The Embedded I/O Company

TEWS TECHNOLOGIES

TPMC821

INTERBUS Master G4 PMC

Version 1.1

User Manual

Issue 1.1.7

August 2014

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

TPMC821-10R

INTERBUS Master G4 PMC module

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#.

Access terms are described as:

W	Write Only
R	Read Only
R/W	Read/Write
R/C	Read/Clear
R/S	Read/Set

Issue	Description	Date
1.0	First Issue	September 1999
1.1	Additions to “Technical Specification”	February 2001
1.2	General Revision (Manual)	March 2003
1.3	Board Version / Firmware Update (Phoenix Firmware 4.6x)	February 2004
1.4	Phoenix Firmware specified as 4.66	April 2004
1.5	Replaced PCI9050 with PCI9030	September 2006
1.1.6	New notation for User Manual and Engineering Documentation	January 2010
1.1.7	General Revision	August 2014

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 Product Description

The TPMC821 is a standard single-width 32 bit PMC module and offers a complete PMC INTERBUS Master Generation 4 (G4) Interface. A MC68332 local controller and the IPMS3 INTERBUS protocol controller are used as controlling units on board of the TPMC821.

The communication between the host CPU and the TPMC821 is handled via a 4 Kbyte Dual Port Memory. The on board firmware running on the MC68332 is the original INTERBUS Master Generation 4 (G4) firmware from Phoenix Contact.

Furthermore the TPMC821 provides a RS232 diagnostic port, the optically isolated INTERBUS interface and status LEDs.

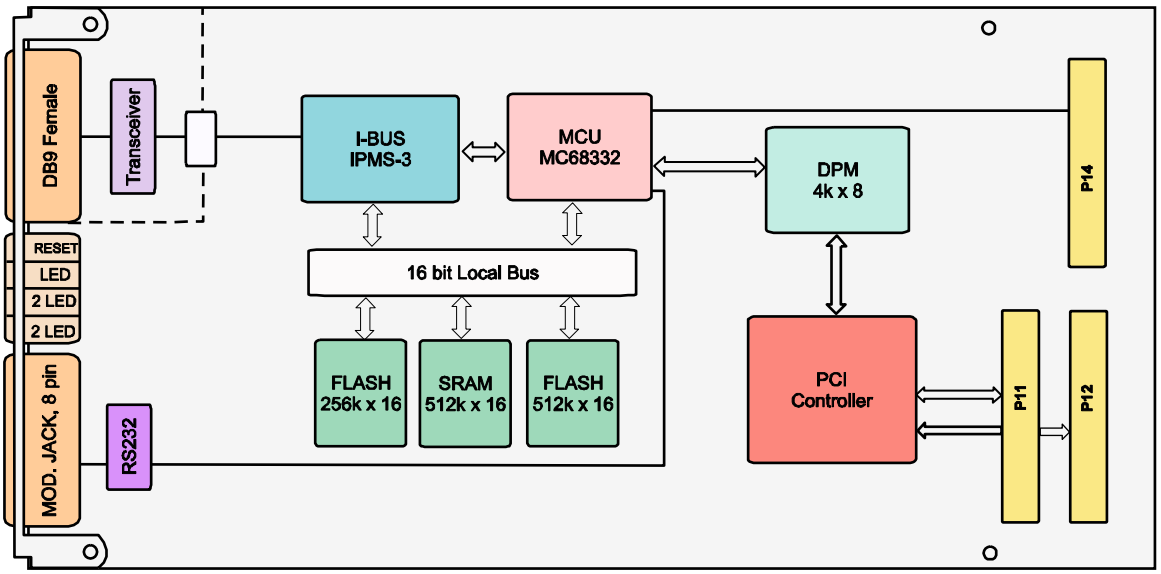

Figure 1-1 : Block Diagram

2 Technical Specification

PMC Interface	
Mechanical Interface	PCI Mezzanine Card (PMC) Interface Single Size
Electrical Interface	PCI Rev. 2.2 compliant 33 MHz / 32 bit PCI 3.3V and 5V PCI Signaling Voltage
On Board Devices	
PCI Target Chip	PCI9030 (PLX Technology)
Local CPU	MC68332-20 MHz
INTERBUS Controller	IPMS-3
Dual Port Memory	4 Kbyte (4k x 8)
SRAM	1 Mbyte (512k x 16)
FLASH	1 Mbyte (512k x 16) + 512Kbyte (256k x 16)
INTERBUS Master	Generation 4 (G4)
Firmware (Factory default)	4.66 Phoenix Contact (TPMC821 V1.1) 4.40 Phoenix Contact (TPMC821 V1.0)
Operating Modes	Asynchronous, bus synchronous, program synchronous, Asynchronous with synchronization pulse
Addressing Modes	Physical and logical
I/O Maximum	4096
Slave Devices Maximum	512
PCP Devices Maximum	62
Non-volatile Parameterization	Yes
PDP Memory	384 Kbyte
I/O Interface	
INTERBUS I/O	RS485, optically isolated, DB9 female connector (front panel)
Serial Diagnostic Interface	RS232, 8 pin Mod. JACK (front panel)
Diagnostic LEDs	RDY/RUN, FAIL, BSA, PF, HF (front panel)
Physical Data	
Power Requirements	1.25A max.

Temperature Range	Operating Storage	0 °C to +70 °C -45°C to +125°C
MTBF	260000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G_B 20°C. The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation.	
Humidity	5 – 95 % non-condensing	
Weight	80 g	

Table 2-1 : Technical Specification

3 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	PCI9030 PCI Base Address (Offset in PCI Configuration Space)	PCI Space Mapping	Size (Byte)	Port Width (Bit)	Endian Mode	Description
0	0 (0x10)	MEM	128	32	LITTLE	PCI9030 Local Configuration Registers
1	1(0x14)	I/O	128	32	LITTLE	PCI9030 Local Configuration Registers
2	2 (0x18)	MEM	4k	8	BIG (lower lane)	Local Memory Space
3	3 (0x1C)	MEM	16	8	BIG (lower lane)	Local Register Space

Table 3-1 : PCI9030 Local Space Configuration

3.2 Local Memory Space

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

3.2.1 Coupling Memory

The coupling memory is the communication interface between the host and the TPMC821. The coupling memory is implemented as a 4 Kbyte Dual Port Memory (DPM) on the TPMC821.

3.2.2 DPM Areas

The DPM is divided into several functional areas.

Offset	Area
0x000	DTA OUT
0x200	DTA IN
0x400	SSGI BOX 0 (Host to TPMC821)
0x800	SSGI BOX 1 (TPMC821 to Host)
0xC00	SLAVE I/O
0xC80	EXT. DATA
0xFC0	REGISTER

Table 3-2 : DPM Area Map

The DTA area is used to store process data. The TPM821 writes INTERBUS input data to the DTA IN region. The host writes INTERBUS output data to the DTA OUT region.

The SSGI area is used for exchanging messages between the TPMC821 and the host. With these mailboxes the host is able to perform firmware service requests of the TPMC821 and to receive service confirmation from the TPMC821.

The SSGI BOX 0 is the mailbox for the host to TPMC821 direction. The SSGI BOX 1 is the mailbox for the TPMC821 to host direction.

The functions of the SLAVE I/O area are not supported by the TPMC821. The EXT DTA area can be used for user defined functions.

The REGISTER area is used to handle the communication protocol between the TPMC821 and the host.

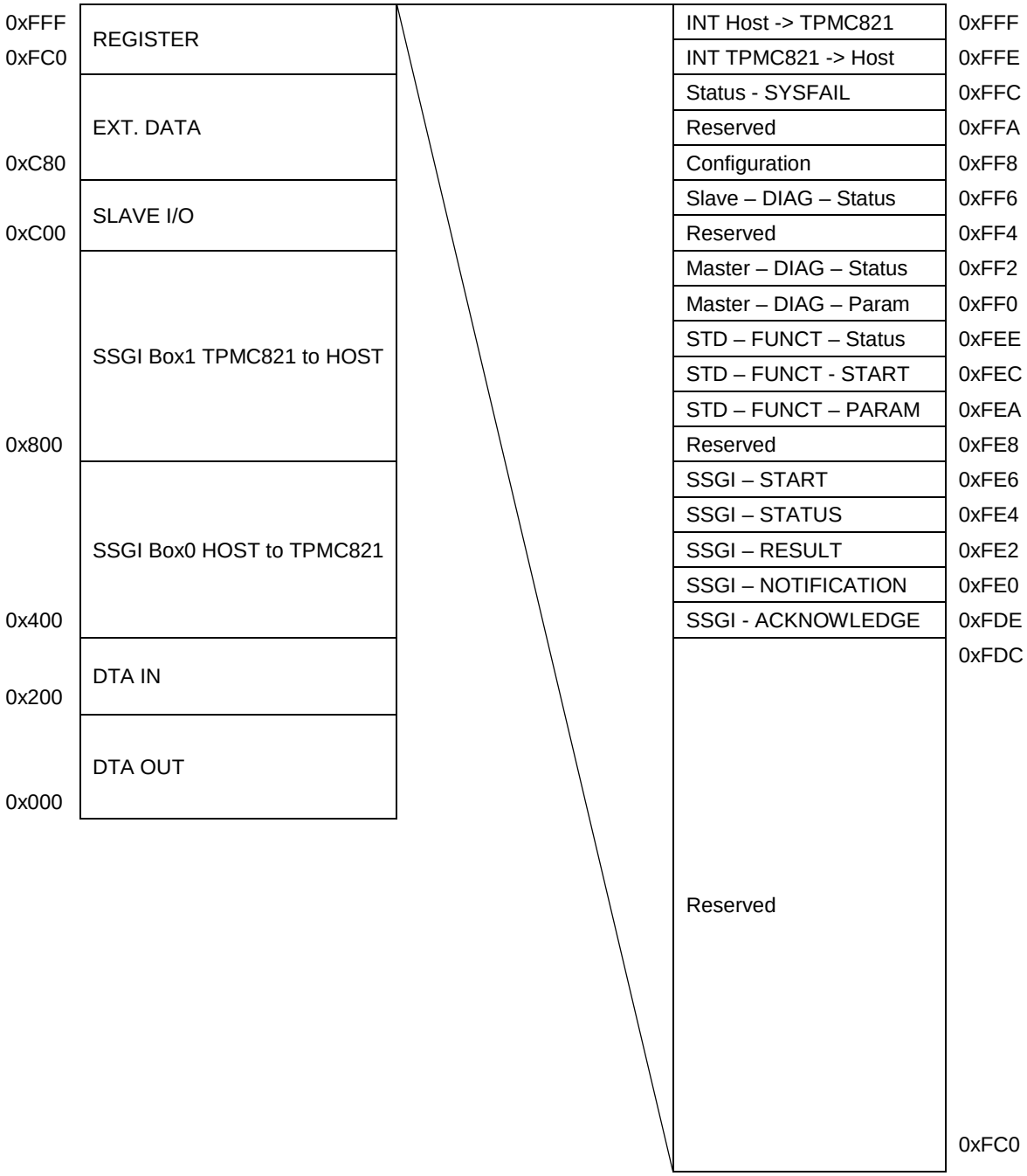

Table 3-3 : DPM Map

3.2.3 DPM Register Area Map

The registers in the DPM register area are used for

- controlling interrupt request communication for the synchronous operating modes
- indicating status data
- controlling standard functions
- controlling SSGI handshake communication protocol (service requests, service confirmation)

Offset	Name
0xFFF	Interrupt Register Host -> Master (INT_H_MA)
0xFFE	Interrupt Register Master -> Host (INT_MA_H)
0xFFC	Status SYSFAIL Register
0xFF8	Configuration Register
0xFF6	Slave Diagnostic Status Register
0xFF2	Master Diagnostic Status Register
0xFF0	Master Diagnostic Parameter Register
0xFEE	Standard Function Status Register
0xFEC	Standard Function Start Register
0xFEA	Standard Function Parameter Register
0xFE6	SSGI Start Register
0xFE4	SSGI Status Register
0xFE2	SSGI Result Register
0xFE0	SSGI Notification Register
0xFDE	SSGI Acknowledge Register

Table 3-4 : DPM Register Map

3.3 Local Register Space

The following registers refer to the host side.

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

Offset to PCI Base Address 3	Register Name		Size (Bit)
0x00	INT_CS	Interrupt Control and Status Register	8
0x01	INT_E	Interrupt Enable Register	8

Table 3-5 : Local Register Space

3.3.1 Interrupt Control/Status Register INT_CS (Offset 0x00)

The Interrupt Control / Status Register is an 8 bit wide register with various access types.

Bit	Symbol	Description	Access	Reset Value
7:3		Not used and undefined during reads		
2	S_SVCIRQ	Status Service Interrupt Request 1 = Service Interrupt Request Pending 0 = No Service Interrupt Request Pending If the local CPU detects a hardware error and the corresponding bit is set in the INT_E Register this service request interrupt is generated on INTA. The service interrupt request is cleared by writing '1'.	R/C	
1	S_DPMIRQ	Status DPM Interrupt Request 1 = DPM Interrupt Request Pending 0 = No DPM Interrupt Request Pending If the local CPU writes to DPM location 0xFFE (Indication Register from TPMC821 to host) and the corresponding interrupt enable bit is set in the INT_E Register this interrupt request is generated on INTA. The DPM interrupt request is cleared by reading the DPM location 0xFFE.	R	
0	C_HOSTIRQ	Control Host Interrupt Request 1 = Set Host Interrupt Request 0 = Clear Host Interrupt Request Interrupt should be issued in case of a serious host system failure. All INTERBUS outputs are reset. An active host interrupt is indicated by the HF LED. **The effect of a host interrupt to the INTERBUS depends on the configuration of the firmware system variables.**	R/W	0

Table 3-6 : Interrupt Control / Status Register INT_CS (Offset 0x00)

3.3.2 Interrupt Enable Register INT_E (Offset 0x01)

The Interrupt Enable Register is an 8 bit wide read/write register that is used to enable/disable interrupt request from the TPMC821 to the host.

Bit	Symbol	Description	Access	Reset Value
7:3		Not used and undefined during reads		
2	E_SVCIRQ	Enable Service Interrupt Request from the TPMC821 1 = Enable Service Interrupt Request 0 = Disable Service Interrupt Request	R/W	0
1	E_DPMIRQ	Enable Dual Port Memory Interrupt Request from TPMC821 1 = Enable DPM Interrupt Request 0 = Disable DPM Interrupt Request	R/W	0
0		Reserved. Write as '0'.		0

Table 3-7 : Interrupt Enable Register INT_E (Offset 0x01)

4 PCI9030 Target Chip

4.1 PCI Configuration Registers (CFG)

4.1.1 PCI9030 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	0335 1498
0x04	Status		Command		Y	0280 0000
0x08	Class Code			Revision ID	N	028000 xx
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	FFFFFF80
0x14	PCI Base Address 1 for I/O Mapped Config. Registers				Y	FFFFFF81
0x18	PCI Base Address 2 for Local Address Space 0				Y	FFFFF000
0x1C	PCI Base Address 3 for Local Address Space 1				Y	FFFFFFF0
0x20	PCI Base Address 4 for Local Address Space 2				Y	00000000
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	000A 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 01 00
0x40	PM Cap.		PM Nxt Cap.	PM Cap. ID	N	4801 00 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 00 00
0x4C	VPD Address		VPD Nxt Cap.	VPD Cap. ID	Y[31:16]	0000 00 03
0x50	VPD Data				Y	00000000

Table 4-1 : PCI9030 Header

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 PCI9030 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'.

Offset in Config.	Description	Usage
0x10	PCI9030 LCR's MEM	Used
0x14	PCI9030 LCR's I/O	Used
0x18	PCI9030 Local Space 0	Used
0x1C	PCI9030 Local Space 1	Used
0x20	PCI9030 Local Space 2	Not used
0x24	PCI9030 Local Space 3	Not used
0x30	Expansion ROM	Not used

Table 4-2 : PCI9030 PCI Base Address Usage

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.

Offset from PCI Base Address	Register	Value	Description
0x00	Local Address Space 0 Range	0x0FFF_F000	4 kBytes Memory Space
0x04	Local Address Space 1 Range	0x0FFF_FFF0	16 Bytes Memory Space
0x08	Local Address Space 2 Range	0x0000_0000	Not used
0x0C	Local Address Space 3 Range	0x0000_0000	Not used
0x10	Local Exp. ROM Range	0x0000_0000	Not used
0x14	Local Re-map Register Space 0	0x0000_0001	Enabled, Base Address 0x0000
0x18	Local Re-map Register Space 1	0x0001_0001	Enabled, Base Address 0x1000
0x1C	Local Re-map Register Space 2	0x0000_0000	Not used
0x20	Local Re-map Register Space 3	0x0000_0000	Not used
0x24	Local Re-map Register ROM	0x0000_0000	Not used
0x28	Local Address Space 0 Descriptor	0x5501_40C2	Local Space 0 Configuration
0x2C	Local Address Space 1 Descriptor	0x5501_40C0	Local Space 1 Configuration
0x30	Local Address Space 2 Descriptor	0x0000_0000	Not used
0x34	Local Address Space 3 Descriptor	0x0000_0000	Not used
0x38	Local Exp. ROM Descriptor	0x0000_0000	Not used
0x3C	Chip Select 0 Base Address	0x0000_0401	Chip Select Local 0
0x40	Chip Select 1 Base Address	0x0000_0C01	Chip Select Local 1
0x44	Chip Select 2 Base Address	0x0001_0009	Chip Select Local 2
0x48	Chip Select 3 Base Address	0x0000_0000	Not used
0x4C	Interrupt Control/Status	0x0049	Local IRQ1 & IRQ2 and PCI IRQ enabled
0x4E	EEPROM Write Protect Boundary	0x0030	Standard write protection
0x50	Miscellaneous Control Register	0x0078_0000	Retry delay = max
0x54	General Purpose I/O Control	0x026D_B6DB	No GPIO
0x70	Hidden1 Power Management data select	0x0000_0000	Not used
0x74	Hidden 2 Power Management data scale	0x0000_0000	Not used

Table 4-3 : PCI9030 Local Configuration Register

4.3 Configuration EEPROM

After power-on or PCI reset, the PCI9030 loads initial configuration register data from the on board configuration EEPROM.

The configuration EEPROM contains the following configuration data:

- Address 0x00 to 0x0F : PCI9030 PCI Configuration Register Values
- Address 0x10 to 0x87 : PCI9030 Local Configuration Register Values
- Address 0x88 to 0xFF : Reserved

See the PCI9030 Manual for more information.

Address	Offset							
	0x00	0x02	0x04	0x06	0x08	0x0A	0x0C	0x0E
0x00	0x0335	0x1498	0x0280	0x0000	0x0280	0x0000	0x000A	0x1498
0x10	0x0000	0x0040	0x0000	0x0100	0x4801	0x4801	0x0000	0x0000
0x20	0x0000	0x4C06	0x0000	0x0003	0x0FFF	0xF000	0x0FFF	0xFFF0
0x30	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
0x40	0x0001	0x0001	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
0x50	0x5501	0x40C2	0x5501	0x40C0	0x0000	0x0000	0x0000	0x0000
0x60	0x0000	0x0000	0x0000	0x0401	0x0000	0x0C01	0x0001	0x0009
0x70	0x0000	0x0000	0x0030	0x0049	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

Table 4-4 : Configuration EEPROM TPMC821-10R

4.4 Software Reset (Controller and LRESET#)

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 PCI9030 and issues a reset to the Local Bus (LRESETo# asserted). The PCI9030 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.

The PCI-9030 LRESET# pin is connected to the TPMC821 on board logic's master reset.

4.5 PCI Interrupt Control/Status

For disabling / enabling PCI interrupts set bit 6 of the PCI9030 Interrupt Control/Status Register (INTCSR, 0x4C) to '0' / '1'. Do not change any other bits of this register.

4.6 Local Space Endian Mode

The local space endian mode is controlled by the local space descriptor register bit 24.

If bit 24 of a local space descriptor register is '0' then little endian mode is selected. A '1' selects big endian mode.

The TPMC821 uses local spaces 0 and 1. Local space 0 is the Local Memory Space, local space 1 is the Local Register Space. Both local spaces are set to big endian mode per default.

The offset of the Local Space 0 Descriptor Register is 0x28, the offset of the Local Space 1 Descriptor Register is 0x2C, in the PCI9030 Local Configuration Register Space. The PCI9030 Local Configuration Register Space is always in little endian mode.

5 Communication between Host and TPMC821

There are two interfaces for INTERBUS communication data exchange between the host and the TPMC821. These interfaces are realized in the DPM:

- **Standard Signal Interface (SSGI)** for the exchange of messages
- **Data Interface (DTA)** for processing data exchange

Each interface has its own memory area in the DPM which is divided into a send and receive area.

The input data provided by the INTERBUS is stored in the “DTA IN” area by the TPMC821. The host writes the output data to the “DTA OUT” area.

Box0 of the SSGI is used to transmit messages from the host to the TPMC821 (service requests). Box1 transmit messages from the TPMC821 to the host (service confirmations).

5.1 Message Exchange via the SSGI

Exchanging messages via SSGI is controlled by SSGI handshake protocol which is controlled by the SSGI registers in the DPM Register area. The message data is exchanged via the DPM SSGI area.

5.1.1 Sending Messages to the TPMC821

Message sending (service request) is controlled by the SSGI Start Register, SSGI Status Register and SSGI Result Register. The host writes the message data to SSGI Box0.

5.1.2 Receiving Messages from the TPMC821

Message receiving (service information) is controlled by the SSGI Notification Register and SSGI Acknowledge Register. The host can read the message data from the SSGI Box1.

5.2 Exchanging Process Data via the Data Interface

5.2.1 Operating Modes

The data interface (DTA) on INTERBUS offers three operating modes which are characterized by their different types of process data transmission to the host system.

After reset the TPMC821 operates in the “asynchronous without consistency locking” operating mode.

5.2.1.1 Asynchronous

- Process data is updated by the TPMC821 synchronously with the INTERBUS data cycles, but asynchronously with the host’s random access to the process image.
- An optional signal protocol can be used to lock accesses to the DPM and thus enable the host to access the complete consistence process image.
- The INTERBUS data cycles executed by the TPMC821 are equidistant in terms of time and can be set with a resolution of 1µs.

5.2.1.2 Asynchronous with Synchronization Pulse

- Interrupt based (handshake) operating mode
- Host sets up DPM data and generates interrupt to the INTERBUS master
- INTERBUS master reads DPM data, performs INTERBUS cycle, updates DPM data and generates interrupt to the host.

5.2.1.3 Bus Synchronous

- Process data is updated by the TPMC821 synchronously with the INTERBUS data cycles.
- At the beginning of each INTERBUS data cycle the TPMC821 sends a synchronization pulse to the host. This can be used by the host to synchronize the application process with the transmission of process data to the I/O devices.
- The INTERBUS data cycles executed by the TPMC821 and the synchronization pulse are very precisely equidistant in terms of time and can be set with a resolution of 1µs.

5.2.1.4 Program Synchronous

- The host application process initializes each individual INTERBUS data cycle with a trigger pulse.
- The process data in the DPM is transmitted and updated by the TPMC821 synchronously with the host trigger pulses.

“Bus Synchronous” and “Program Synchronous” operating modes are still part of the Phoenix Contact firmware code, but are no longer listed in the Phoenix Contact IBS USC4-2 UM document.

6 Firmware

6.1 Startup Behavior

After a successful power-on self test (POST) the TPMC821 is set to the READY state (RDY).

The “Complete_Load_Configuration” service allows loading a planned bus configuration to the TPMC821. When activating the loaded bus configuration with the “Active_Configuration” service the TPMC821 checks if this configuration corresponds to the actual configuration. Differences between the planned and present bus configuration will be indicated by the service confirmation.

The “Create_Configuration” service allows reading in the connected bus configuration. When this is done, the configuration can be read out with the “Read_Configuration” service and can be compared with the planned configuration.

With the CMD Software from Phoenix Contact it is also possible to store preprocessed functions as a boot project to the TPMC821. After power-on self test this program is copied to the on board RAM and can be started automatically.

6.2 State Machine

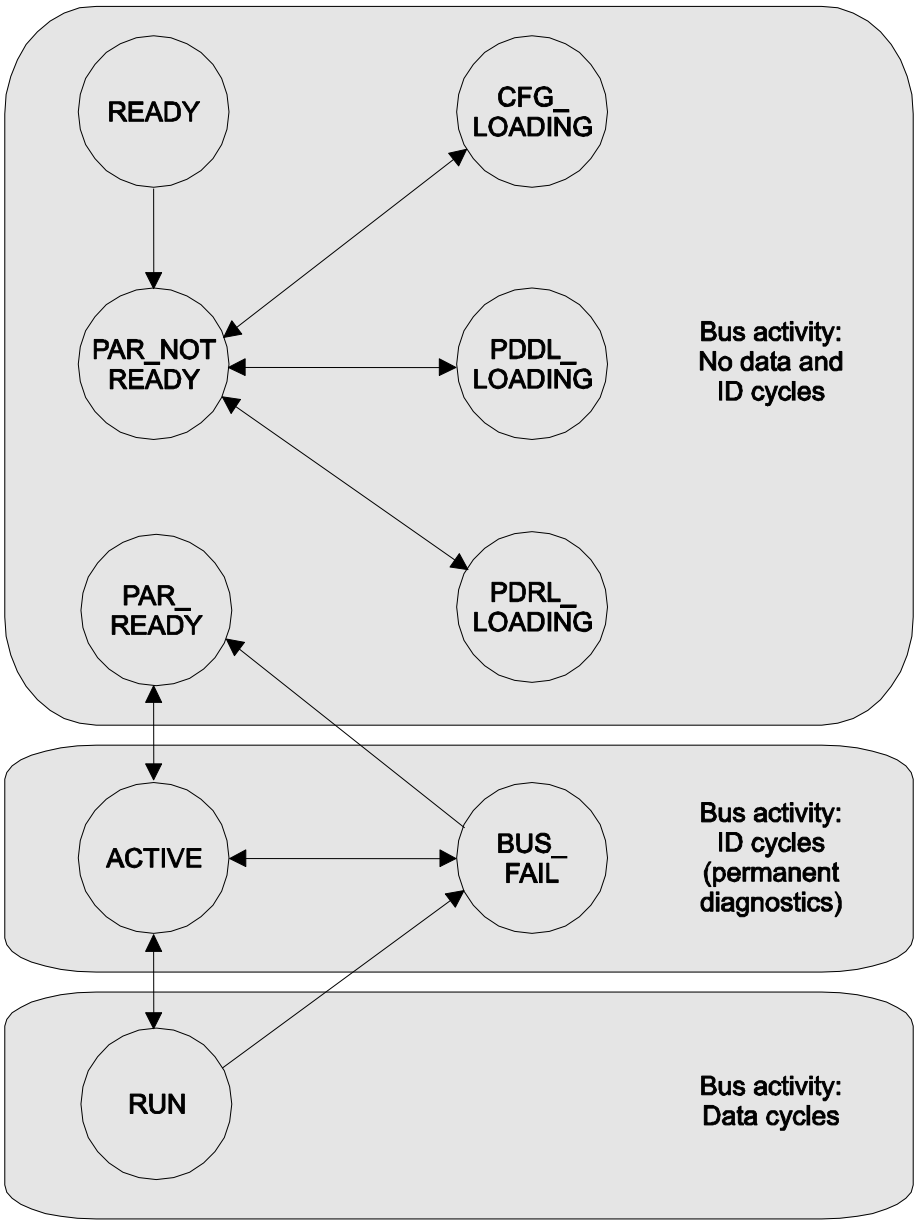

Figure 6-1 : State Machine

6.2.1 Explanation for State Machine

READY

The READY state is the initial state after every startup. In this state is no valid bus configuration available. The bus can not be activated.

PAR_NOT_READY

After initiating the parameterization phase the system management is in the PAR_NOT_READY state. It is possible to initiate the three parameterization sequences for loading the

- configuration frames
- process data description lists (PDDLs)
- process data reference lists (PDRLs)

It is not ensured that the parameterization data which can be read out during the parameterization phase is valid.

CFG_LOADING

The system must be in the CFG_LOADING state to load a configuration frame.

PDDL_LOADING

The system must be in de CFG_LOADING state to define process data.

PDRL_LOADING

The system must be in de CFG_LOADING state to link process data.

PAR_READY

After having terminated the parameterization phase successfully the firmware occupies the PAR_READY state. One configuration frame was defined at least. It was checked for completeness and consistency.

The states PAR_NOT_READY, PAR_READY, CFG_LOADING, PDDL_LOADING and PDRL_LOADING imply that the bus is inactive and the data and ID cycles cannot be run.

ACTIVE

The bus is active. Only ID cycles are running. Data are not transmitted via the bus. The bus is operated with the configuration characterized as “active configuration” in the configuration frame.

RUN

Data cycles are running in this state. Data is actually transferred in this state.

BUS_FAIL

If errors occur in the ACTIVE or RUN state the system management will enter the BUS_FAIL state. ID cycles are still running with the greatest possible configuration.

The states ACTIVE, RUN and BUS_FAIL can directly be read from the Master Diagnostic Status Register.

6.3 Services

Firmware services are requested via the SSGI Box0 area in the DPM. The SSGI registers in the register area of the DPM are used for the communication handshake protocol during service request.

To request a service the host starts the SSGI message sending (Service request) handshake protocol and writes the service code, the parameter count value and the service parameters to the SSGI Box0 area in the DPM.

Then the host can check the service confirmation message with the SSGI message receiving protocol by using the SSGI registers in the register area of the DPM. The confirmation data is stored in the SSGI Box1 area in the DPM by the TPMC821.

Every request will be confirmed (exception: reset and unconfirmed PCP services).

When representing the request code in binary form, bit 15 has always assigned the bit value ‘0’. The confirmation code following to this request is identical, except for bit 15. Bit 15 of the confirmation code has always assigned bit value ‘1’. Thus, in hexadecimal representation, the confirmation code is always 0x8000 higher than the corresponding request code.

All services include a parameter count (Parameter_Count) which indicates how many parameters will follow.

Positive and negative confirmations have the same confirmation code. The result parameter indicates if a service was executed successfully.

7 Diagnostic LEDs

There are 5 Diagnostic LEDs on the TPMC821 to support a simple diagnosis of the current mode of operation.

7.1 LED Description

LED	Color	Description	Display
RUN (RDY / RUN)	Green	Ready	ON OFF
		Run	ON OFF
FA (FAIL)	Red	No Error	ON OFF
		Remote Bus Error	ON OFF
		Local Bus Error	
		Controller Error	
		Watchdog or HW Error	
BS (BSA)	Yellow	Bus segment Switch Off	ON OFF
PF	Yellow	Module Error	ON OFF
HF	Yellow	Host Error	ON OFF

Figure 7-1 : Diagnostic LEDs

7.2 LED Location

The Diagnostic LEDs are located on the TPMC821 front panel.

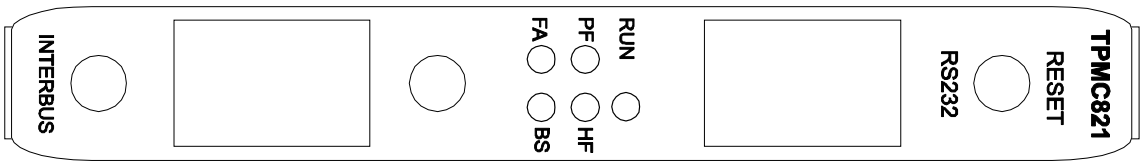

Figure 7-2 : LED Location

8 Pin Assignment – I/O Connector

8.1 INTERBUS Interface DB9 female Connector

Pin	Signal	Level
1	DOH	Data OUT High, RS485
2	DIH	Data IN High, RS485
3	GNDi	Ground for isolated VCCi
4	GND	
5	VCCi	5V +/-5%, 60mA, isolated
6	DOL	Data OUT Low, RS485
7	DIL	Data IN Low, RS485
8	VCC	5V, +/-5%, 30mA, fused 1A
9	NC	

Table 8-1 : DB9 female connector front panel

8.2 RS232 Interface 8 pin Mod. JACK Connector

Pin	Signal	Level
1	NC	
2	GND	
3	CTS	Clear To Send, RS232
4	RTS	Request To Send, RS232
5	RXD	Receive Data, RS232
6	TXD	Transmit Data, RS232
7	NC	
8	NC	

Table 8-2 : RS232 Interface 8 pin Mod. JACK connector

8.3 P14 Mezzanine Connector

Pin	Signal
BDM Port	
1	+5V (F)
2	BD32_CONN
3	+5V (F)
4	IPIPE
5	MC_RES
6	IFETCH
7	GND
8	FREEZE
9	GND
10	BKPT
11	NC
12	NC
Diagnostic Port	
13	+5V (F)
14	PCS3
15	+5V (F)
16	PCS2
17	SW_RES
18	MISO
19	GND
20	MOSI
21	GND
22	SCK*
23…64	NC

Table 8-3 : P14 Mezzanine Connector

* Requires population of R27

8.4 BDM Connector (10 pin field on board)

Pin	Signal
1	GND
2	BKPT
3	GND
4	FREEZE
5	MC_RES
6	IFETCH
7	+5V (F)
8	IPIPE
9	+5V (F)
10	BD32_CONN

Table 8-4 : BDM Connector (10 pin field on board)

9 Important Notes

The hardware of the TPMC821 INTERBUS Master Interface Generation 4 (G4) from TEWS TECHNOLOGIES is based on the original Generation (G4) Master Board USC4-2 from Phoenix Contact.

The original firmware version 4.66 from Phoenix Contact is running on the TPMC821 V1.1 board version.

The original firmware version 4.40 from Phoenix Contact is running on the TPMC821 V1.0 board version.