INTERFACE SYSTEMS, INC.
462 Jackson Plaza
Ann Arbor, Michigan 48104

(313) 769-5900

USER'S MANUAL for ISI 3274 Type A Tester

Written by : Paul R. Patrick

Revision Date : September 20, 1982
Program Version : 0.16

Table of Contents
Table of Contents

1.0.0 GENERAL :

The ISI 3274 Type A Tester simulates an IBM 3274 Type A Control Unit. The IBM Control Urit is controlled by a host computer and can control up to 32 devices at a time. The ISI Tester is controlled by the operator arc can control only one device.

2.C.C FRONT PANEL OPERATION

To operate the tester from the front panel switches, the test switch on the top board MUST be set to 0 (zero) before power is turned on.

When power is turned on, the POWER LED should light. The READY, CHECK, and HOLD LEDs will be off. A few seconds later, the HOLD LED will light if the power on tests pass. The CHECK LED will light if the power on tests fail. If the CHECK LED lights the tester is defective. If the HOLD LED lights the tester is ready to be used.

To run a test the test number is set in the thumb wheel switches labeled FORM LENGTH. The operator presses the rocker switch labeled TEST to start the test. The READY, CHECK, and HOLD LED will turn off and the CU SIGNAL LED will light if the test number is valid. The alarm will sound if the test number is invalid.

NOTE : The LEDs may appear to remain lit because the test is so fast.

The operator may abort a test in progress by pressing the rocker switch labeled CANCEL.

Upon test completion, the HOLD LED will light. The READY LED will light, if the test passed. The CHECK LED will light, if the test failed. The alarm will sound if the test was aborted and the READY and CHECK LEDs will remain off.

SEE APPENDIX A for description of the test numbers.

If a terminal is connected to the tester the terminal operation messages will be displayed.

2.1.0 LED INDICATORS

The POWER LED on indicates that the tester is on.

The READY LED on with the HOLD LED on indicates that the last test passed.

The CHECK LED on with the HOLD LED on indicates that the last test failed.

The HOLD LED on indicates that the last test was completed or the tester is waiting to run the next test.

The CU SIGNAL LED on with the READY, CHECK, and HOLD LEDs off indicates that a test is running.

The alarm will sound when the operator attempts to run an invalid test number. Also, the alarm will sound, if any rocker switch other any the TEST is pressed.

2.2.0 SWITCHES

The FORM LENGTH thumb wheel switches are used to select the desired test number.

The TEST rocker switch is used to start the test selected by the FORM LENGTH thumb wheel switches.

The CANCEL rocker switch is used to stop a test in process.

2.3.0 DESCRIPTION OF THE TEST NUMBERS

The following list of test numbers are related to the "canned" tests that can be run from the front panel or the terminal in "TEST" mode.

Number	Mnemonic	
00	TNX	Tests with No Print for 4k buffer
01	RST	ReSet Test
02	ACT	Address Counter Tests
03	DWT	Data Word Test
04	ZRX	ZeRo Test for 4k buffer
05	SCX	Search Character Tests for 4k buffer
06	DBX	Device Buffer Tests for 4k buffer
07	ZRT	ZeRo Test for 2k buffer
08	SCT	Search Character Tests for 2k buffer
09	DBT	Device Buffer Tests for 2k buffer
10	TPX	Tests with Prints for 4k buffer
11	CH1	LU1 Character print
12	CH3	LU3 Character print
13	FIX	FIX print tests
14	VAR	VARiable print tests
15	WRX	WRap LU1 prints tests for 4k buffer
16	TP1	LU1 Test Prints
17	LLP	Long Line Prints
18	WRP	WRap LU1 Prints test for 2k buffer
19	TPR	Test PRints for 2k buffer
20	ALX	ALL tests for 4k buffer
21	ALL	ALL tests for 2k buffer
22	BTO	Busy TimeOut test
23	DTO	Device TimeOut test
24	POR	Power On Reset test

3.0.0 Terminal Operation

To operate the tester from a terminal, the test switch on the top board MUST be set to 1 (one) before power is turned on.

Before turning on the tester be sure the terminal is operational.

The message "Power on tests " will be displayed on the terminal when the tester is turned on. Also, the POWER LED on the FRONT PANEL will light and the READY, CHECK, and HOLD LEDs will be off. After a few seconds, the tester's power on tests will complete. If the power on tests pass, the message "Passed !" will be displayed after the message "Power on tests ". If the power on tests fail, the message "Failed ???" will be displayed after the message "Power on tests ".

The version message "ISI3274A Version v.vv month day, year" will be displayed on the next line. The 'v.vv' is the current version number of the test software. The 'month day, year' is the date of the last update of the tester software.

On the next line, the prompt message "RUN>" will be displayed.

EXAMPLE :

```
Power on tests Passed !

ISI3274A Version 1.00 September 8, 1982
RUN>
```

When a prompt message is displayed, the tester is waiting for input from the operator. A prompt message is a three character word ending with a ">" symbol.
The "RUN>" is a prompt message.

The operator may enter a new command line at this time. The tester will not process the new command line until the operator depresses the RETURN key. This is to allow the operator to correct any mistakes in the command line before the tester processes the line.

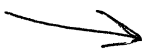
The operator may delete the last character entered by depressing the DELETE key. Each time the DELETE key is depressed the last character entered will be deleted from the command line. If there is no character in the command line to be delet-

ed the tester will respond by displaying the prompt message agair on the next line.

The operator may delete the entire command line by holding the CTRL key down and depressing the 'U' key. The tester will respond by displaying the prompt message again on the next line.

The maximum number of characters that the command line will accept is 127, including the carriage return. When the operator enters the 126 th character, a bell will scund. After the bell sounds, the tester will only accept the following control keys :

RETURN to process the line as entered.
DELETE to delete the last displayed character.
CTRL U to delete the entire line.



The command line accepts input from the operator. The operator must seperate the different entries with a special character refered to as a delimiter. Each delimiter has a unique meaning to the tester.

The COMMA ',' is used to seperate the hex rumbers.
The PERIOD '.' is used to seperate the cecimal numbers.
The SPACE ' ' character is usec to seperate the mnemonic names.
The RETURN character tells the tester to process the commard line.

3.0.1 Modes of Operation

On power up, the tester enters the 'RUN' mode of operation. The tester presently has sever modes of operation. The following is a list of the modes of operation and the operation performed by each :

Build	creates or modifies user macros.
Dump	displays a hex dump of the device's buffer.
Help	displays the help messages.
List	displays the tables or individual user macros.
Move	moves data from one area to another area.
Run	runs a PASS-FAIL user macro by name.
Test	runs a PASS-FAIL user macro by number.

The operator may select any mode of operation by entering the new mode name in response to a prompt.

3.1.0 Run Mode

RUN

RUN will compile the command line into the execute macro buffer 'MA0' and then will execute the macro instructions in the execute macro buffer.
RUN will report any errors detected during the compile or execution of the command line.

Syntax : RUN,<exc>,<parameters>

Where <exc> is any valid execute macro name.
Where <parameters> are the values required by the execute macro.

Example :

```
LIS>RUN JSR ALX          Change mode to 'RUN' and run 'ALX' test
PASSED !                 The 'ALX' test passed !
RUN>ALX                  Re-run 'ALX' test macro
C254 CMD RES = 801       The instruct that failed
C1C1 JSR RST             In the 'RST' test macro
C1B7 JSR TNX             Called by the 'TNX' test macro
C980 JSR ALX             Called by the 'ALX' test macro

RES  813-02  ???-??  = 801      Last coax transmission
FAILED ???               The 'ALX' test failed ?
RUN>
```

3.2.C Help Mode

HELP

Help will display messages to help the operator determine the proper syntax for a given mode of operation.

Syntax : HEL

Example :

```
RUN>HEL                    Display the help messages
•
•
•
'the help messages'
•
•
•
HEL>
```

3.3.0 Test Mode

TEST

Test allows the operatcr to run tests similiar to the frcnt carel operations.

Syntax : TES,<num>

Where <num> is a valid test number.

Example :

```
RUN>TES 20          Run test number '20'
PASSED !            Test rumber '20' passed !
TES>                Wait for next test rumber
```

3.4.0 Dump Mode

DUMP

Dump mode will do a hex dump to the terminal of the device's buffer at the address specified. An entire page of 256 bytes of the device's buffer is displayed at once.

Syntax : DUM,<adr> or <N> or <nul>

Where <adr> is a valid device buffer address.
Where <N> is the letter 'N' will dump the next page.
Where <nul> is a carriage return will dump the same page again.

Example :

```
RUN>DUM,100                Change modes and dump the buffer

C100 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
C110 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
.
.
.
C1E0 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
C1F0 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
DUM>
```

3.5.0 List Mode

LIST

List will the table(s) or user macro(s) specified.

Syntax : LIS,<tab> or <usr>

Where <tab> is the name of the table to list.
Where <usr> is the name of the user macro to list.

Example :

```
RUN>LIS ALX              Change modes and list test macro 'ALX'

ALX                      User macro name
C1B7 JSR TNX             Call user macro 'TNX'
C1BB JSR TPX             Call user macro 'TPX'
C1BF RTS                 Return to Caller
C1C1                     Next address
LIS>
```

3.6.0 Build Mode

BUILD

Build a user macro.

Syntax : BUI,<usr> or <adr>

Where <usr> is a valid user macro name.
Where <adr> is a valid address in the range of a user macro.

Example :

```
RUN>BUI,MAC          Change mode and build user macro 'MAC'
1500>JSR MA1         Call user macro 'MA1'
1504>RTS             Exit building user macro.
BUI>
```

3.7.0 Move Mode

MOVE

Move mode moves data from location to another location within the tester macro area.

Syntax : MOV,<beg>,<nxt>,<to>

Where <beg> is the beginning address of the first byte to move.
Where <nxt> is the next address of the byte after the last byte to be moved.
Where <to> is the starting address to which to move the first byte.

Example :

```
RUN>LIS MA1              List user macro to move

MA1
CA00 JSR MA2             This is the <beg> address
CA04 RTS
CA06                     This is the <nxt> address
LIS>MA2                  List user macro moving to

MA2                      User macro 'MA2' is blank
CB00 RTS                 This is the <to> address
CB02
LIS>MOV A00,A06,B00      Change modes and move data

MOV>
```

Show Result

4.0.0 CONTROL UNIT WORDS

Control unit words fall into two categories, data and command words. Control unit words are 12 bit words (b1-b12) that are transmitted to the device over the coax cable. After the device receives a control unit word from the tester it will send a response word back to the tester over the coax cable. The tester normally does not display the coax transmission while running user macros. If the operator enters CTRL T, the tester will display the coax transmissions in the format below :

EXAMPLE :

```
RUN>ALXT                    Run user macro 'ALX'
RES   813-02   801-00   = 801
      804-00   801-00   = 801
POL   808-01   801-00   = 815
POL   808-01   801-00   = 815
POL   808-01   815-02   = 815
```

The first column is the trarsmittec control unit word mremonic rame. If the word was a command word the command mremonic rame is displayed. If the word was a data word a blank is displayed.

The second column is the trarsmitted control unit word in hex. The 3 hex digits represent bits 1 thru 12 of the word and the 2 hex digits represent bits 2 thru 9 of the word.

The third column is the trarsmittec cevice resporse word in hex. The 3 hex digits represent bits 1 thru 12 of the word and the 2 hex digits represent bits 2 thru 9 of the word.

The fourth column is the word the tester excepts to receive. The 3 hex digits represent bits 1 thru 12 of the word

4.1.0 DATA WORDS

Control unit data words are 12 bit words (b1-b12). The data words contain one byte for information in bits (b2-b9).

4.2.0 COMMAND WORDS

Control unit command words are refered to by a mnemonic name. Control unit command words fall into two categories, read and write command words. The control unit transmits a read command to the device to cause the device to trarsmit data back to the control unit. The control unit transmits a write command to the device to prepare the device to accept data from the control unit.
NOTE : The command mnemonic names for the 'CLC' tester are indicated by "(" ")".

A list of command words by mnemonic name with a brief description.

INB inserts a byte in device buffer.
INV invald command word.
LAH load device's address counter high.
LAL load device's address counter low.
LCR load device's control register.
LME load device's mask register for EAB buffer.
LMK load device's mask register.
PAC acknowledge device's poll response.
PAL sound device's alarm.
PDI disable device.
PEN enable device.
POL read device's poll response register.
RAH read device's address counter high.
RAL read device's address counter low.
RDE read data from device's EAB buffer.
REA read data from device's buffer.
RES reset device.
STA read device's state register.
SBK search backward for pattern.
SFD search forward for pattern.
SOP start operation.
TID read device's terminal identification.
WAE write data to device's EAB buffer.
WME load device's EAB mask register.
WRI write data to device's buffer.
ZER zero device's buffer.

INB

INB commands the device to insert the following control unit data word as a byte into the device's buffer at the location pointed to by the device's address counter.

(BAD) INV

INV is an invalid control unit command word.

LAH

LAH commands the device to load the following control unit data word as a byte into the device's high order address counter.

LAL

LAL commands the device to load the following control unit data word as a byte into the device's low order address counter.

LCR

LCR commands the device to load the following control unit data word as a byte into the device's control register.

LME

LME commands the device to load the following control unit data word as a byte into the device's extended attribute buffer.

LMK

LMK commands the device to load the following control unit data word as a byte into the device's mask register.

(ACK) PAC

PAC commands the device to clear the device's poll response re-

gister. The 'PAC' is sent by the control unit to acknowledge the pending bit(s) set in the poll response register.

(BEL) PAL

PAL commands the device to sound the device's alarm.

(DIS) PDI

PDI commands the device to leave the enable state.

(ENA) PEN

PEN commands the device to enter the enable state.

POL

POL commands the device to trarsmit the contents of the device's poll response register to the control unit.

RAH

RAH commands the device to trarsmit the contents of the high order byte of the device's address counter to the control unit.

RAL

RAL commands the device to trarsmit the contents of the low order byte of the device's address counter to the control unit.

RDE

RDE commands the device to trarsmit the contents of the device alternate buffer location pointed to by the device's address counter to the control unit. The device address counter is advanced to the next location.

(CAT) REA

REA commands the device to trarsmit the contents of the device buffer location pointed to by the device's address counter to the control unit. The device address counter is advanced to the next location.

RES

RES commands the device to reset itself.

STA

STA commands the device to trarsmit the contents of the state register to the control unit.

SBK

SBK commands the device to search backward for the following control unit data word as a byte starting at the location pointed to by the device's address counter.

SFD

SFD commands the device to search forward for the following control unit data word as a byte starting at the location pointed to by the device's address counter.

SOP

SOP commands the device to start the operation specified by the device's control unit output area.

TID

TID commands the device to trarsmit the contents of the device's terminal identification register to the control unit.

WAE

WAE commands the device to write the following control unit data

word as a byte into the device's alternate buffer at the location pointed to by the device's address counter.

WME

WME commands the device to write the following control unit data word as a byte into the device's alternate mask register.

WRI

WRI commands the device to write the following control unit data word as a byte into the device's buffer at the location pointed to by the device's address counter.

(CLR) ZER

ZER commands the device to zero the device's buffer starting at the location pointed to by the device's address counter until the contents of the device's buffer at the current location matches the following control unit data word as a byte under the previously loaded mask register.

5.0.0 EXECUTE MACROS

Execute macros instruct the tester to execute or perform some function or operation. Hence, the name 'EXECUTE MACRO'. The execute macros are executed in sequential order.

All user and user defined macros are made up of one or more execute macros. Each execute macro is refered to by a mnemonic name.

The execute macros can be categorized into the following types of operations or functions :

Execute macros that control the coax transmission and TRANSMIT buffer operations are :

CMD DAT HEX PUT LU1 LU3 RTX STR TXW WRC WRD

Execute macros that control the order of execution are :

FOR IFT JSR PSH PUL REP RPT RTS WHI TIL

Execute macro that control the device address counter are :

CAC DAC LAC RAC PRI PRT

Execute macros that control the PASS/FAIL counter operations are :

CHK DCT FAI PAS ZCT

Execute macros that control terminal operations are :

POF PON TON TOF MSG WOI

Misc. execute macros are :

CPD CPH DLY NOP PRN PRW SUF SUV SU1 ***

A list of execute macros by mremonic rame with a brief description.

CAC compare device address counter.
CAL calls an execute macro.
CHK checks and increments the PASS or FAIL ccunters.
CMD transmits a single command word to the device.
CPD compares the device data word with the passed parameter.
CPH compares the device data words with the hex strirg.
DAC displays the device's address counter.
DAT transmit a single control unit data word to the device.
DCT displays the PASS-FAIL counters.
DLY delays program execution for a specific time in ticks.
FAI increments the FAIL counter.
FOR repeats executing the next instruction for a ccunt.
HEX defines a hex string to be put ir the TX buffer.
IFT conditionally executing the next instruction.
JMP jumps to a specific user macro by name or address.
JSR jumps to a specific user macros by name or address.
LAC loads the device's address ccunter to the address.
LU1 defines a LU1 string to be put in the TX buffer.
LU3 defines a LU3 string to be put in the TX buffer.
MSG defines a ASCII string to be displayed.
NOP does no operation.
PAS increments the PASS counter.
POF disables the display paging.
PON enables the display paging.
PRI sets up the device's control unit area and print.
PRN issues a 'SOP' command.
PRT sets up the device's control unit area and print.
PRW issues a 'SOP' command and waits for 'Crder Complete'.
PSH marks the start of a series of execute macros.
PUL marks the end of a series of execute macros.
PUT sets up the TX buffer for a multi-word transmission.
RAC reads the device's address ccurter.
REP repeats executing the macro 'UNTIL' macro passes.
RPT repeats executing the next macro for specific count.
RTS returns from 'JSR' call.
RTX retransmits the TX buffer.
RXW compares the device's last resporse with the RX word.
STR sets up the TX buffer for a multi-word trarsmission.
SUF sets up the device's control unit area for a FIX print.
SUV sets up the device's control unit area for a VAR print.
SU1 sets up the device's control unit area for a LU1 print.
TIL repeats retransmitting command word 'UNTIL' it passes.
TOF disables displaying the coax trarsmissions.
TON enables displaying the coax trarsmissions.
TXW transmits the multi-word TX buffer to the device.
WHI user macro passes 'DO' user macrc.
WOI waits for operator input.
WRC transmits a command and data word to the device.
WRD sets up the TX buffer for a multi-word trarsmission.
ZCT zeros the PASS and FAIL ccunters.

*** defines a ASCII string as a comment.

The execute macro names can be listed by typing :

```
RUN>LIS EXC<cr>

EXC
CHK     CMD     CPD     CPH
CAC     DAK     DAT     DCT
DLY     FAI     FOR     HEX
IFT     JSR     LAC     LU1
LU3     MSG     NOP     PAS
POF     PON     PRI     PRT
PSH     PUL     PUT     RAC
RPT     RTS     RTX     STR
TIL     TOF     TON     TXW
WOI     WRC     ZCT     ***

LIS>
```

The first execute macro parameter is the address of the assembly language code to execute. This parameter is always present. Depending on the execute macro definition one or more additional parameters may be required. Some execute macros expect one or more execute macros to follow immediately after or must be immediately preceded by a specific execute macro.

The additional parameters can be categorized into the following types.

SYMBOL	TYPE
<exc>	Execute macro
<usr>	User macro
<mac>	User defined macro
<cmd>	Command word
<dat>	Data word
<rec>	Receive word
<byt>	Hex byte
<adr>	Hex address (Device buffer address)
<wrd>	hex word
<hex>	Hex string
<str>	ASCII string

The description of each execute macro is formatted :

<exc>

Description of the use of execute macro and parameters.

Syntax : <exc>,<parameters>

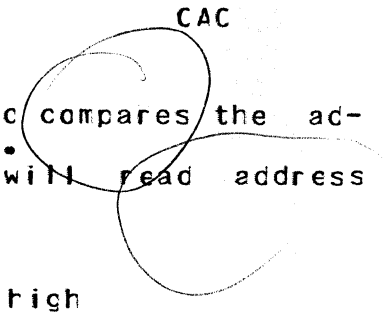
Where <exc> is an execute macro mnemonic name.

Where <parameters> are a list of the additional parameters, if required.

EXAMPLE :

```
BUI>MAC                    Build user defined macro 'MAC'
C001><exc>,<parameters> Execute macro and parameters
C002>RTS          Exit buildling macro 'MAC'
```

CAC



CAC will read the device's address counter and compares the address with the value specified by the address.
CAC calls user macro 'DAK' and then CAC will read address counter high and address counter low.

```
CAC     JSR DAK          Disable the device
        RAH = hhh        Read address counter high
        RAL = lll        Read address counter low
        RTS
```

Where ' = hhh' is the high order of the address word.
Where ' = lll' is the low order of the address word.

Syntax : CAC,<addr>

Where <addr> is a hex value between $000 and $FFF.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>LAC 0000            Load address counter to $000
CA04>CAC 000C            Check address counter is $000
CA08>LAC 0010            Load address counter to $010
CA0C>CAC 0C10            Check address counter is $01C
CA10>LAC 0FFF            Load address counter to $FFF
CA14>CAC 0FFF            Check address counter is $FFF
CA18>RTS                 Exit building macro 'MA1'

BUI>MA2                  Build user defined macro 'MA2'
CB00>LAC 004A            Load address counter to $04A
CB04>STR                 Setup multi-word transmission
CB06>HEX AA 32 74 AA     Control unit area message
CB0D>TXW = 801           Transmit TX buffer to device
CB0F>CAC 004E            Check address counter is $C4E
CB13>RTS                 Exit building macro 'MA2'
BUI>
```

CAL

CAL will execute an execute macro instruction.

Syntax : CAL,<exc>

EXAMPLE :

```
BUI>MA1
CA00>CAL EX1              Call user macro expression  1
CA04>RTS
BUI>LIS EX1

EX1
D96D CMD PCL = 801
D973 RTS
D975

LIS>
```

CHK

CHK will execute the next excute macro. If PASS-FAIL flag is set passed, the PASS counter is incremented. If PASS-FAIL flag is set failed, the FAIL ccurter is incremented. If the PASS-FAIL flag is set tc neither, nc ccunter is incremented and the execute macro is aborted. If either ccunter has overflowec, the execute macro is aborted.

Syntax : CHK

EXAMPLE :

```
BUI>MA1                 Build a user defined macro 'MA1'
CA00>CHK                Check PCLL resporse equal to 801
CAC2>CMD PCL = 801      Trarsmit poll ccmmand anc compare
CAC8>RTS                Exit build macro 'MA1'

BUI>MA2                 Build a user defined macro 'MA2'
CB00>RPT 0000           Repeat checkinc 'ALX' until aborted
CB04>CHK                Check user macro 'ALX' for pass or fail
CB06>JSR ALX            Call user macro 'ALX'
CBCA>RTS                Exit buildinc macro 'MA2'

BUI>MA3                 Build user defined macrc 'MA3'
CC00>CHK                Check the address ccunter for zero
CC02>PSH                Repeat a series of execute macros
CC04>CMD RAH = 804      Transmit reac address high and compare
CCCA>CMD RAL = 804      Trarsmit reac address low anc compare
CC10>PUL                End of series of execute macros
CC12>RTS                Exit buildinc macro 'MA3'
BUI>
```

CMD

CMD transmits a control unit command word as a single word transmission and compares the device's resporse word with the receive word.

Syntax : CMD,<cmd>,<opr>,<rxw>

Where <cmd> is the control unit command word to transmit.

Where <opr> is the receive word operator. The = compares that the response word is equal to the receive word. The # compares that the response word is not equal to the receive word. The - does no compare of the words.

Where <rxw> is the word that the resporse word is compared with.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>CMD REA = 804         Trarsmit read data command and compare
0A06>CMD POL # 815         Transmit poll command anc compare
0A0C>CMD STA - 801         Transmit State command with no compare
0A12>RTS                   Exit building macro 'MA1'
BUI>
```

CPD

CPD reads the data word at the device address counter and compares b2-b9 of the data word with the value passed by the previous execute macro.

Syntax : CPD

Example :

```
BUI>MA4                  Build user defined macro 'MA4'
0DC0>FOR 00 FF 01        For a value of $C0 to $FF by $C1
0DC5>CPD                 Compare data words
0DC7>RTS                 Exit building macro 'MA4'
BUI>
```

CPH

CPH compares a string of hex bytes with a string of data words pointed to by the device address counter. The maximum number of hex bytes in a string is 41.

Syntax : CPH <byte1>,<byte2>,<byte3>,....<byteN><cr>

Where <byteN> is a hex byte.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>LAC 004A            Load address counter to $4A
CA04>CPH AA 32 74 AA     Compare hex byte with control unit area
CA0B>RTS                 Exit building macro 'MA1'
BUI>
```

DAC

DAC will display the message 'Device Address Counter $xxxx' Where 'xxxx' is the hex value of the device address counter. DAC calls execute macro 'DAK' and then DAC will issuse a read address counter high command and a read address counter low command.

Syntax : DAC

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>DAC                   Display device address counter
0A02>RTS                   Exit building macro 'MA1'
BUI>
```

CAT

CAT transmits a control unit data word as a single word transmission and compares the resporse word with the receive word. The only valid receive word is '= 801' for a control unit cata word.

Syntax : CAT <dat>,<opr>,<rxw>

Where <dat> is the control unit cata word to trarsmit.

Where <opr> is the receive word operator. The = compares that the response word is equal to the receive word. The # compares that the response word is not equal to the receive word. The - does no compare of the words.

Where <rxw> is the word that the resporse word is compared with.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>DAT 00 = 801        Transmit data word ($00) and compare
CA06>DAT FF = 801        Transmit data word ($FF) and compare
CA0C>RTS                 Exit building macro 'MA1'
BUI>
```

DCT

DCT will display the message

'PASSED $pppppppp + FAILED $ffffffff = TOTAL $ttttttttt'

Where 'pppppppp' is the passed hex count.
Where 'ffffffff' is the failed hex count.
Where 'ttttttttt' is the totaled passed and failed hex count.

Syntax : DCT

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>DCT                Display the pass/fail count
CA02>RTS                Exit building macro 'MA1'
BUI>
```

DLY

DLY will delay executing other execute macros for a specific time. The maximum delay time is just over 1 minute.

Syntax : DLY,<hex>

Where <hex> is the number of ticks in hex.
NOTE : Each tick is equal to 1 millisecond.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>DLY 0000           No delay
CA04>DLY 0001           1 ms delay
CA08>DLY 000A           10 ms delay
CA0C>DLY 0064           100 ms delay
CA10>DLY 03E8           1.0 second delay
CA14>DLY 2710           10.0 second delay
CA18>DLY EA60           60.0 second delay
CA1C>DLY FFFF           65.535 second delay (max.)
CA20>RTS                Exit building macro 'MA1'
BUI>
```

FAI

FAI will increment the FAIL counter. A use of FAI would be to keep track of the number of times a user defined macro is executed.

Syntax : FAI

Example :

```
BUI>MAA                  Build user defined macro 'MAA'
1300>CMD PAC = 801       Acknowledge condition
1306>FAI                 Increment FAIL counter
1308>RTS                 Exit building macro 'MAA'
BUI>
```

FOR

FOR will repeat executing the next execute macro from the beginning value to the ending value in step increments.

Syntax : FOR,<beg>,<end>,<stp>

Where <beg> is the beginning value ir hex.
Where <end> is the ending value in hex.
Where <stp> is the step value in hex.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>PUT 0050           Setup multi data word trarsmission
0A04>FOR 00,FF,01       For loop
CA09>HEX                Put hex character in TX buffer
CA0C>TXW = 8C1          Transmit TX buffer
0A10>RTS                Exit building macro 'MA1'
BUI>
```

HEX

HEX will put a string of hex bytes in the TX buffer. HEX MUST be preceded by the execute macro 'PUT' or 'STR'. HEX MUST be followed by the execute macro 'TXW' to trarsmit the TX buffer to the device buffer.

Syntax : HEX,<byte1>,<byte2>,<byte3>,...,<byteN><cr>

Where <byteN> is a hex byte.
Where <cr> is a carriage return.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>STR                Setup a multi data word transmission
CA02>HEX AA 32 74 AA    Put hex string in TX buffer
CA09>TXW = 801          Transmit the TX buffer
CA0D>RTS                Exit building macro 'MA1'
BUI>
```

IFT

IFT will first execute the EXPRESSION user macro. If the EXPRESSION user macro passes then the THEN user macro will be executed else the ELSE user macro will be executed.

Syntax : IFT,<user1>,<user2>,<user3>

Where <user1> is the EXPRESSION user macro.
Where <user2> is the THEN user macro.
Where <user3> is the ELSE user macro.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>CMD POL = 801       Is the poll equal to 801 ?
CA06>RTS                 Exit building macro 'MA1'

BUI>MA2                  Build user defined macro 'MA2'
CB00>PAS                 Yes, increment the PASS counter
CB02>RTS                 Exit building macro 'MA2'

BUI>MA3                  Build user defined macro 'MA3'
CC00>FAI                 No, increment the FAIL counter
CC02>RTS                 Exit building macro 'MA3'

BUI>MAC                  Build user defined macro 'MAC'
1500>RPT 0000            Repeat testing the poll response
1504>IFT MA1 MA2 MA3     If MA1 then MA2 else MA3
150C>RTS                 Exit building macro 'MAC'
BUI>
```

JMP

JMP will execute a user macro. A user macro is a series of execute macros.

Syntax : JMP,<user>or<addr>

Where <user> is the user macro to execute.
Where <addr> is the valid user macro address to execute.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>JMP ACK            Jumps to user macro 'ACK'
CA04>JMP MAC            Jumps to user defined macro 'MAC'
CA08>RTS                Exit building macro 'MA1'
BUI>
```

JSR

JSR will execute a user macro. A user macro is a series of execute macros.

Syntax : JSR,<user>

Where <user> is the user macro to execute.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>JSR ACK            Call user macro 'ACK'
CA04>JSR MAC            Call user defined macro 'MAC'
CA08>RTS                Exit building macro 'MA1'
BUI>
```

LAC

LAC will load the device's address counter with the value specified by the address.
LAC calls user macro 'DAK' and then LAC will write address counter high and address counter low.

```
LAC     JSR DAK          Disable the device
        LAH hh = 801     Write address counter high
        LAL ll = 801     Write address counter low
        RTS
```

Where 'hh' is the high order of the address.
Where 'll' is the low order of the address.

Syntax : LAC,<addr>

Where <addr> is a hex value between $000 and $FFF.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>LAC 0000            Load address counter to $000
CA04>RAC 0000            Check address counter is $00C
CA08>LAC 0010            Load address counter to $010
CA0C>RAC 0010            Check address counter is $01C
CA10>LAC 0FFF            Load address counter to $FFF
CA14>RAC 0FFF            Check address counter is $FFF
CA18>RTS                 Exit building macro 'MA1'

BUI>MA2                  Build user defined macro 'MA2'
CB00>LAC 004A            Load address counter to $04A
CB04>STR                 Setup multi-word transmission
CB06>HEX AA 32 74 AA     Control unit area message
CB0D>TXW = 801           Transmit TX buffer to device
CB0F>RAC 004E            Check address counter is $04E
CB13>RTS                 Exit building macro 'MA2'
BUI>
```

LU1

LU1 is an ASCII string of character that is translated into LU1 characters and then stored in the TX buffer. LU1 MUST be preceded by the execute macro 'PUT' or 'STR'. LU1 MUST be followed by the execute macro 'TXW' to transmit the TX buffer to the device buffer.

Syntax : LU1,<delimiter><string><delimter>

Where <delimiter> is any printable ASCII character.

Where <string> is any ASCII characters except <cr> and the delimiter character.

The following control characters are translatec to special LU1 control codes :

CTRL A is translated into a END OF MESSAGE code.
CTRL B is translated into a FORM FEED code.
CTRL C is translated into a NEW LINE code.
CTRL D is translated into a LINE FEED code.
CTRL E is translated into a CARRIAGE RETURN code.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>STR                   Setup a multi cata word transmission
0A02>LU1 'characters'      Put LU1 string in TX buffer
0A10>TXW = 801             Transmit the TX buffer
0A14>RTS                   Exit building macro 'MA1'
BUI>
```

LU3

LU3 is an ASCII string of character that is translated into LL3 characters and then stored in the TX buffer. LU3 MUST be preceded by the execute macro 'PUT' or 'STR'. LU3 MUST be followed by the execute macro 'TXW' to transmit the TX buffer to the cevice buffer.

Syntax : LU3,<delimiter><string><delimter>

Where <delimiter> is any printable ASCII character.

Where <string> is any ASCII characters except <cr> and the delimiter character. The following control characters are translated to special LU3 control codes :

CTRL A is translated into a END OF MESSAGE code.
CTRL B is translated into a FORM FEED code.
CTRL C is translated into a NEW LINE code.
CTRL D is translated into a NEW LINE code.
CTRL E is translated into a CARRIAGE RETURN code.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>STR                 Setup a multi data word transmission
0A02>LU3 'characters'    Put LU3 string in TX buffer
CA10>TXW = 801           Transmit the TX buffer
CA14>RTS                 Exit building macro 'MA1'
BUI>
```

MSG

MSG is an ASCII string of character that is displayed when a user macro is run.

NOTE : These messages are displayed whether or not the coax transmissions are being displayed.

Syntax : MSG,<delimiter><string><delimter>

Where <delimiter> is any printable ASCII character.

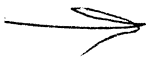
Where <string> is any ASCII characters except <cr> and the delimiter character. The following control characters are translated to special ASCII control codes :

CTRL A is translated into a END OF MESSAGE code.
CTRL B is translated into a FORM FEED code. The FORM FEED will output 3 LINE FEEDS.
CTRL C is translated into a NEW LINE code.
CTRL D is translated into a LINE FEED code.
CTRL E is translated into a CARRIAGE RETURN code.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>MSG /Enabled and Idling/   Operator message
0A16>CMD PEN = 801         Enable device
0A1C>RTS                   Exit building macro 'MA1'
BUI>
```

NOP



NOP does no operation. It is used to leave room in user macros or to create a very short execution delays. It can also be used to "Wipe out" part of a user macro.

Syntax : NOP

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>NOP                   No operation
CA02>RTS                   Exit building macro 'MA1'
BUI>
```

PAS

PAS will increment the PASS counter. A use of PAS would be to keep track of the number of times a user defined macro is executed.

Syntax : PAS

Example :

```
BUI>MAA                    Build user defined macro 'MAA'
1300>CMD PAC = 801         Acknowledge condition
1306>PAS                   Increment PASS counter
1308>RTS                   Exit building macro 'MAA'
BUI>
```

POF

POF will disable paging the terminal display. It does not disable displaying coax transmissions.

Syntax : POF

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>POF                Disable paging
CAC2>RTS                Exit building macro 'MA1'
BUI>
```

PON

PON will enable paging the terminal display. It will also enable displaying the coax trarsmissions. It is used to display the coax transmissions from user macros.

Syntax : PON

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CAC0>PON                   Enable paging and display
0A02>RTS                   Exit building macro 'MA1'
BUI>
```

PRI

PRI will setup the device's control unit output area for a print and then send a 'SOP' control unit command.
PRI does NOT wait for 'Order Complete. It leaves the device enabled and executing the device command.

Syntax : PRI,<mode>,<msa>,<msl>,<orcer>,<mpp>

Where <mode> is a 2 byte hex mode rumber.
Where <msa> is a 2 byte hex message starting address.
Where <msl> is a 2 byte hex message length.
Where <order> is a 2 byte hex order rumber.
Where <mpp> is a 1 byte hex maximum presentation position.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>PUT 0050              Setup for multi word trarsmission
CA04>LU3 "Test messsage"          Put LU3 string in TX buffer
CA15>TXW = 801             Transmit words in device
CA19>RTS                   Exit building macro 'MA1'

BUI>MA2                    Build user defined macro 'MA2'
CB00>PRI 0001,0050,000D,0300,84 Do the print
CB11>*** /Does not check print complete/
CB31>RTS                   Exit building macro 'MA2'
BUI>
```

PRN

PRN will send a 'SOP' control unit command. It assumes that the device's control unit area is already setup.

Syntax : PRN

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>PUT 0050              Setup for multi word trarsmission
0A04>LU3 "Test messsage"          Put LU3 string in TX buffer
0A15>TXW = 801             Transmit words in device
0A19>SUV                   Set up for Variable LU3 print
0A1B>RTS                   Exit building macro 'MA1'

BUI>MA2                    Build user defined macro 'MA2'
0B00>PRN                   Do the print now
0B11>*** /Does not check print complete/
0B31>RTS                   Exit building macro 'MA2'
BUI>
```

PRT

PRT will sets up the Device's control unit output area for a print and then send a 'SOP' control unit command. PRT will poll the device until a new status is availiable. It will check for order complete and acknowledge.

Syntax : PRT,<mode>,<msa>,<msl>,<order>,<mpp>

Where <mode> is a 2 byte hex mode number.
Where <msa> is a 2 byte hex message starting address.
Where <msl> is a 2 byte hex message length.
Where <order> is a 2 byte hex order number.
Where <mpp> is a 1 byte hex maximum presentation position.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>PUT 0050              Setup for multi word transmission
CA04>LU3 "Test messsage"          Put LU3 string in TX buffer
CA15>TXW = 801             Transmit words in device
CA19>RTS                   Exit building macro 'MA1'

BUI>MA2                    Build user defined macro 'MA2'
CB00>PRT 0001,0050,000D,0300,84 Do the print
CB11>*** /Does check print complete/
CB31>RTS                   Exit building macro 'MA2'
BUI>
```

PRT 6,50,0,3400,0 SET BEGINNING OF CHAIN

PRT 6,50,0,3200,0 SET END OF CHAIN

PRW

PRW will send a 'SOP' control unit command to the device. PRW will poll the device until a new status is availiable. It will check for order complete and acknowledge. It assumes that the device's control unit area is setup.

Syntax : PRW

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>PUT 0050              Setup for multi word trarsmission
CA04>LU3 "Test messsage"          Put LU3 string in TX buffer
CA15>TXW = 801             Transmit words in device
CA19>SUV                   Set up for Variable LU3 print
CA1B>RTS                   Exit building macro 'MA1'

BUI>MA2                    Build user defined macro 'MA2'
CB00>PRW                   Do the print and wait
CB11>*** /Does check print complete/
CB31>RTS                   Exit building macro 'MA2'
BUI>
```

PSH

PSH is used to define a series of execute macros as one execute macro. PSH MUST be followed by the execute macro 'PUL'.

Syntax : PSH

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>CHK                   Check the series of execute macros
0A02>PSH                   Mark the start of the series
0A04>CMD PEN = 801         Enable the device
0A0A>CMD STA = 801         Verify that device is enabled
0A10>PUL                   Mark the last of the series
0A12>RTS                   Exit building macro 'MA1'
BUI>
```

PUL

PUL is used to define a series of execute macros as one execute macro. PUL MUST be preceded by the execute macro 'PSH'.

Syntax : PUL

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>CHK                   Check the series of execute macros
CAC2>PSH                   Mark the start of the series
0A04>CMD PEN = 801         Enable the device
0A0A>CMD STA = 801         Verify that device is enabled
CA10>PUL                   Mark the last of the series
CA12>RTS                   Exit building macro 'MA1'
BUI>
```

PUT

PUT will load the device's address counter with the value specified by the address. PUT will setup the TX buffer for a multi-word transmission. A 'WRI' command word will always be the first word to be trarsmitted. PUT MUST be followed by the execute macro 'TXW' to trarsmit the TX buffer to the device.

Syntax : PUT,<addr>

Where <addr> is a hex value between $CCC and $FFF.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>PUT 0101              Start multi-word trarsmission
CA04>LU1 'Print message'         LU1 print message
CA15>TXW = 801             Transmit TX buffer
CA19>RTS                   Exit building macro 'MA1'
BUI>
```

RAC

RAC will read the device's address counter and setup the tester variable 'DEVADR'.
RAC checks device is disabled and then RAC will read address counter high and address counter low.

```
RAC     STA = 900       Check the device is disabled
        RAH - hhh       Read address counter high
        RAL - lll       Read address counter low
        RTS
```

Where 'hhh' is the high order of the address word.
Where 'lll' is the low order of the address word.

Syntax : RAC

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>LAC 0000           Load address counter to $000
CA04>RAC                Read address counter is $000
CA08>LAC 0010           Load address counter to $010
CA0C>RAC                Read address counter is $010
CA10>LAC 0FFF           Load address counter to $FFF
CA14>RAC                Read address counter is $FFF
CA18>RTS                Exit building macro 'MA1'

BUI>MA2                 Build user defined macro 'MA2'
CB00>LAC 004A           Load address counter to $04A
CB04>STR                Setup multi-word transmission
CB06>HEX AA 32 74 AA    Control unit area message
CB0D>TXW = 801          Transmit TX buffer to device
CB0F>RAC                Read address counter is $04E
CB13>RTS                Exit building macro 'MA2'
BUI>
```

what is it good for

REP

REP will first execute the DO user macro. and then the EXPRESSION user macro will be executed. Until the EXPRESSION user macro fails the DO user macro will executed.

Syntax : REP,<user1>,<user2>

Where <user1> is the DO user macro.
Where <user2> is the EXPRESSION user macro.

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>CMD POL - 801      Poll the device
CA06>PAS                Increment the PASS counter
CA08>RTS                Exit building macro 'MA1'

BUI>MA2                 Build user defined macro 'MA2'
CB00>RXW # 801          Is the poll not equal to 801 ?
CB06>RTS                Exit building macro 'MA2'

BUI>MAC                 Build user defined macro 'MAC'
1500>ZCT                Zero the PASS-FAIL counters
1502>CMD PEN = 801      Enable the device
1508>CMD PDI = 801      Disable the cevice
150E>REP MA1 MA2        Repeat MA1 until MA2
1514>DCT                Display PASS-FAIL counters
1516>CMD PAC = 801      Acknowledge poll disable
151C>RTS                Exit building macro 'MAC'
BUI>
```

RPT

RPT will execute the next execute macro for the number of times specified by the count.
NOTE : If the count is zero, the next execute macro will be run indefinitely.

Syntax : RPT,<cnt>

Where <cnt> is the repeat count in hex.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
CA00>RPT 000A            Repeat 10 times
CA04>DAT 00 = 801        Transmitting data words
CA0A>RTS                 Exit building macro 'MA1'
BUI>
```

RTS

RTS will return the execution to the user macro that called this user macro. It is used to mark the end of a user macro.

Syntax : RTS

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>LAC 0050           Load address counter to $050
CAC2>RTS                Exit building macro 'MA1'
BUI>
```

RTX

RTX will re-transmit the last TX buffer to the cevice. RTX MLST be preceded by a valid series of execute macros that setup the TX buffer.

Syntax : RTX

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
0AC0>JSR SCP            Transmit Search Character Pattern
CA04>RPT 0007           Repeat 7 times
CA08>RTX                Re-trarsmitting 'SCP'
CACA>RTS                Exit building macro 'MA1'
BUI>
```

RXW

RXW will compare the last device response to the receive word.

Syntax : RXW,<opr>,<rxw>

Where <opr> is the receive word operator. The = compares that the last response word is equal to the receive word. The # compares that the last response word is not equal to the receive word. The - does no compare of the words.

Where <rxw> is the word that the last resporse word is compared with.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>TIL                   Wait for the 'POL' to change
0A02>CMD POL,801           from '801'
0A08>RXW = 808             Check last resporse 'Operation Complete'
0A0C>RTS
BUI>
```

STR

STR will setup the TX buffer for a multi-word transmission. A 'WRI' command word will always be the first word to be transmitted. STR MUST be followed by the execute macro 'TXW' to transmit the TX buffer to the device. The message is stored at the location pointed to by the device's address counter.

Syntax : STR

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CACO>STR                   Start multi-word transmission
CA04>LU1 'Print message'         LU1 print message
CA15>TXW = 801             Transmit TX buffer
CA19>RTS                   Exit building macro 'MA1'
BUI>
```

STR DOES NOT CONTROL THE SETTING OF THE DEVICE ADDR CTR.

SUF

SUF will setup the device's control unit area for a fixed LU3 print. SUF MUST be preceded by the execute macro 'PUT'. 'PUT' sets up the 'MSA'. SUF sets up the 'MSL' by reading the device's address counter.

Syntax : SUF

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>PUT 100            Start multi-word trarsmission
CA04>LU3 /ABC/          LU3 message to print
CA0A>TXW = 801          Trarsmit TX buffer
CA0C>SUF                Set up C.U. area for LU3 print
CA0E>PRW                Do the prirt and wait
CA10>RTS                Exit building macro 'MA1'
BUI>
```

SUV

SUV will setup the device's control unit area for a variable LU3 print. SUV MUST be preceded by the execute macro 'PUT'. 'PUT' sets up the 'MSA'. SUV sets up the 'MSL' by reading the device's address counter.

Syntax : SUV

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA0C>PUT 100            Start multi-word transmission
CA04>LU3 /ABC/          LU3 message to print
CA0A>TXW = 801          Transmit TX buffer
CA0C>SUV                Set up C.U. area for LU3 print
CA0E>PRW                Do the print and wait
CA10>RTS                Exit building macro 'MA1'
BUI>
```

SU1

SU1 will setup the device's control unit area for a LU1 print. SU1 MUST be preceded by the execute macro 'PUT'. 'PUT' sets up the 'MSA'. SU1 sets up the 'MSL' by reading the device's address counter.

Syntax : SU1

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
0AC0>PUT 100            Start multi-word transmissior
0AC4>LU1 /ABC/          LU1 message to print
0ACA>TXW = 801          Transmit TX buffer
0ACC>SU1                Set up C.U. area for LU1 print
0ACE>PRW                Do the print and wait
0AD0>RTS                Exit building macro 'MA1'
BUI>
```

NOTE: THIS ROUTINE ALSO SETS MPP TO 132.

TIL

TIL will repeat re-transmitting the next 'CMD' execute macro until the device's response is the same as the excepted receive word. The next execute macro MUST be a 'CMD' excute macro.

Syntax : TIL
CMD <cmd>,<opr>,<rxw>

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
0A00>CMD PDI = 801         Disable the cevice
CA06>TIL                   Wait for
CA08>CMD STA # 801         Device not enabled
CA0E>CMD PAC = 801         Acknowledge Poll disable
CA14>CMD STA = 900         Check device is disabled
0A16>RTS                   Exit building macro 'MA1'
BUI>
```

TOF

TOF will disable displaying the coax transmissions.

Syntax : TOF

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>TOF                   Disable the display
CA02>RTS                   Exit building macro 'MA1'
BUI>
```

TON

TON will enable displaying the coax trarsmissions. If paging was previously enable, it will still be honored.

Syntax : TON

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>TON                   Enable display
CA02>RTS                   Exit building macro 'MA1'
BUI>
```

TXW

TXW completes the multi-word trarsmission setup and then transmits the words in TX buffer to the device. TXW MUST be preceded by the execute macro 'PUT' or 'STR'.

Syntax : TXW,<opr>,<rxw>

Where <opr> is the receive word operator. The = compares that the response word is equal to the receive word. The # ccmpares that the response word is not equal to the receive word. The - does no compare of the words.

Where <rxw> is the word that the resporse word is compared with.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>PUT 0101              Start multi-word trarsmission
CA04>LU1 'Print message'          LU1 print message
CA15>TXW = 801             Transmit TX buffer
CA19>RTS                   Exit building macro 'MA1'
BUI>
```

WHI

WHI will first execute the EXPRESSION user macro. If the EXPRESSION user macro passes, then the THEN user macro will be executed.

Syntax : WHI,<user1>,<user2>

Where <user1> is the EXPRESSION user macro.
Where <user2> is the THEN user macro.

Example :

```
BUI>MA1                  Build user defined macro 'MA1'
0AC0>CMD POL # 840       Is the poll rot equal to 840 ?
0A06>RTS                 Exit building macro 'MA1'

BUI>MA2                  Build user defined macro 'MA2'
0B00>DAK                 Disable the device
0B02>CMD PEN = 801       Enable the device
0B08>RTS                 Exit building macro 'MA2'

BUI>MAC                  Build user defined macro 'MAC'
1500>WHI MA1 MA2                  While MA1 then MA2
1506>RTS                 Exit building macro 'MAC'
BUI>
```

WOI

WOI will sound the alarm and wait for operator input. It should be used in conjunction with execute macro 'MSG'.

Syntax : WOI

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>MSG /Connect the coax cable/       Operator message
CA1A>WOI                   Wait for operator to acknowledge
CA1C>RTS                   Exit building macro 'MA1'
BUI>
```

WRC

WRC transmits the control unit command and data word as one multi-word transmission and compares the response word with the receive word. The only valid receive word is '= 801' for a control unit write command.

Syntax : WRC,<cmd>,<dat>,<opr>,<rxw>

Where <cmd> is the control unit command word to transmit.

Where <dat> is the control unit data word to trarsmit.

Where <opr> is the receive word operator. The = compares that the response word is equal to the receive word. The # compares that the response word is not equal to the receive word. The - does no compare of the words.

Where <rxw> is the word that the response word is compared with.

Example :

```
0A00>WRC LAL 50 = 801    Transmit Load A. C. with $50
CA08>WRC SFD A5 = 801    Transmit Search Forward for $A5
CA10>RTS                 Exit building macro 'MA1'
BUI>
```

WRD

WRD sets up the TX buffer for a multi-data word transmission. WRD MUST be followed by the execute macro 'TXW' to transmit the TX buffer to the device.

Syntax : WRD

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>CMD WRI = 801         Write data word command
CA06>WRD                   Set up to transmit multi-data word
CA08>HEX 1,2,3,4           Hex data to send
CA0F>TXW = 801             Transmit TX buffer
CA11>RTS                   Exit building macro 'MA1'
BUI>
```

ZCT

ZCT will zero the PASS and FAIL counters.

Syntax : ZCT

Example :

```
BUI>MA1                 Build user defined macro 'MA1'
CA00>ZCT                Clear PASS and FAIL counters
CA02>RPT 0000           Repeat user macro
CA06>JSR MAC            Call user macro 'MAC'
CA0A>RTS                Exit building macro 'MA1'
BUI>
```

***

*** is an ASCII string of character that is displayed when a user macro this listed. It is to enteddec comments into a user macro.

Syntax : ***,<delimiter><string><delimter>

Where <delimiter> is any printable ASCII character.
Where <string> is any ASCII characters except <cr> and the delimiter character.

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>*** /Wait for power on/    Comment
CA15>TIL
CA17>CMD POL = 815              Transmit a poll
CA2D>RTS                   Exit building macro 'MA1'
BUI>
```

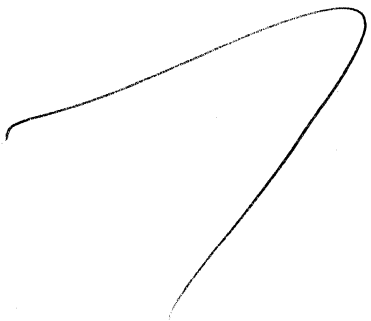
XXX

Description

Syntax : XXX

Example :

```
BUI>MA1                    Build user defined macro 'MA1'
CA00>
CA02>RTS                   Exit building macro 'MA1'
BUI>
```



A list of user macros by mremonic rame with a brief cescription.

ABO aborts the print in progress.
ACK acknowledge order complete and enable device.
ACT address counter test.
AKT acknowledge test.
ALL 2K device buffer print and no print tests.
ALX 4K device buffer print and no print tests.
BPT big character print test.
BTO busy time out test.
CH1 LU1 character table print.
CH3 LU3 character table print.
CLB 2K device buffer clear buffer test.
CLX 4K device buffer clear buffer test.
CS1 LU1 character set.
CS3 LU3 character set.
DAK disable device and acknowledge.
DBT 2K device buffer memory diagnostic test.
DBX 4K device buffer memory diagnostic test.
DEP disable and enable print.
DO1 user macro DO 1.
DTO de-select time out test.
DWT data word test.
EL2 user macro ELSE 2.
EL3 user macro ELSE 3.
ENS enable device and check device is enabled.
EX1 user macro EXPRESSION 1.
EX2 user macro EXPRESSION 2.
EX3 user macro EXPRESSION 3.
FIX LU3 fixed print test.
HP1 select hopper one print.
HP2 select hopper two print.
HPT hopper page test print.
LC1 lower case letter in LU1.
LC3 lower case letters in LU3.
LLP long line print test.
LL1 long line message in LU1.
LL3 long line message in LU3.
MOD set address counter to mode.
MPP set address counter to max. presentation position.
MSA set address counter to message starting address.
MSL set address counter to message length.
NC1 number characters message in LU1.
NC3 number characters message in LU3.
ORD set address counter in order.
POR power on test.
RMD read multiple data (00-0F).
RST reset device test.
SB1 search backward test.
SCP search character pattern.
SCT 2K device buffer search character test.
SCX 4K device buffer search character test.
SF1 search forward test.

SL1 short line message in LU1.
SL3 short line message in LU3.
TH2 user macro THEN 2.
TH3 user macro THEN 3.
TNP 2K device buffer no printing test.
TNX 4K device buffer no printing test.
TP1 LU1 test print.
TPR 2K device buffer printing test.
TPX 4K device buffer printing test.
UC1 upper case letter message in LU1.
UC3 upper case letter message in LU3.
VAR LU3 variable print test.
WMD write multiple data (00-0F).
WMF write multiple FF's.
WAI wait for operation complete in poll response.
WCC wait for order complete.
WPO wait for power on complete in poll resporse.
WRP 2K device buffer LU1 wrap print test.
WRX 4K device buffer LU1 wrap print test.
WSA wait for status available in poll response.
ZRT 2K device buffer zero buffer test.
ZRX 4K device buffer zero buffer test.

ABC

```
ABC
 PUT 0016 SET ABORT ORDER
 HEX 01 00 ABORT ORDER
 TXW = 801 TRANSMIT ORDER
 CMD SOP = 801 START OPERATION
 JSR WOC WAIT FOR ORDER COMPLETE
 RTS
```

ACK

```
ACK
 CMD PAC = 801 ACKNOWLEDGE ANY PREVICUS POLL
 CMD PEN = 801 ENABLE DEVICE
 CMD STA = 801 CHECK DEVICE IS ENABLED
 CMD PDI = 801 DISABLE DEVICE
 TIL WAIT FOR
 CMD POL   801
 RXW = 808 OPERATION COMPLETE
 CMD PAC = 801 ACKNOWLEDGE OPER COMPLETE
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 CMD STA = 900 CHECK DEVICE IS DISABLED
 CMD RAH = 804 CHECK ADDR CNTR HIGH = 00
 CMD RAL = 804 CHECK ADDR CNTR LOW  = 0C
 CMD REA = 804 CHECK DEVICE STATUS = 00
 CMD PEN = 801 ENABLE DEVICE AGAIN
 CMD STA = 801 CHECK DEVICE IS ENABLED
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 RTS
```

ACT

```
ACT
 LAC FFFF
 CAC 0FFF
 LAC EEEE
 CAC 0EEE
 LAC DDDD
 CAC 0DDD
 LAC CCCC
 CAC 0CCC
 LAC BBBB
 CAC 0BBB
 LAC AAAA
 CAC 0AAA
 LAC 9999
 CAC 0999
 LAC 8888
 CAC 0888
 LAC 7777
 CAC 0777
 LAC 6666
 CAC 0666
 LAC 5555
 CAC 0555
 LAC 4444
 CAC 0444
 LAC 3333
 CAC 0333
 LAC 2222
 CAC 0222
 LAC 1111
 CAC 0111
 LAC 0000
 CAC 0000
 LAC AA55
 CAC 0A55
 LAC 55AA
 CAC 05AA
 LAC 0000
 CAC 0000
 WRC LAL 50 = 801
 CAC 0050
 WRC LAH 05 = 801
 CAC 0550
 LAC 0050
 CAC 0050
 RTS
```

AKT

```
AKT
 CMD PAC = 801 CLEAR POLL REGISTER
 CMD POL = 801 CHECK POLL REGISTER CLEARED
 RTS
```

ALL

```
ALL
 JSR TNP NO PRINT TESTS
 JSR TPR PRINT TESTS
 RTS
```

ALX

```
ALX
 JSR TNX NO PRINT TESTS
 JSR TPX PRINT TESTS
 RTS
```

BPT

```
BPT
 PUT 0050
 HEX F0,NL
 LU3 "%P0100100A 01 INCH LINE",NL
 LU3 "%P0200100A 02 INCH LINE",NL
 LU3 "%P0400100A 04 INCH LINE",NL
 LU3 "%P0500100A 05 INCH LINE",NL
 LU3 "%P0600100A 06 INCH LINE",NL
 LU3 "%P0700100A 07 INCH LINE",NL
 LU3 "%P0800100A 08 INCH LINE",NL
 LU3 "%P1000100A 10 INCH LINE",NL
 LU3 "%P1200100A 12 INCH LINE",NL
 LU3 "%P1400100A 14 INCH LINE",NL
 LU3 ,EM
 TXW = 801
 PRT 0001 0050 0111 0300 84
 RTS
```

BTC

```
BTC
 PUT 0050
 RPT 30
 LU3 "GO BUSY NOW",NL
 LU3 ,EM
 TXW = 801
 MSG 'START TIMING WHEN BUSY',BEL,CR,LF
 PRT 0001 0050 0241 0300 84
 MSG 'STOP TIMING ON BELL',BEL,CR,LF
 RTS
```

CH1

```
CH1
 PUT 0050
 LU1 ,NL
 LU1 'LU1 CHAR SET',NL,NL
 HEX 40 40 F4 F5 F6 F7 F8 F9 C1 C2 C3 C4 C5 C6 15
 HEX 40 40 6D 6D 6D 6D 6D 6D 6D 6D 6D 6D 6D 6D 15
 HEX F0 4F 40 50 60 70 80 90 AC B0 CC D0 EC F0 15
 HEX F1 4F 41 51 61 71 81 91 A1 B1 C1 D1 E1 F1 15
 HEX F2 4F 42 52 62 72 82 92 A2 B2 C2 D2 E2 F2 15
 HEX F3 4F 43 53 63 73 83 93 A3 B3 C3 D3 E3 F3 15
 HEX F4 4F 44 54 64 74 84 94 A4 B4 C4 D4 E4 F4 15
 HEX F5 4F 45 55 65 75 85 95 A5 B5 C5 D5 E5 F5 15
 HEX F6 4F 46 56 66 76 86 96 A6 B6 C6 D6 E6 F6 15
 HEX F7 4F 47 57 67 77 87 97 A7 B7 C7 D7 E7 F7 15
 HEX F8 4F 48 58 68 78 88 98 A8 B8 C8 D8 E8 F8 15
 HEX F9 4F 49 59 69 79 89 99 A9 B9 C9 D9 E9 F9 15
 HEX C1 4F 4A 5A 6A 7A 8A 9A AA BA CA DA EA FA 15
 HEX C2 4F 4B 5B 6B 7B 8B 9B AB BB CB DB EB FB 15
 HEX C3 4F 4C 5C 6C 7C 8C 9C AC BC CC DC EC FC 15
 HEX C4 4F 4D 5D 6D 7D 8D 9D AD BD CD DD ED FD 15
 HEX C5 4F 4E 5E 6E 7E 8E 9E AE BE CE DE EE FE 15
 HEX C6 4F 4F 5F 6F 7F 8F 9F AF BF CF DF EF FF 15
 TXW = 801
 PRT 0006 0050 011E 0300
 RTS
```

CH3

```
CH3
 PUT 0050
 LU3 ,NL
 LU3 "LU3 CHAR SET",NL,NL
 HEX 10 10 20 21 22 23 24 25 26 27 28 29 AC A1 C3
 HEX 10 10 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 03
 HEX 20 16 10 10 20 30 40 50 60 70 80 90 A0 B0 03
 HEX 21 16 10 11 21 31 41 51 61 71 81 91 A1 B1 C3
 HEX 22 16 10 12 22 32 42 52 62 72 82 92 A2 B2 C3
 HEX 23 16 10 13 23 33 43 53 63 73 83 93 A3 B3 C3
 HEX 24 16 10 14 24 34 44 54 64 74 84 94 A4 B4 C3
 HEX 25 16 10 15 25 35 45 55 65 75 85 95 A5 B5 03
 HEX 26 16 10 16 26 36 46 56 66 76 86 96 A6 B6 03
 HEX 27 16 10 17 27 37 47 57 67 77 87 97 A7 B7 03
 HEX 28 16 08 18 28 38 48 58 68 78 88 98 A8 B8 C3
 HEX 29 16 09 19 29 39 49 59 69 79 89 99 A9 B9 03
 HEX A0 16 0A 1A 2A 3A 4A 5A 6A 7A 8A 9A AA BA C3
 HEX A1 16 0B 1B 2B 3B 4B 5B 6B 7B 8B 9B AB BB 03
 HEX A2 16 0C 1C 2C 3C 4C 5C 6C 7C 8C 9C AC BC 03
 HEX A3 16 0D 1D 2D 3D 4D 5D 6D 7D 8D 9D AD BD 03
 HEX A4 16 0E 1E 2E 3E 4E 5E 6E 7E 8E 9E AE BE 03
 HEX A5 16 0F 1F 2F 3F 4F 5F 6F 7F 8F 9F AF BF C3
 TXW = 801
 PRT 0001 0050 011E 0300 84
 RTS
```

CLB

```
CLB
 LAC 0050 LOAD ADDRESS COUNTER
 WRC LMK 00 = 801 LOAD MASK REGISTER
 WRC ZER 00 = 801 ZERO UNDER MASK
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0800 CHECK ADDR CNTR
 RTS
```

CLX

```
CLX
 LAC 0050 LOAD ADDRESS COUNTER
 WRC LMK 00 = 801 LOAD MASK REGISTER
 WRC ZER 00 = 801 ZERO UNDER MASK
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0000
 RTS
```

CS1

```
CS1
 STR
 JSR UC1
 JSR NC1
 LU1 '=,;:'
 HEX 7D 7F 4A 6A 6D C0 D0 E0 79 A1
 LU1 ,+-@()% !?*/%% ,
 JSR LC1
 TXW = 801
 RTS
```

CS3

```
CS3
 STR
 JSR UC3
 JSR NC3
 LU3 "=,;:'"
 HEX 13 1B 17 2F 0E 0F 15 37
 HEX 3D 3E 3A 3B 0A 0B
 LU3 "+-@()% !?*/%% "
 JSR LC3
 TXW = 801
 RTS
```

DAK

DBT

DBX

DEP

DC1

DTC

DWT

EL2

ENS

EX1

EX1

EX2

EX2

FIX

FIX

HP1

HP1

HP2

HP2

HPT

HPT

LC1

LC1

LC3

LC3

LLP

LLP

LL1

LL1

LL3

LL3

MCD

MOD

MPP

MPP

MSA

MSA

MSL

MSL

NC1

NC1

NC3

NC3

CRD

CRD

PCR

POR

RMD

RMD

RST

RST

SB1

SB1

SCP

SCP

SCT

SCT

SCX

SCX

SF1

SF1

SL1

SL1

SL3

SL3

TH2

TH2

TNP

TNP

TNX

TNX

TP1

TP1

TPR

TPR

TPX

TPX

UC1

UC1

UC3

UC3

VAR

VAR

WMD

WMD

WMF

WMF

WAI

WAI

WCC

WOC

WPC

WPC

WRP

WRP

WRX

WRX

WSA

WSA

ZRT

ZRT

ZRX

```
ZRX
TNP JSR RST RESET TESTS
 JSR ACT ADDRESS COUNTER TESTS
 JSR DWT DATA WORD TEST
 JSR ZRT CLEAR AND CHECK FOR ZERO TEST
 JSR SCT SEARCH CHARACTER TESTS
 JSR DBT DEVICE BUFFER TESTS
 RTS
TPR JSR CH1 LU1 CHARACTER PRINT
 JSR CH3 LU3 CHARACTER PRINT
 JSR FIX FIX PRINT
 JSR VAR VARIABLE PRINT
 JSR WRP LU1 WRAP PRINT
 JSR TP1 LU1 PRINTS
 JSR LLP LONG LINE PRINTS
 RTS
TNX JSR RST RESET TESTS
 JSR ACT ADDRESS COUNTER TESTS
 JSR DWT DATA WORD TEST
 JSR ZRX CLEAR AND CHECK FOR ZERO TEST
 JSR SCX SEARCH CHARACTER TESTS
 JSR DBX DEVICE BUFFER TESTS
 RTS
TPX JSR CH1
 JSR CH3
 JSR FIX
 JSR VAR
 JSR WRX
 JSR TP1
 JSR LLP
 RTS
 PAGE
*
* POWER ON RESET TEST
*
POR JSR WPO WAIT POWER ON OR RESET
 CMD PAL = 801 SOUND ALARM
 MSG 'IDLING',CR,LF
 TIL WAIT FOR
 CMD POL   801 *
 CMD POL = 840 STATUS AVAILABLE
 CMD PAC = 801 ACKNOWLEDGE STATUS AVAILABLE
 CMD POL = 801 CHECK NOTHIN ELSE PENDING
 CMD STA = 900 CHECK DEVICE IS DISABLED STATE
```

```
 CMD RAH = 804 CHECK ADDR CNTR HIGH = 00
 CMD RAL = 804 CHECK ADDR CNTR LOW  = 00
 CMD REA = 804 CHECK STATUS BYTE = 00
 CMD PEN = 801 ENABLE DEVICE
 CMD STA = 801 CHECK DEVICE IS ENABLED
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 RTS
*
* RESET COMMAND SEQUENCE TO DEVICE
*
RST CMD RES = 801 SEND RESET COMMAND TO DEVICE
 DAT 00 = 801 SEND RESET DATA
WPO TIL DEVICE MAY STOP TRANSMITTING
 CMD POL = 815 ONLY VALID RESPONSE AFTER RES CMD
 CMD PAC = 801 ACKNOWLEDGE POWER ON
 CMD PCL = 801 MUST BE CLEAN RESPONSE
 CMD STA = 900 PRINTER IS DISABLED ON POR
 CMD RAH = 804 ADDRESS COUNTER HIGH MUST BE ZERO
 CMD RAL = 804 ADDRESS COUNTER LOW MUST BE ZERO
 CMD TID = 808 MUST READ IDENTIFICATION AND
 PUT 004A WRITE TO CONTROL UNIT AREA
 HEX AA 32 74 AA THIS STRING
 TXW = 801
 CAC 004E CHECK ADDRESS COUNTER AFTER WRITE
 CMD PEN = 801 BEFORE SENDING FIRST ENABLE POLL
 CMD STA = 801 STATUS MUST BE ENABLED IMMEDIATELY
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 RTS
 PAGE
 PAGE
*
* DATA WORD TEST
*
DWT PUT 0100 WRITE DATA WORD PATTERN
 FOR 00 FF 01
 HEX
 TXW = 801
 CAC 020C CHECK ADDRESS COUNTER
 LAC 0100 RELOAD ADDRESS COUNTER
 FOR 00 FF 01
 CPD COMPARE DATA WORD
 CAC 020C CHECK ADDRESS COUNTER
 RTS
 PAGE
ZRT JSR CLB ZERO 2K DEVICE BUFFER
 LAC 0050
 CMD REA = 804 CHECK FOR ZERO DATA
 RPT 07AF 1967
 RTX
 CAC 0800
 RTS
ZRX JSR CLX ZERO 4K DEVICE BUFFER
 LAC 0050
```

```
 CMD REA = 804 CHECK FOR ZERO DATA
 RPT OFAF 4015
 RTX
 CAC 0000
 RTS
 PAGE
*
* 2K SEARCH CHARACTER TEST
*
SCT JSR CLB ZERO 2K DEVICE BUFFER
 JSR SCP WRITE SEARCH CHARACTER PATTERN
 RPT 07
 RTX
 CAC 07A0
 JSR SF1 SEARCH FORWARD
 CAC 0800
 LAC 07FF
 JSR SB1 SEARCH BACKWARD
 RTS
*
* 4K SEARCH CHARACTER TEST
*
SCX JSR CLX ZERO 4K DEVICE BUFFER
 JSR SCP WRITE SEARCH CHARACTER PATTERN
 RPT 07
 RTX
 CAC 07A0
 JSR SF1 SEARCH FORWARD
 CAC 0000
 LAC 0FFF
 JSR SB1 SEARCH BACKWARD
 RTS
*
* SEARCH CHARACTER PATTERN
*
SCP PUT 0050
 RPT 27
 HEX 00
 RPT 27
 HEX 55
 RPT 27
 HEX 5A
 RPT 27
 HEX A5
 RPT 27
 HEX AA
 RPT 27
 HEX FF
 TXW = 801
 RTS
 PAGE
*
* SEARCH FORWARD TEST
```

```
*
SF1 LAC 0050
 WRC LMK AA = 801
 WRC ZER FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 00EC
 LAC 0050
 RPT 9C
 CMD REA = 804
 WRC SFD F0 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 01AF
 WRC SFD 0F = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0272
 WRC SFD 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 030E
 WRC SFD FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 03AA
 WRC LMK 55 = 801
 WRC SFD 0F = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 046D
 WRC SFD F0 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0530
 WRC SFD FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 WRC SFD 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 WRC SFD FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 WRC SFD 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0668
 WRC LMK 00 = 801
 WRC ZER 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 RTS
 PAGE
*
* SEARCH BACKWARD TEST
*
SB1 WRC LMK FF = 801
 WRC SBK FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 05CB
 WRC SBK 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0508
 WRC SBK 55 = 801
```

```
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0445
 WRC SBK AA = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 03D0
 WRC LMK 0F = 801
 WRC SBK 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0334
 WRC SBK 55 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 028F
 WRC SBK FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0223
 WRC LMK F0 = 801
 WRC SBK 55 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 01AE
 WRC SBK AA = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0112
 WRC SBK 00 = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 00EB
 WRC LMK 00 = 801
 WRC SBK FF = 801
 JSR WAI WAIT FOR OPERATION COMPLETE
 CAC 0FFF
 RTS
 PAGE
*
* 2K DEVICE BUFFER MEMORY TESTS
*
DBT JSR WMF WRITE MULTIPLE FF'S
 RPT 7A
 RTX
 CAC 0800
 LAC 0050
 CMD REA = FFC
 RPT 07AF
 RTX
 CAC 0800
 JSR WMD WRITE MULTIPLE DATA (00-0F)
 RPT 7A
 RTX
 CAC 0800
 LAC 0050
 RPT 7B
 JSR RMD READ MULTIPLE DATA (00-0F)
 CAC 0800
 RTS
*
```

```
* 4K DEVICE BUFFER MEMORY TESTS
*
CBX JSR WMF WRITE MULTIPLE FF'S
 RPT FA
 RTX
 CAC 0000
 LAC 0050
 CMD REA = FFC
 RPT 0FAF
 RTX
 CAC 0000
 JSR WMD WRITE MULTIPLE DATA (00-0F)
 RPT FA
 RTX
 CAC 0000
 LAC 0050
 RPT FB
 JSR RMD READ MULTIPLE DATA (C0-0F)
 CAC 0000
 RTS
 PAGE
*
* WRITE MULTIPLE DATA (16 FF)
*
WMF PUT 0050
 HEX FF FF FF FF FF FF FF FF
 HEX FF FF FF FF FF FF FF FF
 TXW = 8C1
 RTS
*
* WRITE MULTIPLE DATA (00-0F)
*
WMD PUT 0050
 HEX 00 01 02 03 04 05 06 C7
 HEX 08 09 0A 0B 0C 0D 0E CF
 TXW = 801
 RTS
*
* READ MULTIPLE DATA (00-0F)
*
RMD CMD REA = 804
 CMD REA = 808
 CMD REA = 810
 CMD REA = 81C
 CMD REA = 820
 CMD REA = 82C
 CMD REA = 834
 CMD REA = 838
 CMD REA = 840
 CMD REA = 84C
 CMD REA = 854
 CMD REA = 858
 CMD REA = 864
```

```
 CMD REA = 868
 CMD REA = 870
 CMD REA = 87C
 RTS
 PAGE
*
* LU3 FIXED PRINT TEST
*
FIX PUT
 LU3 "NON DISPLAY IS"
 HEX FC
 LU3 "NOT"
 HEX F0
 LU3 "WORKING OK"
 RPT 33
 HEX 00
 LU3 "INVALID ESCAPE SEQUENCE PRINTS AS:% "
 RPT 2B
 HEX 00
 LU3 "A LINE OF NULLS HERE:"
 RPT 8B
 HEX 00
 LU3 "DOES NOTHING"
 RPT 44
 HEX 00
 LU3 "AN %0E%0F%08EXPANDED FIXED PRINT"
 RPT 34
 HEX 00
 TXW = 801
 PRT 0001 0050 01E7 0301 50
*
* CANDY STRIPE PRINT
*
 LAC 0050
 JSR CS3 CHARACTER SET LU3
 RPT 15
 RTX
 PRT 0001 0050 0660 0301 50
*
* TEST OF AUTO FORM FEED AFTER OPERATOR INITIATED LOCAL PRINT
*
 PUT
 LU3 "%L0AAUTO FF AFTER OPERATOR INITIATED "
 LU3 "LOCAL PRINT FORM FEED FROM HERE"
 RPT 0A
 HEX 00
 LU3 "TO HERE SHOULD BE 10 LINES "
 LU3 "(IF AUTO FF SELECTED)"
 RPT 1F
 HEX 00
 TXW = 801
 PRT 0011 0050 0050 0301 50
 PRT 0001 00A0 0050 0301 50
```

```
 RTS
 PAGE
*
* LU3 VARIABLE PRINT TEST
*
VAR PUT
 LU3 "A LINE OF NULLS HERE:",NL
 RPT 84
 HEX 00
 LU3 "LEAVES A BLANK LINE, _A LINE OF SPACES HERE:",NL
 RPT 84
 HEX 10
 LU3 "LEAVES A BLANK LINE",NL
 LU3 "2 ESCAPES HERE:%% SHOULD PRINT AS 1 ESCAPE",NL
 LU3 "2 ESCAPES HERE:%% SHOULD PRINT AS 1 ESCAPE",NL
 LU3 "DOWNLOAD FORM OF 4 LINES NOW%LC4",NL
 LU3 "FROM HERE",NL,FF
 LU3 "TO HERE IS 4 LINES (LU3 FORM FEED)",NL
 LU3 "THIS LINE ENDS WITH NL CR EM",NL,CR,EM
 TXW = 801
 PRT 0001 0050 0780 0300 84
 PUT
 LU3 "THERE SHOULD BE NO BLANK LINE JUST ABOVE THIS,"
 LU3 " AND NO FORM FEED HERE",FF,EM
 TXW = 801
 PRT 0001 0050 0780 0300 84
*
* ABORT TEST
*
 PUT
 LU3 "ABORT HERE",NL
 RPT 35
 HEX 32
 LU3 "FAILED",NL,EM
 LU3 "FOLLOWED BY A VAR PRINT",NL,EM
 TXW = 801
 PRT 0001 0050 000B 03C0 84
 PRI 0001 005B 003D 0300 84
 JSR ABO ABORT PRINT
 PRT 0001 0098 001A 0300 84
 RTS
 PAGE
*
* TEST PRINT LU1
*
TP1 PUT
 LU1 'UNDERSCORE USING BACKSPACES'
 RPT 20
 LU1 ,BS
 RPT 1B
 LU1 ''
 LU1 ,NL
 LU1 'UNDERSCORE USING CR',CR
```

```
 RPT 13
 LU1 ''
 LU1 ,NL
 HEX 2B C1 06 84 01 84 14 28
 LU1 'COL 1 TAB',HT
 LU1 'COL 20 TAB',HT
 LU1 'COL 40 TAB',NL
 LU1 'DOWNLOAD FORM LENGTH OF 3 NOW',NL
 LU1 '%LO3FROM HERE',FF
 LU1 'TO HERE IS 4 LINES (MUST BE INITIAL TEST)',NL
 HEX 2B C2 C6 0A 01 0A 03 C6 0C
 LU1 'LIN 1 TAB(10 BLANK LINES)',VT
 LU1 'LIN 3 TAB(1 BLANK LINE)',VT
 LU1 'LIN 6 TAB(2 BLANK LINES)',NL,FF
 LU1 'LIN 11 (LIN1)(4 BLANK LINES)',NL
 LU1 'NOW DOING FORM DOWNLOAD AFTER SVF',NL
 LU1 '%LO5FROM HERE',FF
 LU1 'TO HERE IS 9 LINES',NL
 LU1 'INP',ENP
 LU1 ' _ENP HERE:',INP
 LU1 'DO NOTHING',NL
 LU1 'LF _VCS HERE',LF
 HEX 04 81
 LU1 'LEAVE 1 BLANK LINE _CC NC CR',NL
 LU1 'IRS HERE',IRS
 LU1 'DOES A NL',NL
 LU1 'TRN HERE '
 HEX 35 02 0C 0D
 LU1 ' IGNORED AN FF CR',NL
 TXW = 801
 PRT 0006 0050 0230 0300
*
* SET LINE DENSITY PRINT TEST
*
 PUT
 LU1 'SLD HERE',NL
 HEX 2B C6 02 18
 LU1 '3',NL
 LU1 'LINES',NL
 LU1 'PER',NL
 LU1 'INCH',NL
 HEX 2B C6 02 12
 LU1 '4',NL
 LU1 'LINES',NL
 LU1 'PER',NL
 LU1 'INCH',NL
 HEX 2B C6 02 0C
 LU1 '6',NL
 LU1 'LINES',NL
 LU1 'PER',NL
 LU1 'INCH',NL
 HEX 2B C6 02 09
 LU1 '8',NL
```

```
 LU1 'LINES',NL
 LU1 'PER',NL
 LU1 'INCH',NL
 HEX 2B C6 02 0C
 TXW = 801
 PRT 0006 0050 001E 0300
 PRT 0006 006E 0015 0300
 PRT 0006 0083 0015 0300
 PRT 0006 0098 0019 0300
*
* LU1 CANDY STRIPE PRINT
*
 PUT 0050
 HEX 2B C1 02 50
 TXW = 801
 JSR CS1 CHARACTER SET LU1
 RPT 14
 RTX
 LU1 ,NL
 TXW = 801
 PRT 0006 0050 0615 0320
 RTS
 PAGE
*
* WRAP TEST FOR LU1 PRINT
*
WRP PUT 07C0
 LU1 'THIS TEST WRAPS AN ESCAPE SEQUENCE '
 LU1 'FOR COMPRESSED PRINT HERE:',NL
 LU1 '%'
 TXW = 801
 PUT
 LU1 '12COMPRESSED',NL
 LU1 'THEN SPLITS AN ESCAPE FOR EXPANDED '
 LU1 'ACROSS MESSAGE BOUNDRY HERE:',NL
 LU1 '%'
 TXW = 801
 PRT 0006 07C0 008F 0300
 PUT
 LU1 '0E%0F%08EXPANDED LU1 PRINT',NL
 TXW = 801
 PRT 0006 0050 001D 0300
 PUT 07EF
 LU1 'NON CONTIGUOUS WR'
 TXW = 801
 PRT 0006 07EF 0011 0300
 PUT 0100
 LU1 'AP TEST HERE',NL
 TXW = 801
 PRT 0006 0100 000D 0300
 RTS
*
* WRAP TEST FOR LU1  -PRINT EXTENDED BUFFER
```

```
*
WRX PUT OFCO
 LU1 'THIS TEST WRAPS AN ESCAPE SEQUENCE '
 LU1 'FOR COMPRESSED PRINT HERE:',NL
 LU1 '%'
 TXW = 801
 PUT
 LU1 '12COMPRESSED',NL
 LU1 'THEN SPLITS AN ESCAPE FOR EXPANDED '
 LU1 'ACROSS MESSAGE BOUNDRY HERE:',NL
 LU1 '%'
 TXW = 801
 PRT 0006 OFCO 008F 0300
 PUT
 LU1 '0E%0F%08EXPANDED LU1 PRINT',NL
 TXW = 801
 PRT 0006 0050 001D 0300
 PUT OFEF
 LU1 'NON CONTIGUOUS WR'
 TXW = 801
 PRT 0006 OFEF 0011 0300
 PUT 0100
 LU1 'AP TEST HERE',NL
 TXW = 801
 PRT 0006 0100 000D 0300
 RTS
 PAGE
*
* LONG LINE PRINT
*
LLP PUT
 LU3 ,EM START AT COLUMN 0
 RPT 42
 LU3 "FFFF"
 RPT 42
 LU3 "VVVV"
 LU3 ,EM
 HEX 2B C1 02 84
 RPT 42
 LU1 '1111'
 LU1 ,NL
 TXW = 801
 PRT 0001 0050 0001 0300 84
 PRT 0001 0051 0108 0301 84
 PRT 0001 0159 0109 0300 84
 PRT 0006 0262 010D 0300
 RTS
 PAGE
*
* DESELECT TIME OUT TEST
*
DTO PUT
 RPT 30
```

```
 LU3 "GO DESELECT NOW",NL
 LU3 ,EM
 TXW = 801
 MSG 'START TIMING WHEN DESELECTED',BEL,CR,LF
 PRT 0001 0050 0301 0300 84
 MSG 'STOP TIMING ON BELL',BEL,CR,LF
 RTS
 PAGE
 PAGE
 PAGE
 PAGE
*
* HOPPER PRINT TEST
*
HPT JSR HP1 SELECT HOPPER  1
 LAC 0050
 JSR SL3
 RPT 21
 RTX
 PRT 0001 0050 06E0 0300 50
 PRT 0001 0050 06E0 0300 50
 PRT 0001 0050 06E0 0300 50
 JSR HP2 SELECT HOPPER  2
 LAC 0050
 JSR SL3
 RPT 21
 RTX
 PRT 0001 0050 06E0 0300 50
 PRT 0001 0050 06E0 0300 50
 PRT 0001 0050 06E0 0300 50
 RTS
 SUBTTL COMMON USER MACRO FOR ISI3274A CONTROL UNIT
 PAGE
*
* LU1 CHARACTER SET
*
*
* UPPER CASE LETTERS LU1
*
UC1 LU1 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
 RTS
*
* LOWER CASE LETTERS LU1
*
LC1 LU1 'abcdefghijklmnopqrstuvwxyz'
 RTS
*
* NUMBER CHARACTERS LU1
*
NC1 LU1 '1234567890'
 RTS
*
* SHORT LINE LU1
```

```
*
SL1 STR
 RPT 8
 JSR NC1
 TXW = 801
 RTS
*
* LONG LINE LU1
*
LL1 STR
 RPT 13
 JSR NC1
 LU1 '12'
 TXW = 801
 RTS
 PAGE
*
* LU3 CHARACTER SET
*
*
* UPPER CASE LETTERS LU3
*
UC3 LU3 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
 RTS
*
* LOWER CASE LETTERS LU3
*
LC3 LU3 "abcdefghijklmnopqrstuvwxyz"
 RTS
*
* NUMBER CHARACTERS LU3
*
NC3 LU3 '1234567890'
 RTS
*
* SHORT LINE LU3
*
SL3 STR
 RPT 8
 JSR NC3
 TXW = 801
 RTS
*
* LONG LINE LU3
*
LL3 STR
 RPT 13
 JSR NC3
 LU3 '12'
 TXW = 801
 RTS
 PAGE
*
```

```
* WAIT FOR OPERATION COMPLETE
*
WAI TIL WAIT FOR
 CMD POL   801 *
 RXW = 808 OPERATION COMPLETE
 CMD PAC = 801 ACKNOWLEDGE OPER COMPLETE
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 CMD STA = 900 CHECK DEVICE IS DISABLED
 RTS
 PAGE
*
* WAIT FOR STATUS AVAILABLE
*
WSA TIL WAIT FOR
 CMD POL   801 *
 RXW = 840 STATUS AVAILABLE
 CMD PAC = 801 ACKNOWLEDGE STATUS AVAILABLE
 CMD POL = 801 CHECK NOTHING ELSE PENDING
 CMD STA = 900 CHECK DEVICE IS DISABLE
 CMD RAH = 804 CHECK ADDR CNTR HIGH = 00
 CMD RAL = 804 CHECK ADDR CNTR LOW  = 00
 RTS
 PAGE
*
* WAIT FOR ORDER COMPLETE
*
WOC JSR WSA WAIT FOR STATUS AVAILABLE
 CMD REA = 900 CHECK ORDER COMPLETE BIT IS SET
 JSR ACK ACK ORDER COMPLETE AND ENABLE DEVICE
 RTS
*
* ENABLE AND CHECK STATE
*
ENS CMD PAC = 801 CLEAR POLL REGISTER
 CMD PEN = 801 ENABLE DEVICE
 CMD STA = 801 CHECK DEVICE IS ENABLED
 RTS
*
* DISABLE AND ACKNOWLEDGE
*
DAK CMD PAC = 801 CLEAR POLL REGISTER
 CMD PDI = 801 DISABLE DEVICE
 TIL WAIT FOR
 CMD STA   801 DEVICE NOT ENABLED
 CMD PAC = 801 CLEAR POLL REGISTER
 CMD POL = 801 CHECK POLL REGISTER CLEARED
 CMD STA = 900 CHECK DEVICE IS DISABLED
 RTS
 PAGE
*
* DISABLE AND ENABLE DEVICE DURING PRINT
*
DEP WHI EX1,DO1 WHILE (POL=801) DO (DIS-ENABLE)
```

```
 RXW = 840 MUST BE STATUS AVAILABLE
 RTS
EX1 CMD POL = 801 CLEAN RESPONSE ?
 RTS
*
* IF DEVICE IS ENABLED THEN DISABLE AND ENABLE DEVICE
*
D01 IFT EX2,TH2,EL2 IF (PDI=801) THEN (RE-ENABLE)
 RTS
EX2 CMD PDI = 801 DISABLE DEVICE ?
 RTS
TH2 TIL WAIT FOR
 CMD POL   801 DEVICE DISABLED
 IFT EX3,TH3,EL3 IF (RXW=808) THEN 'TH3'
 RTS
EL2 RXW = 840 STATUS AVAILABLE ?
 RTS
EX3 RXW = 808 OPERATION COMPLETE
 RTS
TH3 CMD PAC = 801 CLEAR POLL REGISTER
 CMD PEN = 801 ENABLE DEVICE
 DLY 000A DELAY 10 MS
 RTS
EL3 RXW = 840 STATUS AVAILABLE ?
 RTS
 PAGE
*
* CONTROL UNIT AREA USER MACROS
*
*
* SET THE PRINTER MODE
*
MOD LAC 0010
 CMD WRI = 801
 RTS
*
* SET THE PRINTER MESSAGE STARTING ADDRESS
*
MSA LAC 0012
 CMD WRI = 801
 RTS
*
* SET THE PRINTER MESSAGE LENGTH
*
MSL LAC 0014
 CMD WRI = 801
 RTS
*
* SET THE PRINTER ORDER
*
ORD LAC 0016
 CMD WRI = 801
 RTS
```

```
*
* SET THE PRINTER MAX PRESENTATION POSITION
*
MPP LAC 0018
 CMD WRI = 801
 RTS
 PAGE
*
* SELECT HOPPER  1
*
HP1 PUT 07F0
 LU3 '%H1',NL,FF
 TXW = 801
 PRT 0001 07F0 0006 0300 50
 RTS
*
* SELECT HOPPER  2
*
HP2 PUT 07F0
 LU3 '%H2',NL,FF
 TXW = 801
 PRT 0001 07F0 0006 0300 50
 RTS
```

APPENDIX A

Test Numbers

The following list of "CANNED" tests that can be run by number from the front panel or from the terminal in 'TEST' mode, or by name from the terminal in the 'RUN' mode. The list includes a brief description of the test.

Number	Mnemonic	
00	TNX	Tests with No Print for 4k buffer
01	RST	ReSet Test
02	ACT	Address Counter Tests
03	DWT	Data Word Test
04	ZRX	ZeRo Test for 4k buffer
05	SCX	Search Character Tests for 4k buffer
06	DBX	Device Buffer Tests for 4k buffer
07	ZRT	ZeRo Test for 2k buffer
08	SCT	Search Character Tests for 2k buffer
09	DBT	Device Buffer Tests for 2k buffer
10	TPX	Tests with Prints for 4k buffer
11	CH1	LU1 Character print
12	CH3	LU3 Character print
13	FIX	FIX print tests
14	VAR	VARiable print tests
15	WRX	WRap LU1 prints tests for 4k buffer
16	TP1	LU1 Test Prints
17	LLP	Long Line Prints
18	WRP	WRap LU1 Prints test for 2k buffer
19	TPR	Test PRints for 2k buffer
20	ALX	ALL tests for 4k buffer
21	ALL	ALL tests for 2k buffer
22	BTO	Busy TimeOut test
23	DTO	Device TimeOut test
24	POR	Power On Reset test

APPENDIX B

ASCII Control Codes

A list of ASCII control codes with a brief cescription.

Special ASCII codes that control the tester ~~operation~~ behavior. The following control codes cause some action to be taken and the operator need not press the RETURN key before the action occurs.

KEY	ASCII	HEX	
RETURN	CR	$0D	Process the enterec command line.
CTRL P	DLE	$10	Toggle the paging flag.
CTRL Q	DC1	$11	Enable the coax transmission flag.
CTRL R	DC2	$12	Toggle the register flag.
CTRL S	DC3	$13	Disable the coax trarsmission flag.
CTRL T	DC4	$14	Toggle the trarsmission flag.
CTRL U	NAK	$15	Delete the entire command line.
CTRL V	SYN	$16	Display the PASS-FAIL ccunters.
CTRL W	ETB	$17	Toggle the warning bell flag.
CTRL [	ESC	$1B	Return to command lire entry.
DELETE	DEL	$7F	Delete the last character entered.

ASCII codes to LU1 and LU3 codes.

KEY	ASCII	HEX	LU1	LU3	
CTRL A	SOH	$01	$19	$01	END OF MESSAGE
CTRL B	STX	$02	$0C	$02	FORM FEED
CTRL C	ETX	$03	$15	$03	NEW LINE
CTRL D	EOT	$04	$25	$03	LINE FEED
CTRL E	ENQ	$05	$0D	$05	CARRIAGE RETURN
CTRL G	BEL	$07	$2F	-	BELL
CTRL H	BS	$08	$16	-	BACK SPACE
CTRL I	HT	$09	$05	-	HORIZONTAL TAB
CTRL J	LF	$0A	$25	-	LINE FEED
CTRL K	VT	$0B	$0B	-	VERTICAL TAB
CTRL L	FF	$0C	$0C	$02	FORM FEED
CTRL Y	EM	$19	$19	$01	END OF MESSAGE
CTRL ^	IRS	$1E	$1E	-	RECORD SEPERATOR

NOTE : All remaining control codes are converted tc nulls.

APPENDIX C

Tester Messages

NUMBER	MESSAGE
000	'Passed !'
001	'Failed ???'
002	'Exited !'
003	'*** Aborted ***'
004	'Stack underflow'
005	'Stack overflow'
006	'Transmit buffer overflow'
007	'Counter overflow'
100	'Invalid "Test" number'
101	'Valid "Build" macro names'
102	'Invalid "Dump" address'
103	'Invalid "Help" name'
104	'Valid "List" table names'
105	'Invalid "Run" macro'
106	'Invalid "Move" addresses'
110	'Macro name not found'
111	'Command name not found'
112	'Invalid data word'
113	'Invalid hex character(s)'
114	'String overflow'
115	'Illegal delimiter'
116	'Missing receive word'
117	'Invalid receive word'
118	'Too many hex digits'
119	'Invalid hex byte'
120	'Invalid device address'
121	'Invalid hex word'
122	'Too many hex digits'
123	'Not a hex character'
124	'User macro name not found'
125	'Excute macro name not found'
126	'Invalid BCD number'
127	'Test number not found'
128	'Macro buffer overflow'
129	'Not implemented'
130	'Excute macro address not found'
131	'Command word not found'
132	'User macro address not found'

```
133     'Not a decimal number'
249     'Spurious or no interupt service for '
25C     'SW3'
251     'SW2'
252     'FIRQ'
253     'IRQ'
254     'SWI'
255     'NMI'
256     'Message number - '
```

APPENDIX C

Mnemonic Names

A list of command words by mnemonic name with a brief description.

INB inserts a byte in device buffer.
INV invald command word.
LAH load device's address counter high.
LAL load device's address counter low.
LCR load device's control register.
LME load device's mask register for EAB buffer.
LMK load device's mask register.
PAC acknowledge device's poll response.
PAL scund device's alarm.
PDI disable device.
PEN enable device.
POL read device's poll response register.
RAH read device's address counter high.
RAL read device's address counter low.
RDE read data from device's EAB buffer.
REA read data from device's buffer.
RES reset device.
STA read device's state register.
SBK search backward for pattern.
SFD search forward for pattern.
SOP start operation.
TID read device's terminal identification.
WAE write data to device's EAB buffer.
WME load device's EAB mask register.
WRI write data to device's buffer.
ZER zero device's buffer.