

PMC-FIFO

2 * 16bit FIFO (4k, 16k oder kompatibel)

8bit TTL-Input mit IRQ

8bit TTL-Input statisch

8bit TTL-Output

Hardware-Manual

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. esd übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei esd. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch esd gestattet.

esd electronic system design gmbh

Vahrenwalder Str. 205

D-30165 Hannover

Tel.: 0511/37298-0

FAX: 0511/37298-68

E-Mail: info@esd.electronics.com

Internet: <http://www.esd-electronics.com>

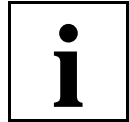
Beschriebene Platinenversion	PMCFIFO-10
------------------------------	------------

Änderungen nach Kapiteln:

Die hier aufgeführten Änderungen im Anwenderhandbuch betreffen sowohl Änderungen in der Hardware als auch reine Änderungen in der Beschreibung der Sachverhalte.

Kapitel	Änderungen
1.2, 3	Bezeichnung des I/O-Steckers P1 korrigiert.

Inhalt	Seite
1. Übersicht	3
1.1 Funktionsbeschreibung	3
1.2 Allgemeine Technische Daten	4
1.3 Technische Daten der I/O-Baugruppen	5
1.4 Bestellhinweise	6
2. FIFO-Speicher	7
3. Anhang	9
3.1 Steckerbelegungen	9
3.1.1 Belegung des 64poligen PMC-Steckers P11	9
3.1.2 Belegung des 64poligen PMC-Steckers P12	10
3.1.3 Belegung des 68poligen I/O-Steckers P1	11



1. Übersicht

1.1 Funktionsbeschreibung

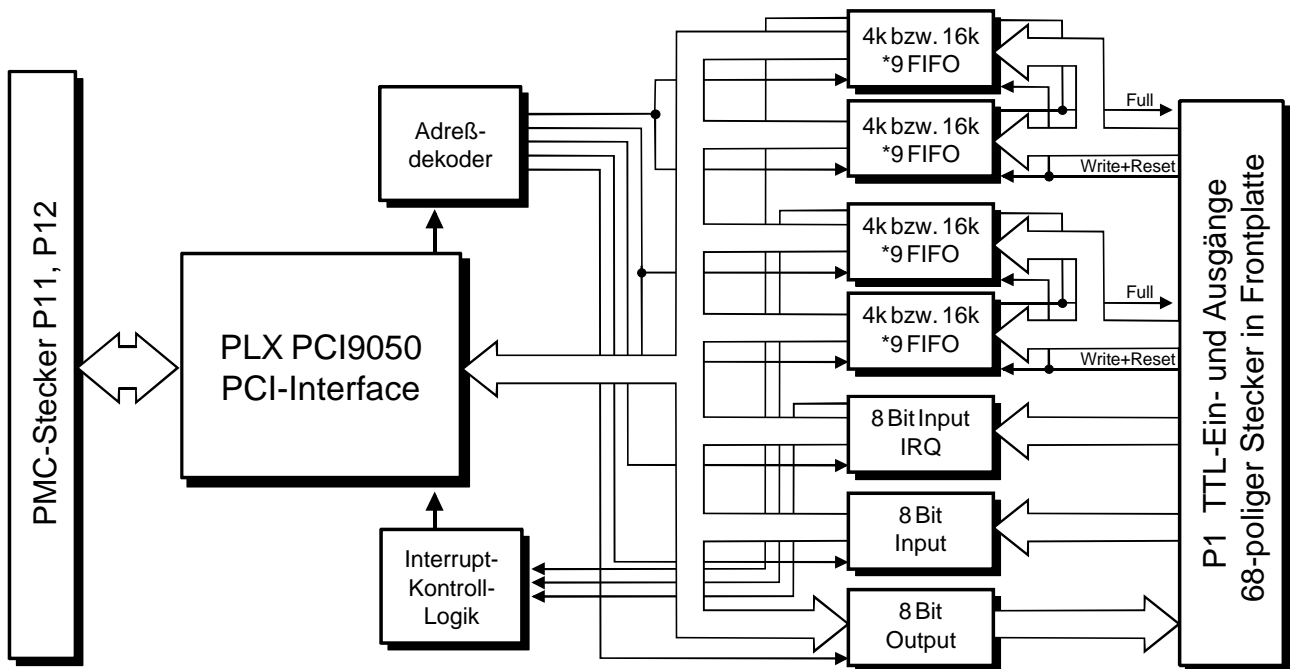


Abb. 1.1.1: Blockschaltbild der PMC-FIFO

Auf der PMC-FIFO befinden sich zwei FIFOs, die mit 4k, 16k oder kompatiblen Bausteinen bestückbar sind. Zum flexiblen Auslesen der FIFOs können diese Interrupts für die Zustände “Empty”, “Half-Full”, und “Full” generieren. Als zusätzliche Interrupt-Quelle ist ein 8-bit Latch vorhanden, das bei einem Interrupt ausgelesen werden kann. Ein weiterer 8-bit Treiber dient als TTL-Input und ein 8-bit Latch als TTL-Output.

Der Anschluß erfolgt über einen 68poligen Stecker mit Stiftkontakten in der Frontplatte.

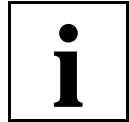
Nach Power-On oder bei Empfang des RESET-Signals werden alle FIFOs und Interrupts gelöscht und die 8 digitalen Ausgänge auf “high” gesetzt.



1.2 Allgemeine Technische Daten

Modul-Interface	<i>PMC</i>
Temperaturbereich	max. zulässige Umgebungstemperatur: 0...70°C
Luftfeuchtigkeit	max. 90%, nicht kondensierend
Steckverbinder	P11 - <i>PMC</i> 64pol P12 - <i>PMC</i> 64pol P1 - 68pol, Typ: Thomas&Betts: 311-068072E (TTL-I/O-Signale in Frontplatte) Notwendiges Gegenstück: Thomas&Betts: 311-068302
Größe der Platine	74 mm x 149 mm
Gewicht	ca. 100 g
Spannungsversorgung	über <i>PMC</i> : 5V / ca. 500mA

Tabelle 1.2.1: Allgemeine Daten der *PMC-FIFO*



1.3 Technische Daten der I/O-Baugruppen

Anzahl der FIFO-Eingänge	2 * 16 (jeweils entweder 4k oder 16k Tiefe) 2 * FIFO-Write, 2 * FIFO-Reset
Zeitverhalten der FIFO-Eingänge (siehe auch S. 7)	Data-Setup-Time: t_s \$ 20 ns Data-Hold-Time: t_H \$ 20 ns Data-Cycle-Time: t_p \$ 50 ns Burst-Read-Rate: # 132 MByte/s Non Burst-Read-Rate: # 4 MByte/s
Anzahl der FIFO-Ausgänge	2 * FIFO-Full-Flag
Anzahl der digitalen Eingänge mit Interrupt	8
Anzahl der digitalen Eingänge	8
Anzahl der digitalen Ausgänge	8
Pegel (alle I/Os)	TTL-Level
Receiver (IRQ-Inputs, Write-Signale der FIFOs und FIFO-Reset)	74LS14 (Schmitt-Trigger mit Hysterese)
Receiver (alle anderen Inputs)	74LS244
Transmitter (FIFO-Full-Flag)	74LS244
Transmitter (8bit Output)	74ALS573
Interrupts	- FIFOs bei "Empty", "Half-Full" und "Full" - 8 digitale Eingänge mit Interrupt (Flanke "high to low")

Tabelle 1.3.1: Technische Daten der I/O-Baugruppen



1.4 Bestellhinweise

Typ	Eigenschaften	Bestell-Nr.
PMC-FIFO	PMC-Modul mit zwei mal 16 Bit FIFO, 8 Bit Input (mit IRQ), 8 Bit Input, 8 Bit Output	V.2010.01
PMC-FIFO-OS9	Treiber als C-Source für OS9	V.2010.50
PMC-FIFO-VxWorks	Treiber als C-Source für VxWorks	V.2010.56
PMC-FIFO-RTOS-UH	Treiber für RTOS-UH	V.2010.54
PMC-FIFO-MD (*)	Anwenderhandbuch in deutsch	V.2010.20

(*) Wird das Handbuch gemeinsam mit der Karte bestellt, so wird es kostenlos mitgeliefert.

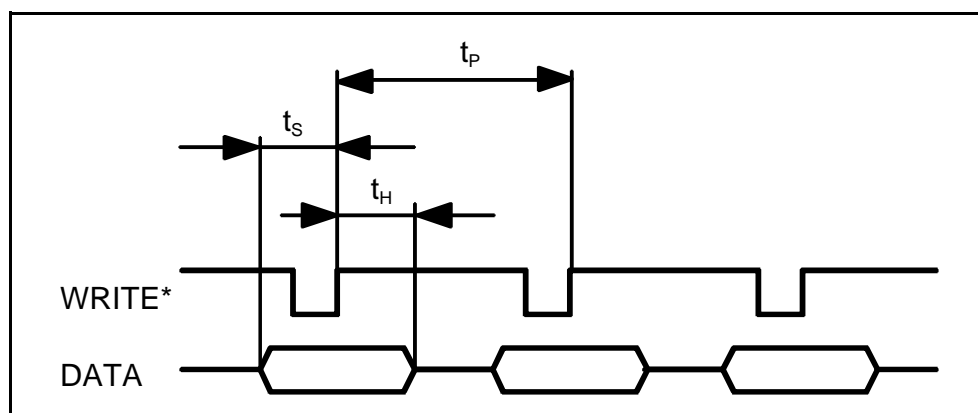
Tabelle 1.4.1: Bestellhinweise



2. FIFO-Speicher

Bei den eingesetzten FIFO-Speicherbausteinen handelt es sich um Cypress CY7C433 (4k Tiefe) bzw. CY7C462 (16k Tiefe) oder kompatible Bausteine.

Die folgende Abbildung zeigt das FIFO-Write-Timing für das Einlesen der Daten. Die Daten werden mit der steigenden Flanke des *WRITE**-Signals übernommen. Sie müssen mindestens 20 ns (Setup-Time t_s) vor der steigenden Flanke anliegen und mindestens weitere 20 ns (Hold-Time t_H) nach der steigenden Flanke des *WRITE**-Signals gehalten werden. Die minimale Periodendauer für das Einlesen der nächsten Daten beträgt 50 ns (t_p), wodurch sich eine Schreibfrequenz von maximal 20 MHz ergibt.



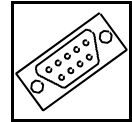
t_s \$ 20 ns

t_H \$ 20 ns

t_p \$ 50 ns

Abb. 2.1: FIFO-Write-Timing

Das Auslesen der FIFOs erfolgt bei Burst-Zugriffen (DMA-Controller) mit einer Datenrate von bis zu 132 MByte/s und ohne Burst-Zugriffe mit einer Datenrate von bis zu 4 MByte/s.



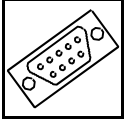
3. Anhang

3.1 Steckerbelegungen

3.1.1 Belegung des 64poligen PMC-Steckers P11

Pin #	Signal Name	Signal Name	Pin #
1	TCK	-12V	2
3	GND	INTA*	4
5	INTB*	INTC*	6
7	PMCPRSNT*	+5V	8
9	INTD*	-	10
11	GND	-	12
13	CLK	GND	14
15	GND	PMCGNT*	16
17	PMCREQ*	+5V	18
19	+5V	AD31	20
21	AD28	AD27	22
23	AD25	GND	24
25	GND	C/BE3*	26
27	AD22	AD21	28
29	AD19	+5V	30
31	+5V	AD17	32
33	FRAME*	GND	34
35	GND	IRDY*	36
37	DEVSEL*	+5V	38
39	GND	LOCK*	40
41	SDONE*	SBO*	42
43	PAR	GND	44
45	+5V	AD15	46
47	AD12	AD11	48
49	AD09	+5V	50
51	GND	C/BE0*	52
53	AD06	AD05	54
55	AD04	GND	56
57	+5V	AD01	58
59	AD02	AD01	60
61	AD00	+5V	62
63	GND	REQ64*	64

PMC SPECIFICATION IEEE1386.1/Draft 2.0 - 04-APR-1995



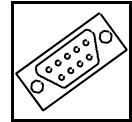
Steckerbelegung

3.1.2 Belegung des 64poligen PMC-Steckers P12

Pin #	Signal Name	Signal Name	Pin #
1	+12V	TRST*	2
3	TMS	TDO	4
5	TDI	GND	6
7	GND	-	8
9	-	-	10
11	-	-	12
13	RST*	-	14
15	-	-	16
17	-	GND	18
19	AD30	AD29	20
21	GND	AD26	22
23	AD24	-	24
25	IDSEL	AD23	26
27	-	AD20	28
29	AD18	GND	30
31	AD16	C/BE2*	32
33	GND	-	34
35	TRDY*	-	36
37	GND	STOP*	38
39	PERR*	GND	40
41	-	SERR*	42
43	C/BE1*	GND	44
45	AD14	AD13	46
47	GND	AD10	48
49	AD08	-	50
51	AD07	-	52
53	-	-	54
55	-	GND	56
57	-	-	58
59	GND	-	60
61	ACK64*	-	62
63	GND	-	64

PMC SPECIFICATION IEEE1386.1/Draft 2.0 - 04-APR-1995

- Dieser Pin ist auf dem Modul nicht belegt.



3.1.3 Belegung des 68poligen I/O-Steckers P1

Pin #	Signal Name	Signal Name	Pin #
1	F1D0	F2D12	35
2	F1D1	F2D13	36
3	F1D2	F2D14	37
4	F1D3	F2D15	38
5	F1D4	F2WRITE*	39
6	F1D5	F2RESET*	40
7	F1D6	F2FULL*	41
8	F1D7	GND	42
9	GND	IRQ0	43
10	F1D8	IRQ1	44
11	F1D9	IRQ2	45
12	F1D10	IRQ3	46
13	F1D11	IRQ4	47
14	F1D12	IRQ5	48
15	F1D13	IRQ6	49
16	F1D14	IRQ7	50
17	F1D15	GND	51
18	F1WRITE*	IN0	52
19	F1RESET*	IN1	53
20	F1FULL*	IN2	54
21	GND	IN3	55
22	F2D0	IN4	56
23	F2D1	IN5	57
24	F2D2	IN6	58
25	F2D3	IN7	59
26	F2D4	GND	60
27	F2D5	OUT0	61
28	F2D6	OUT1	62
29	F2D7	OUT2	63
30	GND	OUT3	64
31	F2D8	OUT4	65
32	F2D9	OUT5	66
33	F2D10	OUT6	67
34	F2D11	OUT7	68

Bauform des I/O-Steckers auf der Platine (Stiftkontakte):

Thomas&Betts: 311-068072E

Bauform des notwendigen Gegensteckers an der Leitung (Buchsenkontakte):

Thomas&Betts: 311-068302

PMC-FIFO

2 * 16bit FIFO (4k, 16k oder kompatibel)

8bit TTL-Input mit IRQ

8bit TTL-Input statisch

8bit TTL-Output

Software-Manual

Der Inhalt dieser Dokumentation wurde mit größter Sorgfalt erarbeitet und geprüft. esd übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei esd. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch esd gestattet.

esd electronic system design gmbh
Vahrenwalder Str. 205
D-30165 Hannover

Tel.: 0511/37298-0
FAX : 0511/37298-198
E-Mail: info@esd-electronics.com
Internet: <http://www.esd-electronics.com>

Beschriebene Software	Revision/Datum
PMC-FIFO C-Schnittstelle	05/05/99

Änderungen in der Software und/oder der Dokumentation

Änderung in diesem Handbuch gegenüber der Vorversion	Änderung in der Software	Änderung in der Dokumentation
Funktion pmcfifoIRQWait neu.	x	x

1. Programmierschnittstelle	3
1.1 Initialisierung	3
pmcfifoOpen()	3
pmcfifoClose()	3
1.2 Auslesen der FIFOs	4
pmcfifoFIFOConfig()	4
pmcfifoFIFOClear()	5
pmcfifoFIFOReset()	5
pmcfifoFIFORead()	5
pmcfifoFIFOStatus()	6
1.3 Interrupt Behandlung	7
pmcfifoIRQSignal()	7
pmcfifoIRQWait()	7
pmcfifoIRQRead()	8
1.4 Lesen und Schreiben von Daten	9
pmcfifoTTLRead()	9
pmcfifoTTLWrite()	9
2. Rückgabewerte	10

1. Programmierschnittstelle

Im nachfolgenden Kapitel wird die C-Programmierschnittstelle zur PMC-FIFO beschrieben. Die Bedeutung der Rückgabewerte im Fehlerfall ist in Kapitel 2 beschrieben.

1.1 Initialisierung

Die nachfolgend beschriebenen Dienste dienen der Initialisierung der PMC-FIFO-Hardware.

pmcfifoOpen()

Name: **pmcfifoOpen()** - Initialisieren der PMC-FIFO-Karte

Aufruf: `int pmcfifoOpen
(
)`

Beschreibung: Diese Funktion schaltet den Speicherbereich der PMC-FIFO frei und initialisiert die PCI-Bausteine auf der Host-CPU und der PMC-FIFO-Karte. Sie muß vor allen anderen Routinen zur Kommunikation mit der PMC-FIFO aufgerufen werden.

Rückgabe: 0 oder einen im Anhang beschriebene Fehlercode.

pmcfifoClose()

Name: **pmcfifoClose()** - Beenden der Kommunikation mit der PMC-FIFO-Karte

Aufruf: `int mcfifoClose
(
)`

Beschreibung: Diese Funktion schaltet alle angeschlossenen Interrupts ab und sperrt alle weiteren Zugriffe auf die PMC-FIFO.

Rückgabe: 0 oder einen im Anhang beschriebene Fehlercode.

1.2 Auslesen der FIFOs

Die nachfolgend beschriebenen Dienste dienen der Konfiguration und der Kommunikation mit den FIFO-Bausteinen auf der PMC-FIFO.

pmcfifoFIFOConfig()

Name: **pmcfifoFIFOConfig()** - Konfigurieren der FIFO-Speicher

Aufruf:

```
int    pmcfifoFIFOConfig
(
    int    i,           /* FIFO-Nummer (1 oder 2) */
    int    mode,       /* Modus (0 ... 2) */
    int    depth,      /* FIFO Tiefe (z.B. 4096) */
    void * buf         /* Zeiger auf freien Speicherbereich */
)
```

Beschreibung: Diese Funktion konfiguriert den FIFO-Speicher *i* als 8 (*mode=0*), 16 (*mode=1*) bzw. 32 (*mode=2*) bit Baustein. Es wird eine Interrupt-Service-Routine angeschlossen, die gewährleistet, daß die FIFOs beim Status "Half-Full" bzw. "Full" automatisch ausgelesen werden. Der Parameter *depth* enthält die Tiefe der eingesetzten FIFO-Bausteine. Dieser Wert muß übergeben werden, da die Bausteine ausgetauscht werden können. Zum Abspeichern der FIFO-Werte wird ein Zeiger auf einen ausreichend großen Speicherbereich *buf* übergeben. Das erste Langwort im Puffer *buf* enthält den Schreibzeiger, der anzeigt, wieviele Werte bisher aus dem FIFO ausgelesen wurden.

Im 8-bit-Modus werden die oberen 8 Bit ignoriert und lediglich die unteren 8 Bit abgespeichert.

Im 32-bit-Modus werden die beiden FIFO-Speicher (jeweils 16-bit) zu einem 32-bit Baustein zusammengefaßt und als 32-bit-Wert abgespeichert. Hierbei liefert FIFO-Nummer 1 die unteren 16 Bit und FIFO-Nummer 2 die oberen 16 Bit. Als FIFO-Nummer *i* muß 1 übergeben werden. FIFO-Nummer 2 darf im 32-bit-Modus nicht konfiguriert werden. Bei dieser Konfiguration dienen die FIFO-Steuersignale (WRITE*, RESET*) des ersten FIFOs als Steuersignale für beide FIFOs.

Der FIFO-Speicher wird zusätzlich gelöscht.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoFIFOClear()

Name: pmcfifoFIFOClear() - Löschen eines FIFO-Speichers

Aufruf:

```
int    pmcfifoFIFOClear
(
    int    i                /* FIFO-Nummer (1 oder 2) */
)
```

Beschreibung: Diese Funktion löscht einen der beiden FIFO-Speicherbausteine der PMC-FIFO.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoFIFOReset()

Name: pmcfifoFIFOReset() - Zurücksetzen eines FIFO-Speichers

Aufruf:

```
int    pmcfifoFIFOReset
(
    int    i,                /* FIFO-Nummer (1 oder 2) */
    int    mode              /* Modus (0 ... 2) */
)
```

Beschreibung: Diese Funktion löscht einen der beiden FIFO-Speicherbausteine der PMC-FIFO und setzt den FIFO erneut mit dem angegebenen Modus auf (siehe **pmcfifoFIFOConfig**).

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoFIFORead()

Name: pmcfifoFIFORead() - Vollständiges Auslesen eines FIFO-Speichers

Aufruf:

```
int    pmcfifoFIFORead
(
    int    i,                /* FIFO-Nummer (1 oder 2) */
    int    signal,          /* Signal-Code */
    int    proc_id         /* Process-ID */
)
```

Beschreibung: Diese Funktion liest den FIFO-Speicher *i* vollständig (bis der FIFO-Baustein "Empty" signalisiert) aus. Darauf wird das Signal *signal* durch den Aufruf *_os_send()* an den Prozeß *proc_id* gesendet.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoFIFOStatus()

Name: **pmcfifoFIFOStatus()** - Lesen des Status der FIFO-Speicher

Aufruf: `char pmcfifoFIFOStatus
(
)`

Beschreibung: Diese Funktion liest den Status der beiden FIFO-Speicher und liefert deren Wert als Rückgabeparameter.

Rückgabe: 8 Bit mit folgender Bedeutung:

Bit 0: FIFO 1 Empty	Bit 4: FIFO 2 Empty
Bit 1: FIFO 1 Half-Full	Bit 5: FIFO 2 Half-Full
Bit 2: FIFO 1 Full	Bit 6: FIFO 2 Full
Bit 3: Reserved	Bit 7: Reserved

1.3 Interrupt Behandlung

Die nachfolgend beschriebenen Dienste dienen der Konfiguration der interruptfähigen Eingänge.

pmcfifoIRQSignal()

Name: pmcfifoIRQSignal() - Konfiguration der interruptfähigen Eingänge (OS-9)

Aufruf:

```
int    pmcfifoIRQSignal
(
    int    mode,           /* Modus (0 oder 1) */
    int    bit,           /* Bit-Nummer (0 ... 7) */
    int    signal,        /* Signal-Code */
    int    proc_id        /* Process-ID */
)
```

Beschreibung: Diese Funktion verknüpft den interruptfähigen Eingang *bit* (0...7) mit *signal*.

Falls als *mode* 1 übergeben wird, so wird bei einem entsprechenden Interrupt ein *_os_send(proc_id, signal)* einmalig ausgeführt. Dadurch kann ein, auf dieses Signal wartender, Prozeß über einen Interrupt einmal "angestoßen" werden.

Falls als *mode* 2 übergeben wird, so ist die Verknüpfung nicht einmalig, sondern auf Dauer hergestellt. Diese Verknüpfung kann jedoch durch einem Aufruf mit *mode* 0 wieder aufgehoben werden.

Falls als *mode* 0 übergeben wird, so wird die Verknüpfung zwischen *bit* und *signal* wieder aufgehoben. Durch den angegebenen Interrupt *bit* wird nun kein Prozeß ausgeführt.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoIRQWait()

Name: pmcfifoIRQWait() - Warten auf einen Interrupt (VxWorks / RTOS-UH)

Aufruf:

```
int    pmcfifoIRQWait
(
    int    bit           /* Bit-Nummer (0 ... 7) */
)
```

Beschreibung: Nach Aufruf dieser Funktion wartet der aufrufende Prozeß, bis an dem angegebenen Interrupt-Eingang *bit* (0...7) ein Interrupt erzeugt wird.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

pmcfifoIRQRead()

Name: pmcfifoIRQRead() - Lesen der 8 interruptfähigen Eingänge

Aufruf: int pmcfifoIRQRead
(
)

Beschreibung: Diese Funktion liest die 8 IRQ-fähigen Eingänge und liefert deren Interrupt-Status als Rückgabeparameter. Hierbei entspricht eine 1 einem vorherigen "high-to-low" Pegelwechsel.

Rückgabe: Die 8 Bit zusammengefaßt als unteres Byte im Integer-Wert. Bit 0 entspricht Eingang Nummer 0 usw.

1.4 Lesen und Schreiben von Daten

Die nachfolgend beschriebenen Dienste dienen dem Auslesen der TTL-Eingänge bzw. dem Schreiben der TTL-Ausgänge.

pmcfifoTTLRead()

Name: `pmcfifoTTLRead()` - Lesen der 8 TTL-Eingänge

Aufruf:

```
int    pmcfifoTTLRead
(
)
```

Beschreibung: Diese Funktion liest die 8 TTL-Eingänge und liefert deren Wert als Rückgabeparameter. Hierbei entspricht eine 1 einem "high" und eine 0 einem "low" am Eingang.

Rückgabe: Die 8 Bit zusammengefaßt als unteres Byte im Integer-Wert. Bit 0 entspricht Eingang Nummer 0 usw.

pmcfifoTTLWrite()

Name: `pmcfifoTTLWrite()` - Schreiben eines der 8 TTL-Ausgänge

Aufruf:

```
int    pmcfifoTTLWrite
(
  int    bit_no,          /* Bit-Nummer (0 ... 7) */
  int    bit_count,      /* Bit-Zähler (1 ... 8) */
  int    value            /* Wert (0 oder 1) */
)
```

Beschreibung: Diese Funktion setzt einen oder mehrere (ab Bit-Nummer *bit_no* *bit_count*-Bits) der 8 TTL-Ausgänge auf den übergebenen Wert *value*. Hierbei entspricht eine 1 einem "high" und eine 0 einem "low" am Ausgang. Die anderen Ausgänge werden nicht verändert.

Rückgabe: 0 oder einen im Anhang beschriebenen Fehlercode.

