The Embedded I/O Company

TEWS TECHNOLOGIES

TPMC806

Conduction Cooled PMC,

Isolated 2 x CAN and MIC Bus

Version 1.0

User Manual

Issue 1.0

July 2002

D 76806800

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7
25469 Halstenbek / Germany
Phone: +49-(0)4101-4058-0
Fax: +49-(0)4101-4058-19
e-mail: info@tews.com
www.tews.com

TEWS TECHNOLOGIES LLC
1 E. Liberty Street, Sixth Floor
Reno, Nevada 89504 / USA
Phone: +1 (775) 686 6077
Fax: +1 (775) 686 6024
e-mail: usasales@tews.com
www.tews.com

TPMC806-10

Conduction Cooled PMC, Isolated 2 x CAN and MIC Bus; Standard front panel; P14 Back-I/O

TPMC806-11

Conduction Cooled PMC, Isolated 2 x CAN and MIC Bus; Without front panel; P14 Back-I/O

Style Conventions

Hexadecimal characters are specified with prefix 0x, i.e. 0x029E (that means hexadecimal value 029E).

For signals on hardware products, an ‚Active Low' is represented by the signal name with # following, i.e. IP_RESET#.

Issue	Description	Date
1.0	First Issue	July 2002

Table of Contents

Table of Figure

1 Product Description

The TPMC806 is a conduction cooled single-width 32-bit PMC module providing two channel of high speed CAN bus interface and a MIC bus interface.

The PLX PCI9030 is used as PCI target chip.

Two Philips SJA1000 CAN controllers (CAN specification 2.0B supported) are used for the two CAN bus channels.

The CAN bus I/O interface provides two independent channels, isolated from system logic, from the MIC bus interface and from each other.

CAN High Speed transceivers are used for the CAN bus I/O interface.

An on board termination option (solder pads) is provided for each CAN bus channel allowing to configure on board termination and/or pass trough mode for the CAN bus.

A Vetronix MIC-320GM MIC bus controller is used for the MIC bus interface.

The MIC bus I/O interface provides two channels isolated from the system logic and from the CAN bus interface.

ESD protected high speed RS485 transceivers are used for each MIC bus channel.

An on board termination option (solder pads) is provided for each MIC bus channel allowing to configure on board termination and/or pass trough mode for the MIC bus.

The TPMC806 uses the P14 I/O connector for the CAN bus and MIC bus I/O interface.

Two versions of the TPMC806 are available: the TPMC806-10 with a standard front panel allows using the board in standard PMC slots and the TPMC806-11 which has no front panel is for use in conduction cooled PMC slots.

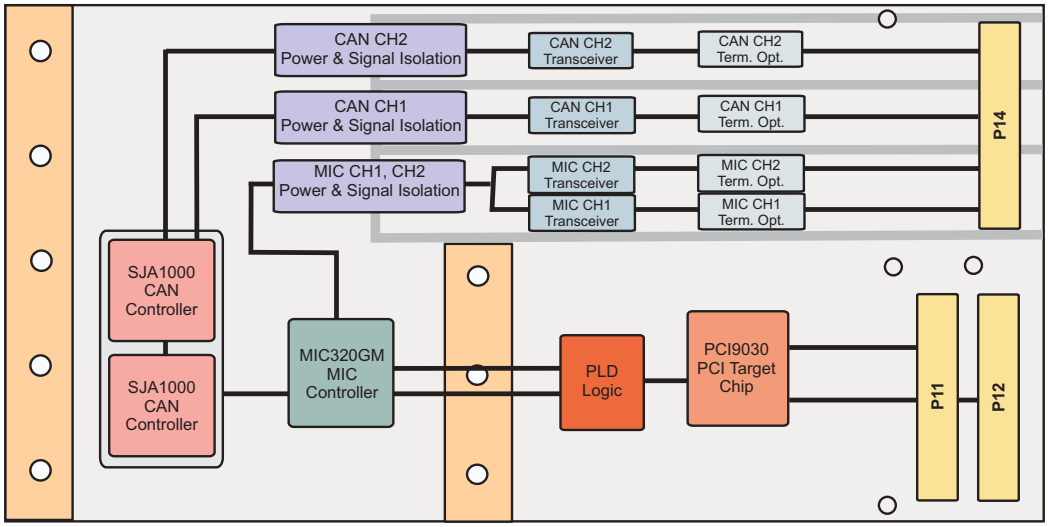


Figure 1-1: Block Diagram TPMC806

2 Technical Specification

Logic Interface	
Mechanical Interface	PCI Mezzanine Card (PMC) Interface Conduction Cooled Single Size
Electrical Interface	PCI Rev. 2.2 compliant 33MHz / 32-bit PCI 3.3V and 5V PCI Signaling Voltage
On board Devices	
PCI Target Chip	PCI9030 (PLX Technology)
MIC Controller	1 x MIC-320GM (Vetronix) (16MHz)
MIC Bus Transceiver	2 x MAX3088E (High Speed RS485 Half Duplex) (Maxim)
CAN Controller	2 x SJA1000 (Philips) (24MHz)
CAN Bus Transceiver	2 x TJA1050 (Philips)
I/O Interface	
Number of MIC Bus Channels	2
MIC Bus Interface	RS485 Half Duplex
Number of CAN Bus Channels	2
CAN Bus Interface	CAN High Speed
I/O Connector	PMC P14 I/O (64 pin Mezzanine Connector)
Physical Data	
Power Requirements	145mA typical @+ 3.3V DC 535mA typical (no load) @+ 5.0V DC 765mA typical (loaded) @+ 5.0V DC
Temperature Range	Storage : -55°C to +125°C Operating : -40°C to + 85°C
MTBF	Tbd
Weight	76g
Humidity	5 – 95 % non-condensing

Figure 2-1: Technical Specification

3 TPMC 806 Local Space Addressing

3.1 PCI9030 Local Space Configuration

The local on board addressable regions are accessed from the PCI side by using the PCI9030 local spaces.

PCI9030 LOCAL SPACE CONFIGURATION						
PCI9030 Local Space	**PCI9030 PCI Base Address (Offset in PCI Configuration Space)**	**PCI Space Mapping**	**Size (Byte)**	**Port Width (Bit)**	**Endian Mode**	**Description**
0	2 (0x18)	MEM	16	8	BIG	PLD Register Address Space
1	3 (0x1C)	MEM	64	32	BIG	MIC Controller Address Space
2	4 (0x20)	MEM	512	8	BIG	CAN Controller Address Space
3	5 (0x24)	-	-	-	-	Not Used

Figure 3-1 : PCI9030 Local Space Configuration

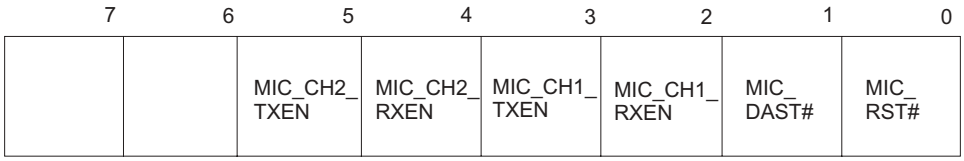
3.2 PLD Register Space

PCI Base Address: **PCI9030 PCI Base Address 2 (Offset 0x18 in PCI Configuration Space).**

PLD REGISTER SPACE		
Offset to PCI Base Address 2	**Register Name**	**Size (Bit)**
0x00	MIC CONTROL REGISTER	8
0x01	MIC STATUS REGISTER	8
0x02	CAN CONTROL REGISTER	8
0x03	CAN STATUS REGISTER	8
0x04	INTERRUPT ENABLE REGISTER	8
0x05	INTERRUPT STATUS REGISTER	8
0x06	MIC INTERRUPT ACKNOWLEDGE REGISTER	8
0x07 … 0x0F	Reserved	-

Figure 3-2 : PLD Register Space

3.2.1 MIC Control Register



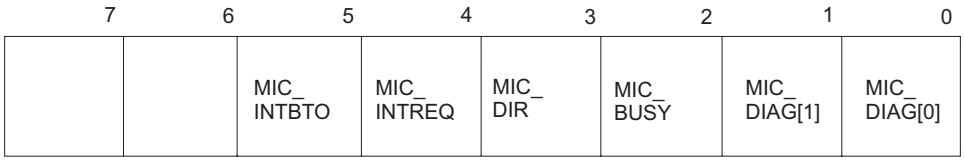
MIC CONTROL REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads. Write as '0'.)	-	X
6	-	Reserved (Undefined for reads. Write as '0'.)	-	X
5	MIC_CH2_TXEN	0 : MIC CH2 Transmitter Disabled 1 : MIC Chip TXEN Signal is used for MIC CH2 Transmitter Enable/Disable	R/W	0
4	MIC_CH2_RXEN	0 : MIC CH2 Receiver Disabled 1 : MIC Chip RXEN# Signal is used for MIC CH2 Receiver Enable/Disable	R/W	0
3	MIC_CH1_TXEN	0 : MIC CH1 Transmitter Disabled 1 : MIC Chip TXEN Signal is used for MIC CH1 Transmitter Enable/Disable	R/W	0
2	MIC_CH1_RXEN	0 : MIC CH1 Receiver Disabled 1 : MIC Chip RXEN# Signal is used for MIC CH1 Receiver Enable/Disable	R/W	0
1	MIC_DAST#	0 : MIC Chip Self-Test Mode Disabled (MIC_DAST# = 0) 1 : MIC Chip Self-Test Mode Enabled (MIC_DAST# = 1)	R/W	0
0	MIC_RST#	0 : MIC Chip in Reset Mode (MIC_RST# = MIC_SRST# = 0) 1 : MIC Chip in Operating Mode (MIC_RST# = MIC_SRST# = 1)	R/W	0

Figure 3-3 : MIC Control Register

After power-up or board reset, the MIC-320 controller is held in reset mode. To bring the MIC-320 chip out of reset mode, a write to the MIC Control Register is required.

If used, the MIC self-test mode must be enabled in an extra access before the MIC chip is set to operating mode (i.e. first write 0x02, then write 0x3F to the MIC Control Register).

3.2.2 MIC Status Register



MIC STATUS REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads)	-	X
6	-	Reserved (Undefined for reads)	-	X
5	MIC_INTBTO	MIC Chip INTBTO Bus Timeout Interrupt Status	R	X
4	MIC_INTREQ	MIC Chip INTREQ Interrupt Request Status	R	X
3	MIC_DIR	MIC Chip DIR Signal Status	R	X
2	MIC_BUSY	MIC Chip BUSY Signal Status	R	X
1	MIC_DIAG[1]	MIC Chip DIAG[1] Signal Status	R	X
0	MIC_DIAG[0]	MIC Chip DIAG[0] Signal Status	R	X

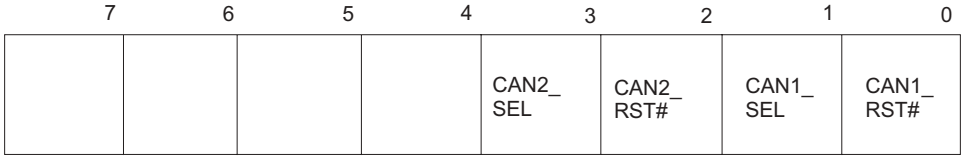
Figure 3-4 : MIC Status Register

The MIC Chip INTREQ interrupt request status is '0' for "no active interrupt request" and '1' for "active interrupt request".

The MIC Chip INTBTO bus timeout interrupt signal status is '0' for "no active interrupt request" and '1' for "active interrupt request".

See the MIC-320 Manual for more information.

3.2.3 CAN Control Register



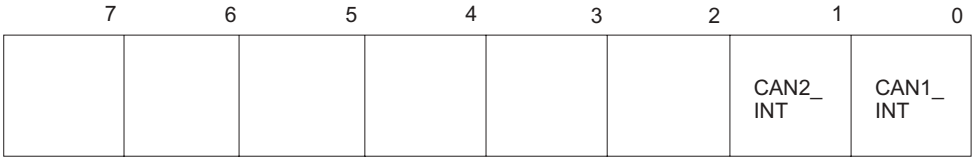
CAN CONTROL REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads. Write as '0'.)	-	X
6	-	Reserved (Undefined for reads. Write as '0'.)	-	X
5	-	Reserved (Undefined for reads. Write as '0'.)	-	X
4	-	Reserved (Undefined for reads. Write as '0'.)	-	X
3	CAN2_SEL	0 : CAN CH2 Transceiver Silent Mode 1 : CAN CH2 Transceiver Operating Mode	R/W	0
2	CAN2_RST#	0 : CAN CH2 Controller Reset Mode (CAN2_RST# = 0) 1 : CAN CH2 Controller Operating Mode (CAN2_RST# = 1)	R/W	0
1	CAN1_SEL	0 : CAN CH1 Transceiver Silent Mode 1 : CAN CH1 Transceiver Operating Mode	R/W	0
0	CAN1_RST#	0 : CAN CH1 Controller Reset Mode (CAN1_RST# = 0) 1 : CAN CH1 Controller Operating Mode (CAN1_RST# = 1)	R/W	0

Figure 3-5 : CAN Control Register

After power-up or board reset the CAN controllers are held in reset mode.

To bring the CAN controller chips out of reset mode, a write to the CAN Control Register is required.

3.2.4 CAN Status Register



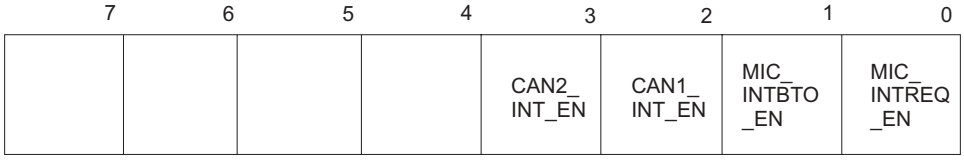
CAN STATUS REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads)	-	X
6	-	Reserved (Undefined for reads)	-	X
5	-	Reserved (Undefined for reads)	-	X
4	-	Reserved (Undefined for reads)	-	X
3	-	Reserved (Undefined for reads)	-	X
2	-	Reserved (Undefined for reads)	-	X
1	CAN2_INT	CAN CH2 Controller Interrupt Request Status	R	X
0	CAN1_INT	CAN CH1 Controller Interrupt Request Status	R	X

Figure 3-6 : CAN Status Register

The CAN CHx controller interrupt request status is '0' for "no active interrupt request" and '1' for "active interrupt request".

See the SJA1000 Manual for more information.

3.2.5 Interrupt Enable Register

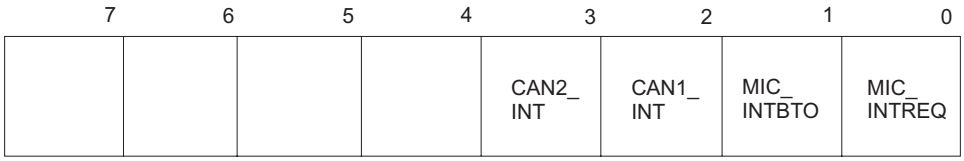


INTERRUPT ENABLE REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads. Write as '0'.)	-	X
6	-	Reserved (Undefined for reads. Write as '0'.)	-	X
5	-	Reserved (Undefined for reads. Write as '0'.)	-	X
4	-	Reserved (Undefined for reads. Write as '0'.)	-	X
3	CAN2_INT_EN	0 : CAN CH2 Interrupt Disabled 1 : CAN CH2 Interrupt Enabled	R/W	0
2	CAN1_INT_EN	0 : CAN CH1 Interrupt Disabled 1 : CAN CH1 Interrupt Enabled	R/W	0
1	MIC_INTBTO_EN	0 : MIC Bus Timeout Interrupt Disabled 1 : MIC Bus Timeout Interrupt Enabled	R/W	0
0	MIC_INTREQ_EN	0 : MIC Interrupt Request Disabled 1 : MIC Interrupt Request Enabled	R/W	0

Figure 3-7 : Interrupt Enable Register

Disabling interrupts in the Interrupt Enable Register only effects the interrupt mapping to the PCI9030 LINT1# local interrupt input, it will not effect the interrupt source.

3.2.6 Interrupt Status Register



INTERRUPT STATUS REGISTER				
Bit	**Name**	**Function**	**Access**	**Reset**
7	-	Reserved (Undefined for reads)	-	X
6	-	Reserved (Undefined for reads)	-	X
5	-	Reserved (Undefined for reads)	-	X
4	-	Reserved (Undefined for reads)	-	X
3	CAN2_INT	CAN CH2 Interrupt Request Status 0 : No Active Interrupt Request 1 : Active Interrupt Request	R	X
2	CAN1_INT	CAN CH1 Interrupt Request Status 0 : No Active Interrupt Request 1 : Active Interrupt Request	R	X
1	MIC_INTBTO	MIC Bus Timeout Interrupt Status 0 : No Active Interrupt Request 1 : Active Interrupt Request	R	X
0	MIC_INTREQ	MIC Interrupt Request Status 0 : No Active Interrupt Request 1 : Active Interrupt Request	R	X

Figure 3-8 : Interrupt Status Register

All four interrupt sources are mapped to the PCI9030 LINT1# local interrupt input.

The PCI9030 LINT1# local interrupt input is used in active low level sensitive mode.

The PCI9030 LINT2# local interrupt input is not used.

MIC interrupts must be acknowledged by a write to the MIC Interrupt Acknowledge register (PLD Register Space).

CAN interrupts must be acknowledged via SJA1000 registers (CAN Controller Register Space).

See the SJA1000 Manual for more information.

3.2.7 MIC Interrupt Acknowledge Register

If either the MIC-320 chip signal INTREQ# or INTBTO# is asserted, any write to the MIC Interrupt Acknowledge Register will assert an active pulse to the MIC chip IACK# input. The IACK# pulse remains active (low) until both MIC-320 interrupt signals INTREQ# and INTBTO # are negated.

MIC interrupts must be acknowledged by a write to the MIC Interrupt Acknowledge register.

See the MIC-320 Manual for more information.

Undefined results for reads.

3.3 MIC Controller Register Space

PCI Base Address: **PCI9030 PCI Base Address 3 (Offset 0x1C in PCI Configuration Space).**

MIC CONTROLLER REGISTER SPACE				
Offset to PCI Base Address 3	**Register Name**	**Access**	**Size (Bit)**	**Bits Used**
0x00	Bus Timeout Counter	W	32	[15:0]
0x04	No Command Timeout Counter	W	32	[15:0]
0x08	No Response Timeout Counter	W	32	[9:0]
0x0C	Interrupt Acknowledge Timeout Counter	W	32	[9:0]
0x10	Setup Register	W	32	[15:0]
0x14	Status Register	R	32	[22:0]
0x18	Module Address / Base Vector Number	W	32	[13:8] / [7:0]
0x1C	Transmit Buffer	W	32	[31:0]
0x20	Receive Buffer	R	32	[31:0]
0x24 … 0x3F	Reserved	-	-	-

Figure 3-9 : MIC Controller Register Space

Access to MIC controller registers when MIC controller is held in reset mode: The access will terminate normally. Write data is ignored. Read data is undefined.

Access to MIC controller registers with wrong access direction: The access will terminate normally. Write data is ignored. Read data is undefined.

Access to MIC controller registers with wrong transfer size: The access will terminate normally. Write data is ignored. Read data is undefined.

The MIC controller must be set to operating mode (MIC Control Register in PLD Register Space) before MIC controller register access.

32-bit transfer size must be used for all MIC controller register access. Other transfer sizes are not supported.

Do not perform read access to the MIC controller write-only registers. Do not perform write access to the MIC controller read-only registers.

See the MIC-320 Manual for a detailed register description.

3.4 CAN Controller Register Space

PCI Base Address: **PCI9030 PCI Base Address 4 (Offset 0x20 in PCI Configuration Space).**

CAN CONTROLLER REGISTER SPACE		
Offset to PCI Base Address 4	**Register Name**	**Size (Bit)**
CAN Controller Channel 1		
0x000	CAN Controller CH1 Address 0	8
0x001	CAN Controller CH1 Address 1	8
0x002	CAN Controller CH1 Address 2	8
…	…	…
0x07F	CAN Controller CH1 Address 127	8
0x080 … 0x0FF	Reserved	-
CAN Controller Channel 2		
0x100	CAN Controller CH2 Address 0	8
0x101	CAN Controller CH2 Address 1	8
0x102	CAN Controller CH2 Address 2	8
…	…	…
0x17F	CAN Controller CH2 Address 127	8
0x180 … 0x1FF	Reserved	-

Figure 3-10 : CAN Controller Register Space

The CAN controllers must be set to operating mode (CAN Control Register in PLD Register Space) before CAN controller register access.

See the SJA1000 Manual for a detailed register description.

4 PCI 9030 Target Chip

4.1 PCI Configuration Registers (PCR)

4.1.1 PCI9030 PCI Header

PCI CFG Register Address	Write '0' to all unused (Reserved) bits				PCI writeable	Initial Values (Hex Values)
	31 24	23 16	15 8	7 0		
0x00	Device ID		Vendor ID		N	bmmm 1498
0x04	Status		Command		Y	0280 0000
0x08	Class Code			Revision ID	N	070200 00
0x0C	BIST	Header Type	PCI Latency Timer	Cache Line Size	Y[7:0]	00 00 00 00
0x10	PCI Base Address 0 for MEM Mapped Config. Registers				Y	FFFFFFE0
0x14	PCI Base Address 1 for I/O Mapped Config. Registers				Y	FFFFFFE1
0x18	PCI Base Address 2 for Local Address Space 0				Y	FFFFFFF0
0x1C	PCI Base Address 3 for Local Address Space 1				Y	FFFFFFC0
0x20	PCI Base Address 4 for Local Address Space 2				Y	FFFFFE00
0x24	PCI Base Address 5 for Local Address Space 3				Y	00000000
0x28	PCI Cardbus Information Structure Pointer				N	00000000
0x2C	Subsystem ID		Subsystem Vendor ID		N	b0oo 1498
0x30	PCI Base Address for Local Expansion ROM				Y	00000000
0x34	Reserved			New Cap. Ptr.	N	000000 40
0x38	Reserved				N	00000000
0x3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	Y[7:0]	00 00 00 01
0x40	PM Cap.		PM Nxt Cap.	PM Cap. ID	Y	4801 48 01
0x44	PM Data	PM CSR EXT	PM CSR		Y	00 00 0000
0x48	Reserved	HS CSR	HS Nxt Cap.	HS Cap. ID	Y[23:16]	00 00 4C 06
0x4C	VPD Address		VPD Nxt Cap.	VPD Cap. ID	Y[31:16]	0000 00 03
0x50	VPD Data				Y	00000000

Figure 4-1 : PCI9030 PCI Header

Vendor-ID:	0x1498	TEWS Technologies
Device-ID (module dependent) :	*bmmm*	TEWS Module Bus-Type & Module-ID
Subsystem-Vendor-ID:	0x1498	TEWS Technologies
Subsystem-ID (module dependent):	*b0oo*	TEWS Module Bus-Type & Board-Option

b TEWS Module Bus-Type Coding

PMC	0x0
PC-MIP	0x1
Compact-PCI	0x2
Standard-PCI	0x3

mmm TEWS Module-ID hexadecimal coded

oo TEWS Module Board-Option hexadecimal coded

Example: TPMC806-10

Device-ID:	0x0326 (TPMC806)
Vendor-ID:	0x1498 (TEWS Technologies)
Subsystem-ID:	0x000A (Board Option -10)
Subsystem-Vendor-ID:	0x1498 (TEWS Technologies)

4.1.2 PCI Base Address Initialization

PCI Base Address Initialization is scope of the PCI host software.

PCI9030 PCI Base Address Initialization:

1. Write 0xFFFF_FFFF to the PCI9030 PCI Base Address Register.
2. Read back the PCI9030 PCI Base Address Register
3. For PCI Base Address Registers 0:5, check bit 0 for PCI Address Space.

   Bit 0 = '0' requires PCI Memory Space mapping

   Bit 0 = '1' requires PCI I/O Space mapping

   For the PCI Expansion ROM Base Address Register, check bit 0 for usage.

   Bit 0 = ‘0’: Expansion ROM not used

   Bit 0 = ‘1’: Expansion ROM used
4. For PCI I/O Space mapping, starting at bit location 2, the first bit set determines the size of the required PCI I/O Space size.

   For PCI Memory Space mapping, starting at bit location 4, the first bit set to '1' determines the size of the required PCI Memory Space size.

   For PCI Expansion ROM mapping, starting at bit location 11, the first bit set to '1' determines the required PCI Expansion ROM size.

   For example, if bit 5 of a PCI Base Address Register is detected as the first bit set to ‘1’, the PCI 9030 is requesting a 32 byte space (address bits 4:0 are not part of base address decoding).
5. Determine the base address and write the base address to the PCI9030 PCI Base Address Register. For PCI Memory Space mapping the mapped address region must comply with the definition of bits 3:1 of the PCI9030 PCI Base Address Register.

After programming the PCI9030 PCI Base Address Registers, the software must enable the PCI9030 for PCI I/O and/or PCI Memory Space access in the PCI9030 PCI Command Register (Offset 0x04). To enable PCI I/O Space access to the PCI9030, set bit 0 to '1'. To enable PCI Memory Space access to the PCI9030, set bit 1 to '1'.

4.2 Local Configuration Register (LCR)

After reset, the PCI9030 Local Configuration Registers are loaded from the on-board serial configuration EEPROM.

The PCI base address for the PCI9030 Local Configuration Registers is PCI9030 PCI Base Address 0 (PCI Memory Space) (Offset 0x10 in the PCI9030 PCI Configuration Register Space) or PCI9030 PCI Base Address 1 (PCI I/O Space) (Offset 0x14 in the PCI9030 PCI Configuration Register Space).

Do not change hardware dependent bit settings in the PCI9030 Local Configuration Registers.

PCI9030 LOCAL CONFIGURATION REGISTER		
Offset from PCI Base Address	**Register**	**Value**
0x00	Local Address Space 0 Range	0x0FFF_FFF0
0x04	Local Address Space 1 Range	0x0FFF_FFC0
0x08	Local Address Space 2 Range	0x0FFF_FE00
0x0C	Local Address Space 3 Range	0x0000_0000
0x10	Local Exp. ROM Range	0x0000_0000
0x14	Local Re-map Register Space 0	0x0000_0001
0x18	Local Re-map Register Space 1	0x0000_1001
0x1C	Local Re-map Register Space 2	0x0000_2001
0x20	Local Re-map Register Space 3	0x0000_0000
0x24	Local Re-map Register ROM	0x0000_0000
0x28	Local Address Space 0 Descriptor	0x1500_C0A0
0x2C	Local Address Space 1 Descriptor	0x1580_C062
0x30	Local Address Space 2 Descriptor	0x1502_4120
0x34	Local Address Space 3 Descriptor	0x0000_0000
0x38	Local Exp. ROM Descriptor	0x0000_0000
0x3C	Chip Select 0 Base Address	0x0000_0009
0x40	Chip Select 1 Base Address	0x0000_1021
0x44	Chip Select 2 Base Address	0x0000_2081
0x48	Chip Select 3 Base Address	0x0000_2181
0x4C	Interrupt Control/Status	0x0041
0x4E	EEPROM Write Protect Boundary	0x0030
0x50	Miscellaneous Control Register	0x0078_0000
0x54	General Purpose I/O Control	0x026D_B6D2
0x70	Hidden1 Power Management data select	0x0000_0000
0x74	Hidden 2 Power Management data scale	0x0000_0000

Figure 4-2 : PCI9030 Local Configuration Register

4.3 Configuration EEPROM

After power-on or PCI reset, the PCI 9030 loads initial configuration register data from the on-board configuration EEPROM.

The configuration EEPROM contains the following configuration data:

- Address 0x00 to 0x27 : PCI9030 PCI Configuration Register Values
- Address 0x28 to 0x87 : PCI9030 Local Configuration Register Values
- Address 0x88 to 0xFF : Reserved

See the PCI9030 Manual for more information.

TPMC806-xx CONFIGURATION EEPROM								
Address	**Offset**							
	0x00	0x02	0x04	0x06	0x08	0x0A	0x0C	0x0E
0x00	0x0326	0x1498	0x0280	0x0000	0x0780	0x0000	s.b.	0x1498
0x10	0x0000	0x0040	0x0000	0x0100	0x4801	0x4801	0x0000	0x0000
0x20	0x0000	0x4C06	0x0000	0x0003	0x0FFF	0xFFF0	0x0FFF	0xFFC0
0x30	0x0FFF	0xFE00	0x0000	0x0000	0x0000	0x0000	0x0000	0x0001
0x40	0x0000	0x1001	0x0000	0x2001	0x0000	0x0000	0x0000	0x0000
0x50	0x1500	0xC0A0	0x1580	0xC062	0x1502	0x4120	0x0000	0x0000
0x60	0x0000	0x0000	0x0000	0x0009	0x0000	0x1021	0x0000	0x2081
0x70	0x0000	0x2181	0x0030	0x0041	0x0078	0x0000	0x026D	0xB6D2
0x80	0x0000	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0x90	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xA0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xB0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xC0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xD0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xE0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xF0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF

Figure 4-3 : Configuration EEPROM TPMC806-xx

Subsystem-ID Value (Offset 0x0C): TPMC806-10 0x000A

TPMC806-11 0x000B

4.4 Local Software Reset

The PCI9030 Local Reset Output LRESETo# is used to reset the on-board local logic.

The PCI9030 local reset is active during PCI reset or if the PCI Adapter Software Reset bit is set in the PCI9030 local configuration register CNTRL (offset 0x50).

CNTRL[30] PCI Adapter Software Reset:

Value of 1 resets the PCI 9030 and issues a reset to the Local Bus (LRESETo# asserted). The PCI 9030 remains in this reset condition until the PCI Host clears this bit. The contents of the PCI9030 PCI and Local Configuration Registers are not reset. The PCI9030 PCI Interface is not reset.

5 Programming Hints

5.1 MIC Controller (MIC-320)

The MIC-320 clock input frequency is 16MHz.

On the TPMC806 the MIC-320 operates in Processor Interface Mode (PIM) only.

After power-up or board reset, the MIC-320 controller is held in reset mode.

To bring the MIC-320 chip out of reset mode, a write to the MIC Control Register is required.

If used, the MIC self-test mode must be enabled in an extra access before the MIC chip is set to operating mode (i.e. first write 0x02, then write 0x3F to the MIC Control Register).

32-bit transfer size access must be used for all MIC-320 registers. The access direction for each MIC controller register must be appropriate.

Any incorrect MIC controller register access (transfer size, access direction, access in reset mode) will terminate normally. However, write data is ignored and read data is undefined.

The TPMC806 does not support vector number bus cycles for interrupt processing. The MIC-320 Base Vector Number register should be set to 0x00. The interrupt vector bits VECTOR[1:0] can be read in the MIC-320 Status Register.

Bit14 of the MIC-320 Setup Register should be set to '1' (use Module Address Register), and a valid module address should be programmed to the MIC-320 Module Address Register.

5.2 CAN Controller (SJA1000)

The SJA1000 clock input frequency is 24MHz (for both SJA1000 controllers).

After power-up or board reset the CAN controller chips are held in reset mode.

To bring the CAN controller chips out of reset mode, a write to the CAN Control Register (PLD Register Space) is required.

After power-up or board reset the CAN transceivers are in the "Silent Mode".

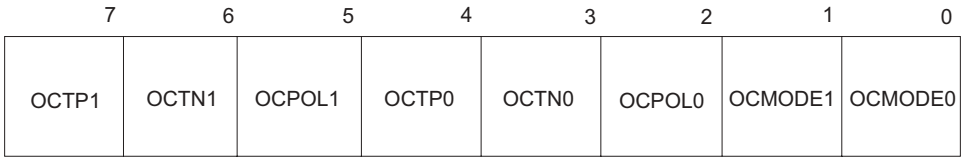
To bring the CAN transceivers to operating mode, a write to the CAN Control Register (PLD Register Space) is required.

The SJA1000 Output Control Register and Clock Divider Register must be programmed as follows (for both SJA1000 controllers in SJA1000 controller internal reset mode. See SJA1000 Control Register):

7	6	5	4	3	2	1	0
CAN MODE	CBP	RXINTEN		CLOCK OFF	CD2	CD1	CD0

CLOCK DIVIDER REGISTER		
Bit	**Symbol**	**Description**
7	CAN MODE	0 : BasiCAN Mode 1 : PeliCAN Mode
6	CBP	1 : Bypass input comparator, Use RX0 only
5	RXINTEN	0 : Disable Interrupts on TX1 output
4	-	0
3	CLOCK OFF	1 : Disable Clock Output (Not used)
2	CD2	0
1	CD1	0
0	CD0	0

Figure 5-1 : Clock Divider Register



OUTPUT CONTROL REGISTER		
Bit	**Symbol**	**Description**
7	OCTP1	11 : Push-Pull output stage
6	OCTN1	
5	OCPOL1	0 : Normal polarity
4	OCTP0	11 : Push-Pull output stage
3	OCTN0	
2	OCPOL0	0 : Normal polarity
1	OCMODE1	01 : Test Output Mode 10 : Normal Output Mode
0	OCMODE0	

Figure 5-2 : Output Control Register

6 Configuration Hints

6.1 I/O Line Configuration

The I/O line configuration is configured by on-board solder pads (there is no jumper solution because the TPMC806 is considered to be used in a vibration-sensitive environment.)

Possible line configuration options for each of the four I/O channels (MIC CH1, MIC CH2, CAN CH1, CAN CH2) are:

- On-board Termination-Mode : on / off
- Bus-Mode : bus_end / pass_through

The on board termination option for a MIC I/O channel node input (see P14 I/O pin assignment) is a 120 ohm termination resistor.

The on-board termination option for a CAN I/O channel node input (see P14 I/O pin assignment) is a 120 ohm split termination network.

For the bus_end option, the I/O lines are **not** passed through to the node output pins on the P14 I/O connector (see P14 I/O pin assignment).

For the pass_through option, the I/O lines are passed through from the node input pins to the node output pins of the P14 I/O connector (see P14 I/O pin assignment).

I/O LINE CONFIGURATION				
	Termination Mode		Bus Mode	
	On-board Termination On	On-board Termination Off	Bus_End	Pass_Through
MIC CH1	J4 Closed	J4 Open	J5, J6 Open	J5, J6 Closed
MIC CH2	J1 Closed	J1 Open	J2, J3 Open	J2, J3 Closed
CAN CH1	J13, J14 Closed	J13, J14 Open	J11, J12 Open	J11, J12 Closed
CAN CH2	J9, J10 Closed	J9, J10 Open	J7, J8 Open	J7, J8 Closed

Figure 6-1 : I/O Line Configuration

FACTORY DEFAULT I/O LINE CONFIGURATION		
	Termination Mode	**Bus Mode**
MIC CH1	On	Pass_Through
MIC CH2	On	Pass_Through
CAN CH1	Off	Pass_Through
CAN CH2	Off	Pass_Through

Figure 6-2 : Factory Default I/O Line Configuration

6.2 Solder Pad Location

TPMC806 PCB, Top View, Upper Right Corner

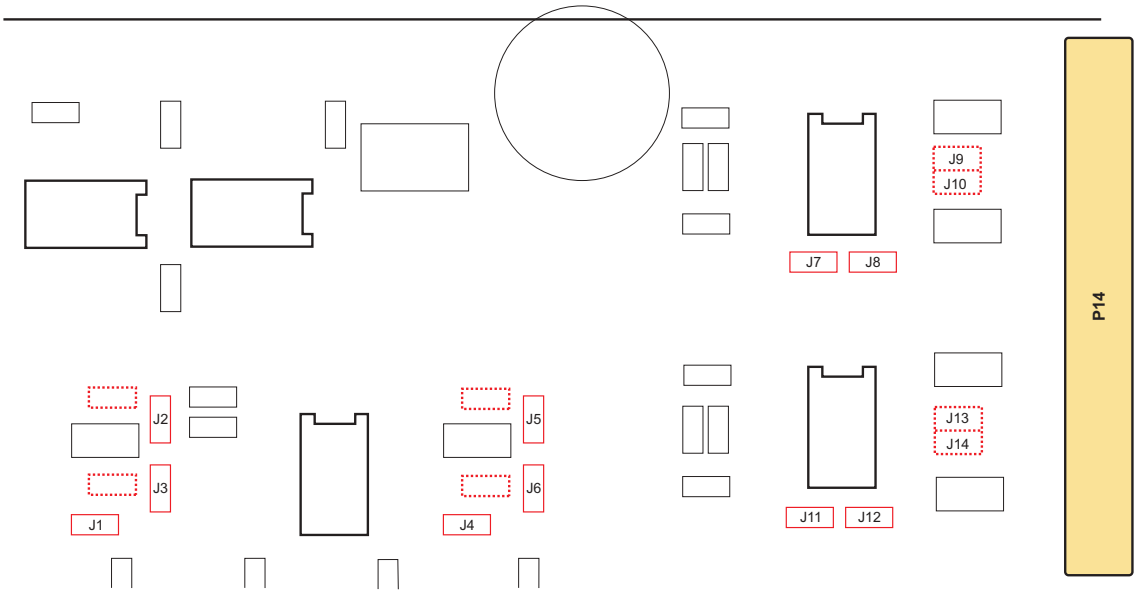


Figure 6-3 : Solder Pad Location

7 Pin Assignment

The complete TPMC806 I/O interface is available on the P14 64p. mezzanine connector ("back-I/O").

Pin	Signal	Interface	Direction
1	MIC_CH1_P	RS485-HD	Node In
2	MIC_CH1_N		
3	MIC_GND		
4	NC		
5	MIC_CH1_P	RS485-HD	Node Out
6	MIC_CH1_N		
7	MIC_GND		
8	NC		
9	MIC_CH2_P	RS485-HD	Node In
10	MIC_CH2_N		
11	MIC_GND		
12	NC		
13	MIC_CH2_P	RS485-HD	Node Out
14	MIC_CH2_N		
15	MIC_GND		
16	NC		
17	NC		
18	NC		
19	NC		
20	NC		
21	CAN_CH1_P	CAN-HS	Node In
22	CAN_CH1_N		
23	CAN1_GND		
24	NC		
25	CAN_CH1_P	CAN-HS	Node Out
26	CAN_CH1_N		
27	CAN1_GND		
28	NC		
29	NC		
30	NC		
31	NC		
32	NC		
33	CAN_CH2_P	CAN-HS	Node In
34	CAN_CH2_N		
35	CAN2_GND		
36	NC		
37	CAN_CH2_P	CAN-HS	Node Out
38	CAN_CH2_N		

Pin	Signal	Interface	Direction
39	CAN2_GND		
40	NC		
41 … 64	NC		

Figure 7-1 : P14 I/O Pin Assignment

Be sure that the P14 connector I/O signals used by the TPMC806 are available and not otherwise used on the J14 connector of the PMC carrier board.

The "Out-Node" for each MIC or CAN channel is only available if the on-board I/O line configuration is set accordingly (pass_through mode).