Classic Calculator Emulator +

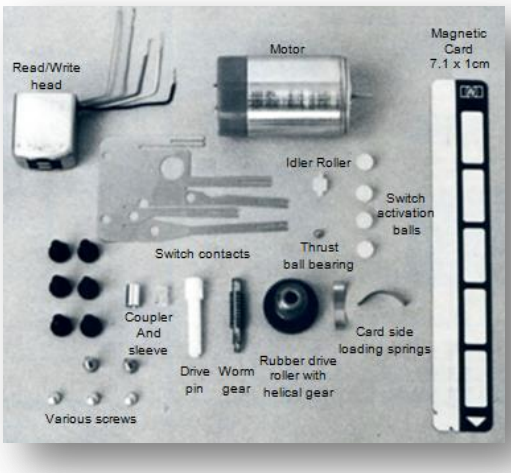
Features:

- Emulates HP01*, HP10, HP19C, HP21, HP22, HP25, HP25C, HP27, HP29C
  HP31E, HP32E, HP33E, HP33C, HP34E, HP37E, HP38E, HP38C
  HP35, HP35 (Bug), HP45, HP55, HP65, HP67, HP70, HP80
  HP46
  HP91, HP92*, HP95C*, HP97
  HP56i*
- PC emulator[1] and/or build a real calculator
- Memory storage for HP65 and HP67 and other programmable models
- Emulates continuous memory for HP19C, HP25C, HP29C, HP33C, HP34C and HP38C
- Actual code from original documents[2]
- Simulator
- Assembler
- Code Trace
- Memory Card Text Editor
- HP65 DemoPac1, MathPac1, EEPac1, StandPac1, AVpac1
- HP-67 Demo programs from manual
- USB interface with ability to upgrade software
- In circuit PIC programming[3]

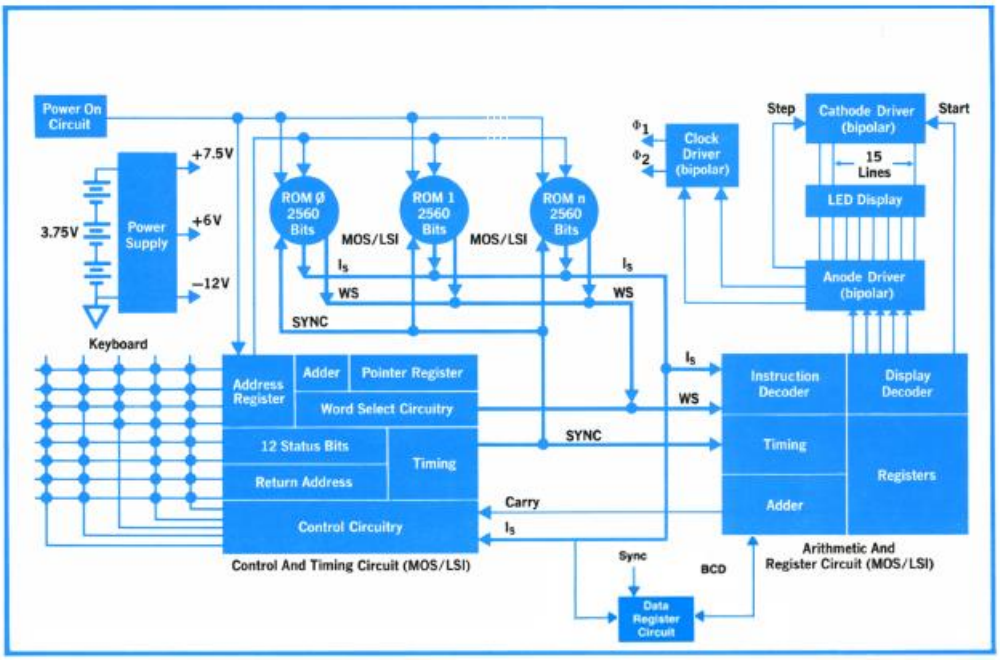
Project Design by Tony Nixon tnixon059@gmail.com
http://www.teenix.org

1 Best viewed in 1920 x 1080 screen resolution. (Minimum is 1280 x 720)
2 See Notes section
3 PIC programmer initially required to enable a boot loader program.
* Not supported with current PIC emulator

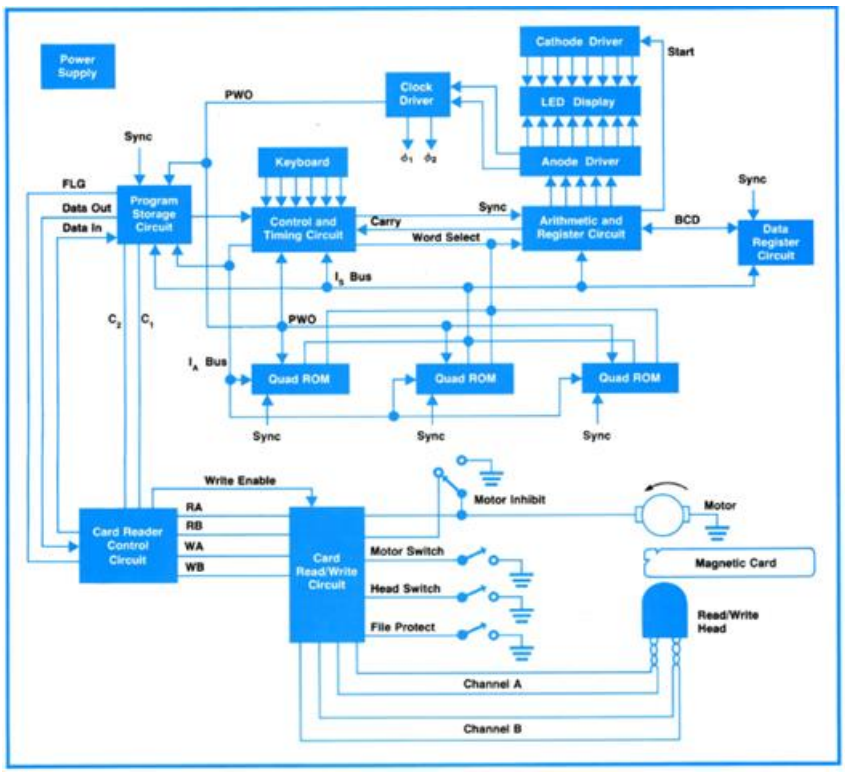
Table of Contents



Classic Block Diagram



General layout of classic series



HP65

Calculator

Menu

The menu is activated by right clicking on the calculator

Menu Items

Calculator Type	Selects the calculator type to use
For The Inquisitive...	Opens the debug screen in Single Step mode
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.
Spice Display Mode	Select Standard or European display mode
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
Print Roll	Print the contents of the print roll buffer
Discard Roll	Clear the contents of the print roll buffer The print roll buffer holds 200 printed lines
Print Head Speed	Adjusts the speed of the print head movement
Auto RAM Bank Switch	Toggles automatic RAM bank switching during Slow or Step modes
Red Dot	Use HP-35 calculator as a red dot model
HP01 Display ON	Enable HP-01 display continuously
Fine Tune Timer	Tweak the timing interval for the calculator code execution and HP01 clock timing PC sync – see HP01 section.
67 Extra Memory	Use RAM at address $3A, $3B and $3C for useable memory. Uses modified HP-67 microcode. Access via [f] [P<>S].
Program	Load/Clear HP65/67 programs and view user card images
Program Trace	View a trace of the current calculator program
67 Continuous Memory	Active when the HP-67 is selected – See 67 ConMem
Program Notes	See Program Notes
Program Capture	See Capture
Memory Fail	Enter HEX address range for simulated failed ROM or RAM
Owners Handbook	Opens the OH in PDF format (If available)
Quick Reference Guide	Opens the QRG in PDF format (If available)
Patent Document	Opens the patent document in PDF format (If available)
Close	Closes the program

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 is clicked on the calculator, the PC will recognise the calculator keyboard – pad colour is pink. When the mouse has been clicked on the assembler pad, the PC will direct key input to the pad – pad colour is white.

Key Short Cuts

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

Additional:

CTRL n	New Assembler file
CTRL o	Open assembler file
CTRL s	Save assembler file
CTRL p	Toggles the calculator power
F1	HP instructions listing
F2	Close the program
F5	Code Debug - Auto step code at slow speed
F7	Code Debug - Single step code
F8	Code Debug – Run until following code address (Eg. Execute a subroutine)
F9	Code Debug - Run code normally
F12	Toggles calculator/debugger screens

Printer Roll

The printer roll can hold up to 200 lines of print text.
If the print roll exceeds this amount then the first line will be deleted to make room for the next line. When this occurs, the print roll lines indicator will turn red.

Constant Memory

The following models have constant memory that work with the PIC based calculator.

HP-19c
HP-25c
HP-29c
HP-34c
HP-33c
HP-38c

When the PIC calculator is set to one of these models, and is turned OFF and then back ON again, the calculator will retain some of its memory. Check with the calculator manual for the specific model to see what is not saved.

If you remove all power from the calculator, then as with the real calculators, the memory will be lost. This equates to changing batteries, or removing USB power without batteries connected.

When you change to a different calculator model, the constant memory is saved to the PIC internal memory prior to the change.

When you change back to this model calculator, the constant memory information will be recalled, however, the restraints for recalling constant memory still apply for a particular model.

Note: When you connect to the calculator via the USB port, the calculator will reset. This has the same effect as removing all power from the calculator and therefore the current memory will be lost.

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

 Sticky key*

 Reset the calculator

 Reload the original calculator micro-code file

 Open the Program Card Editor

 Disassembles an instruction op-code or displays instruction map.

* The Sticky Key button holds a calculator key in the down position while simulating in slow or step modes. Click the Sticky Key button, then click a calculator key and it will stay down. To release the key, click the Sticky Key button again so it is in the up position. When the sticky key button is down, the trace will not be cleared on a key press.

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-67 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
```

HP-01 Mode

Clock timing is derived from the PC system clock and you may notice the seconds update at uneven times. This is a Windows timing problem but should not affect the overall accuracy. If the seconds skip numbers while updating, use the Timer Adjust from the Options menu to speed the time refresh up until the seconds update normally. The stopwatch timer is updated differently due to the 100th second increments and the Timer Adjust has no affect. The timer adjust value is stored in the emulator ini file.

When the simulator screen is open, the calculator display will be active continuously. It will revert to the 3/7 second timed on mode when the simulator screen is closed.

The HP-01 goes into Sleep mode when after a calculator key press has been acted on, but the timer registers still function. While asleep, you cannot simulate the code from the Step, Slow and Run keys as nothing is happening. To simulate the code after wakeup, you can set a break point after the sleep instruction which will stop code execution after a calculator key press. You can also pre-set the code execution speed by clicking on the Step, Slow or Run buttons. Then after the calculator key has been pressed, the calculator will wake up and code will be simulated at a speed according to the new run mode.

Some HP01 registers will be saved on closing the application so that when the HP01 is running again it will operate with the same settings. The [T] button will be pressed on start up if the HP01.ini file is found so that the calculator code can update its own registers.

Note: This file will not be saved unless the Emulator is running in Step or Slow modes.

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               jsb @207
      go to $FE     HEX                 jsb $0E
      go to .100    DECIMAL             jsb .23
      go to Label   LABEL               jsb Label
```

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

Target Label Names

Labels are limited to 6 characters and must be followed by a colon.
Labels names can only have the characters 0 - 9, a - z, A – Z
Labels are 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
	Stop On Buffer Full	If this button is in the down position, the code stops running when the trace buffer is full. After stopping, the trace code is copied to the PC clipboard and the trace buffer is cleared. Press the Step, Slow or Run button to continue.

A new trace begins each time the Run or Slow buttons, or any of keys are clicked. Note that, if the Stick Key is down, the trave will not be cleared.

Capture

This mode can capture trace data only when specific ROM addresses have executed. You can set the ROM address, which registers to display and the way the data is displayed. You can also add some text to describe the capture point.

Menu Items

Editor	Opens the data capture editor
Load File	Load a capture file
	Once loaded, the capture file will begin to work immediately
Clear File	Clear the loaded capture file
Capture To Clipboard	Transfers the capture data to the PC clipboard
Clear Capture	
Now	Clear the captured trace data now
On Run	Clear the captured trace data when [Run] starts the simulator

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.

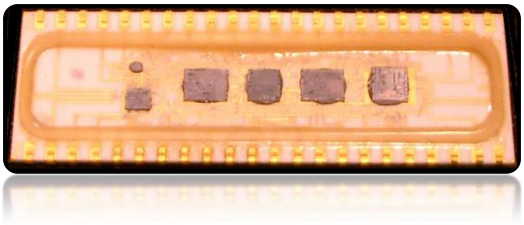
```
002A Example 4              Item #1     Heading only
This is M1                  Item #2     Caption
M1: 22.987                  Item #2     M1 displayed in Decimal
This is M2                  Item #3     Caption
M2: 01000000000000          Item #3     M2 displayed in Nibbles
{Line Feed}
```

Additional Buttons

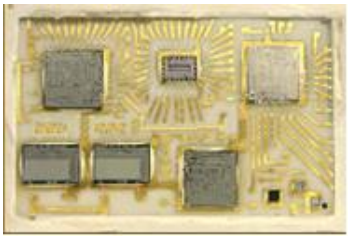
Opens a window to show the internals of the selected calculator

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

Opens a window to see a simulation of how the Classic 7 segment LED display is updated.
The operating screen is fairly basic and should be easy to use.
For a small discussion on Classic Display Operations see:
http://www.teenix.org/ClassicNotes.pdf



Metal can removed, inside the HP-65 hybrid chip set



Inside the HP-01

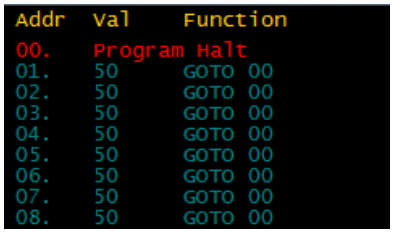
Program Memory

If the screen size permits, both the register panel and the program memory panel will be displayed. If only one panel can be displayed, two extra buttons will appear.

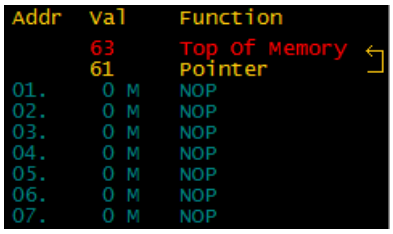
Show register panel

Show program memory panel

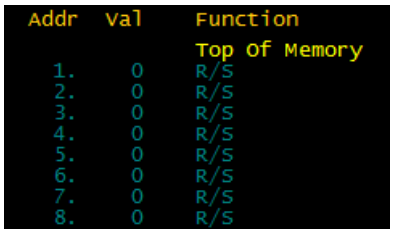
This window shows some example contents of the program memory space.



HP55



HP65



HP67

The display shows the memory location, the decimal key value stored there and the type of instruction the value represents. The HP65 also displays the main and sub pointers and [M] if the key value is 'merged' such as NOP which is a two key function [g and 1], giving a merged key value of 0. The merged keys are described in the HP65 Owners Handbook and the HP65 Patent Document which are available when right clicking the calculator menu.

Buttons – HP65 and HP67


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 HP65 and HP67 Owners Handbook are available ready to use in the installation..\HP65\DemoPac1 or ...\HP67\DemoPac1 directory. There are many HP65 programs available in the installation..HP65\MathPac1, StandPac1, EEpac1 and AVPac1 sub directories and each includes the applications book in PDF format.

To create a new program, first clear the calculator program memory by switching to W/PGM mode and pressing f CLx . Then click on the Clear Memory Card button

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 W/PGM mode and use the SST button to single step through the program to verify it.

Saving Loading Cards for the HP67

The HP67 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 – W/Pgm] 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 – W/Pgm] 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 – W/Pgm] 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.

Buttons – Saving Program Memory – HP25, HP29, HP33, 34, HP38 and HP55

 Save the program memory

 Load the program memory

These calculator types do not normally save and load program data. As such there are special procedures to follow before loading and saving program data files.

To save this type of program memory the simulator must be in **Single Step** mode.
Click the **Save** button to save the program memory to a file.

If you try to load a file while the calculator has been reset and is paused at the start of its microcode, then on running, the calculator microcode will erase the program memory, so to begin loading this type of program memory the simulator must be in **Run** or **Slow** mode and just waiting for a key press. Then select **Step** mode and press the **Load** button and select a file to load into memory. If the selected file does not match the screen calculator, the file load will abort.

The HP34C and the HP38E/C calculators allocate new memory as programming steps are added and this reduces the amount of memory available for storing numbers.

For these calculators, any values stored in the program memory space will be saved along with the program steps. This can be useful if your program needs to access a few stored constants as these can be stored in memory prior to saving a file.

See the respective calculator user handbook to determine the memory available when entering programs.

HP-34C	Memory locations from 0 to 31 will be saved, except for 15.
HP38E/C	Memory locations from 16 to 29 will be saved.

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 **[h] [A]** (HP-65 **[g] [A]**).
This key combination will be shown on the display as [**35 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 **[h] [B]**, (HP-65 **[g] [B]**), [**35 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**
	h A (HP-65 g A)		**003**	**35 11**

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

Press	7	Display	**001**	**07**
	0		**002**	**00**
	h B (HP-65 g B)		**003**	**35 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.

Spice HP-34C additional Program Code

The Spice HP-34C has an extra function added that will generate a random number in the range 0 to 0.9999999999.

To access this function, use the new key combination [g][0].

The random number will be placed into the X register after the stack is raised one level.

You can use this key combination in programs as well.

A program example of the use of this function could be to generate a random number between 0 and 99.

[g] [0]	X = 0.6735
1	
0	
0	
x	X = 67.3542
[h] INT	X = 67.0000

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

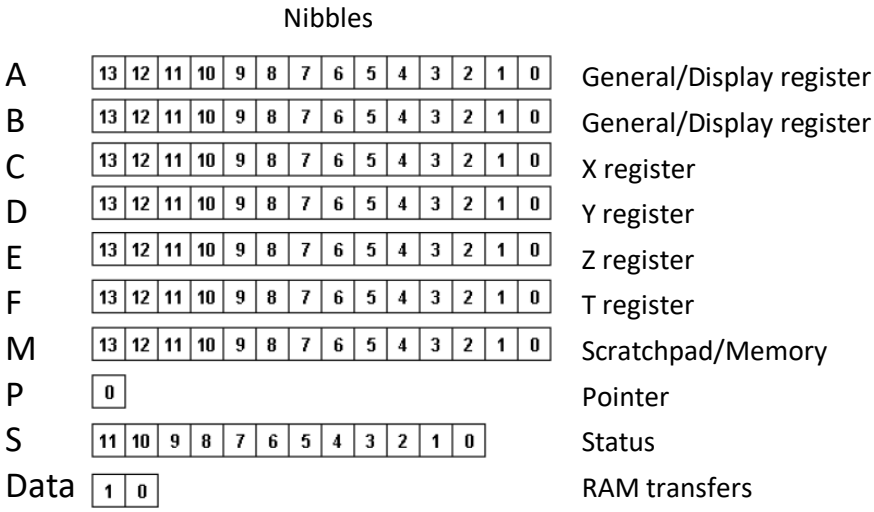
M Used as a scratchpad or memory register and only supported transfers to and from the C register

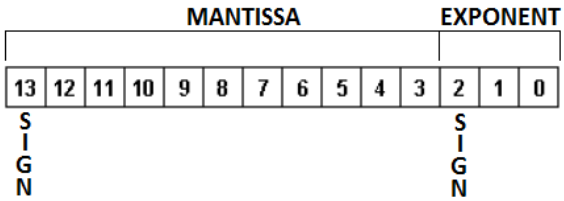
Additional Registers

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-65

              pointer advance         Increment program pointer
              mark and search         search for a label and mark it in buffer
              search for label        search for a label
              memory insert           insert a program element
              memory initialize
              memory full -> a        flag to a register if program memory is full

HP-67         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


HP-01         Calculator Watch commands

              0 -> s1-7  Clear status bits 1 to 7
              0 -> s8-15 Clear status bits 8 - 15
              dsscwp     Disable 1 second updates - not implimemnted
              enscwp     Enable 1 second updates - not implimemnted
```

sleep	Put calculator to sleep
gokeys	Decode key press
swstop	Stop stop watch count
swstrt	Start stop watch count
sw+	Stop watch counts up
sw-	Stop watch counts down
altog	Toggle alarm armed state
blink	Blink the display on and off
al -> a	Alarm register to A register
a -> al	A register to Alarm register
dsp -> a	Display register to A register
a -> dsp	A register to Display register, data feed from A register
a -> cl	A register to Clock register
a -> clrs	A register to Clock register with clock reset
cl -> a	Clock register to A register
a -> sw	A register to Stp Watch register
sw -> a	Stop Watch register to A register
a[p] -> f	A register nibble from P register pointer to F register
f -> a[p]	F register to A register nibble from P register pointer
dsp=sw	Display data feed from Stop Watch
dsp=al	Display data feed from Alarm
dsp=cl	Display data feed from Clock

Programmer Screen

Proper system functioning requires that you:

- connect the calculator to the PC USB port before opening the programming window.
- close the programming window before unplugging the calculator from the USB port.

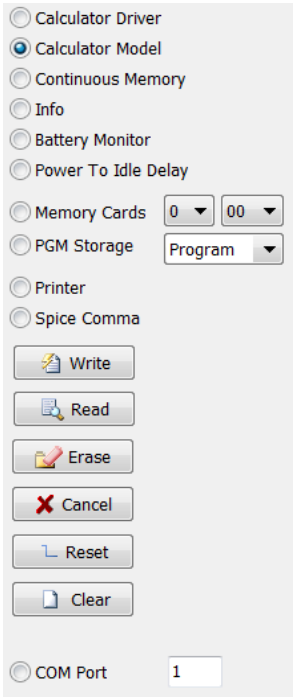
Failure to adhere to this process may result in failed communications between the PC and calculator.

If you disconnect the calculator from the USB port while the programming window is open, then the Emulator program may stop responding, usually for around 10 seconds. This is due to the change in conditions to the open USB interface.

Note: Communications to the calculator are only possible when it is turned ON. When the calculator is turned OFF, the USB communications are not monitored.

Buttons

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



Upgrade the PIC program to latest version
Set the calculator model
Interrogate or clear continuous memory state
Read the current calculator information
Battery Monitor set, read and test
Set the Power To Idle timer
Memory card access
Program storage access
Enable printer output
Enable Decimal Point or Blanks for Spice Comma*

Program the selected function

Read the selected function

Erase the selected function

Cancel an active process

Reset the PIC calculator

Clear the communications response memo

Set a communications COM Port

* Harald mode only

Program/Read Options

1) Calculator Driver — Read/Write

Read Read the current PIC code version

Write Programs the current driver code into the PIC chip (Located in ...install/PIC directory)

There are two modes of reprogramming available and will be automatically determined by the software when new updates become available.

1) Basic This mode only reprograms the basic core operating code
2) All This mode reprograms the entire PIC chip except for a protected area and will take a few minutes to do

It should not be necessary, but you can also do a Basic or All reprogram manually by selecting this option and holding the PC [Shift] key down before clicking on the [Write] button.

IMPORTANT: During the update process, do not remove power from the calculator, PC or unplug the USB cable during updates or the PIC processor may fail to operate.
If reprogramming fails, you may need to do an ISCP reprogram.
Contact the developer for help if required.

See Haralds Controller for USB reprogramming.

2) Calculator Model — Read/Write

A window will open allowing you to choose the operating calculator model.

Note: If the HP-10 is selected and the print mode switch is in the Printer Only position, the display will appear blank but the calculator will still be functioning properly.

3) Continuous Memory — Read/Erase

Read the continuous memory status for HP19c, HP29c and HP34c.
Erase the continuous memory that was saved when the calculator model was changed.

4) Info — Read

Read the calculator type information currently selected in the PIC chip.

If there is a later version of the PIC code in you installation directory, the listing will inform you that there is a later version ready to program. Use the Calculator Driver button and [Write] to update.

5) Battery Monitor — Read/Write

Read Read the battery monitor voltage level setting.

Write Test the Low Battery display, by turning it ON or OFF
Set the battery monitor voltage level – High, Medium, Default and Low.

6) Power To Idle Timer Read/Write

This function, if enabled, will put the calculator into low power idle mode when the timer expires. To wake the calculator up again, press any key. The calculator will resume operation from where it left off. If you turn off the power switch in idle mode, the calculator will switch off.

Read Read the Power To Idle setting.

Write Set the Power To Idle timer. The available options are OFF, or 1 to 10 minutes

7) Memory Cards HP-65, HP-67 Read/Write

Select this option to read/write a file to/from a memory card.
The program data is stored in memory and needs a block and memory location designator.
These can be set by the two drop down selectors.
There are 8 blocks [0] .. [7] and each can hold up to 51 card programs each.

8) PGM Storage HP-19c, HP-25, HP-29c, HP-34c, HP-38e and HP-55 Read/Write

Select this option to read/write program memory to/from a programmable calculator.
The program data is stored in memory and needs a block and memory location designator.
These can be set by the two drop down selectors.
There are 8 blocks [0] .. [7] and each can hold up to 32 calculator programs each.

Memory Card-Program Read/Write Options

- Program Write Mode: Program Read Mode: Read program file
- Data Write Mode: Data Read Mode: Read program file
- Directory Write Mode: Nil Read Mode: Read card directory

When writing program files, the data will be sourced from the PC hard disk. You can store single files or if you have multiple files stored in a specific directory you can transfer them all as a Library as long as the maximum block count is not exceeded.

When reading files from the calculator memory you can select a single file or all files in a block as a Library which can be saved to the PC. You can also transfer and display single files on screen or save it to the PC clipboard as a text file.

9) Printer

Select this option if the selected calculator has a printer, (Eg: HP-97) and you want a copy of the printer output. Up to 200 lines of text can be displayed at any time. These can be printed on any printer for a hard copy if desired.

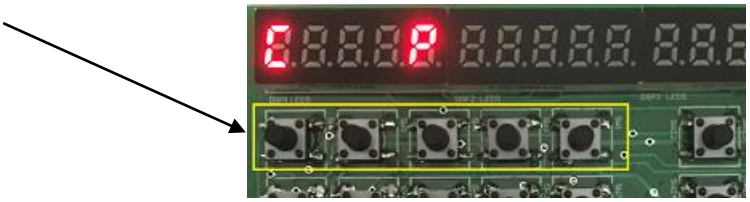
10) COM Port

Select the COM port for communications via the USB port to the calculator
Ports [0] to [9] can only be selected.

Menu Access

The top left row of keys are used for normal calculator operation or they control the menu items.

Menu Entry Button



To enable the menu, hold the Menu Entry button down for at least half a second.

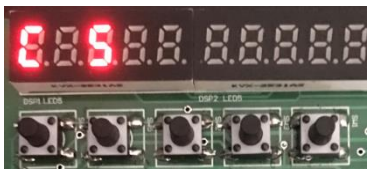
The basic menu display will show with [C] over the Menu entry button. This allows you to select a calculator model.

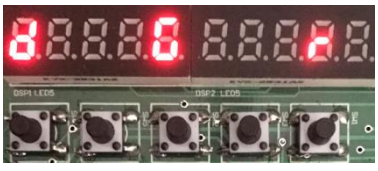
If the selected calculator model is a programmable type, then you will see a [P] displayed as well, so that you can select a program to Store or Recall.

To select a calculator model, press the button under the [C] character.

The display will change to that shown above. The currently selected calculator will be displayed as well as [P] and [N].

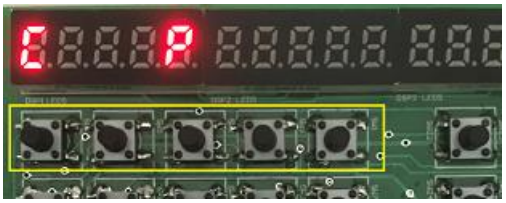
Click the button under the [P] to see the previous model available, or press the button under [N] to see the next model available. When the chosen calculator appears, press the [ENTER] button to select it. Any other button cancels calculator selection.





The HP91 has two 3 position switches on its front panel. The printer mode is activated from the mechanical switch and the DEG RAD GRD modes are selected by pressing the button under the [S] indicator, then select [d] [G] or [r] for the required mode.

To select a program to load or recall, enter the menu mode and press the button under the [P].



If the calculator model is HP-65 or HP-67 then you will be prompted for the program number to store or recall from memory. If the calculator is another programmable type then you need to specify whether you want to store or recall a program.

Press the button under RCL to Recall a program, or press the button under the STR to store a program. After making this selection the display will change to allow choosing the program number.

Note: The programmable models like the HP25 did not originally have the ability to store or recall programs. This means their own software does not have any way of dealing with programs that suddenly "appear" in memory. The most neutral operation of these calculators is when they are sitting idle in the RUN mode without programs running. This is the best state for these calculators to be in when loading or saving a program. The program pointers will be set to zero when a program has been loaded.

For the HP-65 and HP-67 calculators, you do not need to select RCL or STR because that function is determined by the position of the W/PGM - RUN switch.

To store a program for these calculators, move the switch to the W/PGM position before accessing the menu. To recall a program, make sure the switch is in the RUN position. This action mimics the card read/write function for these calculators. The HP-67 makes card transfers under its own micro-code control so the switch must be in the correct position.

There are 8 memory blocks available for Card Program storage and another 8 blocks available for Program storage for the other models.

The HP-65 and HP-67 can store up to 51 programs in each block numbered [00] to [50].

The other programmable models can store up to 32 programs in each block, numbered [00] to [31].

The software knows what type of program the calculator expects. You just need to select a block and a program number from that block.

To select a program from block 2, program 31 for the HP65 key in [2] [3] [1].

If you make a mistake then press the CLx button, or continue entering numbers as they will overwrite themselves after the third digit.

Once the correct digits have been entered, press [ENTER] to select the program.

If the chosen calculator program does not match the current calculator model, then a TYPE ERROR message will be displayed.

If the program selection is vacant then the ERASED message will be displayed.

Press any other key other than a digit or ENTER to terminate the menu without loading or saving a program.

If you try to enter a program number outside the limits for a particular model, the menu process will terminate.

Storing memory (data) to a card. (HP-67 only) Program Mode Switch: RUN Position

Press the [**f**] and [**WDATA**] buttons and the calculator display will show [**Crd**]. Enter the memory card storage location as described before. If two cards are required the calculator will again display [**Crd**]. Repeat the process to store the second data card.

Harald's Controller Display

The display menus for Harald's Classic Controller module are much the same as the MultiCalc displays. It is activated in the same way, by holding down the top left key for longer than about ½ second. In the case of a HP-45 keyboard, this will be the 1/x key.

Each menu item is selected from the key directly below that item. For example, the 1/x key selects [C] and the Shift key selects [A]. The e^x key exits the menu mode in this displayed menu because it has no selection item.

Switch menu items are always displayed for this controller as it is expected that the Classic model does not have any other switch other than the Power switch.

The [SC] menu item is to Store or Clear Continuous Memory. This item is required because when the Classic Power switch is turned OFF, the calculator completely shuts down and all memory is lost. To be able to save the data from models with continuous memory, you need to select this menu item before turning off the calculator.

When selected a sub menu will appear.

Str	Stores the Continuous Memory
Clr	Clears the Continuous Memory

Note: Harald's Classic Controller retains the previous Continuous Memory data when switched off. If this is not required, use the [Clr] function.

The [Str] procedure will stop a running program, and will not reset the registers that are supposed to be cleared according to the respective owner manual. However, when you turn the calculator off then back on, the Constant Memory registers are recovered from storage and the required registers will be cleared.

NOTE: When changing models, the continuous memory is automatically saved.

The HP91 has two 3 position switches on its front panel. When selected, this will be the main menu display. The printer modes are selected from the [SP] menu indicator. Choose [mAn] [ALL] or [nor] as required. The DEG RAD GRD trigonometric modes are selected by pressing the button under the [St] indicator, then select [dEG] [Grd] or [rAd] for the required mode.

The [A] menu item displays a short About message.

Calculator models have different switch text shown on the display depending on the calculator model selected. For example, the HP21 has a DEG or RAD mode select switch.

Normally with a 2 or 3 position switch display, the left, centre and right keys are used for selection.

NOTE: The displayed menu items may change depending on the calculator model that is selected.

Some models have multiple switches for Printer modes and Program Run modes displayed.

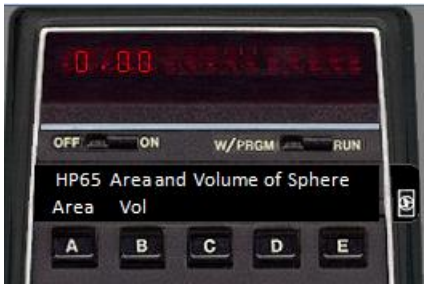
M Selects Printer Manual
Tr Selects Printer Trace
Nr Selects Printer Normal

PG Selects Program Mode
rn Selects Run Mode

PC Program Cards – HP-65 and HP-67

If the selected calculator is the HP-65 or HP-67 then you will be prompted to associate a card image when you save a program or data file from the PC. When you reload the file later on, if an image is associated and is available from the same file location, it will be loaded into the program card viewing slot on the calculator face.

If the card image is not included with a program file, you can open one by using the following method. Right Click on calculator - Select -> Program -> Card -> Load menu item.



Program cards can be made up using any graphics editor and must be saved in 24 bit BMP format. These graphic files can be displayed on the calculator face after loading a program or data file.

The images should be sized at 215 pixels wide by 35 pixels high. If the images are any larger they will be cropped before displaying. White text on a black background works well.

Example “TestCard65.bmp” from ..HP65\DemoPac1 directory

There is also a “Template65.bmp” which can be used as a template for new designs. There are small dots along the bottom of the image which correspond with the centre of the A – E keys.

Experimentation on image size and text positioning may be required depending on the screen resolution of your computer.

There is a Word document called ”HP65TestCard.docx” available in the HP65 module installation file and this can be used to create text at a size that can be transferred onto the card images. You can create the text for the cards with the document template and [Print Screen] the Word document into the PC clipboard. Then paste the image into MS Paint where you can easily shift the card information around and then save it as a Windows 24 bit bitmap with the dimensions shown above.

Each program file in the installation...HP65\DemoPac1, MathPac1, EEPac1, StandPac1 and AVpac1 sub directories have corresponding card images ready to use and will be displayed on the calculator when one of these programs are loaded.

Note: Card images must be saved in the same directory as the program or data file.

Program Card Editor

This window allows you to edit program files for the programmable models. The operating mode is determined by the selected calculator model from the emulator.

 New file

 Clear card image file

 Open text file

 Save text file

 Open formatted card file — Includes status and card image information

 Save formatted card file — Includes status and card image information

 Import calculator program memory

 Export list to calculator memory

 Syntax check the text listing

 Covert text in the form of key coordinates to program codes
Example: Changes 31 25 11 to f LBL A

When a formatted file is loaded or imported from the calculator, the program elements will appear in the correct listing addresses. Formatted files are card files that can be loaded directly into the PC or hardware calculator emulators. They are formatted the same as the original magnetic cards.

Text File Format

Text files can be loaded with the following two formats:

Numbered lines	Without numbering
1. LBL A	LBL A
2. 1	1
3. ENTER	ENTER
4. 2	2
5. +	+
6. RTN	RTN

After loading, the numbering will be stripped from the text. Text files will be saved with the numbering added. The numbering is assumed to terminate with a period (.) character.

When the **Delete trailing blanks on load/save** item is checked, when importing, saving or loading text files, any blank data that may be in the file after the last valid code line is deleted.

Lines may have comments added if they follow the comment delimiter - //.

```
LBL A                          // This program begins with the 'A' key
```

Note: Comments and numbering are not saved in formatted card files.

Sort — If checked, the codes will be sorted into alphabetical order.

No Trailing Blanks — If checked, text files will not be padded with blank code values when files are saved or when importing program memory from a calculator.

HP-67/97 Mode

Drop down lists and check boxes provide a means of setting the HP-67/97 status information for each program card. S5 and S6 are set automatically by the software.

```
STATUS CARD BITS
 S0 S1      S2       S3       S4            S5                          S6
 0  0 SCI   0 - 9    0 DEG    3210 BITS     1 SINGLE SIDED              1 Data Side 1
 0  4 ENG   DSPx     1 RAD    1234 FLAGS    0 PART OF 2 SIDED PAIR      2 Data Side 2
 2  2 FIX             2 GRAD                                             3 Prog Side 1 (001-112)
                                                                         4 Prog Side 2 (113-224)
```

These items are only available to set card information. They will be updated when the calculator program is imported to the editor, but will not be used when the program is exported back to the calculator.

Construction Notes

Note: Observe static handling procedures

The calculator schematic and PCB designs for top and bottom layers can be found in PDF format in the install directory...\schematics. There is a zip file in the same directory which contains the Gerber and drill plot files for the PCB. This file can be sent to a prototyping PCB manufacturer to create the board.

Circuit Description

The circuit is simple and is based on a PIC18F47K40 chip running at 12.288MHz. Its job is to emulate the original calculator code and hardware and communicate with a PC.

The displays used are five digit common cathode LED modules. All of Port A pins with the exception of RA4, plus all of Port E pins are used to drive the 8 segments for each display via 680R resistors. They set the maximum steady forward current per segment to around 5mA, however the duty cycle is only about 6%. This limits the total current that the PIC has to deliver to each digit. Port D pins RD0 – RD7 and Port B pins RB0 – RB6 control the common cathodes and sink the total current for each digit. With the resistors chosen, the maximum current required for the calculator is around 27mA.

The keys are arranged in a six row by six column array. Port D pins RD0 – RD5 are connected to the cathodes of digits one to five and also sequentially drive the key columns low. Port C pins RC0 – RC5 are used to read the key rows which are normally held high via the 10K resistors R9 – R14. If a key is pressed during the scan process, the corresponding row will be brought low. The PIC will detect this and generate the same key code and set internal logic as happens in the original calculators. Diodes D1 – D6 provide isolation to stop the keys interfering with the display cathodes. The two slide switches are also connected to the RC0 key row pin. They are tested for a low state during the display scan process from pins RD6, RD7, RB0 and RB1. Diodes D7, D8, D11, D12 and D13 provide the isolation for these items.

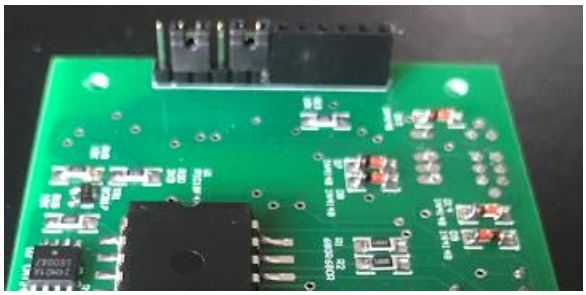
U2 is a CAT24MO1 128KB I2C serial EEPROM chip and is used to store the programs for the programmable calculators. PIC pin RA4 is open collector and transfers data to and from the 24LC64 I2C chip via the SDA pin which is held high by R16. Pin RB7 provides the clock pulses for SCL.

U3 is a FTDI USB to Serial converter chip and communicates to a PC USB port for serial communications to the PIC. RC6 and RC7 are the TX and RX pins respectively and communicate at 19200 baud. The chip is configured for self powered operation and provides power for the calculator when connected to the PC. Diodes D11 and D12 isolates the PC power from the battery. R17 holds the RX pin low when the chip is disconnected from the USB port. The FTDI and PIC chips are programmed to invert the data signals so that the TX pins are held low while in an idle state. This stops the PIC chip trying to drive the FTDI chip when it is not powered.

The PC can reset the calculator at any time under user control. This is accomplished via the DTR serial line from the FTDI chip. This controls TR1 which pulls the MCLR pin low.

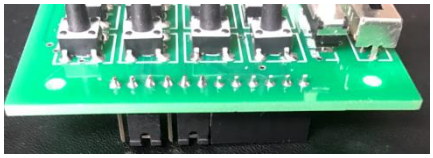
Connector CN1 is used for In Circuit Serial Programming (ICSP) for the PIC chip. Pins 6 to 12 have been configured to accept a Microchip® PICkit™ 3 serial programmer.

For normal calculator operation, 2 links connect pins (2 and 3) and pins (5 and 6).



Construction

Construction requires the placement of surface mount components but the PCB pads were enlarged slightly to accommodate home construction. To avoid using specialist parts, some through hole components are mounted as surface mount types to reduce clutter on the top side of the PCB.



The membrane switches are surface mount types and should be soldered to the calculator board first. The circuit board is widest looking East/West, the switches are also mounted East/West when looking at the two sets of adjacent terminals. To mount a switch, pre solder any one the four PCB pads. Then by using tweezers or small needle nose pliers, carefully place and align the switch central and square over the four mounting pads. Apply a small downwards pressure on the switch and reheat the solder with the soldering iron and the switch will sink down into the pooled solder. Quickly remove the iron to avoid overheating the switch and wait briefly until the solder solidifies. If satisfied with the switch position, the other three pins can then be soldered. Continue the process for the other switches.

Solder the 3 display modules next taking care to observe the correct orientation. Unfortunately, the manufacturer did not make these displays so that they evenly end stack and would have had a gap between each of the five digits. The display modules have been reduced in size to make the overall display more even. The original PCB design did not cater for this so the leads have been bent outwards slightly for the two outside displays. They will be a tight fit when mounting which is ok. **Do not** remove any more material from the display module ends or you risk destroying the miniature PCB inside the module and some LEDs may not work. Use minimal heat to avoid damage. You can solder a pin on display 1, then solder a pin on display 2 then one on display 3 and continue the pin rotation to keep the heat to a minimum.

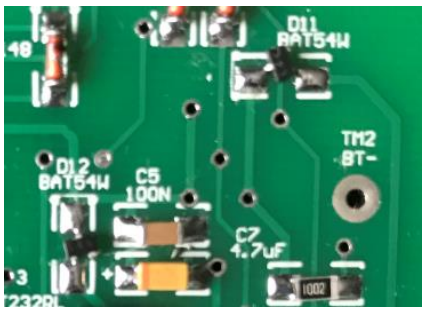
The two switches can be soldered next. Try to make sure the top of the switch actuators are above the height of the push button switches. As a guide the switch connectors should be flush with the underside of the PCB before soldering. After these are in place, the board can be turned over to begin adding the underside components.

A similar method can be used for soldering for the small surface mount parts. Melt a small amount of solder to one PCB pad per component. Using tweezers hold the component aligned properly just

to one side of the PCB pads. With the other hand, melt the solder and hold the soldering iron in place. Slide the component into the melted pool of solder until the edge of the component touches the soldering iron tip. Now remove the soldering iron and the solder will cool and solidify quickly. Complete this process for all the other components except for the ICs..Once completed, check your work, especially the orientation of the diodes and capacitor C7 – 4.7uF Tantalum (+ lead is marked with a line). Now go around and solder the unsoldered terminals.

Once completed, it is possible that some of the first connections didn't take properly and if you suspect a dry joint resolder it remembering that these components are small and too much heat may damage them. Mount transistor TR1 next. Use the same solder technique and connect the top pin first. Allow it to cool and solder the 2 lower pins, also allowing it to cool.

Diodes D11 and D12 may be supplied as a SOT323 - 3 pin package. If so, then mount them diagonally as shown in the diagram. The top single pin is the cathode. The bottom left pin is the anode, and the bottom right pin is not connected. If they are a standard 2 pin type then just mount the same as the other diodes.



Mount the connectors for the ICSP next, as shown in the diagram. Insert the two links for the ISCP connector, so that pins (2 and 3) are shorted and pins (5 and 6) are shorted for normal operation. These connectors can have their pins bent at right angles and can be laid flat on the PCB if you want to reduce the overall height.

The USB connector can be soldered next using a fine tipped soldering iron. If the solder makes a bridge between the small contacts, use some solder wick to remove the excess.

Caution: The chips used in this project may be damaged by static electricity caused by touching the pins or PCB connections. Always handle the PCB by the edges.

Note: Mount the FTDI chip before the PIC and serial EEPROM chips.

The FDTI chip can be soldered using a small chisel tip soldering iron. Try to position the chip and solder a single pin on one of the corners so that is stays put when soldering the other pins. Before continuing, check the orientation. Use small diameter solder and lay it along one side of the chip over the pins to be soldered. Run the chisel tip soldering iron along the solder and it should solder all pins. Do the same for the other side. If there are bridges, remove them using solder wick, but try to keep the heat to a minimum.

FTDI Programming.

Please visit the FTDI web site to download the driver for the FTDI USB chip.

http://www.ftdichip.com/Drivers/D2XX.htm

Before mounting the PIC chip, the FTDI chip needs to be programmed by using a utility from the FTDI web site. It is downloadable from the following link.

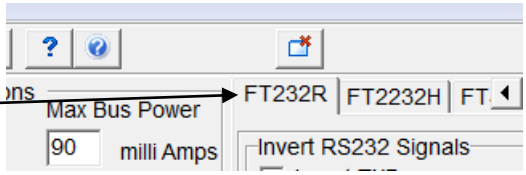
http://www.ftdichip.com/Support/Utilities.htm#MProg

Caution: Check all of your work before continuing, especially the USB connector, FTDI chip, and C7.

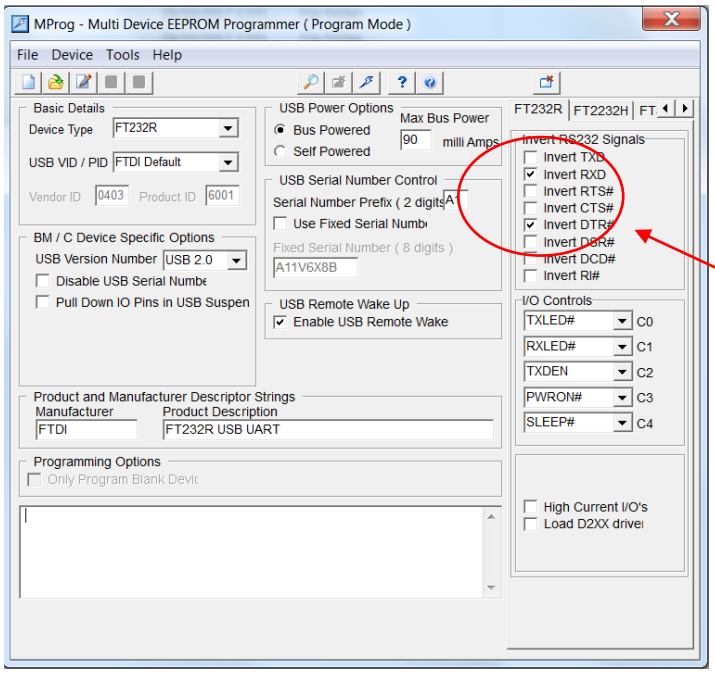
After installing the FTDI driver, connect the PCB to the PC USB port using a suitable cable. If all is well, Windows should recognise the FDTI chip. If Windows reports an error, then unplug and check your construction work.

Extract the files from the MProg download and run the MPROG.exe utility program.

Click [Tools] [Read and Parse]
The software should configure itself to the connected FTDI chip and display FT232R.

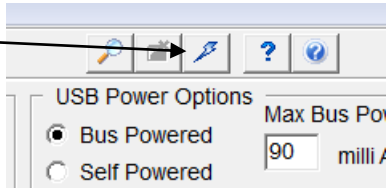


Click [File][Load]
Navigate to the cce33 installation directory and look in the PIC directory.
You should see a file called [calculator.ept]
Select and load this file.
The MPROG screen should look like this. The serial number may not match but that is ok.



Important
Both must be checked

Click the program button to program the FTDI chip.



Close the MPROG program.

Next, install the CAT24M01 serial EEPROM chip. Pin 1 is towards the PIC chip. The pins are spaced far enough apart for easy soldering

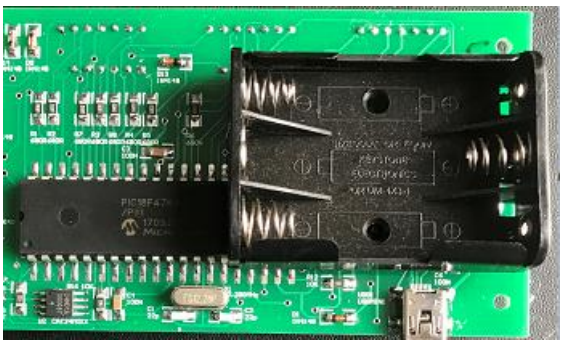
Installing the PIC chip.

The PIC chip needs to have the pins carefully trimmed where they narrow, or they can be bend inwards and lay flush under the chip against the bottom of the plastic. Pin 1 is close to the serial EEPROM chip. Solder one corner pin and check position and orientation and if all is correct, solder a pin on the opposite corner, recheck, then solder the other pins.

The 12.288MHz crystal is mounted on the bottom of the PCB with the pins soldered and trimmed from the top side. Make sure the metal case is not touching the two 22n capacitors next to it. Push the crystal fully down to the PCB, then if you push the crystal gently sideways and solder both pins while in this position, then when you straighten it, it will sit about 0.5mm above the PCB.

Caution: Check all of your work before continuing.

If you desire to use the battery pack, insert he leads for the pack so that the battery pack sits central to the PCB and on top of the PIC processor. Solder the leads to the PCB terminals from the top side.



Make sure the top switch is in the left position and the bottom switch is in the right position. Insert the batteries and switch the top switch to either the middle or right positions. The display should light up as in the diagram. The default calculator will be the HP-35.

If the display did not light up like this, then switch off, remove the batteries and check all of your work, especially for dry joints and components that may have been soldered in backwards.

Installing the emulator software

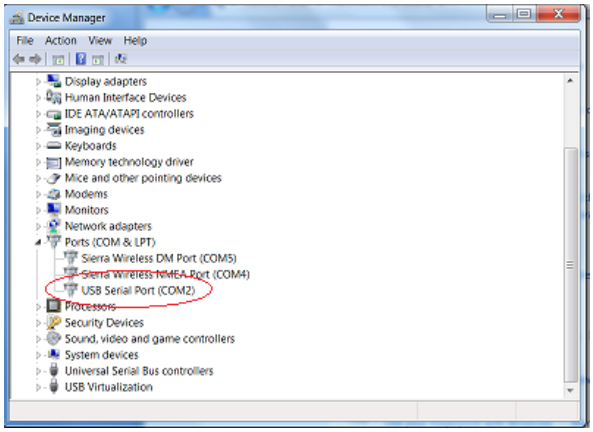
The **cce33.zip** file includes all files for the project. All required calculator modules must be downloaded and unzipped into the same directory if you want to try them on the PC emulator. A CCE33 shortcut should be created on the PC desktop as well. The following procedures assume that some of the calculator modules are installed.

Communications

The calculator PCB communicates with the PC calculator program via a USB port and a general purpose USB cable which will have to be purchased.

Connect the calculator board to a free PC USB port and the board should work but if there are problems visit www.ftdichip.com/ for more details on their drivers.

You need to know the COM port being used for the device and this can be accessed from the Control Panel – Hardware and Sound – Device Manager – Ports (COM & LPT) – USB Serial Port (COMx) where **x** should be set to a value between **1** and **9**. To do this, double click the USB Serial Port entry to open the Property Viewer, click the Port Settings Tab and then click Advanced. You will see a COM Port drop down list where you can set the port number to a value described above.



Note: It is important to make sure the FTDI board is connected to the PC and working **before** opening or closing the programmer screen from the cce33.exe emulator program or the FTDI driver seems to get confused and the PC may hang. The actual calculator can be switched on or off as desired.

Note: The slide power switch does not remove power from the PCB. The PIC will detect that the switch is in the OFF position and will stop executing code. This method is used so that the Continuous Memory models can be emulated.

Run the **cce33.exe** program and the HP-35 calculator should be displayed on the PC screen (if it was installed). Other calculators might appear depending on what modules are installed. Move the mouse over the calculator keyboard and right click. In the menu that pops up, click the item [For the inquisitive...]. Pressing F12 does the same task.

In the debugger screen that opens, click the Programmer Button,  (or press F6).

If you see a "Serial port not active" message then the USB/serial port driver is not functioning properly. Make sure the displayed COM port value, near the screen bottom left, matches the one that was set previously from the Device Manager (1 – 9). If not, click the [COM Port] radio button and enter the same port value that was set earlier and press Enter. If nothing happens, click the [Reset] button to try and establish communications with the calculator.

If all is well, you should see a message appear similar to below.

```
Calculator info query...
Current Model         = HP46
Module Type           = MultiCalc
Battery Monitor Level = Default
PIC Code Version      = 43180 (Current)
```

If none of this happens, go back and recheck your work. Make sure the calculator power switch is ON. Assuming nothing is damaged, the PIC chip may need verifying or reprogramming so you will need a PIC ICSP programmer to accomplish this task.

If everything is ok so far, make sure the [Info] radio button is selected and then click the [Read] button. The same message should appear except for the success message.

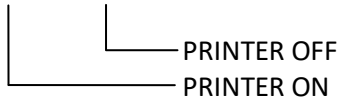
Click the [Calculator Model] radio button and then [Write] button. Select any model you like and click OK. This will now be the calculator's operational model.

If you get an "Access Error" displayed on the calculator when you try to store or recall program data, then the PIC chip is not communicating with the serial EEPROM chip.

Check to make sure that the 2 links are in the correct position for normal operation.

Calculator Switches

Model	Top			Bottom			
HP55	OFF	ON	ON	Timer	Prgm	Run	
HP65	OFF	ON	ON	Run	Prgm	Run	
HP67	OFF	ON	ON	Run	Prgm	Run	
HP21	OFF	ON	ON	RAD	DEG	RAD	
HP22	OFF	ON	ON	END	BEGIN	END	
HP25/C	OFF	ON	ON	RUN	PRGM	RUN	
HP27	OFF	ON	ON	N/A	N/A	N/A	
HP29C	OFF	ON	ON	RUN	PRGM	RUN	
HP25/C	OFF	ON	ON	RUN	PRGM	RUN	
HP33E/C	OFF	ON	ON	RUN	PRGM	RUN	
HP34C	OFF	ON	ON	RUN	PRGM	RUN	
HP37E	OFF	ON	ON	END	BEGIN	END	
HP38E/C	OFF	ON	ON	M.DY	D.MY	M.DY	
HP10	OFF	ON	ON	ON	ON	OFF	(DISPLAY)
HP91	OFF	ON	ON	MAN	ALL	NORM	(+ MENU – DEG GRD RAD)
HP19C	OFF	PRGM	RUN	MAN	TRACE	NORM	
HP97	OFF	PRGM	RUN	MAN	TRACE	NORM	
HP46	OFF	ON	ON	N/A	N/A	N/A	



Project Parts List

Part	Used	Part Type	Designators
1	35		SW1 SW2 SW3 SW4 SW5 SW6 SW7 SW8 SW9 SW10 SW11 SW12 SW13 SW14 SW15 SW16 SW17 SW18 SW19 SW20 SW21 SW22 SW23 SW24 SW25 SW26 SW27 SW28 SW29 SW30 SW31 SW32 SW33 SW34 SW35
2	12	1N4148	D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D13 D14
3	1	3K9	R20
4	1	4.7uF	C7
5	10	10K	R9 R10 R11 R12 R13 R14 R15 R16 R17 R19
6	1	12.288MHz	X1
7	2	22p	C1 C2
8	4	100N	C3 C4 C5 C6
9	8	680R	R1 R2 R3 R4 R5 R6 R7 R8
10	2	BAT54W	D11 D12
11	1	BC817	TR1
12	1	BT+	TM1
13	1	BT-	TM2
14	1	CAT24M01X	U2
15	1	FT232RL	U3
16	1	ISCP	CN1
17	3	LED Module	DSP1 DSP2 DSP3
18	1	OFF/PRG/RUN	SW37
19	1	PIC18F47K40	U1
20	1	TMR PRGM RUN	SW36

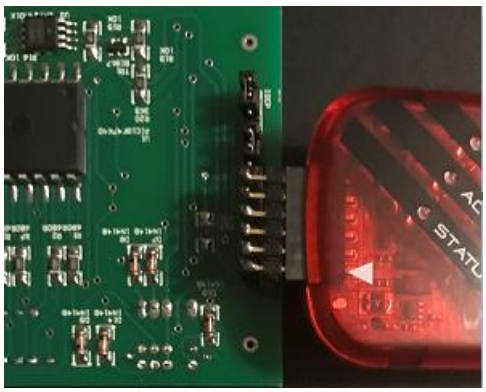
Reprogramming the PIC via ICSP.

Caution: The components used in this project may be damaged by static electricity. Handle the PCB only by the edges.

Connector CN1 is used for In Circuit Serial Programming (ICSP) for the PIC chip.

Remove the batteries and disconnect the USB cable if fitted.

When ICSP is required for reprogramming the PIC, the two links should be moved so that they connect pins (1 and 2) and pins (4 and 5).



Pins 6 to 12 have been configured to accept a Microchip® PICkit™ 3 serial programmer. Pin 1 of the PICkit3 connects to pin 12 of the connector as shown.

If you are using Microchip's IPE programmer interface, then there is an Environment file called `calculator.pen` available in the PIC directory which can be from menu item [File -> Import -> Environment]. This will setup IPE for reprogramming the PIC. Use the Browse button to load the HEX file for reprogramming the PIC chip. It is located in your software installation directory in the PIC sub directory. It is called `PICcalcM.hex`.

When PIC programming is completed, remove the PICkit3 and replace the links so that they connect pins (2 and 3) and (5 and 6).

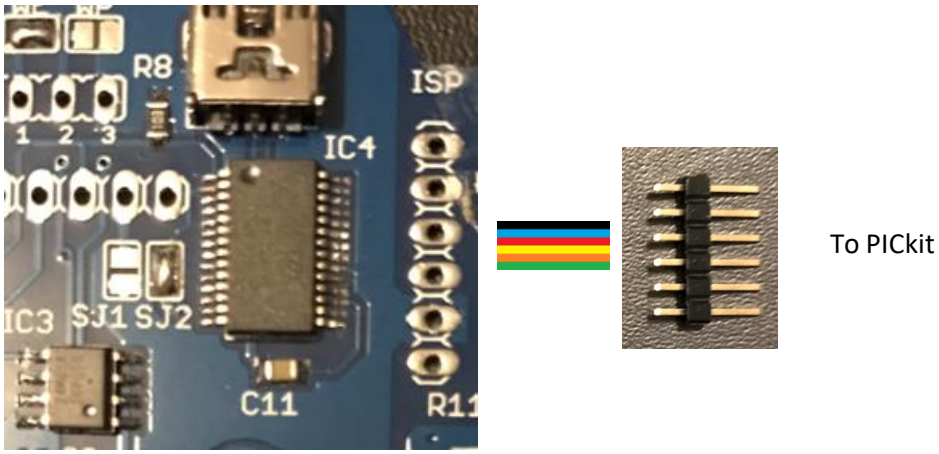
Haralds Classic Controller

To ICSP reprogram with a PICkit3, this module needs the SJ2 link be opened and this will require a soldering iron.

Carefully remove the module from the calculator. A small screwdriver used as a lever alternating at the corners of the PCB near the bottom connectors works well.

If you have some Solder-Wick (preferred) just blot the solder up which will break the link, otherwise you will need to try and draw the solder away using the clean tip of the soldering iron. There are also some specialised desoldering tools that will also do the job if these are handy. The SJ2 link should look like the SJ1 link next to it.

Note: The most important thing to remember is that applying too much heat may lift the solder pads away from the PCB surface and may render the module unusable.



A 6 way flat ribbon cable (about 25mm in length) needs to be soldered to the 6 pin ICP connector. A 6 pin IDC connector can be soldered to the other ends of the wires so that it can be inserted into the PICkit. Once fitted, the wiring can stay on the PCB as long as it is insulated from the PCB. This is only a suggestion and other methods may be suitable as long as the case can be reassembled.

Once assembled, the PICkit3 can be connected and programming can be done according to the Microchip documentation.

The file for reprogramming the PIC chip is located in your software installation directory in the PIC sub directory and is called `PICcalcH.hex`.

After programming, re-solder the shorting link for SJ2 and reassemble the board and calculator.

USB Programming.

The calculator may be switched off for the following procedure.

No PCB changes are required.

The USB cable can be plugged in to the top of the PCB and connected to a PC USB port.. Then run the CCE33.exe emulator program and open the Programmer Interface window. You should see a connected message and if reprogramming is required, information will also appear on screen.

Change the USB port if no connection is made.

FTDI Driver

The calculator project communicates to a PC via the USB port. To do this it needs a special driver to be installed. This can be downloaded from the FTDI web site.

http://www.ftdichip.com/Drivers/D2XX.htm

Please install the correct driver for your system.

USB Communications

The calculator communicates to the PC via the USB port and suitable cable.

The COM Port number must be known before communications can take place.

See Communications to set his up.

Batteries

The batteries used for this project are 3 x AAA 1.5 volt cells.

IMPORTANT: The single cell voltage should not exceed 1.5 volts
The total battery voltage should not exceed 4.5 volts.
Do not use rechargeable batteries if the voltages exceed those listed above.

A 3.7V LiPo battery may also be used if desired, but will need to have its own recharger. The battery should be disconnected from the calculator prior to charging.

Additional Program Steps – CCE33 Emulator

Models affected: HP-19C, HP-25(C), HP29C, HP-34C

The HP-19C and HP29C emulators have additional key functions, most of which can be used in normal run mode, and all can be entered as a program step which will execute in a running program.

The new functions use the following key sequences and are treated as a merged key code and are stored as a single program step.

		Program Text/Print	Key Display
[f][f][0]	Random Number	RAND	16 16 62
[f][f][1]	Constants	CONR	16 16 52
[f][f][2]	Beep	BEEP	16 16 53
[f][f][4]	Data Swap	DATS	16 16 54
[f][f][3]	File access	FILE	16 16 42

		Run Mode	Running Program
Random Number	[f][f][0]	Yes	Yes

A random number will be placed into the X register.
The returned number has a value between 0 and 1.

Example: Generate a random number between 0 and 9.
Press [f][f][0] then [10] [x] [f] [INT].

		Run Mode	Running Program
Constants	[f][f][1]	Yes	Yes

Up to 10 Constants can be accessed and are stored in a special memory area.

To save a constant
Enter constant, then press ENTER to place it into the Y register
Press 1, followed by the Constant address 0 – 9
Press [f][f][1]
The constant will be saved into the selected constant memory
Example: Save Y into Constant 1 [1][1] [f][f][1]

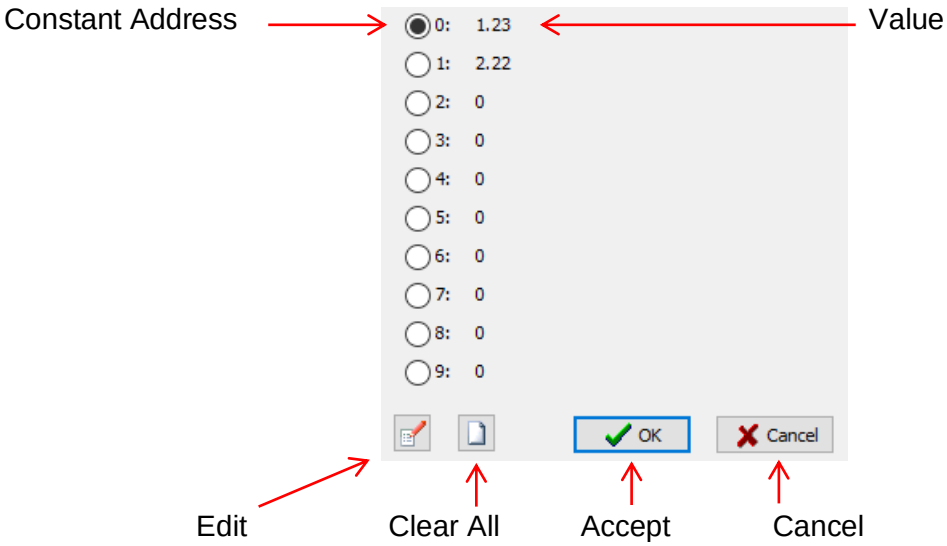
To recall a constant
Enter 2, followed by the Constant address 0 – 9
Press [f][f][1]
The recalled constant will be placed into the X register.
Example: Recall Constant 7 [2][7] [f][f][1]

Recall from I register
Enter 3
Press [f][f][1]
The constant address come from Mantissa digit [1] in the I register (Memory location 0)
Example:
I = **1.**23 Constant 1 recalled
I = **2**345.99 Constant 2 recalled
I = -**9.**345 E+99 Constant 9 recalled

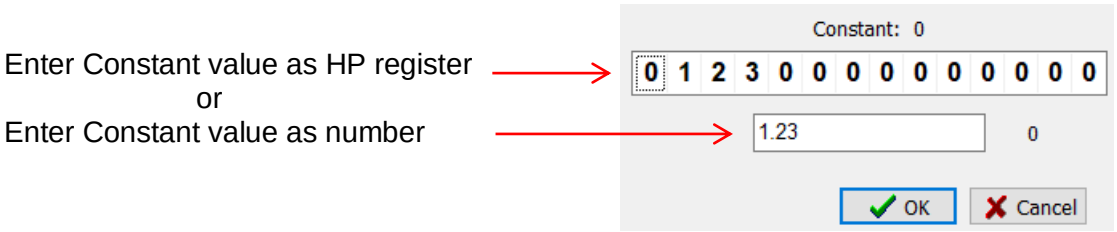
The same key sequences can be entered in a program to access constants during a running program. Constants can also be modified and stored via the emulator program.

To access the Constants from the emulator, select the [Options] HP-19C Constants menu item.

The constants will then be loaded and displayed, and after editing, saved again if the [Accept] button is pressed, otherwise no changes take place,



The [Edit] button opens another screen to set the Constant value. Select the Constant to be changed prior to pressing the button.



		Run Mode	Running Program
Beep	[f][f][2]	Yes	Yes

Regardless of whether it is turned on or off, the beeper will sound briefly when this function is executed. The volume will be as set, either soft or loud.

		Run Mode	Running Program
Data Swap	[f][f][3]	Yes	Yes

This function will swap the Primary and Secondary registers from a separate RAM storage buffer. This effectively gives you twice the RAM number of registers available, 60 instead of 30.

The storage buffer is saved and recalled as part of the Continuous Memory.

		Run Mode	Running Program
File Access	[f][f][4]	No	Yes

This function allows access from program or data files stored on disk. These files must be in the `MultiCalc` install directory and a subdirectory called `\execFiles`. 19C and 29C files will be loaded from additional subdirectories `\files19C` and `\files29C` respectively. These directories are included in the `MultiCalc.zip` file and contain a few example `.pgf` files.

File addresses are made up of a 3 digit number giving a maximum of 360 or 400 files to access.

Block Numbers	0 – 9	HP-29C	0 – 8.
File Numbers	00 – 39		

Files can contain either Program or Data. Program Files contain the 98 steps, and data files contain the 16 primary memory registers `R0` to `R.5`. When you use the Save Button in HP-19C emulator mode, you will be asked which file type to save.

To access a file, the last 3 characters of the file name before the `.pgf` extension must be the 3 file address digits. Two program test files and one data test file are included in the install subdirectory. Multiple files may have the same 3 digits, the first one found matching this data will be loaded.

Examples:	`Prgm19C_000.pgf`	Test program file which loads "`Test19C_001`"
	`Test19C_001.pgf`	Program File
	`Data19C_002.pgf`	Data file

The same files are included in the 29C subdirectory.

The required data for these functions can either be *assumed* to save keystrokes or *explicit*. The least amount of *assumed* keystrokes is the Prefix only followed by `[f][f][3]`. An *explicit* instruction requires the maximum key strokes for a full data entry followed by `[f][f][4]`.

This function has 4 modes which are accessed by a Prefix number.

Prefix	Function	Required Data
1	Load Program File	File - Block, Number, LBL Number
2	Load Program File	File - Block, Number (program starts at Step 1)
3	Load Data File	File - Block, Number
4	Save Data File	File - Block, Number

Prefix 1 Load a 98 step program file and have it start from a LBL.

File Address		LBL
Block	Number	
0 – 9	00 – 39	0 – 9

Data Type	File Keystrokes	Block	Number	LBL Start
Assumed	None	0	00	LBL 0
Assumed	LBL	0	00	LBL n
Assumed	NumU, LBL	0	0 NumU	LBL n
Assumed	NumT NumU LBL	0	NumT NumU	LBL n
Explicit	Blk NumU NumT	Blk	NumT NumU	LBL n

Examples:	Key Sequence
`Block[0] Number[00] LBL0`	`[1 [f][f][4]`
`Block[0] Number[00] LBL9`	`[1] [9] [f][f][4]`
`Block[0] Number[09] LBL2`	`[1] [9] [2] [f][f][4]`
`Block[5] Number[29] LBL7`	`[1] [5] [2] [9] [7] [f][f][4]`

Prefix 2 Load a 98 step program file and have it start from Step 1.

File Address	
Block	Number
0 – 9	00 – 39

Data Type	File Keystrokes	Block	Number	Start
Assumed	None	0	00	Step 1
Assumed	LBL	0	00	Step 1
Assumed	NumU, LBL	0	0 NumU	Step 1
Assumed	NumT NumU LBL	0	NumT NumU	Step 1
Explicit	Blk NumU NumT	Blk	NumT NumU	Step 1

Examples:	Key Sequence
`Block[0] Number[00]`	`[2] [f][f][4]`
`Block[0] Number[03]`	`[2] [3] [f][f][4]`
`Block[0] Number[19]`	`[2] [1] [9] [f][f][4]`
`Block[3] Number[29]`	`[2] [3] [2] [9] [f][f][4]`

Prefix 3 Load Data File containing the 16 Primary registers `0` to `.5`.

File Address
Block Number
0 – 9 00 – 39

Data Type	File Keystrokes	Block	Number
Assumed	None	0	00
Assumed	NumU, LBL	0	0 NumU
Assumed	NumT NumU LBL	0	NumT NumU
Explicit	Blk NumU NumT	Blk	NumT NumU

Examples:	Key Sequence
`Block[0] Number[00]`	`[3] [f][f][4]`
`Block[0] Number[03]`	`[3] [3] [f][f][4]`
`Block[0] Number[19]`	`[3] [1] [9] [f][f][4]`
`Block[3] Number[27]`	`[3] [3] [2] [7] [f][f][4]`

Prefix 4 Save Data File containing the 16 primary registers `0` to `.5`.

File Address
Block Number
0 – 9 00 – 39

Data Type	File Keystrokes	Block	Number
Assumed	None	0	00
Assumed	NumU, LBL	0	0 NumU
Assumed	NumT NumU LBL	0	NumT NumU
Explicit	Blk NumU NumT	Blk	NumT NumU

Examples:	Key Sequence
`Block[0] Number[00]`	`[4] [f][f][4]`
`Block[0] Number[07]`	`[4] [7] [f][f][4]`
`Block[0] Number[39]`	`[4] [3] [9] [f][f][4]`
`Block[3] Number[14]`	`[4] [3] [1] [4] [f][f][4]`

FILE Errors

Errors may occur when using the FILE instruction if, for example, you specify an invalid file address.

Example: Save Data File at: `Block[3] Number[47]`
`[4] [3] [4] [7] [f][f][4]`

This will generate an *Error 1* because the `Number` value can only be between `00` and `39`.

Errors are generated the same as for the normal HP-19C operating code, but FILE errors will have a number shown in the display similar to Spice calculators.

Error	Cause
1	Invalid File Address (not 0 – 9, and 00 – 39)
2	Trying to load a file from a vacant address

3	Loading a Program File, but File is Data
4	Loading a Data File, but File is Program
5	Invalid prefix code (not 1 – 4)
6	Trying to execute `[f][f][4]` from keyboard
7	File did not save
8	File did not load
9	File had errors

HP-25(C)

The HP-25 and HP-25C have 3 extra program steps.

GSB 49	`[g] [GTO]`
RTN	`[g] [BST]`
BEEP	`[g] [SST]`

Some Woodstock's have GTO nn and or GSB nn program steps that allow jumps to the Label nn. The HP-25 did not have the ROM space or program step space to allow GSB nn, and with only 49 steps, it was probably not necessary.

To be able to make a versatile GSB command for the HP-25, GSB 49 was added. After this program step executes, the next is fetched from step 49. If a GTO nn step is placed there, then the GSB has access to most of the available program space.

RTN is used to return from the subroutine called by GSB 49. If it is used without a matching GSB 49, the program will stop running.

Only one subroutine can be active at any time. Additional GSB 49 steps will overwrite the RTN address.

BEEP sounds a beeper once.

This can be used in RUN mode as well.

Example: Add PI to the X register and beep

```
Step  Code          Keys
.1    GSB 49        [g] [GTO]
.2    BEEP          [g] [SST]
.2    R/S           R/S

.40   PI            [g] [.]
.41   +             [+]
.42   RTN           [g] [BST]

.49   GTO 40        GTO [4][0]
```

Test:

```
[5] [R/S]  X = 8.14
[R/S]      X = 11.28
```

HP-34C

The HP-34C has 2 new program steps.

Random
Beep

Both of these are accessed from the key combination `[g][0]`.

Random numbers are >= 0 and < 1.

As there is only one spare program code available for the HP-34C, this new code shares both functions.

If the C register has [1] as the first digit, the Beeper will sound. Any other first digit will place a random number into the C register.

Examples:

0.234	`[f][0]`	Random
1.454	`[f][0]`	Beep
95.034	`[f][0]`	Random

These new functions can be used in a program.

The code will be displayed as:

Normal	**15 0**
Text Mode	**g FuncN**

An example of sounding the beeper

```
[1]
[f][0]
```

An example of creating a random number

```
[0]
[f][0]
```

Project Notes

This project was put together using freely available documentation from the web. Some of these documents proved to be unreliable and others were nonexistent so in these cases a best guess approach was used.

You may notice that a key press, power on/off, TMR or PRG modes affect certain Status register bits in different calculators. There is a logic arrangement in the original calculator hardware that overrides these bits depending on the switch positions. For example s0 is set to 1 in the HP-45 when a key is pressed. Even if you explicitly set this bit to 0 when a key is down, the bit will still stay = 1.

HP45

This calculator has an undocumented timer mode, and to enable it you had to press RCL and then keys CHS, 7 and 8 simultaneously. The ENTER key on the original calculator could be easily modified from inside and doing so makes access to the timer mode less complicated. With the modification you just press RCL then ENTER to do the same thing. Simultaneously pressing CHS, 7 and 8 produces a hardware key code of [60]. The hardware modification makes the ENTER key produce the same key code of [60] instead of key code [62]. The original calculator code was written to deal with both values. For simplicity, this project uses the RCL - ENTER method and the ENTER key code [60] is used for the HP45 emulation.

HP55

The available code listing for this calculator had quite a few irregularities and required some modifications to get it working.

HP65

The available code listing for this calculator had irregularities as well and required some modifications to get it working.

HP80 and HP80 Patent

The HP80 code appears to be released with a bug explained on page 68 of the HP80 Owner's Manual. The HP80 Patent code is the code as listed in the patent document and appears to have this problem fixed.

SPICE SERIES – HP31E etc

The ROM self check procedures have been modified because they were set up to check the original calculator code. As it is possible the user may modify this code, the ROM checks may produce errors. The [`rom checksum`] instruction always returns a no fault found result.

HP10

There is no Microcode available for this calculator, so the code was written by the author and performs as per the contents of the HP10 owner manual.

ENJOY