



ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

GA Basic
Quick Reference

For use with the
Peripheral
Communication
Module
Cat. No. 1775-GA

Table of Contents

i

Section	Title	Page
1	GA Basic Command Summary	2
2	GA Basic Function Summary	4
3	Using GA Basic Commands	6
3.A	Using Commands From the GAI Prompt or Within Procedures	7
3.B	Using I/O Commands	16
3.C	Operating the 1770-M11 System	20
3.D	Operating the 1770-SB Recorder	25
4	Using GA Basic Modifiers	26
5	Assigning Data to User Symbols	33
6	Using GA Basic Functions	34
7	Editing Procedures	42
8	Using GA Basic Expressions	44
9	Formatting Data	46
10	Identifying Channels	48
11	Using 1770-T4 Terminal Characters	50
12	Using the Conversion Tables	52
13	Error Codes	55

To the Reader

This publication does not try to teach you how to program a peripheral communication module. It is a summary of the GA Basic programming language that you use to operate the peripheral communication module and connected devices. For detailed information on the GA Basic programming language, refer to the Peripheral Communication Module User's Manual (publication 1775-6.5.4 formerly 1775-808).

Terminology

In this publication, we refer to the:

- Peripheral Communication Module (cat. no. 1775-GA) as the **1775-GA module**
- Peripheral Interface Adapter (cat. no. 1775-RM) as the **1775-RM adapter**
- Processor Mass Storage System (cat. no. 1770-M11) as the **1770-M11 system**
- Industrial Terminal System (cat. no. 1770-T4) as the **1770-T4 terminal**
- Data Cartridge Recorder (cat. no. 1770-SB) as the **1770-SB recorder**

1 GA Basic Command Summary

Command	Abbreviation	Command	Abbreviation
Addsystem	ADDS	Identify	ID
Adduser	ADDU	If	IF
Allocate	AL	Ifdef	IFD
Argument	AR	Ifndef	IFN
Assign	AS	Increment	INC
Break	BR	Input	INP
Btbidirection	BTB	Inquire	INQ
Btdeallocate	BTD	List	L
Btread	BTR	Makedir	MA
Btwrite	BTW	Message	ME
Case	CA	Next	NE
Changepass	CH	No_verify	NO
Close	CL	On_error	ON

Convert
Copy
Create
Deallocate
Deassign
Decrement
Delay
Delete
Dimension
Directory
Edit
Else
Endcase
Endif
Endwhile
Exit
For
Format
Freemain
Getmain
Goto

CON
COP
CR
DEAL
DEAS
DEC
DELA
DELE
DIM
DIR
ED
EL
ENDC
ENDI
ENDW
EX
FOR
FORM
FR
GE
GO

Open
Outchar
Print
Randomize
Read
Remuser
Rename
Restore
Rewind
Setcontext
Setdir
Setprot
Setrecord
Show
Stop
Store
Verify
Wait
While
Write

OP
OU
P
RA
REA
REM
REN
RES
REW
SETC
SETD
SETP
SETR
SH
STOP
STOR
V
WA
WH
WR

Section

2 GA Basic Function Summary

Function	Abbreviation	Function	Abbreviation
Abs	AB	Locate	LOC
Arctan	AR	Log	LOG
Ascii	AS	Max	MA
Btalloc	BTA	Mid	MID
Btrdone	BTRD	Min	MIN
Btrerror	BTRE	Outstat	O
Btwdone	BTWD	Rad	RAD
Btwerror	BTWE	Rand	RAN
Chr	CH	Repeat	RE
Cos	CO	Right	RI
Date	DA	Round	RO

Decode	DEC	Sign	SIG
Degree	DEG	Sin	SIN
Encode	EN	Sizeof	SIZ
Eof	EO	Sqrt	SQ
Exp	EX	Tab	TAB
Float	FL	Tan	TAN
From_bcd	FR	Testchar	TESTC
Getchar	G	Testline	TESTL
Int	I	Time	TI
Left	LEF	To_bcd	TO
Length	LEN	Trim	TR
Levels	LEV	Waitchar	W
Ln	LN		

Section

3 Using GA Basic Commands

You can enter GA Basic commands from the GA1> prompt or within procedures to tell the 1775-GA module what to do. The following tables summarize the GA Basic commands:

Section

3.A Using Commands From the GA1> Prompt or Within Procedures

If you want to:	Then use this command:	With this format:
Retrieve arguments passed to this procedure	Argument	AR(<user symbols>)
Assign a value to a user symbol	Assignment	<destination> <assignment statement> <source>
Terminate a for or while loop before normal completion	Break	BR
Execute block transfers ¹	Btread Btwrite Btbidirection	BTR, BTW, or BTB/<modifier> \$<logical address>
Call for a procedure to execute based on the value of an expression ²	Case	CA (<expression>) @<procedure list> or block of commands with endcase command (ENDC)

If you want to:	Then use this command:	With this format:
Convert a procedure into a faster executing form	Convert	CON/<modifiers> @<original name> @<converted name>
Copy a procedure or disk	Copy	COP/<modifiers> @<original name> /<modifiers> @<copy name>
Create a symbolic address	Create	CR/<scope modifier> @<system symbol> \$<logical address>
Subtract 1 from a specified value	Decrement	DEC/<scope modifier> <operand>
Pause procedure execution for a specified number of milliseconds	Delay	DELA <value or expression>
Remove a system symbol or user symbol	Delete	DELE/<modifier> <@system symbol> or <user symbol>

Display a list of system symbols stored in PLC-3 memory or files stored on 1770-M11 system	Directory	DIR/<modifier> or DIR/<modifier> <device location or file specification>
Declare an array	Dimension	DIM/<modifier>/<data type> <array name> <size>
Create a procedure or change an existing procedure	Edit	ED/<modifier> @<procedure name>
Execute the command lines that make up a procedure or nest executing procedures	Execute	@<procedure name> or @<procedure name>(<arguments>)
Stop procedure execution and return to GA1> prompt or calling procedure	Exit	EX

If you want to:	Then use this command:	With this format:
Execute a command or block of commands a specified number of times ²	For	FOR (<user symbol>=<expression> <limit>, <step>) <command> or block of commands with next command (NE)
Remove a section of PLC-3 memory	Freemain	FR \$<logical address>
Create an additional section in PLC-3 memory	Getmain	GE \$<logical address>
Jump to a label in the procedure	Goto	GO <label name>
Execute a command or block of commands if a specified condition is true ²	If	IF (<expression>) <command> or block of commands with endif command (ENDI)

Execute commands if the conditions set on the if command line are not true ²	If	IF then EL with endif command (ENDI)
Execute command or block of comand if a specified user symbol or logical address is defined ²	Ifdef	IFD <user symbol or logical address> <command> or block of commands with endif command (ENDI)
Execute command or block of commands if a specified user symbol or logical address is not defined ²	Ifndef	IFN <user symbol or logical address> <command> or block of commands with endif command (ENDI)
Add 1 to a specified value	Increment	INC/<scope modifier> <operand>
Make the 1775-GA module wait for data and a carriage return to be entered before executing the rest of the procedure	Input or Inquire	INP or INQ/<modifiers> <user symbol> <data specifiers>

If you want to:	Then use this command:	With this format:
Access the PLC-3 LIST function	List	L or L @<procedure name>
Send messages to other modules in the PLC-3 system ¹	Message	ME \$<logical address>
Cancel the verify command	No_verify	NO
Execute a command if an error occurs during procedure execution	On_error	ON <action command statement>
Print out the contents of a procedure or other information	Print	P @<procedure name> or P <data specifier>
Set the random number generator	Randomize	RA or RA <integer value>

Change the name of a procedure	Rename	REN/<modifier> @<old name> @<new name>
Change the operating context for the execution task	Setcontext	SETC or SETC <value or expression>
Display status for a logical unit	Show	SH/LU #<logical unit number>
Display the active privileges for the channel	Show	SH/PRIV
Display device attributes for the channel	Show	SH/CONF \$<device location>
Display context for the execution task and the PLC-3 system	Show	SH/CONT

If you want to:	Then use this command:	With this format:
Display 1770-M11 system identification	Show	SH/U \$<device location> <value or expression>
Stop procedure execution and output an error message	Stop	ST <data specifier>
Check procedure execution command line by command line	Verify	V
Pause procedure execution until a specified time is reached	Wait	WA<hour-value><minute-value> <second-value>
Execute a command or block of commands while a specific expression is true ²	While	WH (<expression>) <command> or block of commands with endwhile command (ENDW)

- 1 For the block transfer and message commands, you must set up the control files before executing the commands.
- 2 Commands that use the block format allow you to enter multiple command lines in accordance with the command. You must use the next, endcase, endif, or endwhile commands to end the corresponding command block. For example, the block command format for the IF command would be:

```
IF (<expression>) <command>  
    <command>  
    .  
    .  
    .  
    <command>  
ENDI
```

Section

3.B

Using I/O Commands

If you want to:	Then use this command:	With this format:
Access the communication link for a device	Allocate	AL #<logical unit number>
Set a communication link between a channel on the 1775-GA module and a device or file location	Assign	AS/<modifiers> #< logical unit number> <\$device location and/or @file name>
Define the scope and/or section for a file	Assign	AS/<scope modifier>/<section modifier> @<file name>

Specify a file or device for read or write commands	Assign	AS/M #<logical unit number> <\$device location and/or @file name>
Set the internal PLC-3 file pointer so that it does not advance after each file operation	Assign	AS/NO_A #<logical unit number> @<file name>
Disable character editing for a device or file	Assign	AS/PA #<logical unit number> <\$device location and/or @file name>
Specify a file for read commands only	Assign	AS/REA #<logical unit number> @<file name>
Specify the number of bytes in each record and the fill character for records less than the specified size	Assign	AS/SI=n/FI='<character>' #<logical unit number> <\$device location and/or @file name>
Set a string to define the end of a record	Assign	AS/TERMINAT='<string>' #<logical unit number> <\$device location and/or @file name>

If you want to:	Then use this command:	With this format:
Specify a file for write commands only	Assign	AS/WR #<logical unit number> @<file name>
Release access to the file	Close	CL #<logical unit number>
Release access to the device	Deallocate	DEAL #<logical unit number>
Break the communication link established by the assign command	Deassign	DEAS #<logical unit number>
Make the 1775-GA module wait for data and a carriage return to be entered from logical unit 1 and store the data in a specified variable	Input or Inquire	INP or INQ/<modifiers> <user symbol> <data specifiers>
Access the communication link for a file	Open	OP #<logical unit number>

Print out information at logical unit 0	Print	P/<modifiers> <data specifiers>
Take in values from a logical unit and store these values in specified variables	Read	REA/<modifiers> #<logical unit number> <data specifiers>
Set the internal PLC-3 file pointer to a specific record in a file	Setrecord	SETR #<logical unit number> <record number>
Send data to a logical unit	Write	WR/<modifiers> #<logical unit number> <data specifiers>

Section

3.C Operating the 1770-M11 System

You can operate the 1770-M11 system through the PLC-3 LIST function to save/load the contents of PLC-3 memory. You can also use the following GA Basic commands to create and operate a database:

If you want to:

Add a new system manager
to the 1770-M11 system

Then use

this command:

Addsystem

With this format:

ADDS <disk location> <system manager
name>

Add a new user to the 1770-M11 system	Adduser	ADDU <disk location> <user name>
Collect and acquire files stored on a disk	Copy	COP <disk location>@<original file name> <disk location>@<copy name>
To copy an entire disk to another disk that is the same size	Copy	COP/DEV <original disk location> <copy disk location>
Change your password or a user's password if you are a system manager	Changepass	CH <disk location>
Remove a file from a disk	Delete	DELE/<modifier> <disk location>@<file name>
Display a list of files stored on the specified disk	Directory	DIR<disk location>

If you want to:	Then use this command:	With this format:
Change the name of a file stored on a disk	Rename	REN/<modifier> <disk location>@<old name> @<new name>
To recover the data stored on a series of micro-floppy disks by the store command	Restore	RES <micro-floppy disk location> <Winchester disk file location>@<file name> or RES <micro-floppy disk location> <Winchester disk location>
Set the default directory for the disk	Setdir	SETD <disk location> <directory name>. <subdirectory name>...
Set the default directory back level(s) from the current default directory	Setdir	SETD <disk location> —...

Set the default directory to the top level directory	Setdir	SETD <disk location>
Set file protection for a particular file or all files that you create on the 1770-M11 system	Setprot	SETP/<modifiers> <disk file location> or SETP/<modifiers>
Display the user name associated with an owner number	Show	SH/U<disk location><owner number>
To create a backup copy of a file or disk on a series of micro-floppy disks	Store	STOR <disk file location>@<file name> <micro-floppy disk location> or STOR <Winchester disk location> <micro-floppy disk location>

If you want to:	Then use this function:	With this format:
Display a list of files stored on a subdirectory of the specified disk	Directory	DIR <disk location>.<directory name>
Initialize a new disk or reinitialize an old disk that contains data no longer valid	Format	FORM <disk location> <disk type>
Identify yourself as an identified user to the 1770-M11 system	Identify	ID <disk location> <user name> <password>
To create a subdirectory on a disk	Makedir	MA <disk location> <directory name>
To remove a user or a system manager from the 1770-M11 system	Remuser	REM <disk location> <user or system manager name>

Section

3.D Operating the 1770-SB Recorder

You can operate the 1770-SB recorder through the 1770-T4 terminal or an RS-232-C channel on the 1775-GA module or 1775-RM adapter. Through the LIST function, you can save/load the contents of PLC-3 memory to/from the 1770-SB recorder.

IMPORTANT: Tapes made through the 1770-T4 terminal are not compatible with tapes made through the 1775-GA module. They are of different formats and cannot be interchanged.

In addition, if you use the 1775-GA module, you can read and write data to/from the 1770-SB recorder. However, the 1770-SB recorder does not format the data on tape. To assist you in formatting the data, GA Basic supports the rewind command:

If you want to:	Then use this command:	With this format:
Rewind the 1770-SB recorder	Rewind	REW <1770-SB recorder location>

Section

4 Using GA Basic Modifiers

You can specify modifiers with commands to tell the 1775-GA module how to execute the command. You enter a slash (/) following the command name to delimit a modifier. The GA Basic modifiers include:

Modifier and Abbreviation	Applies to:	Function
Assist /A	Assign, Copy, Edit	Defines the assistance (HELP) section of the message area of PLC-3 memory
Comment /COM	Assign, Copy, Edit	Defines the rung comment section of the message area of PLC-3 memory
Config /CONF	Show	Displays the attributes of the specified device connected to the 1775-GA module

Context /CONT	Show	Displays the current operating context for the execution task and the PLC-3 system
Data /DA	Assign, Copy	Defines a file that contains data such as paragraphs of text.
Delete /DEL	Convert, Copy	Replace a copied or converted procedure with an updated version
Device /DEV	Copy	Copy the entire contents of a disk
Directory /DI	Assign, Copy	Defines a file that contains a subdirectory of files stored on the 1770-M11 system.
Fill /FI	Assign	Defines the character that the 1775-GA module displays if the output contains less characters than the specified size for each record

Modifier and Abbreviation	Applies to:	Function
Float /FL	Dimension	Defines a floating point data type for the specified array
Global /G	Assign, Convert, Copy, Create, Decrement, Delete, Edit, Increment, Rename	Defines that the specified user symbol is recognized in any context
Highway /H	Assign, Copy, Edit	Defines the data highway section of the message area of PLC-3 memory
Integer /IN	Dimension	Defines an integer data type for the specified array
Interprocedural /NP	Decrement, Delete, Dimension, Increment, Input, Inquire	Defines that the specified user symbol is recognized in any procedure

Local /L	Assign, Convert, Copy, Create, Decrement, Delete, Edit, Increment, Rename	Defines that the specified system symbol is recognized only in the procedure that it was created
Lunit /LUN	Show	Displays information on the device or file assigned to the specified logical unit
Memory /ME	Assign, Copy, Show	Defines a file containing the contents of PLC-3 memory stored on a disk drive of the 1770-M11 system or shows the amount of memory left on-board the 1775-GA module
Modify /MO	Assign	Defines that the specified file accepts read or write commands
Noadvance /NO_A	Assign	Defines that the internal PLC-3 file pointer does not advance after reading a record in the specified file

Modifier and Abbreviation	Applies to:	Function
Noecho /NOE	Read, Input, Inquire	Turns off the echo parameter so that the PLC-3 processor does not display the characters for the data stored by the command
Passall /PA	Assign	Disables execution of control characters for the specified device or file
Privileges /PRI	Show	Displays a list of current privileges assigned to the channel that issues the command
Procedural /PRO	Decrement, Delete, Dimension, Increment, Input, Inquire	Defines that the specified user symbol is recognized only in the procedure it was created in
Read /REA	Assign	Defines that the specified file accepts only read commands

Report /REP	Assign, Copy, Edit	Defines the report generation or GA Basic section of the message area
Rgconv /RG	Assign, Copy	Defines the converted procedures section of the message area
Serial I/O /SIO	Read, Write	Controls and monitors the RS-232-C handshaking control lines
Size /SI	Assign	Defines the number of bytes that make up each record in the specified device or file
String /ST	Dimension	Defines a string data type for the specified array
Suppress /SU	Directory, Print, Write	Turns off the pagination function on the directory, or turns off the line number display when outputting data

Modifier and Abbreviation	Applies to:	Function
Terminal /TERMINAL	Assign, Copy, Edit	Defines the industrial terminal macro section of the message area
Terminator /TERMINAT	Assign	Defines a string that represents the end of record for each record in the specified file or device
User /U	Show	Displays the user name that corresponds to the specified owner number on the 1770-M11 system
Write /WR	Assign	Defines that the specified file accepts only write commands
Wait /WA	Btbidirection, Btread, Btwrite, Message, Print, Write	Delays procedure execution until the command completes executing before executing the next command line

5 Assigning Data to User Symbols

You can use GA Basic user symbols to assign numerical values or strings of data to variables at the GA1> prompt or within procedures.

If you want to:

Assign a numeric value to a procedural user symbol

Assign a numeric value to an interprocedural user symbol

Assign a string of data to a procedural user symbol

Assign a string of data to an interprocedural user symbol

Then do this:

<user symbol> = <numeric value>

<user symbol> == <numeric value>

<user symbol> =? '<string of data>'

<user symbol> ==? '<string of data>'

Section

6 Using GA Basic Functions

GA Basic functions are auxiliary commands that you can use with commands to tell the 1775-GA module what to do. On a command line, you must accompany a function with a command for proper execution. These functions include:

If you want to:	Then use this function:	With this format:
Return the absolute value of an expression	Abs	AB(<expression>)
Return the arctangent of an expression	Arctan	AR(<expression>)

Convert an ASCII value to its integer equivalent	Ascii	AS(<character>)
Convert character(s) to the specified code equivalent	Chr	CH(<value>)
Return the cosine of an expression	Cos	CO(<expression>)
Return the current date	Date	DA()
Store values from a specified string in specific variables	Decode	DEC(<string> ,<arg1>,<arg2>,...)
Convert an expression from radians to degrees	Degree	DEG(<expression>)
Write characters from specified variables to create a string	Encode	EN(<string>,< arg1>,<arg2>,...)

If you want to:	Then use this function:	With this format:
Check if the 1775-GA module is pointing at the last record in a file	Eof	EO(#<logical unit number>)
Return the inverse natural logarithm of an expression	Exp	EX(<expression>)
Convert an expression to a floating point number	Float	FL(<expression>)
Convert an expression from BCD to its binary equivalent	From_bcd	FR(<expression>)
Store the next character from a specific file or device	Getchar	G(#<logical unit number>)
Convert an expression to a 32-bit integer	Int	I(<expression>)

Create a substring consisting of the first n characters of a specified string	Left	LEF(<string> ,<length>)
Return the number of characters in the specified string	Length	LEN(<string>)
Return the number of sub-levels at a logical address	Levels	LEV(\$<logical address>)
Return the natural logarithm of an expression	Ln	LN(<expression>)
Return the position of one string inside another string	Locate	LOC(<string1>,<string2>)

If you want to:	Then use this function:	With this format:
Return the base-10 logarithm of an expression	Log	LOG(<expression>)
Return the maximum value from a set of expressions	Max	MA(<expression>,<expression2,&br/><expression3> ,...)
Create a substring from the specified string	Mid	MID(<string> ,<start value>,<length>)
Return the minimum value from a set of expressions	Min	MIN(<expression>,<expression2>, <expression3>,...)
Return the execution status of the last output operation	Outstat	O()
Convert an expression from degrees to radians	Rad	RAD(<expression>)
Return a random number	Rand	RAN()

Return a string of characters n times	Repeat	RE(<string>, <value>)
Create a substring consisting of the last n characters of a specified string	Right	RI(<string>, <length>)
Round a floating point number to the nearest whole number	Round	RO(<floating point value>)
Return the sign of an expression	Sign	SIG(<expression>)
Return the sine of an expression	Sin	SIN(<expression>)
Return the number of words at a specified logical address	Sizeof	SIZ(\$<logical address>)
Return the square root of an expression	Sqrt	SQ(<expression>)

If you want to:	Then use this function:	With this format:
Display uniform columns of text	Tab	TAB(<tab position>)
Return the tangent of an expression	Tan	TAN(<expression>)
Test a specific file or device for a character	Testchar	TESTC(#<logical unit number>)
Test a specific file or device for a line of data terminated by a carriage return	Testline	TESTL(#<logical unit number>)
Return the current time	Time	TI()
Convert an expression to its BCD equivalent	To_bcd	TO(<expression>)

Return a string with leading
and trailing spaces removed

Trim

TR(<string>)

Store the next character from
a specific file or device

Waitchar

W(#<logical unit number>)

Section

7**Editing Procedures**

You can use the following editing commands to edit GA Basic procedures:

If you want to:**Then in the edit mode enter:**Insert lines¹

I

Set the line pointer

<line number>

Advance the line pointer

<number of lines>A

Display the line number

L

Type out lines

<number of lines>T

Search for text²

<number of occurrences>S/<text>/

Change text characters²

<number of occurrences>C/<old text>/

<new text>/<number of lines>A

Delete lines

<number of lines>D

Backup the line pointer	<number of lines>B
Move lines	M<start line>,<end line>
Transfer lines	X<start line>,<end line>
Read in a procedure	R@<procedure name>
Write out lines	<number of lines>W</modifiers> @<procedure name>
List editing commands	H
Exit edit mode and return to GA1> prompt level	E

- ¹ You can then add any number of lines. To exit the insert mode, enter an extra [ENTER] after any line.
- ² You can use any ASCII character as the delimiter as long as it does not occur in the text that you are searching for or changing. We use the slash (/) in this table.

Section

8 Using GA Basic Expressions

You can use the following expression operators to perform mathematical, comparison, or bitwise operations:

Order of Execution	Operator	Description
~ or .BNOT.	1	Bitwise 32-bit complement
.NOT.	1	Logical complement
/	2	Bit test
**	2	Exponent
*	3	Multiply
%	3	Divide
.MOD.	3	Modulo divide
+	4	Add
-	4	Subtract
+	4	String concatenate

>>	5	Shift right
<<	5	Shift left
& or .BAND.	6	Bitwise 32-bit AND
^ or .BXOR.	7	Bitwise 32-bit EXCLUSIVE OR
or .BOR.	8	Bitwise 32-bit OR
.EQ.	9	Compare equal
.GE.	9	Compare greater than or equal
.GT.	9	Compare greater than
.LE.	9	Compare less than or equal
.LT.	9	Compare less than
.NE.	9	Compare not equal
.SNE.	9	String compare not equal
.SEQ.	9	String compare equal
.AND.	10	Logical AND
.OR.	11	Logical OR

Order of execution moves from 1 to 11 with 1 executing first. When operators have the same order of execution number, the order for their execution within an expression is from left to right on the command line.

Section

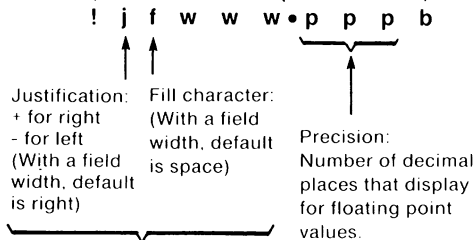
9 Formatting Data

In GA Basic, you can enter the ! delimiter following a user symbol or an expression to format input or output data. In the data format specification given below:

- ! = start of the data format
- j = justification
- f = fill character
- w = width of field
- p = precision
- b = base

Width of field. If not specified or **0**, printing starts on the left and continues through a field width necessary to hold the value. Therefore, justification and fill character specifications do not apply.

Delimits the start of format specification.



Specify both or neither. Do not specify when field width is not specified or **0**.

Base:

H Hexadecimal

O Octal

D Decimal

B Binary

S String

X Spaces

F Fixed Point

E Exponential

L Logical

US Unformatted Signed Integer

UU Unformatted Unsigned Integer

UF Unformatted Floating Point

If you enter a format delimiter, you must specify the base. With no delimiter, the default is decimal for integer values, exponential for floating point values, and string for string values.

Section

10 Identifying Channels

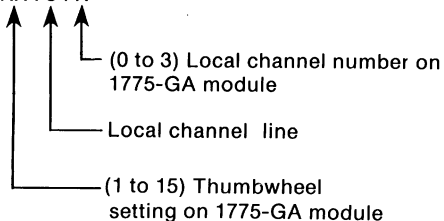
You can use the following formats to identify an RS-232-C channel on the 1775-GA module or 1775-RM adapter, or a disk drive on the 1770-M11 system:

If you are identifying:

Local RS-232-C channel on
1775-GA module

Then the format is:

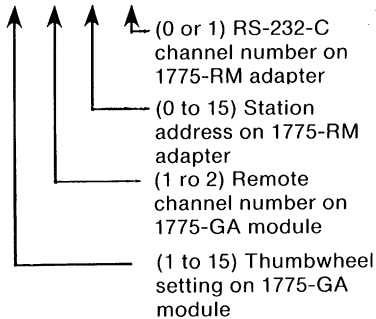
GA XX . 3 . X



If you are identifying: Then the format is:

RS-232-C channel on the 1775-RM adapter

GA XX . X . XX . X

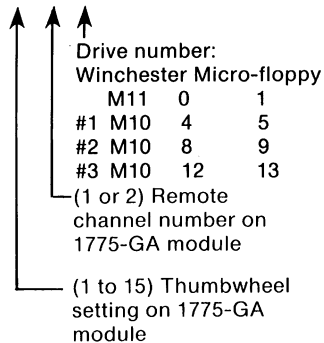


If you are identifying:

Disk drive on the 1770-M11 system

Then the format is:

GA XX . X . X



Section

11 Using 1770-T4 Terminal Characters

The following table lists control codes unique to the 1770-T4 terminal:

Control Code Key Sequence	Function
[CTRL] P [Column #] ; [Line #] A [CTRL] P F [CTRL] P U [CTRL] P 5 C	Positions the cursor at the specified column and line number. [CTRL] P A will position the cursor at the top left corner of the screen. Moves the cursor one space to the right. Moves the cursor one line up in the same column. Turns cursor ON.

[CTRL] P 4 C	Turns cursor OFF.
[CTRL] P 5 G	Turns ON graphics capability.
[CTRL] P 4 G	Turns OFF graphics capability.
[CTRL] P 5 P	Turns Channel C ON.
[CTRL] P 4 P	Turns Channel C OFF.

Key Sequence	Attribute 1
[CTRL] P 0 T [CTRL] P 1 T [CTRL] P 2 T [CTRL] P 3 T [CTRL] P 4 T	Attribute 0 = Normal Intensity Attribute 1 = Underline Attribute 0 = Intensity Attribute 0 = Blinking Attribute 0 = Reverse Video
1 Any three attributes can be used at one time using the following key sequence: [CTRL] P [Attribute #] ; [Attribute #] T.	

12 Using the Conversion Tables

The following table converts an ASCII bit pattern to its decimal, hexadecimal, and octal equivalents. The table is divided into four columns:

- Column 1 contains all the control characters.
- Column 2 contains numbers and symbols.
- Column 3 contains the capital letters. If you press the control key [CTRL] or [^] and a capital letter, the control code matches the control character in the first column. For example: [CTRL] G is the control character [BEL].
- Column 4 contains lower case letters and symbols.

Decimal/Hexadecimal/Octal/ASCII Conversion Table

Column 1				Column 2				Column 3				Column 4			
DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC
00	00	000	NUL	32	20	040	SP	64	40	100	@	96	60	140	\
01	01	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
02	02	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
03	03	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
04	04	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
05	05	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
06	06	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
07	07	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
08	08	010	BS	40	28	050	(72	48	110	H	104	68	150	h
09	09	011	HT	41	29	051)	73	49	111	I	105	69	151	i
10	0A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	S0	46	2E	056	.	78	4E	116	N	110	6E	156	n

15	0F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	:	91	5B	133	[123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135]	125	7D	175	~
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	-	127	7F	177	DEL

Section

13 Error Codes

The 1775-GA module generates the following codes when it detects an error during execution:

Error Code	Description
1	Read error in executing inquire, input, or read command
2-6	Unused
7	Control C has been entered from data terminal keyboard
8-10	Unused
11	Illegal operands for arithmetic operation
12	Illegal arithmetic operator
13	Missing parenthesis in expression
14	Illegal expression syntax
15	Illegal operator in expression
16	Illegal right side of arithmetic assignment

17	Illegal data following address
18	Illegal string assignment
19	Unused
20	Division by zero
21	System symbol already defined
22	Unused
23	System symbol is not a symbolic address
24	Illegal left side of assignment
25	Illegal modifier
26	Must specify system symbol for create command
27	Illegal data for system symbol
28	Unused
29	System or user symbol to delete not found
30	Procedure name already defined
31-32	Unused
33	Must use system symbol
34	Unused
35	Illegal field width in format specification
36	Unused

Error Code	Description
37	Illegal expression on case command line
38	Procedure not found for case command
39	Unused
40	Unrecognized or illegal command at GA1> prompt level
41	Unused
42	Illegal data following goto command
43	Label entered at GA1> prompt
44	Label not found
45	Duplicate label
46	Converted procedure cannot be printed
47	Insufficient privilege
48-50	Unused
51	Error reading original procedure for copy command
52	Unused
53	On_error command entered at GA1> prompt level
54	Field size greater than 128 characters on assign command line
55	Resultant string exceeds 255 characters

56	Unused
57	Symbol undefined
58	Unused
59	Illegal level specified in logical address
60	Unrecognized data table section specifier
61	Illegal timer, counter, or pointer format
62	Illegal timer, counter, or pointer word specification
63	Missing colon between file and word in data table section specifier
64	Unused
65	Illegal context specifier in setcontext command
66	Attempt to execute a system symbol that is not defined as a procedure
67	Insufficient PLC-3 memory for operation
68	Arithmetic overflow
69	Illegal bit specified in bit write
70	Cannot write a bit in user symbols
71	Illegal data for bit write (must be a 1 or 0)
72	Must use = to assign values to logical or symbolic addresses

Error Code	Description
73	Undefined assignment statement
74	Illegal fill character (e.g. a carriage return)
75	Need user symbol in inquire command
76	Illegal string format specified
77	Unused
78	Unused
79	Stop command encountered in procedure
80	Exit command encountered in procedure
81	Attempt to edit invalid system symbol (e.g. symbolic address)
82	Illegal function argument
83	Unexpected end of argument list
84	Unexpected end of input data
85	Unable to open file
86	Error writing to copy procedure for copy command
87	Invalid base specifier for command
88	Attempt to read or write at an illegal address
89	Unable to evaluate value in given base format
90	Illegal data for print command

91	Illegal data entered for inquire, input, or read command
92	Unused
93	Illegal string
94	Expression is too complex
95	Unused
96	Illegal device address
97	Missing logical unit number
98	Illegal logical unit number
99	Logical unit already allocated
100	Assign command execution did not occur
101	Illegal assign parameter
102	Illegal directory, file, or user name
103	Illegal terminator in assign command
104	Illegal fill character in assign command
105	Deassign command execution did not occur
106	Duplicate directory name
107	Duplicate file name
108	Logical unit #1 deallocated or deassigned
109	Logical unit not allocated

Error Code	Description
110	Error decoding input in read command
111	Unable to communicate with scanner for block transfer
112	Illegal block control file; must be a binary file
113	Unused
114	Mismatched block statements
115	Insufficient on-board memory
116	Unused
117	Unused
118	Array not declared with dimension command
119	Array index out of bounds
120	Incorrect number of array dimensions specified
121	Data type does not match array type
122	Illegal index specification
123	Unused
124	Illegal nesting of loop commands (if, ifdef, ifndef, for, while)
125	Illegal array name
126	Block transfer already requested
127	Illegal procedure argument

128	No index specified for array
129	File locked by another user
130	LIST not available
131	Disk or tape error
132	Invalid password
133	Error detected in file operation; setrecord command failed
134	Integer required
135	Illegal value
136	Command line too complex
137	Illegal user number

Important User Information

Because of the variety of uses for the solid state equipment described herein, and because of the differences between it and electromechanical equipment, you must satisfy yourself as to its acceptability for each of your applications. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages that may result from installation or use of this equipment.

The illustrations, charts, and layout examples shown in this manual are intended solely to help you understand the text, not to guarantee operation. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company will not assume responsibility for actual use based upon illustrations of applications.

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

Reproduction of any part of this manual, without written permission of Allen-Bradley Company, is prohibited.

© 1984 Allen-Bradley Company

PLC is a registered trademark of Allen-Bradley Company



ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Programmable Controller Division
747 Alpha Drive, Cleveland, Ohio 44143

P/N 955095-38
© 1984 Allen-Bradley Company

Publication 1775-7.1 — December, 1984