

<h1>A337</h1> <h2>Trolley NMR-Meßsystem</h2>

1. FUNKTION	2
1.1. DATENBLATT	2
1.1.1. Anwendung	2
1.1.2. Daten	2
1.1.3. Besonderheiten	2
1.1.4. Aufbau	2
1.1.5. Stromversorgung (MP42, M26-13, A296)	2
1.2. BLOCKDIAGRAMM	3
1.3. BESCHREIBUNG	4
2. BETRIEB	6
2.1. BETRIEBSSYSTEM	6
2.1.1. MONITOR	6
2.1.2. SYSTEM	8
2.2. TROLLEY	9
2.3. DOWNLOAD	13
2.3.1. Download über S-Records:	13
2.3.2. Download in den Heap:	13
2.3.3. Abspeichern im Flash-Memory:	14
2.4. BEISPIELE	14
2.4.1. Einschalten und Start von TROLLEY	14
2.4.2. Ein- und Ausschalten des Echo	14
2.4.3. Öffnen einer Task	14
2.4.4. Einstellen einer anderen Baudrate	14
2.4.5. Download (S-Records) einer neuen "TROLLEY" Version	15
2.4.6. Löschen einer Task	15
2.4.7. Neuanlegen des Flash-Memories	15
3. FERTIGUNG	16
3.1. JUMPER	16
3.2. CLOCK	16

1. FUNKTION

1.1. Datenblatt

1.1.1. Anwendung

Mobiles Magnetfeldmess-System (NMR) im G-2 Experiment.

1.1.2. Daten

Parameter	Wert	Dimension
Messauflösung für heruntergemischte Frequenz	4	ppm
Verlustleistung im Power-Down-Modus	300	mW
Datenkommunikation RS232	max. 38400	baud

1.1.3. Besonderheiten

- Kleine Abmessung: 6 * 11* 4 cm.
- Weitgehende Verwendung nichtmagnetischer Bauteile.
- Geringe Verlustleistung (P< 1W) im Mittel bei Verwendung von Power-Down.
- Flexibler Programmwechsel durch Flash-Bausteine möglich (Einfaches Laden mit S-Records).
- Standard RS232 Kommunikation.
- Positionssensorik (Dual) mit Synchronisation an absoluten Code-Markierungen (Bar Codes).
- Energieversorgung, Referenzfrequenz-Zuführung, Kommunikation über ein Koaxial-Kabel.

1.1.4. Aufbau

A296: HF-Platine mit Senderumschalter, Preamplifier, RF-Mixer, Filter, Envelope Detector.

MP26-13: IF mit 5V, -5V, 15V Ladungspumpen, PowerSwitch, 8 Kanal * 8Bit ADC, Drucksensor, Temperatursensor, Positionssensorik, 6 Bit NMR-Precounter, Komparatoren für ClockRef, FID, Envelope.

MP42: Controller mit MC68332, 64KB RAM, 128KB Flash-EEPROM

1.1.5. Stromversorgung (MP42, M26-13, A296)

Spannung	Strom	Leistung
+6V (normal)	230mA	1,3W
+6V (Power on)	330mA	2W
+6V (LowPower)	50mA	300mW
Gesamt		

1.2. Blockdiagramm

Das Blockschaltbild der Digital-Elektronik (Bild 1) zeigt die zentrale Stellung des Microcontrollers mit seinem Speicher (ROM, RAM).

Die Verbindung nach außen erfordert lediglich ein 50 Ohm-Koaxkabel. Über eine Filterbaugruppe wird hier die Stromversorgung, die Zuführung der ca. 60 Mhz-Referenzfrequenz sowie eine Voll-Duplex-Kommunikation für die serielle Schnittstelle (max. 38400 Baud) realisiert.

Verschiedene Digital-Ausgänge am Controller dienen zur Auswahl der NMR-Sensoren, zur kompletten Steuerung des Messablaufs sowie zur Erzeugung verschiedener Schwellenspannungen (PWM).

Die Eingänge am Controller ermöglichen die Auswertung der Signale zur Temperatur-, Druck- und Positionsmessung und der Verarbeitung der NMR-Frequenz. Durch die Verwendung 'intelligenter' Sensoren (z.B. mit integrierten ADCs) ist ein direkter digitaler Anschluß möglich, lediglich für das Auszählen der NMR-Frequenz wird eine kleine Zusatzschaltung benötigt.

Für weitere Analogsignale ist ein zusätzlicher 8-Kanal, 8-Bit ADC vorgesehen. Hier kann z.B. auch das FID-Signal direkt digitalisiert werden.

NMR - TROLLEYCONTROLLER

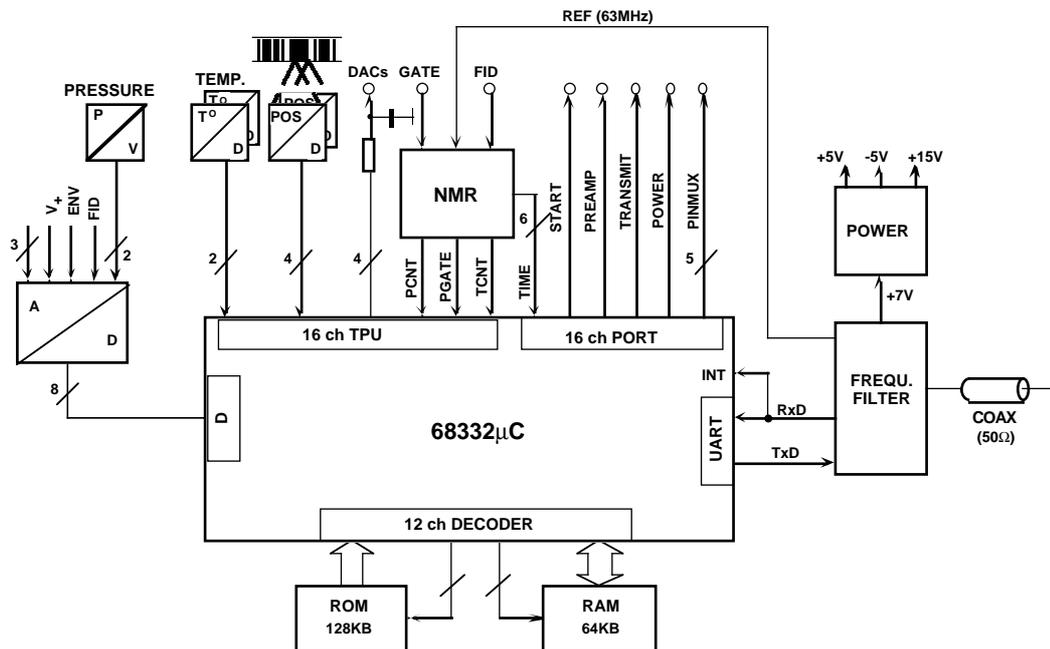


Bild 1: Trolley-Controller

1.3. Beschreibung

Verschiedene Signale (Power, Transmit, Preamp, Gate bzw. TimeOut) mit entsprechenden zeitlichen Abhängigkeiten zur Steuerung des Messzyklus werden direkt vom Controller erzeugt (Bild 2).

Die Signale zur Auswahl der gewählten NMR-Probe werden ebenfalls an diesem Port erzeugt.

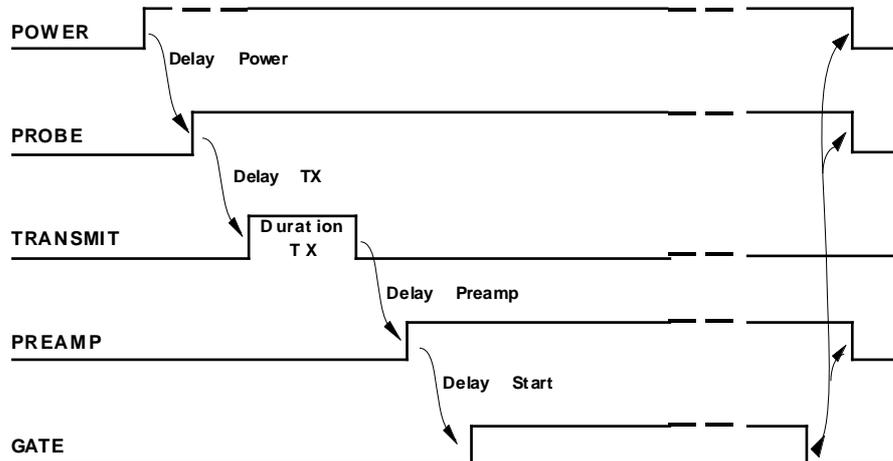


Bild 2: NMR-Ablaufsteuerung

Die Bestimmung der (heruntergemischten) NMR-Frequenz im Bereich von 10...100kHz wird in der Weise durchgeführt, daß für die Dauer eines externen Gates die Anzahl der Perioden gezählt und die dazugehörige Zeitdauer ausgezählt wird. Die Frequenz ergibt sich dann durch den Quotienten.

Das Gate bestimmt die Dauer der Messung und wird in einem Komparator mit einer Schwelle aus der Einhüllenden des NMR-Signals gebildet. Dieses Gate wird in der Schaltung (Bild 3) mit dem digitalisierten NMR-Signal (FID) synchronisiert und liefert so flankengenaue Gatesignale für die Perioden (PRDS) und die Messzeit (TIME).

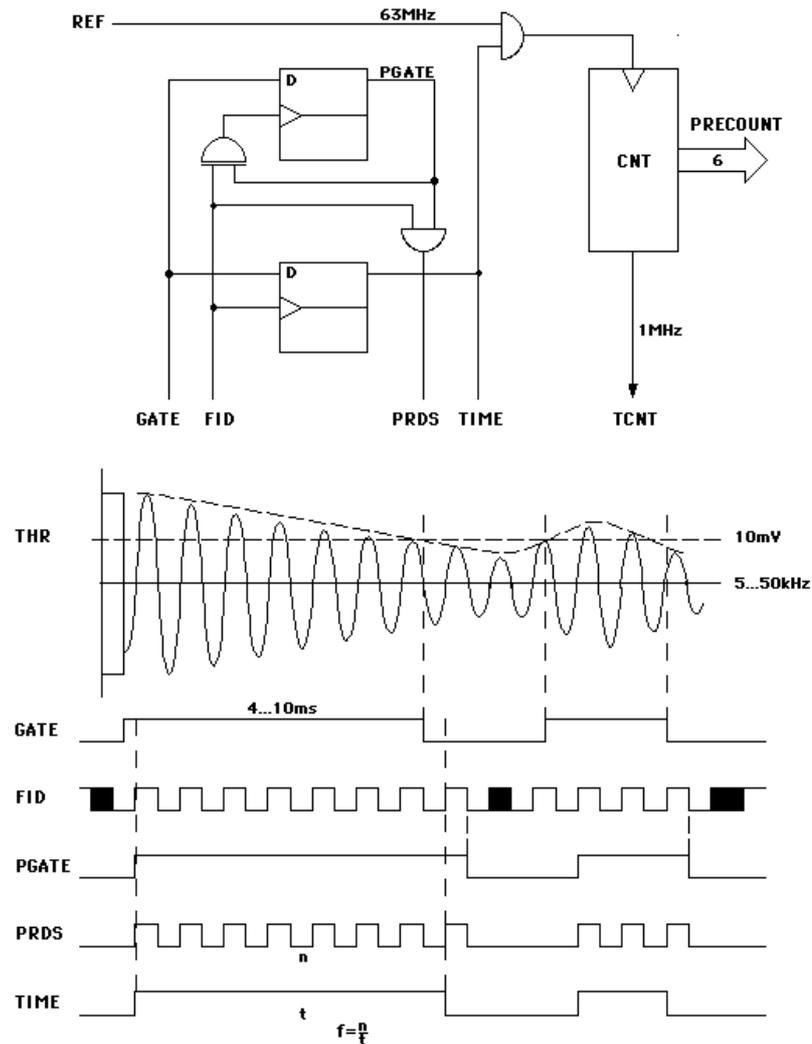


Bild 3: NMR Vorverarbeitung

Durch einen Vorzähler wird die Referenz von ca. 60 MHz vorgeteilt um sie mit ca. 1 MHz (TCNT) der direkten Verarbeitung im Controller zugänglich zu machen.

Diese Schaltung wird komplett in einem PAL realisiert.

Die erreichbare Auflösung liegt in der Unsicherheit der Zeitmessung und beträgt z.B. bei Verwendung von 6 Bit Vorteilung:

$$\begin{aligned} \Delta f/f &= 1 / (\text{Messdauer} * \text{Clock} + 1) \\ &= \text{z.B. } 1 / (4\text{ms} * 60 \text{ MHz} + 1) \\ &= 4 * 10^{-6}; \end{aligned}$$

2. BETRIEB

2.1. Betriebssystem

Für Abarbeitung mehrerer Programme (Tasks) ist ein kooperatives Betriebssystem zu Grunde gelegt. Die Taskumschaltung ist nicht interruptgesteuert sondern wird implizit (beim Warten auf I/O) oder explizit durch Funktionsaufruf gesteuert. Der Vorteil liegt in der gesicherten Abarbeitung von zeitkritischen Programmen, da diese nicht durch System-Interrupts unterbrochen werden. Ein einfaches Memory-Managementsystem sowohl für das RAM (Heap) wie auch für den Flash-Speicher (Storage) erlaubt dynamische Zuweisung und Freigabe von Speicher.

Eine einfache Task-Verwaltung erlaubt modulare und dynamische Erweiterung bzw. Änderung der eingesetzten Programme. Alle Tasks sind in einer Liste eingetragen und können über den Namen für die Kommunikation geöffnet werden. Da im System mindestens eine Task bereits vorhanden sein muß, existiert immer eine Task SYSTEM, die grundlegende Systemsteuerungen erlaubt.

Ein spezieller MONITOR erlaubt unabhängig von gerade aktiven Tasks den unmittelbaren Zugriff auf verschiedenste Systemparameter, das Memory sowie die internen Prozessorregister. Hier wird auch die Kommunikation zu den verschiedenen Tasks geöffnet.

Die im folgenden beschriebenen Kommandos (Buchstaben) für den MONITOR und die unterschiedlichen Tasks werden ohne "CR" erkannt. Alle angegebenen Parameter (Zahlen und Strings) müssen immer durch "CR" abgeschlossen werden. Alle zurückgeschickten Daten sind durch "CR" (ohne "LF"!) abgeschlossen! Falls es sich um Hex-Zahlen handelt ist ein "\$" vorgestellt. Empfangene Zeichen werden nur zurückgeschickt wenn das ECHO eingeschaltet ist (siehe Befehl E/e in MONITOR).

2.1.1. MONITOR

Der Monitor kann jederzeit durch das Zeichen „Esc“ gestartet werden, wobei die aktive Task per Interrupt unterbrochen wird. Der Monitor wartet nunmehr auf ein Kommando, wobei nach der Ausführung das unterbrochene Programm fortgesetzt wird.

Kommandos:

- ? sendet Übersicht über die vorhandenen Kommandos.
- ^ liefert den Namen der aktuellen Task.
- # name sendet Status der Task name.
"Leerstring": Task ist nicht vorhanden.
"O": Task ist 'geöffnet'.
"R": Task ist 'ready', d.h. steht zur Ausführung bereit.
"W": Task is 'waiting', d.h. steht noch nicht zur Ausführung an.
- & size,d.. legt im Heap einen Block mit der Größe size an, lädt size gesendete Bytes (Binärformat) in diesen Block und startet anschließend an der ersten Adresse im Block. Damit kann z.B. eine Task in den Heap nachgeladen und installiert werden.
- @ addr startet ein (beliebiges) Programm an der Adresse addr.
- B addr,d schreibt das Byte (8 Bits) d in die Adresse addr.
- b addr sendet das Byte (8 Bits) von der Adresse addr.

- D/d addr sendet 16 Bytes im Hex- und ASCIIformat ab addr. Jedes weitere Kommando "D/d" stellt die vorherigen/nächsten 16 Bytes dar. Der Befehl wird mit einem beliebigen anderen Kommando beendet.
- E Echo Ein.
Alle empfangenen Zeichen werden grundsätzlich zurückgesendet. Falls ein CR gesendet wird, wird auch noch ein LF angehängt. Sinnvoll bei direkter Bedienung mit Terminal.
- e Echo Aus (standard).
Es erfolgt kein Echo der gegebenen Kommandos. Sinnvoll bei Computeransteuerung. W addr,d schreibt das Wort (16 Bits) d in die Adresse addr.
- H Hex Ein.
Alle Zahlen werden im Hexformat ausgegeben (z.B: \$1A)
- h Hex Aus (standard).
Alle Zahlen werden im Dezimalformat ausgegeben. (z.B: 26)
- I d.. setzt Inhalt aller Prozessorregister.
ACHTUNG: Nur verwenden wenn vorher gestoppt wurde!
- i sendet Inhalt alle Prozessorregister.
- K sendet einen Prüfsumme (Byte), die aus allen gesendeten Daten gebildet wird und setzt diese anschließend auf 0. Kann zur Kontrolle von umfangreichen Datentransfers benutzt werden.
- k sendet einen Prüfsumme (Byte), die aus allen empfangenen Daten gebildet wird und setzt diese anschließend auf 0. Kann zur Kontrolle von umfangreichen Datentransfers benutzt werden.
- L addr,d schreibt das Langwort (32 Bits) d in die Adresse addr.
- l addr sendet das Langwort (32 Bits) von der Adresse addr.
- N a,s,d.. schreibt einen Block mit der Länge s in die Adresse a. Die gesendeten Daten sind unformatiert binär.
- n addr,s sendet einen Block im Binärformat mit der Länge s ab der Adresse addr.
- O name 'öffnet' Task name zur Kommunikation.
- o sendet Name der gerade 'geöffneten' Task.
- S stop ein Stopbefehl mit einer bestimmten Priorität wird ausgeführt:
 $0 \leq \text{stop} \leq 7$: Die Prozessorpriorität wird auf entsprechend 0..7 gesetzt.
 $8 \leq \text{stop} \leq 15$: Der Prozessor geht in STOP mit Priorität entsprechend 0..7.
 $16 \leq \text{stop} \leq 23$: Der Prozessor geht in LowPower STOP mit Priorität 0..7.
- s sendet den momentanen Stop Modus.
- W addr,d schreibt das Wort (16 Bits) d in die Adresse addr.
- w addr sendet das Wort (16 Bits) von der Adresse addr.
- (..) Aufruf einer Systemroutine. (Beschreibung an anderer Stelle!).
- "CR" sendet entweder einen Leerstring ("CR") oder einen String mit dem zuletzt aufgetretenen Fehler im System! Kann als Quittierung, ob der vorherige Befehl korrekt abgearbeitet worden ist, benutzt werden.

2.1.2. SYSTEM

Diese grundlegende Task ist mit der Möglichkeit der Kommunikation zur Außenwelt (z.B. über die serielle Schnittstelle) ausgestattet und hält eine Reihe von Kommandos zum Anzeigen und Setzen von Systeminformationen und -parametern bereit. Zur Erweiterung des Systems können Tasks neu angelegt bzw. über S-Records nachgeladen oder wieder gelöscht werden. Andere Tasks können durch die Einstellung der Zeit-Parameter gestartet oder verzögert werden.

Kommandos:

?	sendet Übersicht über hier vorhandenen Kommandos.
& size,..	Daten mit der Länge size Bytes werden in den Heap geladen. Die Daten werden im Binärformat erwartet.
B rate	Die Baudrate für die serielle Kommunikation wird auf rate gesetzt (300..38400 Baud).
b	Die serielle Schnittstelle wird initialisiert. Der Buffer gelöscht.
C clk	Die Systemuhr wird auf den Wert clk gesetzt. Falls der Periodic Interrupt Timer nicht eingeschaltet ist, wird die Clock per Software (jeweils bei Taskwechsel) weitergezählt.
c	Der aktuelle Wert der Systemuhr wird gesendet.
D name	Die Task name wird gelöscht.
d name	Die Task name wird neu gestartet (entsprechend delay)
E	Der komplette Storage wird gelöscht (nicht System!)
e blk	blk zeigt auf einen Block. Falls addr in Heap wird der Block freigegeben. Falls addr in Storage wird der Block als ungültig markiert.
F	sendet Informationen über belegte Blöcke im Storage: 1) Adresse des Blocks. 2) Block-Kennung. 3) Größe des Blocks.
f	Summe des noch verfügbaren Storage im Flash.
G blk	sendet Einsprungadressen und Namen aller Routinen im Block blk.
g name	sendet die Adresse des ersten Globals (erste erklärte Variable im Stack) von unterbrochener Task name.
H	sendet eine Liste der angelegten Blöcke im Heap 1) Adresse des Blocks. 2) Größe des Blocks. 3) tatsächlich beanspruchter Speicherplatz.
h	Summe des noch verfügbaren Heaps im RAM.
L	sendet eine ausführliche Liste der geladenen Tasks in der Reihenfolge der Eintragung. 1) Adresse des Task-Records. 2) Adresse des Task-Codes. 3) Einsprungadresse. 4) Adresse des im Heap abgelegten Stacks 5) Adresse des zurückzuspeichernden Stacks. 6) maximale Größe des Stacks 7) Time-Parameter: bestimmt wann Task wieder gestartet wird.

	8) Delay-Parameter: bestimmt wie Time bei Unterbrechung gesetzt wird.
	9) Status
	10) Name der Task
l	sendet nur eine Namensliste der geladenen Tasks in der Reihenfolge der Eintragung.
M name	legt eine Task name (falls noch in Heap) im Storage ab.
m blk,tag	legt einen Block a im Heap mit der Kennung tag im Storage ab.
O name	generiert Startup-Block mit aktueller Baudrate und 'Öffnen' der Task name.
o name,n	dupliziert die Task name mit neuem Name n. Dabei wird lediglich ein neuer Task-Control-Block angelegt. code und entry werden von name übernommen.
R	Reboot: das System wird neu gestartet.
r	Reset: ein Hardware-Reset wird durchgeführt.
S rec	empfängt S-Records (Motorola-Format) und legt die Daten im Memory ab. Bei einem Start-Record (S7,S8,S9), wird entsprechend gestartet.
s addr,s	sendet einen Memorybereich mit der Größe s ab Adresse addr im S-Record format.
T name,t	setzt den Delay-Parameter der Task name auf t.
t name,t	setzt den Time-Parameter der Task name auf t.

2.2. TROLLEY

Dies ist das eigentliche Anwenderprogramm, wobei (durch entsprechende Programmierung in SYSTEM „O name“) die Kommunikation mit dieser Task nach dem Einschalten immer bereits automatisch geöffnet ist. Ansonsten muß diese über den MONITOR mit dem Befehl „Esc O Trolley“ geöffnet werden.

Alle im folgenden beschriebenen Kommandos werden ohne "CR" erkannt. Alle zusätzlich notwendigen Parameter müssen immer durch "CR" (also z.B.: A132"CR") abgeschlossen werden. Alle zurückgeschickten Daten sind immer durch "CR" (ohne "LF"!) abgeschlossen!

Empfangene Zeichen werden nur zurückgeschickt wenn das ECHO eingeschaltet ist.

Kommandos:

?	sendet Übersicht über die vorhandenen Kommandos .
!	sendet Version des Programms (z.B. "Trolley vWalter 30.9.98").
*	versetzt Mess-System in Power-Down Modus. Alle Werte im Speicher werden erhalten.
@	Leerkommando: versetzt das Mess-System nach Power-Down wieder in den Power-Active Modus bzw. bricht wiederholende NMR-Sequenz ab.
"CR"	sendet entweder einen Leerstring ("CR") oder einen String mit dem letzten aufgetretenen Fehler . Kann auch als Quittierung für die Abarbeitung eines vorangegangenen Befehls benutzt werden!
A ch	startet den internen ADC und sendet einen entsprechenden 8-Bit Wert (val) zurück. ch bestimmt welcher Kanal in welchem Modus gewandelt wird und ergibt sich aus ch=Modus+KanalNummer (0..7).

Modus	Bedeutung	Standard
128	Single	Configured Conversion
64	Differential (2 Kanäle)	Single Ended
32	Bipolar	Unipolar
16	Load Mux	Start Conversion
8	PowerDown	Normal

Beispiele:

Messgröße	Kanal	Kommando	Wert
Druck	0&1 (differentiell)	A192	$U = \text{val} * 2.5/256$ Volt; $P = U * 1024/2.1$ mBar;
Envelope	2	A130	$U = \text{val} * 2.5/256$ Volt;
FID-Clock	3 (bipolar)	A163	if val > 127 $U = (\text{val}-256) * 2.5/128$ Volt else $U = \text{val} * 2.5/128$ Volt;
Power VB	4	A132	$U = \text{val} * 2.5/256 * 6.1$ Volt;

Die Spannung VB wird vor dem Low-Drop Spannungsregel zur 5V Versorgung gemessen und sollte >5.5V (typ. 6V) betragen! Dies wird durch die zugeführte Spannung von außen bestimmt.

- B ch startet interne **Block Conversion**. D.h. es wird ein normaler NMR-Messzyklus begonnen, während der eigentlichen Messung (Gate) wird das FID-Signal durch den (onboard) ADC mit der maximalen Geschwindigkeit (über eine Software-Loop) abgetastet und die Werte zwischengespeichert. Der Wert für TimeOut (0..1000) bestimmt die Anzahl der Samples.
- b Die im Befehl "B" aufgezeichneten Werte werden im **Block** an den Host gesendet. Das Format ist 8-Bit im 2-er Komplement (das Bit7 ist das Vorzeichenbit, siehe auch FID im Befehl "A"). Jeder Wert ist durch "CR" beendet.
- C thr setzt Schwelle für Diskriminierung der **FID-Clock** (0..255) (typ.120)
- c sendet aktuellen Wert für Schwelle **FID-Clock**.
- D 5*n setzt 5 Werte für **Timing** des NMR-Messablaufs: DelayPower, DelayTx, DurationTx, DelayPreamp, DelayStartMeasurement (Alle 5 Zahlen (0..32767) jeweils mit CR beenden!) ($t = 15\mu\text{s} + n * 1.5\mu\text{s}$). Falls DelayPreamp=0 wird Preamplifier bereits mit Power eingeschaltet!
- d sendet 5 Werte für **Timing** (DelPower, DelTx, DurTx, DelPreamp, DelSM).
- E thr setzt Schwelle für **Envelope** (0..255) (typ.10).
- e sendet aktuellen Wert für Schwelle **Envelope**.
- F thr setzt Schwelle für Pos.Sensorgruppe **ClockInterrpt** (0..255) (typ.128).
- f sendet aktuellen Wert für Schwelle **ClockInterrpt**.
- G thr setzt Schwelle für Pos.Sensorgruppe **DirCode** (0..255) (typ.128).
- g sendet aktuellen Wert für Schwelle **DirCode**.
- H schaltet Positionssensoren (**LEDs Clock**) ein.
- h schaltet Positionssensoren (**LEDs Clock**) aus.
- I ticks setzt die **Systemuhr** auf Interruptsteuerung. Ticks bestimmt, mit welcher Rate der Interrupt erfolgt (z.B. ticks=272: 1 sekunde):

- ticks=0; 256: Interrupt AUS (Uhr wird per Software weitergezählt!)
 ticks=1..255: 122µs .. 31.1ms;
 ticks=257..511: 62,5ms .. 15.9sec;
- i sendet **Systemlock** (Setzen in SYSTEM mit Befehl "C").
- J schaltet LEDs für **DirBit** ein.
 Dies ist nur für Testzwecke notwendig, da ansonsten die Steuerung automatisch bei Interrupt erfolgt!
- j schaltet LEDs für **DirBit** aus.
- K schaltet LEDs für **CodeBit** ein.
 Dies ist nur für Testzwecke notwendig, da ansonsten die Steuerung automatisch bei Interrupt erfolgt!
- k schaltet LEDs für **CodeBit** aus.
- L time setzt **Timeout** (0..32767) für Abbruch der NMR-Messung ($t = \text{time} * 1.5\mu\text{s}$).
 Bei $\text{time} \geq 30000$, wird für den Abbruch der NMR-Messung auch die FID-Envelope herangezogen!
- l sendet aktuellen **Timeout**.
- M n,ch setzt Schritt n (1..1000) in der **Mess-Sequenz** auf Kanal (Probe) ch=(1..17).
 Die Kanalangabe ch=0 stoppt die Messung;
 Die Kanalangabe ch=18 wiederholt die Sequenz (ab Schritt 1)
 Bis zu 1000 Schritte (n=1..1000) sind möglich.
- m n sendet Kanalnummer ch von Schritt n in der **Mess-Sequenz**.
- N startet vordefinierte **NMR-Mess-Sequenz**:
 Falls die Mess-Sequenz ständig wiederholt wird (ch=18), wird durch ein beliebiges Kommando (bzw. Leerkommando '@') dieser Vorgang abgebrochen. Eine anschließend zurückgesendete Zahl gibt immer den zuletzt bearbeiteten Mess-Schritt an.
 Die Daten werden mit dem Kommando O/o ausgelesen.
 ACHTUNG: Durch DelayTx kann der zeitliche Abstand der Messungen zueinander eingestellt werden.
- n ch startet **eine** NMR-Messung von Kanal ch und sendet anschließend die Daten in folgendem Format mit 5 Zahlen zurück:
 1: ch=Kanalnummer (Probe)
 2: posA= Position Sensor A (Trolley) bei der Messung
 3: posB= Position Sensor B (Trolley) bei der Messung
 4: TC= timeCounter aus NMR-Messung (Referenz=fr)
 5: PC= periodCounter aus NMR-Messung
 Mit einer Referenzfrequenz fr (=61.7MHz) für TC ergibt sich die NMR-Frequenz nach folgender Formel:

$$f = \text{fr} * \text{PC}/\text{TC}$$
- O sendet alle Daten in oben beschriebenem Format aus **NMR-Mess-Sequenz** von Schritt 1..bis Schritt n (mit ch=0 oder ch=18).
- o n sendet Daten in oben beschriebenem Format aus **NMR-Mess-Sequenz** von Schritt n.
- P n setze **Positionscouter A** auf n (0..2147483647).
- p sende aktuellen **Positionscouter A**.

Q n	setze Positionscouter B auf n (0..2147483647).
q	sende aktuellen Positionscouter B .
R	schaltet den Interrupt für Positionscouter A an.
r	schaltet den Interrupt für Positionscouter A ab (nur für Testzwecke)
S	Positionszähler synchronisieren sich an den Absolut-Codemarken.
s	Positionszähler arbeiten nur relativ.
T n	Auslese des internen Temperatur-Sensors. Rückgabe von 2 Zahlen H, P. n bestimmt die Genauigkeit bzw. Geschwindigkeit der Messung (typ 1000) $T = (H * 212.77/P) - 68.085 \text{ } ^\circ\text{C}$;
t n	Auslese des externen Temperatur-Sensors. Rückgabe von 2 Zahlen H, P. n bestimmt die Genauigkeit bzw. Geschwindigkeit der Messung (typ 1000) $T = (H * 212.77/P) - 68.085 \text{ } ^\circ\text{C}$;
U	schaltet den Interrupt für Positionscouter B an.
u	schaltet den Interrupt für Positionscouter B ab (nur für Testzwecke)
V ch	setzt Sensorkanal auf Probe ch .
v	schaltet Sensorkanäle ab.
W	aktiviert Power (Stromversorgung ohne Controller!)
w	deaktiviert Power
X	aktiviert Transmit (Versorgung Sender)
x	deaktiviert Transmit
Y	aktiviert Preamp (Versorgung Vorverstärker)
y	deaktiviert Preamp
Z	setze Gate für Messung
z	lösche Gate für Messung

Die Kommandos V..z sind nur für Testzwecke vorgesehen. Die eigentliche Ablaufsteuerung der NMR-Messung wird immer automatisch nach Bild 2. mit den vorgeladenen Timing-Parametern durchgeführt.

2.3. Download

Durch Download kann im Prinzip das gesamte Betriebssystem ersetzt werden. Dies kann in diesem Fall jedoch nur durch ein entsprechendes Loaderprogramm erfolgen um zu verhindern, daß durch einen einfachen Befehl das Betriebssystem zerstört wird. Der Bereich des Betriebssystems ist per Software gegen Überschreiben geschützt. Falls das Betriebssystem zerstört wurde, muß durch ein spezielles Interface (BDM) das Flash-Memory neu geladen werden.

Typischerweise werden neue Anwenderprogramme als Tasks nachgeladen und im Flash-Memory installiert. Eine Task muß so geschrieben sein, daß sie sich selber als Task installiert und dann über das System gestartet werden kann.

Für das Austesten eines Programms kann dieses auch zunächst in den Heap (RAM) geladen werden und dort gestartet werden.

2.3.1. Download über S-Records:

Eine einfache Möglichkeit besteht darin, das Programm über S-Records in das System zu laden. Ein Problem liegt darin, dass die Adressen in den S-Records absolut angegeben sind und deshalb das Memory-Management des Systems (Heap) nicht benutzt werden kann. Es ist deshalb ein Adressbereich zu verwenden, der mit dem System-Memory nicht kollidiert. Im Prinzip wird zwar das komplette RAM für Heap und Stack verwendet, es kann aber in der Regel die Adresse \$1000..\$F000 für diesen Zweck benutzt werden (falls der Heap noch nicht über \$1000, bzw. der Stack noch nicht unter \$F000 gewachsen sind!)

Die Task SYSTEM versteht den ersten Buchstaben "S" eines jeden S-Records als Befehl und führt dessen Bedeutung aus:

<u>Befehl</u>	<u>Bedeutung</u>
S0..3 record	S-Record lädt sich an die angegebene Adresse
S7..9 record	startet das Programm

Das Programm kann jetzt so geschrieben sein, dass es sich entweder an der geladenen Stelle startet und im RAM läuft oder sich gleich in das Flash kopiert. Im letzteren Fall muss es wieder so geschrieben werden, daß es sich selber als Task (beim Einschalten) installiert.

ACHTUNG: um eine Task mit dem Befehl "M" vom Heap in den Storage transferieren zu können, muß diese über das Betriebssystem im Heap installiert sein. Zu diesem Zweck ist das geladene S-Record-Programm so zu schreiben, daß es sich zuerst regulär über das Betriebssystem im Heap installiert.

2.3.2. Download in den Heap:

Hier liegt der Vorteil darin, daß der notwendige Speicherbereich vom System selber zur Verfügung gestellt wird und damit keine Kenntnis über die Speicherbelegung vorhanden sein muß.

Der MONITOR stellt mit dem Befehl "&" eine entsprechende Funktion zur Verfügung: Nach dem Befehl "&" erwartet der Monitor eine Zahl, die die Größe des gesendeten Blocks (in Byte) angibt. Als nächstes wird der komplette Block in Binärform mit der entsprechenden Anzahl von Bytes gesendet und anschließend an der ersten Stelle im Block gestartet.

Das Programm kann natürlich so geschrieben werden, daß es sich anschließend gleich in das Flash kopiert!

2.3.3. Abspeichern im Flash-Memory:

Falls eine Task im Heap ausgetestet ist, kann sie in das Flash-Memory transferiert werden und ist anschließend immer verfügbar.

Die notwendigen Befehle sind in der Task SYSTEM verfügbar. Falls die Kommunikation mit dieser Task nicht geöffnet ist, muß diese zunächst mit dem Befehl „Esc O System“ geöffnet werden!

Ein entsprechender Befehl in der Task SYSTEM "M name" kopiert die Task name (aus dem Heap) in das Flash-Memory. **ACHTUNG:** Die Task muß regulär über das Betriebssystem im Heap installiert sein.

Der Befehl „O name“ speichert die Kommunikation bezüglich der Task name mit der aktuellen Baudrate ab. Damit ist die gewählte Kommunikation auch nach dem Einschalten sofort verfügbar.

2.4. Beispiele

ACHTUNG: Im folgenden sind die einzelnen Befehle (zur besseren Darstellung) durch Leerzeichen voneinander getrennt. Diese dürfen nicht eingegeben werden!

Die meisten Befehle im MONITOR bzw. in der SYSTEM-Task sind für den normalen Betrieb nicht erforderlich. Hier nochmal eine Zusammenfassung der unbedingt notwendigen Bedienungsschritte für einige Betriebsarten:

2.4.1. Einschalten und Start von TROLLEY

- 1) Anlegen der Stromversorgung (ca. 7V) über die DC-Einspeisung der Referenz. Es sollte ein Strom von ca. 350mA fließen. Es kann vorkommen, daß der Autoreset des Controllers nicht richtig erfolgt (Strom nur ca. 250mA, Controller meldet sich nicht). In diesem Fall muß die Stromversorgung noch einmal ausgeschaltet und dann wieder eingeschaltet werden.
- 2) Die serielle Schnittstelle arbeitet üblicherweise (falls nicht umgestellt) mit 9600 Baud, 8 Bits, 1 StopBit, NoParity. Da standardmäßig die Task TROLLEY geöffnet ist, sollte diese sich z.B. nach dem Kommando „!“ durch Senden eines Versions-Strings (z.B. "Trolley vWalter 14.4.98") melden. Durch Senden des Kommandos „?“ kann eine Liste der Kommandos abgerufen werden.

2.4.2. Ein- und Ausschalten des Echo

Falls die Bedienung an einem normalen Terminal durchgeführt wird, sollte zunächst das Echo eingeschaltet werden, um eine leichtere Bedienung zu erlauben.

- 1) Das Kommando „Esc E“ schaltet das Echo ein.
- 2) Das Kommando „Esc e“ schaltet das Echo aus.

2.4.3. Öffnen einer Task

Das Öffnen einer Task verbindet die Kommunikation über die serielle Schnittstelle mit dieser Task. Eine Task, die (momentan oder überhaupt) keine Kommunikation benötigt, ist davon unberührt. Falls eine Task eine Eingabe erfordert oder eine Ausgabe machen möchte, wird diese Task unterbrochen bis die Kommunikation zu dieser Task geöffnet wird.

- 1) Das Kommando „Esc O name CR“ öffnet die Kommunikation mit der Task name.

2.4.4. Einstellen einer anderen Baudrate

- 1) Öffnen der Task SYSTEM mit „Esc O System CR“.

- 2) Ändern der Baudrate mit dem Befehl z.B. „B 38400 CR“. Die neue Baudrate wird sofort eingestellt und ist für die folgende Kommunikation maßgeblich.
- 3) Gegebenenfalls die Task TROLLEY mit „Esc O Trolley CR“ wieder öffnen.

2.4.5. Download (S-Records) einer neuen "TROLLEY" Version

In diesem Fall wird vorausgesetzt, daß das neue Programm („Trolley_neu“) so geschrieben wurde, daß es sich zunächst im Heap (RAM) als Task installiert. Falls im übrigen eine neue Task mit dem gleichen Namen (Trolley) installiert wird, wird der Name der alten Task mit einem Punkt am Ende gekennzeichnet (Trolley.), usw.

- 1) Öffnen der Task SYSTEM mit „Esc O System CR“.
- 2) Über den Host wird das File mit den **S-Records** einfach als ASCII-Stream zum Trolley geschickt. Dabei wird für jeden S-Record der S-Rec-Typ und die Ladeadresse zurückgemeldet (z.B: „S3: \$1200“). Durch einen abschließenden, entsprechenden S-Record wird das Programm automatisch gestartet und installiert sich automatisch.
Kontrolle: Mit dem Kommando „I“ kann überprüft werden, ob das neue Programm korrekt installiert wurde. Es sollte in der Liste der Tasks erscheinen (z.B. „Trolley_neu CR Trolley_alt CR System“)!
- 3) Das Programm wird jetzt mit dem Befehl „M Trolley_neu CR“ in den Storage (Flash-Memory) kopiert.
Kontrolle: Mit dem Befehl „F“ kann dies verifiziert werden! Es sollte ein neuer Block mit der ID=TASK angelegt sein!
- 4) Da sich mehrere Tasks im Speicher befinden können, muß schließlich noch hierfür durch den Befehl „O Trolley_neu CR“ die Default-Kommunikation festgelegt werden.
Kontrolle: Nach dem Einschalten sollte damit das neue Programm automatisch geöffnet sein und sich auf ein entsprechendes Kommando (z.B. „!“) melden.

2.4.6. Löschen einer Task

- 1) Öffnen der Task SYSTEM mit „Esc O System CR“.
- 2) Das Kommando „D name“ löscht die Task name.
- 3) Gegebenenfalls die Task TROLLEY mit „Esc O Trolley CR“ wieder öffnen.

2.4.7. Neuanlegen des Flash-Memories

Blöcke im Flash-Memory werden zunächst nicht gelöscht sondern nur als „gelöscht“ markiert. Bei der Anlage von neuen Blöcken wird somit immer wieder neuer Speicher verwendet. Falls der Speicher für neue Blöcke nicht mehr reicht (aber ausreichend „gelöschter“ Speicher zur Verfügung stehen würde), muß der Flash-Speicher insgesamt gelöscht und neu angelegt werden.

- 1) Öffnen der Task SYSTEM mit „Esc O System CR“.
- 2) Das Kommando „F“ gibt einen Überblick über die momentane Belegung.
- 3) Das Kommando „E“ löscht das Flash-Memory komplett (nicht das Betriebssystem).
- 4) Die gewünschten Tasks (siehe 2.4.5) wieder neu einspielen.
- 5) Gegebenenfalls die Task TROLLEY mit „Esc O Trolley CR“ wieder öffnen.

3. FERTIGUNG

3.1. Jumper

Am MP42-2 muss der Jumper J7.2-3 gesetzt sein!

3.2. Clock

Um ein verbessertes Anschwingen der Clock zu erreichen, sollte der Widerstand R1 auf der MP42-2 von 20M Ω auf 10 M Ω erniedrigt werden!