HP-97 HP-97S Calculator Emulators

Project Design by Tony Nixon tnixon059@gmail.com
http://www.teenix.org

Best viewed in 1920 x 1080 screen resolution. (Minimum is 1280 x 720)

Disclaimer

The material contained within this package is supplied without representation or warranty of any kind. The author therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this material or any part thereof.

HP-97 Microcode Bug

There is a bug in the HP97 microcode that will cause the calculator to crash.

If you power up the calculator and press [Sigma –] (Shift and Sigma +), then press [X bar] (Shift and X^2), then the Y register will contain negative zero. (See the debug screen, or print the stack) If you now press [% Change] (Shift and %), the calculator will treat the Y register as non zero, but will actually do a division by zero which results in a crash. Turn [Off] then [On] or use the [Reset] debug button to restore
.

Calculator (HP-97S)

Menu

The menu is activated by right clicking on the calculator

Menu Items

Calculator Type	Select the type of calculator to emulate
For The Inquisitive...	Opens the debug screen
Copy Display	Copies the display value to the PC clipboard
Options	
Turbo Mode	Runs the emulator at the fasted speed it can
Clear Stack Error	If the calculator subroutine stack pointer under or overflows the stack address indicator will change colour. Click this item to restore the non error colour.
Low Battery Display	Shows the calculator display in low power mode
Calculator Key Legend	Toggles the PC key legend which displays on the bottom of the calculator face when the mouse is over a button.
Sound	Toggles card read/write motor or printing sounds playing
Auto RAM Bank Switch	Toggles automatic RAM bank switching during Slow or Step modes
Fine Tune Timer	Tweak the timing interval for the calculator code execution and HP01 clock timing PC sync – see HP01 section.
97 Extra Memory	Use RAM at address $3A, $3B and $3C for useable memory. Uses modified HP-97 microcode. Access via [f] [P<>S].
Printer	
View Roll	View the entire printer roll
Print Roll	Print the used part of the print roll
Discard Used Roll	Clear the used part of the print roll
New Roll	Replace the printer roll with a new one The print roll buffer holds 500 printed lines
Print Head Speed	Adjusts the speed of the print head movement
Help	Open this help file
Program	Load/Clear HP-97programs and view user card image
Program Trace Log	View a trace of the current calculator program
Program Notes	See Program Notes
Program Capture	See Capture
Minimize	Minimize the window
Close	Closes the program (F2)

Switches

All switches can be activated by clicking on the labels adjacent to the switch.
The mouse cursor will change to a hand pointer

Moving The Calculator

Move the mouse over the calculator display
The mouse will change to a hand
Left click to “grab” the calculator and drag to a new position

Assembler Pad Focus

As the PC keyboard is shared by the calculator keyboard and the assembler pad, either of these items must have focus when required. To accomplish this, when the mouse has moved over the calculator, the PC will recognise the calculator keyboard. When the mouse has moved over the assembler pad, the PC will direct key input to the pad.

Key Short Cuts

The PC keyboard short cuts for the calculator buttons are shown if the [Calculator Key Legend] menu item is enabled.

Additional:

F2	Close the program
F5	Assembler - Auto step code at slow speed
F7	Assembler - Single step code
F9	Assembler - Run code normally
F12	Toggles calculator/debugger screens

Simulator

 Run the code at normal speed

 Run the code at slow speed (Adjust with the [Adj Slow Speed] control)

 Step the code one instruction at a time

 Execute code from assembler listing cursor position

 Skip over all break points

 Clear all simulator break points

 Reset the calculator

 Reload the original calculator micro-code file

 Open the Program Card Editor

Toggle Break Points

Break points can be toggled on/off by double clicking on any valid code address in the assembler left gutter while in Single Step Mode. Break points will be highlighted in red.

Register Modify

The calculator registers can be modified by placing the mouse over the register and clicking the left mouse button.

A small window will open allowing the value to be modified.

Note: Changing some values might alter the operation of the calculator and some changes might not occur. For example HP-45, change regS bit 0 to 1 when calculator buttons are not pressed will remain 0.

RAM Address Pointer

You can set the storage RAM address pointer by left clicking on a RAM address number.

RAM Value Modify

The RAM register values can be modified by clicking over the register. An input window will open so that you can change the value. You can modify the register directly or any numeric value that will fit into the register will be accepted.

Examples:
10
10.99999
100 E9
-90012 E-09
4532.0009 E10

Assembler Code Line Format

```
Label          Code              Comment
Label1:        c -> data         // comments
```

Labels must be followed by the colon (:) character.
Labels must be unique in each ROM page of 256 words.
Text that follows the [//] characters on a code line is considered to be a comment.

ORG directive

This sets the program counter to the value shown which saves adding [no operation] to fill between code lines.

```
Eg Org $023      Org $789
```

The code editor has tab indenting for three listing items.

1) Code following labels with up to 6 characters
2) then go to...
3) Comment fields.

Example:

```
Label1:     if s0 # 0                      // comment
                then go to Label1 // comment
```

HP-97 If...Go To

When a [go to] instruction follows an [if...] type of instruction in the HP-67, the [go to] instruction becomes a 10 bit address after assembly.

Example:

```
            org $000

            if s0 # 0               // $000
               then go to Addr1     // $001
            ....
            ....
            org $110

Addr1:      no operation            // $110
```

Normally the [then go to Addr1] instruction will be compiled as $210, but because the code is for the HP-67 and the [go to] follows an [if...] instruction, the compiled code will be $110 and is the actual code address of Addr1.

Note: If your code includes a [go to] or [jsb] which puts the Program Counter on a [go to] instruction which is an address only, a run time error will occur.

Example: (from above code)

```
                jsb $001          // causes run time error
```

Classic Calculators CPU and Programming

56 Bit Registers

The HP calculator CPU's were optimized for floating point numbers. The main registers contain 14 x 4 bit nibbles (56 bits per register). This allowed each register to hold a 10 digit mantissa, a 2 digit exponent and a sign each for the mantissa and exponent in BCD format.

The 56 bit registers were:

A, B General purpose registers for math and scratchpad use

C Similar to A and B but also dedicated to memory reads and writes and handled transfers to
M the C register also contained the value to be displayed in the X register

D, E, F These registers hold the user stack levels Y, Z and T and are accessed by user rotating the stack from the keyboard or by using stack related commands

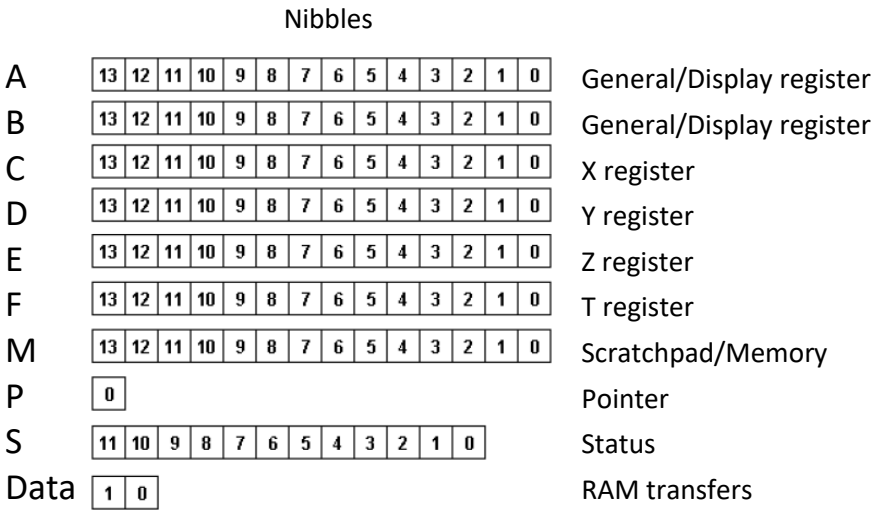
M Used as a scratchpad or memory register and only supported transfers to and from the C register

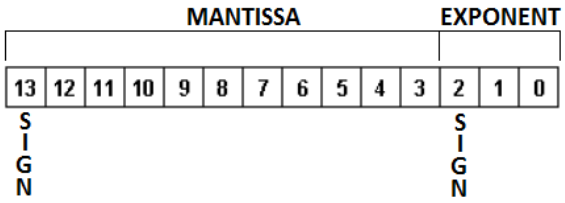
Additional Registers

P Is a 4 bit offset into register nibbles used with the P and WP field select instructions.

Data Transfers data to RAM memory via the C register

Status Holds 12 programmable bits including some that are controlled by hardware





Instructions Listing

```
TYPE 1 - ADDRESS CHANGE

              jsb nn [label]          Jump to Sub Routine
              go to nn [label]        go to  (Carry bit must be clear to work)
                then go to            nn = address or [label] in same ROM page
                if no carry go to

TYPE 2 - ARITHMETIC

  Clear       0 -> a[fs]              Clear a
              0 -> b[fs]              Clear b
              0 -> c[fs]              Clear c

  Transfer/   a -> b[fs]              Transfer a into b
  Exchange    b -> c[fs]              Transfer b into c
             c -> a[fs]              Transfer c into a
              a exchange b[fs]        Exchange a and b
              b exchange c[fs]        Exchange b and c
              a exchange c[fs]        Exchange a and c

  Add/        a + c -> c[fs]          Data Select [fs]
  Subtract    a + b -> a[fs]                      [p]   -  determined by P register
              a + c -> a[fs]                      [m]   -  mantissa
              c + c -> a[fs]                      [x]   -  exponent
              a - c -> c[fs]                      [w]   -  word (entire register)
              a - b -> a[fs]                      [wp]  -  word up to and including P register
              a - c -> a[fs]                                  Eg P=3, nibbles 0,1,2,3
                                                  [ms]  -  mantissa and sign
  Compare     if b[fs] = 0                        [xs]  -  exponent and sign
              if c[fs] = 0                        [s]   -  mantissa sign
              if a[fs] >= 1
              if c[fs] >= 1
              if a >= b[fs]
              if a >= c[fs]           if.... just sets/clears carry bit

  Complement  0 - c -> c[fs]          Tens Complement   0-C->C
              0 - c - 1 -> c[fs]      Nines Complement  0-C-1->C

  Increment   a + 1 -> a[fs]
              c + 1 -> c[fs]

  Decrement   a - 1 -> a[fs]
              c - 1 -> c[fs]

  Shift       shift right a[fs]
              shift right b[fs]
              shift right c[fs]
              shift left a[fs]

TYPE 3 - STATUS OPERATIONS

              1 -> s(Stat Bit)        HP35, HP45  - Stat Bit = 0 - 11
              0 -> s(Stat Bit)        HP55        - Stat Bit = 0 - 12
              clear status
              if s(Stat Bit) # 1      (not equal to 1) - clears carry bit
              if s(Stat Bit) = 0      (equals 0) - same as above

TYPE 4 - POINTER OPERATIONS

              nn -> p                 p = nn  (nn = 0 - 15)  p - 1 -> p
              p + 1 -> p              Increment pointer
              p - 1 -> p              Decrement pointer
              if p # nn               If P not equal to nn - clears carry bit
              if p = nn               If P equal to nn - clears carry bit

TYPE 5 - DATA ENTRY / DISPLAY

              load constant nn        nn -> c[p]  (n = 0 - 15)  p - 1 -> p
              display toggle          Toggle display on/off
              i8 -> a                 Not implemented - no operation
              swap c -> data address  c[11] = c[12], c[12] = 1
                                      (created to fix HP55 listing bug)
              c exchange m            Swap c and m
              m1 exchange c           Swap m1 and c
              m2 exchange c           Swap m2 and c
              c -> stack              Push c onto stack  C->C->D->E->F
```

```
              stack -> A              Pop stack into a   F->F->E->D->A
              display off             Turn off display
              display on              Only for HP-45 in timer paused loop
              m -> c                  m register into c register
              m1 -> c                 m1 register into c register
              m2 -> c                 m2 register into c register
              down rotate             Rotate stack down  C->F->E->D->C
              clear registers         Clears registers A B C D E F M
              clear data registers    Clear bank of RAM registers
              data -> c               RAM memory into c - set RAM address first
              c -> data register n    register c into RAM (current bank + n)
              data register n -> c    RAM (current bank + n) into c register
              f exchange a            swap f register and a register[0]
              f -> a                  a register[0] = f register

TYPE 6 - ROM SELECT, MISC

              select rom n            Set program counter to ROM Page n
              delayed select rom n    As above but delayed until next branch
              delayed select group n  Select a group of 8 roms, delayed as above
              return                  Subroutine return
              keys -> rom address     Key code to program counter
              keys -> a               Key code into a register
              a -> rom address        Set program counter from data in a register
              y -> a                  y register into a register
              c -> data address       Set RAM address pointer
                                      HP45   c[12]
                                      HP55   c[12] x 10 + c[11]
              c -> data               c into RAM memory - set RAM address first

TYPE 10

             no operation             Does nothing

OTHER

              binary                  Set binay math mode   - base 16
              decimal                 Set decimal math mode - base 10
              bank switch             Toggle a bank of 16 roms
              display reset kmf       Not implemented = no operation
              hi i'm woodstock        Not implemented = no operation

HP-97         Card reader circuit commands

              CRC  060  Set display digits
              CRC  100  Test ready
              CRC  160  Test display digits
              CRC  260  Motor = On
              CRC  300  Test W/PGM switch
              CRC  360  Motor = Off
              CRC  400  A key was pressed
              CRC  500  Test of a key was pressed
              CRC  560  Test if card inserted
              CRC  660  Set card write mode
              CRC  760  Set card read mode
              CRC 1000  Set default function keys
              CRC 1100  Test if default function keys set
              CRC 1200  Set merge flag
              CRC 1300  Test merge flag
              CRC 1400  Set waiting for card side 2 flag
              CRC 1500  Test waiting for card side 2 flag
              CRC 1700  Read/Write data to/from card via RAM $99 and $9B

             CRC TestF1       Program / Run
             CRC TestF2       Printer Manual
             CRC TestF3       Printer Norm
             CRC TestF4       KBE line for SST BST keys
```

Break Points

Break points can be set by double clicking on a valid code address in the code editor greyed left gutter. These addresses will be highlighted in red for active break points and a break point identifier will appear as a comment in the code line.

Break points can also be set/cleared by adding/deleting a break point identifier in the comment field

```
Break Point Identifier:    /b     (case sensitive)
```

Example: Without a comment

```
$123   Label1       no operation
$123   Label1       no operation          /b
```

Example: With a comment

```
$123   Label1:      no operation          // no break point
$123   Label1:      no operation          /b break point
```

Go To / JSB Instruction Format

```
go to @203    OCTAL
go to $FE     HEX
go to .100    DECIMAL
go to Label   LABEL

jsb @207
jsb $0E
jsb .23
jsb Label
```

Note: All code line addresses are displayed in the gutter as hexadecimal.

Target Label Names

Limited to 6 characters and must be followed by a colon, not case sensitive.
Due to the design of the [Classic] calculator series, labels must be unique within any 256 word ROM page boundary.

```
Example:    Incval:     // appears in ROM $000 - $0FF
            Incval:     // also appears in ROM $000 - $0FF    -> Error!!
            Incval:     // appears in ROM $100 - $1FF   -> Ok
```

Jump To Label

Double click a label to jump to the matching target label in the current ROM page.
Double click a target label to jump to the first matching label in the current ROM page.

Trace

The trace screen shows the previously executed code lines up to a maximum count of 4000. After a count of 4000 is reached the first trace line will disappear as new lines are added.

Buttons


New Trace — Starts a new trace


Copy To Clipboard — Copies the trace buffer to the PC clipboard


Freeze — Disables the trace function

A new trace begins each time the Run or Slow buttons are clicked.
Note: When printing multiple lines, the HP-97 microcode has a code loop that waits until the carriage has homed before printing the next line. This can fill up the trace buffer many times over so the trace will log the first time these instructions execute – (ROM addresses $038D to $038F, and $039D to $039F), then ignore following execution.

Data Capture Editor

This screen allows you to create capture items. It is a simple text interface and allows up to 20 items to be entered. If too many items were added, then the extra processing will begin to slow the simulator down.

Buttons


Clear all items


Open capture file


Save capture file

Format `ROM Address {space} [Item #1] / [Item #2] / ... [Item #X]/`

The capture event occurs when the ROM address is executed in the calculator microcode.

Items

Items describe the text output for each capture event and have the following format.

```
[Capture Type - RAM Address - Display Type] {space} [Caption]/
```

`Capture Type`, `RAM Address` and `Display Type` are represented by up to 4 characters if required. Depending on the Capture Register selection, some of these may not be required, otherwise they must be entered in the order specified. Mandatory spaces are added after the ROM address and these characters to improve readability.

Multiple items can be entered for each capture event, but each item must be separated by a forward slash character [/]. This may be included or omitted on the last item.

The **ROM Address** is a 4 digit hexadecimal number. `Example: 045F`
Only one address is required per capture item.

The **Capture Register** is a single character from the following list.

```
A       Register A
B       Register B
C       Register C
D       Register D
E       Register E
F       Register F
H       Heading                 Displays caption only
L       Line Feed
M       Register M1 or M2
P       P Register              Displays decimal value only
S       Status Register         Displays [1]=True, [0]=False for Status Flag Bits
R       RAM                     Data displayed is a RAM register
```

The **RAM Address** is a 2 digit hexadecimal number Example: 2E
A 2 digit RAM address must be specified if the [R] type is used

The **Display Mode** is a single character.

```
N       Data is displayed in nibbles          Example:  91230000000023
D       Data is displayed as decimal          Example:  -1.23E23
```

The **Caption** is a text message that will appear for each item and can be up to 20 characters long. Only the characters [0..9], [a..z], [A..Z] and [space] are allowed.

The following examples show a capture file with 2 items added. One is for capturing the start of the Division routine, and another for capturing the end of the division routine with an answer.

Example 1: The HP32E division routine can be captured starting at ROM Address 064F

```
ROM Addr  Item #1    Item #2       Item #3     ... Item #(x) etc
  064F H Division/DD Numerator/CN Denominator/  (More if required)
```

The item above captures this data when ROM address 064F is executed.

```
064F Division               Item #1       Heading only
Numerator                   Item #2       Caption
D: 555                      Item #2       Register D displayed in Decimal
Denominator                 Item #3       Caption
C: 03300000000001           Item #3       Register C displayed as Nibbles
```

Example 2: The division result can be captured at ROM Address: 005C

```
ROM Addr Item #1 Item #2
  005C H Answer/CD
```

Note that Item #2 has no [Caption] text

The item above captures this data when ROM address 005C is executed.

```
005C Answer                 Item #1       Heading only
C: 16.81818182              Item #1       Register D displayed in Decimal
```

Example 3: Capturing RAM address [2F] when ROM address 031D executes.

```
ROM Addr      Item #1    Item #2
   031D H Memory Capture/R24D
```

The item above captures this data when ROM address 031D is executed.

```
031D Memory Capture          Item #1        Heading only
R24: 254.9                   Item #2        RAM 24 displayed in Decimal
```

Example 4: Capturing M1 and M2 Registers when ROM address 002A executes with a line feed.

```
ROM Addr  Item #1      Item #2        Item #3    Item #4
   002A H Example 4/M1D This is M1/M2N This is M2/L
```

Notes:
A forward slash has been included on the end of the last item, but as it is the last item, it may be omitted. To match the 4 character rule, there are 3 periods after the H definition, and a period after the [1] and [2] for the [M] register definitions.

The item above captures this data when ROM address 002A is executed.

```
002A Example 4               Item #1        Heading only
This is M1                   Item #2        Caption
M1: 22.987                   Item #2        M1 displayed in Decimal
This is M2                   Item #3        Caption
M2: 01000000000000           Item #3        M2 displayed in Nibbles
{Line Feed}
```

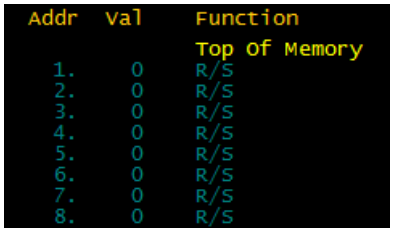
Additional Buttons

Opens a window to show the internals of the selected calculator

Opens a window to show the back panel label of the selected calculator – if available

Program Memory

This window shows the contents of the program memory space.



The display shows the memory location, the decimal key value stored there and the type of instruction the value represents.

Program File Buttons

 Clear the program memory card

 Open a saved program card file

 Save the current program card to a file

When saving program files, you will be prompted to associate a program card with this file. If a program card bitmap file is available with the same filename, it will be displayed in the save dialog as a first choice. You will also be prompted to enter a name for the program, up to 20 characters in length.

Note: The program card file does not include path information, only the filename. Therefore, to load correctly, the card file must be stored in the same directory as the program file.

Most of the programming examples from the HP67 Owners Handbook are available ready to use in the installation directory.

An easy way to review the newly entered program is to switch to RUN mode, press RTN to reset the program pointer and then switch back to PRGM mode and use the SST button to single step through the program to verify it.

Saving Loading Cards for the HP97

The HP-97 microcode handles the actual saving and loading of files, therefore the simulator must be in [Run Mode] to save or load files to and from the PC hard disk.

Program	If the program memory is greater than 112 steps then 2 cards will be saved.
Data	If any of the secondary data memory is non zero, then 2 cards will be saved

To save program cards, make sure the [Run – PRGM] switch is in the **W/Pgm** position.
Press the Save Program Card File button.
If 2 cards are to be saved, [Crd] will be displayed after the first save.
Press the Save Program Card File button again to save card #2.

To save data cards, make sure the [Run –PRGM] switch is in the **Run** position.
Click the calculator buttons [f] [W/Data] to begin.
When prompted by [Crd] on the display, press the Save Program Card File button.
If 2 cards are to be saved, [Crd] will be displayed again.
Press the Save Program Card File button again to save card #2.

When loading program or data files, make sure the [Run – PRGM] switch is in the **Run** position.
Press the Load Program File Button.
If 2 cards are to be loaded, [Crd] will be displayed again.
Press the Load Program Card File button to load card #2.

PAUSE Card Load

The software supports parking cards in the reader slot ready to load programs from the PAUSE instruction, or save data with the W/Data instruction. See HP-67 owner manual Page 292.

To park a card in the slot ready for when a PAUSE instruction executes, make sure a program is running and press and hold [SHIFT] on the computer keyboard then click the Load Program File button. The card file will be loaded and the decimal point LED for the exponent units digit will be lit on the display. When a PAUSE instruction executes, the HP microcode will load the card data.

To park a card in the slot ready for when a W/Data instruction executes, make sure a program is running and press and hold [CTRL] on the computer keyboard then click the Load Program File button. The decimal point LED for the exponent units digit will be lit on the display. When a W/Data instruction executes, the HP microcode will save the card data.

The menu item [Program] -> [Card] -> [Clear] will clear a parked card.

If the W/PGRM switch is in the wrong position when trying to load or save files, an error will occur and the load/save process will be aborted, or in the case of parked cards, the actual read/write will not work.

Programmer Screen Buttons

This window is used to communicate with the hardware calculator via the USB port.

Program the selected function

Read the selected function

Cancel an active programming process

Reset the PIC calculator

Clear the communications response memo

Set the displayed COM Port

Memory card index

Program/Read Options

1) Calculator Driver — Write

Program the driver code into the PIC chip [PICcalc1519.hex]

2) Calculator Code — Write

Program the current calculator assembler listing code into the PIC chip.
The code will be assembled prior to programming and any errors abort the programming process.

3) Info — Read

Read the calculator type information currently programmed into the PIC chip.

4) Memory Card HP-65, HP-67 — Read/Write

Select this option to read/write a file to/from the memory card.
The memory card must be inserted prior to performing this option otherwise an error will result.

Memory Card Read/Write Options 

- Program — Write Mode: Program* — Read Mode: Nil
- Data — Write Mode: Data* — Read Mode: Nil
- Directory — Write Mode: Nil — Read Mode: Read card directory

* Current program or data stored in the PC calculator, or from a PC disk file

5) Display Type — Write

Select this option to change the display type between the default LED display or the LCD option.
The alternate PIC code is for a generic LCD 2 line x 16 character type with a ST7066U controller.

Memory Card Write - Source File Options

The source file for memory card writes can be one of the following...

- Current PC calculator program stored to current memory card index
- Load a single file stored to current memory card index
- Load a library of files stored in a single PC disk directory stored starting at card index 00

The memory card read options can be one of the following...

- Read memory at current memory card index
- Read all card files as a library and store in a single PC disk directory

Read single card - File save options... (from selected memory card index)

- List in responses memo
- Save to disk file
- Send to PC clipboard as a text file

HP-97S Calculator

Note: Please read the HP-97S Installation and Operation Guide to fully understand how the Interface works.

The HP-97S calculator works exactly like the HP-97 except that it has a 50 pin interface that can connect the calculator to external equipment.

To select the HP-97S calculator, right click the calculator and select [Calculator Type] -> HP97S.

If you open the Debug Screen, (F12), you can click on the 97S I/O tag to view some operational options.

The Interface has 10 inputs (Digits A to J) that can be set to 0 to 9, Decimal Point, EEX, ENTER, Continue at A, CHS and NO-OP.

All of these inputs, when processed by the Interface, are the equivalent of pressing the associated calculator key, with the exception on NO-OP which does nothing.

The four flags (F0, F1, F2, and F3) that are integral to the HP-97 calculator have LEDs that indicate whether the Flag is SET or CLEAR.

The LE LED indicates when the Interface is ready for use.

The LOAD button activates the Interface and is Equivalent to the LOAD input.

The Inhibit button disables the Interface.

The T/C button alternates between positive and negative true.

The Reset button resets the interface.

So far, there are 3 options available to experiment with the interface. Each of the options has its own program card available in the HP97.zip file. These programs are the same as some of the examples in the HP-97S manual.

Manual Input

Please load the [Manual Input] program card into the calculator before use.
Press [B] to initialise the Interface. – The LE LED should be lit
Set any number in the Digit items and make sure the [A] is selected last
NO-Ops can follow the [A] item.
Press [Load]
The data from the Digit inputs should be printed on the calculator.

Volt Meter

Please load the [Mean Stan] program card into the calculator before use
Press [B] to initialise the Interface. – The LE LED should be lit
There will be a slider to adjust the voltage and will set the digits
Press [Load]
The data from the Digit inputs should be printed on the calculator and after 10 values have been loaded, the Mean and Standard deviation will be printed

Continous Sample

Please load the [Continuous] program card into the calculator before use

Select Continuous Sample from the drop down list.
If the Interface is ready, the LE output should trigger the Load input and send random digit data to the calculator. There will be 10 pauses and the Interface will be reset causing the LE to re-trigger the Load input. Each time, a random value will be printed on the calculator.
To stop the process, press [Reset], Inhibit, or select a different experiment.

Program Notes

It may be handy to have the calculator display a note when a certain event happens during program execution. This function allows that to happen and up to 100 messages can be stored for use.

Notes can have up to 11 characters displayed and only these listed characters can be used.

To open the Notes Editor, the HP-67 must be selected and the simulator must be in Step Mode.

Enter a note by typing any of the allowed characters into the list. When completed, press OK. You will be notified if any errors occur.

To enable any of these notes while a program is running you need to include the following keys in your program.

0	E	b	q
1	F	c	r
2	G	d	s
3	H	f	t
4	I	g	u
5	J	h	y
6	L	i	=
7	N	j	?
8	O	l	_
9	P	n	-
A	Q	o	[spc]
C	U	p	

A single digit number may be used to specify the note numbers 0 – 9.
A 2 digit number is required to specify the note numbers 10 – 99.
The note number is accessed from the X Register in the calculator during program execution.
Nibble [11] holds the 10's value and Nibble [12] holds the units value. Any number can be in the X register, so a blank note will be displayed for note number values outside the useable range.
The note number needs to be entered units first and tens second. This is due to the way the X register is normalised in the calculator.

A normally unused key combination is used to initiate the note. This is **[f] [f] [A]**.
This key combination will be shown on the display as [**16 16 11**] and in the program list as **RSNT**.

During program execution, when the notes program token is encountered, the program will stop as if **R/S** was pressed and the selected note will be displayed. To continue running the program press **R/S**.

Another key combination is available. This is **[f] [f] [B]**, [**16 16 12**] and in the program list as **PSNT**. This function displays the note during a PAUSE sequence, then the program will continue.

Examples:
You wish note number 29 to be displayed and the program will stop.

Press	9	Display	**001**	**09**
	2		**002**	**02**
	f f A		**003**	**16 16 11**

You wish note number 07 to be displayed briefly and then the program continues.

Press	7	Display	**001**	**07**
	0		**002**	**00**
	f f B		**003**	**16 16 12**

This feature can be enabled or disabled by clicking on the appropriate menu item.

Notes: The note address is pushed onto the stack so the T register contents will be lost.
Note displays cannot be accessed from the keyboard in RUN mode.

Each time one of these functions executes, the C register will be incremented by 1. This can help with concatenating notes to form a message.