

ZKPCI

Manual



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 1 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

1 Contents

| | | |
|----------|--|-----------|
| 1 | CONTENTS | 2 |
| 2 | DEVICE CONCEPT | 3 |
| 2.1 | GENERAL | 3 |
| 2.2 | BLOCK DIAGRAM | 3 |
| 2.3 | MEASUREMENT PROCEDURE..... | 3 |
| 2.4 | MEASUREMENT PROCEDURE WITH TRIGGER PROBES | 4 |
| 2.5 | OTHER FUNCTIONS | 4 |
| 3 | PCI-ADDRESS AREA | 5 |
| 3.1 | PCI-I/O | 5 |
| 3.2 | FPGA-REGISTER..... | 5 |
| 3.3 | REGISTER DESCRIPTION FPGA | 6 |
| 4 | BOARD LAYOUT..... | 9 |
| 4.1 | MEASURING SYSTEM CONNECTION | 9 |
| 4.2 | TRIGGER PROBE CONNECTION..... | 10 |
| 4.3 | LVTLL-IN/OUTPUT..... | 10 |
| 5 | IMPORTANT CHARACTERISTIC VALUES | 10 |
| 6 | PC – SOFTWARE | 11 |
| 6.1 | DEVICE DRIVER | 11 |
| 6.2 | DYNAMIC LINK LIBRARY ZKPCI.DLL AND DEMO PROGRAM | 11 |
| 6.2.1 | <i>Program start and selection of counter board.....</i> | <i>11</i> |
| 6.2.2 | <i>Time-triggered measuring mode.....</i> | <i>11</i> |
| 6.2.3 | <i>The interrupt triggered measuring mode.....</i> | <i>11</i> |
| 6.2.4 | <i>Other Functions</i> | <i>11</i> |
| 7 | THE ZKPCI DYNAMIC LINK LIBRARY (DLL) | 12 |
| 7.1 | PROJECT FILES | 12 |
| 7.2 | INTEGRATION IN YOUR PROJECT | 12 |
| 7.2.1 | <i>Implicit binding</i> | <i>12</i> |
| 7.2.2 | <i>Explicit binding</i> | <i>13</i> |
| 7.2.3 | <i>CreateCZKPCIInstance</i> | <i>14</i> |
| 7.2.4 | <i>DllGetVersion</i> | <i>14</i> |
| 7.3 | THE CLASS CZKPCI..... | 15 |
| 7.3.1 | <i>CZKPCI::DestroyCZKPCIInstance.....</i> | <i>16</i> |
| 7.3.2 | <i>CZKPCI::GetCardCount</i> | <i>16</i> |
| 7.3.3 | <i>CZKPCI::Open</i> | <i>16</i> |
| 7.3.4 | <i>CZKPCI::Close.....</i> | <i>16</i> |
| 7.3.5 | <i>CZKPCI::Init</i> | <i>17</i> |
| 7.3.6 | <i>CZKPCI::ReadCardInfo.....</i> | <i>17</i> |
| 7.3.7 | <i>CZKPCI::WriteCardInfo</i> | <i>18</i> |
| 7.3.8 | <i>CZKPCI::MeasureAll</i> | <i>18</i> |
| 7.3.9 | <i>CZKPCI::ReadFPGARegister</i> | <i>19</i> |
| 7.3.10 | <i>CZKPCI::WriteFPGARegister.....</i> | <i>19</i> |
| 7.3.11 | <i>CZKPCI::TriggerInterruptEnable</i> | <i>19</i> |
| 7.3.12 | <i>CZKPCI::TriggerInterruptDisable.....</i> | <i>20</i> |
| 7.3.13 | <i>CZKPCI::GetErrorDescription</i> | <i>20</i> |
| 8 | REVISION HISTORY | 21 |



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 2 of 21

Title: **ZKPCI**

Name of Document:
457-HB -1-4-E-ZKPCI

2 Device concept

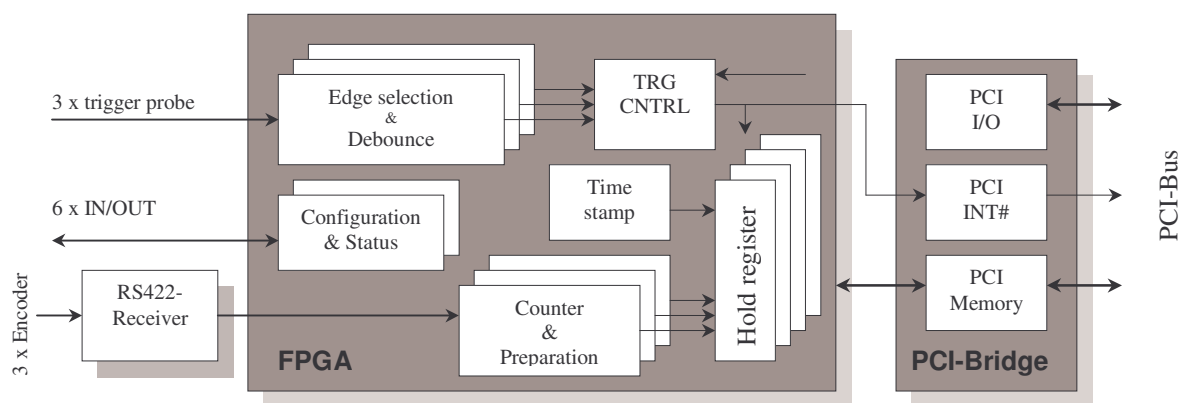
2.1 General

The counter board ZKPCI analyses output signals of incremental measuring systems with RS422 rectangular signals. These are usually two rectangular sequences shifted by 90° as well as a third signal to define a zero point (reference signal). These output signals are counted applying a 1-, 2- or 4-edge evaluation.

Each of the three counter channels contains a predetermined 32 bits up/down counter with Reset and a separate hold register as well as a logic for the adaptation of the input signals to the different counter modes. Additionally, a reference point signal can be processed and an external error signal can be stored. It is possible to connect up to three probes for the simultaneous triggering of all counter channels. The active trigger edge as well as a debouncing option can be configured for each of the three trigger probes. By using an additional 32-bit counter, the speed of the connected measuring system can be determined during every measurement.

The counter board ZKPCI is realised as a PCI-plug-in board. A memory area which is mapped into the PC-address area allows the communication with the user software.

2.2 Block diagram



2.3 Measurement procedure

When working with the counter board ZKPCI the following typical steps have to be performed:

Initialisation

- Ø Configuration of the counter and the trigger probes
- Ø Reset of all status flags
- Ø Initialisation of the PCI-Register for interrupt processing
- Ø Reference run to define the zero point reference

Measurement

- Ø Readout of the status register - if no unconfirmed trigger (probe interrupt) exists, the counter values are transferred with this command to the trigger hold registers
- Ø Readout of the counter hold registers and the time stamp register
- Ø Evaluation of all bits of the status register. If necessary, the measurement has to be rejected.
- Ø Confirmation of trigger events to unlock the counter hold registers - only necessary if a trigger event was recognized.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 3 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

Reference run

- Ø Set the reference request flags
- Ø Move the measuring sensors over their reference points (=Index- or zero point). When the reference point has been recognised, the reference request flag is automatically reset by the counter board.
- Ø By reading the status register it can be checked, which of the counter channels have already recognized the reference point.

2.4 Measurement procedure with trigger probes

When a trigger edge has been recognised (see also register `TCFG`), the current counter values are immediately transferred in the counter hold register, and the associated `TRG`-flag is set in the status register. Then the hold registers are locked. Unlocking is inhibited by executing the software command `CTRG` (see also register `CMD`).

During the readout process by the user software the counter hold registers are not locked (read out of the `STAT`-register). This means that in case of a recognised `TRG`-bit in the status register the previous measurement released by the software has to be rejected. Otherwise, in case of a simultaneous appearance of a trigger event and a readout of the counter registers by the user software, the synchronous measurement of all channels would not be guaranteed.

For each of the three trigger inputs it is possible to define the active edge, to activate a debouncing and to enable the PCI-Interrupt. Every trigger input can be activated or deactivated separately. The logical OR-combination of the active trigger inputs works on **all** counter channels.

2.5 Other Functions

Error pulses

For external error signals, in every channel a fourth input (`ENC_ST`) is implemented. A rising edge at this input is stored and reported as active `ERR`-bit in the status register. These errors flags can be reset by means of a command in the `CMD` register.

Interrupt

The ZKPCI can be configured in such a way, that a recognised trigger signal releases a PCI-Interrupt. This interrupt is level sensitive and **must** be confirmed by deleting the `TRG`-bits (see also register `TCFG` and `CMD` and description in section 3.1).

Boot flag

On the counter board a supervisor-IC is implemented. It arranges a reset and the reload of the implemented FPGA circuit in case of a PC-reset or too low supply voltage. In this case all registers are set to their default values. A boot flag (`BTF`) in the status register indicates this state. If the `BTF`-Flag is recognised, the user software has to execute a new initialisation and to delete this flag.

Speed measurement

Another 32 bits counter (`TSTAMP`) is incremented with a frequency of 33MHz. Its value is transferred with every measurement in a hold register. In a simple manner, the speed of the connected sensor systems can be determined by using the difference of two consecutive counter values and the time stamps as variables for calculation.

Common in- and outputs

The counter board contains 6 common in- and outputs for LVTTTL-level. They can be accessed by appropriate registers.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 4 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

3 PCI-Address area

3.1 PCI-I/O

The PCI-connection of the ZKPCI is realised in form of a PCI bridge on the board. The following blocks are mapped into the PC address area:

| Block | Type | Size (Byte) | Description |
|-------|--------|-------------|------------------------|
| BAR0 | Memory | 256 | Register of PCI-Bridge |
| BAR1 | I/O | 256 | |
| BAR2 | Memory | 64 | FPGA-register |

The registers of the PCI Bridge are mapped in the I/O address area as well as in the memory area of the PC. **It is not allowed to change the content of the registers to guarantee the correct functionality of the counter board.**

The following bits can be used by the user software:

| Register | I/O-Address offset | Bit | Access | Default | Description |
|----------|--------------------|-----|--------|---------|--------------------------------------|
| INTCSR | 0x68 | 8 | R/W | 1 | PCI-Interrupt Enable (H-active) |
| | | 11 | R/W | 0 | FPGA-Interrupt Enable (H-active) |
| | | 15 | R | 0 | FPGA-Interrupt active (H-active) |
| | | 16 | R/W | | Bit must be reset |
| CNTRL | 0x6C | 16 | R/W | 1 | Reset FPGA (L-active) |
| | | 17 | R | 0 | FPGA-Configuration loaded (L-active) |



Do not change the other bits in these both registers! If bit 17 of the register CNTRL (0x6C) is set, an access to the FPGA is not allowed! Otherwise, the PC may crash.


The Vendor-ID of the counter card ZKPCI is 0x10B5, the Device-ID 0x2749. The serial number of the device is stored in the EEPROM of the PCI-bridge. For the access the PCI-VPD functions are used.

3.2 FPGA-register

The FPGA registers are mapped as a 64 byte area into the memory address area of the PC. The base address of this address area is set at boot time of the PC or when starting the operating system.

| Address | Write | Read | Description |
|---------|-------|--------------------|---------------------------------------|
| 0x00 | CNT0 | CNT0 | Counter channel 0 |
| 0x04 | CNT1 | CNT1 | Counter channel 1 |
| 0x08 | CNT2 | CNT2 | Counter channel 2 |
| 0x0C | CNT3 | CNT3 = 0xFFFF_FFFF | Counter channel 3 |
| 0x10 | CNT4 | CNT4 = 0xFFFF_FFFF | Counter channel 4 |
| 0x14 | CNT5 | CNT5 = 0xFFFF_FFFF | Counter channel 5 |
| 0x18 | CNT6 | CNT6 = 0xFFFF_FFFF | Counter channel 6 |
| 0x1C | CNT7 | CNT7 = 0xFFFF_FFFF | Counter channel 7 |
| 0x20 | CMD | STATUS/Getval-CMD | Command / Status |
| 0x24 | TCFG | TCFG | Trigger configuration, LVTTIL-Outputs |
| 0x28 | CCFG | CCFG | Counter configuration |
| 0x2C | | TSTAMP | Time stamp |
| 0x30 | | 0xFFFF_FFFF | |
| 0x34 | | 0xFFFF_FFFF | |
| 0x38 | BTEST | BTEST | Bus-Test |
| 0x3C | | VER | FPGA-Version |

Write accesses to unused counter channels have no effect, read accesses to these channels return 0xFFFF_FFFF.

| | | |
|--|---|---------------------|
|  <p>GEMAC Gesellschaft für Mikroelektronikanwendung Chemnitz mbH</p> <p>Zwickauer Straße 227 D-09116 Chemnitz, Germany</p> | Phone: +49 371 33 77 - 0 Fax: +49 371 33 77 272 Internet: www.gemac-chemnitz.de Email: interpolation@gemac-chemnitz.de sales@gemac-chemnitz.de | Title: ZKPCI |
| | Date: 24.03.04 | Page 5 of 21 |

3.3 Register description FPGA

CNT Counter

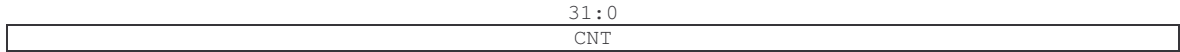
Address: 0x00 – 0x1C

Access: R/W

Reset value: 0x00000000

Description

Read accesses act on the counter hold registers. Write accesses to these registers allow pre-adjustment of the counters.



CNT Counter value in 32 bit twos-complement format

TSTAMP Time stamp

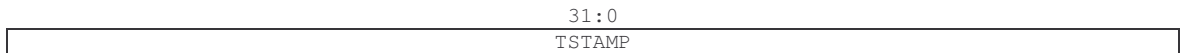
Address: 0x2C

Access: R

Reset value: 0x00000000

Description

With the help of the time stamp the speed of each of the measurement systems can be determined. For every transfer of the counter values to the counter hold register (caused by the user software or a trigger), the time stamp is stored, too.



TSTAMP Time stamp in unsigned 32Bit-Format,
1 increment = 30.30ns (1/33MHz)

STAT Status register / Transfer of the counters into holding register

Address: 20

Access: R \bar{L} Lock counter value

Reset value: 0x80000000

Description

From this register all status information relevant for measurements can be read. With every read access to this register the values of all counters will be transferred to the corresponding hold registers. Exception: Unconfirmed trigger (Interrupt). In this case, one of the bits TRG1, TRG2 or TRG3 will be set.

For software-controlled measurements this register must be selected before the readout of the counter channels.

| | | | | | | | | | | | | | | | |
|----|----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BT | PI | IN5 | IN4 | IN3 | IN2 | IN1 | IN0 | 0 | 0 | 0 | 0 | 0 | TRG3 | TRG2 | TRG1 |

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR7 | ERR6 | ERR5 | ERR4 | ERR3 | ERR2 | ERR1 | ERR0 | REF7 | REF6 | REF5 | REF4 | REF3 | REF2 | REF1 | REF0 |

| Bit | Description | 0 | 1 |
|------|-------------------------|---|---|
| BT | Boot-Flag | Standard use | The FPGA of ZKPCI has been reset |
| PI | PI203-Flag | No PI203 at X9 connected | PI203 at X9 recognised |
| TRGx | Trigger-Flag | No trigger (probe impulse) recognised The counter registers are unlocked | Trigger (probe impulse) recognised. The counter registers are locked |
| ERRx | Error-Flag | No error (L/H-Edge) recognised | Error (L/H-Edge) recognised |
| REFx | Reference point request | No activated | Activated |
| INx | LVTTTL-Input | Associated Input leads L-Level | Associated Input leads H-Level |

i The bits *BT*, *TRG* and *ERR* should be reset during the initialisation of the ZKPCI (see *CMD* register) and should be tested after every read access.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
 Fax: +49 371 33 77 272
 Internet: www.gemac-chemnitz.de
 Email: interpolation@gemac-chemnitz.de
 sales@gemac-chemnitz.de

Date: 24.03.04

Page 6 of 21

Title: **ZKPCI**

Name of Document:
457-HB -1-4-E-ZKPCI

CMD Command execution

Address: 20

Access:W (no storage)

Reset value: -

Description

Write accesses to this register allow the command execution. Every bit corresponds to a command. These bits can be combined (e.g., command 0x8100_FFFF as master-reset for the initialization). The command word is not stored.

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CBF | - | - | - | - | - | - | CTRG | SRF7 | SRF6 | SRF5 | SRF4 | SRF3 | SRF2 | SRF1 | SRF0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLE7 | CLE6 | CLE5 | CLE4 | CLE3 | CLE2 | CLE1 | CLE0 | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |

- CBF** Clear Boot-Flag
The boot Flag (BTF) will be deleted. This command should be implemented during the device initialisation.
- CTRG** Clear Trigger-Flag (Clear Interrupt)
Command for confirmation of a trigger event or interrupt. The counter holding registers will be unlocked.
- CLEx** Clear Error-Flag
Command for the confirmation of a recognised error signal
- SRF** Setting of reference point request
Command to the activation of the reference point processing
- RC** Reset Counter and Reset reference point request
Counter and reference request-flags will be reset.



The command RC does not affect the counter hold registers and the error Flags (ERRx).

TCFG Configuration Trigger

Address: 24

Access: R/W

Reset value: 0x00000000

Description

Register for the configuration of the three trigger inputs of the measuring probes.

| | | | | | | | | | | | | | | | |
|----|----|------|------|---------------|------|------|---------------|----|----|---------------|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | - | OUT5 | OUT4 | OUT3 | OUT2 | OUT1 | OUT0 | - | - | - | - | - | - | - | - |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | SIG | Probe mode T3 | | | Probe mode T2 | | | Probe mode T1 | | | | | |

| Probe mode Tx | | | |
|---------------|------|------|-----|
| 3 | 2 | 1 | 0 |
| ENAIRQ | FAST | EDGE | ENA |

- T1: Probe connection labelled with T1 (card-mounting bracket)
T2: Probe connection labelled with FS (card-mounting bracket)
T3: Probe connection via 40-pole pin header X9 (PI203)

| Bit | Description | 0 | 1 |
|--------|---------------------------------------|---------------------------------|----------------------------------|
| SIG | Acoustic signal while triggering | Deactivated | Activated |
| ENA | Global activate probe input | Deactivated | Activated |
| EDGE | Active signal edge of the probe input | Falling (opening probe switch) | Rising (closing probe switch) |
| FAST | Debouncing of the probe input | No debouncing (Time 90 ns) | Debouncing (Time constant >60ms) |
| ENAIRQ | PCI-Interrupt | No activated | Activated |
| OUTx | LVTTTL-Output | Associated output leads L-Level | Associated output leads H-Level |



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 7 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

CCFG Counter Configuration

Address: 28

Access: R/W

Reset value: 0x55555555

Description

Register for counter configuration.

| | | | | | | | | | | | | | | | |
|-------|--------|----|----|-------|--------|----|----|-------|--------|----|----|-------|--------|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIR 7 | Mode 7 | | | DIR 6 | Mode 6 | | | DIR 5 | Mode 5 | | | DIR 4 | Mode 4 | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIR 3 | Mode 3 | | | DIR 2 | Mode 2 | | | DIR 1 | Mode 1 | | | DIR 0 | Mode 0 | | |

| Mode | Counter | Counting direction forward | Counting direction backwards |
|---------|---------------------|-------------------------------|-------------------------------|
| 000 (0) | Up/Down-Counter | Impulse on signal A | Impulse on signal B |
| 001 (1) | Rising Edge A | Signal B = L | Signal B = H |
| 010 (2) | Rising Edge B | Signal A = H | Signal A = L |
| 011 (3) | Both Edges A | Signal A ≠ Signal B | Signal A = Signal B |
| 100 (4) | Both Edges B | Signal A = Signal B | Signal A ≠ Signal B |
| 101 (5) | 4-Edges (A/B) | Like Mode 3 and Mode 4 | Like Mode 3 and Mode 4 |
| 110 (6) | Counter deactivated | | |
| 111 (7) | Counter deactivated | | |

DIR 0: Mode 0: Falling edge
Mode 1-5: Counting direction as table (Signal A leads to Signal B by 90°)
1: Mode 0: Rising edge
Mode 1-5: Counting direction inverse to table (Signal A lags to Signal B by 90°)

BTEST Bus-Test

Address: 38

Access: R/W

Reset value: 0x12488421 (Write)

Description

Register for the test of the interface between FPGA and PCI Bridge. The read value corresponds to the register content rotated 1 bit to the right.

| | |
|--------------|------------|
| 31:1 | 0 |
| BTEST (30:0) | BTEST (31) |

VER Version

Address: 3C

Access: R

Reset value: 0x4846_VERSION

Description

Hardware-Version

| | |
|--------|---------|
| 31:16 | 15:0 |
| 0x4846 | VERSION |



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

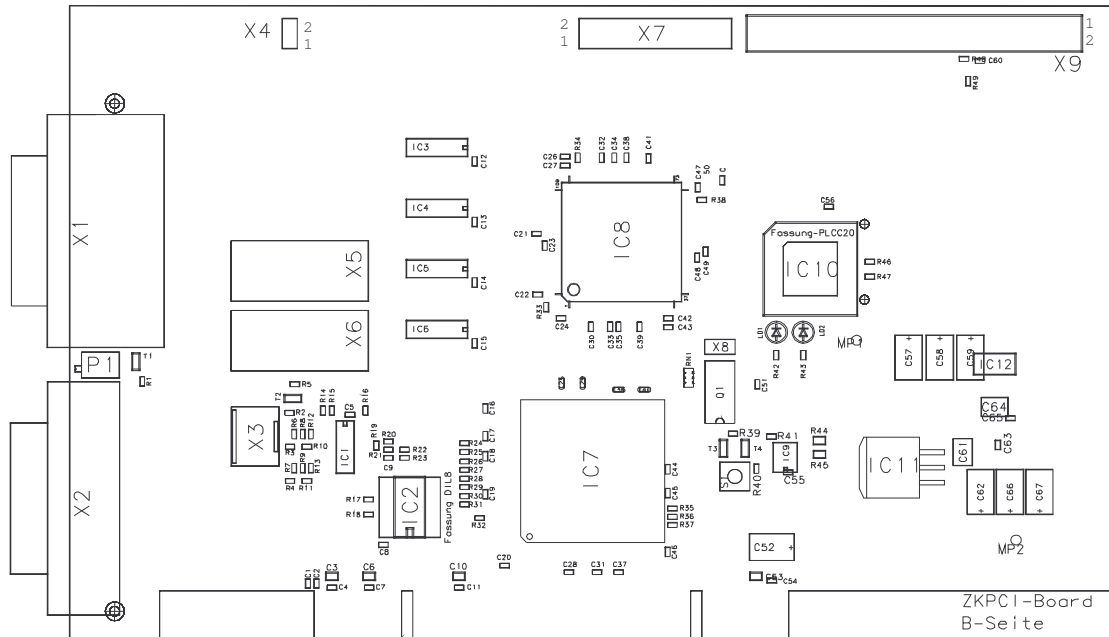
Page 8 of 21

Title:

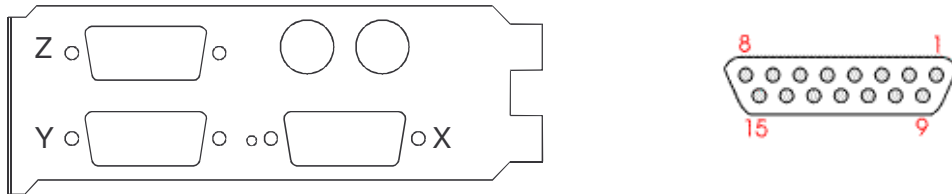
ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

4 Board layout



4.1 Measuring system connection



| Signal | Description | Channel X | | Channel Y | | Channel Z | |
|----------------|-------------------------|-----------|----------|-----------|------------|-----------|------------|
| | | X9 | SUB-D X2 | X9 | SUB-D X1/1 | X9 | SUB-D X1/2 |
| ENC_A+ | Counting input A + | 31 | 14 | 27 | 14 | 23 | 14 |
| ENC_A- | Counting input A - | 32 | 6 | 28 | 6 | 24 | 6 |
| ENC_B+ | Counting input B + | 33 | 13 | 29 | 13 | 25 | 13 |
| ENC_B- | Counting input B - | 34 | 5 | 30 | 5 | 26 | 5 |
| ENC_R+ | Reference point input + | 7 | 12 | 5 | 12 | 3 | 12 |
| ENC_R- | Reference point input - | 8 | 4 | 6 | 4 | 4 | 4 |
| ENC_ST+ | Error input + | 15 | 11 | 9 | 11 | 11 | 11 |
| ENC_ST- | Error input - | 16 | 3 | 10 | 3 | 12 | 3 |
| Load+ | Trigger T3 + | 17 | - | 17 | - | 17 | - |
| Load- | Trigger T3 - | 18 | - | 18 | - | 18 | - |
| U ₀ | Auxiliary voltage | 13 | 10 | 13 | 10 | 13 | 10 |
| GND | 0 V | 37 | 2 | 37 | 2 | 37 | 2 |
| GND | 0 V | 38 | 9 | 38 | 9 | 38 | 9 |
| VCC | 5 V | - | 7 | - | 7 | - | 7 |
| VCC | 5 V | - | 8 | - | 8 | - | 8 |
| PI203 | PI203 – Identification | 40 | - | 40 | - | 40 | - |

If no difference signals RS422 are available, the inputs for the inverted signals must be connected to the auxiliary voltage U₀.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

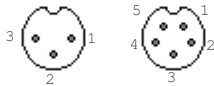
Page 9 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

4.2 Trigger probe connection



| Signal | Mini-Din 5-poles T1 | Mini-Din 3-poles T2 (FS) | Header X4 |
|----------------|---------------------|--------------------------|-----------|
| Trigger input | 5 | 1 | |
| GND | 4 | 2 | |
| VCC | 3 | | |
| GND | 2 | | |
| LED | 1 | | |
| Piezo buzzer + | | | 2 |
| Piezo buzzer - | | | 1 |



The LED in T1 lights, if the trigger probe is inactive.
The volume of the piezo buzzer can be adjusted with the potentiometer P1.

To configure the trigger probe T1 as a closing contact, the trigger probe T2 as a opening contact and to deactivate the trigger connection in X9, the register TCFG must contain the value 0x1031. In this case both probes are debounced and the piezo buzzer is active. No PCI-Interrupt is released.

4.3 LVTTLL-In/Output

| Signal | Pin on X7 | Signal | Pin on X7 |
|---------|-----------|--------|-----------|
| GND | 1 | OUT2 | 11 |
| GND | 2 | OUT1 | 12 |
| D.C | 3 | OUT0 | 13 |
| D.C | 4 | IN5 | 14 |
| D.C | 5 | IN4 | 15 |
| TestOut | 6 | IN3 | 16 |
| TestOut | 7 | IN2 | 17 |
| OUT5 | 8 | IN1 | 18 |
| OUT4 | 9 | IN0 | 19 |
| OUT3 | 10 | 3.3V | 20 |

All signals lead LVTTTL level (3.3-V logic), the inputs are 5V tolerant. The connections "D.C." and "TestOUT" must be open.

5 Important characteristic values

| Parameter | Min. | Type. | Max. | |
|---|------|-------|------|----|
| General operating conditions | | | | |
| Supply current | | 140 | | mA |
| Operating voltage | 4,75 | 5 | 5.25 | V |
| Temperature area | 0 | | 70 | °C |
| Input signals | | | | |
| Edge distance A-B | 35 | | | ns |
| Pulse width error-signal s_T | 130 | | | ns |
| Pulse width trigger signal (no debouncing active) | 130 | | | ns |
| Pulse width trigger signal (debouncing active) | 150 | | | ms |



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 10 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

6 PC – Software

6.1 Device driver

For the counter card ZKPCI the driver `windrivr6.sys` with the associated files `mvspci.inf` and `windrivr6.inf` for Windows 98/ME/2000/XP/NT is applied. **This driver may only be used in conjunction with the Library `zkpci.dll`.**

Start the script `Install.bat` for installation. Windows will recognise the ZKPCI board automatically and will install the driver. Under Windows98/ME/NT a reboot is necessary. To delete the device driver start the script `Deinstall.bat`.

6.2 Dynamic Link Library ZKPCI.DLL and demo program

The library `zkpci.dll` contains all functions to access the FPGA registers, functions for the initialisation of the ZKPCI as well as functions for interrupt processing. A detailed description of the library functions can be found in section 7.

The demo program shows the capabilities of the library and introduces to the work with the PCI card.

6.2.1 Program start and selection of counter board

To start the demo program, the `ZKPCI.DLL` is needed. It must be in the same directory as the program `ZKPCIDemo.exe`.

All counter boards of the type `ZKPCI` in the system are listed by the program and the first available board is selected when starting the program. If you have several boards in the system, you can switch between them.

6.2.2 Time-triggered measuring mode

This is the default mode after the program start. You should use this mode to become familiar with the functionality of the counter board, as well as to test the correct behaviour of a connected sensor.


In this mode, the program reads and displays every 20ms the counter values of all three channels, the status as well as the time stamp. When a trigger event occurs, one of the bits `TRG0`, `TRG1` or `TRG2` is set in the status register. Then the current measuring line is shifted downwards to the list standing below. At first, a trigger source must be enabled in the register `TCFG`. If you have selected the option `AutoClearIRQ`, trigger events are automatically confirmed by sending the command `CTRG` (see `CMD`-register).

6.2.3 The interrupt triggered measuring mode.

In this mode the interrupt functionality of the board is demonstrated. In opposite to the time triggering mode, in this mode no continuous reading is performed anymore, is done here as a response to an interrupt of the counter board. (See also register `TCFG` and `CMD` as well as section 3.1). Then the count values of all three channels, the status and the time stamp are read and the whole measuring line is shifted downwards to the list standing below. A trigger source must be enabled first in the register `TCFG` as interruptible. Trigger events are automatically confirmed by sending the command `CTRG` into the `CMD` register.

6.2.4 Other Functions

The further FPGA registers `CMD`, `TDFG`, `CCFG` and `BTEST` in hexadecimal and binary representation are ready for reading and writing. By means of another selector the arbitrary FPGA register can be read or written. In the lower part of the program window, a success or error message is indicated for every operation.

| | | |
|---|---|--|
|  GEMAC <small>Gesellschaft für Mikroelektronikanwendung Chemnitz mbH</small> | Phone: +49 371 33 77 - 0 | Title: ZKPCI |
| | Fax: +49 371 33 77 272 | |
| Zwickauer Straße 227 D-09116 Chemnitz, Germany | Internet: www.gemac-chemnitz.de | Name of Document: 457-HB -1-4-E-ZKPCI |
| | Email: interpolation@gemac-chemnitz.de sales@gemac-chemnitz.de | |
| | Date: 24.03.04 | Page 11 of 21 |

7 The ZKPCI Dynamic Link Library (DLL)

Important Note: The methods described here to integrate and to handle the instruction library are limited to MICROSOFT VISUAL C++ 6.0. For other programming environments, such as BORLAND C++, VISUAL BASIC or DELPHI, no hints are given.

7.1 Project files

For a comfortable work with the device " ZKPCI 3 channel counter " you have got an instruction library (*Dynamic link Library*). The provided software consists of the following files:

| | |
|--------------|---|
| ZKPCI.DLL | Contains the class CZKPCI with all necessary functions, as well as the function <code>CreateCZKPCIInstance</code> , for explicit binding of ZKPCI.DLL |
| ZKPCI.H | Definition of the class CZKPCI and its function headers for integration into your project |
| ZKPCITYPES.H | Definition of data types for the data transfer between your project and the functions of the class CZKPCI |
| ERRORCODES.H | Definition of error constants |

7.2 Integration in your project

The provided CZKPCI.DLL contains a class CZKPCI, as well as a function `CreateCZKPCIInstance`, which both are exported and available for your project. They allow to integrate the ZKPCI.DLL statically (implicitly) as well as dynamically (explicitly) in your project. Both methods are described here.

7.2.1 Implicit binding

Implicit binding of the ZKPCI.DLL is the simpler, but less flexible way. The functions of the library are linked to your project at compilation time. ZKPCI.DLL must be available each time when starting the program. Otherwise, a start of the program is not possible.

Steps to bind the ZKPCI.DLL implicitly into your project:

- (1) Copy all four provided files (see 7.1) in your project directory.
- (2) Add the header file ZKPCI.H to your project (this integrates the other required header files ZKPCITYPES.H, ERRORCODES.H.)
- (3) Define a member variable of the class CZKPCI in your project, e.g. `CZKPCI m_ZKPCI;`

Now you can access all member functions of the class with the help of the instance of the class CZKPCI.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 12 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

7.2.2 Explicit binding

Explicit binding of the `ZKPCI.DLL` is little more difficult, but offers more flexibility. The `ZKPCI.DLL` is not linked at the compilation time to your project, i.e. for the program start `ZKPCI.DLL` is not needed. When the functions are needed for the first time, the `ZKPCI.DLL` is dynamically loaded. If the `ZKPCI.DLL` does not exist, your program works nevertheless, however, cannot access the functions of the `ZKPCI.DLL`.

Steps to bind the `ZKPCI.DLL` explicitly into your project:

- (1) Copy the data `ZKPCI.DLL`, `ZKPCI.H`, `ZKPCITYPES.H`, and `ERRORCODES.H` in your project directory.
- (2) Add the header file `ZKPCI.H` to your project (This integrates the other required headerfiles `ZKPCITYPES.H`, `ERRORCODES.H`)
- (3) Set the global known member variable on the data type `HINSTANCE`, for example:

```
HINSTANCE m_hZKPCILibrary;
```

- (4) Furthermore set a member variable (Pointer) to the class `CZKPCI` in your project, for example.:

```
CZKPCI *m_pZKPCI;
```

- (5) To load the `ZKPCI.DLL` dynamically and to use functions contained in the class `CZKPCI`, implement the following source code (further details you will find in the source code of the demo program):

```
m_hZKPCILibrary = NULL;
m_pZKPCI = NULL;

m_hZKPCILibrary = LoadLibrary(_T("ZKPIC.DLL"));

if(m_hZKPCILibrary != NULL)
{
    typedef CZKPIC* (*CreateCCZKPCIInstance) (void);

    CreateCZKPCIInstance pfnCreateCZKPCIInstance =
        (CreateCZKPCIInstance)
        GetProcAddress(m_hZKPCILibrary,
            _T("CreateCCZKPCIInstance "));

    if (pfnCreateCZKPCIInstance)
    {
        m_pZKPCI = pfnCreateCZKPCIInstance();
    }

    if (m_pZKPCI)
    {
        //Your function orders
        ...

        m_pZKPCI->DestroyCZKPCIInstance();
    }

    FreeLibrary(m_hZKPCILibrary);
}
```



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 13 of 21

Title: **ZKPCI**

Name of Document:
457-HB -1-4-E-ZKPCI

7.2.3 CreateCZKPCIInstance

Function header:

```
CZKPCI* CreateCZKPCIInstance(void)
```

Arguments:

→ no arguments

Return parameter:

→ Pointer to new instance (in the error case ZERO)

The member function `CreateCZKPCIInstance` creates a new instance of the class `CZKPCI`. It can be called only once. With the `CZKPCI::DestroyCZKPCIInstance` the instance is released.

7.2.4 DllGetVersion

Function header:

```
HRESULT CALLBACK DllGetVersion(DLLVERSIONINFO *pdvi)
```

Arguments:

→ `pdvi` Pointer to `DLLVERSIONINFO` structure
The variable `cbSize` must be set before calling the function.

Return parameter:

→ `HRESULT` is `NOERROR` if successful, or OLE- defined error value.

The member function `DllGetVersion` is identical to the `DllGetVersion-Functions` of `Windows-Shell-DLLs`.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

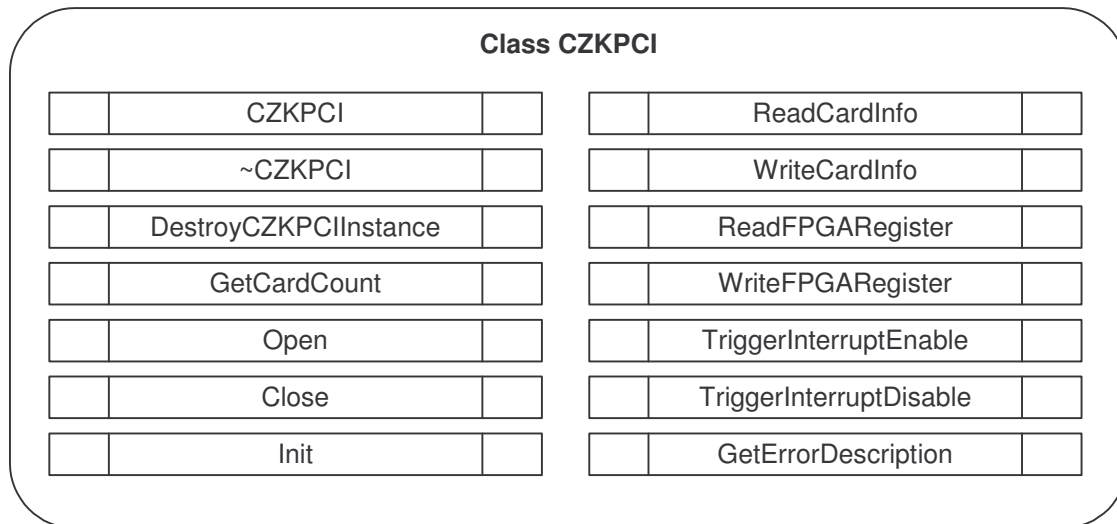
Page 14 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

7.3 The class CZKPCI



The class CZKPCI and its member functions

The library `ZKPCI.DLL` contains an exported class `CZKPCI`. In this class all functions (member functions) are contained which are necessary to communicate with the device " *ZKPCI 3 channel counter* " of your PC. The diagram shows an overview of the class `CZKPCI` and their member functions.

All functions of the class `CZKPCI` (besides of the function `CIPH001::ErrorText` and `CZKPCI::DestroyCZKPCIInstance`) are of the return type `DWORD`. After the execution of one of these functions an error value returns. These error values are defined in the header file `ERRORCODES.H` as constants. If no error occurs, the return value is `ZKPCI_ERR_SUCCESS (0x0000000L)`. During the execution of one of these functions only one error can occur.

For consistent data exchange between these functions and your application different structures exist. They are defined in the provided header file `ZKPCITYPES.H`. You find further explanations in the detailed function descriptions.

7.3.1 CZKPCI::DestroyCZKPCIInstance

Function header:

```
void CZKPCI::DestroyCZKPCIInstance(void)
```

Arguments:

→ no parameters

The member function `CZKPCI::DestroyCZKPCIInstance` destroys an instance of the `CZKPCI`-class created by `CreateCZKPCI Instance`.

7.3.2 CZKPCI::GetCardCount

Function header:

```
DWORD CZKPCI::GetCardCount(unsigned into *puiCardCount)
```

Arguments:

→ `puiCardCount`: Pointer to unsigned int variable for the return value (is loaded by the function)

The member function `CKPCI::GetCardCount` returns the number of the counter boards `ZKPCI` existing in the system.

7.3.3 CZKPCI::Open

Function header:

```
DWORD CZKPCI::Open(unsigned int uiCardNumber)
```

Arguments:

→ `uiCardNumber`: Card number in the system.
 $0 \leq \text{uiCardNumber} < \text{CardCount}$

The member function `CKPCI::Open` opens a board for the use in the system. Several boards can be used simultaneously in your application. Therefore the board number given in the `CZKPCI::Open` card function has to be submitted for all card relevant functions.

7.3.4 CZKPCI::Close

Function header:

```
DWORD CZKPCI::Close(unsigned int uiCardNumber)
```

Arguments:

→ `uiCardNumber`: Card number in the system.
 $0 \leq \text{uiCardNumber} < \text{CardCount}$

The member function `CKPCI::Close` closes a board which was opened by `CKPCI::Open`.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 16 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

7.3.5 CZKPCI::Init

Function header:

```
DWORD CZKPCI::Init(unsigned int uiCardNumber,  
                  ZKPCI_TCFGREG unionTCFG,  
                  ZKPCI_CCFGREG unionCCFG);
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- unionTCFG: TCFG Register (union for bit access)
- unionCCFG: CCFG Register (union for bit access)

Structure definition:

- ZKPCI_TCFGREG see ZKPCITYPES.H
- ZKPCI_CCFGREG see ZKPCITYPES.H

The member function `CKPCI::Init` executes a bus and version test of the FPGAs and sets the configuration registers `TCFG` and `CCFG`. It is recommended to call this function after the `CKPCI::Open` Function. The `TCFG` and `CCFG` register can be set also without this function by `CKPCI::WriteFPGA-Register`.

7.3.6 CZKPCI::ReadCardInfo

Function header:

```
DWORD CZKPCI::ReadCardInfo(unsigned int uiCardNumber,  
                           CARDINFO *pstcCardInfo)
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- pstcCardInfo: Pointer to `CARDINFO` Structure
(is loaded by the function)

Structure definition:

- `CARDINFO`
 - ↳ `char szSerialNumber[]`: Serial number string (12 characters)
 - ↳ `DWORD dwBus`: Bus number
 - ↳ `DWORD dwSlot`: Number of a slot, with a card inside
 - ↳ `DWORD dwFunction`: Card number

The member function `CKPCI::ReadCardInfo` `ReadCardInfo` supplies information about the applied PCI-plugin location and the serial number of the board.



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 17 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

7.3.7 CZKPCI::WriteCardInfo

Function header:

```
DWORD CZKPCI::WriteCardInfo (unsigned int uiCardNumber,  
                             CARDINFO stcCardInfo)
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- stcCardInfo: CARDINFO Structure

Structure definition:

- CARDINFO
 - ↳ char szSerialNumber[]: Serial number string (12 characters)
 - ↳ DWORD dwBus: not writable and is ignored
 - ↳ DWORD dwSlot: not writable and is ignored
 - ↳ DWORD dwFunction: not writable and is ignored

The member function `CKPCI::WriteCardInfo` allows the manufacturer to set the serial number of the board. If the user calls this function, no data are not changed and the error code `ZKPCI_ERR_MANUFACTUREFUNCTION` is sent back.

7.3.8 CZKPCI::MeasureAll

Function header:

```
DWORD CZKPCI::MeasureAll (unsigned int uiCardNumber,  
                          ZKPCI_VAL *pstcData,  
                          BOOL bAutoClearIRQ)
```

Arguments:


- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- pstcData: Pointer to `ZKPCI_VAL` structure
(is loaded by the function)
- bAutoClearIRQ: if TRUE, then the ClearIRQ-Bit (CTRG) is automatically set in the register
CMD, if a trigger event was recognised.

Structure definition:

- ZKPCI_VAL
 - ↳ DWORD dwCnt0: Value of the register CNT0
 - ↳ DWORD dwCnt1: Value of the register CNT1
 - ↳ DWORD dwCnt2: Value of the register CNT2
 - ↳ DWORD dwTStamp: Value of the register TSTAMP
 - ↳ ZKPCI_STATREG stcStatus: Value of the register STAT

The member function `CKPCI::MeasureAll` automatically supplies all register values required for a separate measurement and confirms the appearance of the required trigger events by setting the bit `CTRG` in the `CMD`-register to unlock the counters for a new measurement.

The register values can also be read by `CKPCI::ReadFPGARegister`. In this case, please note the remarks about the measuring process in the sections 2.3. and 2.4.

| | | |
|---|---|--|
|  GEMAC <small>Gesellschaft für Mikroelektronikanwendung Chemnitz mbH</small> | Phone: +49 371 33 77 - 0 | Title: ZKPCI |
| | Fax: +49 371 33 77 272 | |
| Zwickauer Straße 227 D-09116 Chemnitz, Germany | Internet: www.gemac-chemnitz.de | Name of Document: 457-HB -1-4-E-ZKPCI |
| | Email: interpolation@gemac-chemnitz.de sales@gemac-chemnitz.de | |
| | Date: 24.03.04 | Page 18 of 21 |

7.3.9 CZKPCI::ReadFPGARegister

Function header:

```
DWORD CZKPCI::ReadFPGARegister (unsigned int uiCardNumber,  
                                unsigned long ulAdr,  
                                unsigned long *pulData)
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- ulAdr: Address in the FPGA, definitions of valid addresses can be found in
ZKPCITYPES.H
- pulData: Pointer to register content (is loaded by the function)

The member function CKPCI::ReadFPGARegister reads a 32Bit register from the FPGA.

7.3.10 CZKPCI::WriteFPGARegister

Function header:

```
DWORD CZKPCI::WriteFPGARegister (unsigned int uiCardNumber,  
                                  unsigned long ulAdr,  
                                  unsigned long ulData)
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- ulAdr: Address in the FPGA, definitions of valid addresses can be found in
ZKPCITYPES.H
- ulData: new register content

The member function CKPCI::WriteFPGARegister writes a 32Bit Register in the FPGA.

7.3.11 CZKPCI::TriggerInterruptEnable


Function header:

```
DWORD CZKPCI::TriggerInterruptEnable  
(unsigned int uiCardNumber,  
 ZKPCI_INTERRUPT_HANDLER fnInterruptFunction)
```

Arguments:

- uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount
- fnInterruptFunction: Pointer on a CALLBACK interrupt function
(must be statically)

The member function CKPCI::TriggerInterruptEnable switches the trigger-Interrupt on. To release an interrupt by the trigger, one of the trigger sources in the TCFG register must be enabled, too. If the interrupt function of the user is finished, the library automatically sets the ClearIRQ-Bit (CTRG) in the CMD-register to release the next interrupt. If the MeasureAll-function is called in your own interrupt function, you do not need to set the parameter bAutoClearIRQ.

| | | |
|---|---|--|
|  GEMAC <small>Gesellschaft für Mikroelektronikanwendung Chemnitz mbH</small> | Phone: +49 371 33 77 - 0 | Title: ZKPCI |
| | Fax: +49 371 33 77 272 | |
| Zwickauer Straße 227 D-09116 Chemnitz, Germany | Internet: www.gemac-chemnitz.de | Name of Document: 457-HB -1-4-E-ZKPCI |
| | Email: interpolation@gemac-chemnitz.de sales@gemac-chemnitz.de | |
| | Date: 24.03.04 | Page 19 of 21 |

7.3.12 CZKPCI::TriggerInterruptDisable

Function header:

DWORD CZKPCI::TriggerInterruptDisable(unsigned int uiCardNumber)

Arguments:

→ uiCardNumber: Card number in the system.
0 ≤ uiCardNumber < CardCount

The member function CKPCI : :TriggerInterruptDisable switches the trigger interrupt off.

7.3.13 CZKPCI::GetErrorDescription

Function header:

void CZKPCI::GetErrorDescription (DWORD dwErrorCode,
char *szErrDescription,
unsigned int uiErrDescriptionLength)

Arguments:

→ dwErrorCode: Error code
→ *szErrDescription: Pointer to character string as a buffer for the error description
→ uiErrDescriptionLength: Maximum character number of the character string

The member function CZKPCI::GetErrorDescription supplies a textual description of the error code (only for errors defined in ERRORCODES.H).



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 20 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI

8 Revision History

| <i>Date</i> | <i>Revision</i> | <i>modification</i> |
|-------------|--|---|
| 08-21-2002 | Hardware Revision 2 | First Revision |
| | DLL Revision 1.02.438 | First Revision |
| | Driver Revision 5.20 RC2 | First Revision |
| | Manual Revision 1.1 | First Revision |
| 04-04-2003 | Driver Revision 5.22 | Install-Script |
| | Manual Revision 1.2 | Chapter 6.1 / Page 11: Installation of driver |
| 07-23-2003 | Hardware Revision 3 | Trigger inputs: timing-bug fixed |
| | Manual Revision 1.3 | Chapter 7 / Page 10: Minimal edge distance A/B Chapter 8 / Page 21: new |
| 03-24-2004 | Driver Revision 6.20 Software Package V3 Manual Revision 1.4 | Install-Script Adapted to driver version 6.20 Chapter 4.1 / Page 9: X9: 9/10, 11/12 , 15/16 Chapter 5 / Page 10: supply current Chapter 6.1 / Page 11: Installation of driver |



Zwickauer Straße 227
D-09116 Chemnitz, Germany

Phone: +49 371 33 77 - 0
Fax: +49 371 33 77 272
Internet: www.gemac-chemnitz.de
Email: interpolation@gemac-chemnitz.de
sales@gemac-chemnitz.de

Date: 24.03.04

Page 21 of 21

Title:

ZKPCI

Name of Document:
457-HB -1-4-E-ZKPCI