PROPRIETARY NOTICE

CAN-PMC/FD

PMC CAN FD Interface

(Hardware Manual)

Version 0.9

refers to product revision no.
V 1.0

Title:	CAN-PMC/FD User's Manual
File:	j:\as\entwicklung\projekte\hw\misc\can-pmc_zync\docs\manual\can-pmcfd_hardware_manual.doc
Pattern:	
Created:	ime, 22.10.2008
Last Update:	ime, 07.03.2013

Im Doerener Feld 8
D-33100 Paderborn

Tel.:	+49-5251-1550-0
Fax:	+49-5251-1550-190
Email:	support@janztec.com
Internet:	www.janztec.com

Contents

List of Figures

List of Tables

About this Manual

This is the Hardware Manual for the CAN-PMC/FD CAN fieldbus controller board for PCI Express. It gives all necessary information to users and programmers of the CAN-PMC/FD. We try to keep it compact, so there are no special section for users and programmers. Users that are no programmers might therefore only read the first sections (Introduction and Installation), and come back to the detailed sections when they need a special information.
In any case, this manual assumes that the users, especially programmers, are familiar with their job. We do not give basic information about programming, computer hardware or describe operation of bus-systems. You will find pointers to advance information in section 4.2.
The manual starts with some **Introduction** to the CAN-PMC/FD. This is some prose that describes the features and architecture of the CAN-PMC/FD. You should already know this if you have bought the board! It then discusses some topics about **Installation** and **Configuration** of the product.

The rest of the manual will then cover **technical details** about the CAN-PMC/FD.

Conventions

If numbers are specified in this manual, they will be either decimal or hexadecimal. We use C-notation to identify hexadecimal numbers (the 0x prefix).

If we refer to low active signal names, they will suffixed by a "#" character.

Some parts of the manual contains notices you have to observe to ensure your personal safety, or to prevent damage to property. These are visually marked with the following alert symbols:

DANGER

indicates that death or severe personal injury *will* result if proper precautions are not taken.

WARNING

indicates that death or severe personal injury *may* result if proper precautions are not taken.

CAUTION

indicates that *minor* personal injury can result if proper precautions are not taken.

NOTICE

indicates that damage to equipment can result if proper precautions are not taken.

indicates information that we think you should have read to save your time by avoiding common problems. Important suggestions that should be followed will also be marked with this sign.

Acronyms and Abbreviations

EMC	Electromagnetic capability.
ESD	Electrostatic discharge.
RO	Read Only.
RW	Read and Write.
WO	Write Only.
PCI	Peripheral Component Interconnect

1 Introduction

CAN-PMC/FD

The CAN-PMC/FD is a PCI Mezzanine Card (PMC) compatible intelligent CAN FD field bus controller module. It can be plugged onto a mezzanine carrier board for specific host computer systems. In this way the CAN-PMC/FD can be used for CAN host connections to VMEbus, CompactPCI etc.

Material-No.	Product	Description
BO-FPC-36242	CAN-PMC/FD	4 channel, isolated

table 1: CAN-PMC/FD standard products

1.1 Hardware Features

- Intelligent PMC CAN FD fieldbus controller
- Zynq ARM Soc up to to 667 MHz
- 512Mbyte DDR3 16bit organized
- 4 x CAN FD via IP Core inside Zynq
- 1Mbit high speed transfer rates
- ISO/DIS 11898-2 interface
- 25 pin DSUB connectors at front bezel
- CAN interfaces isolated from logic
- CANbus termination software switchable
- Optional RS232 serial interface
- PMC Specification P1386
- Single PMC module (stacking height 10mm)
- Dimensions 149mm x 74mm

1.2 Functional Overview

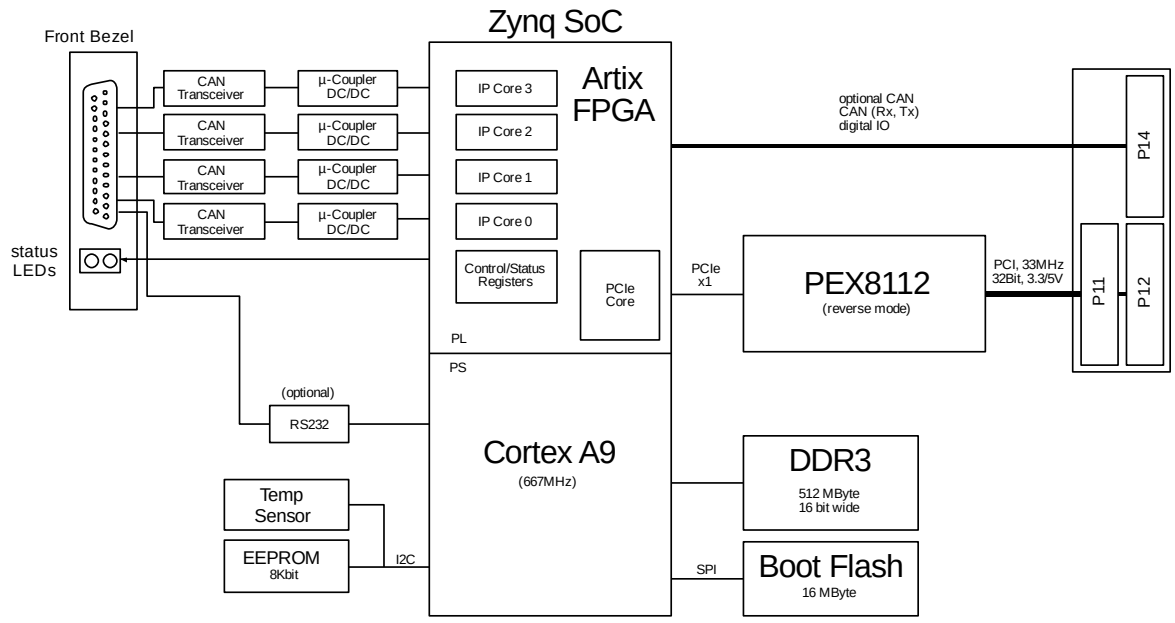

figure 1: CAN-PMC/FD block diagram

The CAN-PMC/FD offers a powerful CAN connection for host systems with PMC sockets. The CAN-PMC/FD supplies 4 CAN connections which are controlled by the on board Cortex A9 processor.

The block diagram of the CAN-PMC/FD is shown in figure 1. It shows the heart of the module with the Cortex A9 Zynq SoC, the Bootflash and with the DDR3. A PCI express endpoint – integrated in the Zynq FPGA section - allows the user to access the DDR3 and the control/status registers from the PCI as well as from the local processor. The address window accessible from the PCI is programmable. Connectivity to PMC is implemented with a PCI express bridge.

The CAN logic is implemented by IP cores inside the FPGA. All CAN interfaces are connectable via 25pin D-Sub connectors at the front bezel. The CAN interfaces on the CAN-PMC/FD are isolated from system logic and among each other. Optionally it is possible to make the CAN interfaces available at the PMC connector P14.

Additionally two user programmable general purpose LED´s are available on the front panel. Furthermore an optional RS232 serial interface can be used for debugging purposes.

With the Cortex A9, 16 MByte Flash and 512 MByte DDR3, the module is able to manage large-scale applications.

The CAN-PMC/FD comes default with the ICANOS firmware. ICANOS manages all base functions for a CAN network, such as the management of the timers and the CAN interfaces.
All interrupt handling for a real-time clock and the CAN controller is also carried out by ICANOS. ICANOS multiplexes time-out indications, receives CAN messages and bus events and transmits confirmations to it's user. Software filtering of received CAN messages may be requested. It is possible to request that single identifiers or ranges of identifiers be accepted or not.

The CAN-PMC/FD offers enough power to run on-board CANopen. The interfacing is done by CANopen service request/indications and by a DPM hosted real-time-data process image.
The services can be used in a very convenient way by an OS independent library, so that migration between different operating systems can be done easily.

2 Safety Instructions

Refer to page 5 for explanation of the warning notice system.

The product described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products.

2.1 Installation and Maintenance

CAUTION

This product is designed to be integrated into (industrial) computer systems.

DANGER

This product may only be connected to power supply systems that are free of hazardous voltages (e.g. SELV).

WARNING

The IO interfaces (connectors) of the product are only suited to be connected to SELV circuits.

2.2 Ambient and Environmental Conditions

WARNING

This product does not provide a fire enclosure according to EN 60950-1. Installation is only permitted into computer systems that provide such enclosure.

CAUTION

Do not operate the product beyond the specified ambient conditions.

DANGER

Do not operate this product in potentially explosive atmosphere.

3 Installation

When opening the shipping package of the CAN-PMC/FD, you should immediately check the contents of the package. In the package you will find information about the scope of delivery, as this depends on the options that you have ordered.

3.1 Handling Instructions

When installing the CAN-PMC/FD onto a mezzanine carrier board you need to obey some handling precautions to avoid damaging your modul with an electrostatic discharge.

When taking the modul from the anti-static bag, in which it is delivered, you should wear a grounded anti-static wrist strap. The strap should also be connected to your working environment. It is recommended to first touch the modul at the front panel, as this has a defined resistive path of about 1MΩ to the sensitive modul electronics, thereby a controlled discharge will take place.

Before placing the modul onto your mezzanine carrier board, you should check that the power supply has enough strength to provide the extra load for the CAN-PMC/FD. You should also check whether the supplied voltage levels are appropriate. See section **Fehler! Verweisquelle konnte nicht gefunden werden.** for the specification. Before inserting the CAN-PMC/FD modul onto your mezzanine carrier board, you *must* turn the power off.

When you insert your mezzanine carrier board with the CAN-PMC/FD into your system a second discharge defect could arise. This can be avoided by the ESD clips in the card guides of your mezzanine carrier board. These clips make contact to the moduls ESD strips, and discharge any electrostatic on the modul through a defined resistive path (2MΩ) *before* the carrier board connectors mate with the backplane.

If you have a system that does not have such ESD clips in the card guides, then *you* need to ensure that this discharge takes place! Do so by „connecting“ the CAN-PMC/FD front panel with the ground of your VMEbus or PCIbus system. This can be ensured if you touch the front panel, while wearing an anti-static wrist strap that is connected the ground of the VMEbus or PCIbus system.

3.2 Components and Jumper Locations

figure 2: Component side of CAN-PMC/FD

There are no jumpers or switches on the CAN-PMC/FD.

3.3 Connecting IO signals

This is the connector assignment for CAN0 to CAN3:

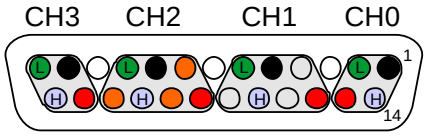

figure 3: D-SUB Connector Pin Layout

DSUB	Flat cable
1	EXGND_CH0
2	CAN-L_CH0
3	NC
4	NC
5	EXGND_CH1
6	CAN-L_CH1
7	NC
8	RxD[2)]
9	EXGND_CH2
10	CAN-L_CH2
11	NC
12	EXGND_CH3
13	CAN-L_CH3
14	CAN-H_CH0
15	EXVCC_CH0[1)]
16	EXVCC_CH1[1)]
17	NC
18	CAN-H_CH1
19	NC
20	EXVCC_CH2[1)]
21	TxD[2)]
22	CAN-H_CH2
23	GND[2)]
24	EXVCC_CH3[1)]
25	CAN-H_CH4

Notes	
1	Optional the EXVCC voltage can connected to the D-Sub connector by placing option resistors. This can be used to power external transceiver.
2	Optional connection of serial interface for debugging purposes

Optionally, the board can also be modified to route the isolated CAN interface signals to the PMC P14 IO connector. However, this breaks the isolation voltage. Hence this is an on-request option and not installed normally.

3.4 Software Considerations

For any information about installing drivers and other software packages related to the CAN-PMC/FD, refer to the documentation that comes with the driver or software.

4 Programming Information

This describes the methods that are needed, when working with the CAN-PMC/FD. It definitely relies on the Zynq configuration as it is factory supplied. If you manage to clear or alter the boot SPI flash, then this chapter will no longer apply to you.

4.1 PCI Bus configuration space

The board (the interface chip) is identified by a set of IDs in PCI configuration space as listed below:

Purpose	Value	Found in
Vendor ID	0x13c3	CFG space register 0
Device ID	0x7000	CFG space register 0
Subsystem Vendor ID	0x13c3	CFG space register 0x2c
Subsystem ID	0x7000	CFG space register 0x2c

The LSB of the Subsystem ID codes the hardware revision of the CAN-PMC/FD board. Currently the following version coding is used:

LSB of Subsystem ID	CAN-PMC/FD revision	Features
0x00	V1.0	-

The PCI express interface provides address spaces to access the boards dual port memory (a fraction of the DDR3 memory) and a register space with various control registers.

PCI base address register	Description	Size
0	Dual Port Memory	1 MByte
1	Control/Status Registers	1 MByte

The actual addresses for these memory spaces are configured by the PCI-BIOS of your system every time the computer is booted. If you wish to access one of these spaces, then you need to read the actual addresses from the PCI configuration space!

4.2 Dual Port Memory

The dual port memory is fully under control of the firmware installed on the CAN-PMC/FD.

4.3 Control/Status Registers

The registers are as follows:

Address Offset	Device	Access
0x90000	Control-register	read/write
0x90004	Interrupt-status-register	read
0x90008	Feature-register	read
0x9000c	Power-on-reset-register	read/write
0x90010	PCI-interrupt-set-register	read/write
0x90014	PCI-interrupt-clear-register	read/write
0x90018	Flag-set-register	write
0x9001c	Flag-clear-register	write
0x90020	Local-L-interrupt-set-register	read/write
0x90024	Local-L-interrupt-clear-register	read/write
0x90028	Local-T-interrupt-set-register	read/write
0x9002c	Local-T-interrupt-clear-register	read/write
0x90030	Scratchpad-register	read/write
0x9003c	Reset-register	write

table 2: Address assignment control/status egisters

The on-board registers can be accessed by the ARM SoC processor and by the host. The host can access these registers relative to PCI BAR1. Firmware defines with registers are used by SoC and which are used by the host.

4.3.1 Control Register

On the front bezel of the CAN-PMC/FD are located two user programmable LED´s. These LED´s can be switched by bit D24 and D25 of the control-register. Furthermore the CAN-termination can be switched on/off by the bits D26 and D27.

The control register should normally not be used by the host, as the LEDs and the CAN-termination are controlled by the firmware of the CAN-PMC/FD.

CTR_REG **0x90000 (lword, r/w)**

31 30	29	28	27	26	25	24	23 – 0
Reserved	T3	T2	T1	T0	LED1	LED0	not used

LED[1..0] Theses bits controls the LED´s on the front bezel of the CAN-PMC/FD.
LED0 = 1 LED0 (green) is on.
LED0 = 0 LED0 (green) is off.

LED1 = 1 LED1 (red) is on.
LED1 = 0 LED1 (red) is off.

T[3..0] Theses bits controls the CAN-termination of the four CAN interfaces.
T0 = 1 CAN-termination CAN_1 is on.
T0 = 0 CAN-termination CAN_1 is off.
T1 = 1 CAN-termination CAN_2 is on.
T1 = 0 CAN-termination CAN_2 is off.
T2 = 1 CAN-termination CAN_3 is on.
T2 = 0 CAN-termination CAN_3 is off.
T3 = 1 CAN-termination CAN_4 is on.

T3 = 0 CAN-termination CAN_4 is off.

Reserved Reserved positions are undefined, and must not be considered. Software must mask them off.

4.3.2 Interrupt Status Register

All CAN-controllers are able to generate interrupts to the ARM processor. All interrupt lines are logical ored.

For differentiation of the interrupt source a bit in the Interrupt Status Register (INT_STAT) is activated (active low).

INT_STAT **0x90004 (lword, ro)**

31	30	29	28	27	26	25	24	23 – 0
Reserved				I3	I2	I1	I0	not used

I[3..0] Interrupt status info. Each defined bit in this register reflects the status of the INT# pin of the corresponding CAN-Controller. A zero will be read when an interrupt is pending.
I0 coresponds to CAN-controller 1; I1 coresponds to CAN-controller 2 and so on.

Reserved Reserved positions are undefined, and must not be considered. Software must mask them off.

4.3.3 Feature Register

The feature register is used for identification of additional features in further versions. By reading this register it is possible to distinguish different versions or extensions.
The bits D24 – D31 of the feature register are zero on the CAN-PMC/FD.

FTR_REG **0x90008 (lword, ro)**

31	30	29	28	27	26	25	24	23 – 0
F7	F6	F5	F4	F3	F2	F1	F0	not used

F[3..0] Feature info. Each bit combination defines the corresponding feature in future designs. On the CAN-PMC/FD these bits are all zero.

F[7..4] These bits are reserved to read on further versions the position of a hex switch, if implemented. On the CAN-PMC/FD these bits are all zero.

4.3.4 Power-on reset Register

The power-on reset register has the feature, that the contents are not cleared by a local reset. The implemented bit R0 is only cleared by a power-on reset.

PWOR_REG **0x9000c (lword, r/w)**

31	30	29	28	27	26	25	24	23 – 0
Reserved							R0	not used

R0 The content of R0 is not affected by a local reset, but is only cleared by a power-on reset.

Reserved Reserved positions are undefined, and must not be considered. Software must mask them off.

4.3.5 PCI Interrupt Registers

On the CAN-PMC/FD are several registers to generate interrupts from the local processor to the PCI. PCI requests can be masked off by the CPU. This is done through the Flag set/clear registers. PCI interrupts from the local CPU are disabled after RESET, and must be enabled before using.

FLAG_SET **0x90018 (lword, wo)**

31 – 25	24	23 – 0
Reserved	P0	not used

FLAG_CLEAR **0x9001c (lword, wo)**

31 – 25	24	23 – 0
Reserved	P0	not used

P0	Writing this bit enables/disables interrupts from the local CPU to the PCIbus. Both registers are accessed in hot-1 technique: Writing a one to P0 enables/disables further interrupts to the PCIbus, writing zero to P0 does not affect the interrupt mask status.
Reserved	Reserved positions are undefined, and must not be considered. Software must mask them off.

If the PCI Interrupt is enabled, the bits PI0 – PI3 in the PINT_SET and PINT_CLEAR Register can be used to generate interrupts to the PCI.

PINT_SET **0x90010 (lword, r/w)**

31 – 28	27	26	25	24	23 – 0
Reserved	PI3	PI2	PI1	PI0	not used

PINT_CLEAR **0x90014 (lword, r/w)**

31 – 28	27	26	25	24	23 – 0
Reserved	PI3	PI2	PI1	PI0	not used

PI[3..0]	Writing one of these bits sets/resets interrupt requests from the local CPU to the PCI. Both registers are accessed in hot-1 technique: Writing a one to PI[3..0] sets/resets the corresponding bit in the registers.
Reserved	Reserved positions are undefined, and must not be considered. Software must mask them off.

4.3.6 Local Interrupt Registers

Local Interrupts on the CAN-PMC/FD module can be generated from the host in two different ways. The registers LINT_SET, LINT_CLEAR and TINT_SET, TINT_CLEAR are available to set/clear host interrupt requests to the local CPU.
The bits LIO-LI3 and TI0 - TI3 are ored and connected to the local CPU. An interrupt is generated, if a bit in the LINT or TINT register is set.

LINT_SET **0x90020 (lword, r/w)**

31 – 28	27	26	25	24	23 – 0
Reserved	LI3	LI2	LI1	LI0	Not used

LINT_CLEAR **0x90024 (lword, r/w)**

31 – 28	27	26	25	24	23 – 0
Reserved	LI3	LI2	LI1	LI0	Not used

LI[3..0]	Writing one of these bits sets/resets interrupt requests from the host to the local CPU. Both registers are accessed in hot-1 technique: Writing a one to LI[3..0] sets/clears the corresponding bit in the registers.

Reserved	Reserved positions are undefined, and must not be considered. Software must mask them off.

TINT_SET **0x90028 (lword, r/w)**

31	30	29	28	27	26	25	24	23 – 0
Reserved				TI3	TI2	TI1	TI0	Not used

TINT_CLEAR **0x9002c (lword, r/w)**

31	30	29	28	27	26	25	24	23 – 0
Reserved				TI3	TI2	TI1	TI0	Not used

TI[3..0]	Writing one of these bits sets/clears interrupt requests from the host to the local CPU. Both registers are accessed in hot-1 technique: Writing a one to TI[3..0] sets/clears the corresponding bit in the registers.
Reserved	Reserved positions are undefined, and must not be considered. Software must mask them off.

4.3.7 Scratchpad register

The scratchpad register is implemented on the CAN-PMC/FD for storage of hardware specific driver informations.

SCPR_REG **0x90030 (lword, r/w)**

31	30	29	28	27	26	25	24	23 – 0
S7	S6	S5	S4	S3	S2	S1	S0	not used

S[7..0]	The bits can be used to store firmware specific driver informations.

4.3.8 Reset register

The reset register allows the PCI host to request a firmware reset to the CAN-PMC/FD.

RESET_REG **0x9003c (lword, r/w)**

31	30	29	28	27	26	25	24	23 – 0
RV								not used

RV	To request reset, the values 1,2,4 and 8 have to be written to this register.

Despite of older products (CAN-PMC, CAN-PCIH) issuing a reset through this interface does not reset the processor or the PCI interface. Hence saving and restoring PCI configuration is not required.

After issueing reset the host should wait 0.1 s before attempting to access the CAN-PMC/FD again.

5 Appendix

5.1 Technical Data

CPU	
Type	Zynq ARM SoC,
Clock Frequency	Up to 667 MHz
Memory	
SDRAM	64Mbyte DDR3 (16bit wide) accessible from PCI and local bus.
Boot FLASH	16MByte SPI flash
Interfaces	
4x CAN FD	25 pin D-SUB connector, IP Core inside Zynq
CAN Transceiver	MCP2561FD, rated up to 8 MBit
Serial Interface	Optional on 25 pin D-SUB connector. RS232 level.
PCIbus interface specification	
Interface Controller	PCI V2.2 compliant Integrated 32 bit PCI interface, operating up to 33 MHz. 3.3 and 5 V universal.
PCIbus connector	64 pins, U1/U2 connector (P1386, Draft 2.0)
Physical Dimensions	
Modul Size	149mm x 74mm
Height	As defined by PMC Specifications
Slot requirements	1 PMC mezzanine slot
Weight	90 g
Power Requirements	
+3V3 (±5%)	**TBD**
+5V (±5%)	**TBD**
Power dissipation	**typ. 90**
Environmental Specifications	
Temperature range	0..+TBD°C (operating)
Humidity	0%..80%, non condensing
CAN isolation	500V (test voltage)
Reliability	
MTBF	**TBD**

5.2 Pin Assingment PMC P14 (U5) I/O connector

Pin #	Signalname	Pin #	Signalname
1	-	2	GND
3	VCC[8)]	4	TxD_S0[2)]
5	IO_E06	6	IO_E07
7	RxD_S0[2)]	8	+3V3_IO[7)]
9	IO_E05	10	IO_E04
11	EX-VCC_CH1[1)]	12	-
13	-	14	EXGND_CH1[9)]
15	-	16	-
17	CAN-L_CH1[9)]	18	-
19	CAN-H_CH1[9)]	20	-
21	EXVCC_CH4[9)]	22	CAN-L_CH3[9)]
23	CAN-H_CH3[9)]	24	EXGND_CH4[9)]
25	-	26	EX-VCC_CH3[1)]
27	CAN-L_CH4[9)]	28	EXGND_CH3[9)]
29	CAN-H_CH4[9)]	30	VCC
31	EXVCC_CH2[1)]	32	CAN-L_CH2[9)]
33	CAN-H_CH2[9)]	34	EXGND_CH2[9)]
35	-	36	-
37	CAN-L_CH2[9)]	38	-
39	CAN-H_CH2[9)]	40	-
41	-	42	-
43	+3V3_IO[7)]	44	Tx0_CH1[4)]
45	Tx0_CH0[3)]	46	IO_E00
47	GND	48	VCC
49	Rx0_CH1[4)]	50	Rx0_CH0[3)]
51	IO_E01	52	GND
53	VCC	54	Tx0_CH3[6)]
55	Tx0_CH2[5)]	56	IO_E02
57	GND	58	VCC[8)]
59	Rx0_CH3[6)]	60	Rx0_CH2[5)]
61	IO_03	62	GND
63	-	64	-

table 3: Pin assignment of PMC J14 I/O connector

Notes	
1	Optional the EXVCC voltage can be routed to the connector by placing option resistors. This can be used to power external transceiver.
2	Serial port signals from Zynq UART (LVTTL signals)
3,4,5,6	Rx/Tx from CAN controller, LVTTL signals. Tx may only be driven when no transceiver is equipped on the CAN-PMC/FD board.
7	Internal 3.3V power supply for the Zynq IO banks. All IOs – except CAN-L and CAN-H – **must not be driven** until +3V3_IO is powerd up. Best use +3V3_IO to power all external logic (e.g. level shifters). Note that +3V3_IO must not be connected to +3V3_PMC or any other +3V3 supply.
8	This is the power supply as provided by the PMC connector.
9	By default the isolated CAN signals are not routed to the PMC IO connector, as this breaks the isolation clearance. As for such an option.

5.3 References

These references direct you to manuals and specification that you might need to know when you attempt programming the CAN-PMC/FD. Most of the documents can be downloaded from the Internet. Look for the web servers of the chip manufacturers.

[1] **PCI local bus specification**, PCI Special Interest Group, Revision 2.2, December 1998.

WWW-References

Janz Tec AG	www.janztec.com
PCI Special Interrest Group	www.pcisig.org

5.4 Product History

Version	Release Date	Name	Changes
V1.0			• Protoype
V1.1	2016-06-14	As	• Protoype bugs fixed (wires, etc), layout improvements • Board temperature sensor added • Removed expansion routing option (CH5..CH8) • Added possibility for non-isolation option • PMC P14 now has +3V3_IO instead of +3V3_PMC • PEX8112 is now powered by +3V3_PMC instead of +3V3_IO
			•

5.5 Manual History

Version	Release Date	Name	Changes
1.0	2016-06-14	As	• Initial
			•