

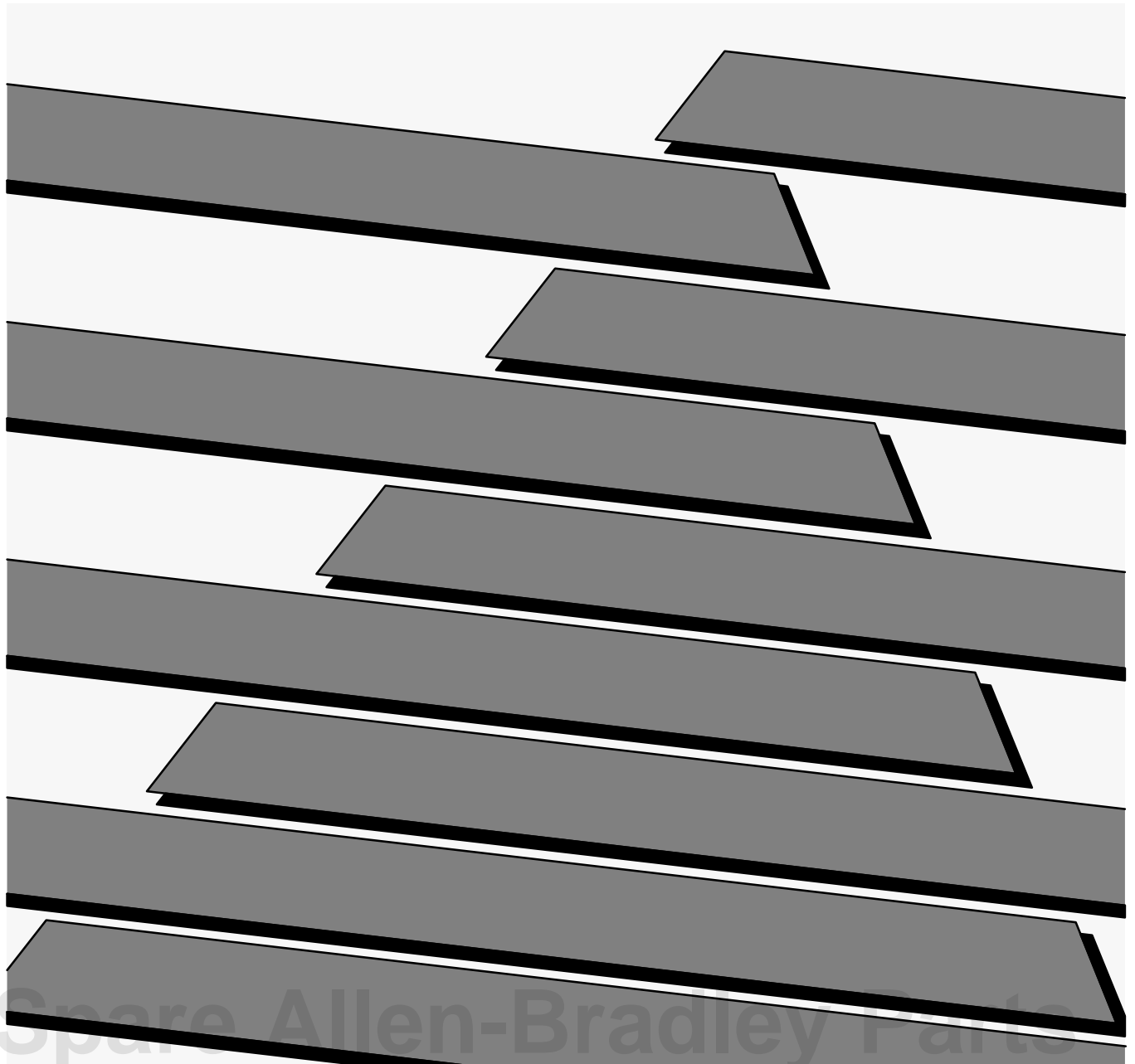


ALLEN-BRADLEY

# Bulletin 2708 Application Generator Software

(Catalog No. 2708-NAG)

User Manual



## Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. “Application Considerations for Solid State Controls” (Publication SGI-1.1) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will the Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, the Allen-Bradley Company cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of the Allen-Bradley Company is prohibited.

© 1991 Allen-Bradley Company, Inc.

PLC is a registered trademark of Allen-Bradley Company, Inc.  
Pyramid Integrator, Data Table Library and CVIM are trademarks of Allen-Bradley Company, Inc.  
MicroVAX, VAX, VAXstation, VAXcluster, VMS, DECnet, DECwindows, VAXserver, VAX FMS, VAX DATATRIEVE, VAX GKS, VAX DEC/CMS, VAX VTX, and VAX Printserver are trademarks of DIGITAL Equipment Corporation.

<b>A-B BASIC Applications Generator .....</b>	<b>3</b>
Overview .....	3
What It is .....	3
Who Can Use It .....	3
Installation and Requirements .....	4
Normal Operation Tutorial .....	5
Key Stroke Quick Reference .....	5
Normal Operation Tutorial (cont'd) .....	6
Special Purpose Keys .....	6
Variables .....	6
The AG Screen .....	7
Main Operations Menu .....	7
F1 – General Help .....	7
F4 – Read From Laser Or Device .....	12
F5 – Send To Host Or Device .....	12
F6 – Move, Transfer, Add, Etc. ....	13
Adding A-B BASIC Functions .....	19
F8 – IF (Conditional) .....	19
F9 – GoTo Or Loop To Position .....	20
F10 – File (New, Load, Save, Generate) .....	20

## A-B BASIC Applications Generator

### Overview

The Application Generator (AG) is an interactive program development tool for Catalog No. 2708–DH5XX products.

An AG developer trains a DH5 workstation by interactively modifying its current behavior until the desired functions are achieved.

Because the developer is working with the actual operation of the program, instead of trying to relate the program's behavior to a listing of program source code, development is both faster and more intuitive than using traditional programming techniques.

### What It is

AG is a program which simulates a workstation.

When the program simulates the application properly, an A-B BASIC program is generated by AG. This program will perform identically to the A-B BASIC computer simulator.

### Who Can Use It

An AG developer should be a person with a good grasp of the problem to be solved, and be familiar with use of PC based products. Programming skills may be beneficial, although they are not required.

While AG substantially decreases development time, and improves overall program quality, it is not a cure all. A well designed application demands in depth problem analysis and an attention to detail that no tool can replace.

## Overview (cont'd)

### How It Works

The simulation is controlled by the AG master program. The AG developer loads an initial personality (.AIP) file into the simulator program.

The personality will contain a set of operations that are eventually compiled into an A-B BASIC program. The operations are selectable from menus within AG.

Unlike a program, a personality can be executed stepwise, both forwards and in reverse. This means that any operation the simulator program performs can be undone. This provides true 'hands on' access to programming instead of the more common indirect approach.

The AG developer steps forwards and backwards through the operation of the current personality. Where operations are missing, the developer inserts them. If the personality performs extraneous operations then the developer deletes them.

## Installation and Requirements

AG requires an IBM PC or IBM PC compatible which has at least 512K of RAM, a hard disk, and DOS 3.2 or later.

Before running install, you should be pathed to the floppy drive which has the INSTALL.BAT file on it. You should be at the A: or B: prompt.

To install on drive C, type :     **INSTALL C:**

To install on drive D, type :     **INSTALL D:**

You should also have a complete and functional copy of MicroSoft QUICKBASIC V4.5 or later on your system. The QB command must be accessible in the normal manner.

## Normal Operation Tutorial

### Starting AG

At the C or D prompt, depending on the location of the AG directory, path to AG.

```
C:\ > CD \ AG
```

Then type AG to start the operations.

```
C:\ AG > AG
```

On the last line of your CRT there will be a message:

```
Personality name to load: Default
```

You can execute the default program by pressing ENTER. The default personality is a program which

Displays “Use F1 to F4”

Waits for user to enter a badge or press a function key.

The program then branches to various editing and entry routines to illustrate many of the features of AG.

Sends the badge data to the host.

Loops back to the start.

### Key Stroke Quick Reference

Commonly used keys for controlling the Applications Generator.

Up Arrow	: Undo the previous line (if stepping) or up one line
Down Arrow	: Execute the next line (if stepping) or down one line
INS	: Switches into single step mode
ESC	: Stop running or abandon the current operation
DEL	: Delete an operation (press it twice)
ENTER	: Edit the current operation
Alt R	: RUN the program starting at line 1 (ESC to stop)
Alt G	: Go (RUN) starting at this line
Alt U	: Undo back to this line
Alt E	: Edit the current program in full screen mode
Alt V	: List the variables in the program

## Normal Operation Tutorial (cont'd)

### Simulated Device Input Keys

Alt L	: Simulate input from the bar code port (key entry)
Alt H	: Simulate input from the host computer (key entry)
Alt C	: Simulate input from the COM1 RS232 line (key entry)
Alt A	: Simulate input from the COM2 RS232 line (key entry)
Alt F	: Select a DOS file to use for simulated inputs. Specify the input device by entering the first letter of: Laser, Host, Com, or Aux.
Alt N	: Simulate the next line of input from the response file (file entry)

### Special Purpose Keys

Page Up	: Skip up 6 lines or change field size
Page Down	: Skip down 6 lines or change field size
END	: Move to the end of the program
HOME	: Undo or Move to the start of the program
Alt I	: Set or clear the interrupt (breakpoint)
Alt D	: Allows a device log file to be specified that will store I/O transactions
-	: Moves the current line to the edit buffer
+	: Copies the current line to the edit buffer
*	: Inserts the edit buffer above the current position

### Variables

Variables are named storage locations. They can each hold up to 80 characters of information. When you start AG, you are given the variables "IN", "OUT", "SRC", "NET", "COM", and "AUX". Any time you are asked for a variable name, you can key in one of the built in names, or make up one of your own. Use Alt V to see all of your variables.

By using variables, you can move data from place to place such as from the badge reader to the display.

The names of the default variables give you a hint as to how to use them. The name of a variable does not in any way affect what you can use it for. A name is just a way of referring to the same group of characters.

Variable names must not start with a quote mark (") or a digit. It is good practice to use just one word. Upper and lower case does not matter.

In many cases where you are asked for a variable name, you can key in a number or a string. (A string is characters enclosed in quotes.) These are constants.

## The AG Screen

The AG screen will be divided into various functional areas similar to:

SIMULATED DH5 DISPLAY	PROGRAM STATUS
SIMULATED DH5 INPUT AND OUTPUT DEVICES (access using Alt L, Alt H, Alt C or Alt A)	OPERATIONS MENU (use function key or the highlighted letter to select)
Contents of variables your program is using	
The program operations you have entered (use up and down)	
STATUS AND HELP LINES	

## Main Operations Menu

- F1: Help
- F2: Display to screen
- F3: Keyboard read/laser
- F4: Read from laser or device
- F5: Send to host or device
- F6: Move, transfer, add, etc.
- F7: Other operations
- F8: IF (conditional)
- F9: Goto or loop to position
- F10: File (new, load, save, generate)

### F1 – General Help

The last two lines of the display contain 'field specific' help information at all times.

Selecting F1 brings up the context sensitive help system.



## Main Operations Menu (cont'd)

### F2 – Display To Screen

The Display Menu (F2) is used to display text, variables, date or time on the workstation. It is also used to CLEAR the display.

The size and format of each field is controlled by pressing the plus key <+> and the minus key <->. The arrow keys are used to position the field on the screen.

Note that if you select a variable to display, make sure to use the plus key <+> to make the field as long as the data you want to display. If the field is too small then characters will be lopped off.

When F2 is pressed the main menu is replaced with these options:

- f1 : Help
- f2 : Display text to screen
- f3 : Display a variable
- f4 : Display time
- f5 : Display date
- f6 : Clear the screen
- f7 : Clear and show text
- f8 : Clear line 1 with text
- f9 : Clear line 2 with text
- f10 : Other display operation

In options **f2**, **f3** and **f7–f9**, you are prompted to enter the text or the variable name. Then it is shown on the workstation display in the upper right corner of the CRT. You then use the arrow keys to position it, and the plus and minus keys to chose the size of the field. When both are to your satisfaction, press enter to write the statement in the program.

In options **f4** and **f5**, the time or date is displayed on the screen representing the workstation display. You use the arrow keys to move the time/date to the desired position, then press enter to write the statement.

Option **f6** clears the workstation's display.

Option **f10** offers various clear functions.

- f1 : Help
- f2 : Clear and show variable
- f3 : Clear line 1 with var
- f4 : Clear line 2 with var
- f5 : Clear line 1 with date
- f6 : Clear line 2 with date
- f7 : Clear line 1 with time
- f8 : Clear line 2 with time

## Main Operations Menu (cont'd)

### F3 – Keyboard Read/Laser

The Keyboard Read menu is used to read fields and characters from the keyboard. When you chose one of these attributes a message is displayed on the next to last line of the CRT. Then the attribute is added to the read statement in the program. When f3 is pressed the main menu is replaced with these options:

- f1 : Help
- f2 : Read a single keystroke
- f3 : Read a number
- f4 : Read an alpha field
- f5 : Special options
- f6 : Accept laser input also
- f7 : Set an input range/lookup
- f8 : Set a timeout

**Note:** Since there are many options that can be selected for reading, you will have to press ENTER until the main menu is shown in order to store a READ operation. Keep an eye on the help line at the bottom of your computer screen.

If you choose **f2 – READ A SINGLE KEYSTROKE** then the key number will be placed into the input variable. An example use of this choice would be in a Time & Attendance program when you need to know if the employee is checking IN or OUT; the entry could be the single keystroke of the <IN> or <OUT> key. Some of the more important numbers are :

F1=225	F2=226	F3=227	F4=228	F5 =229
F6=230	F7=231	F8=232	F9=233	F10=234
ENTER=13	IN=43	OUT=45	CLEAR=127	SPACE=32
EXIT=159	UP=24	LEFT=8	DOWN=25	RIGHT=26

The **f3 – READ NUMBER** will create a numeric field. The input will be converted into a number.



#### CAUTION:

The limits of this numeric field are +2, 147, 483, 647 maximum to - 2, 147, 483, 647 minimum. If a value is entered that is outside of this range the number will overflow to an incorrect value. If very large positive or negative numbers are expected, then set an input range to the above limits (see **f7** option).

---

## Main Operations Menu (cont'd)

The **f4 – READ ALPHA** creates a field which will allow entry of all displayable characters.

The **f5 – SPECIAL OPTIONS** which can be selected for reading fields are :

**AUTOEXIT:** When the operator has entered as many characters as are in the field, then the read is completed. 'Autoexit' is added to the Keyboard read statement.

**DON'T CLEAR:** Prevents the field from being cleared on the first character typed (see default). 'No Clear' is added to the Keyboard read statement.

**DEFAULT:** The previous value of the variable is shown instead of displaying a blank field. 'Default' is added to the Keyboard read statement.

**ZERO FILL:** Numeric field is right justified with zeros. 'Zerofill' is added to the Keyboard read statement.

**PASSWORD:** \*\*\*'s are shown instead of the characters typed. 'Secure' is added to the Keyboard read statement.

If the **f6 – ACCEPT LASER INPUT ALSO** is selected then the laser is included in the read statement.

**Note:** Range checks and other edit checks do not apply to this option.

When the **f7 – SET AN INPUT RANGE/LOOKUP** is entered the following sub-menu is displayed:

- (f1) – Help
- (f2) – Set smallest valid number
- (f3) – Set biggest valid number
- (f4) – Select valid characters
- (f5) – Lookup in DH5 file
- (f6) – Lookup in host file

**Smallest #:** You enter the lowest numeric number for the input value. For example, it can be entered as the number 20, or if the variable AGE = 20 then you can enter the variable name, AGE.

**Biggest #:** You enter the biggest numeric number for the input value. For example, it can be entered as the number 9999, or if the variable MOST = 9999 then you can enter the variable name, MOST.

When the program is running, if the value entered is not within the range, then \* OUT OF RANGE \* is displayed and the user is prompted to enter the number again.

## Main Operations Menu (cont'd)

**VALID CHARACTERS:** You specify a set of characters that can be used during entry. Anything not in the set will not be displayed.

This applies only to the one read statement you attach this attribute to. An example use might be: Departments are coded A, B, C, and D. When the operator enters the Department field, if the read statement has the valid character set as {A,B,C,D}, then no other characters will be displayed.

**Lookup in workstation file:** You enter the name of the Lookup File and the starting character position to compare.

**Lookup in Host file:** You enter the name of the Lookup File and the starting character position to compare. These lookups are performed by your host computer. Allen-Bradley Network Manager Software (Catalog No. 2708–NNM) V5.0 must be installed and RUNNING on your host computer in order for host lookups to work.

This allows a workstation program to always have up to the minute information about a file which is being maintained by the host computer.

The LOOKUP options allow you to restrict input to values which match those in a workstation file or in a HOST file. It is a method of verifying data against a file or obtaining records from a file. A text file must first be created using a text editor or another program.

Lookups can be selected here or in the sub–menu from the Move/Transfer/Add menu.

**Note:** When using Read lookups, the data read from the file is **ALWAYS** placed in the OUT variable.

If the **f8 – SET A TIMEOUT** option is chosen, you are given the choice of setting the timeout from seconds to minutes. The following sub–menu is displayed:

- (f1) – Help
- (f2) – Timeout after 5 seconds
- (f3) – Timeout after 15 seconds
- (f4) – Timeout after 30 seconds
- (f5) – Timeout after 1 minute
- (f6) – Timeout after 2 minutes
- (f7) – Timeout after 5 minutes
- (f8) – Key in timeout period

Timeouts are a way of stopping a read operation before the operation has finished. If your program is waiting for the user to enter his password and the user walks off and leaves the workstation, you want your program to restart after a given amount of time.

## Main Operations Menu (cont'd)

When you select a timeout you are also asked to specify where you want to branch to when the timeout occurs. To select the GOTO statement, you can use the arrow keys to select the destination statement or enter the statement number.

This option is appended to the Read statement and replaces any previous timeout.

**Note:** You stay in the Read from Keyboard/badge Menu after selecting an attribute. More than one attribute can be selected for a single read statement. Be sure to press ENTER and return to the Main Menu for the read statement to be entered into your program.

### **F4 – Read From Laser Or Device**

The Read From Laser or Device menu is used to read fields and characters from devices other than the keyboard. When **f4** is pressed the main menu is replaced with these options:

- f1 : Help
- f2 : Read laser
- f3 : Read line from Host
- f4 : Read line from Com port
- f5 : Read line from Aux port (second communications port)
- f6 : Read single keystroke
- f7 : Get DSR for Com 1 to In
- f8 : Set a timeout

In the options **f2** through **f7**, you are prompted on the next to the last line: 'Read into Variable Name : ". **IN** is the default variable. Press ENTER to store the operation.

The **f8** option is appended to the Read statement and replaces any previous timeout.

### **F5 – Send To Host Or Device**

The **f5 – SEND TO HOST OR DEVICE** option is used to send data to the host, send data out a communications port, or to control the workstation LED indicators. When F5 is pressed the main menu is replaced with these options:

- f1 : Help
- f2 : Send variable to indirectly to Host
- f3 : Send variable to directly to Host (NET)
- f4 : Send variable to Com port
- f5 : Send variable to Aux port
- f6 : Set a communications line or LED

## Main Operations Menu (cont'd)

In the **f2** – Send variable to indirectly to Host option, the record sent is held in the workstation queue memory until the Host polls. The number of records sent when the Host is off-line is dependent of the amount of memory in the workstation.

In the **f3** – Send variable directly to Host (NET) option, a record can be sent only when the Host is on-line. If you send a record when the Host is off-line then the next record cannot be sent until the Host comes on-line and receives the record. Use this option only when you know the Host is on-line.

In the options **f2** through **f5**, you are prompted on the next to the last line: “Variable to send to ... .”. OUT is the default variable. Press ENTER to store the operation.

When **f6** is chosen the following menu appears:

- (f1) : Help
- (f2) : Turn ON an LED (light)
- (f3) : Turn OFF an LED (light)
- (f4) : Turn ON the COM DTR line
- (f5) : Turn OFF the COM DTR line
- (f6) : Turn ON the RTS line
- (f7) : Turn OFF the RTS line

In options (**f2**) and (**f3**), you are prompted : “Enter LED number (1 to 10):”. The numbers reference the function keys. The other options have no prompts, they just write a statement in your program to perform the operation.

### **F6 – Move, Transfer, Add, Etc.**

When **f6** is pressed, you are first prompted for a destination variable which will be modified by the operation. Then the main menu is replaced with these options:

- f1 : Help
- f2 : Move to a variable
- f3 : Append to a variable
- f4 : Append comma separated
- f5 : Append quoted
- f6 : Append with comma and quotes
- f7 : Add to a variable
- f8 : Subtract from a variable
- f9 : Transfer field to a variable
- f10 : Other operations

## Main Operations Menu (cont'd)

These options are described below:

**MOVE:** This is used to copy the contents of one variable to another, or to set a variable to a value or text string. To clear a variable set it to “ ”.

**APPEND:** Appends a variable or text string onto the end of another. This is often used in building a record for sending to the host.

**ADD:** This will convert the variable into a number and add a value.

**SUBTRACT:** This will convert the variable into a number and subtract a value.

**TRANSFER:** Transfers are the most powerful way of manipulating variables. These allow you to edit the contents of a variable in the same way you can edit the screen using the display operations.

The following sub-menu lists your options here:

- (f1) : Help
- (f2) : Transfer text
- (f3) : Transfer a variable
- (f4) : Transfer time
- (f5) : Transfer date

Variables, text, date and time can be positioned within the variable (this is shown on the second to the last line of the display). The <+> and <-> will change the field size and format, while left and right arrows are used for positioning.

It is best to be in single step mode while entering transfers since that is the easiest way to see the relationship between fields that are being formatted.

Transfers are fixed length operations that always pad with spaces. If this is not what you want, you may want to use Append or the advanced operations. When you position a variable using a transfer, be sure the field is long enough (using <+>).

**Note:** Be sure to observe the second to the last line on the CRT monitor for the position of the data being transferred into your variable.

**OTHER OPERATIONS:** The following sub-menu appears:

- (f1) : Help
- (f2) : Multiply
- (f3) : Divide
- (f4) : Negate
- (f5) : Make Numeric
- (f6) : Make Integer
- (f7) : Lookup in host file
- (f8) : Still Other Operations

## Main Operations Menu (cont'd)

**MULTIPLY:** Will operate on any variable.

**DIVIDE:** Will operate on any variable. It may result in a floating point number.

**NEGATE:** Is used to alter the sign of a value.

**MAKE NUMERIC:** Will get rid of any non-numeric parts to a text value. This is a side effect of other math operations.

**MAKE INTEGER:** Will remove the fractional part of a number.

**LOOKUP:** This option is described on page 9.

**STILL OTHER OPERATIONS:** The following sub-menu appears:

- (f1) : Help
- (f2) : Length of a variable
- (f3) : Left part of a variable
- (f4) : Right part of a variable
- (f5) : Middle part of a variable
- (f6) : Trim spaces off a variable
- (f7) : Upper case of a variable
- (f8) : Character to ASCII
- (f9) : ASCII Value to Character
- (f10) : Take Modulus of Variable

**LENGTH:** Returns the number of characters in a variable

**LEFT:** Isolates the left 'n' characters from a variable. Left 4 characters of IN = 092890, returns 0928.

**RIGHT:** Isolates the right 'n' characters from a variable. Right 2 characters of IN = ABCDEFG, returns FG.

**MIDDLE:** Is used to get data from the middle of a variable. Warning! If you use middle on a variable that is too short and try to get characters that are not present, AG will abort. The 2 middle characters, starting in position 3 will return the following:

Variable = ABCDEFG	returns	CD
Variable = ABC	returns	program abort

**TRIM:** Removes spaces from the front and back of a variable.

**UPPER CASE:** Converts a variable into upper case.

**CHAR and ASCII:** converts the exchange character values into ASCII and back to characters.

**MODULUS:** Is the positive remainder from a division operation.

**Note:** Watch the 'Contents of Variables' lines to see what each variable is set equal to as the program runs in the stepping mode.



## Main Operations Menu (cont'd)

### F7 – Other Operations

When **f7 – OTHER OPERATIONS** is selected the main menu is replaced with these options:

- f1 : Help
- f2 : Beep
- f3 : Pause (delay)
- f4 : Chain to personality
- f5 : Comment line
- f6 : GoSub (call subroutine)
- f7 : Return (from subroutine)
- f8 : Label
- f9 : On event GoTo line/label

**f2 – Beep:** When the **f2 – BEEP** option is chosen the following sub-menu appears:

- (f1) : Help
- (f2) : Simple Beep
- (f3) : Short Chirp
- (f4) : Error Boop
- (f5) : Key in tone & duration

**Beep:** The tone is set to 1000 hz for 0.25 (1/4) of a second.

**Chirp :** The tone is set to 2000 hz for 0.15 of a second.

**Boop:** The tone is set to 300 hz for 1 second.

**Key in.. .:** The workstation can produce sounds in the range of 100 to 3000 hz. A low sounding tone is around 200 hz. A high pitched tone is above 2000 hz.

The loudness of the sound will vary depending on the frequency. Your PC will only produce an approximation of the actual sound produced by the workstation.

**f3 – Pause:** You can pause your program for any number of seconds. Use this to leave a prompt on the display long enough for the user to read it.

While you are testing your program, even a pause as short as 0.1 seconds will take a full second to execute on the PC. When you generate your program, the downloaded version will pause for the selected interval.

## Main Operations Menu (cont'd)

**f4 – Chain to Personality:** If your program becomes too large to run as a single program then you will need to use CHAINING. This is a method of having one program start the execution of another.

In order to use chaining, you will need to put CHAIN operations in the main program to start up your subprogram. Then in the subprogram, you will need a CHAIN that restarts the main program when the operation is completed.

The chain operation does not save the contents of user variables so you will need to make sure that the operations are entirely separate in order for chaining to be useful.

**Chaining Requirements:** Note that to download the files, you must use the FILE DOWNLOAD feature of the Network Manager Software (Catalog No. 2708–NNM) V5.0. This is in the FILE SERVICES menu of the Network Manager Software and is NOT the same as PROGRAM DOWNLOAD. Both your main program and your subprograms must be loaded using FILE DOWNLOAD.

You can start the main program by making a small program containing only the statement: CHAIN “MAIN” and using PROGRAM DOWNLOAD.

Alternatively you can start the main program using the Start Program feature: NM 5.0. This feature is NOT NORMALLY accessible. You will need to add it to one of the NM menus before you can get to it.

**Global Variables:** Chained programs may make use of GLOBAL variables. These are variables whose value is accessible by all of the chained programs. The names of the global variables must be placed in a text file named “GLOBAL.DEF” in your program directory using a text editor. Put one variable name on each line.



**WARNING:**

Once you have created GLOBALS.DEF you may only ADD variable names to the end of the list. If you insert or delete old names then you will damage your existing AG Personalities that use globals.

---

A maximum of 99 global variables may be defined. Using a lot of globals will significantly slow down chaining your programs.

## Main Operations Menu (cont'd)

**Generating A-B Basic program:** Once your AG personality behaves as you desire, you will need to generate an A-B BASIC program that can be downloaded into the DH5 network. This program will behave almost exactly like the personality you have been interacting with in the AG environment.

**f5 – Comment line:** This allows you to put comments into your program. For example, the first line of the program might be a Comment line stating the program name and purpose. The comment line is identified by a single quote (‘) at the beginning of the line.

**f6 – GoSub (Call Subroutine):** A GOSUB is a way of calling a subroutine. The subroutine can do its job and then return to the line immediately after the GOSUB statement. This makes it easier to write programs that have the kind of structure found in high level languages such as A-B BASIC.

**f7 – Return (From Subroutine):** When you write the subroutine that is called by the GOSUB statement, you must be sure to put a RETURN operation at the last line of the GOSUB. If you execute too many GOSUBs without returning you will get a stack overflow error.

**f8 – Label:** Line numbers work well for writing simple programs that you will not have to change often. For better structured programs you should use LABELS. A label is a name for a line in your program. The advantage is that the GOTO or GOSUB will have a name that makes sense instead of a meaningless line number.

You can place a LABEL anywhere in your program using the LABEL option in the OTHER OPERATIONS menu. When you key in a GOTO, you can use the label name instead of a line number.

When you are prompted for the line/label name, if you key in a label name that does not currently exist, you will be asked if you would like to put the label somewhere in your program. This is a nice shortcut for entering the labels.

## Main Operations Menu (cont'd)

**f9 – ON Event GoTo Line/Label Operations:** The ON event operation is a way of branching to more than one place depending on the condition of a previous operation. If the condition is met then the program will branch to the line number or label you define. The previous operation is defined from the following menu selections:

- On Function key
- On In key
- On Out key
- On Enter key
- On Exit key
- On Laser input
- On Host input
- On Com input
- On Other menu

The other menu contains these conditions that you can test for:

- On Key input
- On Aux input
- On Clear key input

**Note:** The destination variable must be in the IN variable.

### Adding A-B BASIC Functions

A-B BASIC Functions are added to the Other Operations Menu by modifying the file AGUser.BAS. You may add up to 10 named statements which will appear in a sub-menu named Additional Operations.

### F8 – IF (Conditional)

An IF statement is another way of changing the order or program execution. It allows you to GOTO a statement IF a variable is in some state.

When **f8 – IF** is chosen you are asked for the Variable to compare. Then the main menu is replaced with this sub-menu:

- f1 : Help
- f2 : If equals ...
- f3 : If not equal ...
- f4 : If less than ...
- f5 : If greater than ...
- f6 : If in a range ...
- f7 : If out of range ...
- f8 : If starts with ...
- f9 : If ends with ...

## Main Operations Menu (cont'd)

In **f2** through **f5**, you are prompted with “Value (text in quotes) or Variable name to compare : ”, ie, you might have entered **IN** as the variable to compare immediately after pressing **f8**, then after you chose **f4** (If less than ..) you might enter 9999, if you want to be sure the number that was input into the IN variable was less than 9999. Then you are prompted to enter the GOTO statement number if your compare is true. You can enter the number or use the up/down arrows to find the statement and press enter.

In **f6** and **f7**, you are prompted to enter both the lower limit and the upper limit of the range. Then you are prompted to enter the GOTO statement number if your compare is true. You can enter the number or use the up/down arrows to find the statement and press enter.

In **f8** and **f9** the ‘starts with’ and ‘ends with’ are used to branch (GOTO) if the variable starts or ends with some piece of text. You will usually type a quote (“) in front of the text being checked.

After you have enter the compare data, (text as above, a numeric, or another variable), you are prompted to enter the GOTO statement number if your compare is true. You can enter the number or use the up/down arrows to find the statement and press enter.

### **F9 – GoTo Or Loop To Position**

A GOTO is a way of changing the order in which operations are performed in your program. A program which contains no GOTOs will do one step after another until it reaches the last step. Then it will stop.

Each GOTO specifies the LINE NUMBER that the program should advance to when the GOTO operation is reached. You can either key in a line number, enter a label or use the up and down arrow keys to find where you would like to go. Press ENTER when you have made your selection.

When choosing this option, note the prompt on the next to last line of the CRT; “Go to line number : “. You can enter the number or use the up/down arrows to find the statement and press enter.

### **F10 – File (New, Load, Save, Generate)**

When **f10 – FILE** is chosen the main menu is replaced with these options:

- f1 : Help
- f2 : Load personality file
- f3 : Save personality file
- f4 : New personality file
- f5 : Generate an A-B BASIC Program
- f6 : Generate & Download using NM
- f7 : Transfer to NM to download
- f8 : Temporary DOS Shell
- f9 : Exit to DOS (also ALT X)

## Main Operations Menu (cont'd)

The **f2** or **f3** options are for loading or saving your personalities.

A personality name is an 8 character name that contains only the letters A to Z and 0 to 9. The name is used to create DOS files which contain the personality.

Select personality names that will make them easy to remember. If you are in the \ AG directory, you can list all of your personalities from the DOS prompt by typing : DIR \*.AIP

The **f4** option will clear the operations area of the CRT of all program statements. The area will have only:

<Start of Program>

==> <End of Program>

You can now start creating your program.

The **f5** option is used to generate a A-B BASIC Program that can be downloaded into the workstation network.

Once your AG personality behaves as you desire, you will need to generate this program. It will behave almost exactly like the personality you have been interacting with in the AG environment.

The **f6** option is used to generate and download using Network Manager Software (Catalog No. 2708–NNM). The PC must be running NM V5.0 or later.

The **f7** option transfers you to NM to download you program. Use the Network Manager Software program to download your generated personality to your DH5 workstation.

The **f8** – Temporary DOS Shell option provides a temporary exit to DOS. Typing EXIT in DOS will return you to AG.

The **f9** – Exit to DOS (also ALT X) option allows you to save your personality before exiting to DOS.

© 1991 Allen-Bradley Company



A subsidiary of Rockwell International, one of the world's largest technology companies, Allen-Bradley meets today's automation challenges with over 85 years of practical plant floor experience. More than 13,000 employees throughout the world design, manufacture and apply a wide range of control and automation products and supporting services to help our customers continuously improve quality, productivity and time to market. These products and services not only control individual machines, but also integrate the manufacturing process while providing access to vital plant floor data that can be used to support decision-making throughout the enterprise.

With offices in major cities worldwide.

**WORLD HEADQUARTERS**  
Allen-Bradley  
1201 South Second Street  
Milwaukee, WI 53204 USA  
Tel:(414) 382-2000  
Telex:43 11 016  
FAX:(414)382-4444

**EUROPE/MIDDLE EAST/  
AFRICA HEADQUARTERS**  
Allen-Bradley Europe B.V.  
Amsterdamseweg 15  
1422 AC Uithoorn  
The Netherlands  
Tel:(31) 2975/43500  
Telex:(844) 18042  
FAX:(31) 2975/60222

**ASIA/PACIFIC HEADQUARTERS**  
Allen-Bradley (Hong Kong) Limited  
Room 1006, Block B, Sea View Estate  
2-8 Watson Road  
Hong Kong  
Tel:(852)887-4788  
Telex:(780) 64347  
FAX:(852)510-9436

**CANADA HEADQUARTERS**  
Allen-Bradley Canada Limited  
135 Dundas Street  
Cambridge, Ontario N1R 5X1  
Canada  
Tel:(519)623-1810  
FAX:(519)623-8930

**LATIN AMERICA  
HEADQUARTERS**  
Allen-Bradley  
1201 South Second Street  
Milwaukee, WI 53204 USA  
Tel:(414)382-2000  
Telex:43 11 016  
FAX:(414)382-2400