	
3		0	Reserved
		1	
4	MC2_INT_EN	0	Disabled
		1	Enabled
5		0	Reserved
		1	
6	MC3_INT_EN	0	Disabled
		1	Enabled
7	MC5_INT_EN	0	Disabled
		1	Enabled

4.2.8 Device Interrupt Pending Register

The Device Interrupt Pending Register is used to identify the source of the pending interrupt request of the following onboard devices:

- CompactFlash
- Temperature sensor
- CAN controller
- UARTs (SER1 and SER2)

Table 4-11: Device Interrupt Pending Register

REGISTER NAME	DEVICE INTERRUPT PENDING						ACCESS	
ADDRESS	0x5100 0007						R	
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	CF	TEMP	res.	CAN	res.	res.	SER2	SER1
DEFAULT	0	0	0	0	0	0	0	0

For information regarding the required hardware resources, refer to Chapter 4.4 Interrupt Mapping in this manual. A logical ’1’ indicates that an interrupt has been asserted.

4.2.9 Watchdog Control Register

The Watchdog Control register is the interface between applications and the operating system for controlling the functioning of the Watchdog timer. In normal mode, the watchdog timer can either assert a system reset or an interrupt at timeout. In cascaded mode, an interrupt is asserted at the first timeout, and then if a second watchdog timeout occurs, a system reset is asserted. The corresponding interrupt pending bit is WD (bit 0) in the Event Register.

27854.02.VC.060131/114707

Table 4-12: Watchdog Control Register

REGISTER NAME	WATCHDOG CONTROL						ACCESS	
ADDRESS	0x5100 0009						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	WD_EN	WD_R	WD_CCD	WD_TRG	WDT3	WDT2	WDT1	WDT0
DEFAULT	0	0	n/a	0	0	0	0	0

BIT	NAME	VAL	DESCRIPTION
0	WDT0	0	Watchdog timeout time: Settings: **WDT3 WDT2 WDT1 WDT0** 0 0 0 0 0.5 seconds 0 0 0 1 1.0 seconds 0 0 1 0 1.5 seconds 0 0 1 1 2.0 seconds 0 1 0 0 2.5 seconds 0 1 0 1 3.0 seconds 0 1 1 0 3.5 seconds 0 1 1 1 4.0 seconds 1 0 0 0 4.5 seconds 1 0 0 1 5.0 seconds 1 0 1 0 5.5 seconds 1 0 1 1 6.0 seconds 1 1 0 0 6.5 seconds 1 1 0 1 7.0 seconds 1 1 1 0 7.5 seconds 1 1 1 1 8.0 seconds
		1	
1	WDT1	0	
		1	
2	WDT2	0	
		1	
3	WDT3	0	
		1	
4	WD_TRG	-	-
		1	Writing a '1'causes the Watchdog to be retriggered (Resets Watchdog timer to value indicated by bits 0 to 3.
5	WD_CCD	0	Normal watchdog functionality
		1	Cascade mode: when watchdog timeout occurs, an interrupt will be generated, the watchdog timer resets, a further timeout will result in a system reset (when WD_R is first set to 1)
6	WD_R	0	Causes hardware reset of system upon Watchdog timeout
		1	Causes generation of an interrupt upon Watchdog timeout
7	WD_EN	0	Watchdog timer disabled
		1	Watchdog timer enabled  ***Note ...*** Once the Watchdog timer is enabled it cannot be disable except by resetting the system.

27854.02.VC.060131/114707

4.2.10 Board Logic / Revision Register

The Board Revision Register may be used to identify the hardware (BRn) and logic status of the board by the software (LRn). It is set at the factory and starts with the value 0x00 for the initial board prototypes and will be incremented with each redesign / logic release.

Table 4-13: Board Logic / Revision Register

REGISTER NAME	BOARD LOGIC/REVISION						ACCESS	
ADDRESS	0x5100 000a						R	
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	LR3	LR2	LR1	LR0	BR3	BR2	BR1	BR0
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

4.2.11 External I²C Command Register

The External I²C Command Register is used to initialize and control the bit stream from the BPCC to the I²C devices. For Read and Write access to this register, two different sets of control bits are provided.

Table 4-14: External I²C Command Register

REGISTER NAME	EXTERNAL I²C COMMAND						ACCESS	
ADDRESS	0x5100 000b						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
WRITE CONTENT	STROBE	SET ACK	res.	res.	IRQ_EN	IACK	MODE1	MODE0
READ CONTENT	BUSY	GET ACK	res.	res.	IRQ_EN	IRQ	MODE1	MODE0

Note ...

A description of the I²C controller usage is provided in Appendix A.

The following table indicates the mode bit description for the I²C controller.

Table 4-15: Mode Bit Description

MODE 1	MODE 0	DESCRIPTION
0	0	Generate Stop Condition
0	1	Generate Start Condition
1	0	Send Byte
1	1	Receive Byte

27854.02.VC.060131/114707

4.2.12 External I^{2}C Data Register

The External I^{2}C Data Register provides data which can be either read from or written to the I^{2}C devices. When reading data, the register provides the data byte that was read from an I^{2}C device. When writing data, the register provides address and data information for the I^{2}C controller.

Table 4-16: External I^{2}C Data Register

REGISTER NAME	EXTERNAL I^{2}C DATA						ACCESS	
ADDRESS	0x5100 000c						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	I^{2}C_D7	I^{2}C_D6	I^{2}C_D5	I^{2}C_D4	I^{2}C_D3	I^{2}C_D2	I^{2}C_D1	I^{2}C_D0
DEFAULT	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Note ...

A description of the I^{2}C controller usage is provided in Appendix A.

4.2.13 External I^{2}C Divider Register

The External I^{2}C Divider Register provides clock rate information for the I^{2}C controller.

Table 4-17: External I^{2}C Divider Register

REGISTER NAME	EXTERNAL I^{2}C DIVIDER						ACCESS	
ADDRESS	0x5100 000d						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	res.	res.	res.	res.	res.	CR2	CR1	CR0
DEFAULT	n/a	n/a	n/a	n/a	n/a	0	0	0

The following clock rates can be achieved by setting CR2 down to CR0.

CR[2...0] = “000”->SCL = 50 KHz

CR[2...0] = “001”->SCL = 100 KHz

CR[2...0] = “010”->SCL = 150 KHz

CR[2...0] = “011”->SCL = 200 KHz

CR[2...0] = “100”->SCL = 250 KHz

CR[2...0] = “101”->SCL = 300 KHz

CR[2...0] = “110”->SCL = 350 KHz

CR[2...0] = “111”->SCL = 400 KHz

27854.02.VC.060131/114708

4.2.14 Serial Interrupt Pending 1 Register

The Serial Interrupt Pending Register 1 in conjunction with the Serial Interrupt Pending Register 2 is used to identify the source of the pending interrupt request. All serial interrupts are coupled together to one CPU Interrupt (GPIO6). A logical 1 indicates that an interrupt has been asserted.

Table 4-18: Serial Interrupt Pending Register 1

REGISTER NAME	SERIAL INTERRUPT PENDING 1						ACCESS	
ADDRESS	0x5100 0010						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SIRQ7	SIRQ6	SIRQ5	SIRQ4	SIRQ3	SIRQ2	SIRQ1	SIRQ0
DEFAULT	0	0	0	0	0	0	0	0

4.2.15 Serial Interrupt Pending 2 Register

The Serial Interrupt Pending Register 2 in conjunction with the Serial Interrupt Pending Register 1 is used to identify the source of the pending interrupt request. All serial interrupts are coupled together to one CPU Interrupt (GPIO6). A logical 1 indicates that an interrupt has been asserted.

Table 4-19: Serial Interrupt Pending Register 2

REGISTER NAME	SERIAL INTERRUPT PENDING 2						ACCESS	
ADDRESS	0x5100 0011						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SIRQ15	SIRQ14	SIRQ13	SIRQ12	SIRQ11	SIRQ10	SIRQ9	SIRQ8
DEFAULT	0	0	0	0	0	0	0	0

4.2.16 Serial Interrupt Mask 1 Register

The Serial Interrupt Mask Registers 1 and 2 enable the generation of a CPU interrupt. Writing a '1' on the Bit SIN_ENx enables the generation of a CPU interrupt and enables the corresponding bit in the Serial Interrupt Pending Registers 1 and 2.

Table 4-20: Serial Interrupt Mask Register 1

REGISTER NAME	SERIAL INTERRUPT MASK 1						ACCESS	
ADDRESS	0x5100 0012						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SIN_EN7	SIN_EN6	SIN_EN5	SIN_EN4	SIN_EN3	SIN_EN2	SIN_EN1	SIN_EN0
DEFAULT	0	0	0	0	0	0	0	0

4.2.17 Serial Interrupt Mask 2 Register

The Serial Interrupt Mask Registers 1 and 2 enable the generation of a CPU interrupt. Writing a '1' on the Bit SIN_ENx enables the generation of a CPU interrupt and enables the corresponding bit in the Serial Interrupt Pending Registers 1 and 2.

Table 4-21: Serial Interrupt Mask Register 2

REGISTER NAME	SERIAL INTERRUPT MASK 2						ACCESS	
ADDRESS	0x5100 0013						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SIN_EN15	SIN_EN14	SIN_EN13	SIN_EN12	SIN_EN11	SIN_EN10	SIN_EN9	SIN_EN8
DEFAULT	0	0	0	0	0	0	0	0

4.2.18 Serial Interrupt Polarity 1 Register

The Serial Interrupt Polarity 1 Register bits define the polarity of their corresponding serial interrupt. A '1' written to the required bit position results in an active high sensitivity of the corresponding interrupt and vice versa.

Table 4-22: Serial Interrupt Polarity 1 Register

REGISTER NAME	SERIAL INTERRUPT POLARITY 1						ACCESS	
ADDRESS	0x5100 0014						R	W
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SerInt_POL 7	SerInt_POL 6	SerInt_POL 5	SerInt_POL 4	SerInt_POL 3	SerInt_POL 2	SerInt_POL 1	SerInt_POL 0
DEFAULT	0	0	0	0	0	0	0	0

4.2.19 Serial Interrupt Polarity 2 Register

The Serial Interrupt Polarity 2 Register bits define the polarity of their corresponding serial interrupt. A '1' written to the required bit position results in an active high sensitivity of the corresponding interrupt and vice versa

Table 4-23: Serial Interrupt Polarity 2 Register

REGISTER NAME	SERIAL INTERRUPT POLARITY 2						ACCESS	
ADDRESS	0x5100 0015							
BIT POSITION	MSB 7	6	5	4	3	2	1	0 LSB
CONTENT	SerInt_POL 15	SerInt_POL 14	SerInt_POL 13	SerInt_POL 12	SerInt_POL 11	SerInt_POL 10	SerInt_POL 9	SerInt_POL 8
DEFAULT	0	0	0	0	0	0	0	0

4.3 UART Registers Address Mapping

4.3.1 UART A (SER1)

The following table indicate the address mapping of the UART A (SER1). For a more detailed description please refer to the EXAR XR16C2850 or XR16C2750 DUART manual.

Table 4-24: UART A General Register Set

READ MODE	WRITE MODE	ADDRESS
Receive Holding Register	Transmit Holding Register	0x5200 0000
n/a	Interrupt Enable Register	0x5200 0001
Interrupt Status Register	FIFO Control Register	0x5200 0002
n/a	Line Control Register	0x5200 0003
n/a	Modem Control Register	0x5200 0004
Line Status Register	n/a	0x5200 0005
Modem Status Register	n/a	0x5200 0006
Scratchpad Register	Scratchpad Register	0x5200 0007

Accessible only when CS A/B is logical 0.

Table 4-25: UART A Baud Rate Register Set

READ MODE	WRITE MODE	ADDRESS
LSB of divisor latch	LSB of divisor latch	0x5200 0000
MSB of divisor latch	MSB of divisor latch	0x5200 0001

Accessible only when CS A/B is logical 0 and LCR bit 7 is a logical 1.

Table 4-26: UART A Enhanced Register Set

READ MODE	WRITE MODE	ADDRESS
Trigger Level Register	Trigger Level Register	0x5200 0000
Feature Control Register	Feature Control Register	0x5200 0001
Enhanced Feature Register	Enhanced Feature Register	0x5200 0002
Enhanced Mode Select Register	Enhanced Mode Select Register	0x5200 0007
Xon-1	Xon-1	0x5200 0004
Xon-2	Xon-2	0x5200 0005
Xoff-1	Xoff-1	0x5200 0006
Xoff-2	Xoff-2	0x5200 0007

Accessible only when LCR is set to “BF” hex.

27854.02.VC.060131/114709

4.3.2 UART B (SER2)

The following table indicate the address mapping of the UART B (SER2). For a more detailed description please refer to the EXAR XR16C2850 or XR16C2750 DUART manual.

Table 4-27: UART B General Register Set

READ MODE	WRITE MODE	ADDRESS
Receive Holding Register	Transmit Holding Register	0x5200 0008
n/a	Interrupt Enable Register	0x5200 0009
Interrupt Status Register	FIFO Control Register	0x5200 000A
n/a	Line Control Register	0x5200 000B
n/a	Modem Control Register	0x5200 000C
Line Status Register	n/a	0x5200 000D
Modem Status Register	n/a	0x5200 000E
Scratchpad Register	Scratchpad Register	0x5200 000F

Accessible only when CS A/B is logical 0.

Table 4-28: UART B Baud Rate Register Set

READ MODE	WRITE MODE	ADDRESS
LSB of divisor latch	LSB of divisor latch	0x5200 0008
MSB of divisor latch	MSB of divisor latch	0x5200 0009

Accessible only when CS A/B is logical 0 and LCR bit 7 is a logical 1.

Table 4-29: UART B Enhanced Register Set

READ MODE	WRITE MODE	ADDRESS
Trigger Level Register	Trigger Level Register	0x5200 0008
Feature Control Register	Feature Control Register	0x5200 0009
Enhanced Feature Register	Enhanced Feature Register	0x5200 000A
Enhanced Mode Select Register	Enhanced Mode Select Register	0x5200 000F
Xon-1	Xon-1	0x5200 000C
Xon-2	Xon-2	0x5200 000D
Xoff-1	Xoff-1	0x5200 000E
Xoff-2	Xoff-2	0x5200 000F

Accessible only when LCR is set to “BF” hex.

4.4 Interrupt Mapping

The interrupt controller of the IXP42x (CPU) supports 12 external interrupts. The interrupt mapping is listed below.

Table 4-30: Interrupt Mapping

INTERRUPT INPUT	SOURCE
GPIO0	WDG Interrupt
GPIO1	MC2/MC3/MC5 Interrupt
GPIO2	CAN Interrupt
GPIO3	CF Interrupt
GPIO4	UART 5/6 Interrupt
GPIO5	I^2C Interrupt
GPIO6	Serial Interrupt (LPC)
GPIO7	Alarm, Temperature
GPIO8	P_INTA
GPIO9	P_INTB
GPIO10	P_INTC
GPIO11	P_INTD

4.5 Real-Time Clock

Access to the real-time clock (RTC) is effected via the I²C bus. The RTC uses address 0xD0. For more detailed information please refer to the manuals for the ST - Microelectronics M41T81.

Table 4-31: Register Map RTC M41T81

ADR (HEX)	ADDRESS BITS								FUNCTION RANGE IN BCD FORMAT
	D7	D6	D5	D4	D3	D2	D1	D0	
00	0.1 Seconds				0.01 Seconds				Seconds: 00 - 99
01	ST	10 Seconds			Seconds				Seconds: 00 - 59
02	0	10 Minutes			Minutes				Minutes: 00 - 59
03	CEB	CB	10 Hours		Hours				Century: 0 - 1 Hours: 00 - 23
04	0	0	0	0	0	Day			Day: 00 - 07
05	0	0	10 Date		Date				Date: 01 - 31
06	0	0	0	10M.	Month				Month: 01 - 12
07	10 Years				Year				Year: 00 - 99
08	OUT	FT	S	Calibraton					Control:
09	0	BMB4	BMB3	BMB2	BMB1	BMB0	RB1	RB0	Watchdog:
0A	AFE	SQWE	ABE	Al 10M	Alarm Month				Alarm Month: 01 - 12
0B	RPT4	RPT5	Al 10 Date		Alarm Date				Alarm Date: 01 - 31
0C	RPT3	HT	Al 10 Hour		Alarm Hour				Al Hour: 00 - 23
0D	RPT2	Alarm 10 Minutes			Alarm Minutes				Al Min: 00 - 59
0E	RPT1	Alarm 10 Seconds			Alarm Seconds				Al Sec: 00 - 59
0F	WDF	AF	0	0	0	0	0	0	Flags:
10	0	0	0	0	0	0	0	0	Reserved:
11	0	0	0	0	0	0	0	0	Reserved:
12	0	0	0	0	0	0	0	0	Reserved:
13	RS3	RS2	RS1	RS0	0	0	0	0	SQW:

Legend for Table 4-22

0	=	Must set to ’0’
ABE	=	Alarm in battery back-up mode enable bit
AF	=	Alarm flag (read only)
AFE	=	Alarm flag enable flag
BMBn	=	Watchdog multiplier bit(s)
CB	=	Century bit
CEB	=	Century enable bit
FT	=	Frequency test bit
HT	=	Halt update bit
OUT	=	Output level
RBn	=	Watchdog resolution bit(s)
RPTn	=	Alarm repeat mode bit(s)
RSn	=	SQW frequency
S	=	Sign bit
SQWE	=	Square wave enable
ST	=	Stop bit
WDF	=	Watchdog flag (read only)

Note ...

When the RTC has once been stopped due to low voltage, it is necessary to re-initialize the “Seconds” “Minutes” and “Hours” registers before it will run again.

4.6 CompactFlash

Write access to the CompactFlash address area is only possible using word-wide (16-bit) write commands.

Table 4-32: CompactFlash Register

REGISTER	READ/WRITE	ADDRESS
Data Register	R/W	0x5400 0000
Error Register	R	0x5400 0002
Feature Register	W	0x5400 0002
Sector Count Register	R/W	0x5400 0004
Sector Number Register	R/W	0x5400 0006
Cylinder Low Register	R/W	0x5400 0008
Cylinder High Register	R/W	0x5400 000A
Drive/Head Register	R/W	0x5400 000C
Status Register	R	0x5400 000E
Device Control Register	W	0x5400 000E
Alternate Status Register	R	0x5400 001C
Digital Output Register	W	0x5400 001C
Card Drive Address Register		0x5400 001E

4.7 EEPROM

Access to the EEPROM is effected via the I²C bus of the EB42x. The EEPROM uses the I²C address 0xA0. Write protection is achieved by installing 0 ohm resistor R130 and removing 0 ohm resistor R140. Default is unprotected.

For more detailed information please refer to the manuals for the MICROCHIP 24LC64 or Catalyst 24WC64.

4.8 Digital Temperature Sensor, LM75

Access to the optional onboard digital temperature sensor (DTS) is effected via the I²C bus of the BPCC. The DTS uses the I²C address 0x90.

For more detailed information please refer to the manuals for the National Semiconductor LM75.

27854.02.VC.060131/114711

This page has been intentionally left blank.

Chapter 5

NetBootLoader

This page has been intentionally left blank.

5. NetBootLoader

This E²Brain™ module is delivered with the NetBootLoader software already programmed into the onboard soldered Flash memory. The NetBootLoader itself is a software utility which initializes the module for operation before turning control over to either an application or to an operator. This software also provides the capability to monitor and control the operation of the NetBootLoader itself, display system status information, to program executable code and data to the Flash memory, and to load and start application software.

To attain full operational capability, the NetBootLoader FLASH must be programmed by the user with application software. Once the application has been programmed to Flash memory, the NetBootLoader will support the complete boot operation. The following chapters describe the functioning of the NetBootLoader and how to program the Flash memory.

Note ...

The following description assumes a standard CPU board with appropriate hardware. In the event such hardware is not available, disregard the text that applies to the missing hardware and proceed as appropriate.

5.1 General Operation

Upon power on or a system reset, the NetBootLoader is started. The CPU board is configured for operation and control is either passed to an application or an operator. In the event a valid application has been programmed into the Flash memory and no operator intervention takes place, the application is copied from FLASH into SDRAM and control is passed to the application. If the NetBootLoader does not find a valid application or operator intervention has occurred, control is passed to the operator. The operator now has control to determine the system status, make configuration changes, read or program the Flash memory, or to restart or shut down the system.

The operator command interfacing with the NetBootLoader is accomplished either via the TERM serial port or the Ethernet port. During the boot operation a command interpreter is started which allows the operator to input commands to the NetBootLoader. Prior to interfacing via the Ethernet port the network must be configured. This is done via the TERM port.

5.2 NetBootLoader Interfaces

There are four possibilities to interface with the NetBootLoader:

- Via the MC1 (Abort) signal
- Via the TERM serial interface
- Via the SER0 serial interface
- Via the Ethernet interface

Gaining access to the NetBootLoader is a function of the contents of the Flash memory and the “BootWaitTime” setting. If there is no valid application programmed into the Flash memory, the boot operation automatically terminates after the module has been initialized and control is passed to the command interpreter. If there is a valid application in the Flash memory the boot operation is delayed according to the setting of the boot wait time, and the MC6 (LED1) output signal is alternately asserted indicating that the boot operation is in a wait state. During this time the operator may intervene in the boot operation either by asserting the MC1 (Abort) signal, entering the “abort” command via the TERM interface, or by performing a successful telnet login via the Ethernet interface. If the operator does not intervene, the boot operation is continued after the boot wait time has been exceeded.

5.2.1 MC1 (Abort) Signal

The MC1 (Abort) signal is routed to the EB42x carrier board via the System Interface (CON1 connector) and, if made available on the carrier, provides the operator with the ability to directly terminate the boot operation during the boot wait time which is indicated by the alternately asserted MC6 (LED1) signal. This is the sole purpose of the MC1 (Abort) signal during the NetBootLoader operation.

5.2.2 TERM Serial Interface

The TERM serial port, if realized on the carrier board, is used to provide direct operator interfacing to the NetBootLoader. As soon as the CPU board has been initialized this port is activated and the operator may input commands. During the boot wait time the operator may terminate the boot operation and take control of the NetBootLoader. Once the boot wait time is exceeded the command interpreter is deactivated and the operator no longer has access to the NetBootLoader.

The TERM serial interface may either be directly connected to a terminal device or may interface with a terminal emulator.

5.2.3 SER0 Serial Interface

The SER0 serial port is used to provide the NetBootLoader with the ability to access Motorola S-Records for programming an application to FLASH. No command interpreter is available for this interface.

5.2.4 Ethernet Interface

The Ethernet interface provides the capability of remotely interfacing with the NetBootLoader. Prior to using this interface it is necessary to configure the NetBootLoader network settings. This is accomplished via the TERM interface. Once the network settings have been made, the remote operator has the same capabilities as with the TERM interface. During the boot wait time the operator gains control of the NetBootLoader by logging into it via the Ethernet interface. This causes the boot operation to be terminated and gives control to the remote operator.

The Ethernet interface uses the telnet protocol for operator interfacing with the NetBootLoader. In addition to the operator interface via Ethernet, the NetBootLoader also uses the Ethernet interface for ftp server access.

5.3 NetBootLoader Functions

In addition to initializing the CPU board for operation and the loading and starting of applications, the NetBootLoader provides the following operator monitor and control functions:

- NetBootLoader control
- system status monitoring
- ftp server access
- FLASH reading and programming operations
- Motorola S-Record acquisition

These functions are described in detail in the following chapters.

Note ...

The command title (CMD TITLE) is expressed in capital letters and is not the same as the syntax of the command. The command syntax is always written using small letters.

5.3.1 NetBootLoader Control

The NetBootLoader provides various functions for controlling the operation of the NetBootLoader itself as well as the setting of operational parameters. The following table provides an overview of available NetBootLoader control functions.

Table 5-1: NetBootLoader Control Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
ABORT	-	Terminate boot wait	
BW	Boot Wait	Set or display BootWaitTime	
HELP or ?	-	Display online HELP pages	
LOGOUT	-	Terminate telnet session	
NET	-	Set network parameters	Must be set before attempting telnet login
PASSWD	Password	Set telnet password	Must be set before attempting telnet login
PF	Port Format	Set serial port parameters	Used for both TERM and SER0 ports
RS	Reset	Resets system	

5.3.2 System Status Monitoring

The NetBootLoader provides various functions for monitoring the overall status of the system during the operation of the NetBootLoader. The following table provides an overview of available system status monitoring functions.

Table 5-2: System Status Monitoring Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CHECK	-	Application validation	Verifies validity of user image programmed to FLASH
INFO	-	Display system information	
MD	Memory Display	Display memory contents	Applies to all visible memory
PCI	-	Display PCI device information	
PING	-	Verify network status	
VER	Version	Display version number of NetBootLoader	

27854.02.VC.060131/114712

5.3.3 ftp Server Access

The NetBootLoader provides various functions for interfacing with an ftp server. The following table provides an overview of available ftp server functions.

Table 5-3: ftp Server Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
BYE	-	Terminate session with ftp server	
CD	Change Directory	Change ftp server directory	
GET	-	Download a file from ftp server	Only for executable applications. Data buffer is target.
LOGIN	-	Login to ftp server	
LS	List Directory	List ftp server directory	Lists contents of directory.
PUT	-	Upload a file to ftp server	Data buffer is source.
PWD	Print Working Directory	Display current ftp server directory	Lists name of directory

5.3.4 FLASH Operation

The NetBootLoader provides various functions for performing operations with Flash memory. The following table provides an overview of available FLASH operation functions.

Table 5-4: FLASH Operation Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CLONE	-	Program NetBootLoader to FLASH	Uses data buffer or socket as source
LF	Load FLASH	Program application to FLASH	Uses data buffer as source
SF	Store FLASH	Reads FLASH to data buffer	Uses data buffer as target

5.3.5 Motorola S-Records

The NetBootLoader provides one function for acquiring Motorola S-Records. The following table provides an overview of this function.

Table 5-5: Motorola S-Records Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
SL	SLoad	Download Motorola S-Records	Uses data buffer as target

5.4 Operating the NetBootLoader

5.4.1 Initial Setup

The CPU board is delivered with the NetBootLoader already installed in the onboard soldered FLASH and is ready for operation. However, in order for the CPU board to be used in a system, application software must be made available for use. This is accomplished by programming the application also to the onboard soldered Flash memory where the NetBootLoader is located.

Upon initial power up the NetBootLoader is started automatically. As soon as the NetBootLoader has completed initialization of the CPU board, it checks to see if there is a valid application programmed in FLASH and at the same time initiates a command interpreter which the operator can access either via the TERM or telnet interfaces. If there is no valid application in memory, the NetBootLoader terminates the boot operation, and waits for operator intervention. As this is the case when the CPU board is first powered up, the operator now has the opportunity to program an application.

Prior to programming an application it may be necessary to configure the NetBootLoader or perform other functions depending on the user’s application development environment or application requirements. Once this has be accomplished and the application has been programmed, the CPU board is ready for operation.

The following chapters provide information on how to set up and operate the NetBootLoader itself, initiation of the telnet interface, and how to program an application to FLASH.

5.4.2 Accessing the NetBootLoader

Initial access to the NetBootLoader can only be achieved via the TERM interface. Prior to using the telnet interface, the Ethernet parameters must be set and this can only be accomplished initially via the TERM interface. Once valid Ethernet parameters and the telnet login password have be set, the telnet interface is available for operation.

Use of the TERM interface requires either a terminal or a terminal emulator. Use of the telnet interface requires a remote telnet login to the NetBootLoader.

Availability of the command interpreter depends on the system status. If there is no valid application programmed, the command interpreter is available as long as the operator requires it. If a valid application is programmed, the command interpreter is only available for the duration of the boot wait time. If the operator requires the command interpreter for a longer time he must terminate the boot operation before the boot wait time is exceeded.

Upon initiation of the command interpreter, a prompt is sent to the TERM interface and commands may be entered. To gain access to the NetBootLoader from a remote location via Ethernet a telnet login must be performed. If the boot wait time has not been exceeded, a telnet login automatically terminates the boot operation and a command prompt is sent to the telnet remote interface.

Once the operator has control of the NetBootLoader, he may perform any required action. To continue with the operation of the CPU board, the system must either be cold started or the operator must issue a “reset” command. In either event, the NetBootLoader is restarted and the boot operation begins anew.

5.4.3 NetBootLoader Configuration

There are several NetBootLoader commands which provide the operator with the capability to configure specific parameters which are used by the NetBootLoader for interfacing operations. These commands are:

- BW (BootWait)
- NET
- PASSWD
- PF (Port Format)

Default settings are available for all the above commands except for “net” which is dependent on the application environment.

5.4.3.1 BW

This command is used to display or set the actual boot wait time used by the NetBootLoader to delay the boot operation before proceeding with the loading and starting of an application. If this time is set too short it may only be possible to gain access to the NetBootLoader via the MC1 (Abort) signal.

The BootWaitTime value is stored in the boot section of the serial EEPROM. This section is validated with a CRC code to avoid the setting of random parameters.

Note ...

If the CRC of the boot section is not valid, changing the BootWaitTime will have no effect because the “bw” command does not validate an invalid CRC. In this case, a default timing of 5 seconds is always used.

To validate an invalid CRC, an operating system utility must be used, or, alternatively, the “-f” option of the “bw” command must be issued.

Warning!

Using the “bw -f” command to validate invalid entries may adversely impact the operation of the operating system.

5.4.3.2 NET

This command is used to set or display the parameters for the configuration of the Ethernet interface of the CPU board. The Ethernet interface is only available after these settings have been made. Once these settings have been made, the system must be cold started or reset for them to take effect.

5.4.3.3 PASSWD

This command is used to set the password used by the NetBootLoader for the operation of the telnet interface. No password is required for access from the TERM interface.

5.4.3.4 PF

This command is used to set the port parameters for the TERM and SER0 serial interfaces only for the current operator session. The next system restart will cause these settings to revert to the default settings of: 9600 Baud, 8 bits per character, 1 stop bit, and no parity. This is done to preclude a system lockout when restarting due to incompatible settings.

5.4.4 telnet Login

A telnet login to the NetBootLoader is only possible during the boot wait time and only after the Ethernet network parameters have been set.

To effect a telnet login the operator performs the standard telnet login procedure during the boot wait time. The NetBootLoader responds by suspending the boot wait and requests a login password. The operator then enters a password. If the password is valid, the boot wait is terminated and the operator can now access the NetBootLoader. If the password is invalid, the telnet login procedure is terminated and the boot operation continues.

In the case of an invalid password, the login procedure may be repeated as often as required within the boot wait time. Once the boot wait time is exceeded, a telnet login is no longer possible.

5.4.5 FLASH Operations

To achieve an operable system for an application, the application software must be programmed to FLASH. The NetBootLoader supports the programming of the application to FLASH. In addition to this, it also supports the updating of the NetBootLoader itself as well as data transfer from the FLASH to the data buffer and from the data buffer to an ftp server. The following chapters provide information on performing the various types of FLASH operations.

5.4.5.1 FLASH Offsets

All FLASH is treated as one uniform FLASH, regardless of the physical addresses of the devices involved. All offsets are based from the beginning of the FLASH area. This means that 0x0 is the beginning of the first FLASH bank. The NetBootLoader itself is located at the beginning of the FLASH area and for this reason this area cannot be used for application image programming. To display an overview of the current FLASH organization use the “info” command.

If the application image is an operating system (which is the default case), it must be programmed without an offset. When such an image is programmed to FLASH, the image length and CRC information is also programmed along with the image to FLASH. This information is used by the NetBootLoader to determine the validity of the image during the boot operation. During system startup, a valid image is copied to SDRAM address 0x0 and started at offset 0x0 after the boot wait time is exceeded.

If an offset is specified, the image will be programmed exactly at this offset without adding length or CRC information. This option is intended for the storing of configuration information which is required to be located in FLASH.

5.4.5.2 Programming an Application

The application image itself must be compiled and linked to run from the SDRAM base address 0x0 of the CPU. The image must contain executable ARM/XScale code at offset 0x0 which is the usual case with ROM/Flash images.

Gaining access to the image for programming to FLASH depends on where it is located. The NetBootLoader can access three different sources for images:

- ftp server
- Motorola S-Records
- memory within the visible address range of the CPU board

The NetBootLoader uses a single data buffer for downloading an image from an ftp server or an image as Motorola S-Records. These images must first be downloaded to the data buffer prior to being programmed to FLASH. An image located within the visible address range of the CPU board is directly accessible for programming.

To access an image located on an ftp server, the “get” command is used. To perform Motorola S-Record acquisition, the “sl” (SLoad) command is used. Once the image is in the data buffer, the FLASH is programmed using the “lf” (Load Flash) command. For an image within visible memory, the “lf” (LoadFlash) command is used to program directly to FLASH.

5.4.5.3 ftp Server Access

To gain access to an application image file stored on an ftp server the Ethernet interface is used. Images are downloaded to the data buffer using the ftp protocol. To use this interface the Ethernet parameters must first be set and then the system must be restarted. During boot wait the operator must gain control of the NetBootLoader and perform an ftp server login. After a successful login, the operator then locates the image file required and downloads it to the data buffer. As with any type of server session, the operator should logout when the session is finished.

Note ...

The commands “get” and “ls” use the same data buffer. Therefore if an “ls” command is issued after a “get” command the data buffer will be overwritten. If an “lf” command follows the “ls” the NetBootLoader refuses to program the overwritten data buffer to the FLASH.

5.4.5.4 Motorola S-Records

The NetBootLoader will also accept Motorola S-Records as an application image. The “sl”’ command accepts S1, S2 and S3 records. Operation is terminated by the appropriate S9, S8 or S7 record. Other types of records are ignored.

The checksum of every record except end records is checked. Bad records are rejected by the NetBootLoader. The address range of every record is also checked. Records which fall outside of the internal buffer are rejected.

The records must be 0-based. This means that it’s address must correspond to the address where they will be loaded in the data buffer relative to its start. If necessary, the base address can be modified with the -o option of the “sl” command.

Note ...

If the data buffer is programmed to FLASH without the -o option (program a startable image) the downloaded image is copied to RAM during startup and is executed there. For this reason application images which require to be programmed must start at the address 0x0.

The image must start at the absolute address 0x0 and must contain executable ARM/XScale code at the absolute address 0x0. If S1 or S2 record input is preferred, please note that these records only include 16 and 24-bit wide addresses. If no switch to another record type is included it must be ensured that the code is not larger than the address range covered.

Note ...

Neither the “sl” nor “lf” command can be used to program Motorola S-Records to RAM areas.

For accessing the Motorola S-Records, both the TERM and SER0 interfaces can be used. The MC6 (LED1) signal is asserted alternately at a low rate while downloading indicating that the transfer is in progress. The transfer itself may take several minutes to complete.

Ensure that the XON/XOFF protocol is used on the host side. This is a fixed setting and cannot be changed. Additionally, ensure that the host does not stop transmission after a number of lines (e.g. OS-9: use the ‘nopause’ attribute).

The TERM and SER0 serial interface parameters can be modified with the “pf” command.

5.4.6 Updating the NetBootLoader

In addition to programming an application to FLASH, the NetBootLoader itself can be updated. The new version of the image is made available via an ftp server.

5.4.6.1 Updating With an Image Loaded Via an ftp Server

The image is downloaded in the same way as an application image (refer to chapter 5.4.5.3). The new version of NetBootLoader image is then programmed using the “clone -n” command.

5.4.7 Uploading a FLASH Area

The NetBootLoader also has the possibility to upload certain areas of the FLASH to a host using the Ethernet interface. To use this interface the Ethernet parameters must first be set and then the system must be restarted. During boot wait the operator must gain control of the NetBootLoader and perform an ftp server login. After a successful login, the operator then stores the FLASH area to be uploaded to the local data buffer using the “sf” command. Using the “put” command transfers the contents of the data buffer to the ftp server. As with any type of server session, the operator should logout when the session is finished.

5.5 Plug and Play

On the CPU board the NetBootLoader includes “Plug and Play” functionality. This ensures that the board is completely initialized and that all resources necessary for PCI devices (addresses, interrupts etc.) are assigned automatically. This important feature has the advantage that conflicts do not arise when PCI devices are added or removed. Furthermore, the operating system itself does not include the board initialization code.

5.6 Porting an Operating System to the CPU Board

The image should be linked for the absolute address 0x0 with an entry point at the absolute address 0x0.

One should not attempt to reassign the PCI BAR registers. The assigned values should be read back and these should always be used in the drivers.

The “interrupt line” field in the PCI configuration header is initialized with the IRQ line number to which the INTA of the device is routed.

Downloaded images are never executed from the FLASH due to the fact that on the CPU board it is paged. The programmed image is always downloaded to SDRAM, the absolute address 0x0 being downloaded first. There is no configuration option available to amend this process. If it is necessary to relocate the image to another address after download, simply add a small assembly routine at the beginning of the code which will move the image to the correct address.

5.7 Commands

The following commands are available with the NetBootLoader. Where an ellipsis (…) appears in the command syntax it means that the command is continued from the previous line. Observe any spaces that may be between the ellipsis and the remainder of the command.

ABORT

FUNCTION:	Terminate the NetBootLoader boot operation
SYNTAX:	**`abort`**
DESCRIPTION:	This command is used by the operator to terminate the boot operation during the boot wait time to allow the operator to perform other NetBootLoader operations. To be asserted it must be issued during the boot wait time which is indicated by the alternating assertion of the MC6 (LED1) signal.

BW

FUNCTION:	Set or display the parameters of the boot wait function of the NetBootLoader
SYNTAX:	**`bw [<time>| -f]`** where: bw — command <time> — parameter: value: seconds 0, 1, 2, 5, 10, 20, 50 -f — option: force CRC update

BW

DESCRIPTION:	The command “bw” displays the parameter “<time>” setting. The parameter “<time>” stipulates the waiting time in seconds that the boot operation is delayed before the application is loaded and started. No values other than these are supported. Bear in mind when setting the boot wait time that the MC6 (LED1) signal is asserted alternately at the rate of two times a second. Therefore, if the boot wait is set to 1 second the MC6 signal will only be alternately asserted two times. The option “-f” is used to force updating of the CRC value of boot section of the EEPROM. For further information refer to chapter 5.4.3.1.
USAGE:	Display setting of “<time>” parameter COMMAND / RESPONSE: **bw** **WaitTime: 20**
	Set boot wait time to 50 seconds COMMAND / RESPONSE (none): **bw 50**

BYE

FUNCTION:	Terminate an ftp server session
SYNTAX:	**bye**
DESCRIPTION:	An ftp server session which has been established with the command “login” is terminated with the command “bye”.

27854.02.VC.060131/114715

CD

FUNCTION:	Change the current ftp server directory
SYNTAX:	`cd <new-path>` where: cd — command <new-path> — parameter: string new directory path
DESCRIPTION:	If an ftp server session has been established with the “login” command, the command “cd” is used to change the current ftp server directory. The argument “<new-path>” may be an absolute or relative path. The format depends on what the server accepts. For example, UNIX hosts require that the directory names must be entered exactly in the same case.

CHECK

FUNCTION:	Verify validity of application programmed to FLASH
SYNTAX:	`check`
DESCRIPTION:	When an application is programmed to FLASH, a CRC is performed and the results are stored in FLASH along with the application. The “check” command is used to verify that the current application image in FLASH is valid.

USAGE:	Veriy valid application is stored in FLASH
	
	COMMAND / RESPONSE:
	
	`check`
	`Check userimage CRC: ok`

27854.02.VC.060131/114715

CLONE

FUNCTION:	Program the NetBootLoader to FLASH
SYNTAX:	**`clone [-n]`** where: clone — command -n — option: program from data buffer
DESCRIPTION:	To update the NetBootLoader itself, the command “clone” is used. The application image source for programming is the data buffer. The image must first be downloaded to the data buffer from an ftp server. To program from the data buffer, the command “clone -n” is used. The new image is checked for validity. If an image is invalid, the update is aborted. Additionally, the operation must be confirmed by typing the word “yes”. Any other or no input will cancel the operation.
USAGE:	Program NetBootLoader (normal operation) COMMAND / RESPONSE: **`NetBtLd> clone -n`** **`clone: Fixup FLASH info from ftp buffer`** **`This will overwrite the current ...`** **`NetBootLoader, are you sure? [no] yes`** **`clone: System transferred; Start again, ...`** **`assure that Bootjumper is removed.`** **`NetBtLd>`** Note: When responding to the overwrite query, “yes” must be spelled out. Any other response will terminate the cloning operation.

27854.02.VC.060131/114715

CLONE

	Program NetBootLoader (image not valid) COMMAND / RESPONSE: **`NetBtLd> clone -n`** **`clone: Fixup FLASH info from ftp buffer`** **`Image length invalid, image is damaged, abort.`** **`NetBtLd>`**

GET

FUNCTION:	Download file from ftp server
SYNTAX:	**`get <filename>`** where: get — command <filename> — parameter: string name of image file to be downloaded, or path and name of image file to be downloaded
DESCRIPTION:	To download a file from the ftp server to the local data buffer, the command “get” is used. A successful ftp server login must be carried out before a file can be downloaded and the file must be in binary format. The argument “<filename>” must refer to an existing and accessible file on the server and the syntax must follow the requirements on the server, e.g. case sensitiveness. The argument may also include a path specification, if the server supports this.

HELP or ?

FUNCTION:	Display online help pages
SYNTAX:	`help` `?`
DESCRIPTION:	This command displays the online help pages. The display of the help text varies between the different CPU's reflecting their differences. The syntax of every command and a brief description is shown. The display output pauses after every page. The output can be continued with any key. Entering a "." (period) aborts the help function.

INFO

FUNCTION:	Display system information
SYNTAX:	`info`
DESCRIPTION:	The command "info" is used to display an information summary for the running system. The CPU type, the board type, and the detected FLASH layout are displayed.

LF

FUNCTION:	Load Flash
SYNTAX:	**lf [-o[=]<offset> [-k]]** **… [-m[=]<adr> -l[=]<len>]** where: lf — command -o — option: offset <offset> — parameter: value: hexadecimal program to FLASH offset of ... -k — option: keep retain surrounding contents -m — option: memory (address) <adr> — parameter: value: hexadecimal absolute address of image to be programmed -l — option: length <len> — parameter: value: hexadecimal length of image to be programmed
DESCRIPTION:	Without parameters, the FLASH is programmed using the contents of the data buffer. If no image is available in the data buffer, the FLASH programming is terminated. If no offset option (“-o”) is specified the image is considered to be valid and is therefore added along with CRC and length information. If the CRC is determined to be valid during the next startup, the image is copied to the absolute address 0x0 and started at 0x0 after the boot wait time has been exceeded. Normally, the local data buffer holds the image to be programmed. However, if the “-m” and “-l” parameters are specified, the image is programmed from the absolute address specified. If “<offset>” is specified, the contents are programmed exactly at this offset in FLASH. No length and no CRC information is added. The “-k” option can be specified to prevent deletion of the surrounding FLASH contents.

LF

DESCRIPTION:	FLASH memory can only be erased sector-wise. If an image is programmed to a certain offset with the “-o” option, at least this sector (and maybe one or more of the following sectors depending on the size of the image) will be erased. The “-k” option can be used to retain the surrounding data, however, this slows down the operation significantly. To achieve fast programming of parameter images without destroying other FLASH contents, the data should be placed at a sector boundary and the sector(s) must not contain any other data or executable images. If organized this way, use of the “-k” option can be avoided. Note: The “lf” command cannot be used to program the NetBootLoader.
USAGE:	Program FLASH from data buffer and add CRC and image length COMMAND / RESPONSE (none): **`lf`**
	Program FLASH from data buffer to offset 0xF4240 COMMAND / RESPONSE (none): **`lf -o=f4240`**
	Program FLASH from visible address at 0x87000000 for length of 0x123456 COMMAND / RESPONSE (none): **`lf -m=87000000 -l=123456`**
	Program FLASH from data buffer to offset 0xF4240 and retain adjacent FLASH contents COMMAND / RESPONSE (none): **`lf -o=f4240 -k`**

27854.02.VC.060131/114715

LOGIN

FUNCTION:	Initiate ftp server session
SYNTAX:	`login <ip-of-host> <username> [<password>]` where: login — command <ip-of-host> — parameter: value: numerical string IP address of host: nnn.nnn.nnn.nnn <username> — parameter: value: string ftp server “username” <password> — parameter: value: string user’s password
DESCRIPTION:	The command “login” is used to establish an ftp server session. The “<ip-of-host>” must be specified as four numbers separated by single dots. The “<password>” parameter is not necessary if the server does not request one.
USAGE:	Initiate ftp server session COMMAND / RESPONSE: `login 192.168.47.12 johndoe mypassword` (Response is dependent on the server accessed)

LOGOUT

FUNCTION:	Terminate telnet session with NetBootLoader
SYNTAX:	`logout`
DESCRIPTION:	A remote telnet session will be terminated with the command “logout”. No application is loaded and started if the session is terminated with “logout”. The NetBootLoader waits for a new session to be initiated or for a command entry from the serial console.

27854.02.VC.060131/114715

LS

FUNCTION:	Display listing of the current ftp server directory
SYNTAX:	`ls`
DESCRIPTION:	To display a listing of the current ftp server directory the command “ls” is used. This command downloads the listing to the data buffer and then the listing is displayed. Any previously loaded image in the data buffer is overwritten. If an attempt is then made to program the FLASH after the “ls” command has been issued it will fail.

MD

FUNCTION:	Display visible memory
SYNTAX:	`md [<adr>]` where: md command <adr> parameter: value: hexadecimal starting address of a visible memory area
DESCRIPTION:	To display a visible memory area the command “md” is used. The first time the command “md” is issued, visible memory contents starting at the address 0x0 are displayed if no “<adr>” parameter is used. If issued again without the “<adr>” parameter, the display starts with the end address of the previous display. Data is displayed as hexadecimal 32-bit words and as ASCII dump.

NET

FUNCTION: Set or display the parameters for the Ethernet interface

SYNTAX:

```
net [<ip-addr>][-netmask <netmask>]
…[-gw <gateway>][-f]
```

where:

net	command
<ip-addr>	parameter: value: numerical string IP address of CPU board: nnn.nnn.nnn.nnn
-netmask	option: netmask
<netmask>	parameter: value: numerical string netmask of CPU board: nnn.nnn.nnn.nnn
-gw	option: gateway
<gateway>	parameter: value: numerical string gateway address for network: nnn.nnn.nnn.nnn
-f	option: force CRC update

DESCRIPTION: To set or display the parameters of the Ethernet interface the command “net” is used.

Initially the CPU board does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done from the TERM interface using the command “net ... -f”.

Using the “-f” option forces a CRC to be performed and stored along with the other configuration parameters in the serial EEPROM.

Once the initialization of the Ethernet interface is done, the CPU board must be restarted for the parameters to take effect. Later changes to the parameters do not require the use of the “-f” option to force a CRC. This is done automatically. Only in the event that the Ethernet interface does not properly initialize, may it be necessary to re-enter the parameters using the “-f” option.

PASSWD

FUNCTION:	Set the telnet password
SYNTAX:	**`passwd [-f | -d]`** where: passwd command -f option: if password is not known -d option: disable disable telnet login (remote access)
DESCRIPTION:	To set the password for telnet sessions with the NetBootLoader the command “passwd” is used. This command is interactive, meaning that after it is issued, the NetBootLoader responds with an appropriate request to the operator which must be properly acknowledged or the operation fails (refer to USAGE below). To set the password in the event it is unknown, use the option “-f”. This is can only be accomplished from the TERM interface and not from the Ethernet interface. With the option “-d”, the remote telnet login can be disabled by invalidating the password.
USAGE:	Set password COMMAND / RESPONSE: **`NetBtLd> passwd`** **`Old Password: *****`** **`New Password: *****`** **`Type again   : *****`** **`NetBtLd>`** (The old password must be known)
	Set password when the old password is not known COMMAND / RESPONSE: **`NetBtLd> passwd`** **`New Password: *****`** **`Type again   : *****`** **`NetBtLd>`**

PCI

FUNCTION:	Display PCI information
SYNTAX:	**`pci`**
DESCRIPTION:	The command “pci” is used to display detailed information on all detected PCI devices. The bus number, device number, function number, vendor, and device ID’s are displayed together with the configured base addresses and the assigned IRQ number.

PF

FUNCTION:	Set or display the serial port parameters (format)	
SYNTAX:	**pf [<port> [<baud>][/[<bitschar>]** **…[/[<parity>][/<stops>]]]]**	
	where:	
	pf	command
	<port>	parameter: string: “term” or “ser0” defines serial port to be configured
	<baud>	parameter: value: numeric: “50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 115200” defines the baud rate for the port
	<bitschar>	parameter: value: numeric: “7” or “8” defines the number of bits per character
	<parity>	parameter: string: “n” (none), “o” (odd), “e” (even) defines parity to be used
	<stops>	parameter: value: number: “1”, “2” defines number of stop bits

PF

DESCRIPTION:	To set or display the operational parameters for the available serial interfaces the command “pf” is used. At startup the settings for the “TERM” and “SER0” interfaces are always set to the default values (9600/8/n/1). This is to avoid a possible system lockout. If other settings are required during operation of the NetBootLoader they may be made. If changes are made, it must be ensured that corresponding parameters are used for the operator consle. Issuing this command without parameters being specified will display the current serial port settings. Syntax-wise, no spaces are permitted between the parameters and they must be separated with a slash. Not all parameters must be specified, but the “/” characters must be present to distinguish the different parameters from each other. The sequence can be aborted after every option.
USAGE:	Set “TERM” to 300 Baud, 7 Bits/char, odd parity, and 2 stop bits COMMAND / RESPONSE (none): **`pf term 300/7/o/2`**
	Set the bits per character parameter of “SER0” to 7 COMMAND / RESPONSE (none): **`pf ser0 //7`**
	Set the stop bits parameter of “SER0” to 2 COMMAND / RESPONSE (none): **`pf ser0 ///2`**

27854.02.VC.060131/114716

PING

FUNCTION: Verify operability of the Ethernet interface

SYNTAX:

```
ping <ip_addr> [-c <count>] [-s <size>]

… [-w <wait>]
```

where:

ping	command
<ip-addr>	parameter: value: numerical string IP address of target: nnn.nnn.nnn.nnn
-c	option: count
<count>	parameter: value: numeric: “[n ...]n” number of packets to send
-s	option: size
<size>	parameter: value: numeric: “[n ...]n”: bytes size of packet to send
-w	option: wait
<wait>	parameter: value: numeric: “[n ...]n”: seconds wait time between packets

DESCRIPTION: To verify the operational status of the Ethernet interface the command “ping” is used. This command tests the network connection and target server’s ability to respond.

If no other parameters are specified, four requests will be sent. This can be changed with the parameter “-c”. The typical size of a ping packet can be changed with the parameter “-s” and the time between requests, which is typically one second, can be changed with the parameter “-w”.

Reponses to the “ping” command are dependent on the performance of the network.

USAGE: Send four packets

COMMAND / RESPONSE:

```
ping 192.192.158.7
```

Send ten packets, 100 bytes long, and wait two seconds between packets

COMMAND / RESPONSE:

```
ping 192.192.158.7 -c 10 -s 100 -w 2
```

PUT

FUNCTION:	Upload contents of the data buffer to the ftp server.
SYNTAX:	**`put <filename>`** where: put — command <filename> — parameter: string file name to be used for contents of data buffer to be uploaded
DESCRIPTION:	To upload the contents of the data buffer to a file on an ftp server, the command “put” is used. The file indicated by the parameter “<filename>” is created on the server. In the event that a file with this name already exists, its contents will be overwritten.

PWD

FUNCTION:	Display the current ftp server directory.
SYNTAX:	**`pwd`**
DESCRIPTION:	If a ftp connection has been established with the “login” command, the command “pwd” is used to display the complete path of the current directory on the ftp server.

RS

FUNCTION:	Reset the system
SYNTAX:	**`rs`**

RS

DESCRIPTION:	To permit the operator to force a restart of the system, the command “rs” is used. This command terminates the NetBootLoader command interpreter and resets the entire system, generating a system reset with the onboard watchdog. If this command is issued over a remote telnet connection, the telnet session is terminated prior to the generation of the reset.

SF

FUNCTION:	Store FLASH contents to data buffer
SYNTAX:	`sf -o[=]<offset> -l[=]<length>` where: sf — command -o — option: offset <offset> — parameter: value: hexadecimal relative offset to start of FLASH contents to be stored to the data buffer -l — option: length <length> — parameter: value: hexadecimal length of FLASH contents to be stored to the data buffer
DESCRIPTION:	With the command “sf” a selected portion of the FLASH contents may be copied to the local data buffer, e.g. for a subsequent upload to the ftp server with the “put” command. The “<offset>” parameter refers to the relative offset within the FLASH area similar to the “lf” command. The parameter “<length>” specifies the length to store.
USAGE:	Store 64 kB of FLASH contents to the data buffer beginning at an offset of 1 MB COMMAND / RESPONSE (none): `sf -o=100000 -l=10000`

SL

FUNCTION:	Download Motorola S-Records to data buffer
SYNTAX:	`sl [-o[=]<offset>] [-u]` where: sl command -o option: offset <offset> parameter: value: hexadecimal: unsigned offset to be subtracted from each record's address -u option: upper
DESCRIPTION:	With the command “sl” Motorola S-Records are downloaded to the data buffer and the record addresses modified accordingly as required for SDRAM operation (for copying to 0x0). The “<offset>” parameter may be used to change the record base to 0x0. The “-u” option selects the SER0 interface as source for the S-Records.
USAGE:	Download S-Records to data buffer and reduce each record’s address by 0x10000. COMMAND / RESPONSE (none): `sl -o=10000`

VER

FUNCTION:	Display version number
SYNTAX:	`ver`
DESCRIPTION:	The command “ver” displays the actual version number of the NetBootLoader.

27854.02.VC.060131/114717

Appendix *A*

I²C Controller

27854.02.VC.060131/114717

This page has been intentionally left blank.

A. I²C Controller

The I²C controller has been designed to simplify and decouple the direct communication between the CPU and the I²C devices. It is realized using the BPCC and reduces the number of instructions required to transfer user data to and from the I²C devices, such as memory devices, temperature sensors, real-time clocks, etc. Thus, the only instructions to be executed are the writing and reading of the individual bytes and fulfilling the start and stop conditions. Further instructions such as reading and writing of registers or entire blocks must be realized at the software level. The time-dependent conditions are fulfilled by the controller. In the servicing software only the corresponding bits must be polled (Poll mode) or the user must wait for the corresponding interrupt. For further information on the I²C protocol, refer to the I²C specification from Philips Semiconductors, Rev. 02.

The operation of the controller is represented using the following example.

The EEPROM (8 kB capacity, 2 address bytes, I²C address: 0xa0) implemented on the EB42x module is used as the device to be addressed. The contents of the 0x10 and 0x11 EEPROM addresses are read out in the Poll mode.

The basic procedure for accessing an I²C device is described below.

- Set the EEPROM internal address register to 0x10
  - Generate Start Condition
  - Select device 0xa0 for write access
  - Send High Byte Address: 0x0
  - Send Low Byte address: 0x10
  - Generate Stop Condition
- Read out 2 bytes
  - Generate Start Condition
  - Select device 0xa0 for read access
  - Read 1 byte, result: contents of 0x10 address
  - Generate acknowledge
  - Read 1 byte, result: contents of 0x11 address
  - Generate “no” acknowledge
  - Generate Stop Condition

The following steps must be taken to operate the controller.

First, the speed must be configured, for example to 100 kHz. This can be made by writing 0x1 to the “External I²C Divider Register”, address 0x5100000d.

The controller is being operated using the “External I²C Command Register”, address 0x5100000b, referred to as EICR.

The writing and reading of data is made using the “External I²C Data Register”, address 0x5100000c, referred to as EIDR.

The controller is being operated using the EICR, whereby the next transaction can be defined by setting the mode bits and can be started by setting the strobe bits. By reading the EICR, bit “Busy”, the user can determine when the controller has finished the transaction and can start a new transaction. The waiting for “Busy” = 0 required after every step is referred to as Wait For Busy Zero (WFBZ).

Realization of the procedure described above.

- Generate Start Condition
  - write 0x01 to EICR
  - write 0x81 to EICR
  - WFBZ
- Select EEPROM for write access
  - write 0xa0 to EIDR
  - write 0x02 to EICR
  - write 0x82 to EICR
  - WFBZ

By reading out the EICR, it can now be determined whether the EEPROM has been successfully selected, i.e. if an ACK has been received.

- Send High Byte Address: 0x0
  - write 0x00 to EIDR
  - write 0x02 to EICR
  - write 0x82 to EICR
  - WFBZ
- Send Low Byte Address: 0x10
  - write 0x10 to EIDR
  - write 0x02 to EICR
  - write 0x82 to EICR
  - WFBZ
- Generate Stop Condition
  - write 0x00 EICR
  - write 0x80 to EICR
  - WFBZ
- Generate Start Condition
  - write 0x01 to EICR
  - write 0x81 to EICR
  - WFBZ
- Select EEPROM for read access
  - write 0xa1 to EIDR
  - write 0x02 to EICR
  - write 0x82 to EICR
  - WFBZ
- Read 1 byte with acknowledgement (ACK)
  - write 0x43 to EICR
  - write 0xc3 to EICR
  - WFBZ
  - read received byte from EIDR
- Read 1 further byte without acknowledgement
  - write 0x03 to EICR
  - write 0x83 to EICR
  - WFBZ
  - read received byte from EIDR

27854.02.VC.060131/114717

- Generate Stop Condition
  - write 0x00 to EICR
  - write 0x80 to EICR
  - WFBZ

Instead of active polling of the “Busy” bit, an interrupt can be activated (IRQ_EN in EICR). By writing 0x04 in the EICR (IACK bit), the interrupt can be acknowledged. The interrupt must be acknowledged by writing 0x04 to the EICR (Bit IACK) in the Interrupt Service Routine.

Moreover, when writing the mode and the strobe bits it is important to pay attention not to accidentally delete the IRQ_EN.

This page was intentionally left blank.

27854.02.VC.060131/114717

Appendix **B**

WDOG Functionality

27854.02.VC.060131/114718

This page has been intentionally left blank.

B. WDOG Functionality

To assist in understanding the functionlality of the EB42x watchdog (WDOG), the following functional state diagram is provided.

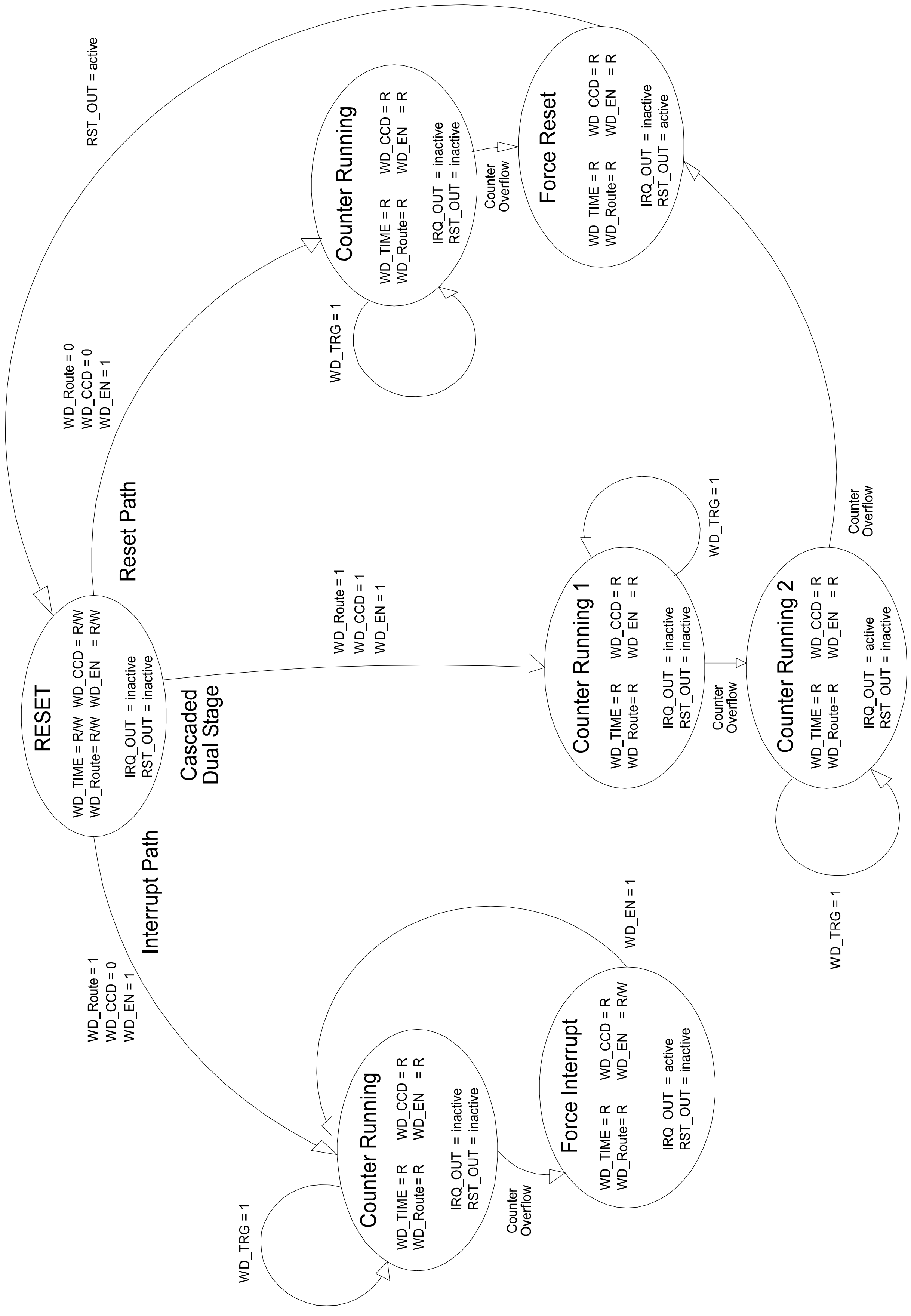


Figure B-1: WDOG Functional States

27854.02.VC.060131/114718

