FEB Transientenrecorder

J. Bengtsson, G. Bialek, K. H. Meß, R. Pforte, M. Plintovic, L. Steffen
(Beratung und Spezifikation: R. Bacher)
Stand 26.4.99

A. Übersicht

Bei HERA ist eine Reihe von vorzüglichen und der jeweiligen Aufgabe angepaßten Transientenrecordersysteme im Einsatz. Da gibt es zum Beispiel die Post-Mortem Funktionen in der Strahllage- und Strahlverlust Auslese. Ein anderes Beispiel ist die Bakker Karte, von denen viele bei MHF und anderswo im Einsatz sind. Insgesamt gibt es wohl gegen 7 verschiedene Systeme, die natürlich weitgehend inkompatibel miteinander sind.
Die hier vorgestellte Lösung ist der Versuch gleichzeitig möglichst einheitlich und vielseitig HERA-spezifische Probleme zu lösen. Dazu wurden folgende Eckwerte von Herrn Bacher gefordert:

- Nahtlos zum Archivsystem kompatibel
- Unabhängigkeit vom Rechnertyp (PC, VME...)
- Abtastrate ≤ 100ks/s oder HERA-synchron (halber Umlauf)
- Bandbreite > 300kHz[1]
- Potentialtrennung >100V (tatsächlich ca. 500V, kabelbegrenzt)
- Speichertiefe von 8000 Worten
- Extern oder intern (Bitkombination) triggerbar.
- Weiterreichen von intern erzeugten Stopppulse auf andere Recorder und gegebenenfalls auf die Alarmloop
- Auflösung analoger Signale mit 12 Bit Genauigkeit bei 4 Eingangsspannungsbereichen

B. Aufbauvarianten

Da das System rechnerunabhängig sein sollte, wurde ein Industriestandard (industrial Package IP) zur Anknüpfung an Rechner gewählt. Trägerkarten für IP Module sind von Motorola (aktiv) für VME Umgebungen, Greenspring für VME und PC Umgebungen erhältlich. Darüber hinaus gibt es auch Feldbusanbindungen; hier sei besonders auf den General Purpose Fieldbus Controller (GPFC) von MKS2 verwiesen, für den es auch mit dem Archiv kompatible Software gibt.
Das IP Modul trägt nicht nur das Interface zum Rechner, sondern auch serielle Links zu 8 Frontends, im folgenden Tastköpfe genannt, einen programmierbaren Oszillator, 8 x 8kWorte Memory und eine Logikeinheit zum Erkennen einer einstellbaren Stoppbedingung. Die Verbindung zur Außenwelt wird über das IP Modul spezifische 50 polige Flachbandkabel hergestellt.
Dies Kabel endet immer auf einer DTU (Daten und Trigger Unit). Es gibt mehrere, funktionsgleiche Typen, angepaßt an die Anwendung. Alle DTUs erlauben das Synchronisieren mit einer externen Clock. Für den Spezialfall HERA ist zur umlaufsynchronen Abtastung auch ein Eingang für die HERA Bunch Clock vorgesehen. Diese Pulskettenfrequenz wird dann um den Faktor 110 herunter gesetzt. Die DTUs sind auch zur Verteilung der externen oder internen Stoppsignale zuständig und in Ihnen findet die Potentialtrennung statt.
Die Signalaufnehmer, die sogenannten Tastköpfe, werden normalerweise mit einem Kabel an die SubD Steckdosen angeschlossen. Auf diese Weise werden die Kapazitäten und

[1] Das ist kein Widerspruch zum Nyquist Theorem. Die Bandbreite wird benötigt, wenn vorher abgetastete Signale an das System weitergereicht werden.

Induktivitäten, die das Eingangssignal verändern können, und die Einstreuung fremder Signale möglichst klein gehalten. Abstände bis zu 40m lassen sich so überbrücken.
Alternativ lassen sich die Tastköpfe in die DTU einbauen. Das ist nützlich, wenn die zu überwachenden Signale bereits nahe bei der DTU vorhanden sind und/oder Einstreuungen vernachlässigbar sind.

C. Tastköpfe

Zur Zeit stehen drei Typen von Tastköpfen bereit.

1. Analogtastkopf

Der analoge Tastkopf hat einen quasi differentiellen Eingang, der mittels zweier Steckverbinder mit der Quelle verbunden wird. Sind die Tastköpfe in die DTU eingebaut, werden Lemo 00 Buchsen verwendet. Die folgende Tabelle faßt die Eigenschaften zusammen.

Max. Eingangsspannung (differentiell)	Kurzzeitig: 150 V Dauernd: 75 V
Einstellbare Spannungsbereiche	+/- 30 V +/- 10 V +/- 1 V +/- 0.1 V
Eingangswiderstand	100 kOhm
Bandbreite (einstellbar)	100 kHz 25 kHz 10 kHz 1 kHz
Isolationsspannung (abhängig vom verwendeten Anschlußkabel)	500 V
Interne Testspannung	Uref/3
ADC Referenzspannung (Uref)	2.460 V
Konversionsrate	100 kHz oder weniger
Auflösung	12 Bit (bipolar)

Abbildung 1 zeigt einen solchen Tastkopf. Das Gerät von der Größe einer Streichholzschachtel läßt sich leicht überall befestigen.

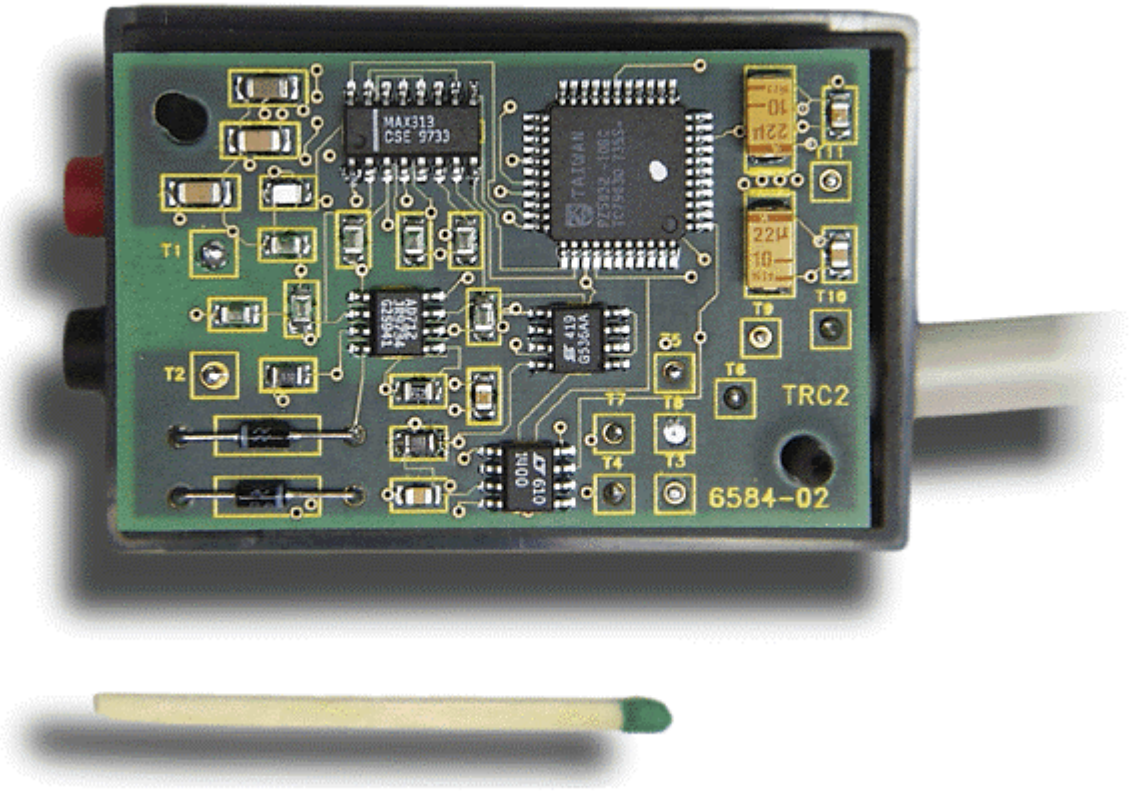
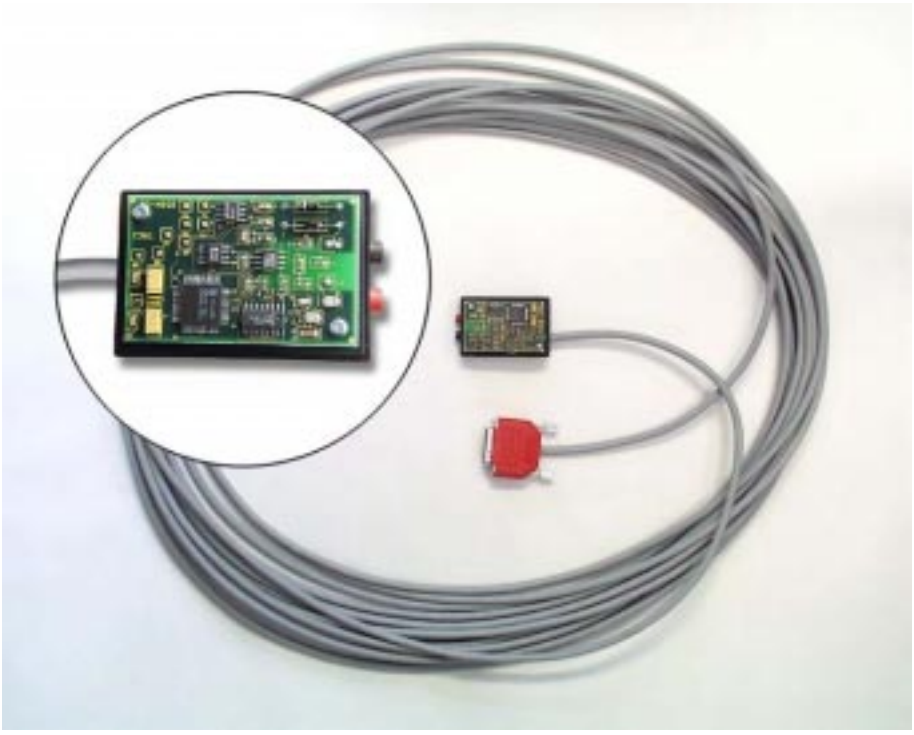


Abbildung 1) Analogtastkopf



Auf Abbildung 2 ist der Tastkopf mit Kabel dargestellt:

Abbildung 2) Analogtastkopf mit Anschlußkabel

2. Digitaltastopf

Äußerlich dem analogen Tastkopf sehr ähnlich ist der digitale Tastkopf. Er wird benutzt um den Zustand eines Bits zum Clockzeitpunkt festzuhalten oder um festzustellen ob sich das Bit seit dem letzten Sampling verändert hat. Spikes bis unter 100 ns sind meßbar. Die Kenndaten sind in der folgenden Tabelle zusammengefaßt.

Anzahl der Eingänge	12
Meßbereiche (einstellbar)	+24 V +5 V
Meßmodi: Sample Edge	 Pegel bei sampling Clock >= 1 Pegeländerung seit sampling Clock
Mindestimpulsbreite	100 ns
5 V Schwellen positiv gehende Flanke negativ gehende Flanke Hysterese	 1.3 ... 2.0 V 1.3 ... 0.55 V > 0.4 V
24 V Schwellen positiv gehende Flanke negativ gehende Flanke Hysterese	 6.6 ... 10.5 V 6.6 ... 2.55 V > 2.2 V
Eingangsimpedanz 5 V Betrieb 24 V Betrieb	 12 kOhm || 12 pF 12 kOhm || 68 pF
Bandbreite	> 10 MHz
Eingangsspannung	< | +/- 50V |
Isolationsspannung (kabelabhängig)	> |+/- 500V |

3. 3 Mode Counter

Der dritte, derzeit verfügbare Tastkopftyp, ist ein Zähler, der sogenannte 3 Mode-Counter .
Funktion
Der 3 Mode-Counter zählt die Eingangsimpulse, die zwischen 2 Clock Signalen, die das Zählintervall der Auslese bestimmen, eintreffen. Dazu kann jede Clock mit weniger als 100 kHz genommen werden. Zum Beispiel generiert das IP Modul alle Frequenzen 100/n kHz, wobei n=1... 255. Für die Messung der Verlustraten bei HERA ist es denkbar die Clock mit der halben Umlaufzeit des Strahls gleichzusetzen. Das sind 94.7 KHz entsprechend 10.56µs. Dafür wird die Bunchclock für 220 Bunche des gesamten Strahlumlauf = 21µs im DT-Modul durch 110 dividiert, um die halbe Umlaufzeit zu. Danach wird diese Clock an die IP -Karten weitergereicht, die daraus die Ausleseintervalle generieren. Innerhalb dieser 10.56µs können mit dem Counter, der eine Breite von 9Bit hat, somit 512 Counts max. gezählt werden.
Der Ausleseimpuls stoppt den Counter und lädt seine Daten in ein Schieberegister. Anschließend werden die 9Bit Counter-Daten und 4 Bit Status-Daten mit einer seriellen Clock von 2 MHz ausgelesen.
Danach wird der Counter gelöscht und wieder zur neuen Zählung freigegeben.
Dies nimmt einige Zeit (400ns) in Anspruch, in der keine Bunchzählung erfolgt. Dieses 400ns-Fenster jittert mit 100ns, bedingt durch die Synchronisation mit dem 10MHz Counter, der als Referenz eingebaut ist.

a) Eingänge

Der 3Mode-Counter hat 4 programmierbare Eingänge

Loss-Monitor-Puls. Pw = 25ns Ph = 2.8V an 90 Ohm
Digital-Pulse. TTL-Norm
NIM-Pulse. L= 0V H = - 0.7V an 50 Ohm
Referenz-Pulse von internem Quarz = 10 MHz

und 2 programmierbare Ausgänge (TTL) , die als Selektleitungen für die Verlustmonitor-Arbeitsbereiche dienen.

b) Anschlüsse

Es gibt folgende Anschlüsse:
Der Signalanschluß ist eine LEMO-Buchse 00
Der Anschluß speziell für die Steuerung der Verlustmonitore wird über eine 6-polige Mini-DIN-Buchse und über ein 6-poliges abgeschirmtes Kabel vorgenommen.

Der Anschluß zum Driver- und Trigger- Modul (DT- Mod.) geschieht über einen 12- poligen DIN- Stecker und einem 10- poligem Kabel jeweils 2- paarig verdrillt. Dieses Kabel führt sowohl die Read / Write- Daten und die serielle clock als auch die Stomversorgung.

c) Stomversorgung

Geliefert werden 3 Spannungen vom DT- Modul und zwar:

+ 9V - 9V und + 24V

Die + 24V werden im 3Mode-Counter nur zum Verlustmonitor durchgereicht, während die +/- 9V im 3Mode- Counter in +/- 5V gewandelt werden.

Folgende Ströme werden für den 3Mode- Counter mit angeschlossenem Verlustmonitor benötigt:

+ 9V / 300mA - 9V / 80mA + 24V / 7mA

d) Inbetriebnahme des 3Mode- Counters

Der obere Gehäusedeckel muß entfernt werden und mit einem Schraubendreher wird anschließend der Mode-Select-Switch auf
der Platine auf den gewünschten Meßmode eingestellt.

Stellung 1 = Verlustmonitor- Mode
Stellung 5 = Digital- Mode
Stellung 9 = NIM- Mode

Die verschiedenen Modi werden durch entsprechende LED's am Gehäuse angezeigt.

An dem per Mode-Select- Switch voreingestelltem Mode kann dann per Kommando zwischen Referenz = 10MHz und Signal- Eingang jeweils umgeschaltet werden.

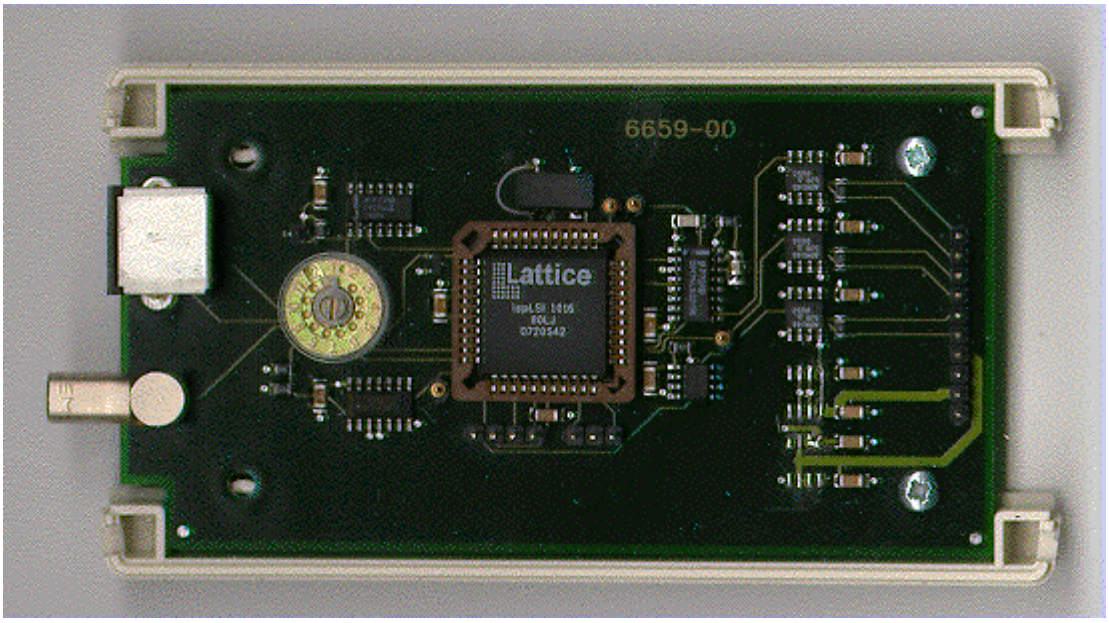


Abbildung 3) 3 Mode Counter

4. *Daten- und Trigger Einheit (DTU)*

Die Daten- und Triggerunit erfüllt drei Zwecke. Zum einen dient sie als Stromversorgung und Potentialtrenner. Intern ist die Potentialtrennung besser als 1000V. De facto wird sie bestimmt durch den eingebauten Sub D Stecker (Achtung Spezialversion 700 V), den verwendeten Kabelstecker (Achtung nur die mitgelieferten roten Stecker verwenden) und das verwendete Kabel. Zum zweiten werden in der DTU 8 Tastköpfe mit seriellen Daten versorgt und ausgelesen. Die Daten werden dann an das IP Modul weitergeleitet. Drittens kann hier die HERA Bunch Clock (wird intern um durch 110 geteilt) oder eine andere Abtastclock (TTL, 50% duty cycle) eingespeist werden. Ferner steht ein Eingang für ein externes Stoppsignal und ein Ausgang für das im IP Modul generierte Stoppsignal, bzw. das durchgeschleifte externe Stoppsignal, zur Verfügung.
Es gibt mehrere Varianten des DTU Moduls. Die normale Variante ist auf der Abbildung 4 gezeigt.

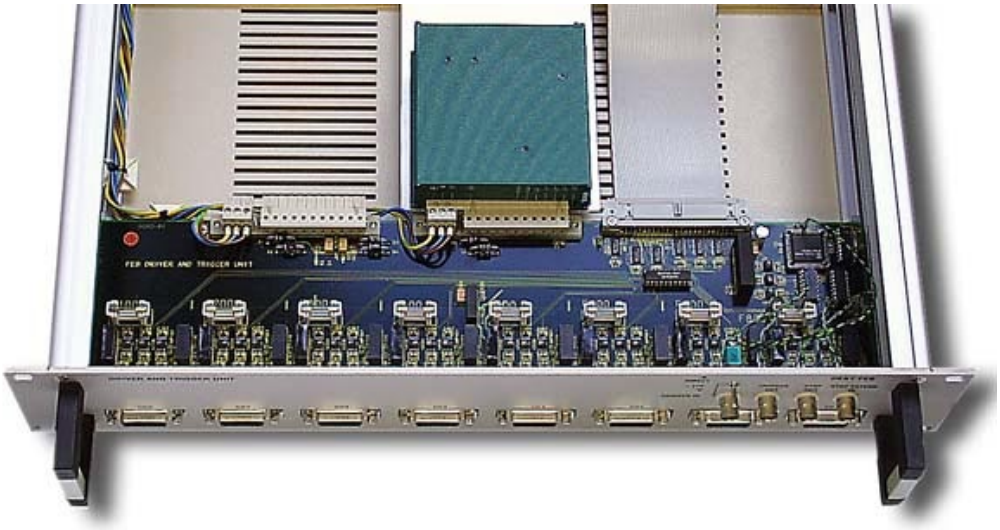


Abbildung 4) Normale DTU

Für MKK wurde eine verkürzte Variante gebaut. MHFp wünschte alle Tastköpfe in die DTU eingebaut. Dann ist die Isolationsfestigkeit durch die verwendeten isolierten Lemo Buchsen bestimmt. Für MKS2 wird eine Einschubvariante entwickelt, passend zu deren GPFC.

5. *IP Modul und Trägerkarten*

Es handelt sich bei den IP Modulen um eine Entwicklung von FEA.
Das industriekonforme Modul enthält die serielle Kommunikation, ein 8k x 8 Worte Memory und eine Logikeinheit. Das IP-Modul unterstützt alle Standard IP-Funktionen bis auf DMA. Die IP-Busfrequenz beträgt 8MHz. Jeder Kanal hat eine Speichergröße von 8192 Worten (1 Wort = 16 Bit).

Abbildung 5) IP Modul

Als Trägerkarte eignen sich die diversen Motorola VME Rechner mit IP Steckplätzen (Vertrieb EBV). Alternativ gibt es intelligente und einfache Trägerkarten von Greespring (Vertrieb EBV). Ähnliche Karten gibt es von Greenspring auch für PC (ISA und PCI). Für Feldbusanwendungen stellen mehrere Hersteller IP Träger bereit. Die Gruppe MKS2 hat einen eigenen Controller entwickelt, für den es auch Softwareunterstützung gibt.

a) Betriebsarten

Die IP-Module befinden sich in einem der folgenden Betriebsmodi.

(1) Data taking

Das Stopp Signal ist nicht aktiv (logisch L). Ein Clockpuls startet einen Lesezyklus aller 8 Kanäle. Die Clockquelle ist entweder intern oder extern. Der interne Clock erfolgt alle 0.5µs, das ergibt eine Abtastfrequenz von 95238.095 Hz. Die seriellen Daten kommen von der Frontend-Elektronik mit einer Taktrate von 2MHz, werden in parallele Daten konvertiert und im Speicher abgelegt. Der Datenspeicher ist als Ringspeicherstruktur ausgelegt. Sobald das Ende des Speichers erreicht ist, werden die ältesten Daten überschrieben. In diesem Betriebsmodus kann der Datenspeicher nicht ausgelesen werden. Allerdings kann man auf einen Buffer zugreifen und somit den zuletzt gemessenen Wert auslesen. Die Software kann Clock und Stoppsignale sperren und freigeben sowie den Betriebsmodus wechseln.

(2) Stop Transition

Vorausgesetzt das Modul befindet sich in normalem Data taking Modus und ein Stoppsignal trifft ein oder die Stoppbedingung wird erfüllt, dann nimmt das IP-Modul noch eine vordefinierte Anzahl von Meßwerten. Diese Zahl muß vor der Messung im Software Control Betriebsmodus in das Register cy_post_reg geschrieben werden. Am Ausgangspin ioSTOP_SYN wird der Stoppzustand der Außenwelt mitgeteilt. Damit kann man mehrere IP-Module miteinander synchronisieren. Wenn die Messung beendet ist wird ein Interrupt erzeugt (IP Interrupt 0). Die Stoppbedingung, die bereits erwähnt wurde, ergibt sich aus der Verknüpfung der Meßdaten mit den Inhalten mehrerer Register. Zur Anwendung kommt folgende Formel:

condition = ((data[i] AND mask[i]) XOR xor[i]) operator[i] level[i]

wobei

i	Nummer des Kanals
data[]	Daten vom Tastkopf
mask[]	Inhalt des 16 Bit Registers mask
xor[]	Inhalt des 16 Bit Registers xor
operator[]	= < > <= >= !=, im 16 Bit Register config
level[]	Inhalt des 16 Bit Registers level

(3) Data read-out

Der Datenspeicher kann von der Software ausgelesen werden. Im 16 Bit Register rx_address steht die Nummer (0-8191) des letzten Meßwertes.

Software Control

In diesem Betriebsmodus lassen sich die meisten Einstellungen machen. Die Tastköpfe können konfiguriert werden. Ein einzelner Lesezyklus kann ausgelöst werden. Die letzten beiden Eigenschaften sind mit einem gemeinsamen Interrupt verbunden (IP Interrupt 1).

Der Abtastzeitpunkt der seriellen Eingangsdaten kann relativ zur ioCLK verschoben werden um Zeitverzögerungen zu kompensieren die durch unterschiedliche Kabellängen der Tastköpfe hervorgerufen werden. Damit das funktioniert, muß die Frontend-Elektronik in der Lage sein, genau definierte Pulsketten zum IP-Modul zu senden. Nach einem Reset befindet sich das IP-Modul in folgendem Zustand:

- Die Timing Eingänge sind gesperrt.
- Die Anzahl der Lesezyklen ist auf 0 gesetzt.
- Die Interrupts sind gesperrt.
- Die Pointer zeigen auf den Speicheranfang, aber der Speicherinhalt ist nicht gelöscht.

Jeder Initialisierungszyklus wird erkannt. Alle IO Übertragungszyklen erfolgen so schnell wie möglich, bis auf das Lesen der letzten Meßdaten (rx_dio_sel), welches bis zu 9.5µs dauern kann. Das IP-Modul verwendet einen wait state auf alle Speicherzugriffe

b) Programmierung

(1) Übersicht: IO Adressen

Zugriff	Größe	Adresse						Erforderlicher Betriebsmodus	Registername
		A6	A5	A4	A3	A2	A1		
write	16Bit	0	1	0	0	0	0	SoftwareControl	rx_Clock
read	16Bit	0	0	1	C	C	C	beliebig	rx_dio_sel(CCC)
write	16Bit	0	0	1	C	C	C	SoftwareControl	tx_write(CCC)
read/write	8Bit	0	0	0	0	1	0	beliebig	control_word
write	8Bit	0	0	0	0	0	0	beliebig	intrpt_vec0
write	8Bit	0	0	0	0	0	1	beliebig	intrpt_vec1
read	16Bit	0	0	0	0	1	1	beliebig	rx_address
read	8Bit	0	0	0	1	0	0	beliebig	status
write	16Bit	0	1	1	0	0	0	DataTaking	cy_sw_stop
write	16Bit	0	1	1	0	0	1	SoftwareControl	cy_post_reg
write	16Bit	0	1	1	0	1	0	SoftwareControl	clock_shift
write	16Bit	1	0	0	C	C	C	SoftwareControl	mask(CCC)
write	16Bit	1	0	1	C	C	C	SoftwareControl	level(CCC)
write	16Bit	1	1	0	C	C	C	SoftwareControl	xor(CCC)
write	16Bit	1	1	1	C	C	C	SoftwareControl	config(CCC)

(2) Register:

(Hier läuft aus historischen Gründen die Nomenklatur etwas Amok, Trigger und Clock werden synonym benutzt.)

(a) rx_Clock:

Startet einen einzelnen Lesezyklus im Betriebsmodus Software Control, ähnlich einem Puls am Eingang ioCLOCK. Das ist aber nur möglich, wenn die Clock freigegeben ist. Dazu muß im control_word das Bit 5 (Clock enable) auf 1 gesetzt werden. rx_Clock ist in erster Linie für Hardwaretests während der Inbetriebnahme oder Wartung gedacht. Für den eigentlichen Meßbetrieb wird diese Funktion nicht benötigt.

(b) rx_dio_sel (CCC):

Liest den letzten Meßwert des Kanals CCC aus einem Zwischenspeicher.

(c) tx_write (CCC):

Sendet 16 Bit zur Frontend-Elektronik des Kanals CCC. Diese Funktion dient der Konfiguration der Tastköpfe. Die Bedeutung der 16 Bit ist von der Art des Tastkopfes abhängig. Außerdem sollte man daran denken, daß die Datenübertragung seriell stattfindet. Es kommt nicht so sehr darauf an, daß auch wirklich 16 Bit gesendet werden. Es können durchaus weniger sein. Aber die Reihenfolge ist sehr wichtig. Am besten geht man so vor, daß man die Konfigurationsdaten in eine 16 Bit Integervariable schreibt und soweit nach links verschiebt, daß keine unbenutzten Bits mehr links von den Konfigurationsdaten stehen.

(d) control_word:

Dieses Register dient der Steuerung des IP-Modul. Die einzelnen Bits haben folgende Bedeutung:

Bit	Bedeutung
D7	mode (1)
D6	mode (0)
D5	Clock enable (1)
D4	stop enable (1)
D3	interrupt0 enable (1) post cycles
D2	interrupt1 enable (1) tx & rx
D1	Clock source (1=ext, 0=int)
D0	io_stop_syn enable (1)

(e) mode

Hiermit wird der Betriebsmodus vorgewählt und so müssen die Bits gesetzt werden:

mode 1	mode 0	Betriebsmodus
1	1	datataking(DT)
1	0	stoptransition(ST)
0	1	dataread-out(DR)
0	0	softwarecontrol(SW)

Für einen reibungslosen Betrieb ist zu beachten, daß nicht beliebig von einem Betriebsmodus in einen anderen gewechselt werden kann. Nur folgende Wechsel sind erlaubt:

- data taking (DT) - data read-out (DR)
- data read-out (DR) - software control (SW)
- software control (SW) - data taking (DT) Für einen reibungslosen Betrieb ist es ratsam diesen Wechsel des Betriebsmodus in zwei Schritten durchzuführen.
  1. software control (SW) - stop transition (ST)
  2. stop transition (ST) - data taking (DT)
- software control (SW) - data read-out (DR)

(f) Clock enable

0 - IP-Modul reagiert nicht auf Clockpulse
1 - IP-Modul regiert auf Clockpulse

Dieses Bit wirkt auf alle Clockarten.

- interner Clock
- externer Clock am Pin io_CLOCK
- EinzelClock durch Schreibzugriff auf das Register rx_Clock

stop enable

0 - IP-Modul reagiert nicht auf Stopsignale
1 - IP-Modul reagiert auf Stopsignale

Dieses Bit wirkt auf alle Arten des Stoppsignals.

- externes Stoppsignal am Pin ioSTOP
- interne Stoppbedingung die aus den Meßwerten generiert wird.
- Softwarestop durch Schreibzugriff auf das Register cy_sw_stop.

(g) interrupt0 enable

0 - Interrupt 0 wird nicht erzeugt
1 - Interrupt 0 wird erzeugt, wenn die Erfassung der Meßwerte beendet wurde, weil eine Stoppbedingung eingetreten ist und die Anzahl der Meßwerte nach dem Stoppsignal, die im Register cy_post_reg steht, aufgezeichnet wurde.

(h) interrupt1 enable

0 - Interrupt 1 wird nicht erzeugt
1 - Interrupt 1 wird erzeugt,
wenn nach einem Schreibzugriff auf das Register rx_Clock das IP-Modul einen Meßwert komplett eingelesen hat und der serielle Empfänger für den nächsten Meßwert bereit ist. Alternativ könnte man auch das Statusregister auslesen und prüfen ob das Bit D5 (rx_rdy) gesetzt ist.
nach einem Schreibzugriff auf das Register tx_write das IP-Modul die Konfigurationsdaten an einen Tastkopf gesendet hat und der serielle Sender bereit ist, den nächsten Tastkopf zu

konfigurieren. Alternativ könnte man auch das Statusregister auslesen und prüfen ob das Bit D4 (tx_rdy) gesetzt ist.

(i) Clock source

0 - interner Clock
1 - externer Clock

(j) io_stop_syn enable

0 - IP-Modul gibt keine Stoppsignale am Pin ioSTOP_SYN aus.
1 - IP-Modul gibt Stoppsignale am Pin ioSTOP_SYN aus.

Beim Auslesen dieses Registers erhält man nur die Daten zurückgeliefert die man selber geschrieben hat. Eine Reaktion des IP-Moduls läßt sich nicht daraus ableiten. Wenn man Software entwickelt, die mit dem IP-Modul zusammen arbeiten soll, kann man das control_word Register gut für die ersten Schritte benutzen, weil es das einzige Register ist, dessen Inhalt man schreiben und lesen kann.

(k) intrpt_vec0:

(l) intrpt_vec1:

Vor der Aufnahme des Meßbetriebs muß auf diese beiden Register ein Schreibzugriff erfolgen. Dadurch wird die Interruptunterstützung des IP-Moduls initialisiert. Der Wert der geschrieben wird, stellt den Interruptvektor dar. Diesem Interruptvektor kommt allerdings nur eine Bedeutung zu, wenn das IP-Modul auf einen VME-Rechner gesteckt wird. Setzt man das IP-Modul mittels IP-Carrier in einem PC ein, dann funktioniert die Interruptbehandlung nach einem anderen Prinzip. Der Wert, der in die Register geschrieben wird, ist dann ohne Belang.

(m) rx_address:

Enthält die Nummer des zuletzt gespeicherten Meßwertes (0 - 8191).

(n) status:

Zeigt den augenblicklichen Betriebszustand des IP-Moduls

Bit	Bedeutung	Zustand nach einem Reset
D7	mode(0)	0
D6	mode(1)	0
D5	rx_rdy receiver ready(1)	1
D4	tx_rdy transmitter ready(1)	1
D3	IoSTOP input	0
D2	IoCLOCK input	0
D1		0
D0		0

(o) cy_sw_stop:

Stoppt die Aufzeichnung von Meßwerten. Dieses Register ist gleichbedeutend wie ein Puls am Eingang ioSTOP

(p) cy_post_reg:

Anzahl der Meßwerte, die nach dem Eintreten der Stoppbedingung noch aufgezeichnet werden sollen.

(q) clockshift:

Zeitverzögerung

Bit	Shift Bit	Kanal
D15	sh1	7
D14	sh0	7
D13	sh1	6
D12	sh0	6
D11	sh1	5
D10	sh0	5
D9	sh1	4
D8	sh0	4
D7	sh1	3
D6	sh0	3
D5	sh1	2
D4	sh0	2
D3	sh1	1
D2	sh0	1
D1	sh1	0
D0	sh0	0

Die Bedeutung der Shift Bits geht aus folgender Tabelle hervor:

Shift Bit		Zeitverzögerung
sh1	sh0	
0	0	0 ns
0	1	156 ns
1	0	312 ns
1	1	467 ns

(r) mask (CCC):

Parameter für die interne Stoppbedingung.

(s) level (CCC):

Parameter für die interne Stoppbedingung.

(t) xor (CCC):

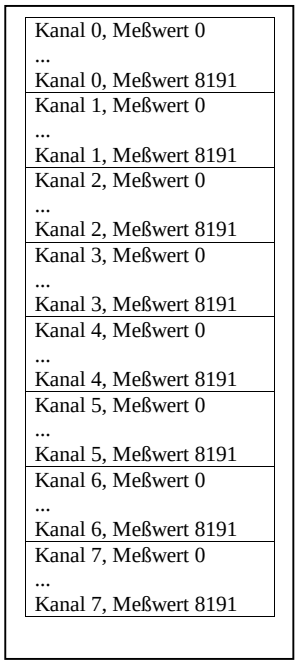
Parameter für die interne Stoppbedingung.

(u) config (CCC):

Parameter für die interne Stoppbedingung.

Wert	Bedeutung
0	=
1	<
2	>
3	>=
4	<=
5	!=
6	disable
7	disable

(3) Zugriff auf den Datenspeicher



Die Meßwerte aller Kanäle stehen hintereinander im Speicher. Als erstes kommen die 8192 Meßwerte des Kanals 0 und als letztes die 8192 Meßwerte des Kanals 7. Zum Auslesen der Werte benötigt man die Adressen der Speicherplätze. Die Ermittlung setzt etwas Rechnerei voraus. Kompliziert wird die Sache dadurch, daß die IP-Module eine wortweise Organisation des Speichers besitzen, aber einige Rechnersysteme eine byteweise Adressierung praktizieren. Byteadressen und Wortadressen dürfen nicht durcheinander gebracht werden. Angenommen, das IP-Modul hat die Datennahme beendet und steht im Betriebsmodus data read-out DR. Nun muß der Speicher ausgelesen werden. Wir lesen das Register rx_address aus und erfahren, welcher Meßwert zuletzt geschrieben wurde. Das Register rx_address enthält einen Wert im Bereich von 0 bis 8191, also gewissermaßen einen Offset in wortweiser Adressierung, man könnte auch Meßwertnummer dazu sagen. Um die Adresse innerhalb des IP-Modulspeichers zu erhalten, müssen wir die Anzahl der Meßwerte pro Kanal (8192) multipliziert mit der Kanalnummer (0 bis 7) hinzu addieren.

IPModulMemory(16 Bit) =
Meßwertnummer + Anzahl der Meßwerte pro Kanal * Kanalnummer

Besitzt das System das die IP-Module steuert eine byteweise Adressierung dann muß dieser Wert mit 2 multipliziert werden. Dieser Fall ist z.B. gegeben wenn das IP-Modul mittels IP-Carrier Steckkarte in einem Standard PC eingesetzt wird.

IPModulMemory(8 Bit) =
2 * (Meßwertnummer + Anzahl der Meßwerte pro Kanal * Kanalnummer)

Hat man die richtige Adresse ermittelt und ausgelesen, dann müssen die 16Bit Daten noch in ein Standarddatenformat umgewandelt werden. Die AD-Wandler liefern nur 12Bit-Daten

inklusive Vorzeichenbit. Also enthalten 4 der 16Bit keine Information. Außerdem steht das Vorzeichenbit an der falschen Stelle. Das folgende Bild zeigt den Zustand einer 16Bit-Variablen (signed short integer) nach dem Auslesen des IP-Moduls. Die relevanten Daten befinden sich in den Bits 2 bis 13. Der Zustand der Bits 0,1,14 und 15 ist undefiniert.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	x	x

Vorzeichen

Der Inhalt der 16Bit-Variablen muß um zwei Bit nach rechts verschoben werden. Die Daten befinden sich dann in den Bits 0 bis 11. Die Bits 12 bis 15 müssen auf den gleichen Wert gesetzt werden wie das Vorzeichenbit 11. Wenn das geschehen ist sieht die 16Bit-Variable so aus.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D11	D11	D11	D11	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Vorzeichen

Die Daten entsprechen jetzt einem Standarddatenformat (signed short integer) und sind fertig für die Ablieferung an das Archivsystem.

c) Anschlüsse

Alle Ein- und Ausgangssignale sind mit RS422-Treiber bestückt. Die Pulsbreite der externen Clock liegt zwischen 500ns und 5µs.

output		input	
Pin	Signal	Pin	Signal
1	ground	26	ground
2	-	27	-
3	ioCLK_	28	ioCLOCK_
4	ioCLK	29	ioCLOCK
5	ioSTOP_SYN_	30	ioSTOP_
6	ioSTOP_SYN	31	ioSTOP
7	ioDtoFE_(0)	32	ioDtoIP_(0)
8	ioDtoFE(0)	33	ioDtoIP(0)
9	ioDtoFE_(1)	34	ioDtoIP_(1)
10	ioDtoFE(1)	35	ioDtoIP(1)
11	ioDtoFE_(2)	36	ioDtoIP_(2)
12	ioDtoFE(2)	37	ioDtoIP(2)
13	ioDtoFE_(3)	38	ioDtoIP_(3)
14	ioDtoFE(3)	39	ioDtoIP(3)
15	ioDtoFE_(4)	40	ioDtoIP_(4)
16	ioDtoFE(4)	41	ioDtoIP(4)
17	ioDtoFE_(5)	42	ioDtoIP_(5)
18	ioDtoFE(5)	43	ioDtoIP(5)

19	ioDtoFE_(6)	44	ioDtoIP_(6)
20	ioDtoFE(6)	45	ioDtoIP(6)
21	ioDtoFE_(7)	46	ioDtoIP_(7)
22	ioDtoFE(7)	47	ioDtoIP(7)
23	-	48	-
24	-	49	-
25	ground	50	ground

D. Software

Für LINUX Rechner stellt FEB ein Auslesesystem mit Anbindung an das PKTR Archivsystem zur Verfügung. Anfragen bei Herrn L. Steffen. Für die Feldbuslösung gibt es Unterstützung bei MKS2. Herr Zimmer (FEE) und Herr Plintovic (FEA) helfen bei der Implementierung eines VME basierten Systems. (Für Windows NT hat sich noch keiner gefunden.)

E. Weiterentwicklungen

Neben eventuell anfallenden Spezialadaptierungen gibt es zwei weitere Entwicklungen, die auf dem TRC2 basieren. Zum einen wird an einem langsameren System mit mehr Kanälen zur Temperaturüberwachung der Synchrotronstrahlabsorber (Lumi Upgrade) gearbeitet. Hier gilt es hauptsächlich bei Beibehaltung der Zuverlässigkeit, die Kosten pro Kanal weiter zu drücken. Das System kann abschalten bei individuell einstellbaren Temperaturschwellen. Ehrgeiziger ist das zweite Projekt für MKK. Die Senderstromerzeugung soll überwacht werden. Dazu ist bei gleicher Speichertiefe eine Abtastrate von 1 MHz gewünscht. Ferner muß ein Kanal von vier eine Isolationsspannung von mehr als 15 kV (angestrebt wird mehr) erreichen. Die anderen drei Kanäle sind im Potential nahe (100 V) am Schutzleiter.
Beide Systeme sind selbstverständlich in der Software so weit möglich kompatibel.