

ARE Autorun Editor

Eine sehr nützliche Funktion an den Funkmessgeräten der Stabilock Familie ist das Messen mittels selbstablaufender Routinen. Für das Erstellen solcher Programme gibt es im Handbuch, hier das des 4015, im Kapitel 9 eine umfassende Erklärung sowie die Auflistung der zur Verfügung stehenden Kommandos für das AUTORUN.

Eine Erstellung solcher Routinen kann direkt auf dem Stabilock erfolgen hierzu ist jedoch eine Tastatur erforderlich.

Da nur wenige der speziellen Tastaturen zum Anschluss an den Stabilock verkauft wurden kann davon ausgegangen werden das die Erstellung meist auf einem PC erfolgt. Für einen auf DOS basierenden Rechner wurde die Software ARE von Wavetek angeboten.

Dieser spezielle Editor ist in der Lage die erfassten Routinen mittels IEEE / GPIB an den Stabilock zu übertragen.

Voraussetzung, egal ob Anschluß der Tastatur am Stabilock oder Steuerung mittels IEEE ist natürlich das der Stabilock über die Option des Interfaces IEEE / Keyboard / RS-232 verfügt.

Der Besonderheit des Transfer der Autorun-Programme zwischen PC und Stabilock möchte ich kurz erläutern.

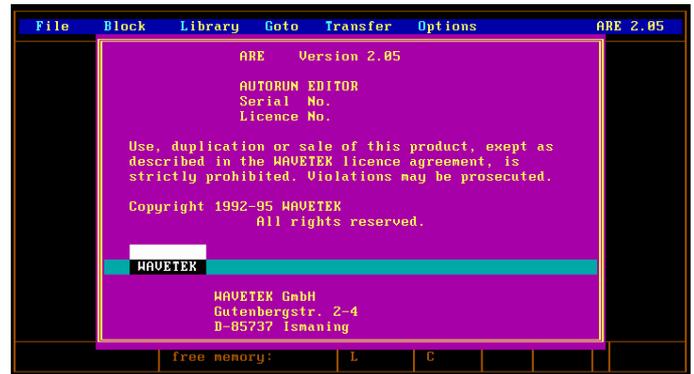
Schon beim Aufruf des Software ARE prüft diese als erstes ob der GPIB-Handler angesprochen werden kann, ist dies nicht möglich erscheint ein entsprechender Hinweis (Bild2).

Ist der GPIB-Handler, es handelt sich hier um das Programm gpib.com, funktionsfähig wird als zweites geprüft ob ein Stabilock mittels IEEE angesprochen werden kann. Hierzu ist jedoch erforderlich das in der ARE-Software unter Optionen bereits der Typ des Stabilock z.B. 4015 und dessen IEEE Adresse korrekt eingetragen ist.

Dieser ersten Kontaktaufnahme möchte ich noch eine weitere Betrachtung zu kommen lassen.

Möchte man eventl. mit anderen IEEE basierender Software ebenfalls mit den Stabilock Geräten kommunizieren so muss man natürlich Wissen welche Abfolge der Kommunikation auf dem IEEE-Bus erfolgt.

Diese Kommunikation habe ich mittels des Tools APPMON, der zur National Instruments AT-GPIB II Karte mitgeliefert wird, protokolliert. Die oben angesprochen Verbindungsaufnahme zwischen ARE-Software und Stabilock erfolgt eigentlich auf einer vergleichsweise einfachen Abfrage. ARE sendet hierzu die Zeichenfolge „UNIT“ zum Stabilock worauf dieser mit Ausgabe des Types, hier also „4015“ antwortet. ARE setzt dann den Stabilock mit dem Kommando GTL (go to local) wieder in die Bereitschaft diesen mittels dem Frontpanel zu bedienen. Eine detaillierte Aufzeichnung dieser IEEE Kommunikation ist im Anhang aufgelistet.



Erstellung der Autorun Routine

Nun kann die Erstellung einer Autorun Routine im ARE Editor beginnen.

Die Befehle bzw. Kommandos sind eine Mischung aus BASIC und IEEE Kommandos. Bei der Eingabe muß keine führende Nummerierung der Zeilen eingegeben werden.

Diese Nummerierung, Vorgabe über Nummerierungsbeginn und Schrittweite erfolgt über die Optionen, wird somit von ARE erzeugt.

```

File  Block  Library  Goto  Transfer  Options  ARE 2.05

REM *** RECEIVER SENSITIVITY ***
SETRX
REM *** All values are suggestions ***
REM
RXFREQUENCY 85.155MHz
MODAF 1KHz
RXAFM 1.5KHz
REM *** Jump into the SPECIAL mask ***
SOFT_SPECIAL
SOFT_SENS
SELECT SINAD
WRTVA 21,12dB
SOFT_RUN
REM *** Result is read out in var A ***
A=RESULT 1
PRINT "Receiver Sensitivity: ";A
PRINT "===== "

NONAME.ART  free memory: 15930  L 2  C 1  Ins  * 4015

```

Eine Anzeige der Zeilennummerierung kann zur Kontrolle eingeschaltet werden. Hierbei wird das Listing in grüner Schrift angezeigt, ein Editieren ist jedoch bei diesem Anzeigemodus nicht möglich. Ebenfalls eine grüne Anzeige mit Zeilennummerierung erhält man wenn man die Routine zum Transfer aktiviert.

Für die schnelle Erstellung von Routinen können aus der so genannten Library Programmteile in Form von Modulen geladen werden. Bei Aufruf des Ladens aus der Library wird eine Übersicht der verfügbaren Module sowie der zugehörige Gerätetype also 4015 oder 4031 für den das Modul erstellt wurde angezeigt.

```

File  Block  Library  Goto  Transfer  Options  ARE 2.05

0010 REM AUTORUN STABILOCK 4015 SW Vers. 2.4
0020 REM *** RECEIVER SENSITIVITY ***
0030 SETRX
0040 REM *** All values are suggestions ***
0050 REM
0060 RXFREQUENCY 85.155MHz
0070 MODAF 1KHz
0080 RXAFM 1.5KHz
0090 REM *** Jump into the SPECIAL mask ***
0100 SOFT_SPECIAL
0110 SOFT_SENS
0120 SELECT SINAD
0130 WRTVA 21,12dB
0140 SOFT_RUN
0150 REM *** Result is read out in var A ***
0160 A=RESULT 1
0170 PRINT "Receiver Sensitivity: ";A
0180 PRINT "===== "

<ESC> to quit  free memory: 15930  L 1  C 1  Ins  * 4015

```

Mittels der Funktion Block-Anfang und Block-Ende können auch Programmteile in die Library gespeichert werden. Die Ablage im Filesystem von ARE erfolgt im SubDirectory „LIBRARY“ als Filetype *.alb. Diese Files sind normale ASCII-Files die die Befehle ohne Zeilennummerierung enthalten. Jede Zeile wird mit CR+LF (Hex 0D und Hex 0A) abgeschlossen. Als letztes Zeichen in der Datei steht ein Hex 1A. Die Übersicht des SubDir-Inhaltes wird im File

```

File  Block  Library  Goto  Transfer  Options  ARE 2.05

Library
-----
Name      Type  Description
AF_RESP   4031  Modul for AF RESPONSE measurement
DISTO     4031  Modul for AF DISTORTION measurement
NOISE     4031  Modul for HUM&NOISE measurement
OFFSET    4031  Modul for OFFSET measurement
SENS      4015  Modul for receiver SENSITIVITY
SEQUENT   4015  Modul for SEQUENTIAL measurement

LIBRARY.DIR  free memory: 16025  L 1  C 1  Ins  * 4015

```

Die fertige Routine kann ebenfalls gespeichert werden. Hierbei wird das Listing ohne führende Zeilennummerierung als ASCII-File in das SubDirectory „PROGRAMS“ als Filetyp *.art abgespeichert. Auch in dieser Datei ist als letztes Zeichen wieder ein Hex 1A enthalten. Beispiele der Dateien befinden sich ebenfalls Im Anhang.

```

File  Block  Library  Goto  Transfer  Options  ARE 2.05

Load
Save
save as  es are suggestions ***
Delete
diRectory  55MHz
Print
dOs
clear Editor  o the SPECIAL mask ***
Quit AltX

SELECT SINAD
WRTVA 21,12dB
SOFT_RUN
REM *** Result is read out in var A ***
A=RESULT 1
PRINT "Receiver Sensitivity: ";A
PRINT "===== "

NONAME.ART  free memory: 15974  L 1  C 1  Ins  * 4015

```

Der Transfer

Ein fertiges Autorun Programm kann über die Auswahl im Menü Transfer nun zum Stabilock übertragen werden.

Auch hier will ich den Vorgang bezüglich der IEEE Kommunikation im Detail betrachten.

Als einleitende Zeichenfolge erhält der Stabilock „LEARN programm“, wobei der Name „programm“ dem im ARE vergebenen Programmnamen des *.art File entspricht.

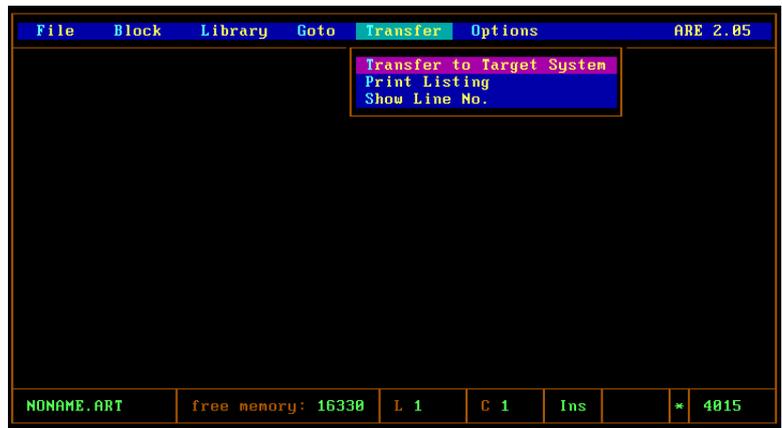
Diese Zeichenfolge veranlasst den Stabilock nun jede nachfolgende Zeichenfolge, abgeschlossen mit CR+LF in das RAM aufzunehmen. Drückt man am Stabilock die Taste MEMORY so ist in der oberen Zeile, diese zeigt den Inhalt des Arbeitsspeicher an, der Eintrag „PC AUTORUN.AUT“ zu sehen. Am Bildschirm von ARE wird in einer, in der Mitte des Displays, erscheinenden Statusanzeige die Anzahl der bisher übertragenen Zeilen heraufgezählt.

Als Erkennung für den Abschluss des Transfer wird die einleitende Zeichenfolge „LEARN programm“, nochmals gesendet.

Ist das Programm vollständig im Arbeitsspeicher des Stabilock kann dieses mittels STORE auch auf einen freien Platz auf die Speicherkarte übertragen werden. Hierbei behält diese Datei den Namen „PC AUTORUN“ – dieser Name kann jedoch mittels der Systemfunktionen des Stabilock innerhalb der Memory Card Anzeige umbenannt werden.

So wie Programme zum Stabilock übertragen werden kann auch ein Re-Transfer von

Autorun-Programmen, die im RAM des Stabilock geladen sind, zum ARE erfolgen. Beim Start des Re-Transfer wird als einleitende Zeichenfolge ein „ALIST“ an den Stabilock gesendet. Bei dem Nachfolgende Lesen auf der IEEE Schnittstelle sendet der Stabilock dann zeilenweise den Inhalt der Autorun Routine aus dem RAM Speicher an die ARE Software. Hierbei wird nach jeder Zeile die Talker und Listener Funktion erneut durchgeführt.



Der Transfer kann mittels des Programm TAP.exe auch in Batchroutinen durchgeführt werden.

Anhang

GPIB Logfile einer Erstinitialisierung von ARE und Stabilock:

IBSTA	IBERR	IBCNTL	GPIB Function Call
0130	1	0	ibsic(GPIB0)
0130	1	0	ibstop(GPIB0)
0130	13	0	ibtmo(GPIB0, T10s)
0130	10	0	ibeos(GPIB0, C0Ah)
0130	0	0	ibsre(GPIB0, 1)
0138	0	4	ibcmd(GPIB0, "?_@6", 4)
0128	0	6	ibwrt(GPIB0, "UNIT..", 6) .. = x0D x0A
0174	0	4	ibcmd(GPIB0, "?_ V", 4)
2164	0	6	ibrd(GPIB0, "4015.....!...7x\$", 255)
0178	0	4	ibcmd(GPIB0, "?_@6", 4)
0178	0	1	ibcmd(GPIB0, ".", 1)
0130	1	0	ibsic(GPIB0)

GPIB Logfile eines Transfer von ARE zum Stabilock

IBSTA	IBERR	IBCNTL	GPIB Function Call
0130	1	0	ibsic(GPIB0)
0130	1	0	ibstop(GPIB0)
0130	13	0	ibtmo(GPIB0, T10s)
0130	10	0	ibeos(GPIB0, C0Ah)
0130	0	0	ibsre(GPIB0, 1)
0138	0	4	ibcmd(GPIB0, "?_@6", 4)
0128	0	12	ibwrt(GPIB0, "LEARN TEST..", 12) .. = x0D x0A
0128	0	19	ibwrt(GPIB0, "0010 REM * TEST *..", 19)
0128	0	12	ibwrt(GPIB0, "0020 SETRX..", 12)
0128	0	28	ibwrt(GPIB0, "0030 RXFREQUENCY 85.155MHz.", 28)
0128	0	7	ibwrt(GPIB0, "0050 ..", 7)
0128	0	12	ibwrt(GPIB0, "LEARN TEST..", 12)
0138	0	1	ibcmd(GPIB0, ".", 1)
0130	1	0	ibsic(GPIB0)

GPIB Logfile eines Re-Transfer vom Stabilock zum ARE

IBSTA	IBERR	IBCNTL	GPIB Function Call
0130	1	0	ibsic(GPIB0)
0130	1	0	ibstop(GPIB0)
0130	13	0	ibtmo(GPIB0, T10s)
0130	10	0	ibeos(GPIB0, C0Ah)
0130	0	0	ibsre(GPIB0, 1)
0138	0	4	ibcmd(GPIB0, "?_@6", 4)
0128	0	7	ibwrt(GPIB0, "ALIST..", 7)
0174	0	4	ibcmd(GPIB0, "?_ V", 4)
2164	0	37	ibrd(GPIB0, "10 REM *** RECEIVER SENSITI", 255)
0174	0	4	ibcmd(GPIB0, "?_ V", 4)
.....			
2164	0	35	ibrd(GPIB0, "170 PRINT "=====", 255)
0174	0	4	ibcmd(GPIB0, "?_ V", 4)
2164	0	7	ibrd(GPIB0, "ALIST..NT "=====", 255)
0178	0	4	ibcmd(GPIB0, "?_@6", 4)
0178	0	1	ibcmd(GPIB0, ".", 1)

Beispiel einer Datei vom Typ *.art und *.alb

```
REM *** RECEIVER SENSITIVITY ***
SETRX
REM *** All values are suggestions ***
REM
RXFREQUENCY 85.155MHz
MODAF 1KHz
RXAFM 1.5KHz
REM *** Jump into the SPECIAL mask ***
SOFT_SPECIAL
SOFT_SENS
SELECT SINAD
WRTVA 21,12dB
SOFT_RUN
REM *** Result is read out in var A ***
A=RESULT 1
PRINT "Receiver Sensitivity: ";A
PRINT "===== "_
^ = x1A
```