

HP Series 200 Models 16, 20, 26, 36 and 36C BASIC 2.0 Language System and BASIC Extensions 2.1

 HEWLETT
PACKARD



Technical Data, July 1983



Introduction

Hewlett-Packard's Series 200 Personal Technical Computers are based on the MC68000, a powerful 16-bit Motorola microprocessor with 32-bit internal architecture and 8 MHz clock rate. Series 200 is available in five models — Models 16, 20, 26, 36, and 36C.

The smallest, Model 16, fits easily on a desk. It features a 9-inch (229 mm) CRT, small, detached keyboard and memory from 128K to 768K bytes. A variety of disc drives, printers and other peripherals are available to convert it into a stand-alone system; or it can be connected to a Shared Resource Management (SRM) system with other HP computers, using printers and disc drives in common. It features both HP-IB* and RS-232-C built-in interfaces. The Model 16 is a high-performance personal computer.

The Model 20 is a modular, rack-mountable computer that can be separately configured with a keyboard, monitor, disc drive, and an assortment of peripherals. The Model 20 can be configured with up to 3.9 megabytes of memory. With extra interface and memory slots, plus the capability to remove the keyboard and monitor, the Model 20 is well-suited for computer-aided test (CAT) areas such as a factory assembly line or testing.

The Model 26 features a 7-inch CRT, built-in 5¼-inch flexible disc drive with 264K-byte capacity and up to 2 megabytes memory. Having a larger base and keyboard than the Model 16, it is suited to placing in a rack with other electronic devices and instrumentation. With built-in HP-IB interface, it finds its ideal

application in computer-aided test (CAT) applications where a more integrated system is required.

The Model 36 features a full 12-inch (310 mm) CRT, dual, built-in 5¼-inch flexible disc drives, and up to 2 megabytes of memory. Its larger screen adapts it more to computer-aided engineering (CAE) applications such as engineering design and analysis. Like the Model 26, it features a built-in HP-IB interface. Its large mass storage capacity (528K bytes) also supports these more demanding applications.

The Model 36C is the top-of-the-line Series 200 computer, with a full 12-inch (310 mm) color CRT, dual, built-in 5¼-inch flexible disc drives and up to 2 megabytes of memory. With its color capability, this is the ideal machine for computer-aided engineering (CAE) applications where the additional graphics versatility of color is useful.

Series 200 machines feature BASIC, Pascal and HPL,[†] graphics, user-definable soft keys, a special "rotary control knob" for easy editing and simulations, memory-mapped I/O, prioritized interrupt, internal system clock and timers, and built-in slots for additional memory or interface cards. In addition, a special backplane expander allows up to an additional 4 megabytes of memory or eight interface cards to any Series 200 computer.

Graphics, keyboard, and monitors are ordered separately on the Model 20.

* HP-IB is Hewlett-Packard's implementation of IEEE Standard 488-1978.
† HPL is not supported on Models 20 and 36C.

BASIC 2.0 Capabilities

The BASIC 2.0 Language System is anything but basic. It adds to the inherent simplicity of BASIC the computational power usually found in FORTRAN, ALGOL and APL. Beyond that, high-performance, I/O-intensive constructs are incorporated, giving this BASIC the highest performance found in interactive instrument control systems on the market today.

BASIC 2.0 puts mini-computer architecture and power on YOUR desk, bench or test-bay, not down the hall. With the BASIC Language System, high performance and easy programming are not mutually exclusive – both are provided for your benefit. This increase in your software productivity is applicable to a wide spectrum of task and project demands because this BASIC provides an impressively rich command set at your finger-tips. Sophisticated applications can be met with state-of-the-art hardware power as your resource. At the same time simple tasks can be solved in minutes, not hours.

BASIC 2.0 increases your software *PRODUCTIVITY* by providing:

- A computer system at your fingertips. You don't wait to "log-on" – just sit down and start developing solutions.
- Interactive editing. The editor is always there. A rotary control knob lets you quickly scroll through your program.
- Statement syntaxing at program entry. You don't have to run your program to find syntax errors; BASIC 2.0 proofreads your code.
- Extensive debug and trace tools. You can single-step your program one line at a time to check program logic flow and variable assignments. Tracing tools allow a line-by-line log of line numbers and variable changes to be printed while your software is running.
- Structured programming constructs that organize your code. Complex algorithms are easier to develop, document and maintain.
- Independent subprograms. You can break your software down into smaller modules for easier development and coding. Subprograms can be stored and loaded separately so you can develop disc-resident subprogram utility libraries.
- Dynamic variable allocation. This increases the flexibility of your subprogram libraries so you can efficiently manage memory space as you go.
- Labeled COM, to increase the flexibility of your subprograms. You can have several independent "common" variable blocks, allowing easy subprogram variable usage and subprogram communication.
- A unified device and mass storage I/O system. You can easily redirect I/O to and from devices and files. In essence, your I/O routines can access a mass-storage file in the same way as an external device providing an excellent test harness for your I/O dependent code.

BASIC 2.0 increases your software *PERFORMANCE* by providing:

- 16M bytes of address space including memory-mapped I/O with the Motorola MC68000. As applications grow, you won't outgrow your memory space or system performance. High-volume data acquisition can be done directly to memory.
- Intrinsic I/O drivers. BASIC 2.0 is optimized for I/O so you don't have to write or "link in" separate I/O drivers – they're already there.
- Optimized transactional I/O. I/O path set-up times are reduced to the bare minimum. Short data transfers are just that – short. I/O paths can be turned around quickly, allowing you to go from input to output extremely fast. BASIC 2.0 gives your I/O-dependent software unparalleled "agility."
- The richest HP-IB I/O command set in the industry. Instrument control over HP-IB is quick and to-the-point. More than 20 commands are dedicated to HP-IB I/O to give you performance and flexibility in letting the bus work for you.

- 15 levels of prioritized software interrupt. You can optimize I/O operations to closely match peripheral device speeds without tying down the system. With 15 levels of interrupts, a multitude of external events can be serviced with varying levels of priority set by you. Even the keyboard rotary control knob can be used as a system interrupt source in addition to the soft keys.

These are just a few reasons why the BASIC 2.0 Language System can increase your software performance and productivity. BASIC 2.0 gives you high performance *AND* easy software development, taking you from problem definition to solution quickly and efficiently. You generate results, not just software.

BASIC 2.0 Operating Characteristics

Language System Memory Requirements

ROM-based

1 ROM board AND 21K bytes RAM.

RAM-based

277K bytes RAM.

Range

Real Precision:

– 1.797693134862315E + 308 to – 2.225073858507202E – 308
0
2.225073858507202E – 308 to 1.797693134862315E + 308

Integer Precision:

– 32768 to + 32767

Math Hierarchy

Highest priority: ()

^
User defined and built-in functions

*, /, MOD, DIV

+, – (unary and binary + and –)

=, <>, <, >, <=, >=

NOT

AND

Lowest priority: OR, EXOR

Built-In Functions

Mathematical and trigonometric functions and operations are included in the following with typical execution times in milli-seconds.

| | | |
|------------------------|-------|---------------|
| Absolute (ABS) | 0.041 | integer 0.033 |
| Integer quotient (DIV) | 0.449 | integer 0.075 |
| Digit round (DROUND) | 1.498 | |
| e ^x (EXP) | 3.485 | |
| Integer (INT) | 0.090 | |
| Ln (LOG) | 3.706 | |
| Log (LGT) | 3.800 | |
| Modulus (MOD) | 0.801 | integer 0.074 |
| Random number (RND) | 0.419 | |
| Sign (SGN) | 0.055 | |
| Square root (SQR) | 1.743 | |
| Sine (SIN) | 3.621 | |
| Cosine (COS) | 4.200 | |
| Tangent (TAN) | 4.200 | |
| Arcsine (ASN) | 5.009 | |
| Arccosine (ACS) | 5.004 | |
| Arctangent (ATN) | 3.398 | |
| ^ | 7.155 | |
| / | 0.432 | |

| | |
|-----------------------------|--------------------------|
| * | 0.302 integer 0.080 |
| + | 0.137 integer 0.069 |
| - | 0.161 integer 0.069 |
| Logical Operators | |
| AND | |
| OR | |
| EXOR | |
| NOT | |
| Relational Operators | |
| = | Equal |
| < | Less than |
| > | Greater than |
| <= | Less than or equal to |
| >= | Greater than or equal to |
| <> | Not equal to |
| String Operator | |
| & | Concatenation |

BASIC 2.0 Keyword Summary

General Functions

ABS – returns the absolute value of its argument.
ACS – returns the principal value of the arccosine of its argument.
AND – returns the logical conjunction of its arguments.
ASN – returns the principal value of the arcsine of its argument.
ATN – returns the principal value of the arctangent of its argument.
BINAND – returns the value of a bit-by-bit AND of its arguments.
BINCMP – returns the value of the 1's complement of its argument.
BINEOR – returns the value of a bit-by-bit exclusive OR of its arguments.
BINIOR – returns the value of a bit-by-bit inclusive OR of its arguments.
BIT – returns an INTEGER representation of the contents of one bit of its argument.
COS – returns the cosine of its argument.
DIV – returns the integer portion of the quotient of the dividend divided by the divisor.
DROUND – returns the value of a numeric expression, rounded to the specified number of significant digits.
ERRL – returns a value of 1 if the most recent execution error occurred during the specified line.
ERRN – returns the number of the most recent program execution error.
EXOR – returns the exclusive disjunction of its arguments.
EXP – raises the base of the natural logarithm (Naperian e , = 2.718281828459045...) to the power of the argument.
INT – returns the greatest integer which is less than or equal to the evaluated expression.
KNOBX – returns the net number of pulses generated by the rotary control knob since the last interrupt.
LGT – returns the logarithm (base 10) of its argument.
LOG – returns the natural logarithm (base e) of its argument.
MOD – returns the remainder after performing division of its arguments.
NOT – returns the logical complement of its argument.
NPARG – returns the number of parameters passed in the call to the currently executing subprogram or multi-line function.
OR – returns the logical disjunction of its arguments.
PI – returns an approximate REAL value for pi.
RND – returns a pseudo-random number which is greater than zero and less than one.
ROTATE – returns an INTEGER value representing the value obtained by creating a bit-string version of its argument and rotating the argument the number of bit positions specified.

SGN – returns an INTEGER value of one if the given expression is positive, zero if it equals zero, and -1 if it is negative.
SHIFT – returns an INTEGER value representing the value obtained when its argument is converted to a 16-bit string pattern and shifted the number of positions specified.
SIN – returns the sine of its argument.
SQR – returns the square root of its argument.
TAB – moves the print position to the specified column on the current printing device.
TABXY – moves the print position to the column and line specified on the internal CRT.
TAN – returns the tangent of its argument.
TIMEDATE – returns the current value of the real-time clock.

String Functions

CHR\$ – converts a numeric value into a character byte.
KBD\$ – returns the contents of the keyboard buffer.
LEN – returns an integer representing the current number of characters in a string expression.
NUM – converts the first character of the string expression to its equivalent decimal value ASCII character code.
POS – determines the position of a substring within a string.
VAL – converts a string expression into a numeric value.
VAL\$ – returns a string which represents the value of its argument.



General Statements

ALLOCATE – dynamically reserves memory space for string variables, string arrays and numeric arrays during execution.
ALPHA OFF/ON – enables or disables the CRT alphanumeric area for viewing.
BEEP – outputs an audible tone with programmable frequency and duration.
CALL – transfers program execution to the specified subroutine subprogram and specifies values for the pass parameters.
CASE, CASE ELSE – structured sequence used with SELECT for selecting alternative actions depending on the result of a conditional test (see SELECT).
COM – dimensions and reserves space for simple and array variables in a special "common" memory area so more than one program segment can access the variables.
CONT – resumes program execution without pre-run initialization (see PAUSE).
DATA – allows numeric values or string literals to be defined and assigned with the READ statement.
DEALLOCATE – reclaims the memory space reserved for use by the ALLOCATE statement.
DEF FN – defines a multi-line function which returns either a single REAL value or a string value to a calling program segment.
DEG – sets degree mode for results and arguments of trigonometric functions and graphics rotational commands.
DEL – deletes all program lines within the range specified by the beginning and ending line identifiers.
DELSUB – deletes one or more subprograms or multiple line functions from memory.
DIM – dimensions and reserves memory for REAL numeric arrays, simple strings and string arrays.
DISABLE – disables all active ON (event) statements except ON ERROR, ON END and ON TIMEOUT.
DISP – causes the values of the print list to be sent to the display line on the CRT.
DISP USING – causes the values of the print list to be sent to the display line on the CRT according to the format specified by the image specifier.
DUMP ALPHA – transfers the contents of the CRT screen to the device currently specified by DUMP DEVICE IS.
EDIT – accesses the editor to enter a new program or for modification of program lines.
ELSE – is part of the IF...THEN...ELSE... END IF construct, provides an alternative action to be performed.

ENABLE – re-enables all ON (event) statements which were suspended by DISABLE.

END – marks the end of a main program segment.

END IF – marks the end of an IF...THEN...ELSE...END IF construct.

END LOOP – marks end of LOOP construct.

END SELECT – marks end of SELECT construct.

END WHILE – marks end of WHILE construct.

EXIT IF – provides conditional exit from a structured loop. (see LOOP.)

FN – is used to call a user-defined, multiple-line function subprogram.

FNEND – is the last statement in a multiple-line function subprogram.

FOR...STEP – defines a loop which is repeated until the loop counter exceeds its final value.

GOSUB – transfers program control to the subroutine that begins at the specified line.

GOTO – transfers program execution to the specified line.

IF...THEN, IF...THEN...ELSE...END IF – provides conditional branching or execution of one or more statements when the specified condition is true.

INTEGER – dimensions and reserves memory for integer variables and arrays.

LET – assigns a value to a simple numeric variable or assigns a set of characters to a simple string variable.

LIST – causes the entire program, or lines specified within a range list to be output to the current PRINTER IS device.

LIST # – causes the entire program, or lines specified within a range list to be output to the specified printing device.

LOOP – repeats statements in a structured loop as long as the EXIT IF expression is FALSE.

NEXT – used with FOR in looping (see FOR).

OFF/ON ERROR – disables/enables an event-controlled branch to occur whenever a trappable error occurs.

OFF/ON KBD, OFF/ON KBD ALL – disables/enables interrupt branching when a key is pressed.

OFF/ON KEY [LABEL] – disables/enables an event-controlled branch to occur when an Special Function Key is pressed.

OFF/ON KNOB – disables/enables an event-controlled branch to occur every specified number of seconds if the rotary control knob has generated pulses since the last interrupt.

ON event GOTO/GOSUB/CALL/RECOVER – is used with ON ERROR, ON END, ON KBD, ON KEY, ON INTR, ON KNOB and ON TIMEOUT to cause a branch in program execution based on the specified GOTO, GOSUB, CALL or RECOVER statement. With RECOVER, the branch will occur regardless of the current program environment.

ON expression GOSUB – transfers program execution to one of several subroutines depending on the value of the expression.

ON expression GOTO – transfers program execution to one of several line identifiers depending on the value of the expression.

OPTION BASE – specifies the lower bound of subscript values for all array dimensions when the lower bound is not explicitly stated in the array declaration.

OPTIONAL – is used in formal parameter lists to declare which parameters are optional when passing to a subprogram or FN function.

PAUSE – halts program execution without altering the data or state information so program operation can be continued.

PRINT – prints the specified items in the print list to the current system printer.

PRINT USING – outputs the items in the list according to the format specified by the image specifier.

PRINTALL IS – assigns a destination printing device for output which is normally sent to the display line of the CRT.

PRINTER IS – specifies the current printing device for any PRINT, PRINT USING, CAT and LIST statements.

RANDOMIZE – is used to modify the seed used by the computer's random number generator.

RAD – sets radian mode for results and arguments of trigonometric functions and graphics rotational commands.

READ – assigns values to program variables in conjunction with the DATA statement.

REAL – reserves storage for full precision, floating point variables and arrays.

REM or ! – allows comments to be inserted into your program.

REN – renumbers the program line numbers.

REPEAT...UNTIL – this loop construct repeats the statements in a structured loop until the expression following UNTIL is true.

RESTORE – repositions the data pointer for a program segment.

RESUME INTERACTIVE – enables the EXECUTE, PAUSE, STOP, STEP, CLR I/O, and RESET keys after a SUSPEND INTERACTIVE statement.

RETURN – marks the end of a subroutine.

RETURN expression – the last executed statement of a multi-line function.

RUN – enables program execution at the specified line identifier.

SCRATCH – erases the program in memory.

SCRATCH A – erases all variables including those in COM, programs, and scratchable (normal) binaries.

SCRATCH C – clears all variables including those in COM.

SELECT – this structured statement allows the execution of several different actions depending upon the result of a conditional test when used with CASE or CASE ELSE. SELECT must end with END SELECT.

SET TIME – sets the time-of-day given by the real-time clock.

SET TIME DATE – sets the time-of-day and the date given by the real-time-clock.

STEP – provides an increment size in a FOR...NEXT loop.

STOP – terminates execution of the program.

SUB – is the first statement in a subroutine subprogram and specifies the formal parameter list for the subprogram.

SUBEND – is the last statement of a subroutine subprogram and transfers execution back to the calling program segment.

SUBEXIT – can be used within the body of a subroutine subprogram to transfer execution back to the calling program segment.

SUSPEND INTERACTIVE – disables EXECUTE, PAUSE, STOP, STEP, CLR I/O and (optionally) RESET keys.

TRACE ALL – allows the tracing of program flow and variable assignments while a program is running.

TRACE OFF – disables all tracing activity enabled by TRACE ALL or TRACE PAUSE.

TRACE PAUSE – causes program execution to pause before executing the specified line.

UNTIL – used with REPEAT for looping (see REPEAT).

WAIT – nondestructively suspends program execution for an approximate amount of time in seconds.

WHILE – repeats a structured loop as long as its expression is true.

Mass Storage Statements

ASSIGN @ name – allows the current attributes associated with the @ name to be changed without cancelling it.

ASSIGN @ name TO – links an @name to a device or file, providing a unified I/O structure to a program.

ASSIGN @ name TO * – cancels an @name assignment and its attribute.

CAT, CAT TO – list the contents of a mass storage medium's directory.

COPY – provides the capability to copy single files or to BACKUP entire media.

CREATE ASCII – creates an ASCII type file on the mass storage device.

CREATE BDAT – creates files which hold binary data types.

GET – reads the specified ASCII file into the computer's program memory.

INITIALIZE – sets up a directory for a new medium. Data previously on a disc is destroyed.

LOAD – retrieves and places in memory any program file (PROG) which was previously stored with the STORE statement.

LOAD BIN – retrieves and puts into memory a binary program file (BIN) which was previously stored with the STORE BIN statement.

LOADSUB ALL FROM – loads all of the BASIC subprograms in a program file (PROG) into the computer's memory.

MASS STORAGE IS – specifies the current system mass storage device. (See MSI below.)

MSI – can be used to abbreviate MASS STORAGE IS.

OFF END – deactivates the end-of-statement branch previously activated by an ON END statement.

ON END – enables an event-controlled branch to occur when an end-of-file condition occurs for a mass storage device.

PROTECT – establishes (or changes) the protect code used on non-ASCII files.

PURGE – deletes a file's entry from the mass storage medium's directory.

RE-SAVE – copies all, or part, of the program currently in the computer's memory into an ASCII file as source code strings.

RE-STORE – copies a BASIC program and all binary programs in the computer's memory onto a mass storage medium in internal format.

RE-STORE BIN – writes all binary programs currently in the computer to the specified file.

RENAME – changes the file's name on the mass storage medium's directory.

SAVE – creates an ASCII file and stores all, or part, of the BASIC program in the computer's memory into it as source code strings.

STORE – creates a PROG file and stores the entire BASIC program and all binary programs in computer memory into it in internal format.

STORE BIN – creates a BIN file and stores all of the binary programs currently in the computer's memory into it.

I/O Functions

PPOLL – returns a byte representing the 8 status-bit messages of those devices on the HP-IB capable of responding to a parallel poll.

READIO – allows the reading of either bytes or words from any interface register.

SPOLL – returns an integer whose low order byte contains the serial poll response from the addressed device.

I/O Statements

ABORT – resets the interface functions for an HP-IB interface.

ASSIGN @name – allows the current attributes associated with the @name to be changed without cancelling it.

ASSIGN @name TO – links an @name to a device or file, providing a unified I/O structure to a program.

ASSIGN @name TO * – cancels an @name assignment and its attribute.

CLEAR – allows the active controller to put HP-IB devices into a defined device-dependent state.

CMD – is used with the SEND statement to send numeric or string expressions over HP-IB with ATN true.

CONTROL – sends control information to an interface or to the internal table associated with an @name.

DATA – is used with the SEND statement to send numeric or string expressions over HP-IB with ATN false if the computer is the active controller and is addressed to talk.

DISABLE INTR – sends a word to the interrupt-enable register of the specified interface, disabling all interrupts from that interface.

ENABLE INTR – enables the specified interface to generate an interrupt which can cause end-of-statement branches. Also allows the setup of Powerfail interrupts (see ON INTR).

END – outputs on EOI to an HP-IB device when used in an OUTPUT or SEND list.

ENTER, ENTER @name – is used to read data from a device, file or string and assign the values read to the variables in the list.

ENTER USING – is used to read data from a device, file or string and assign the values read to the variables in the list according to the specified IMAGE.

FORMAT OFF/ON – used in an ASSIGN statement to specify whether data is to be interpreted as ASCII (ON) or Series 200 internal format (OFF).

IMAGE – is referenced by the USING clause of the PRINT, OUTPUT, DISP, LABEL and ENTER statements to provide formats for I/O operations.

INPUT – is used to assign keyboard input to program variables.

LINPUT – assigns characters entered from the keyboard to a string variable or substring.

LISTEN – is used with the SEND statement to specify one or more primary listen addresses.

LOCAL – returns all specified devices to their local state.

LOCAL LOCKOUT – sends the LLO (local lockout) message, preventing an operator from returning the device to local control by its front panel.

MTA – is used with the SEND statement to send the HP-IB interface card's talk address.

MLA – is used with the SEND statement to send the HP-IB interface card's listen address.

OFF/ON INTR – disables/enables an event-controlled branch to occur when an interface card (enabled by ENABLE INTR) requests an interrupt. Priority may also be specified. With the Powerfail option this statement allows branching to an interrupt service routine if power should fail.

OFF/ON TIMEOUT – disables/enables an event-controlled branch to occur when an I/O timeout occurs on the specified interface.

OUTPUT, OUTPUT @name – copies data from the variables in the output list to the specified destination.

OUTPUT USING – copies data from the variables in the output list to the specified destination according to the specified IMAGE.

PPOLL CONFIGURE – programs the logical sense and data bus line on which the specified device responds to a parallel poll.

PPOLL UNCONFIGURE – disables the parallel poll response of the specified devices.

REMOTE – places HP-IB devices having remote/local capabilities into the remote state of operation.

SEC – is used with the SEND statement to send secondary commands and addresses over HP-IB.

SEND – sends control information and data to an HP-IB interface.

STATUS – provides the status value from an interface register, or the internal table associated with an @name, into the specified numeric variables.

TALK – is used with the SEND statement to address a device to talk on the HP-IB.

TRIGGER – initiates device-dependent action from either a selected device or all devices addressed to listen on the HP-IB.

UNL – is used with the SEND command to send the bus unlisten message (UNL) on the HP-IB.

UNT – is used with the SEND command to send the bus untalk message (UNT) on the HP-IB.

USING – used by the PRINT, OUTPUT, DISP, LABEL and ENTER statements, provides formats for I/O operations.

WRITEIO – writes either bytes or words to any interface register.

Graphics Functions

RATIO – returns a value equal to the ratio of the physical dimensions of the graphic device's hard clip limits.

Graphics Statements

AXES – draws a pair of axes with optional equally spaced tick marks.

CLIP – redefines the soft clip area.



CLIP ON/OFF – specifies whether the current clipping area is the soft clip area (ON) or the hard clip area (OFF).

CSIZE – sets the size and aspect ratio for labelled characters.

DRAW – draws a line from the pen's current position to the specified X,Y coordinate position using the current line type and pen number.

DUMP GRAPHICS – copies the contents of graphics memory onto the device currently specified by DUMP DEVICE IS.

DUMP DEVICE IS [EXPANDED] – specifies which device receives the data when DUMP GRAPHICS or DUMP ALPHA is executed. EXPANDED allows a 2 for 1 expansion along each axis, and rotates the resulting image 90°.

FRAME – draws a frame around the current clipping area using the current pen number and line type.

GCLEAR – clears the plotter's background.

GLOAD – allows you to load the contents of an integer array into graphics R/W memory for display.

GINIT – resets all global graphics parameters to their power-on values.

GRAPHICS ON/OFF – turns the internal CRT graphics raster on/off.

GRID – draws a full grid pattern.

GSTORE – allows the copying of graphics R/W memory contents in coded form into an integer array.

IDRAW – draws to a position specified as an increment to the current position of the pen.

IMOVE – moves to a position specified as an increment to the current position of the pen.

IPLOT – incremental plot is similar to IDRAW and IMOVE. The pen control is determined by a value following the X and Y coordinates.

LABEL – directs text and the contents of variables to the current plotting device.

LABEL USING – directs text and the contents of variables to the current plotting device according to the specified IMAGE.

LDIR – determines the angle at which labeling statements draw the characters.

LINE TYPE – selects a line type and repeat length for lines, labels, frames, axes and grids.

LOCATE – syntaxes to VIEWPORT.

LOGR – specifies the position of the characters being labelled relative to the current pen position.

MOVE – updates the current position of the pen to the specified X,Y coordinate position.

PEN – selects the pen used by the plotter. PEN -1 erases lines on the CRT.

PENUP – lifts the physical pen from the plotting surface.

PIVOT – specifies a rotation of axes which is applied to all lines drawn by DRAW and IDRAW statements.

PLOT – moves or draws to the specified X,Y coordinate. Pen control is specified following the X,Y coordinates.

PLOTTER IS – selects a plotter to receive the plotting statements.

RPLOT – relative plot moves or draws to the X,Y position relative to the last absolute pen position. Pen control follows the X,Y values.

SCALE – syntaxes to WINDOW.

SHOW – isotropically defines the plotting units mapped on the VIEWPORT area.

VIEWPORT – specifies an area on which a scale specified by the WINDOW, and SHOW statements, are mapped.

WINDOW – specifies the minimum and maximum values for the plotting area specified by VIEWPORT.

BASIC Extensions 2.1 Capabilities

BASIC Extensions 2.1 includes over 200 keyword additions or extensions to enhance the already powerful BASIC 2.0 Language System to make Series 200 BASIC the most powerful desktop language in the computer industry. The command set for BASIC Extensions 2.1 was designed to contribute to the features already available in other HP desktop language sets. BASIC Extensions 2.1 also adds the ability to make use of the color screen graphics of the Model 36C. These features include advanced I/O such as DMA, fast handshake and interrupt buffer transfers, matrix operations, new interface cards, and mass storage enhancements. Other features were added to support new peripherals including the 7908/11/12 Winchester discs, external color monitors, and Shared Resource Management.

BASIC Extensions 2.1 is provided as six separate binary programs: Advanced Programming, Color Graphics, XREF, and HP-IL/BCD. ROM or RAM-based BASIC 2.0 must be resident in the computer prior to loading any of these binaries. The binary programs can be loaded separately or stacked together to provide optimum memory utilization. This means that if you are only using the matrix statements, you only need the Advanced Programming binary loaded.

The 2.1 versions of BASIC Extensions should be loaded together. 2.0 versions of BASIC Extensions should *not* be loaded with 2.1 versions of BASIC Extensions.

Features of BASIC Extensions 2.1

The features provided by the BASIC Extensions can be divided into these categories:

- Entry and Editing Enhancements
- Debugging Extensions
- Matrix Operations
- String Utilities
- Timer Routines and Event Controls
- I/O Enhancements including DMA and Fast Handshake
- Buffered I/O Capabilities
- Formatting Enhancements
- Mass Storage Enhancements
- 913XB Winchester Disc Support
- Command Set 80 Disc Support
- Callable Pascal or Assembly Language Subroutines (with separate CSUB utility programs)
- 98623A BCD Interface Support
- Shared Resource Manager Support
- Color Video Graphics Extensions for 98627A Interface
- Model 36C Color CRT Support
- 98253A EPROM Development Kit Support
- 98255A EPROM Card Support
- 98625A High-Speed Disc Interface Support
- 98259A 128K byte Bubble Memory Card Support
- RAM Memory Mass Storage Volume Support

Extensions Memory Requirements

| | |
|----------------------------|------------|
| Advanced Programming | 175K bytes |
| Shared Resource Management | 53K bytes |
| Graphics Enhancements | 45K bytes |
| AP on Shared Resource Mgr. | 34K bytes |
| XREF | 7K bytes |
| BCD | 12K bytes |

Advanced Programming Binary Keyword Summary

General Functions

CRT – returns the select code for the internal CRT.
ERRDS – Error Device Status – returns the address of the I/O resource involved in the most recent error.
ERRL – Error Line – extended for TRANSFER.
ERRN – Error Number – returns the number of most recent error; extended for TRANSFER.
FRACT – returns the fractional part of the specified value.
KBD – returns select code of the built-in keyboard.
MAX – returns the largest value in a list of values.
MIN – returns the smallest value in a list of values.
PROUND – the power-of-ten rounding function returns a value rounded to the power of ten specified.
PRT – returns the value 701, the most common select code for a peripheral printer.
SC – returns the select code of an @ name.

String Functions

DATE – accepts a date in the form of DD MMM [-]YYYY and converts it to the number of seconds between that date and the Julian date: 24 Nov – 4713.
DATE\$ – computes date in DD MMM [-]YYYY format from Julian date.
DVAL – returns the whole number value of the string expression in the radix specified.
DVAL\$ – returns the ASCII string containing the specified whole number converted to the radix specified.
ERRM\$ – Error Message – returns the text of the error message of the most recent error.
IVAL – returns the integer value of the string expression from –32768 to 32767 in the radix specified.
IVAL\$ – returns the ASCII string of the integer value converted to the specified radix.
LWC\$ – Lower Case – converts alpha characters to their lower case equivalents in the current lexical order.
REV\$ – returns the specified string in reverse order.
RPT\$ – returns a string containing a string repeated the specified number of times.
SYSTEM\$ – returns system status and configuration information. SYSTEM\$ request items are:

| | |
|-------------------|-----------------|
| AVAILABLE MEMORY | PLOTTER IS |
| CRT ID | PRINTALL IS |
| DUMP DEVICE IS | PRINTER IS |
| GRAPHICS INPUT IS | SERIAL NUMBER |
| KEYBOARD LANGUAGE | SYSTEM ID |
| LEXICAL ORDER IS | SYSTEM PRIORITY |
| MASS MEMORY | TRIG MODE |
| MASS STORAGE IS | VERSION: BASIC |
| MSI | |

TIME – converts a string of the form HH:MM:SS to seconds between midnight and the specified time.
TIME\$ – converts the Julian time to a string of the form HH:MM:SS.
TRIM\$ – returns a string with any leading or trailing blanks removed.
UPC\$ – Upper Case – converts all lower case alpha characters in a string to the corresponding upper case characters in the current lexical order.

Commands

CHANGE – searches the program for the specified string and replaces it with the specified replacement string.
COPYLINES – copies lines of a program from one location to be inserted at another location in a program.
EDIT KEY – allows “edit-key” mode to be entered for defining the special function keys as typing aids.
FIND – searches a program for the specified string.
INDENT – indents your program to reflect the structure based on the statements used.

LOADSUB FROM – provides a library feature by loading all undefined subroutines or functions from a mass storage file.
MOVELINES – moves lines from one location in a program to another.
REN – Renumber – extended to selectively renumber portions of a program.
SCRATCH – erases the program in memory; extended to provide for immediate termination of TRANSFER operations.
SCRATCH A – erases all variables, programs, scratchable binaries and common; extended to provide for immediate termination of TRANSFER operations.
SCRATCH C – erases common area; extended to provide for immediate termination of TRANSFER operations.
SCRATCH KEY – deletes special function key definitions.
XREF – the cross reference command lists the variables, I/O path names line labels and other identifiers used in a program. Provided as a separate binary program.

General Statements

CALL – extended to access Pascal or Assembly generated subprograms (see CSUB also).
COM – extended to allow for BUFFER specification.
CSUB – the compiled sub statement is used to identify a compiled Pascal or Assembly language subprogram.
DELSUB – the delete subprogram statement is extended to allow for deletion of compiled Pascal or Assembly language subprograms.
DIM – extended to allow for BUFFER specification.
INTEGER – extended to allow for BUFFER specification.
LIST KEY – list the definition of the special function keys.
REAL – extended to allow for BUFFER specification.

Event Programming Statements

OFF CYCLE – disables ON CYCLE event branching.
OFF DELAY – disables ON DELAY event branching.
OFF EOR @ – cancels ON EOR @ event branching.
OFF EOT @ – cancels ON EOT @ event branching.
OFF SIGNAL – disables ON SIGNAL software interrupts (see SIGNAL).
OFF TIME – disables event branches set by ON TIME.
ON CYCLE – sets up and enables a periodic event branch each time the specified number of seconds elapses.
ON DELAY – sets up and enables an event branch to occur the specified number of seconds after ON DELAY is executed.
ON EOR @ – sets up and enables an event branch whenever a record terminator is encountered in a TRANSFER operation.
ON EOT @ – sets up and enables an event branch when the last byte is encountered in a TRANSFER operation.
ON SIGNAL – sets up and enables an event branch when a SIGNAL statement is executed.
ON TIME – sets up and enables an event branch when the time specified matches the time in the computers real time clock.
SIGNAL – generates a software interrupt; used with ON SIGNAL.
SYSTEM PRIORITY – sets system priority for event interrupts.

I/O Statements

ABORTIO @ – causes early termination of a transfer to an I/O path.
ASSIGN @ – extended for assigning an I/O path name to a buffer. Also extended to support the following attributes: BYTE or WORD, CONVERT IN or OUT, EOL or EOL OFF, PARITY OFF or ODD or EVEN or ZERO or ONE and RETURN.
BREAK – causes Break sequence to be sent on the RS-232 or Datacomm interface.
BUFFER – used in the DIM, COM, INTEGER, REAL and ASSIGN @ statements to define the buffer area to hold data for the TRANSFER statement.
BYTE – defines byte (8-bit) type transfer for an interface as opposed to word type transfer (used with ASSIGN @).
CONT – used in the TRANSFER statement to indicate that the transfer is to continue until an end-of-transfer condition is encountered.

CONTROL – extended for new device registers and control operations on a buffer.

CONVERT IN or OUT – used in the ASSIGN @ statement to define conversion tables for ENTER (IN) or OUTPUT (OUT). The conversion table can be set up to convert by INDEX into a table or by PAIRS of characters in the table: the original character and its replacement.

CONVERT IN/OUT OFF – causes code conversion set up by CONVERT IN/OUT to be discontinued.

COUNT – used in the TRANSFER statement to terminate a transfer when the specified number of bytes have been transferred.

DELIM – used with the TRANSFER statement to assign a delimiter for terminating an inbound transfer.

END – used with the TRANSFER statement to specify an interface-dependent message for terminating an inbound transfer (such as EOI for HP-IB).

ENTER – extended to allow entering data from a buffer.

EOL – defines End-Of-Line sequence to be sent with PRINT and OUTPUT.

EOL ... [END, DELAY] – allows terminating I/O attributes when used with the ASSIGN statement.

EOL OFF – Resets EOL to Carriage Return/Line Feed.

EOR – used in the TRANSFER statement to specify the End-of-Record delimiter.

IMAGE – extended to include several new format specifiers.

LEXICAL ORDER IS – defines the collating order used by all string relational operators and the UPC\$ and LWC\$ functions. Six language tables are preset: ASCII, STANDARD, FRENCH, GERMAN, SPANISH, and SWEDISH.

OUTPUT – extended for output to buffers.

PARITY – used with ASSIGN @ to specify parity type.

PASS CONTROL – used to pass active control on the HP-IB to another controller on the bus.

PPOLL RESPONSE – stores the specified value in the parallel poll response register of the HP-IB interface.

PRINT – extended to support WIDTH and EOL sequence.

PRINTALL IS – extended to support WIDTH and EOL sequence.

PRINTER IS – extended to support WIDTH and EOL sequence.

RECORDS – used in the TRANSFER statement to specify the number of records to be transferred.

REQUEST – used to request service from the active controller on HP-IB.

RESET – resets an interface, file or buffer.

RETURN – when used with the ASSIGN @ statement, returns a value indicating the outcome of an assign statement.

STATUS – extended to return status information of buffers and additional device registers.

TRANSFER – provides for unformatted DMA, interrupt and fast handshake buffer transfers.

WAIT – used in the TRANSFER statement to cause the transfer to execute in serial mode.

WAIT FOR EOR @ – suspends overlapped program execution until a record boundary is reached in a transfer operation.

WAIT FOR EOT @ – suspends overlapped program execution until the completion of a transfer operation.

WIDTH – used in the PRINTER IS and PRINTALL IS to specify the printer width.

WIDTH OFF – used in the ASSIGN @, PRINTER IS, and PRINTALL IS statements to set the printer width to infinite.

WORD – used with ASSIGN @ to define word (16-bit) type of interface transfer.

Mass Storage Statements

CAT – extended to catalog the binary programs and subprograms needed by a PROG type file and the binaries in a BIN type file. New control options have also been added (see COUNT, NO HEADER, SELECT and SKIP).

CAT TO – extended to catalog a mass storage directory to a string array.

CHECKREAD OFF – disables verification of data written to a mass storage file.

CHECKREAD ON – enables verification of data written to a mass storage file.

CONT – see I/O statements. Can also be used on files.

COUNT – used in the CAT statement to return the number of items listed.

DELIM – See I/O statements. Can also be used on files.

END – See I/O statements. Can also be used on files.

EOR – See I/O statements. Can also be used on files.

LOAD KEY – loads the definitions of the special function keys from the specified mass storage file.

LOADSUB subprogram FROM – used to load a single subprogram by specifying the subprogram name.

MASS STORAGE IS – extended to include support for 7908, 7911 and 7912 discs, the HP9885M/S, HP913X, BUBBLE, EPROM, and MEMORY.

NO HEADER – used with the CAT statement to disable header information in a catalog listing.

RECORDS – See I/O statements.

RE-STORE KEY – stores special function keys into a previously used file name.

SELECT – used with the CAT statement to select entries that begin with the specified characters.

SKIP – used with the CAT statement to skip the specified number of items before listing catalog entries.

STORE KEY – stores the special function key definitions onto the specified file.

WIDTH – see I/O statements. Can also be used on files.

Matrix Functions

BASE – returns the lower bound for the specified dimension of an array.

DET – returns the determinant of the specified matrix or the last matrix inverted.

DOT – returns the inner (dot) product of the two specified vectors.

RANK – returns the number of dimensions in an array.

SIZE – returns the number of elements in the specified dimension of an array.

SUM – returns the sum of all elements in an array.

Matrix Statements

MAT +, -, *, /, <, <=, =, <>, >=, > – perform the specified array math, element by element, on the specified numeric arrays (. is for element by element multiply.)

MAT * – matrix multiplication.

MAT ... = – array assignment or initialization.

MAT REORDER – used to rearrange the data in an array according to an associated pointer vector.

MAT SORT – used to reorder an array or set up a reorder vector according to specified keys in either ascending or descending order.

MAT CSUM – computes a column sum on the specified matrix.

MAT IDN – sets a square matrix to the identity matrix: the main diagonal becomes all 1's, all other elements become 0's.

MAT INV – produces the inverse of the specified matrix.

MAT RSUM – computes the row sum on the specified matrix.

MAT TRN – computes the transpose of the specified matrix.

REDIM – Redimension – changes the subscript bounds of a list of arrays.

Graphics Binary Keyword Summary

Graphics Statements

- AREA COLOR – specifies a fill color using the HSL color model. Used with statements requiring area fill.
- AREA INTENSITY – specifies the fill color using the RGB color model. Used with statements requiring area fill.
- AREA PEN – specifies a pen number to use as the fill color. This is used with statements requiring area fill.
- CONTROL – extended to allow control of alpha color.*
- DIGITIZE – determines the X and Y coordinates of the current graphics input device's locator when the digitize button (or stylus) is pressed.
- DISP – extended to support color highlights.*
- DUMP GRAPHICS – extended to allow specification of the source device.
- EDGE – used in the IPLOT, PLOT, POLYGON, RECTANGLE, RPLOT and SYMBOL statements to specify that the lines making up the edges of an area are to be drawn using the current line type and pen.
- FILL – used in the IPLOT, PLOT, POLYGON, RECTANGLE, RPLOT and SYMBOL statements to specify that the figure drawn is to be filled with the current fill color.
- GESCAPE – is used for making a request to a graphics device driver. This is a device and driver dependent command.
- GLOAD – extended to allow specification of the destination device and loading of a subset of the graphics memory.
- GRAPHICS INPUT IS – defines the device for graphics input.
- GSTORE – extended to allow specification of the source device and storing of a subset of the graphics memory.
- IPLOT – extended to allow incremental plotting by specifying a sequence of commands in an array.
- PEN – specifies the pen or line color to be used. Allows selection of colors on the CRT of the Model 36C.
- PLOT – extended to allow a sequence of plot commands in an array.
- PLOTTER IS [j COLOR MAP] – initializes a plotting device and enables it for subsequent output. Color map extension to allow graphics output to the Model 36C or external color displays.
- POLYGON – outputs a closed regular polygon or part of a regular polygon.
- POLYLINE – outputs an unfilled regular polygon or part of a regular polygon.
- PRINT – extended to support color highlights.*
- READ LOCATOR – determines the X and Y coordinates of the current graphics input device's locator without requiring the digitize button (or stylus) to be pressed.
- RECTANGLE – draws a rectangle with dimensions specified as displacements from the current position.
- RPLOT – extended to allow relative plotting by specifying a sequence of commands in an array.
- SET ECHO – sets the current graphics output device's locator position to the specified position.
- SET PEN ... INTENSITY – defines one or more entries in the color map using the RGB color model.
- SET PEN ... COLOR – defines one or more entries in the color map using the HSL color model.
- SYMBOL – allows labelling with user-defined symbols.
- TRACK ... IS ON/OFF – enables the graphics input device's locator to be tracked by the current graphics output device's cursor (or pen) position.
- WHERE – returns the current pen position into X and Y variables. It also can return information on whether the pen is up, down, inside or outside the viewport and hard clip boundaries.

* Model 36C only.

Shared Resource Management Binary Keyword Summary

The following BASIC 2.0 Mass Storage Statements are expanded to encompass the use of Shared Resource Management mass storage files:

| | | |
|--------------|-------------------|--------------|
| ASSIGN | LOAD BIN | PURGE |
| CAT | LOAD SUB | RENAME |
| CONTROL | LOAD SUB ALL FROM | RE-SAVE |
| COPY | MASS STORAGE IS | RE-STORE |
| CREATE ASCII | PROTECT DELETE | RE-STORE BIN |
| CREATE BDAT | PROTECT MANAGER | SAVE |
| GET | PROTECT READ | STORE |
| LOAD | PROTECT WRITE | STORE BIN |

The following BASIC keywords are added for use with Shared Resource Management mass storage files.

CREATE DIR – creates a directory file.

LOCK ... CONDITIONAL – prevents other Shared Resource workstations from accessing a shared file which is currently in use.

UNLOCK – removes exclusive access to a file which was protected by the LOCK statement.

Shared Resource Management Advanced Programming Binary Keyword Summary

The following BASIC Extensions 2.1 Mass Storage Statements are expanded to encompass the use of Shared Resource Management mass storage files.

CAT – with new options ALL, PROTECT.

CAT PROG

CAT TO

LOAD KEY

LOADSUB FROM – command only.

LOADSUB ... FROM

RE-STORE KEY

STORE KEY

TRANSFER

The Advanced Programming capabilities on SRM require that the Advanced Programming and the Shared Resource Management Binary programs are loaded in before the AP SRM binary.

XREF Binary

Adds the XREF keyword (described in Advanced Programming keyword summary). This was included as a keyword in AP2.0. It can now be used standalone with BASIC 2.0 or loaded in with the other BASIC Extensions.

BCD Binary

Contains the drivers for the BCD (98623A) Interface card. BCD card drivers provide BASIC 2.0 programming capabilities. The BCD card drivers cannot take advantage of the advanced programming capabilities. The BCD Binary can be used standalone with BASIC 2.0 or loaded in with the other BASIC Extensions.



Series 200 BASIC Interfacing Capabilities

In addition to the built-in HP-IB interface,[†] there is a choice of external interface cards and internal memory enhancements:

- 2-channel DMA Controller (98620A)*
- GPIO (98622A)
- BCD (98623A)*
- HP-IB (98624A)
- High-Speed Disc (98625A)*
- Serial (98626A)
- Color Video (98627A)*
- Datacomm (98628A)
- Shared Resource Management (98629A)*
- HP-IL (98634A)*
- EPROM Development (98253A)*
- EPROM (98255A)*
- 128K byte Bubble Memory (98259A)*
- Programmable Datacomm (98691A)

* Requires BASIC Extensions 2.1.

† Serial interface is also built into the Model 16.

DMA Controller Card

The 98620A DMA Controller Card enhances the Series 200's interfacing capability by providing two DMA channels for I/O data transfers. This high-speed I/O capability works with the 98622A GPIO, 98624A HP-IB, and internal HP-IB interfaces. The 98620A requires the Advanced Programming Binary in BASIC Extensions 2.1.

GPIO Interface

The 98622A GPIO interface provides 16 bits of latched input and output data for bidirectional transfer of information. Extended control and status lines are available for applications that require more than one signal from the computer. Several handshake modes are also available to permit interfacing to a variety of equipment.

BCD Interface

The 98623A BCD interface connects the Series 200 computer with bit-parallel, digit-parallel, binary-coded decimal devices for data input. Up to eight significant BCD digits, two sign bits (mantissa and exponent), exponent digit, function code digit, and an overload bit can be read. Input format is selectable, allowing two independent instruments to be read from one 98623A Interface Card. Data can also be accepted as five input bytes of pure binary information. Eight data output lines are also provided for use as general purpose control and/or data output. The 98623A requires the BCD binary in BASIC Extensions 2.1.

HP-IB Interface

In addition to the standard built-in HP-IB interface, there is an optional external 98624A HP-IB Interface. Both interfaces implement the IEEE 488-1980 Standard Digital Interface for Programmable Instrumentation. Both interfaces can communicate with as many as 14 HP-IB compatible instruments, connected with a maximum of 20 metres (65.6 ft.) of cable.

Disc Interface

The 98625A High-Speed Disc Interface provides an interface to the Command Set 80 discs (7908/11/12/14) that offer the maximum transfer rate available. The 98625A, when used with the 98620A/B DMA card allows buffered DMA data transfers between the Series 200/210 computer and the Command Set 80 disc. The 98625A allows up to four discs to be connected to one interface card. It is recommended that a 98620A/B DMA card be included in any machine using a 98625A as little performance improvement is seen over a 98624A unless a DMA card is installed.

Serial Interface

The 98626A Serial Interface provides bit-serial communication between the Series 200 computer and asynchronous EIA RS-232-C (CCITT V.28/V.24) devices. Data rates range from 50 to 19200 baud (bits/sec.). A variety of cabling options allow for current loop, modem and terminal connections. Series 200 Terminal Emulator Software takes advantage of this card for communication to other computers. The Serial Interface is built into the Model 16 computer.

Color Video Interface

The 98627A Color Video Interface provides the interconnection to an external color monitor. This interface connects to a high-performance, high-resolution color monitor via three outputs – Red, Green/synch, and Blue (RGB). The capabilities provided by this interface make it appear as a “soft plotter.” The 98627A requires the Graphics binary in BASIC Extensions 2.1.

Data Communications Interface

The 98628A Data Communications Interface provides both protocol management and electrical levels for asynchronous serial communications. This card also supports the Distributed System Network/Data Link (DSN/DL) protocol for communications to an HP 1000 series minicomputer. A terminal emulation program, which takes advantage of this card for communication to other computers, is also available.

Shared Resource Management Interface

The 98629A Shared Resource Management provides both protocol management and electrical levels for communications on the Shared Resource Management (SRM) System. The SRM system allows Series 200 computers as well as the 9835, 9845 and Series 500 computers to share common disc and printer resources. The 98629A Interface is used by Series 200 computers to connect to the SRM Controller so that files and printer resources can be shared between computers on the system. The 98629A requires the Shared Resource Management Binary in BASIC Extensions 2.1.

Breadboard Card

The 98630A Breadboard Interface allows experienced hardware designers to design their own custom interface to Series 200 Computers when none of the interface cards provided by HP will fit a particular requirement. The interface consists of a printed circuit board with the necessary buffering to properly interface to the Series 200 backplane. Most of the space on the board contains tinned holes on standard centers to allow a prototyping area for custom interface design.

EPROM Development Kit

The 98253A EPROM Development Kit consists of one EPROM Programmer Card and a 98255A EPROM Card. The EPROM Development Kit allows you to create your own memory volumes in EPROM. The EPROM Programmer Card allows you to program EPROMs for the EPROM Card using simple commands and the Series 200 Computer. The 98253A requires the Advanced Programming Binary in BASIC Extensions 2.1. The Model 16 does not support the 98253A EPROM Development Kit.

EPROM Card

The 98255A EPROM Card contains 16 sockets for EPROMs to allow up to 256K bytes of storage using Intel 27128 EPROMs or equivalent. The 98255A allows program or data files to be stored permanently for maximum reliability, durability, and program security. The 98255A is ideal for repetitive execute-only applications. The 98255A requires the Advanced Programming Binary in BASIC Extensions 2.1.

128K Byte Bubble Memory Card

The 98259A Magnetic Bubble Memory Card features 128K bytes of non-volatile mass storage. The 98259A provides increased reliability and durability over flexible disc storage. The 98259A is ideal for remote data acquisition applications where non-volatile storage is necessary. The 98259A requires the Advanced Programming Binary in BASIC Extensions 2.1.

Programmable Datacomm Interface

The Programmable Datacomm Interface provides a spectrum of capabilities that can be tailored to meet your special datacomm and/or serial interfacing needs. The programmable Datacomm interface consists of two pieces – the Development Package (98690A) and the interface card (98691A).

The 98690A Development Package contains the essential information and tools required by the sophisticated user to do firmware programming of the 98691A Programmable Datacomm Interface. The 98691A accepts the personality ROM developed with the 98690A Development Package.

BASIC Language Configurations

Four separate products comprise the BASIC language offerings for Series 200 products:

98601A ROM-based BASIC 2.0 Language System

Contains BASIC 2.0 on a single plug-in ROM board with utility programs on flexible discs.

98602A ROM-based BASIC 2.0 Language System With 2.1 Extensions

Contains BASIC 2.0 plus BASIC Extensions 2.1 on a single plug-in ROM board. The following binary programs are not contained in ROM, but are provided on a flexible disc shipped with the 98602A product:

- AP on Shared Resource Manager
- XREF
- BCD

All BASIC 2.0 plus Extensions 2.1 utilities are provided on flexible discs.

98611A RAM-based BASIC 2.0 Language System

Contains BASIC 2.0 plus utility programs on flexible discs.

98612A RAM-based BASIC Extensions 2.1

Contains BASIC Extensions 2.1 with utility programs on flexible discs. Requires BASIC 2.0 (ROM- or RAM-based).

Utilities

Compiled Subprogram Utilities P/N 09800-10X40*

Converts modules compiled (or assembled) in Pascal (or assembly language) for use as BASIC subprograms. Compiled subprograms contain the following capabilities and restrictions:

Capabilities

- Ability to call subprograms written in Pascal or assembly from BASIC.
- Parameter passing from BASIC to the subprogram.
- Code is linkable with subprogram libraries. There is no code size restriction and code is relocatable.
- Compiled subprograms have access to BASIC common space.
- Pascal global variable space is allowed so Pascal global variables can be used.
- A fixed amount of HEAP space can be specified by the user.
- Simple READ(LN) and WRITE(LN) to the PRINTER IS device can be done.

Restrictions

- Pascal device, file and graphics I/O are not supported.
- Not all Series 200 Pascal language capabilities are supported. Procedure variables and interrupt service routines are not available. The GOTO statement in Pascal cannot be used to jump to another procedure.

Pascal 2.0, BASIC 2.0 with Extensions 2.0 (or later systems) are required.

BASIC Utilities Library P/N 09800-10X00*

Provides media and system management utilities and programming aids. Also, mainframe feature and application demonstration programs for the BASIC language system. (Shipped with every BASIC 2.0 operating system.)

Loader Utility P/N 09800-10X10*

Allows loading of BASIC 2.0 system and binary programs with one operation. Uses disc drivers in the 3.0 Boot ROM to load other drivers from an external disc or SRM. (Shipped with every BASIC 2.0 operating system.)

9835/45 to Series 200 BASIC Language Translator P/N 09800-10X90*

Provides aid in converting programs from HP 9835/45 BASIC to Series 200 BASIC 2.0. The pack also translates 8-inch flexible disc data files created on HP 9835/45 to LIF data files. Requires BASIC 2.0 operating system. (Translator does not use the BASIC Extensions capabilities.)

* The "X" shown in the part numbers specifies the media on which the software is supplied. It can have the following values:

- 3 for use in 3½-in. external flexible disc drives
- 5 for use in 5¼-in. external flexible disc drives
- 6 for use in 5¼-in. internal flexible disc drives



