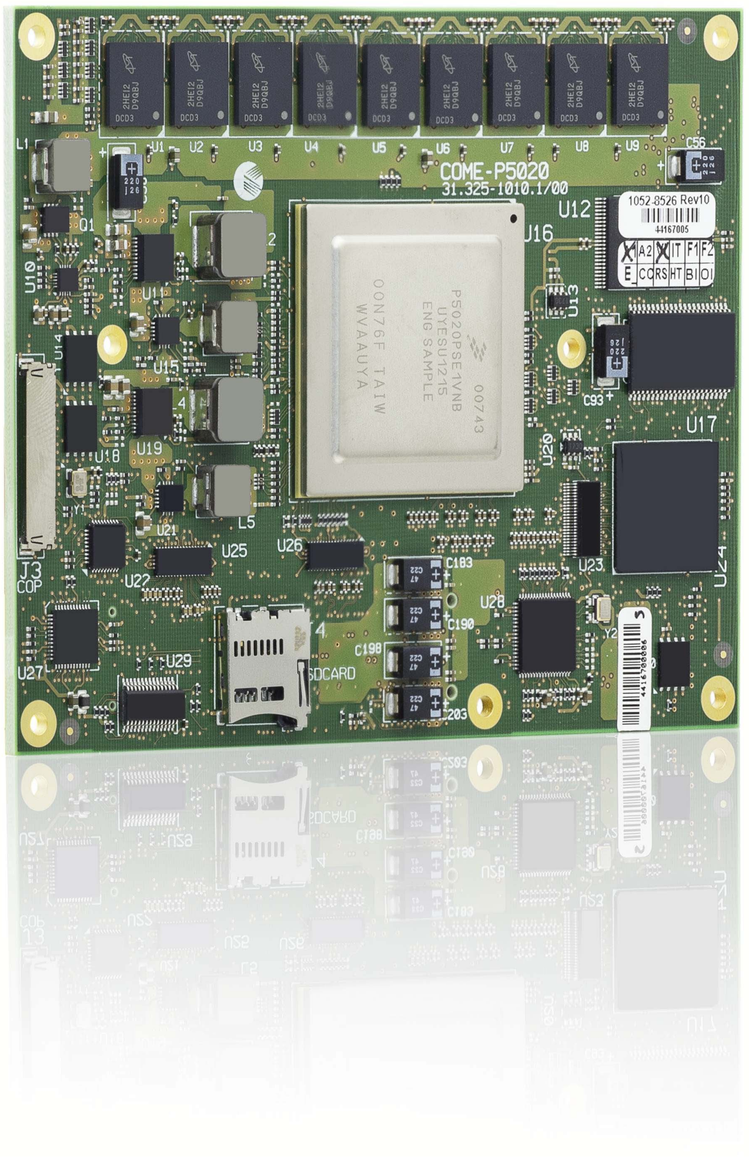
» User Guide «



COMe-bP5020

Doc. ID: 1053-4773, Rev. 1.0

Date: January 22, 2013

The pulse of innovation

Revision History

Publication Title:	COMe-bP5020 “COM Express® Power Architecture® CPU Module”	
Doc. ID:	1053-4773	
Rev.	Brief Description of Changes	Date of Issue
1.0	Initial issue	22-Jan-2013

Imprint

Kontron Europe GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Europe GmbH
Sudetenstraße 7
D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

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

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

Disclaimer

All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

CONTENTS

TABLES

FIGURES

1 Introduction

1.1 COMe-bP5020 Overview

The COMe-bP5020 is a COM Express® form factor compliant Power Architecture® processor module based on the Freescale™ QorIQ™ 64-bit P5020 processor.

Designed in the COM Express® basic (95 mm x 125 mm) form factor the module incorporates the Freescale™ QorIQ™ P5020 dual-core Power Architecture® processor operating at 2.0 GHz - other processor versions (P5010 and P3041) and operating speeds are available on request. Featuring 64-bit technology, it integrates up to 8 GByte of soldered DDR3 SDRAM at 1300 MHz and ECC support. Two additional MBytes of shared third level cache facilitate core-to-core communications to minimize accesses to main memory.

Up to 2 GB of NAND Flash as well as a socket for a MicroSD card provide flexible and reliable storage space for application data. In terms of I/Os, the module interfaces the QorIQ-specific I/Os to the carrier board. In addition to USB 2.0 ports there are also UART (TxD, RxD, RTC and CTS) and Gigabit Ethernet interfaces.

Flexible interface support is guaranteed by 18 SERDES lanes, which can be configured according to application-specific needs. A comprehensive range of different combinations, for example as PCIe x4, sRIO x4, Serial Gigabit Media Independent Interface (SGMII), XAUI and SATA interfaces is available.

The COMe-bP5020 with its innovative Data Path Acceleration Architecture (DPAA) assures that even heavy network traffic does not affect the processing performance of the cores. With DPAA the cores are relieved of the common packet-handling tasks, which leaves more headroom for the relevant processing even at full load.

The COMe-bP5020 targets high-bandwidth telecommunication and data processing applications. With its long-term availability of more than 10 years, it is also a very good choice for use in long life cycle network applications in the medical, military and transportation markets.

1.2 Board Diagrams

Figure 1: COMe-bP5020 Block Diagram

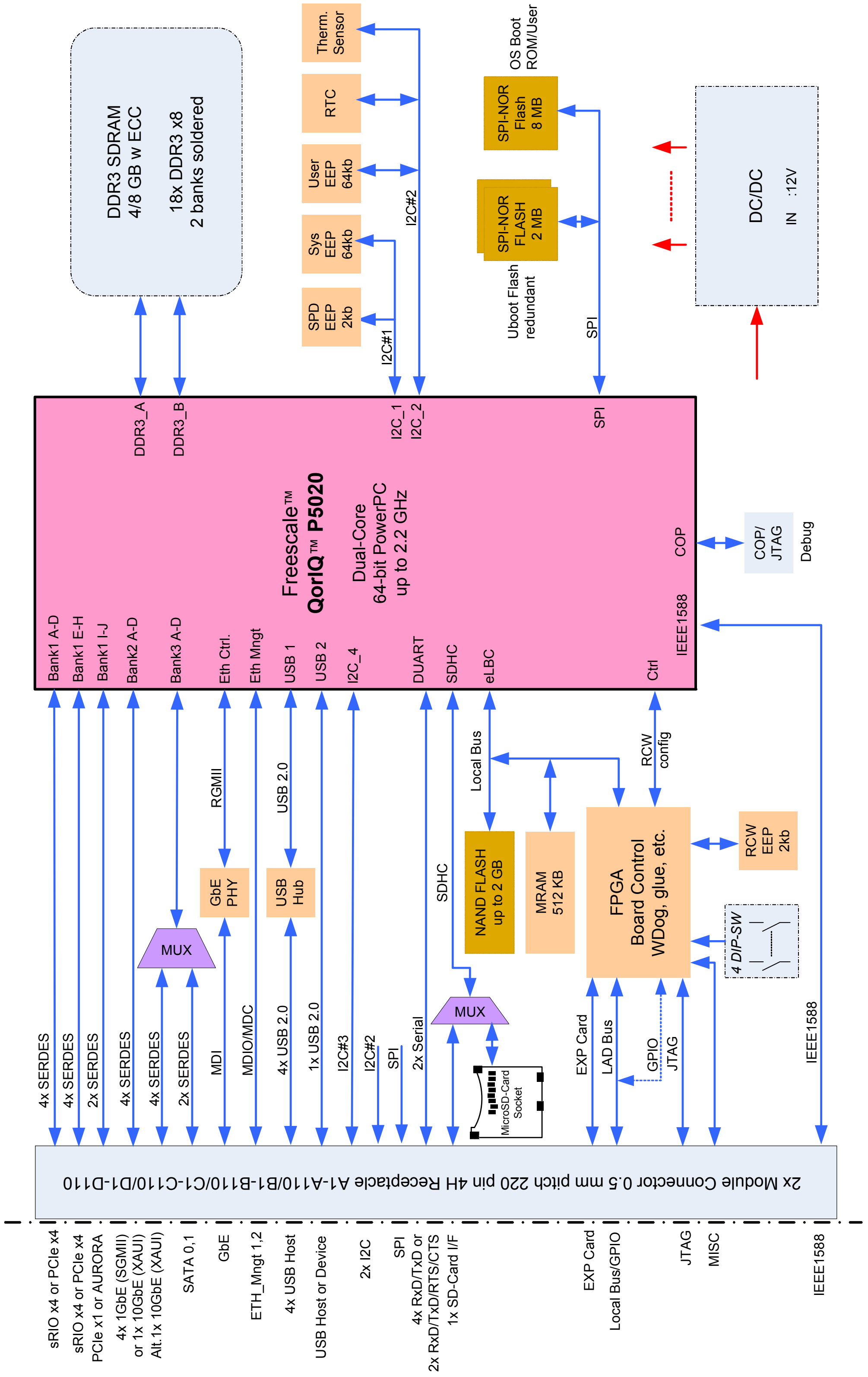


Figure 2: COMe-bP5020 Board Layout Top View

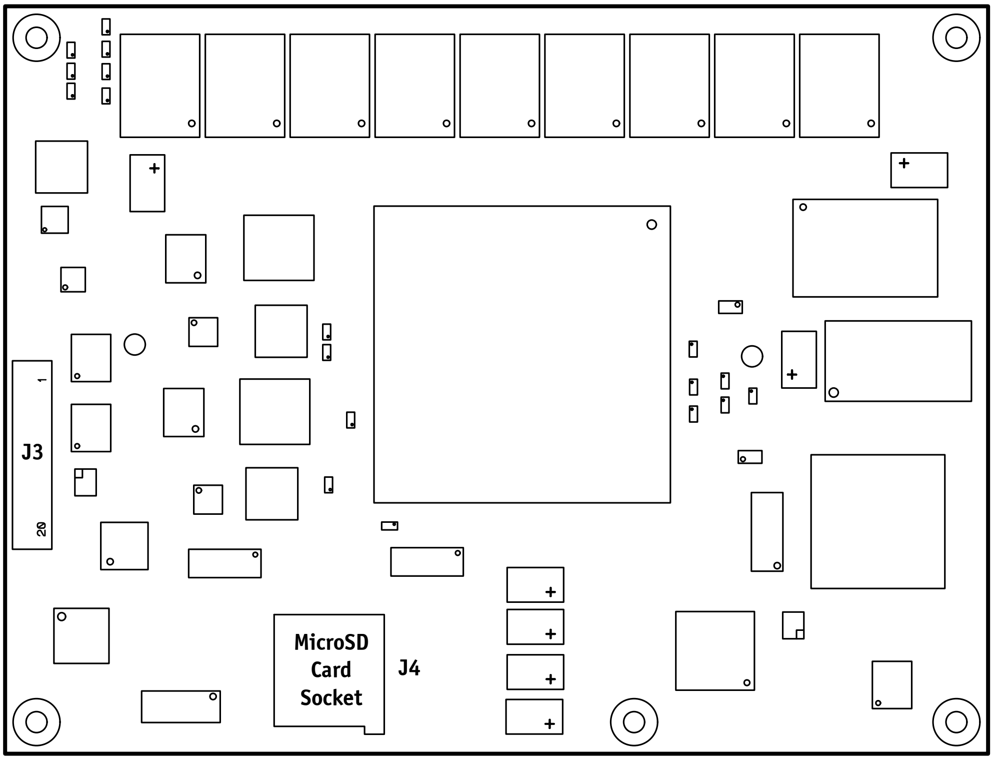
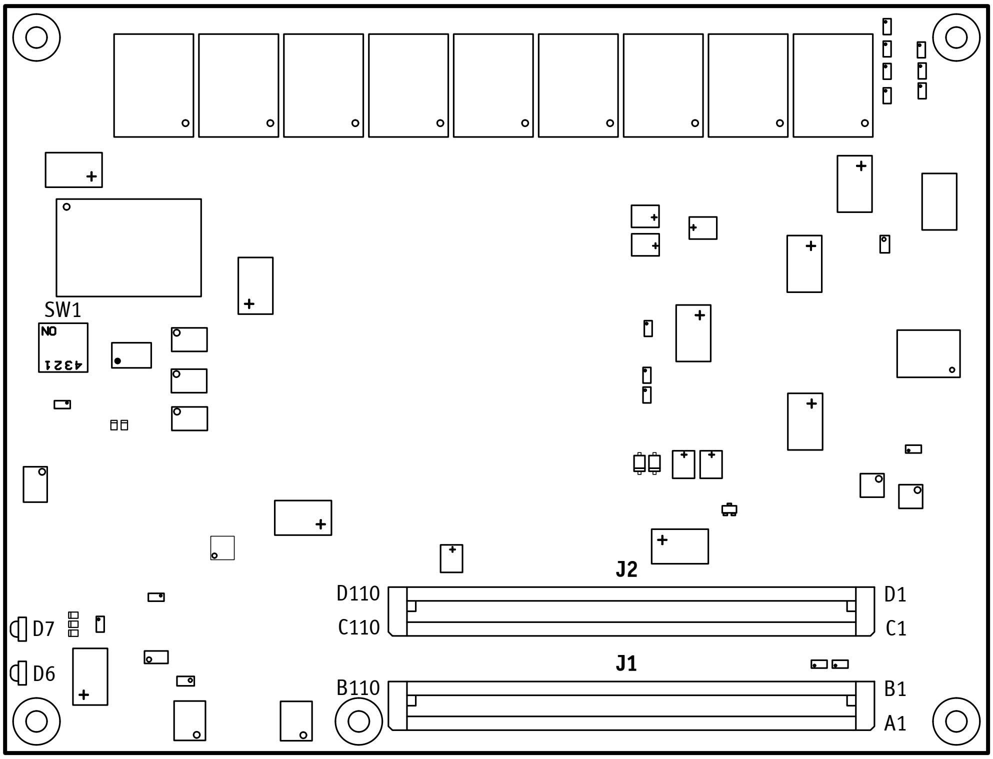


Figure 3: COMe-bP5020 Board Layout Bottom View



1.3 Technical Specifications

Table 1: COMe-bP5020 Main Specifications

COMe-bP5020		SPECIFICATIONS
PROCESSOR	CPU	The COMe-bP5020 supports the following microprocessor: » Freescale™ QorIQ™ P5020 processor, 2.0 GHz (other operating speeds and processor variants (P5010/P3041) are available on request) Further processor features: » Two 64-bit execution cores » System Memory interface with optimized support for dual-channel DDR3 SDRAM memory at 1300 MHz with ECC for the QorIQ™ P5020 processor with 2.0 GHz CPU frequency
	Integrated Controllers	Controllers integrated in the CPU and utilized by the COMe-bP5020: » eSDHC, eLBC, DUART, dTSEC, PCIe, sRIO, SPI, I2C
MEMORY	Memory	Main memory: » Up to 8 GB, dual-channel DDR3 SDRAM memory with ECC running at up to 1300 MHz Cache structure: » 64 kB L1 cache for each core » 32 kB instruction cache » 32 kB data cache » 512 kB backside L2 cache for each core » 2 MB shared L3 CoreNet platform cache (1 MB per memory channel) Flash memory: » Two SPI boot flashes (2 x 2 MB) for U-Boot selectable via the DIP switch » One 8 MB SPI flash for operating system or application Mass storage device: » Up to 2 GB NAND flash via an integrated/embedded NAND flash controller » Up to 32 GB microSDHC flash via an integrated SDHC controller MRAM memory: » 512 kB of non-volatile memory Two serial EEPROMs with 64 kbit: » One for system data storage » One free for user data storage

Table 1: COMe-bP5020 Main Specifications (cont'd)

COMe-bP5020		SPECIFICATIONS
INTERCONNECTION	Gigabit Ethernet	Up to five Gigabit Ethernet ports: » One Gigabit Ethernet port through COMe MDI interface » Up to four Gigabit Ethernet ports through SGMII interface
	SATA	Two SATA ports
	SRIO	Up to two x4 Serial RapidIO interfaces operating in host or agent configuration, depending on configuration
	PCI Express	Up to two x4 PCI Express interface operating in root complex configuration If interface is configured for PCI Express, SRIO is not possible
	Debug Interface	One debug port
	Serial Interface	Up to four serial ports: » 2x 4-wire UART interfaces (RxD, TxD, RTS, CTS), or » 4x 2-wire UART interfaces (RxD, TxD)
	GPIO	Up to 12 GPIOs
Connectors	Onboard Connectors	Two 220-pin connectors for interfacing with a carrier board
		One JTAG/COP connector, J3, for debugging
	microSD card Socket	Standard microSD socket, J9, accepts microSD and microSDHC cards
Switch	DIP Switch	One DIP switch for board configuration, SW1, consisting of four switches
LEDs	Module Health Monitor LEDs	LED7: indicates Reset Status LED9: indicates "Power Good" status
TIMER	Watchdog Timer	Software-configurable, two-stage Watchdog with programmable timeout ranging from 125 ms to 4096 s in 16 steps Serves for generating IRQ or hardware reset
	System Timer	There are several timers implemented in the CPU. For further information regarding these timers, refer to the CPU reference manual from Freescale™.
THERMAL	Thermal Monitoring	One onboard temperature sensor for monitoring the board temperature
GENERAL	Power Consumption	Refer to Chapter 4, "Power Considerations" for information related to the power consumption of the COMe-bP5020.
	Temperature Range	Operational: Refer to Chapter 5, "Thermal" for further information Storage: -40°C to +70°C
	Mechanical	COM Express® basic
	Dimensions	125 mm x 95 mm
	Board Weight	99 grams (without heat spreader) 220 grams (with heat spreader)

Table 1: COMe-bP5020 Main Specifications (cont'd)

COMe-bP5020		SPECIFICATIONS
SOFTWARE	Bootloader	DENX U-Boot (Universal Boot Loader) with Kontron-specific modifications to support the COMe-bP5020 requirements
	Operating Systems	The board is offered with various Board Support Packages including VxWorks and Linux operating systems. For further information concerning the operating systems available for the COMe-bP5020, please contact Kontron.

1.4 Standards

The COMe-bP5020 complies with the requirements of the following standards.

Table 2: Standards

COMPLIANCE	TYPE	STANDARD	TEST LEVEL
CE	Emission	EN55022 EN61000-6-3	--
	Immission	EN55024 EN61000-6-2	--
	Electrical Safety	EN60950-1	--
Mechanical	Mechanical Dimensions	COM Express® basic	--
Environmental and Health Aspects	Vibration (sinusoidal, operating)	tbs	tbs
	Shock (operating)	tbs	tbs
	Climatic Humidity	IEC60068-2-78	93% RH at 40°C, non-condensing (see notice below)
	WEEE	Directive 2002/96/EC	Waste electrical and electronic equipment
	RoHS-II	Directive 2011/65/EC	Restriction of the use of certain hazardous substances in electrical and electronic equipment

NOTICE

Kontron performs comprehensive environmental testing of its products in accordance with applicable standards.

Customers desiring to perform further environmental testing of Kontron products must contact Kontron for assistance prior to performing any such testing. This is necessary, as it is possible that environmental testing can be destructive when not performed in accordance with the applicable specifications.

In particular, for example, boards without conformal coating must not be exposed to a change of temperature exceeding 1K / minute, averaged over a period of not more than five minutes. Otherwise, condensation may cause irreversible damage, especially when the board is powered up again.

Kontron does not accept any responsibility for damage to products resulting from destructive environmental testing.

1.5 Related Publications

Table 3: Related Publications

SPECIFICATION / ORGANIZATION	PUBLICATION
COM Express®	PICMG® COM.0, COM Express® Module Base Specification, Revision 2.0, August 8, 2010 Freescale™, Kontron and Emerson Common Pinout Definition
PCI Express	PCI Express Base Specification Revision 2.0, Dec. 20, 2006
Serial RapidIO	RapidIO™ Interconnect Specification Part 6: LP-Serial Physical Layer Specification, Rev. 2.0.1, March 2008
Serial ATA	Serial ATA International Organization: Serial ATA Revision 2.6, 15th February 2007
Ethernet	IEEE802.3: Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification, Clause 22 and Clause 45
Platform Firmware	DENX “U-Boot” (Universal Boot Loader) online documentation at www.denx.de
Kontron	Kontron’s Product Safety and Implementation Guide, ID 1021-9142

2 Functional Description

2.1 Processor

The COMe-bP5020 supports the high-performance, 64-bit, 45nm dual-core Freescale™ QorIQ™ P5020 processor with the following functions and features:

- Two e5500 cores built on Power Architecture® technology, each with a private 512-Kbyte backside cache, running up to 2.0 GHz clock speed
- 2x 1-Mbyte shared CoreNet platform cache
- Two 64-bit DDR3 SDRAM memory controllers with ECC and chip-select interleaving support
- Data path acceleration architecture incorporating acceleration for Packet-/Buffer- and Queue-Management
- One 10 Gbps Ethernet (XAUI) controller
- Five 1 Gbps Ethernet controllers
- Four PCI Express 2.0 controllers/ports running at up to 5 Gbps
- Two serial RapidIO controllers/ports version 1.3 with features of 2.1 running at up to 5 Gbps
- Two SATA 2.0 interfaces supporting 1.5 and 3.0 Gbps operation
- Two USB 2.0 controllers with integrated PHY
- One SD/MMC controller
- One SPI controller
- Four I2C controllers
- Two DUARTs
- One enhanced local bus controller
- Multicore programmable interrupt controller
- Two 4-channel DMA engines

2.2 Memory

2.2.1 DDR3

The COMe-bP5020 supports a soldered, dual-channel (72-bit), Double Data Rate (DDR3) memory with Error Checking and Correcting (ECC) running at up to 1300 MHz (memory error detection and reporting of 1-bit and 2-bit errors and correction of 1-bit failures). The available memory configuration can be either 4 GB or 8 GB.

2.2.2 Flash Memory

2.2.2.1 SPI Boot Flash

Two 2 MB SPI boot flashes are provided for two separate U-Boot images: a standard SPI boot flash and a recovery SPI boot flash. The fail-over mechanism for the U-Boot recovery can be controlled via the DIP switch SW1, switch 2. Refer to Chapter 6.8 for further information.

The SPI boot flashes include a hardware write protection option. If write protection is enabled, writing to the SPI boot flashes is not possible.

NOTICE

The U-Boot code and settings are stored in the SPI boot flashes. Changes made to the U-Boot settings are available only in the currently selected SPI boot flash. Thus, switching over to the other SPI boot flash may result in operation with different U-Boot code and settings.

2.2.2.2 SPI OS/User Flash

There is an 8 MB soldered flash memory available for the OS or application usage.

2.2.2.3 NAND Flash

The COMe-bP5020 supports up to 2 GB of soldered SLC-based NAND flash memory. It is optimized for embedded systems providing high performance, reliability and security.

2.2.2.4 MRAM Memory

The COMe-bP5020 supports 512 kB of MRAM memory (Magnetorestrictive Random Access Memory) for fast non-volatile data storage.

2.2.2.5 SDHC Socket

The COMe-bP5020 has a microSDHC card socket, J4, which accepts microSD and microSDHC cards up to 32 GB. If used, the card must be installed prior to installation of the COMe-bP5020 in a system.

If the SDHC interface is routed to the COM Express® connector (via the U-Boot “sconf” command), the onboard socket J4 cannot be used.

2.2.3 System/User Data EEPROMs

The COMe-bP5020 provides two 64-kBit EEPROMs: one for system data storage and one which is free for user data storage. The user data EEPROM is accessible via the OS or an application. The system data EEPROM is reserved for system usage.

2.3 Timer

The COMe-bP5020 is equipped with the following timer:

- Real-Time Clock (RTC)

The onboard high-precision real-time clock RV-8564-C2 (RTC) is register-compatible with the PCF8564A RTC from Philips/NXP. In addition, it provides a very rigid frequency tolerance at low power consumption. The COMe-bP5020 does not include a 3 V lithium battery or a GoldCap power source for RTC backup. Power for the RTC is supplied by the carrier via the VCC_RTC pin.

2.4 Watchdog Timer

The COMe-bP5020 provides a Watchdog timer that is programmable for a timeout period ranging from 125 ms to 4096 s in 16 steps. Failure to trigger the Watchdog timer in time results in a interrupt or a system reset or both. In dual-stage mode, it results in a combination of both interrupt and reset if the Watchdog is not serviced. A hardware status flag will be provided to determine if the Watchdog timer generated the reset. Refer to the Watchdog Timer Control Register (WTIM) in Chapter 3 for further information.

There are four possible modes of operation involving the Watchdog timer:

- Timer only mode
- Reset mode
- Interrupt mode
- Dual stage mode

At power on the Watchdog is not enabled. If required, the appropriate bits of the Watchdog Timer Control Register must be set according to the application requirements. To operate the Watchdog, the mode and time period required must first be set and then the Watchdog enabled. Once enabled, the Watchdog can only be disabled or the mode changed by powering down and then up again. To prevent a Watchdog timeout, the Watchdog must be retriggered before timing out. This is done by writing a ‘1’ to the WTR bit. In the event a Watchdog timeout does occur, the WTE bit is set to ‘1’. What transpires after this depends on the mode selected.

The four operational Watchdog timer modes can be configured by the WMD[1:0] bits, and are described as follows:

Timer only mode - In this mode the Watchdog is enabled using the required timeout period. Normally, the Watchdog is retriggered by writing a ‘1’ to the WTR bit. In the event a timeout occurs, the WTE bit is set to ‘1’. This bit can then be polled by the application and handled accordingly. To continue using the Watchdog, write a ‘1’ to the WTE bit, and then retrigger the Watchdog using WTR. The WTE bit retains its setting as long as power is not cycled (off - on). Therefore, this bit may be used to verify the status of the Watchdog.

Reset mode - This mode is used to force a hard reset in the event of a Watchdog timeout. In addition, the WTE bit is not reset by the hard reset, which makes it available if necessary to determine the status of the Watchdog prior to the reset.

Interrupt mode - This mode causes the generation of an interrupt in the event of a Watchdog timeout. The interrupt handling is a function of the application. If required, the WTE bit can be used to determine if a Watchdog timeout has occurred.

Dual stage mode - This is a complex mode where in the event of a timeout two things occur: 1) an interrupt is generated, and 2) the Watchdog is retriggered automatically. In the event a second timeout occurs following the first timeout, a hard reset will be generated. The second timeout period is the same as the first. If the Watchdog is retriggered normally as specified above, operation continues. The interrupt generated at the first timeout is available to the application to handle the first timeout if required. As with all of the other modes, the WTE bit is available for application use.

Pin B27 on the COM Express® J1 connector offers a signal that can be asserted when a Watchdog timer has not been triggered within time. It can be configured to any of the 2 stages. Deassertion of the signal is automatically done after reset. If deassertion during runtime is necessary please contact Kontron for further assistance.

2.5 Connectors

2.5.1 COM Express® Connectors

Table 4: Connector J1 Row A Pinout

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
A1	GND	PWR	PWR		
A2	GBE0_MDI3-	GigE MDI	DP-I/O		logically connected to dTSEC5
A3	GBE0_MDI3+	GigE MDI	DP-I/O		logically connected to dTSEC5
A4	GBE0_LINK100#	GigE MDI	O-3.3		8mA max.
A5	GBE0_LINK1000#	GigE MDI	O-3.3		8mA max.
A6	GBE0_MDI2-	GigE MDI	DP-I/O		logically connected to dTSEC5
A7	GBE0_MDI2+	GigE MDI	DP-I/O		logically connected to dTSEC5
A8	GBE0_LINK#	GigE MDI	O-3.3		8mA max.
A9	GBE0_MDI1-	GigE MDI	DP-I/O		logically connected to dTSEC5
A10	GBE0_MDI1+	GigE MDI	DP-I/O		logically connected to dTSEC5
A11	GND	PWR	PWR		
A12	GBE0_MDI0-	GigE MDI	DP-I/O		logically connected to dTSEC5
A13	GBE0_MDI0+	GigE MDI	DP-I/O		logically connected to dTSEC5
A14	N/C	GBE0 CTREF			not needed
A15	SUS_S3#	MISC	O-3.3		for use as general purpose output
A16	SATA0_TX+	SATA	DP-O	AC coupled on module (10n)	
A17	SATA0_TX-	SATA	DP-O	AC coupled on module (10n)	
A18	N/C				
A19	SATA0_RX+	SATA	DP-I	AC coupled on module (10n)	
A20	SATA0_RX-	SATA	DP-I	AC coupled on module (10n)	
A21	GND	PWR	PWR		
A22	N/C		N/C		
A23	N/C		N/C		
A24	Reserved				leave unconnected
A25	N/C		N/C		
A26	N/C		N/C		
A27	BATLOW#	BOARD CTRL	I-3.3	PU 10k 3.3V	
A28.. A30	N/C		N/C		
A31	GND	PWR	PWR		
A32.. A34	N/C		N/C		
A35	THRMTRIP	MISC	O-3.3		for use as general purpose output
A36	DMA2_DDONE0#	DMA	O-3.3	series 0-Resistor	leave open if not needed
A37	DMA2_DACK0#	DMA	O-3.3	series 0-Resistor	leave open if not needed
A38	LWE[1]#	Local Bus	O-3.3	series 0-Resistor	
A39	USB4-	USB	DP-I/O		
A40	USB4+	USB	DP-I/O		
A41	GND	PWR	PWR		
A42	USB2-	USB	DP-I/O		
A43	USB2+	USB	DP-I/O		
A44	USB_2_3_OC	USB	I-3.3	PU	
A45	USB0-	USB	DP-I/O		

Table 4: Connector J1 Row A Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
A46	USB0+	USB	DP-I/O		
A47	VCC_RTC	PWR	PWR 3V		
A48	EXCD0_PERST#	EXP CARD	O-3.3		
A49	EXCD0_CPPE#	EXP CARD	I-3.3	PU 10k 3.3V	
A50	LA16 / GPIO8	Local Bus / GPIO	O-3.3 / I/O-3.3	PU (weak) if configured for GPIO	function depending on SCONF
A51	GND	PWR	PWR		
A52	SERDES_TX5+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 F
A53	SERDES_TX5-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 F
A54	SD_DATA0	SDIO	I/O-3.3	PU 4k7 3.3V	series resistor 33R recommended on carrier
A55	SERDES_TX4+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 E
A56	SERDES_TX4-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 E
A57	GND	PWR	PWR		
A58	SERDES_TX3+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 D
A59	SERDES_TX3-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 D
A60	GND	PWR	PWR		
A61	SERDES_TX2+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 C
A62	SERDES_TX2-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 C
A63	SD_DATA1	SDIO	I/O-3.3	PU 4k7 3.3V	series resistor 33R recommended on carrier
A64	SERDES_TX1+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 B
A65	SERDES_TX1-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 B
A66	GND	PWR	PWR		
A67	SD_DATA2	SDIO	I/O-3.3	PU 4k7 3.3V	series resistor 33R recommended on carrier
A68	SERDES_TX0+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 A
A69	SERDES_TX0-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1 A
A70	GND	PWR	PWR		
A71.. A79	N/C		N/C		
A80	GND	PWR	PWR		
A81	N/C		N/C		
A82	N/C		N/C		
A83	Reserved				leave unconnected
A84	Reserved				leave unconnected
A85	SD_DATA3	SDIO	I/O-3.3	PU 4k7 3.3V	series resistor 33R recommended on carrier
A86	LA18 / GPIO10	Local Bus / GPIO	O-3.3 / I/O-3.3	PU (weak) if configured for GPIO	function depending on SCONF
A87	LA17 / GPIO9	Local Bus / GPIO	O-3.3 / I/O-3.3	PU (weak) if configured for GPIO	function depending on SCONF
A88	SERDES_CK_REF+	SERDES	DP-O	HCSL termination on module	
A89	SERDES_CK_REF-	SERDES	DP-O	HCSL termination on module	
A90	GND	PWR	PWR		
A91	SPI_POWER	PWR	PWR	series 0-Resistor	provide 3.3V to external SPI device
A92	SPI_MISO	SPI	I-3.3	PU 4k7 3.3V	

Table 4: Connector J1 Row A Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
A93	SD_CLK	SDIO	O-3.3	series 33R Resistor	
A94	SPI_CLK	SPI	O-3.3	series 33R Resistor	
A95	SPI_MOSI	SPI	O-3.3	PU 4k7 3.3V	
A96	N/C				
A97	TYPE10#	TYPE	PDS		not connected on module
A98	SER0_TX	UART	O-3.3		UART[1] on CPU
A99	SER0_RX	UART	I-3.3	PU 10k 3.3V	UART[1] on CPU
A100	GND	PWR	PWR		
A101	SER1_TX	UART	O-3.3		UART[2] on CPU
A102	SER1_RX	UART	I-3.3	PU 10k 3.3V	UART[2] on CPU
A103	N/C				
A104.. A109	VCC_12V	PWR	PWR		nominal 12V
A110	GND	PWR	PWR		

Table 5: Connector J1 Row B Pinout

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
B1	GND	PWR	PWR		
B2	GBE0_ACT#	GigE MDI	O-3.3		8mA max.
B3	1588_CLK_OUT	IEEE1588	O-2.5	series resistor 39R	
B4	1588_PULSE_OUT1	IEEE1588	O-2.5	series resistor 39R	
B5	1588_PULSE_OUT2	IEEE1588	O-2.5	series resistor 39R, PU 4k7 2.5V	
B6	1588_ALARM_OUT1	IEEE1588	O-2.5	series resistor 39R	
B7	1588_ALARM_OUT2	IEEE1588	O-2.5	series resistor 39R, PU 4k7 2.5V	
B8	1588_TRIG_IN1	IEEE1588	I-2.5	series resistor 39R, PU 4k7 2.5V	
B9	1588_TRIG_IN2	IEEE1588	I-2.5	series resistor 39R, PU 4k7 2.5V	
B10	1588_CLK_IN	IEEE1588	I-2.5	series resistor 39R, PD 475R	
B11	GND	PWR	PWR		
B12	Reserved				leave unconnected
B13	SMB_CK	SMB	O-3.3	PU 1k 3.3V	I2C[2] Bus on CPU
B14	SMB_DAT	SMB	I/O-3.3	PU 1k 3.3V	I2C[2] Bus on CPU
B15	N/C				
B16	SATA1_TX+	SATA	DP-O	AC coupled on module (10n)	SATA2 controller on CPU
B17	SATA1_TX-	SATA	DP-O	AC coupled on module (10n)	SATA2 controller on CPU
B18	N/C				
B19	SATA1_RX+	SATA	DP-I	AC coupled on module (10n)	SATA2 controller on CPU
B20	SATA1_RX-	SATA	DP-I	AC coupled on module (10n)	SATA2 controller on CPU
B21	GND	PWR	PWR		
B22	N/C		N/C		
B23	N/C		N/C		
B24	PWR_OK	BOARD CTRL	I-3.3	PD 10K	used to start onboard supply
B25	N/C		N/C		
B26	N/C		N/C		
B27	WDT	BOARD CTRL	O-3.3		

Table 5: Connector J1 Row B Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
B28.. B30	N/C		N/C		
B31	GND	PWR	PWR		
B32	N/C				
B33	I2C_CK	I2C	O-3.3	PU 1k 3.3V	I2C[4] Bus on CPU
B34	I2C_DAT	I2C	I/O-3.3	PU 1k 3.3V	I2C[4] Bus on CPU
B35	Reserved				leave unconnected
B36	DMA2_DREQ0#	DMA	I-3.3	series resistor 0R, PU 4,7 3.3V	leave open if not needed
B37	DMA1_DDONE0#	DMA	O-3.3	series resistor 0R	leave open if not needed
B38	USB4_OC#	USB	I-3.3	PU	
B39	DMA1_DACK0#	DMA	O-3.3	series resistor 0R	leave open if not needed
B40	DMA1_DREQ0#	DMA	I-3.3	series resistor 0R, PU 4,7 3.3V	leave open if not needed
B41	GND	PWR	PWR		
B42	USB3-	USB	DP-I/O		
B43	USB3+	USB	DP-I/O		
B44	USB_0_1_OC#	USB	I-3.3	PU	
B45	USB1-	USB	DP-I/O		
B46	USB1+	USB	DP-I/O		
B47	EXCD0_PERST#	EXP CARD	O-3.3		
B48	EXCD0_CPPE#	EXP CARD	I-3.3	PU 10k 3.3V	
B49	SYS_RESET#	BOARD CTRL	I-3.3	PU 10k 3.3V	
B50	CB_RESET#	BOARD CTRL	O-3.3		
B51	GND	PWR	PWR		
B52	SERDES_RX5+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1F
B53	SERDES_RX5-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1F
B54	SD_CMD	SDIO	O-3.3	PU 4k7 3.3V	
B55	SERDES_RX4+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1E
B56	SERDES_RX4-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1E
B57	SD_WP	SDIO	I-3.3	PU 4k7 3.3V	
B58	SERDES_RX3+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1D
B59	SERDES_RX3-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1D
B60	GND	PWR	PWR		
B61	SERDES_RX2+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1C
B62	SERDES_RX2-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1C
B63	SD_CD#	SDIO	I-3.3	PU 4k7 3.3V	
B64	SERDES_RX1+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1B
B65	SERDES_RX1-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1B
B66	WAKE0#	MISC	I-3.3	PU 10k 3.3V	for use as interrupt input
B67	WAKE1#	MISC	I-3.3	PU 10k 3.3V	for use as interrupt input
B68	SERDES_RX0+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1A
B69	SERDES_RX0-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1A
B70	GND	PWR	PWR		
B71.. B79	N/C		N/C		
B80	GND	PWR	PWR		

Table 5: Connector J1 Row B Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
B81.. B83	N/C		N/C		
B84.. B87	VCC_5V_SBY	PWR	PWR		
B88	BIOS_DIS1#	BOARD CTRL	I-3.3	PU-10k	external Boot-Flash Select
B89	JTAG TCK	JTAG/PROG	I-3.3	series resistor 30R. PD 1k	manufacturing use
B90	GND	PWR	PWR		
B91	JTAG TDI	JTAG/PROG		series resistor 39R, PU 4k7 3.3V	manufacturing use
B92	JTAG TMS	JTAG/PROG		series resistor 39R, PU 4k7 3.3V	manufacturing use
B93	JTAG TDO	JTAG/PROG		series resistor 39R	manufacturing use
B94	N/C				
B95	Reserved				leave unconnected
B96	Reserved				leave unconnected
B97	SPI_CS#	SPI	O-3.3		SPI_CS0# of CPU if BIOS_DIS = low / SPI_CS2# of CPU if BIOS_DIS = N/C or high
B98	EMI2_MDC	ETH MGT	O-1.2	PU 180R 1.2V	Ethernet Management Clock for XAUI usage
B99	EMI2_MDIO	ETH MGT	I/O-1.2	PU 330R 1.2V	Ethernet Management In/Out for XAUI usage
B100	GND	PWR	PWR		
B101.. B103	N/C		N/C		
B104.. B109	VCC_12V	PWR	PWR		nominal 12V
B110	GND	PWR	PWR		

Table 6: Connector J2 Row C Pinout

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
C1	GND	PWR	PWR		
C2.. C10	N/C		N/C		
C11	GND	PWR	PWR		
C12.. C16	N/C		N/C		
C17	LOE#	Local Bus	STRAP/ O-3.3	PU 4k7	Local Bus output enable
C18	LWE[0]#	Local Bus	O-3.3		
C19	SERDES_RX6+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 G
C20	SERDES_RX6-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 G
C21	GND	PWR	PWR		
C22	SERDES_RX7+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 H
C23	SERDES_RX7-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 H
C24.. C26	N/C		N/C		
C27	LAD0	Local Bus	I/O-3.3		multiplexed CPU address/data signal

Table 6: Connector J2 Row C Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
C28	LAD1	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C29	N/C		N/C		
C30	N/C		N/C		
C31	GND	PWR	PWR		
C32	LAD2	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C33	LAD3	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C34	LAD4	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C35	LAD5	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C36	LAD6	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C37	LAD7	Local Bus	I/O-3.3		multiplexed CPU address/data signal
C38	LAD8 / GPIO0	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C39	LAD9 / GPIO1	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C40	LAD10 / GPIO2	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C41	GND	PWR	PWR		
C42	LAD11 / GPIO3	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C43	LAD12 / GPIO4	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C44	LAD13 / GPIO5	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C45	LAD14 / GPIO6	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C46	LAD15 / GPIO7	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
C47	EMI1_MDC	ETH MGT	O-2.5		Ethernet Management Clock
C48	EMI1_MDIO	ETH MGT	I/O-2.5	PU 3k3 2.5V	Ethernet Management In/Out
C49	IRQ1#	IRQ	I-3.3	PU 4k7 3.3V	
C50	IRQ2#	IRQ	I-3.3	PU 4k7 3.3V	
C51	GND	PWR	PWR		
C52	SERDES_RX8+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 I
C53	SERDES_RX8-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 I
C54	TYPE0#	TYPE		PD 4k7	
C55	SERDES_RX9+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 J
C56	SERDES_RX9-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 1 J
C57	TYPE1#	TYPE			
C58	SERDES_RX10+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2A
C59	SERDES_RX10-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2A
C60	GND	PWR			
C61	SERDES_RX11+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2B
C62	SERDES_RX11-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2B
C63	LA25	Local Bus	O-3.3		
C64	LA24	Local Bus	O-3.3		
C65	SERDES_RX12+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2C
C66	SERDES_RX12-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2C

Table 6: Connector J2 Row C Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
C67	LA23	Local Bus	0-3.3		
C68	SERDES_RX13+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2D
C69	SERDES_RX13-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 2D
C70	GND	PWR	PWR		
C71	SERDES_RX14+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3A
C72	SERDES_RX14-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3A
C73	GND	PWR	PWR		
C74	SERDES_RX15+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3B
C75	SERDES_RX15-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3B
C76	GND	PWR	PWR		
C77	LA22	Local Bus	0-3.3		
C78	SERDES_RX16+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3C
C79	SERDES_RX16-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3C
C80	GND	PWR	PWR		
C81	SERDES_RX17+	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3D
C82	SERDES_RX17-	SERDES	DP-I	AC coupled on module (100n)	SerDes Bank 3D
C83	LA21	Local Bus	0-3.3		
C84	GND	PWR	PWR		
C85	N/C		N/C		
C86	N/C		N/C		
C87	GND	PWR	PWR		
C88	N/C		N/C		
C89	N/C		N/C		
C90	GND	PWR	PWR		
C91	N/C		N/C		
C92	N/C		N/C		
C93	GND	PWR	PWR		
C93	N/C		N/C		
C94	N/C		N/C		
C96	GND	PWR	PWR		
C97	LA20	Local Bus	0-3.3		
C98	N/C		N/C		
C99	N/C		N/C		
C100	GND	PWR	PWR		
C101	N/C		N/C		
C102	N/C		N/C		
C103	GND	PWR			
C104.. C109	VCC_12V	PWR	PWR		nominal 12V
C110	GND	PWR	PWR		

Table 7: Connector J2 Row D Pinout

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
D1	GND	PWR	PWR		
D2.. D10	N/C		N/C		
D11	GND	PWR	PWR		
D12.. D16	N/C		N/C		
D17	LCS0#	Local Bus	0-3.3		connected to CPU LCS1#
D18	LCS1#	Local Bus	0-3.3		connected to CPU LCS7#
D19	SERDES_TX6+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1G
D20	SERDES_TX6-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1G
D21	GND	PWR	PWR		
D22	SERDES_TX7+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1H
D23	SERDES_TX7-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1H
D24	LA31	Local Bus	0-3.3		
D25	LA30	Local Bus	0-3.3		
D26	N/C		N/C		
D27	N/C		N/C		
D28	GND	PWR			
D29	N/C		N/C		
D30	N/C		N/C		
D31	GND	PWR	PWR		
D32.. D34	N/C		N/C		
D35	LALE	Local Bus	STRAP/ 0-3.3	PU 4k7	Local Bus address latch enable
D36	N/C		N/C		
D37	N/C		N/C		
D38	GND	PWR			
D39	SER0_CTS#	UART	I-3.3	PU 10k	UART[1]_CTS on CPU when configured for 2UART-Mode / UART[3]_RX in 4 UART-Mode
D40	SER0_RTS#	UART	0-3.3		UART[1]_RTS on CPU when configured for 2UART-Mode / UART[3]_TX in 4 UART-Mode
D41	GND	PWR	PWR		
D42	SER1_CTS#	UART	I-3.3	PU 10k	UART[2]_CTS on CPU when configured for 2UART-Mode / UART[4]_RX in 4 UART-Mode
D43	SER1_RTS#	UART	0-3.3		UART[2]_RTS on CPU when configured for 2UART-Mode / UART[4]_TX in 4 UART-Mode
D44	LBCTL	Local Bus	0-3.3		Local Bus buffer control
D45	LGTA#	Local Bus	0-3.3		Local Bus external access termination signal
D46	IRQ3#	IRQ	I-3.3	PU 4k7	can be routed to CPU-IRQ[7..11]#
D47	IRQ4#	IRQ	I-3.3	PU 4k7	can be routed to CPU-IRQ[7..11]#
D48	LA29	Local Bus	0-3.3		
D49	LA28	Local Bus	0-3.3		

Table 7: Connector J2 Row D Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
D50	IRQ_OUT#	MISC	OD-3.3	PU 4k7	for use as general purpose output
D51	GND	PWR	PWR		
D52	SERDES_TX8+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1I
D53	SERDES_TX8-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1I
D54	Reserved				leave unconnected
D55	SERDES_TX9+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1J
D56	SERDES_TX9-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 1J
D57	TYPE2#	TYPE	PDS		open
D58	SERDES_TX10+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2A
D59	SERDES_TX10-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2A
D60	GND	PWR	PWR		
D61	SERDES_TX11+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2B
D62	SERDES_TX11-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2B
D63	LA27	Local Bus	O-3.3		
D64	LA26	Local Bus	O-3.3		
D65	SERDES_TX12+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2C
D66	SERDES_TX12-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2C
D67	GND	PWR	PWR		
D68	SERDES_TX13+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2D
D69	SERDES_TX13-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 2D
D70	GND	PWR	PWR		
D71	SERDES_TX14+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3A
D72	SERDES_TX14-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3A
D73	GND	PWR	PWR		
D74	SERDES_TX15+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3B
D75	SERDES_TX15-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3B
D76	GND	PWR	PWR		
D77	IRQ5#	IRQ	I-3.3	PU 4k7	can be routed to CPU-IRQ[7..11]#
D78	SERDES_TX16+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3C
D79	SERDES_TX16-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3C
D80	GND	PWR	PWR		
D81	SERDES_TX17+	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3D
D82	SERDES_TX17-	SERDES	DP-O	AC coupled on module (100n)	SerDes Bank 3D
D83	TYPE3#	TYPE		PD 4k7	
D84	GND	PWR	PWR		
D85	N/C		N/C		
D86	N/C		N/C		
D87	GND	PWR	PWR		
D88	N/C		N/C		
D89	N/C		N/C		
D90	GND	PWR	PWR		
D91	N/C		N/C		
D92	N/C		N/C		
D93	GND	PWR	PWR		
D94	N/C		N/C		
D95	N/C		N/C		

Table 7: Connector J2 Row D Pinout (cont'd)

PIN	SIGNAL	SIGNAL GROUP	TYPE	TERMINATION	COMMENT
D96	GND	PWR	PWR		
D97	LA19 / GPIO11	Local Bus / GPIO	I/O-3.3	weak PU when configured for GPIO	multiplexed CPU address/data signal or GPIO depending on SCONF setting
D98	N/C	SERDES			
D99	N/C	SERDES			
D100	GND	PWR	PWR		
D101	N/C		N/C		
D102	N/C		N/C		
D103	GND	PWR	PWR		
D104.. D109	VCC_12V	PWR	PWR		nominal 12V
D110	GND	PWR	PWR		

Table 8: General Signal Description

TYPE	DESCRIPTION
I/O-3.3	Bi-directional 3.3V IO signal
I-3.3	3.3V input
O-3.3	3.3V output
OD-3.3	Open-Drain output
I-2.5	2.5V input
O-2.5	2.5V output
I/O-1.2	Bi-directional 1.2V IO signal
O-1.2	1.2V output
DP-I/O	Differential pair input/output
DP-I	Differential pair input
DP-O	Differential pair output
PDS	PullDown Strap / COM Express® type coding
STRAP	Straping input during power-up (do not connect any external Pullup or Pulldown resistor)
PWR	Power connection
PWR 3V	RTC data retention 3.3V power
PWR 5V	Standby power
N/C	Not connected

2.5.2 Signal Descriptions COM Express® Connectors

2.5.2.1 Ethernet (Group GigE MDI/GBE0 CTREF)

The COMe-bP5020 module provides one Gigabit Ethernet interface whose signals are already at copper wire Ethernet transmission voltage levels (physical levels / MDI) in accordance to the COM Express® Base Specification. 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 (GBE0_ACT#), Ethernet link (GBE0_LINK#), Ethernet speed 100Mbit/s (GBE0_LINK100#) and Ethernet speed 1000Mbit/s (GBE0_LINK1000#).

A reference voltage for the carrier board Ethernet magnetics center tap (GBE0_CTREF) is not required.

2.5.2.2 Ethernet Management (ETH MGT)

The management communication between the Ethernet MACs and the external connected Ethernet PHYs is realized by using the signal group ETH MGT. The CPU provides here two Ethernet management interface types (EMI1 and EMI2) which are dedicated to the supported transfer speed.

EMI1 is the PHY management interface for 10/100/1000 Mbps transfer rates and is therefore dedicated to the dTSEC MACs of the CPU.

EMI2 is the PHY management interface for 10Gbps transfer rates and is therefore dedicated to the XAUI interface of the CPU. For a more detailed description of the Ethernet management interfaces refer to the CPU’s reference manual or the appropriate IEEE standards (IEEE802.3: Part3, Clause 22 and Clause 45).

2.5.2.3 IEEE 1588

The Freescale™ QorIQ™ CPUs provide support for the Ethernet Precision Time Protocol (PTP) defined in the IEEE 1588 specification. In order to utilize this functionality the CPUs provide additional IEEE 1588 time stamp signals. For a more detailed description of those signals please refer to the CPU’s reference manual.

2.5.2.4 Serial ATA

Two standard SATA interfaces are provided on the COM Express® connector. These signals are “ready-to-use” and can therefore be routed directly to the SATA connectors/devices on the carrier.

2.5.2.5 SerDes

The signal group SerDes reflects all of the high speed low voltage differential signals provided by the CPU. The SerDes signals are grouped into so called lanes and links.

A set of differential signal pairs, one pair for transmission and one pair for reception is called a lane. One or more lanes together form a link which can support various logical protocols such as: PCIe, sRIO, SGMI, XAUI, etc.

The P5020 Processor provides 18 lanes which are grouped into so called “banks” (Bank 1, Bank 2, Bank 3). Bank 1 consists of 10 lanes (Bank1 A-J), whereas Bank 2 and Bank 3 consist of 4 lanes each (Bank2 A-D and Bank3 A-D).

The logical protocols which run on the SerDes lanes are specified by the SRDS_PRTCL configuration value programmed into the CPU at system power-up. To obtain a complete overview about all theoretical protocol combinations please refer to the Freescale™ “P5020 QorIQ Integrated Multicore Communication Processor Family Reference Manual”, Chapter 3.5.11 SerDes Lane Assignments and Multiplexing.

To handle the SerDes configuration in a more comfortable way, Kontron provides the configuration tool “sconf”. “sconf” provides a very easy way to configure the functionality of the SerDes lanes. Refer to Chapter 6, U-Boot for further information.

The following table illustrates the SerDes protocol combinations which can be selected by using the “sconf” command:

Table 9: SerDes Protocol Mapping

	CONNECTOR SIGNALS / CPU BANKS				
BASE CONFIG.	SERDES_ TX/RX[0..3]+/-	SERDES_ TX/RX[4..7]+/-	SERDES_ TX/RX[8..9]+/-	SERDES_ TX/RX[10..13]+/-	SERDES_ TX/RX[14..17]+/-
	Bank1 A-D	Bank1 E-H	Bank1 I-J	Bank2 A-D	Bank3 A-D
1	PCIe1	PCIe2	Debug (Aurora)	4 x SGMII	--
2	PCIe1	PCIe2	Debug (Aurora)	4 x SGMII	XAUI
3	PCIe1	PCIe2	Debug (Aurora)	XAUI	--
4	SRIO2 (2.5 or 5 Gbps)	SRIO1 (2.5 or 5 Gbps)	Debug (Aurora)	4 x SGMII	--
5	SRIO2 (3.125 Gbps)	SRIO1 (3.125 Gbps)	Debug (Aurora)	4 x SGMII	--

2.5.2.6 DMA

In order to support external application circuits which demand Direct Memory Access (DMA) the QorIQ™ CPU family provides two external sets of control signals which are routed to the COM Express® connectors. For a more detailed description of the DMA control signals please refer to the CPU’s reference manual

2.5.2.7 Local Bus / GPIO

2.5.2.7.1 Local Bus

The COMe-bP5020 provides a Local Bus interface for connecting directly to memory mapped parallel bus devices (SRAM-style). The Local Bus implementation on the COMe-bP5020 supports 8-bit and 16-bit data signal paths depending on the Local Bus chip select configuration and an 8 Mbyte address range for each of the two Local Bus chip selects.

Per default, LCS0# (LCS1# on the CPU) is configured for 16-bit operation, whereas LCS1# (LCS7# on the CPU) is configured for 8-bit operation.

The Local Bus signals designated as LAD0..15 incorporate multiplexed address and data information, whereby the Local Bus signals LA16..31 are dedicated address lines. Please be aware that external address latches must be provided on the LAD8..15 lines if an address range greater than 64kB is to be addressed.

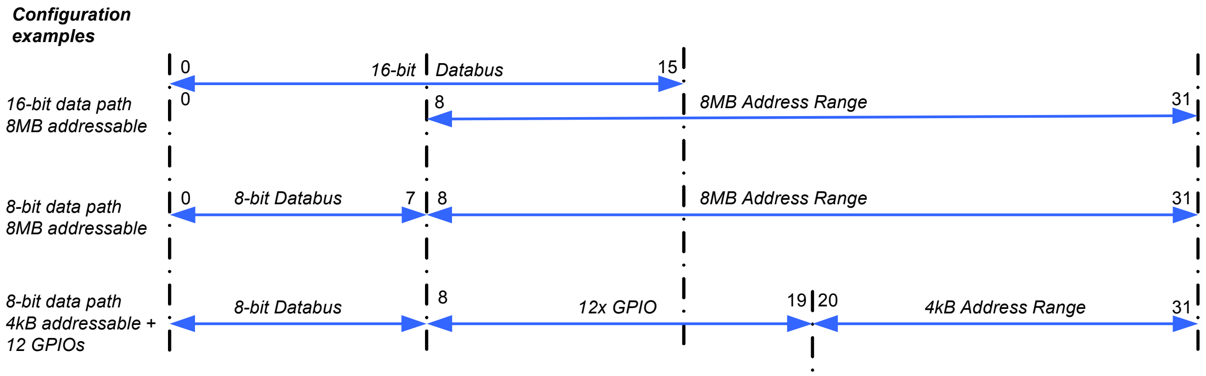
The numbering scheme for the Local Bus LA/LAD pins is noted in Power Architecture® style, meaning that LAD0 is the most significant bit and LA31 is the least significant bit.

For a better understanding of the QorIQ™ P5020 Local Bus functionality and the involved control signals please refer to the CPU’s reference manual.

2.5.2.7.2 GPIO

The COMe-bP5020 provides the possibility to convert some of the Local Bus signals to GPIO functionality. There are 12 signals on the COM Express® connector which can be multiplexed between Local Bus functionality and GPIO functionality in groups of 4 signals.

Figure 4: Examples of Local Bus and GPIO Configurations



The selection which function the dedicated group should have can be done with the “sconf” command in U-Boot depending on the required Local Bus data path width and Local Bus address range.

2.5.2.8 USB

There are five USB 2.0 high speed USB ports available on the COMe-bP5020.

The USB ports USB0..3 at the COM Express® connectors are provided using a 4-port USB hub with its Uplink-Port connected to the USB controller USB1 on the QorIQ™ P5020.

USB port USB4 at the COM Express® connector is connected directly to the USB2 controller of the QorIQ™ P5020. This port is capable of providing USB host or USB device functionality.

The configuration for host or device functionality is done using the U-Boot “sconf” command.

2.5.2.9 SDHC (SDIO)

The Freescale™ QorIQ™ CPUs incorporate an enhanced Secure Digital Host Controller (eSDHC) which provides support for MultiMediaCards (MMC) and Secure Digital (SD) Cards.

The interfacing signals of the CPU are multiplexed between the onboard SD card socket and the dedicated SDIO signals on the COM Express® connectors. The selection between the onboard socket and external interfacing is done via the DIP Switch SW1, switch 3.

2.5.2.10 SPI

The Serial Peripheral Interface Bus or SPI bus is a synchronous serial data link standard developed by Motorola that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (chip select) lines. Sometimes SPI is called a “four wire” serial bus, contrasting with three, two, and one wire serial buses.

For a detailed signal description please refer to the COM Express® base specification, chapter 4.3.12.

The COMe-bP5020 supports boot from an external SPI flash. Therefore it can be configured via pin B88 (BIOS_DIS1#) for the following configurations:

Table 10: SPI Signal Configurations

BIOS_DIS1#	FUNCTION	SIGNAL ROUTING
Open	Boot from on-module flashes	P5020 eSPI chip select SPI_CS2# is available on the carrier
Pulled to GND	Boot from external flash	P5020 eSPI chip select SPI_CS0# (boot chip select) is available on the carrier

The BIOS_DIS0# signal defined in the COM Express® Base specification is not used by the COMe-bP5020.

2.5.2.11 Serial Interface

Up to four UART interfaces are available on the COMe-bP5020. The following configurations are possible:

» 2x 4-wire UARTs (factory configuration)

» 4x 2-wire UARTs

The configuration of the UART modes can be done using the U-Boot “sconf” command.

2.5.2.12 SMB / I2C

The COMe-bP5020 provides two I2C controllers with speeds up to 400 kHz for application usage. The signals on the COM Express® connector labeled SMB_CK and SMB_DAT are connected to the I2C controller IIC2 of the P5020. The resources occupied by the devices are as follows:

Table 11: I2C Device Resources

DEVICE	I2C ADDRESS (binary)	I2C ADDRESS (hex)
User EEPROM	1010 110x	0xAC
RTC	1010 001x	0xA2
Thermal Sensor	1001 001x	0x92

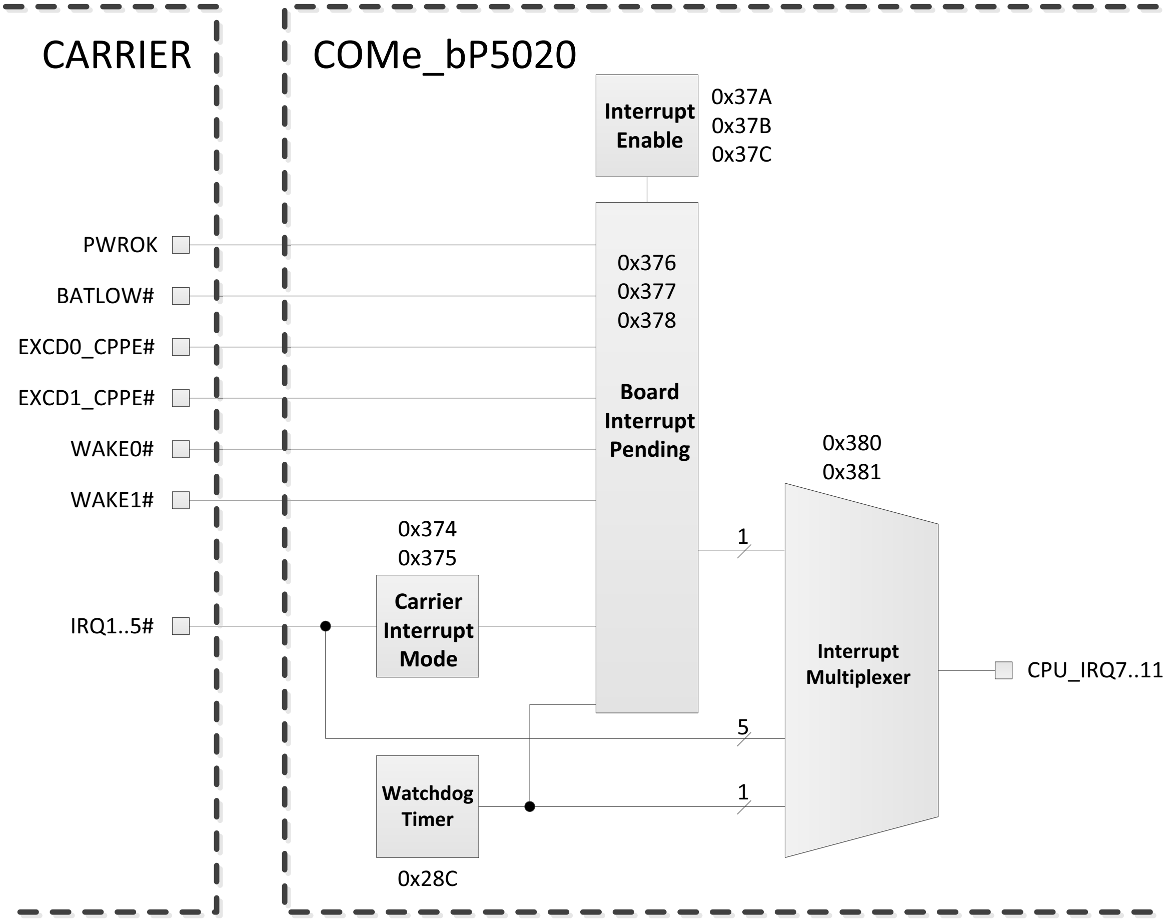
The signals on the COM Express® connector labeled I2C_CK and I2C_DAT are connected to the I2C controller IIC4 of the P5020. This controller is reserved for application use only on the COMe-bP5020.

2.5.2.13 IRQs

The COMe-bP5020 supports five IRQ inputs which can be configured for edge/level, high and low active usage. The operational mode of the IRQs is programmed via the Carrier Interrupt Mode1 and Carrier Interrupt Mode2 registers. Refer to Chapter 3 for further information.

The following figure demonstrates the IRQ routing of the COMe-bP5020.

Figure 5: IRQ Routing Scheme



2.5.2.14 Miscellaneous (MISC)

These signals are normally predefined for an X86 architecture board and have no defined functionality on Power Architecture® CPUs. On the COMe-bP5020 these signals may be used as general purpose output.

2.5.3 JTAG/Debug Interface

The COMe-bP5020 provides one JTAG/Debug connector, J3, to facilitate software debugging using an emulation probe. The connector type is: Hirose DF19G-20P-1H.

The following table provides pinout information for the debug connector J3.

Table 12: JTAG/Debug Connector J3 Pinout

PIN	SIGNAL	FUNCTION	I/O
1	3V3	Power supply	--
2	COP_TDO	JTAG data output	O
3	COP_TDI	JTAG data input	I
4	COP_TRST	JTAG test reset	I
5	NC	--	--
6	COP_TCK	JTAG test clock	I
7	COP_CKSTP_IN	BDM/COP checkstop input	I
8	COP_TMS	JTAG test mode select	I
9	COP_SRST	BDM/COP soft reset	I
10	COP_HRST	BDM/COP hard reset	I
11	COP_CKSTP_OUT	BDM/COP checkstop output	O
12	GND	Ground signal	--
13	NC	--	--
14	NC	--	--
15	NC	--	--
16	AURORA_HALT#	Aurora HALT	I
17	AURORA_EVTI#	Aurora Event in	I
18	AURORA_EVTO#	Aurora Event out	O
19	Reserved	--	--
20	Reserved	--	--

3 Configuration

3.1 DIP Switch Configuration

The COMe-bP5020 is equipped with one 4-bit DIP switch, SW1, for board configuration.

Table 13: DIP Switch SW1 Configuration

SWITCH	SETTING	DESCRIPTION
1	OFF	Uses the SerDes configuration which is defined via the U-Boot “sconf” command.
	ON	The COMe-bP5020 uses a fail-safe SerDes configuration. With this configuration the SerDes lanes are powered off.
2	OFF	Boot from the standard SPI boot flash
	ON	Boot from the recovery SPI boot flash
3	OFF	The SDHC interface is routed to the onboard MicroSD
	ON	The SDHC interface is routed to the COM Express connector
4	OFF	Reserved
	ON	

The default position for the above settings is: OFF

3.2 Board Memory Map

Table 14: COMe-bP5020 Virtual and Physical Memory Address Map

	VIRTUAL ADDRESS		PHYSICAL ADDRESS	
AREA NAME	START ADDR.	SIZE	START ADDR.	SIZE
PCIe3 IO	0xFFC2_0000	64 kB	0xE_FFC2_0000	64 kB
PCIe2 IO	0xFFC1_0000	64 kB	0xE_FFC1_0000	64 kB
PCIe1 IO	0xFFC0_0000	64 kB	0xE_FFC0_0000	64 kB
Onboard Logic	0xFF00_0000	4 kB	0xF_FF00_0000	4 kB
CCSR	0xFE00_0000	16 MB	0xF_FE00_0000	16 MB
NAND 4	0xF809_8000	512 kB	0xF_F809_8000	32 kB
NAND 3	0xF809_0000	512 kB	0xF_F809_0000	32 kB
NAND 2	0xF808_8000	512 kB	0xF_F808_8000	32 kB
NAND 1	0xF808_0000	512 kB	0xF_F808_0000	32 kB
MRAM	0xF800_0000	512 kB	0xF_F800_0000	512 MB
LocalBus (8-bit)	0xF580_0000	16 MB	0xF_F580_0000	16 MB
LocalBus (16-bit)	0xF500_0000	16 MB	0xF_F500_0000	16 MB
QMAN	0xF420_0000	2 MB	0xF_F420_0000	2 MB
BMAN	0xF400_0000	2 MB	0xF_F400_0000	2 MB
DCSR	0xF000_0000	4 MB	0xF_0000_0000	4 MB
PCIe3 Memory	0xE000_0000	256 MB	0xE_C000_0000	512 MB
SRIO2	0xD000_0000	256 MB	0xD_E000_0000	512 MB
SRIO1	0xC000_0000	256 MB	0xD_C000_0000	512 MB
PCIe2 Memory	0xA000_0000	512 MB	0xE_A000_0000	512 MB
PCIe1 Memory	0x8000_0000	512 MB	0xE_8000_0000	512 MB
DDR3 SDRAM	0x0000_0000	2 GB	0x0_0000_0000	8 GB

3.3 I/O Address Map

For the COMe-bP5020, the register address is composed of the base address of the Onboard Logic indicated in the virtual memory map (see Table 14) and the respective address offset indicated in the I/O address map (Table 15):

register address = 0xFF00_0000 base + address offset.

Table 15: I/O Address Map

ADDRESS OFFSET	DEVICE	ACRONYM
0x003	GPIO Direction Register 0	GPDIR0
0x004	GPIO Direction Register 1	GPDIR1
0x005	GPIO Data Register 0	GPDAT0
0x006	GPIO Data Register 1	GPDAT1
0x280	Status Register 0	STAT0
0x284	Device Protection Register	DPROT
0x285	Reset Status Register	RSTAT
0x288	Board ID High Byte Register	BIDH
0x289	Board and PLD Revision Register	BREV
0x28C	Watchdog Timer Control Register	WTIM
0x28D	Board ID Low Byte Register	BIDL
0x374	Carrier Interrupt Mode 1	CIM1
0x375	Carrier Interrupt Mode 2	CIM2
0x376	Board Interrupt Pending Register 1	BIPR1
0x377	Board Interrupt Pending Register 2	BIPR2
0x378	Board Interrupt Pending Register 3	BIPR3
0x37A	Board Interrupt Enable 1	BIE1
0x37B	Board Interrupt Enable 2	BIE2
0x380	Interrupt Multiplexer 1 Register	IMUX1
0x381	Interrupt Multiplexer 2 Register	IMUX2
0x390	Carrier Control Register	CCR

3.4 Board Control and Status Registers

The following registers are special registers which the COMe-bP5020 uses to monitor and control the onboard hardware special features.

NOTICE

Take care when modifying the contents of these registers as the system may be relying on the state of the bits under its control.

Table 16: Address: 0x003 - GPIO Direction Register 0 (GPDIR0)

BIT	7	6	5	4	3	2	1	0
NAME	DIR7	DIR6	DIR5	DIR4	DIR3	DIR3	DIR1	DIR0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

BITFIELD		DESCRIPTION
[7..0]	DIR[7..0]	GPIO[7..0] Direction 0 = GPIO[7..0] is configured for Input 1 = GPIO[7..0] is configured for Output

Table 17: Address: 0x004 - GPIO Direction Register 1 (GPDIR1)

BIT	7	6	5	4	3	2	1	0
NAME	reserved				DIR11	DIR10	DIR9	DIR8
ACCESS	R				R/W	R/W	R/W	R/W
RESET	0000				0	0	0	0

BITFIELD		DESCRIPTION
[3..0]	DIR[11..8]	GPIO[11..8] Direction 0 = GPIO[11..8] is configured for Input 1 = GPIO[11..8] is configured for Output

Table 18: Address: 0x005 - GPIO Data Register 0 (GPDAT0)

BIT	7	6	5	4	3	2	1	0
NAME	GPDAT7	GPDAT6	GPDAT5	GPDAT4	GPDAT3	GPDAT2	GPDAT1	GPDAT0
ACCESS	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

BITFIELD		DESCRIPTION
[7..0]	GPDAT[7..0]	GPDAT[7..0] Data If GPIO[7..0] is configured for INPUT: 0 = GPDAT[7..0] indicates that the signal GPIO[7..0] is at a 'low' level 1 = GPDAT[7..0] indicates that the signal GPIO[7..0] is at a 'high' level If GPIO[7..0] is configured for OUTPUT: 0 = GPDAT[7..0] indicates that the signal GPIO[7..0] is being driven 'low' 1 = GPDAT[7..0] indicates that the signal GPIO[7..0] is being driven 'high'

Table 19: Address: 0x006 - GPIO Data Register 1 (GPDAT1)

BIT	7	6	5	4	3	2	1	0
NAME	reserved				GPDAT11	GPDAT10	GPDAT9	GPDAT8
ACCESS	R				R/W	R/W	R/W	R/W
RESET	0000				0	0	0	0

BITFIELD		DESCRIPTION
[3..0]	GPDAT[11..8]	GPDAT[11..8] Data If GPIO[11..8] is configured for INPUT: 0 = GPDAT[11..8] indicates that the signal GPIO[11..8] is at a 'low' level 1 = GPDAT[11..8] indicates that the signal GPIO[11..8] is at a 'high' level If GPIO[7..0] is configured for OUTPUT: 0 = GPDAT[11..8] indicates that the signal GPIO[11..8] is being driven 'low' 1 = GPDAT[11..8] indicates that the signal GPIO[11..8] is being driven 'high'

Table 20: Address: 0x280 - Status Register 0 (STAT0)

BIT	7	6	5	4	3	2	1	0
NAME	reserved		BFSS		DIP4	DIP3	DIP2	DIP1
ACCESS	R		R		R	R	R	R
RESET	00		N/A		N/A	N/A	N/A	N/A

BITFIELD		DESCRIPTION
[5:4]	BFSS	SPI boot flash selection status: 00 = Standard SPI boot flash active 01 = Recovery SPI boot flash active 10 = external SPI boot flash active 11 = reserved
[3..0]	DIP[4..1]	DIP switch SW1, switches[4..1] (provides current status/position information of the SW1 switches): 0 = on 1 = off

Table 21: Address: 0x284 - Device Protection Register (DPROT)

BIT	7	6	5	4	3	2	1	0
NAME	SWP	reserved	SDWP	NFWP	SEWP	SFWP	EEWP	BFWP
ACCESS	R	R	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	0	1	0
BITFIELD		DESCRIPTION						
[7]	SWP	System write protection: 0 = onboard non-volatile memory devices not write protected 1 = onboard non-volatile memory devices write protected This bit reflects the state of the system hardware write protection signal (SYS_WP#).						
[5]	SDWP	microSD/microSDHC module write protection: 0 = module not write protected 1 = module write protected						
[4]	NFWP	SPI flash for OS write protection: 0 = flash not write protected 1 = flash write protected						
[3]	SEWP	SPD (Serial Presence Detect) EEPROM write protection: 0 = SPD EEPROM not write protected 1 = SPD EEPROM write protected						
[2]	SFWP	SATA Flash module write protection: 0 = SATA Flash module not write protected 1 = SATA Flash module write protected						
[1]	EEWP	System EEPROM write protection: 0 = System EEPROM not write protected 1 = System EEPROM write protected						
[0]	BFWP	Boot Flash write protection 0 = Boot flash not write protected 1 = Boot flash write protected						

Table 22: Address: 0x285 - Reset Status Register (RSTAT)

BIT	7	6	5	4	3	2	1	0
NAME	PORS	reserved						WTRS
ACCESS	R/W	R						R/W
RESET	N/A	00 0000						0
BITFIELD		DESCRIPTION						
[7]	PORS	Power-on reset status: 0 = system reset generated by software (warm reset) 1 = system reset generated by power-on (cold reset) Writing a ‘1’ to this bit clears the bit.						
[0]	WTRS	Watchdog timer reset status: 0 = system reset not generated by Watchdog timer 1 = system reset generated by Watchdog timer Writing a ‘1’ to this bit clears the bit.						

Table 23: Address: 0x288 - Board ID High Byte Register (BIDH)

BIT	7	6	5	4	3	2	1	0
NAME	BIDH							
ACCESS	R							
RESET	0xD0							
BITFIELD		DESCRIPTION						
[7:0]	BIDH	Board identification: 0xD0C8 0xD0 = COMe-bP5020 (high byte)						

Table 24: Address: 0x289 - Board and PLD Revision Register (BREV)

BIT	7	6	5	4	3	2	1	0
NAME	BREV				PREV			
ACCESS	R				R			
RESET	N/A				N/A			
BITFIELD		DESCRIPTION						
[7:4]	BREV	Board revision						
[3:0]	PREV	PLD revision						

Table 25: Address: 0x28C - Watchdog Timer Control Register (WTIM)

BIT	7	6	5	4	3	2	1	0
NAME	WTE	WMD		WEN/WTR	WTM			
ACCESS	R/W	R/W		R/W	R/W			
RESET	0	00		0	0000			
BITFIELD		DESCRIPTION						
[7]	WTE	Watchdog timer expired status bit 0 = Watchdog timer has not expired 1 = Watchdog timer has expired. Writing a ‘1’ to this bit resets it to 0.						
[6:5]	WMD	Watchdog mode 00 = Timer Only mode 01 = Reset mode 10 = Interrupt mode 11 = Cascaded mode (dual-stage mode)						
[4]	WEN/WTR	Watchdog enable / Watchdog trigger control bit: 0 = Watchdog timer not enabled Prior to the Watchdog being enabled, this bit is known as WEN. After the Watchdog is enabled, it is known as WTR. Once the Watchdog timer has been enabled, this bit cannot be reset to 0. As long as the Watchdog timer is enabled, it will indicate a ‘1’. 1 = Watchdog timer enabled Writing a ‘1’ to this bit causes the Watchdog to be retriggered to the timer value indicated by bits WTM[3:0].						
[3:0]	WTM	Watchdog timeout settings: 0000 = 0.125 s 0001 = 0.25 s 0010 = 0.5 s 0011 = 1 s 0100 = 2 s 0101 = 4 s 0110 = 8 s 0111 = 16 s 1000 = 32 s 1001 = 64 s 1010 = 128 s 1011 = 256 s 1100 = 512 s 1101 = 1024 s 1110 = 2048 s 1111 = 4096 s						

Table 26: Address: 0x28D - Board ID Low Byte Register (BIDL)

BIT	7	6	5	4	3	2	1	0
NAME	BIDL							
ACCESS	R							
RESET	0xC8							

BITFIELD		DESCRIPTION
[7:0]	BIDL	Board identification: 0xD0C8 0xC8 = COMe-bP5020 (low byte)

Table 27: Address: 0x374 - Carrier Interrupt Mode 1 Register (CIM1)

BIT	7	6	5	4	3	2	1	0
NAME	IRQ4_MODE		IRQ3_MODE		IRQ2_MODE		IRQ1_MODE	
ACCESS	R/W		R/W		R/W		R/W	
RESET	00		00		00		00	

BITFIELD		DESCRIPTION
[7:6.. 1:0]	IRQ[4..1]_MODE	Interrupt mode definition of the COM Express IRQ[4..1]# lines: 00 = edge triggered high-to-low (falling edge) 01 = edge triggered low-to-high (rising edge) 10 = level triggered low-active 11 = level triggered high-active

Table 28: Address: 0x375 - Carrier Interrupt Mode 2 Register (CIM2)

BIT	7	6	5	4	3	2	1	0
NAME	reserved						IRQ5_MODE	
ACCESS	R						R/W	
RESET	00 0000						00	

BITFIELD		DESCRIPTION
[1:0]	IRQ5_MODE	Interrupt mode definition of the COM Express IRQ5# line: 00 = edge triggered high-to-low (falling edge) 01 = edge triggered low-to-high (rising edge) 10 = level triggered low-active 11 = level triggered high-active

Table 29: Address: 0x376 - Board Interrupt Pending Register 1 (BIPR1)

BIT	7	6	5	4	3	2	1	0
NAME	reserved			CE_IRQ5	CE_IRQ4	CE_IRQ3	CE_IRQ2	CE_IRQ1
ACCESS	R			R/W	R/W	R/W	R/W	R/W
RESET	000			0	0	0	0	0

BITFIELD		DESCRIPTION
[4..0]	CE_IRQ[5..1]	COM Express IRQ[5..1] request: 0 = no interrupt requested 1 = interrupt requested (is masked until enabled) Writing a ‘1’ to this bit resets it to 0.

Table 30: Address: 0x377 - Board Interrupt Pending Register 2 (BIPR2)

BIT	7	6	5	4	3	2	1	0
NAME	WAKE1_RQ	WAKE0_RQ	reserved	BATLOW_RQ	PWROK_RQ	reserved		
ACCESS	R/W	R/W	R	R/W	R/W	R		
RESET	0	0	0	0	0	000		
BITFIELD		DESCRIPTION						
[7..6]	WAKE[1..0]_RQ	WAKE[1..0] request: 0 = no interrupt requested 1 = interrupt requested (is masked until enabled) Writing a ‘1’ to this bit resets it to 0.						
[4]	BATLOW_RQ	BATLOW IRQ request (battery supply is not OK) 0 = no interrupt requested 1 = interrupt requested (is masked until enabled) Writing a ‘1’ to this bit resets it to 0.						
[3]	PWROK_RQ	PWROK IRQ request: (used to indicate a power anomaly) 0 = no interrupt requested 1 = interrupt requested (is masked until enabled) Writing a ‘1’ to this bit resets it to 0.						

Table 31: Address: 0x378 - Board Interrupt Pending Register 3 (BIPR3)

BIT	7	6	5	4	3	2	1	0
NAME	reserved					EXCD1	EXCD0	WDT_IRQ
ACCESS	R					R	R	R
RESET	0 0000					0	0	0
BITFIELD		DESCRIPTION						
[2..1]	EXCD[1..0]	ExpressCard Detect IRQ[1..0] request 0 = no interrupt requested 1 = interrupt requested						
[0]	WDT_IRQ	Watchdog timer IRQ request 0 = no interrupt requested 1 = interrupt requested Writing a ‘1’ to bit WTE in the Watchdog Timer Control Register (WTIM) resets this bit to 0.						

Table 32: Address: 0x37A - Board Interrupt Enable Register 1 (BIE1)

BIT	7	6	5	4	3	2	1	0
NAME	reserved			IRQ5_EN	IRQ4_EN	IRQ3_EN	IRQ2_EN	IRQ1_EN
ACCESS	R			R/W	R/W	R/W	R/W	R/W
RESET	000			0	0	0	0	0
BITFIELD		DESCRIPTION						
[4..0]	IRQ[5..1]_EN	Enable bit for COM Express IRQ[5..1]# signals 0 = interrupt not enabled 1 = interrupt enabled						

Table 33: Address: 0x37B - Board Interrupt Enable Register 2 (BIE2)

BIT	7	6	5	4	3	2	1	0
NAME	WAKE1_EN	WAKE0_EN	reserved	BATLOW_EN	PWROK_EN	reserved		
ACCESS	R/W	R/W	R	R/W	R/W	R		
RESET	0	0	0	0	0	000		

BITFIELD		DESCRIPTION
[7..6]	WAKE[1..0]_EN	WAKE[1..0] IRQ enable: 0 = interrupt not enabled 1 = interrupt enabled (unmasked)
[4]	BATLOW_EN	BATLOW IRQ enable: 0 = no interrupt 1 = interrupt enabled (unmasked)
[3]	PWROK_EN	PWROK IRQ enable: 0 = no interrupt 1 = interrupt enabled (unmasked)

Table 34: Address: 0x380 - Interrupt Multiplexer 1 Register (IMUX1)

BIT	7	6	5	4	3	2	1	0
NAME	CIRQ10		CIRQ9		CIRQ8		CIRQ7	
ACCESS	R/W		R/W		R/W		R/W	
RESET	00		11		00		00	

BITFIELD		DESCRIPTION
[7:6.. 1:0]	CIRQ[10..7]	Interrupt/GPIO Multiplexer for CPU IRQ[10..7]# lines: 00 = line is not used 01 = COM Express IRQ[4..1] signal direct 10 = IRQ pending registers (collective) 11 = Watchdog timer IRQ

Table 35: Address: 0x381 - Interrupt Multiplexer 2 Register (IMUX2)

BIT	7	6	5	4	3	2	1	0
NAME	reserved						CIRQ11	
ACCESS	R						R/W	
RESET	00 0000						00	

BITFIELD		DESCRIPTION
[1:0]	CIRQ11	Interrupt/GPIO Multiplexer for CPU IRQ11# lines: 00 = line is not used 01 = COM Express IRQ5 signal direct 10 = IRQ pending registers (collective) 11 = Watchdog timer IRQ

Table 36: Address: 0x390 - Carrier Control Register (CCR)

BIT	7	6	5	4	3	2	1	0
NAME	CNTRL7	reserved			CNTRL3	reserved	CNTRL1	CNTRL0
ACCESS	R/W	R			R/W	R	R/W	R/W
RESET	0	0			0	0	0	0

BITFIELD		DESCRIPTION
[7]	CNTRL7	Control bit for COM Express IRQ_OUT# line: 0 = line is deasserted (logic level is HIGH-Z) 1 = line is asserted (logic level is LOW)
[3]	CNTRL3	Control bit for COM Express SUS_S3# line: 0 = line is deasserted (logic level is HIGH) 1 = line is asserted (logic level is LOW)
[1]	CNTRL1	Control bit for COM Express THRMTRIP# line: 0 = line is deasserted (logic level is HIGH) 1 = line is asserted (logic level is LOW)
[0]	CNTRL0	Control bit for COM Express CB_RESET# line: 0 = line is deasserted (logic level is HIGH) 1 = line is asserted (logic level is LOW)

4 Power Considerations

4.1 Supply Voltage

Following supply voltage is specified at the COM Express® connector.

Table 37: Supply Voltages

TYPE	RANGE
VCC	8.5V - 18V (12V nominal)
STANDBY	5V DC +/- 5%
RTC	2.5V - 3.3V

The 5V Standby and the RTC voltage are not mandatory for operation.

4.2 Power Supply Rise Time

The input voltages shall rise from ≤ 10% of nominal to within the regulation ranges within 0.1ms to 20ms. There must be a smooth and continuous ramp of each DC input voltage from 10% to 90% of its final set-point as specified in the ATX specification.

4.3 Supply Voltage Ripple

The supply voltage ripple must not be greater than 100 mV peak to peak 0 – 20 MHz.

4.4 Power Consumption

The maximum power consumption of the COMe-bP5020 is a function of clock frequencies, workload/utilization, temperature and component variations/tolerances.

The following tables indicate the typical power consumption of the COMe-bP5020 with 2GHz core clock and 8GB DDR3 memory under various conditions.

Table 38: Workload Dependency

APPLICATIONS	POWER CONSUMPTION
U-Boot (idle)	21.1 W
Linux (idle)	20.3 W
Linux (Memtest)	23.5 W
Linux (Drystone)	23.3 W
U-Boot (idle) + GigE	22.1 W
Linux (idle) + GigE	21.3 W

Table 39: Power Consumption vs. Ambient Temperature (U-Boot in Idle)

AMBIENT AIR TEMPERATURE	12V RAIL LOAD	POWER CONSUMPTION
23° C	1.6 A	21.1 W
40° C	1.94 A	23.3 W
50° C	2.08 A	25.0 W
60° C	2.25 A	27.07 W
65° C	2.36 A	28.3 W
70° C	2.47 A	29.6 W

5 Thermal

5.1 Heatspreader

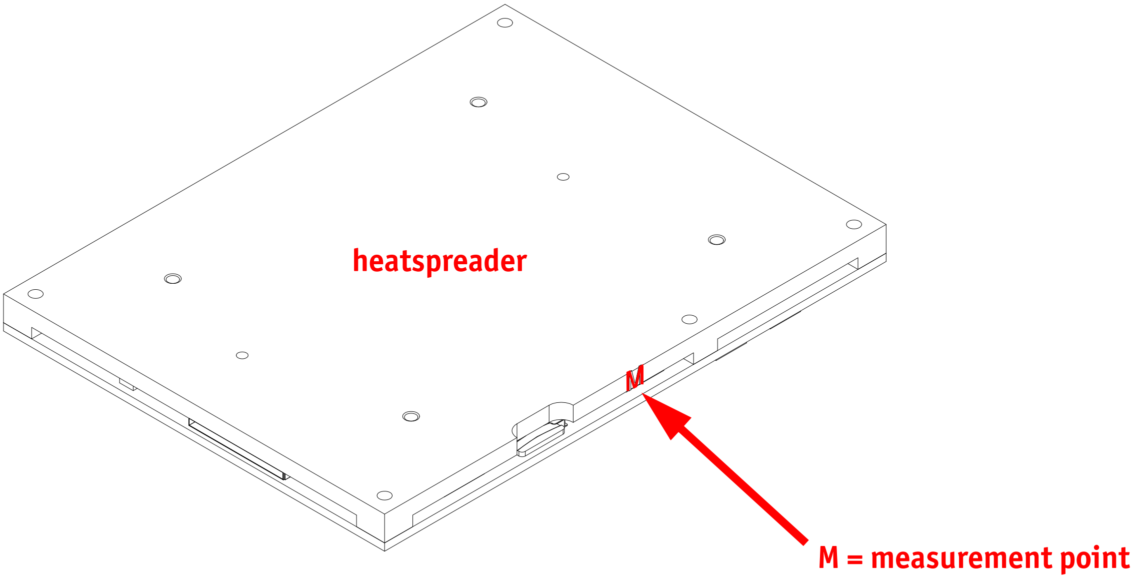
The thermal concept of the COMe-bP5020 is based on a specially designed full-board heatspreader which contacts the main hot spots of the board and therefore provides optimal heat transfer from the board's top surface.

The heatspreader plate is NOT a heat sink. It serves as a COM Express® standard thermal interface for use with a heat sink or other cooling solution. Various thermal management solutions can be used with the heatspreader plate, including active and passive approaches. The optimal cooling solution will vary, depending on the COM Express® application and environmental conditions.

Cooling of the COMe-bP5020 is a function of the attached heatsink or other heat transfer mechanism which must provide adequate cooling capability. To determine cooling performance, the module temperature can be measured at the temperature measurement point "M" indicated in the figure below.

The cooling solution, be it active or passive, must in any event maintain a heatspreader plate temperature of 80°C or less.

Figure 6: COMe-bP5020 Heatspreader



5.2 Cooling Considerations

The COMe-bP5020 is delivered either with a heatspreader plate or a Kontron off-the-shelf cooling solution preassembled.

Kontron provides two off-the-shelf cooling solutions. One for passive cooling and the other for active cooling. The usage of one or the other of these two cooling solutions will result in the following maximum possible ambient air operating temperatures as indicated in the table below. Please contact Kontron Support for more information about available off-the-shelf solutions.

Table 40: Maximum Ambient Air Temperature for Two Kontron Cooling Solutions

COOLING SOLUTION	MAXIMUM AMBIENT AIR TEMPERATURE	ORDER NUMBER
PASSIVE (specially designed heat sink without heatspreader)	55° C	tbs
ACTIVE (with heatspreader and fan)	75° C	36099-0000-99-0-R2.0

The following figures demonstrate the off-the-shelf passive and active cooling solutions available from Kontron.

Figure 7: Passive Cooling Solution

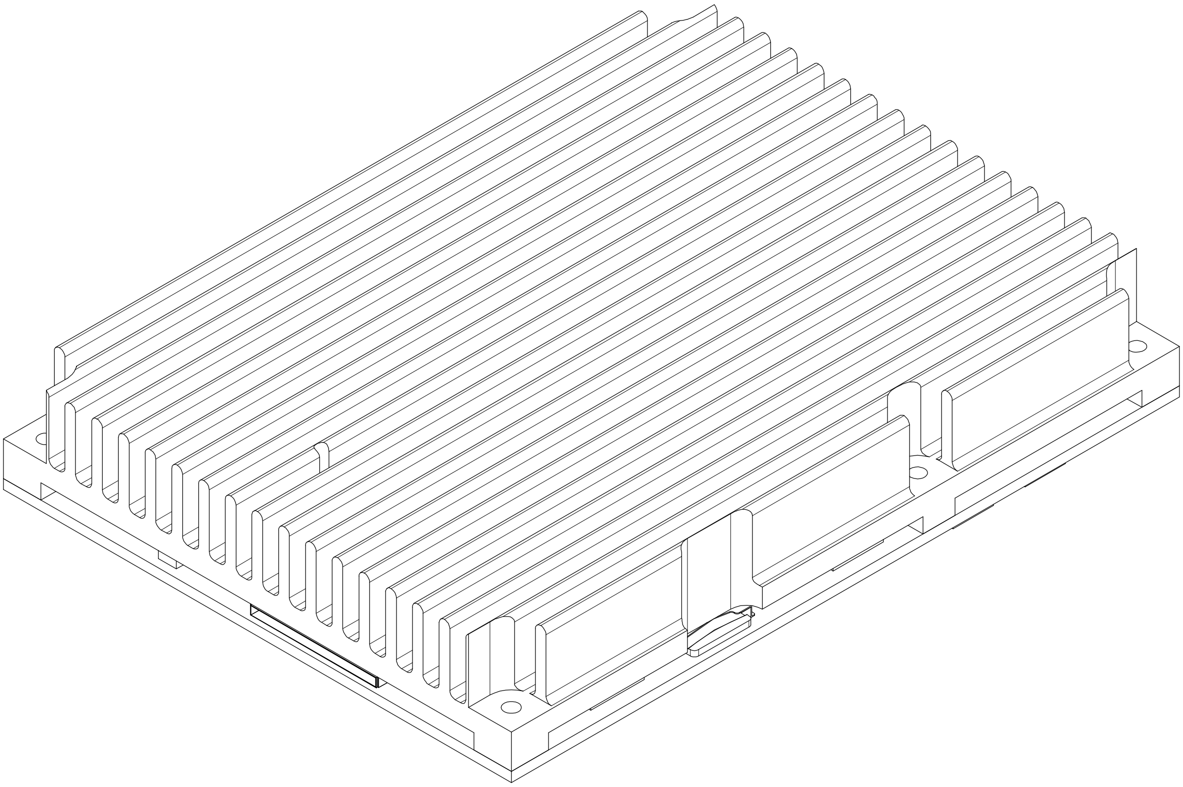
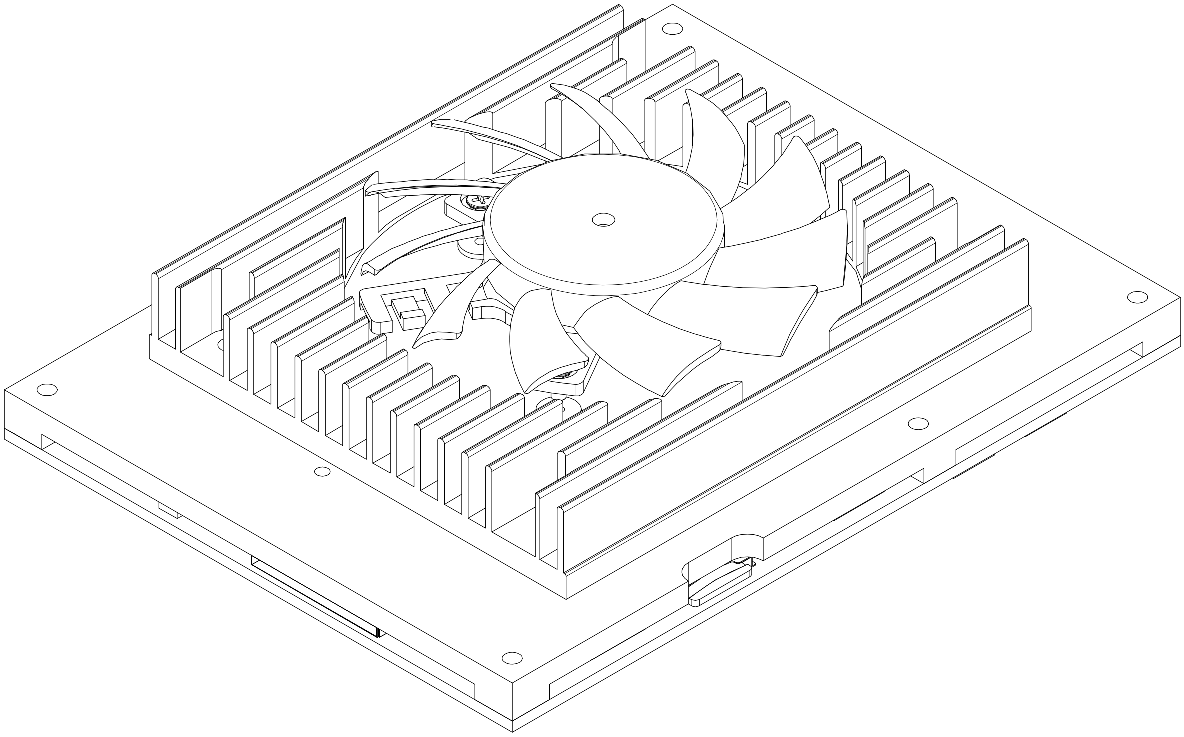


Figure 8: Active Cooling Solution



6 U-Boot

6.1 Introduction to U-Boot

U-Boot is an open source bootloader software developed and maintained by DENX Software Engineering GmbH (http://www.denx.de). Kontron provides U-Boot with all its standard features as well as Kontron-specific features for usage with Kontron’s COMe-bP5020 module. This software is pre-installed at the factory and is ready for use on power-up.

This chapter provides specific information on Kontron’s implementation of U-Boot and its usage. Please refer to the DENX web site for up-to-date on-line documentation of all of U-Boot’s standard features.

6.2 Standard U-Boot Commands

U-Boot is provided with a set of standard commands for which documentation is available on the DENX web site. Some of the standard commands have sub-groups which can be displayed when help for the main group command is requested. Where relevant, further information concerning the usage of standard commands is provided in this guide to assist users in performing specific functions.

The following table indicates the standard U-boot commands configured for the COMe-bP5020. The blue-shaded table cells indicate standard U-Boot commands tested by Kontron. Only the standard U-Boot commands relevant for the normal operation of the COMe-bP5020 U-Boot bootloader have been tested by Kontron.

Table 41: Standard U- Boot Commands Configured for the COMe-bP5020

COMMAND	DESCRIPTION
?	Alias for 'help'
base	Print or set address offset
bdinfo	Print Board Info structure
boot	Boot default, i.e., run 'bootcmd'
bootd	Boot default, i.e., run 'bootcmd'
bootelf	Boot from an ELF image in memory
bootm	Boot application image from memory
bootp	Boot image via network using BOOTP/TFTP protocol
bootvx	Boot vxWorks from an ELF image
chpart	Change active partition
cmp	Memory compare
coninfo	Print console devices and information
cp	Memory copy
cpu	Multiprocessor CPU boot manipulation and release
crc32	Checksum calculation
dhcp	Boot image via network using DHCP/TFTP protocol
echo	Echo args to console
editenv	Edit environment variable
env	Environment handling commands
errata	Report errata workarounds
exit	Exit script
ext2load	Load binary file from an Ext2 filesystem
ext2ls	List files in a directory (default /)
false	Do nothing, unsuccessfully
fatinfo	Print information about filesystem
fatload	Load binary file from a dos filesystem
fatls	List files in a directory (default /)
fdt	Flattened device tree utility commands
fsinfo	Print information about filesystems
fsload	Load binary file from a filesystem image
go	Start application at address 'addr'
grepenv	Search environment variables

Table 41: Standard U- Boot Commands Configured for the COMe-bP5020 (cont'd)

COMMAND	DESCRIPTION
help	Print command description/usage
i2c	I2C subsystem
iminfo	Print header information for application image
imxtract	Extract a part of a multi-image
interrupts	Enable or disable interrupts
irqinfo	Print information about IRQs
itest	Return true/false on integer compare
loadb	Load binary file over serial line (kermit mode)
loads	Load S-Record file over serial line
loady	Load binary file over serial line (ymodem mode)
loop	Infinite loop on address range
ls	List files in a directory (default /)
md	Memory display
mdio	MDIO utility commands
mii	MII utility commands
mm	Memory modify (auto-incrementing address)
mmc	MMC sub system
mmcinfo	Display MMC info
mtdparts	Define flash/nand partitions
mtest	Simple RAM read/write test
mw	Memory write (fill)
nand	NAND subsystem
nboot	Boot from NAND device
nfs	Boot image via network using NFS protocol
nm	Memory modify (constant address)
pci	List and access PCI Configuration Space
ping	Send ICMP ECHO_REQUEST to network host
printenv	Print environment variables
reginfo	Print register information
reset	Perform RESET of the CPU
run	Run commands in an environment variable
sata	SATA sub-system
saveenv	Save environment variables to persistent storage
saves	Save S-Record file over serial line
setenv	Set environment variables
setexpr	Set environment variable as the result of eval expression
sf	SPI flash subsystem
showvar	Print local hushshell variables
sleep	Delay execution for some time
source	Run script from memory
test	Minimal test like /bin/sh
tftpboot	Boot image via network using TFTP protocol
true	Do nothing, successfully
ubi	ubi commands
ubifsload	Load file from an UBIFS filesystem
ubifsls	List files in a directory
ubifsmount	Mount UBIFS volume
ubifsumount	Unmount UBIFS volume
usb	USB sub-system
usbboot	Boot from USB device
version	Print monitor, compiler and linker version

6.3 Kontron-Specific Commands

Kontron's implementation of U-Boot includes certain enhancements to provide specific functions not incorporated in the standard U-Boot. The following table provides a complete listing of all Kontron-specific U-Boot commands implemented on the COMe-bP5020.

Table 42: Kontron-Specific Commands

COMMAND	DESCRIPTION
flsw	FLash SWitch Indicates or selects the currently active SPI boot flash
kboardinfo	Kontron Board Information Displays a summary of board and configuration information
md5sum	Message digest 5 checksum Creates or checks the md5 message digest over a memory area
sconf	Kontron Board Configuration Provides functions for software-based configuration of external interfaces available on the COM Express® connectors
tlbdbg	Translation Look-aside Buffer DeBuG Displays current configuration of TLB0 and TLB1
vpd	Vital Product Data Provides display and importing functions for vital product data entities

The following tables provide command syntax reference information, a short description, and, in some cases, usage examples.

Table 43: flsw Command

flsw	Indicates the currently selected SPI boot flash or selects either the standard or recovery SPI boot flash for flash operations other than booting

SYNTAX: `flsw [s|r]`

flsw command

Issuing the command without arguments will indicate the currently selected SPI flash

Also returns “true” or “false” depending on the currently selected flash

s option:

Selects the standard SPI boot flash for flash operations

r option:

Selects the recovery SPI boot flash for flash operations

DESCRIPTION: This command is used to determine the currently selected SPI boot flash or to select either the standard SPI boot flash or the recovery SPI boot flash for flash operations other than booting

In addition, this command returns “true” if the standard SPI boot flash is selected or “false” if the recovery SPI boot flash is selected. This is used in the update scripts to prevent the recovery SPI boot flash from being updated.

To select the recovery SPI boot flash for booting use the DIP Switch SW1, switch 2. For further information, refer to Chapter 3.1, Table 13 and Chapter 6.8.

The output of this command always shows the current state.

USAGE: 1. Query flash status:

```
=> flsw
standard boot flash active
=>
```

2. Select the standard SPI boot flash for flash operations:

```
=> flsw s
=>
```

Table 44: kboardinfo Command

kboardinfo	Displays a summary of board and configuration information

SYNTAX: `kboardinfo`

kboardinfo command

DESCRIPTION: This command compiles information from various board sources and provides a summary listing of this information:

USAGE: 1. Display board information:

```
=> kboardinfo
Board id:              0xd0c8
Hardware rev.:         0xf
Logic rev.:            0x4
Boot flash:            Standard Flash
In system slot:        na
Geographic address:    na
Material number:       na
Serial number:         0400168722
U-Boot article name:   SK-FIRM-UBOOT-D0C01
U-Boot material num:   1052-6335
=>
```

Table 45: md5sum Command

md5 sum	Creates or checks the md5 message digest over a memory area

SYNTAX: `md5sum <data-address> <length> [<cksum-address>]`

md5sum	command
<data-address>	parameter: hexadecimal start address of memory area
<length>	parameter: hexadecimal length of memory area
<cksum-address>	parameter: hexadecimal If present: compares the calculated md5 message digest with the md5 message digest available at this address. If absent: calculates the md5 message digest over the specified memory range and prints it to the console.

DESCRIPTION: This command is used to create or check the md5 message digest over a memory area.

If the optional third parameter <checksum-address> is omitted, the md5 message digest is calculated over the specified memory range and printed to the console.

If the optional third parameter <cksum-address> is specified, the md5 message digest is calculated over the specified memory range and compared with the md5 message digest at <cksum-address>. If the digest is identical, the command returns 0. If the digests do not match, a value other than zero is returned. When a comparison is made, nothing is printed to the console since this usage of the command is intended for use within scripts.

The md5 message digest at <cksum-address> may be specified in ASCII or binary format.

USAGE: 1. Calculate an md5 message digest:

```
=> md5sum 100000 80000
8fe7006660a2df2265b7cd707eb98786
=>
```

2. Check the md5 message digest of a file previously loaded to 100000 with a size of 80000 and its md5 message digest loaded to 10000 in a script

```
=> setenv check_crc “if md5sum 100000 80000 10000; then echo ‘md5 message
digest OK’; else echo ‘md5 message digest BAD’; fi”
=>run check_crc
md5 message digest OK
=>
```

Table 46: sconf Command

sconf	Provides functions for configuration of external interfaces

SYNTAX:	`sconf info|select <num>|set [<par> <val>]|status|save|undo`
sconf	command
info	option: displays available configurations
select	option: selects base configuration <num>
<num>	parameter: text string (decimal) <0, 1, … n> number of base configuration
set	option: indicate or configure parameter for new base configuration
<par>	parameter: text string <[x …]x> parameter for new base configuration
<val>	parameter: text string <[x …]x> value assigned to <par>
status	option: displays information concerning the status of board configuration
save	option: saves the current settings
undo	option: ignore current settings and use saved settings again
DESCRIPTION:	This command is used to configure external interfaces available on the COMe-bP5020's connectors. The "sconf info" command shows the possible configurations as well as the currently selected configuration. The active configuration is indicated using the "sconf status" command. To configure external interfaces, select a base configuration via the "sconf select" command. Then, the parameters can be defined more exactly via the "sconf set" command. To apply the configuration, invoke the "sconf save" command then perform a module powercycle (a hardware reset is not sufficient to activate the new configuration).

Table 46: sconf Command (cont'd)

USAGE: 1. Display available configurations:

```
=> sconf info
List of available base configurations:
  *<  >* Stored base configuration
  -[  ]- New base configuration (work in progress)
============+======================================================================
 No   SRDS  |  Bank1         Bank1        Bank1     Bank2           Bank3
      PRTCL |   A-D          E-H          I-J       A-D             A-D
============+======================================================================
            |
*[ 1]* 0x11 | PCIe1         PCIe2         DBG       GbE[4x1]        SATA[2x1]
            |
 < 2>  0x13 | PCIe1         PCIe2         DBG       GbE[4x1]        XAUI
            |
 < 3>  0x15 | PCIe1         PCIe2         DBG       XAUI            SATA[2x1]
            |
 < 4>  0x17 | sRIO2 [@5.0]  sRIO1 [@5.0]  DBG       GbE[4x1]        SATA[2x1]
            |
 < 5>  0x17 | sRIO2 @3.125  sRIO1 @3.125  DBG       GbE[4x1]        SATA[2x1]
            |

Board's Port Configuration for base config 1 (serdes protocoll 0x11):

Option/Setting | Value  #  command   option  <and value>
---------------+--------+------------------------------------
PCIe1          : @5.0G  # sconf set  pcie1   <@5.0G | @2.5G | off>
PCIe2          : @5.0G  # sconf set  pcie2   <@5.0G | @2.5G | off>
Aurora         : off    # sconf set  aurora  <off | @5.0G | @2.5G>
dTSEC1         : on     # sconf set  dtsec1  <on | off>
dTSEC2         : on     # sconf set  dtsec2  <on | off>
dTSEC3         : on     # sconf set  dtsec3  <on | off>
dTSEC4         : on     # sconf set  dtsec4  <on | off>
SATA1          : on     # sconf set  sata1   <on | off>
SATA2          : on     # sconf set  sata2   <on | off>
USB            : host   # sconf set  usb     <host | dev>
UART           : 4wire  # sconf set  uart    <4wire | 2wire>
SDHC           : ext    # sconf set  sdhc    <onb | ext>
GPIOA          : gpio   # sconf set  gpioa   <gpio | lbus>
GPIOB          : gpio   # sconf set  gpiob   <gpio | lbus>
GPIOC          : gpio   # sconf set  gpioc   <gpio | lbus>
=>
```

2. Select new base configuration

```
=> sconf select 1
New base configuration 1
=>
```

Table 47: tlbdbg Command

tlbdbg	Displays current configuration of TLB0 and TLB1
SYNTAX:	`tlbdbg`
tlbdbg	command
DESCRIPTION:	This command provides information on the translation look-aside buffers TLB0 ad TLB1 for debugging purposes during U-Boot development or for debugging OS startup issues.
USAGE:	1. Display TLB0/TLB1 information

```
=> tlbdbg
TLBx Configuration Register : 04110200 401bc040


TLB0: [check 512 entries]
IDX  PID EPN       SIZE  V TS     RPN       U0-U3 WIMGE UUUSSS
-------------------------------------------------------------

TLB1: [check 64 entries]
IDX  PID EPN       SIZE  V TS     RPN       U0-U3 WIMGE UUUSSS
-------------------------------------------------------------
  1d: 00  fe000000  16MB  V 0d  -> f_fe000000 0000 -I-G- ---RWX
  2d: 00  00000000  1GB   V 0d  -> 0_00000000 0000 ----- ---RWX
  3d: 00  80000000  1GB   V 0d  -> e_80000000 0000 -I-G- ---RWX
  4d: 00  40000000  1GB   V 0d  -> 0_40000000 0000 ----- ---RWX
  5d: 00  ffc00000  64kB  V 0d  -> e_ffc00000 0000 -I-G- ---RWX
  6d: 00  ffc10000  64kB  V 0d  -> e_ffc10000 0000 -I-G- ---RWX
  7d: 00  ff000000  4kB   V 0d  -> f_ff000000 0000 -I-G- ---RWX
  9d: 00  f4000000  1MB   V 0d  -> f_f4000000 0000 ----- ---RWX
 10d: 00  f4100000  1MB   V 0d  -> f_f4100000 0000 -I-G- ---RWX
 11d: 00  f4200000  1MB   V 0d  -> f_f4200000 0000 ----- ---RWX
 12d: 00  f4300000  1MB   V 0d  -> f_f4300000 0000 -I-G- ---RWX
 13d: 00  f0000000  4MB   V 0d  -> f_00000000 0000 -I-G- ---RWX
 16d: 00  f8080000  64kB  V 0d  -> f_f8080000 0000 -I-G- ---RWX
 17d: 00  fffff000  4kB   V 0d  -> 0_7ffff000 0000 -I-G- ---RWX
 18d: 00  f8090000  64kB  V 0d  -> f_f8090000 0000 -I-G- ---RWX
 19d: 00  c0000000  256MB V 0d  -> d_c0000000 0000 -I-G- ---RWX
 20d: 00  d0000000  256MB V 0d  -> d_d0000000 0000 -I-G- ---RWX
=>
```

Table 48: vpd Command

vpd	Provides functions for configuration of external interfaces

SYNTAX: `vpd print [<name>]|import (<name>|all_params)`

vpd command

print option:
displays VPD information (source: System EEPROM)
(if <name> is not used, all VPD entities are displayed)

<name> parameter: text string
<[x ...]x>

name of VPD entity addressed by option

import option:
imports VPD information to the U-Boot environment
(source: System EEPROM; target: RAM)

all_params parameter: text constant
all_params

selects all VPD entities for importing to the U-Boot environment

DESCRIPTION: Vital Product Data are information stored in the System EEPROM which are required for proper operation of the board. With this command the VPD entities can be displayed or imported to the U-Boot environment in RAM.

Among the VPD entities are, for example, the board serial number and the board's Ethernet MAC addresses.

If the option "import" is invoked, existing VPD entities in the environment in RAM are overwritten. If a "saveenv" is then invoked, the previously stored values in the currently selected SPI boot flash environment area are overwritten.

USAGE: 1. Display all VPD entities:

```
=> vpd print
<response: displays all VPD entities>
=>
```

2. Display eth1addr entity

```
=> vpd print eth1addr
eth1addr=00:80:82:47:12:02
=>
```

3. Import eth1addr entity to environment

```
=> vpd import eth1addr
import eth1addr = 00:80:82:47:12:02 to … environment
=>
```

4. Import all VPD entities to environment

```
=> vpd import all_params

<response: displays all imported VPD entities; format for each imported VPD
entity as follows:>

import <name> = <value> to environment
.
.
.
import <name> = <value> to environment
=>
```

6.4 U-Boot Access and Startup

Communication with U-Boot is achieved via a serial console configured for 115200 baud, 8N1, no hardware handshake.

Initially, U-Boot executes the commands defined in the environment variable “preboot”. Then, if not otherwise interrupted, U-Boot pauses for the time defined in the environment variable “bootdelay” and then executes the statements stored in the environment variable “bootcmd”. To gain access to the U-Boot command prompt, type in any single character during the boot delay time.

If required, the boot delay function can be configured in such a way that even when the boot delay is set to “0” to have characters, which are sent over the serial interface prior to the boot wait time, be recognized to allow operator intervention in the boot process.

6.5 Working with U-Boot

6.5.1 General Operation

Most operations are carried out using the main memory as an intermediate step. It is not possible, for example, to boot a kernel image directly from a tftp server. Instead, the kernel image is first loaded to memory and then booted from there with another command.

The same is true when writing new contents to the SPI boot flashes.

This concept is very flexible since it separates the commands which handle the loading of data from the commands that carry out actions like booting.

6.5.2 Using the sconf Command

In previous board designs, DIP switches were used to configure the fabric interfaces. In response to evolving application requirements, the “sconf” command has been designed to provide increased configuration flexibility.

The COMe-bP5020 is delivered with a default configuration for the external interfaces routed to the COM Express® connectors. If required, these interfaces may be configured via the “sconf” command according to the application requirements.

The factory default configuration for the COMe-bP5020 is as follows:

- “sconf” base configuration: 0
- SRIO system size: small
- SRIO interface mode: agent
- GbE (DTSEC4): port 1

To obtain information about the currently active configuration, invoke the “sconf status” command.

6.5.3 Examples of sconf Command Usage

6.5.3.1 sconf select

To change the setting, invoke the “sconf select” command.

Example:

```
=> sconf select 2
New base configuration 2
=>
```

6.5.3.2 sconf set

The setting of the chosen base configuration can be changed via the “sconf set” command. If the “sconf set” command is invoked without parameters, all changeable options are shown. In the following example, first, all possible settings are shown, and then the SDHC routing is changed from onboard to external.

```
=> sconf set
Board's Port Configuration for base config 3 (serdes protocoll 0x15):

Option/Setting | Value  #  command   option  <and value>
---------------+--------+------------------------------------
PCIe1          : @5.0G  # sconf set  pcie1   <@5.0G | @2.5G | off>
PCIe2          : @5.0G  # sconf set  pcie2   <@5.0G | @2.5G | off>
Aurora         : off    # sconf set  aurora  <off | @5.0G | @2.5G>
XAUI           : on     # sconf set  xaui    <on | off>
SATA1          : on     # sconf set  sata1   <on | off>
SATA2          : on     # sconf set  sata2   <on | off>
USB            : host   # sconf set  usb     <host | dev>
UART           : 4wire  # sconf set  uart    <4wire | 2wire>
SDHC           : onb    # sconf set  sdhc    <onb | ext>
GPIOA          : gpio   # sconf set  gpioa   <gpio | lbus>
GPIOB          : gpio   # sconf set  gpiob   <gpio | lbus>
GPIOC          : gpio   # sconf set  gpioc   <gpio | lbus>
=> sconf set sdhc ext
```

After each “sconf select” or “sconf set” a system powercycle must be performed.

6.5.4 Using the Network

6.5.4.1 Interface Selection

U-Boot provides support for multiple Ethernet interfaces for transferring files from a file server. This is accomplished using the environment variables: “ethprime”, “ethact” and “ethrotate”.

The following table indicates the assignment of the Ethernet interfaces to the COM Express® SerDes signals depending on the active “sconf” base configuration.

Table 49: Ethernet Interface Assignment Depending on the Base Configuration

BASE CONFIGURATION	ETHERNET INTERFACE	COM EXPRESS® SIGNALS
1, 4, 5	FM1@DTSEC1	SERDES_TX/RX10+/-
	FM1@DTSEC2	SERDES_TX/RX11+/-
	FM1@DTSEC3	SERDES_TX/RX12+/-
	FM1@DTSEC4	SERDES_TX/RX13+/-
2	FM1@DTSEC1	SERDES_TX/RX10+/-
	FM1@DTSEC2	SERDES_TX/RX11+/-
	FM1@DTSEC3	SERDES_TX/RX12+/-
	FM1@DTSEC4	SERDES_TX/RX13+/-
	FM1@TGEC1	SERDES_TX/RX[14..17]+/-
3	FM1@TGEC1	SERDES_TX/RX[10..13]+/-

6.5.4.1.1 ethprime

“ethprime” is used to select the required interface after power-up or reset. During boot-up, the U-Boot checks if “ethprime” is set. If set, “ethprime” is used as the first active Ethernet interface (“ethact”). Please note that the setting of the “ethprime” is lost after a reset. To retain the environment permanently, use the command “saveenv”, which saves the complete environment to flash.

Example:

```
=> setenv ethprime FM1@DTSEC3
=> saveenv
Saving environment to SPI Flash...
2 MiB
SF: Detected AT25DF161 with page size 256 Bytes, total 2 MiB
Erasing SPI flash...Writing to SPI flash...done
=> reset
...
=> printenv ethact
ethact=FM1@DTSEC3
=>
```

6.5.4.1.2 ethact

“ethact” is used to define the currently active interface and to change the required interface without rebooting. If a reboot or a power cycle is done, the active Ethernet interface will be set back to the interface defined in “ethprime” or selected by the “ethrotate” functionality.

Example:

```
=> setenv ethact FM1@DTSEC2
=> ping 172.100.100.35
Using FM1DTSEC2 device
host 172.100.100.35 is alive
=>
```

6.5.4.1.3 ethrotate

“ethrotate” can be used to force the selection of the next available interface if, for example, there is no link available for the selected interface.

If set to “yes” or undefined, U-Boot updates the “ethact” variable accordingly and tries to download the file again. This is repeated until either the file is downloaded or all interfaces have been exhausted.

In the event the link is active for the selected interface and “ethrotate” is “yes” or undefined, U-Boot tries to download the file. If it cannot download the file, it tries the next available interface. If the file is not available on the server, U-Boot stops trying and issues an error message.

If “ethrotate” is set to “no”, only the interface defined in “ethact” is used.

Please note that the setting of the “ethrotate” is lost after a reset. To retain the environment permanently, use the command “saveenv”, which saves the complete environment to flash.

6.5.4.2 Contacting the Server

In addition, to be able to transfer files from a tftp server to a module, the module’s IP address (environment variable “ipaddr”) and the IP address of the server must be set (environment variable “serverip”). Alternatively, it is possible to use the “dhcp” or “bootp” commands.

They can be set using the “setenv” command. Please note that these settings are lost after a reset. To retain the environment permanently, use the command “saveenv”, which saves the complete environment to flash.

To transfer a file from a tftp server to memory, the “tftpboot” command is used, for example:

```
tftpboot 100000 filename
```

6.5.5 Using SD Cards

SD cards are supported (read only) with the “ext2” or “fat” file system.

In both cases, the card must be rescanned first.

```
mmc rescan 0
```

After that, the contents can be verified with:

```
ext2ls mmc 0
```

in case of the ext2 file system, or with

```
fatls mmc 0
```

in case of the fat file system.

To load a file into memory, the commands “ext2load” or “fatload” can be used, for example:

```
ext2load mmc 0 100000 kernel.bin
```

which loads the file “kernel.bin” from the SD card to memory address 0x100000.

6.5.6 Using SATA Devices

SATA devices are supported (read only) with the “ext2” or “fat” file system.

In both cases, the SATA devices must be initialized first.

```
sata init
```

After that, the contents can be verified with:

```
ext2ls sata 0
```

in case of the ext2 file system, or with

```
fatls sata 0
```

in case of the fat file system.

To load a file into memory, the commands “ext2load” or “fatload” can be used, for example:

```
ext2load sata 0 1000000 kernel.bin
```

which loads the file “kernel.bin” from the SATA device to memory address 0x1000000.

6.5.7 Using USB Devices

USB devices are supported (read only) with the “ext2” or “fat” file system.

In both cases, the USB devices must be initialized first.

```
usb start
```

After that, the contents can be verified with:

```
ext2ls usb 0
```

in case of the ext2 file system, or with

```
fatls usb 0
```

in case of the fat file system.

To load a file into memory, the commands “ext2load” or “fatload” can be used, for example:

```
ext2load usb 0 1000000 kernel.bin
```

which loads the file “kernel.bin” from the USB device to memory address 0x1000000.

6.5.8 Using the Onboard NAND Flash

The onboard NAND Flash is supported with the “ubi” filesystem. The access is read only. Thus, the filesystem and its contents must be prepared with Linux first.

As a prerequisite, the environment variables “mtdids” and “mtdparts” must be set correctly.

“mtdids” identifies the NAND chip to use while “mtdparts” defines the partitions.

Example:

```
setenv mtdids nand0=chip1
setenv mtdparts mtdparts=chip1:-(all)
```

This defines the first NAND chip (nand0) to be used with the name “chip1”. The chip contains one partition “all” which occupies the whole chip.

The next command sets the partition “all” to be used with the “ubi” layer:

```
ubi part all
```

Now, an “ubi” volume can be mounted; in this example volume “boot”:

```
ubifsmount boot
```

After the volume is mounted, its contents can be listed:

```
ubifsls
```

or a file loaded, in this case “kernel.bin” to address 0x100000:

```
ubifsload 100000 kernel.bin
```

6.5.9 Using the SPI Flash for OS

The SPI flash for OS is not used together with a file system, it is used raw. It does not contain any U-Boot components and is completely free for user usage. It's primary function is to store VxWorks® boot ROMs and images.

Before making any changes to the flashes, ensure that the correct flash is selected. To select the SPI flash for OS, execute the “sf probe 3” command (SPI flash for OS is routed to the processor’s SPI controller chip select 3).

The SPI flash must be erased before it is programmed. To achieve this, use the “sf erase” command.

To program an image to the SPI flash, it must first be loaded to memory from an arbitrary source. It can then be programmed with the “sf write” command.

Example: Programming a test file “test.img” from an SD card using the “ext2” file system:

```
mmc rescan 0
ext2load mmc 0 100000 test.img
sf probe 3
sf erase 0 10000
sf write 100000 0 ${filesize}
```

This example assumes that the size of “test.img” is less than 64 kB. The environment variable “filesize” is set automatically when a file is loaded to memory and can be used for convenience here.

6.5.10 Booting an OS

6.5.10.1 Booting Linux

To boot Linux, at least a kernel image and a FDT (Flattened Device Tree) must be loaded to memory. Optionally, an “initrd” can be loaded.

Furthermore, a command line must be prepared in the environment variable “bootargs”.

The boot itself is initiated with the “bootm” command.

To simplify the setup of the board, four predefined scripts are already programmed in the default environment:

- “nfsboot” to boot from a tftp server and mount the root over NFS
- “nandboot” to boot from the NAND flash and also mount it as root
- “sdboot” to boot from a SD Card and also mount it as root
- “multi_img_boot” to boot from the multi-image provided. The multi-image consists of a FDT, a kernel and a rootfs

For a one-time-only bootup, this can be accomplished with the “run” command, for example:

```
run nfsboot
```

To make this permanent and have the board execute it automatically, it must be stored in the “bootcmd” environment variable and the environment must be saved to flash.

Example:

```
setenv bootcmd 'run nandboot'
saveenv
```

6.5.10.2 Booting VxWorks ELF Images

The U-Boot boot process of a VxWorks ELF image requires that the following steps be performed:

- Load the VxWorks ELF image from media (Ethernet, Flash) into RAM
- Load the VxWorks FDT (Flattened Device Tree) from media (Ethernet, Flash) into RAM
- Setup and prepare the VxWorks FDT accordingly with the appropriate U-Boot commands:

```
fdt addr $fdtLoadAddr
fdt resize
fdt boardsetup
```

- Boot from the VxWorks ELF image in RAM with U-Boot command

```
bootelf $imgLoadAddr
```

The VxWorks ELF image is generated with a suitable Wind River Workbench project based on Kontron’s VxWorks BSP (Board Support Package) for this product and with an appropriate profile.

Following typical build profiles are supported by the VxWorks BSP:

Table 50: VxWorks Build Profiles

NAME	DESCRIPTION
PROFILE_BOOTAPP	VxWorks boot loader ELF image
PROFILE_STANDALONE_DEVELOPMENT	VxWorks standalone ELF image

The VxWorks FDT binary blob is also provided with the Kontron VxWorks BSP. This binary blob is only a basic FDT framework. The previously mentioned U-Boot fdt-commands are required to prepare and add module specific information to the FDT in RAM which is later retrieved by the VxWorks ELF image from the FDT.

The most common way to load VxWorks ELF images and VxWorks FDT's during development is by transferring them using TFTP over the Ethernet interface. For a finalized application the VxWorks ELF image and VxWorks FDT blob are typically stored in and loaded from the SPI flash for OS.

The boot itself is initiated with the "bootelf" command. To perform autobooting of a VxWorks image requires that appropriate U-Boot environment variables or script(s) be defined for the boot operation to be performed. For more detailed information with examples to boot command sequences, please refer to the Kontron VxWorks BSP online documentation

For more information on how to setup, configure and build VxWorks images and how to utilize them e.g. for a subsequent VxWorks boot process, please refer to the appropriate Wind River VxWorks documentation.

6.6 Getting Help

U-Boot was configured with support for longhelp. This means that online help is available for every command while working with the system. To access the online help, enter "?" or "help" at the console prompt. This will show an overview over all available commands. To get specific help, enter "? <command/command group" or "help <command/command group".

For example to get help on the "saves" command enter "? saves".

```
=> ? saves
saves - save S-Record file over serial line
Usage:
saves [ off ] [size] [ baud ]
    - save S-Record file over serial line with offset 'off', size 'size' and
      baudrate 'baud'
=>
```

To get help on the mmc command group enter "? mmc".

```
=> ? mmc
mmc - MMC sub system
Usage:
mmc read <device num> addr blk# cnt
mmc write <device num> addr blk# cnt
mmc rescan <device num>
mmc part <device num> - lists available partition on mmc
mmc list - lists available devices
=>
```

6.7 Update

The environment contains two scripts which allow an update of various components, e.g. U-Boot, bootrom for VxWorks, data in EEPROMs, etc.

The script "update" checks for a U-Boot script "update" in the directory "update_d0c81" in the first partition of the SD card with "ext2" or "fat" filesystem. If unsuccessful, the check continues with the first NAND chip, volume "boot", and again U-Boot searches in the subdirectory "update_d0c81" for the script "update". If the script "update" is found, it is loaded to memory and executed.

So, to actually execute an update, e.g. an SD card should be prepared with a directory "update_d0c81" on the first partition. Kontron provides an update e.g. for U-Boot as a compressed archive (zip, tar.bz2, tar.gz) which must be unpacked in the directory "update".

After the SD card is inserted, U-Boot should be stopped at the console after power-up. To manually start the update, enter the following command:

```
run update
```

In the case of a U-Boot update, only the standard SPI boot flash is updated.

The script “netupdate” tries to load a U-Boot script “update_d0c81/update” from the server. If found, it is loaded to memory and executed as in the case of the SD card.

As the script “netupdate” requires access to a server, the environment variable “serverip” must be set correctly. Alternatively, it is possible to use the “dhcp” or “bootp” commands.

An automatic run of the update script at every startup takes place if the update script is started in the preboot environment variable:

```
setenv preboot 'run update'
saveenv
```

6.8 Recovery Mechanism

There are two SPI boot flashes available with each device holding a copy of U-Boot. In the event the contents of the standard SPI boot flash have been corrupted (e.g. as a result of a power failure during an update), the recovery SPI boot flash must be selected. This is done by powering the system down, deinstalling the COMe-bP5020 module, setting switch 2 of the SW1 DIP switch to the “on” position, reinstalling the COMe-bP5020 module and then restarting the system.

The board now starts from the recovery SPI boot flash. In this state, the standard SPI boot flash can be programmed again with the “update” or “netupdate” scripts described in Chapter 6.7 Update.

The update scripts provided ensure that prior to the update the standard SPI boot flash is selected and the U-Boot update image is available and correct. Once the update is completed, switch 2 of the SW1 DIP switch must be set to “off” to again allow booting from the standard SPI boot flash.

The contents of the recovery SPI boot flash should never be updated in order to avoid a completely inoperable system with no accessing capability.

6.9 Copyrights and Licensing

U-Boot is Free Software. It is copyrighted by Wolfgang Denk and many others who contributed code (see the actual source code for details). You can redistribute U-Boot and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation. Most of it can also be distributed, at your option, under any later version of the GNU General Public License -- see individual files for exceptions.

NOTE! This license does *not* cover the so-called "standalone" applications that use U-Boot services by means of the jump table provided by U-Boot exactly for this purpose - this is merely considered normal use of U-Boot, and does *not* fall under the heading of “derived work”.

The header files “include/image.h” and “include/asm-*/u-boot.h” define interfaces to U-Boot. Including these (unmodified) header files in another file is considered normal use of U-Boot, and does *not* fall under the heading of “derived work".

Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the U-Boot source code) is copyrighted by me and others who actually wrote it.

-- Wolfgang Denk

===

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software -- to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

6.10 Obtaining Source Code

The software included in this product contains copyrighted software that is licensed under the GPL. A copy of that license is included in this document beginning on page 5. You may obtain the complete corresponding source code from Kontron for a period of three years after our last shipment of this product. Please contact Kontron for further assistance in obtaining the source code.

7 Installation

7.1 Safety

This 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 this product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of personnel safety and of the correct operation of this product, it is recommended to conform with the following guidelines.

- Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspection 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 the system housing before touching the product.

7.2 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 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.

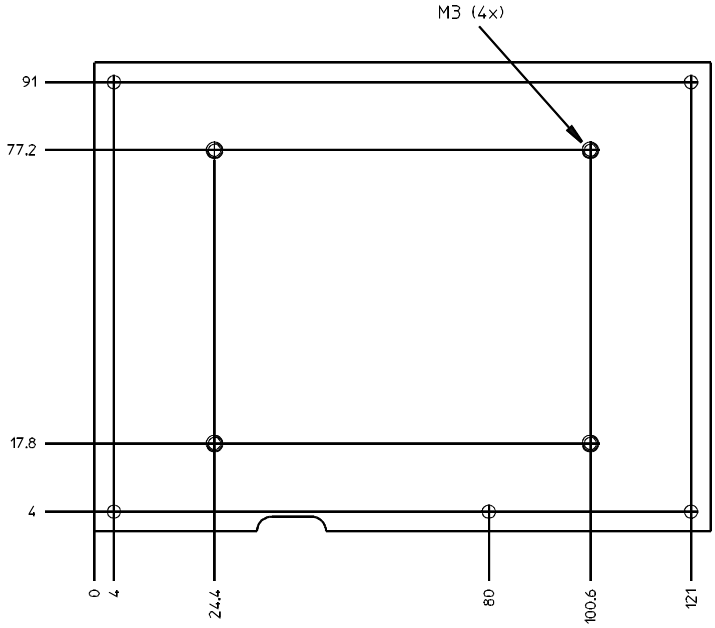
7.3 COM Express® Module-to-Carrier Assembly Considerations

The COMe-bP5020 has been designed to the COM Express® specification for form factor, mechanical dimensions and mounting hole layout. Provisions have also been made for assembly of a heat spreader (two dedicated mounting holes). Kontron offers three cooling solutions (as indicated in Chapter 5) all of which cover the entire upper area of the board. All of the solutions have appropriate holes for mounting hardware (screws, standoffs, etc.).

As each cooling solution is a function of the application, it is the responsibility of the implementer to ensure proper assembly of the COMe-bP5020 with the carrier board and where appropriate attachment of the combined assembly to a chassis wall, a heatpipe or other such devices.

The heat spreader of the COMe-bP5020 has four threaded mounting holes for attaching cooling solutions as indicated in the figure below. Screws used for mounting must not extend beyond the bottom side of the heat spreader when installed, otherwise damage to the COMe-bP5020 will result. The torque applied to these screws when assembling must not exceed 0.7 Nm.

Figure 9: COMe-bP5020 Heatspreader Cooling Solution Mounting Hole Layout



Appendix

1 Warranty

This Kontron product is warranted against defects in material and workmanship for the warranty period from the date of shipment. During the warranty period, Kontron will at its discretion decide to repair or replace defective products.

Within the warranty period, the repair of products is free of charge as long as warranty conditions are observed.

The warranty does not apply to defects resulting from improper or inadequate maintenance or handling by the buyer, unauthorized modification or misuse, operation outside of the product’s environmental specifications or improper installation or maintenance.

Kontron will not be responsible for any defects or damages to other products not supplied by Kontron that are caused by a faulty Kontron product.

2 Proprietary Note

This document contains information proprietary to Kontron. 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 or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron 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 reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron without further notice.

3 Trademarks

This document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

4 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.

CORPORATE OFFICES

Europe, Middle East & Africa

Oskar-von-Miller-Str. 1
85386 Eching / Munich
Germany
Tel.: + 49 (0) 8165 / 77 777
Fax: + 49 (0) 8165 / 77 219
info@kontron.com

North America

14118 Stowe Drive
Poway, CA 92064-7147
USA
Tel.: + 1 888 294 4558
Fax: + 1 858 677 0898
info@us.kontron.com

Asia Pacific

17 Building,Block #1, ABP.
188 Southern West 4th Ring Road
Beijing 100070, P.R.China
Tel.: + 86 10 63751188
Fax: + 86 10 83682438
info@kontron.cn