BNB

Brauchli Netsolution Berg

Brauchli Netsolution Berg
Bahnhofstrasse 7
CH 8572 Berg / Switzerland
Tel: +41/71/6380045
Fax: +41/71/6380046
Email: info@brauchli-netsolution.ch
Web: www.brauchli-netsolution.ch

PCB BNB 446-0-00
QUAD STEP.-MOTOR CNTRL./DRIVER

Table of Contents

1. Functional Description in general

The PCB BNB446 is a digital motor controller and driver for four independent bipolar two phase stepper motors. The motors run on a voltage in the range from 30 V to 50 V in a current controlled mode. The current of each phase can be varied via two switched half-bridges in a range from 100 mA up to 10 A with a resolution of 15 mA and a maximum dc offset of smaller than 30 mA. An over current detection is implemented on a hardware level to prevent damage of the board. To control a wide range of motors with different time constants, it can be choosen between 2 different switching modes and 6 different chopper frequencies in a range from 30 kHz to 240 kHz (30/60/90/120/180/210/240kHz).

Because the current vector is digitally controlled, the motors are always driven in microsteps. With increasing step frequency the number of microsteps per half step is reduced from 32 microsteps per half step at 5 Hz (1 Hz = 1 half step / second) to 1 microstep per half step at 30 kHz.

There are two different drive profiles implemented. For both the trapezoidal and the sinusoidal drive profile, moving distance, acceleration and deceleration, base and top speed as well as hold, acceleration, deceleration and run current can be set. These parameters depend on the physics of the mechanical system but values from 1 kHz / s up to 1000 kHz / s for acceleration resp. deceleration and from 5 Hz to 32760 Hz or switch able from 10Hz to 65520Hz for speed (= half step frequency) are possible (for current range see first section).

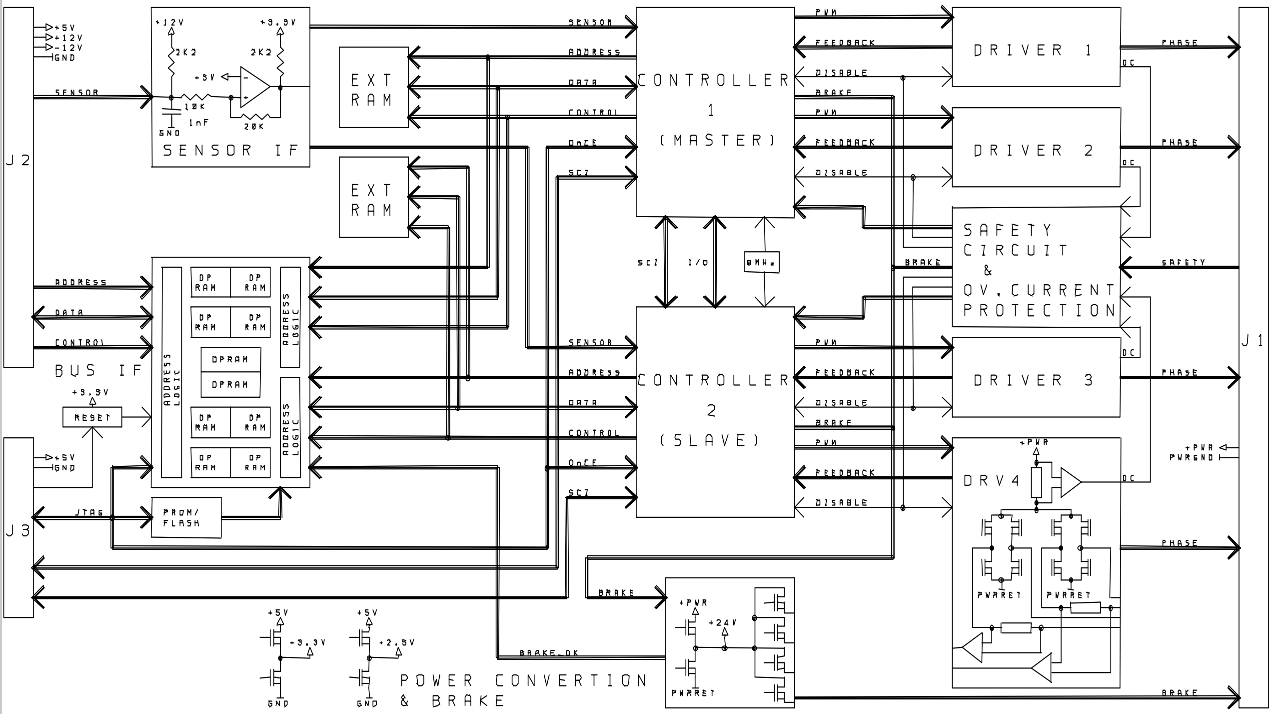
Each motor has four inputs for sensors (i.e. home, limit 1, limit 2, reference) or the digital input can measure PWM signals up to 10 kHz i.e. to drive the motor closed loop. The possibility of a single ended quadrature encoder feedback is also given (with the signals A/B/Index/Zero). The sensor interface is built for 12 V sensors or sensors with open collector outputs. The PWM signal runs on a base frequency of approximately 10 kHz with 12-bit resolution. There is also a sensor controlled safety loop for each motor implemented. This safety loop disables the driver if the software doesn’t react in a certain time. This allows a controlled reaction on an unsafe machine condition.

Each motor has its own brake release output. This output is a high side driver with the capability of driving 0.5 A dc. It automatically gets set / reset with a driver enable / disable signal. The voltage for the four brake outputs is adjustable between 6 V and 35 V (or +PWR - 5 V, whichever is lower).

The electrical / software interface is a BNB owned bus (basically an expanded ISA bus). The mechanical form factor of the BNB446 is 233 mm x 160 mm x 20 mm (9.2” x 6.3” x 0.8”). The needed supply voltages are +5V dc / +12V dc / -12V dc / +PWR (30 V to 50 V dc). Each board voltage is monitored and their value is available on software request.

2. PCB (Hardware)

3. Block diagram

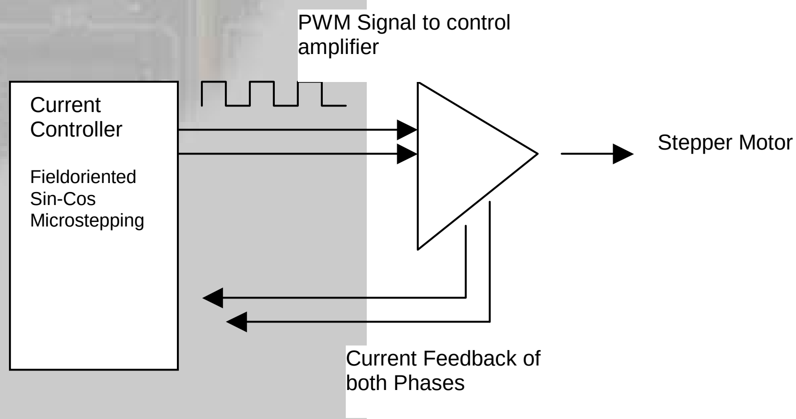


4. Stepper Motor Controller

4.1 Feature Current Controller

The Current is controlled on the BNB446 digitally with a PI regulation. For this type of the current control we need to know the control parameter such as the P-I Value, Control Mode, Chopper frequency and phase-compensation angle. This Parameter needs to be set up prior starting the stepper-motor or engaging the amplifier.

This Current- Controller runs on a 33us base, reads the actual current in both phases (synchronous), uses a field-oriented PI Calculation to calculate the new PWM Value to control and adjust the current, set up by the sinuns-cosinus microstepping.

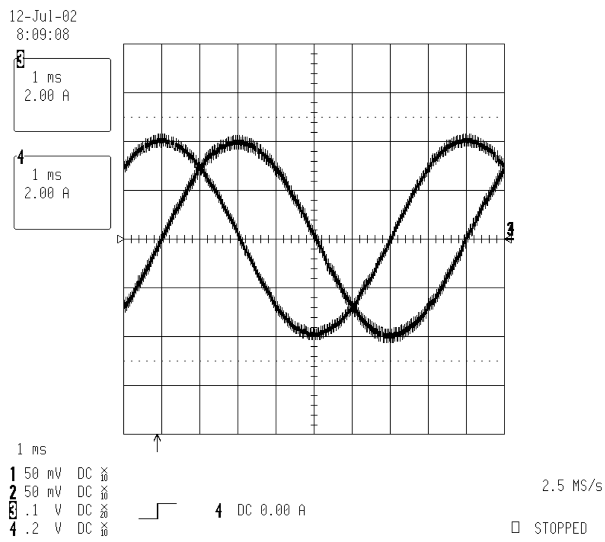


4.3 Feature Motor Identification

Because the Stepper Controller is based on a Digital Current Control we need to know the regulation parameter such as PI, Chopper frequency and so on. This information is saved in a database. When powering up the Stepper-Controller, the stepper controller is going to measure the I vs. t curve. Together with the applied voltage and the R, L Parameter saved in the Database for each Stepper Motor the controller calculates the theoretical I vs. t curve and determines, compares finally which stepper motor is connected to the board. Knowing which stepper is connected; the software is able to setup the parameter for P, I, the chopper frequency, max current of the motor, max. Speed and max. Acceleration. All this information is saved in the motor database on the controller itself. Because the controller is going to measure both phases, the controller is even able to inform the user if there is no stepper motor connected or if one phase has got a short or is not connected. (The theoretical I vs t curve must fit for both phases.)

4.2 Feature Microstepping

The Stepper Controller runs with a Sinus-Cosines-Microstepping with 64 Microsteps. The Stepper uses Microstepping throughout the whole Frequency range from 5Hz up to 32760Hz. See the following picture where the Stepper-Motor runs in Sin-Cos Microstepping at a frequency of 1330Hz (Halfstepspeed) with a setup current of 4 Ampere in each phase. Besides of stopping the Axis in Half or Full- Step the BNB446 is able to stop and hold the stepper motor at any of the 64Microstep positions. Because of the mechanical deviations, mechanical bounce, changing torques, current-resolution, stepper motor magnetism it doesn't make sense to stop it with that high resolution. Reasonable position accuracy will be ¼ or 1/8 Steps. This actually depends only on the specific application and will vary. But the Stepper Controller so far is programmed for 64Micro-Steps.



5. Sensor Interface

The sensor interface is built for 12 V sensors or sensors with open collector outputs with the following specification: V_{iL} 0..5.2V, V_{iH} 7.7..13.2V

5.1 Use as digital Inputs

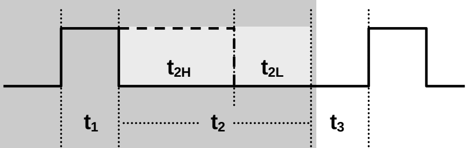
Each motor has four inputs for sensors (i.e. home, limit 1, limit 2, reference). These sensors can be configured to act on different ways. The possible input configuration is as follow: Limit is active if the stepper-motor runs in positive direction, in negative direction or both. If the input is changing either from low to high or high to low (the polarity is configurable) it will cause an action which can be one of the following: decelerate and stop, stop immediately, turn amplifier off or store the actual position into the memory.
Also the inputs can be used and configured as home-inputs and can be setup to drive the stepper into the home position. There is a complex home algorithm implemented to get the best accuracy and to reproduce the home position 100%. This homing procedure compensates the mechanical bounce and play. A digital glitch filter can be configured for all digital inputs.

The value of digital inputs can be read back by the software.

5.2 Use as digital input to measure PWM Signal

The digital input (DI3) can measure PWM signals up to 10 kHz.

Analog Values cannot read directly on the BNB446 Board. So for that reason an analog value must be converted from 0..10V / 0..20mA or whatever it is, into a digital PWM Signal. To convert an analog Signal into a PWM Signal use the BNB448 Board. The Frequency of the PWM Signal is 10kHz. With a sampling rate of 25ns we are able to measure PWM Signals up to a resolution of 2^{10} Bit (=1024)



The signal is modulated as in the following description:
At the beginning and at the end there is at least a high signal of 4us (t_1) (beginning of the cycle) or 4us (t_3) low (at the end of the cycle). Between these times there is the modulated value (t_2), which is split into t_{2H} and t_{2L} corresponding to the analog value. The controller measures the overall time $t_1+t_2+t_3$ and also the time of t_1+t_{2H} and calculated out of this value the analog value. With this calculation the transmission is more or less insensitive to frequency-changings.

Reason of (t_1) and (t_3): The Software can detect a wire break of the PWM Signal transmission. This fault is signalized to the user.

Reason of converting the analog signal into a PWM signal: You are able to convert the analog signal right at the sensor position into a digital signal. Now you can run this digital signal unshielded through your complete machine/ equipment without loosing quality information.

5.3 Use as Encoder Input

The possibility of a single ended quadrature encoder feedback is also given (with the signals A/B/Index/Zero) connected directly to the digital inputs.

5.4 Example of a mixture use of the digital inputs

The first 2 digital inputs can be used as the Encoder Inputs for A and B. The third as a general purpose input configurable as mentioned in the previous chapter and the last can be configured to measure a PWM Signal.

5.5 Safety Interface

There is also a sensor controlled safety loop for each motor implemented. This safety loop disables the driver if the software doesn't react in a certain time. This allows a controlled reaction on an unsafe machine condition.

The Safety loop is realized with hardware and software. The DSP has to signalize with an alive Pin (Toggle a Pin) to the hardware that the DSP is running. If this Pin is stopping from toggling the hardware will turn off the amplifier within 300..500us to bring the amplifier into a save state. Also if the safety switch is open, the DSP has to start another Pin with toggling to keep the amplifier on. If the safety Pin is open, the controller will stop the stepper-motor from running by deceleration. The action, what has to be done on a safety event, can be configured by software.

6. Driver Output / Feedback

Each Amplifier consists of 2, with Power-FET discrete builded, H – Bridges, where each H-Bridge is controlled in a complementary Mode or in a special Mode (actually used for small stepper motor with a low inductivity).
On each Amplifier there is on the High- Voltage side a current supervision build in. This current supervision shuts down the amplifier, if the current will exceed the maximum of 13Amps. This protects the Power-FET and the Board from over current. The measuring of the current for the Current Controller is build into each phase current directly using a current converter.

7. Brake Output

The brake outputs are used to control motor-brakes. The corresponding brake for each axis is turned on as soon as the amplifier is turned off (Holding break). As soon as the amplifier is turned on the brake-output can be controlled by software to release or hold the brake.

The voltage for the brake is generated on board with a PWM Signal out from the DSP Controller. Therefore the Brake voltage can be somewhere between +6V and PWR-5V. Usually it will be 24V. The maximum current should not exceed 2 Amps for all brakes together.

8. Voltage Measurement

All voltages on the BNB446 can be measured and read back to the Host system. This can be used to check the functionality of the BNB445 Board.

9. Software

In the following list there is just a summary/Overview of the commands available on the board. A detailed list with the parameter specification is available.

9.1 Movement Commands

Move Relative, Move Absolute, Homing, Move Forever, Move Sensor controlled (Used for Winder/Unwinder Application), Move Hunt (With Fast and Slow Section, Stops precisely on a sensor input). Max Acceleration, Top- Speed, Base- Speed (The Base speed is the starting speed, where the constant acceleration will start from), Reverse Axis, Trapeze constant Time,

9.2 Current Controller Commands

Amplifier On / Off, Setup Controller Parameter (PI, Chopper frequency), Misc. Current Settings (mA Resolution) for Hold-, Acceleration-, Deceleration- and Run current

9.3 Digital Input Command

Get Actual State of Input, Set Action on Input Event, Configure Input (Polarity, Action, Event), Set Glitch Filter,

9.4 PWM Signal

Get PWM Signal as a fractional Number

9.5 Brake Command

Configure brake (Voltage and regulation parameter), set brake, release brake.

This list is not completed; there are many useful functions not mentioned like set up microstepping resolution, Getpos and so on.

10. Interface to Host System

10.1 Controlling Stepper Ctrl BNB446 by using RS232

The Stepper Controller is controllable using an AxtiveX Element within for example a Homepage. You will be able to create a Homepage placing Buttons, Inputfields/ Outputfields on an Internet Explorer Homepage. Together with an ActiveX Component included into your Homepage which manages all the communication to the stepper controller over RS232 you are able to launch all the commands mentioned under the chapter “Software”.

10.2 BNB owned bus (basically an expanded ISA bus).

For fast communication we recomand to use a parallel communication by connecting the Quad Stepper Quard directly to the ISA Bus. Each Stepper Controller has got a Dual-Ported-Ram wich can be accessed directly out of your Application running on a PC

10.3 Controlling Stepper Ctrl BNB446 by using CAN

A CAN interface including a terminal-block is directly mouted on the PCB.

BNB - Bahnhofstrasse 7 - CH-8572 Berg/TG - Switzerland

11. Technical Specification

11.1 Mechanical

Description	unit	Version 001		
		min	typ	max
PCB lengh	mm	159.70		160.00
PCB height	mm	233.05		233.35
PCB thickness	mm		1.6	
Component height component side	mm			16.00
Component height solder side	mm			2.40
Weight	g		450	

11.2 Electrical

Electrical characteristics and recommended operating conditions over temperature range.

Symbol	Description	unit	Version 001		
			min	typ	max
T_A	Ambient temperature	°C	0		55
V_{+5V}	Supply voltage input logic	V	4.75	5.0	5.25
V_{+3V3}	Supply voltage logic (internal)	V	3.35	3.4	3.45
V_{+2V5}	Supply voltage logic (internal)	V	2.4	2.5	2.6
V_{REF}	Reference volltage A/D (internal)	V	3.25	3.30	3.35
V_{+12V}	Supply voltage input sensors	V	10.8	12.0	13.2
V_{-12V}	Supply voltage input sensors	V	-10.8	-12.0	-13.2
V_{+PWR}	Supply voltage input motors	V	32.4		52.8
V_{+24V}	Supply voltage brakes (internal)	V	6.0	24.0	35.0
I_{+5V}	Supply current input logic	A			1.0
I_{+3V3}	Internal supply current logic (internal)	A			0.6
I_{+2V5}	Internal supply current logic (internal)	A			0.4
I_{+12V}	Supply current input sensors	A			0.5
I_{-12V}	Supply current input sensors	A			0.5
I_{+PWR}	Supply current input motors	A			16.0
I_{+24V}	Supply current brakes (internal)	A			2.0
I_{phase}	Continuous motor phase output current	A			8.0
$I_{solenoid}$	Continuous brake output current	A			0.5
V_{IH}	System address SA[0..11] input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	System data SD[0..7] input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{OH}	System data SD[0..7] output voltage	V	2.4		3.4
V_{OL}		V	0		0.5
V_{IH}	Memory read /MEMR input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Memory write /MEMW input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Bus reset /RST input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Power down /PD input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Interrupt ackn. in /IACKIN input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{OH}	Interrupt ackn. Out /IACKOUT (open drain) output voltage	V	2.4		5.5
V_{OL}		V	0		0.5
V_{OH}	Interrupt request out IRQ (open drain)	V	2.4		5.5

Symbol	Description	unit	Version 001 min	typ	max
V_{OL}	output voltage	V	0		0.5
V_{IH}	Bus synchronisation SYNC input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Broadcast select /BS input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Board select /CS input voltage	V	2.0		5.5
V_{IL}		V	-0.5		0.8
V_{IH}	Sensor signal M[1..4]I[A..D] input voltage	V	7.7		13.2
V_{IL}		V	0		5.2
I_{IH}	Sensor signal M[1..4]I[A..D] input current	mA	0		-2.6
I_{IL}		mA	-3.7		-6.0
V_{IH}	Safety signal input voltage	V	9.8		28.8
V_{IL}		V	0		5.7
I_{IH}	Safety signal input current	mA	0		-6.1
I_{IL}		mA	0		-1.5
V_{IH}	General board /RESET input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8
V_{IH}	JTAG /TRST input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8
V_{IH}	JTAG TCK input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8
V_{IH}	JTAG TMS input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8
V_{IH}	JTAG TDIFPGA input voltage	V	2.0		5.5
V_{IL}	JTAG TDIFLASH input voltage	V	-0.3		0.8
V_{OH}	JTAG TDOFPGA output voltage	V	2.5		3.4
V_{OL}	JTAG TDOFLASH output voltage	V			0.4
V_{IH}	JTAG TDIDSPx input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8
V_{OH}	JTAG TDODSPx output voltage	V	2.5		3.4
V_{OL}		V			0.4
V_{OH}	JTAG /DEDSPx output voltage	V	2.5		3.4
V_{OL}		V			0.4
V_{OH}	TXDDSPx output voltage	V	2.5		3.4
V_{OL}		V			0.4
V_{IH}	RXDDSPx input voltage	V	2.0		5.5
V_{IL}		V	-0.3		0.8

12. Interface Description

12.1 Mechanical

There are three connectors on the BNB446 placed, two of them on the backplaneside. The upper connector (J1) is according to standard DIN 41612 D 32-pole A+C and the lower connector (J2) according to standard DIN 41612 C 64-pole. The third Connector (J3) is on the front side mounted. This 20-pole connector s used to program the Flash of the DSP’s for serial communication with a HostSystem and for debugging purpose.

12.2 Electrical

12.2.1 Connector J1

Pin	Signal	Description
A2	+PWR	Supply voltage input motors
A4	1-OUT1	Motor 1, Phase A
A6	1-OUT3	Motor 1, Phase C
A8	1-SAFETY	Safety signal 1 input
A10	+PWR	Supply voltage input motors

Pin	Signal	Description
A12	2-OUT1	Motor 2, Phase A
A14	2-OUT3	Motor 2, Phase C
A16	2-SAFETY	Safety signal 2 input
A18	+PWR	Supply voltage input motors
A20	3-OUT1	Motor 3, Phase A
A22	3-OUT3	Motor 3, Phase C
A24	/1-BRAKE	Brake 1 high side output
A26	+PWR	Supply voltage input motors
A28	4-OUT1	Motor 4, Phase B
A30	4-OUT3	Motor 4, Phase D
A32	/2-BRAKE	Brake 2 high side output
C2	PWRRET	Supply voltage input motor ground
C4	1-OUT2	Motor 1, Phase B
C6	1-OUT4	Motor 1, Phase D
C8	3-SAFETY	Safety signal 3 input
C10	PWRRET	Supply voltage input motor ground
C12	2-OUT2	Motor 2, Phase B
C14	2-OUT4	Motor 2, Phase D
C16	4-SAFETY	Safety signal 4 input
C18	PWRRET	Supply voltage input motor ground
C20	3-OUT2	Motor 3, Phase B
C22	3-OUT4	Motor 3, Phase D
C24	/3-BRAKE	Brake 3 high side output
C26	PWRRET	Supply voltage input motor ground
C28	4-OUT2	Motor 4, Phase B
C30	4-OUT4	Motor 4, Phase D
C32	/4-BRAKE	Brake 4 high side output

12.2.2 Connector J2

Pin	Signal	Description
A1	+5V	Supply voltage input logic
A2	+5V	Supply voltage input logic
A3	M1IA	Sensor signal motor 1, input A
A4	M2IA	Sensor signal motor 2, input A
A5	M3IA	Sensor signal motor 3, input A
A6	M4IA	Sensor signal motor 4, input A
A7	M1IC	Sensor signal motor 1, input C
A8	M2IC	Sensor signal motor 2, input C
A9	M3IC	Sensor signal motor 3, input C
A10	M4IC	Sensor signal motor 4, input C
A11	NC	Not connected
A12	SD0	System data input / output (ISA bus)
A13	SD2	System data input / output (ISA bus)
A14	SD4	System data input / output (ISA bus)
A15	SD6	System data input / output (ISA bus)
A16	SA0	System address input (ISA bus)
A17	SA2	System address input (ISA bus)
A18	SA4	System address input (ISA bus)
A19	SA6	System address input (ISA bus)
A20	SA8	System address input (ISA bus)
A21	SA10	System address input (ISA bus)
A22	NC	Not connected (board counter circuit on test board)
A23	/MEMR	Memory read input (ISA bus)
A24	/RST	Bus reset input (ISA bus)
A25	/IACKIN	Interrupt ackn. input (ISA bus)

Pin	Signal	Description
A26	/IACKOUT	Interrupt ackn. output (ISA bus)
A27	NC	Not connected
A28	SYNC	Bus synchronisation input (ISA bus)
A29	+12V	Supply voltage input sensors
A30	-12V	Supply voltage input sensors
A31	GND	Supply voltage input logic ground
A32	GND	Supply voltage input logic ground
C1	+5V	Supply voltage input logic
C2	+5V	Supply voltage input logic
C3	M1IB	Sensor signal motor 1, input B
C4	M2IB	Sensor signal motor 2, input B
C5	M3IB	Sensor signal motor 3, input B
C6	M4IB	Sensor signal motor 4, input B
C7	M1ID	Sensor signal motor 1, input D
C8	M2ID	Sensor signal motor 2, input D
C9	M3ID	Sensor signal motor 3, input D
C10	M4ID	Sensor signal motor 4, input D
C11	NC	Not connected
C12	SD1	System data input / output (ISA bus)
C13	SD3	System data input / output (ISA bus)
C14	SD5	System data input / output (ISA bus)
C15	SD7	System data input / output (ISA bus)
C16	SA1	System address input (ISA bus)
C17	SA3	System address input (ISA bus)
C18	SA5	System address input (ISA bus)
C19	SA7	System address input (ISA bus)
C20	SA9	System address input (ISA bus)
C21	SA11	System address input (ISA bus)
C22	NC	Not connected
C23	/MEMW	Memory write input (ISA bus)
C24	/PD	Power down input (ISA bus)
C25	/IRQ	Interrupt request output (ISA bus)
C26	NC	Not connected
C27	/BS	Broadcast select input (ISA bus)
C28	/CS	Board select input (ISA bus)
C29	+12V	Supply voltage input sensors
C30	-12V	Supply voltage input sensors
C31	GND	Supply voltage input logic ground
C32	GND	Supply voltage input logic ground

12.2.3 Connector J3

Pin	Signal	Description
1	GND	Supply voltage output logic ground
2	+5V	Supply voltage output logic
3	/RESET	General board reset input
4	/TRST	JTAG I/F test reset input
5	TCK	JTAG I/F test clock input
6	TMS	JTAG I/F test mode select input
7	TDIFPGA	JTAG I/F test data input to FPGA
8	TDOFPGA	JTAG I/F test data output from FPGA
9	TDIFLASH	JTAG I/F test data input to FLASH
10	TDOFLASH	JTAG I/F test data output from FLASH
11	TDIDSP1	JTAG I/F test data input to DSP 1
12	TDODSP1	JTAG I/F test data output from DSP 1
13	/DEDSP1	On chip emulation debug event output from DSP 1

Pin	Signal	Description
14	TDIDSP2	JTAG I/F test data input to DSP 2
15	TDODSP2	JTAG I/F test data output from DSP 2
16	/DEDSP2	On chip emulation debug event output from DSP 2
17	TXDDSP1	SCI transmitter from DSP 1
18	RXDDSP1	SCI receiver from DSP 1
19	TXDDSP2	SCI transmitter from DSP 2
20	RXDDSP2	SCI receiver from DSP 2

Revision History:

Rev:	Date	By:	Description:
A	06.11.02	CB	Initial Version
B	05.03.03	CB	Add new Features