The new ACT

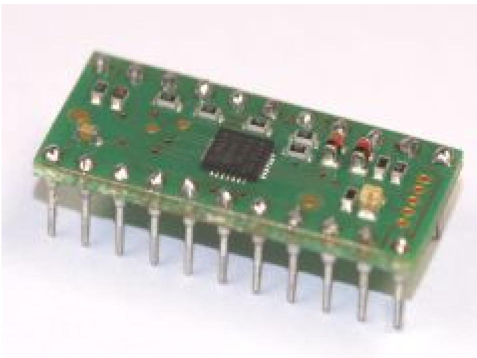
Arithmetic Control Timing

a new generation PMOS/NMOS circuit

Repair Kit for HP-21, HP-22, HP-25, HP-25C, HP-27, HP-29C, HP-67 calculators

or

Upgrade Kit to HP-21E, HP-25E, HP-29E, HP-34E, HP-67E infrared printing calculators



Bernhard Emese , July 2023, Rev. 1.18

Contents

ACT Manual Revisions

Rev 1.00 Jan 25th 2015 initial release

Rev 1.01 Feb 1st 2015 added welcome string and hexadecimal, octal conversion, schematic

Rev 1.02 Feb 17th 2015 HP-27 added, new image of inserted ACT

Rev 1.03 Apr 7th 2015 chapter for HP-21, HP-22, HP-27, HP-29C added

Rev 1.04 May 17th Infrared printing added.

Rev 1.05 Oct 21st HP-34E, HP-67E, Classics, ACT Flash Update added

Rev 1.06 Nov 20th Vinyl Overlays added

Rev 1.07 Mar 25th 2016 GPS added, Apr 17th, GPS program updated

Rev 1.08 May 20th HP-01 added

Rev 1.09 Dec 1st Flags 10-15 added, Continuous Memory saves also stack registers

Rev 1.10 Jan 27th 2017 Program Overlay, load programs dynamically

Rev 1.11 May 9th 2017 Placing the IR diode in HP-25 and HP-67

Rev 1.12 July 31st 2017 Multitasking, Flicker Supression, Memory usage, Voltage, Temperature

Rev 1.13 Dec 19th 2017 HP-65

Rev 1.14 Apr 6th 2018 HP-19C

Rev 1.18 July 11th 2023 Version 1.18 New PIC Processor

Introduction

Congratulations! There are only a few steps to go, and you will be able to give your HP calculator a new life. After more than 35 years without having a replacement, there has finally arrived the new ACT chip, which you are holding in your hands.

As you might already know, ACT is the abbreviation for „Arithmetic Control Timing“. This chip was used throughout the second generation of HP calculators, the “Woodstocks”, like in the HP-21, HP-22, HP-25, HP25C, HP-27, HP-19C/HP-29C, and later in HP-67, HP-91 and HP-97 as well. Hence Hewlett-Packard was very innovative at that times, the models differ significantly, even more inside, than can be assumed from the outside. There were two major changes made in silicon techniques between the first and the last models, starting with PMOS (P-channel Metal Oxide Semiconductor). The PMOS design of the HP-21, HP-22, HP-25 used P-doped silicon, the signal levels were different to our modern mainly N-doped silicon chips, they used negative voltages for signals and the threshold voltages were above the normal TTL logic levels. With the advent of the HP-25C, another technique came into life: CMOS (Complementary Metal Oxide Semiconductor). Although the rest of the circuit remained PMOS, the RAM chips were made of the new CMOS material. Soon came another change, introduced with the HP-27 hardware, HP used NMOS (N-channel Metal Oxide Semiconductor) silicon, which needs the more common voltages that we use in our modern circuits, however today with lower voltages.

Meanwhile the new ACT can handle all these calculators, only the HP-27 needs a slightly modified hardware version of the new ACT, because it has NMOS voltage levels. And HP-10C, HP-19C and HP-97 are not included, because still some details about the original ACT chip are not yet known.

The new ACT does work well in HP-21, HP-22, HP-25, HP-25C, HP-27, HP-29C and HP-67 calculators.

Contributions

Undoubtedly, the *“new ACT”* could not have been built without the previous work of *Eric Smith*, who, over many years, explored these calculators and found out how they worked, he collected precious knowledge about ROMs, RAMs and of course the ACTs. Thus he wrote the first ACT emulator „nonpareil“ for HP calculators and put it into the public domain.

Emulators try to perform exactly the same machine instructions as the original chip did. They execute the original program of the calculator and therefore show the same results on the screen; the *“new ACT”* is such an emulator. It is based on *Eric's* pioneer work. Although it must do some very tricky additional tasks in real time, to establish communication with the other components inside, and maintains a stable keyboard scan and display timing, which was never done before, the arithmetic unit remains the heart of the machine.

Many, many thanks and appreciation to *Eric Smith* for his really great pioneer work!

There are many others to mention and give thanks, that have made previous reengineering research and found out hardware details, that were helpful in understanding all these undocumented things. I want to mention another name, whom I don't know personally, *Jaques Laporte*, who gave his excellent website with hardware, firmware and mathematics analysis of the “Classic” and the “Woodstock” calculators to the public.

Of course, I shouldn't forget to mention the “*Museum of HP Calculators”* as the central source of anything related to HP calculators. It's the “Navel of the HP world”, the center web with all thinkable HP stuff caught over the years. If you want to find out something about HP calculators, you will find it here. Possible you also found the *“new ACT“* somewhere here.

The idea of the “*new ACT”*

I remember that I made a posting in the MoHPC forum in 2003 about my HP-25, asking if anybody could help me repairing it. Despite I didn't got any response, I could repair my HP-25, because there were only some contact problems. And I enjoyed switching it on from time to time. But many years later, only some months ago, in October 2014, I took it from the drawer again and plugged in the power supply for charging. This time I was not careful enough and made the great mistake of switching on the calculator while charging. The batteries were completely empty and didn't buffer the voltages and, to my great horror, the calculator was dead, all chips inside were burned by overvoltage, there was nothing I could do.

Its well known, that many ACT chips of the “Woodstock” series HP calculators were damaged while using its power supply without batteries, due to a design flaw of the original power switching unit.

Not only this one, but my HP-21 was gone as well, because I charged them together. I could not believe what I had done, and thought about what to do. Of course, I could buy another ebay calculator, but this would not be “mine”. As a hardware developer, I thought about the possibilities, and there seemed plenty fixes: replacing the hardware by putting a cheap old LED calculator board inside, developing a complete new board with modern calculator chips and scanning the keyboard and multiplexing the original display. Or at last, buying a working HP-25, exchange the ACTs, making one dead for the life of the other?

I decided No, none of these were acceptable options.

Then, to my own surprise, I discovered the existing HP-25 emulators for IOS, Mac and PC. Most or all of them were based on the „nonpareil“ software. The idea to get a perfect imitation of the originally HP-25, even getting the same flickering of the display between calculations, was charming. So I went on with the idea to leave as much as possible intact, perhaps only one chip, the ACT, was defective, and RAM and ROM were still alive. I built a prototype, studied the available software, and within seven days I saw for the first time a running display on my HP-25. It turned out, that RAM and ROM of both calculators were still working, but I really didn't need them, so I designed the “*new ACT”* with internal RAM and ROM, so that even in those cases, when more chips had gotten damaged, it could bring the calculator back to life. There was still a lot of work ahead of me, and hundreds of oscilloscope and debugging hours were spent until finally the replacement now is a tiny small circuit, ready to be plugged in.

I hope, not only for me it was worth the effort, it should be worth it also for those who can afford a smile when these charming **0.00** LED digits appear again on their calculator, after a hopefully successful repair.

The Repair Set

The repair set consists of the “new ACT” circuit, the manual as .pdf, the laminated Quick Guide card, some solder, a piece of copper mesh wire and two ultra low profile socket strips.

What you need else is a small Phillips screw driver, tweezers, a soldering iron, a vacuum pump optional some flux means, and some moderate skill in soldering.

Disclaimer:

When using these instructions for repairing your calculator, or in any other case of using the new ACT for any purpose, you are aware, that you alone have the responsibility. Due to the limited availability of vintage calculators and their versatility, PANAMATIK was not able to predict any possible circumstances, which may occur. Therefore it does NOT take responsibility for any damage that can occur directly or indirectly from using these instructions or using the new ACT. We assure, that we tried to take care and took any precautions to provide you with all the information required for a successfully repair. Proceed only if you agree with these terms.

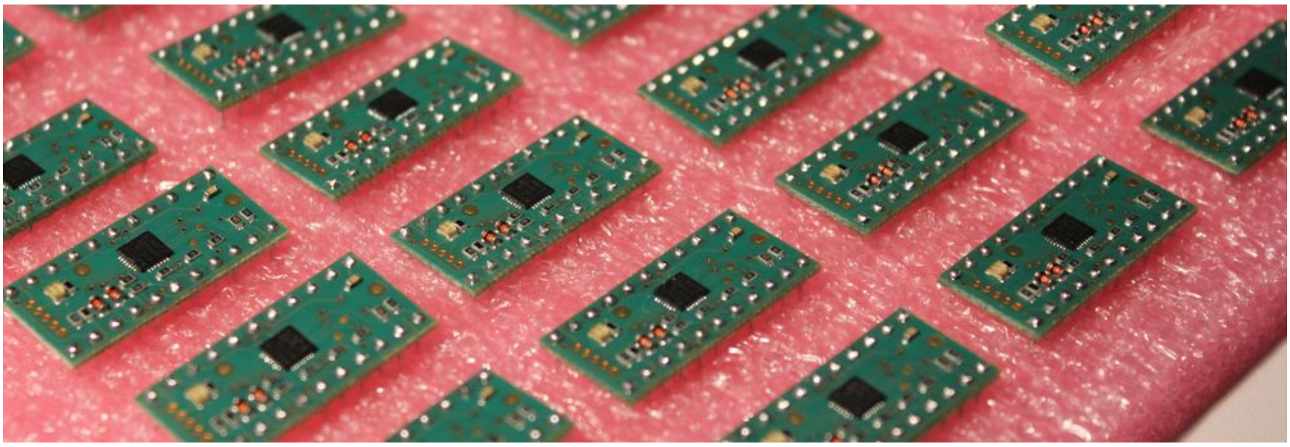
PANAMATIK

Replacement

What you have to ask yourself first: Do I need to replace the ACT Chip or not?

In most cases you need to replace the ACT when it is damaged by overvoltage, when you charged the calculator without any or with deeply discharged batteries. This causes the display showing only a chaos of irregular flashing numbers or it remains totally dark. If however your ACT is running, and you can use the calculator for calculations, but you are unable to enter program steps, seeing only **GTO 00** instructions and you can't use the storage registers, then your external RAM chips are damaged. Also in that case you can replace the ACT with the new ACT, because it uses internal RAM and everything will be fine[1]. As for another case, if your external ROM chip is defective, the original ACT will not have the chance to come up, but the new ACT will fix that too, because it is able to run the original HP-21, HP-22, HP-25, HP-27 or HP-29C code from its *“Internal ROM”*.

If there is another problem with your calculator like no battery power, corroded battery contacts, faulty Power Switching Unit - many other reasons other than the ACT could be responsible for your calculator not working any more - the new ACT would not be able to make your calculator live again. You must first eliminate all of the other repair possibilities, before you decide to pick up your soldering iron. If everything points to a defective ACT, then you should carefully read this manual and follow its instructions. It will describe as well as possible how to replace the ACT chip without damaging anything else that is still working. I hope this manual will give you the best advice, that I can form in words.

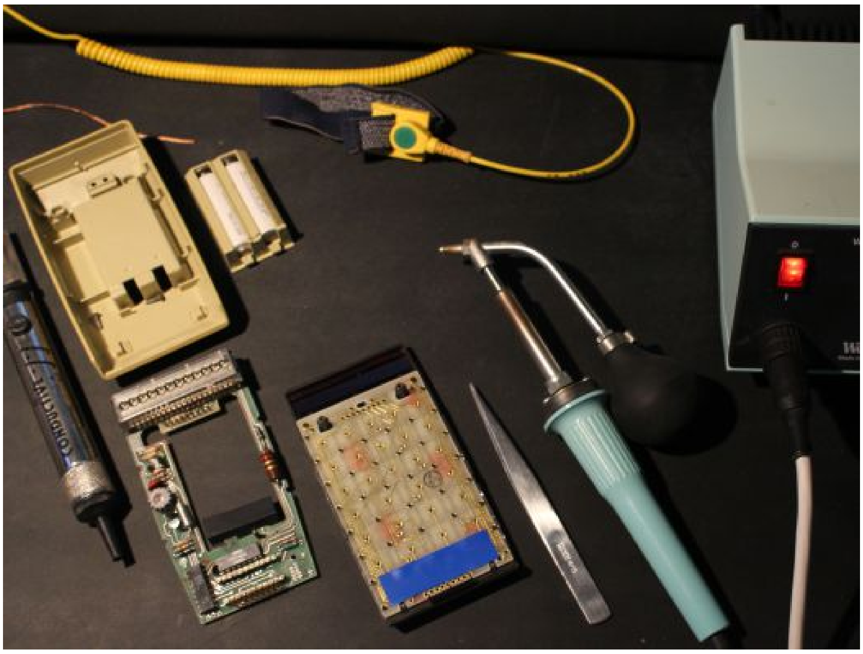


There will be no supply short of ACTs any more in the future.

[1] Keep the original ACT for some other calculator, that is not supported by the new ACT.

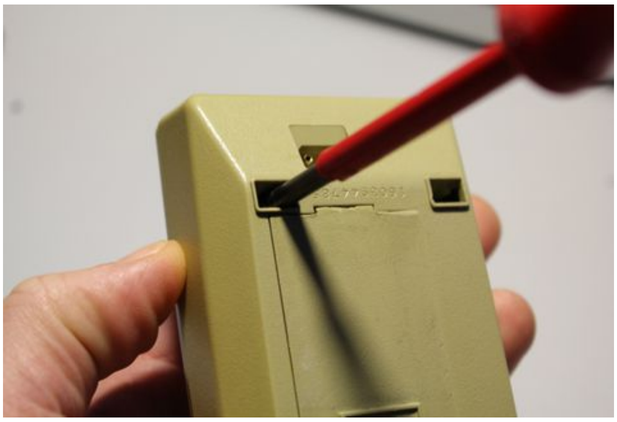
Preparations

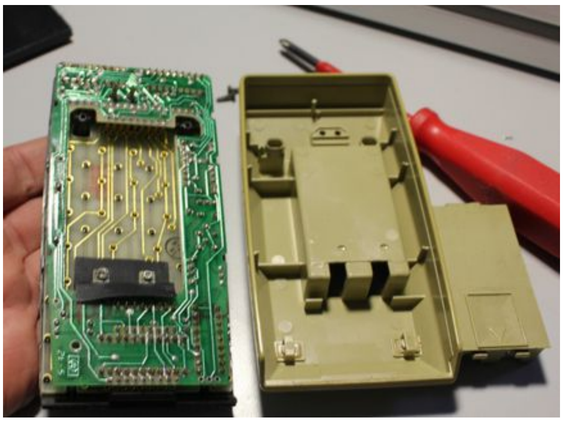
Do you need an electrostatic protection matte? Yes, if you have one. If not and if your room humidity is not too dry and hasn't electrostatic active carpets and materials, and you don't wear shirts which are electrostatically charged, you can go on without it. You are not going to damage more than is already damaged when your ACT is dead.



How to open the case?

There are only two screws to open, they are covered below the upper rubber feet near the display side of the calculator. There are a lot of descriptions to find, how this is done without hurting and making scratches into the rubber feet.





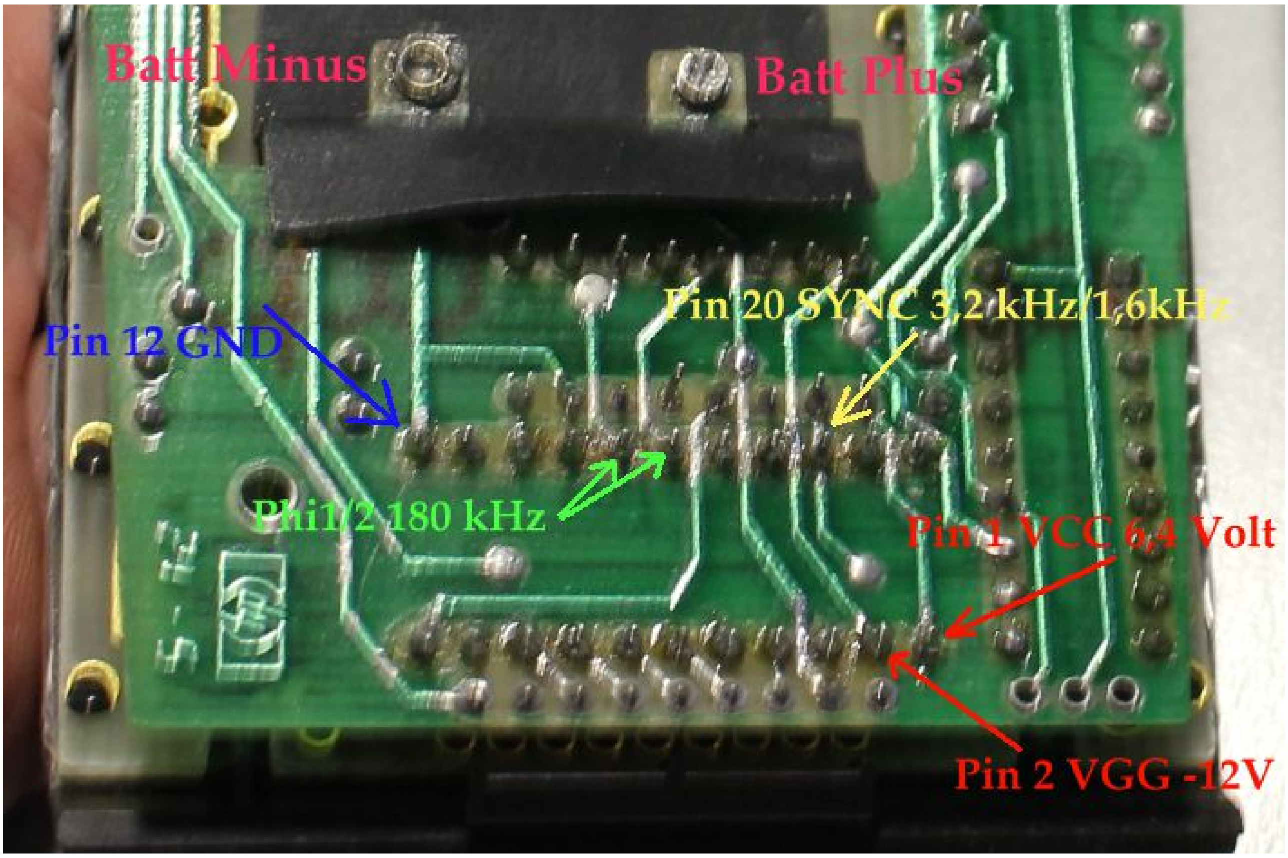
ACT Repair Kit Manual

After removing the battery case, push gently down the golden charger contacts with the screw driver, then guide the two battery contacts of the printed circuit board through the rectangular holes in the housing. Then while pressing with your thumb a little bit at the front side cover, where „Hewlett Packard“ is written, you will take both sides apart, the housing and the keyboard with the printed circuit boards.

Now it's time to measure some voltages. Note that the Voltages can only be measured when the calculator is switched ON and that can be done only when the keyboard is attached, so you have to measure from the bottom side, not the components side of the board.

The anode display driver, cathode display driver and power switch unit are assumedly OK.

Test the voltages at pin 1 VCC and pin 2 VGG of the ACT, they must be 6,4 Volt and -12 Volt relative to Ground GND Pin 12. I knew I had a defective ACT, which pulled the VCC voltage down, because of a short inside. If you measure significantly less than 5 Volt at Pin 1 of the ACT, then it is either defective or the power supply is not working correct. Measure again at the empty socket, when you have removed the ACT.

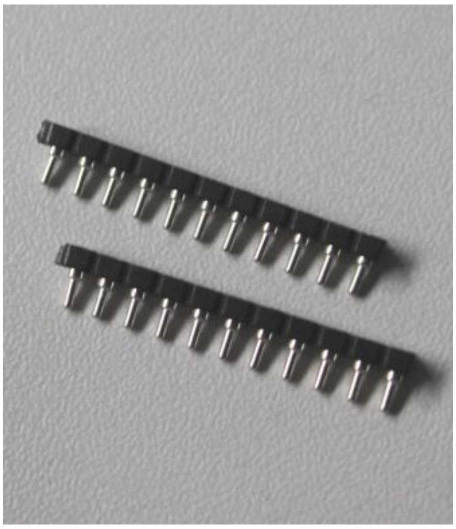


If you have an oscilloscope, measure also Pins 16,17 which should show a +6V to -12V phase shifted Phi1/Phi2 clock. If they are not running with 180kHz rate, the calculator cannot work. Also the SYNC must show a clock, but this is variable depending on the program executed, Normally its running at about 1,6 kHz, maximum 3,2 kHz.

How to unsolder ?

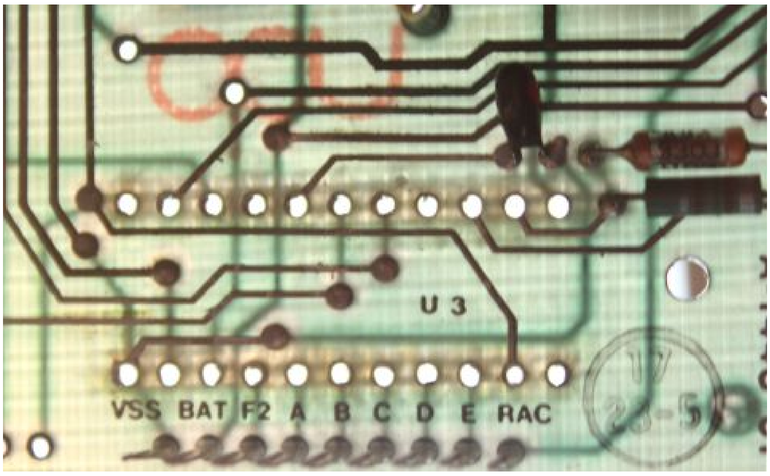
Now you need a unsoldering tool for doing the job. This can be either a manual vacuum pump or some more sophisticated tool. I do not recommend to cut the pins of the chip, which is an common method of removing defective chips, but destroys them. Even if you “just know”, that it is defective, you shouldn't do it, you could be wrong. Instead, take some time to remove all the tin from each pad and free them step by step from any remaining metal by using the copper strip that is part of the repair kit like a sponge. This could be done in 5-10 Minutes. If some pads refuse to get free, put some fresh tin inside, use the flux of any decent soldering tin, fill the hole and repeat pumping out the liquid tin again. Be careful not to damage the pads. And be careful not to solder too long, 5 seconds should be the maximum per pad to prevent overheating. Treat the chip as if it were still alive. Use tweezers to try if the pins is already freely movable in the pad holes and use them also to free each pin from small sticky tin surplus. If all pins are free, **carefully** lever the chip with a screw driver out of its pads. It's your responsibility to know, when to do that. Don't use too much force, it is a sign, that not all pins are yet free. The unsoldering is the most difficult part of the repair, and ideally you have some experience with it before.

I made good experience with, and would therefore recommend, an unsoldering tool like that on the image left with the rubber ball. But it is not mandatory. A hand vacuum pump like seen beneath can do the same job and you don't want to buy a complete electronic workshop just for one repair action.



To ensure that this will be the only difficult part and you will not have to do another unsoldering action, I provide you with ultra low profile socket strips as part of the repair kit. I recommend to use these strips instead of soldering the “new ACT” directly into the board. It gives safe contacts and you can replace it whenever you want by the original chip (if you have one), or another *“new ACT”.*

Luckily the pad holes of the old *“woodstock”* boards are big enough for the pins of the low profile socket strips, but you must carefully remove any tin residue from the pad holes, before you can put them in. I found the pad holes of HP-29C boards were smaller, but still big enough. If you have difficulties to insert the sockets you can also use normal profile socket strips. They will leave enough space though.

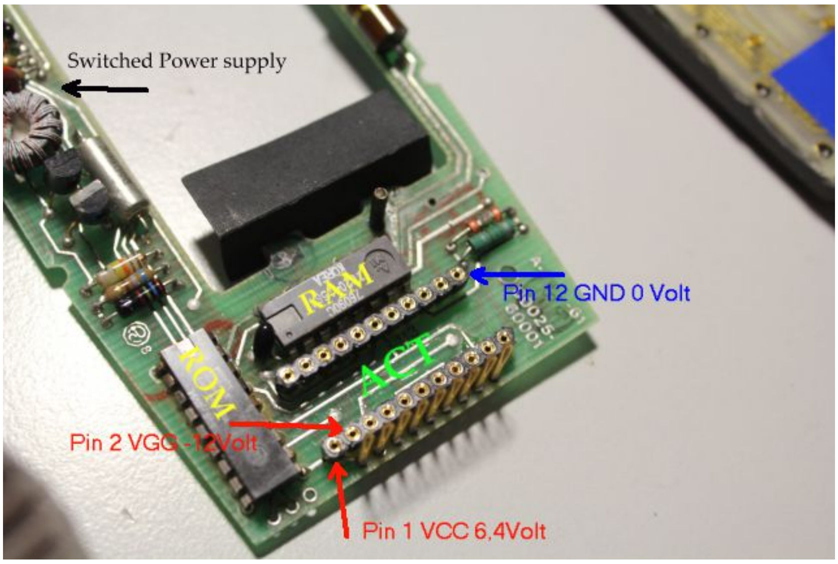


If your board looks like in this picture then you have done your job well. It should be easy now to insert the socket strips.

Attention!

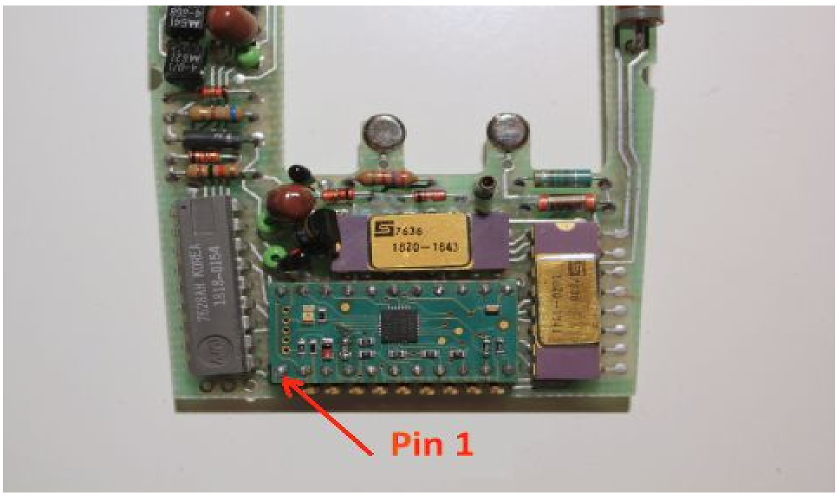
Take care, that the socket strips are inserted all the way in, until they touch the board, and are sitting straight! You will not have much space to waste.

Where to place the new ACT ?



This image shows an HP-25.

Be sure, that you place the ACT in the right direction with Pin 1 to the left into the socket as shown in the image below.



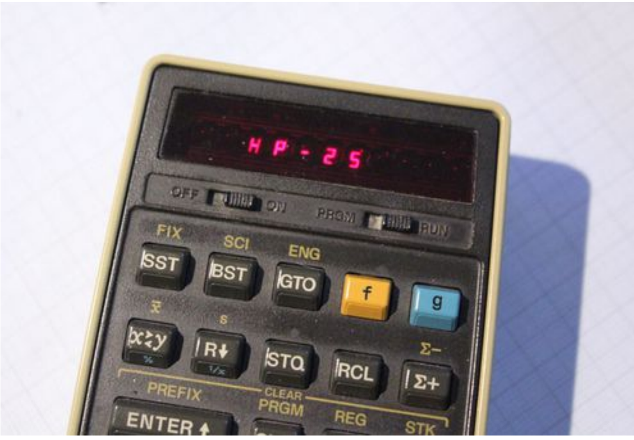
This image shows an HP-25C.

Here you can see how the new ACT is inserted. All woodstock models have the same chip orientation.

Press the ACT circuit carefully into the socket strips.

Ready to go!

Now you are ready to stick the two boards together, again do it carefully to not bend or break the golden contacts between them and place them back into the housing. Fix the screws and put the freshly charged batteries inside. Switch on the calculator! If hopefully all went well your calculator will show a running display with **"HP-25"** or **"HP-21" "HP-22" "HP-27" "HP-29"** showing. Then you can play around and I hope you will have the same little excitement and amazement I felt, when it is finally possible after so many years, to resurrect a machine with modern technology.



A running "new ACT" HP-25!

If it's not running?

Don't panic!

Check the battery contacts. Check the batteries, did you get plus and minus right? Try a power supply with 2.5 Volts directly at the battery contacts. Is the ACT properly seated? Did you really press the pins all the way inside the socket? Check the power switch. Make sure you inserted the ACT right with pin 1 as shown.

If all else fails, but you could measure 6 Volts at VCC and -12V at VGG, and you could measure other ACT pins with the oscilloscope, and they are running, then it must be assumed, that the display driver chips are not working. In that case you should give up. You must find some of these chips somewhere. Or must you really...?

No, there is still hope, even in that case.

Indeed, meanwhile I made also a replacement for the display driver chips. I used the same PIC controller with different software and some modifications. I offer you to send your calculator to PANAMATIK if the “new ACT” is not working. I will have a look inside to find the cause of malfunction. Only if additional chips are necessary it will affect your costs, otherwise I would be glad just to have repaired another old HP calculator for you.

With the availability of display driver replacements now every “Woodstock” calculator can be repaired, regardless of which chip has failed to work. The power supply can easily be repaired by using todays transistors and diodes and everything else can be done by the new ACT and the new display drivers.

Indeed, when I first planned the new ACT, I thought about a PBC, which should be located at the display part of the calculator, that leaves the ACT where it was, and replaced only the display drivers, integrating the ACT functionality in them, but it wouldn't have had access to the keyboard columns and some additional wires had been necessary. Therefore I decided for the ACT replacement. But in case of damaged display drivers it doesn't play a role to add just 5 wires.

The newACT always will bring your calculator to life again!

New PIC Processor

In the past years (beginning from 2020) there was a worldwide short of electronic components. One of the victims was the PIC processor PIC15F1618, which I used for the ACT circuit; it was not longer available. Therefore I had to find another more or less compatible processor for the ACT boards. The new processor which I found was the PIC16F15256. It is more advanced and has more features than its predecessor, like different Timer architecture, double speed and the possibility to map special pins to any of the available peripheral pins. The different timing caused me to make a lot of software adaptions until everything worked again. The software revision for the new PIC Processor is r 1.18 from July 2023.

The new processor makes the “new ACT” more than double as fast as before. It runs with 32MHz instead of 16MHz. But why is the speed then “more” than doubled? The reason is because the Interrupt handler for multiplexing the display and keyboard, which must be called more than 20000 times per second, runs much faster now, and leaves the main program more time to do the mathematical calculations.

Unfortunately the new PIC architecture needed more program space, even with the best optimized compilier options. I had already optimized the size by assembler code to any possible limits. The only possibility to get everything into the new processors program space was just to remove some minor “features”.

I removed the “Original Mode”. I think nobody used this mode to hide the new electronics inside.

I removed the “Power Up Keys”, because showing revision and serial number and toggling the “User Flags” can be done by key sequences.

The “Slow/Fast Speed” User Flag, which reduced calculator speed in Internal ROM mode, is not longer supported. The calculator is always running as fast as possible. Only the PAUSE instruction for displaying intermediate results during a running program, is delayed to give you enough time for reading the display.

I removed also the accuracy adjustment of the stopwatch, because the new processor internal clock curcuit seems to be much more accurate. No adjustment needed any more!

The HP-25E IR version, programmed with Infrared printing instructions, still needed more program space. For the IR printing ACT I decided to remove the “Show ROM Checksum” feature.

All other features are still there and remain untouched.

Features

If you purchased the new ACT replacement, it would be sufficient for having the complete calculator back with all its functions alive again. Yet the new ACT adds some really new features, that you probably never thought an HP-21 or HP-25 could have. Indeed they convert the original “Repair kit” into an “Upgrade kit”, that you might consider of replacing even a healthy ACT with the “new ACT”.

Probably it is much too late for these features, because our modern calculators do have them integrated much better, much faster, much easier. Yes, that's right. But then you will have to ask yourself, why you do use the vintage calculator at all?

Wouldn't it be nice, if your new reborn old calculator could do things, the original never could? Wouldn't it be nice, if the new feature is suddenly there, like a miracle, injected from the future into the past?

Well, everybody has to answer this question for himself. At least there *are* some features, not many, because the limited space of ROM and RAM cannot turn your HP-25 into a smartphone.

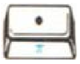
All features introduced here will be available if you purchased the new *ACT* for HP-25 or HP-29C. If you use the new ACT in an HP-21 however, you will not find all features, those related to programming are missing because the HP-21 doesn't have programming. But nevertheless it expands even the HP-21 by adding 88 registers and most of the described features are available.

There are some features, that are inherently not programmable by its intended function, like the *Repeat* or *Insert* and *Delete* feature . But it must be mentioned here, that none of the features will be programmable. I know that you and I would have liked them very much used in a program, I'm sorry, but integrating them into the firmware as programmable instructions was beyond my abilities.

Since V1.04. The “new ACT” will give your vintage calculator the ability of printing if you own an HP82240B Infrared printer. This goes far beyond of a repair, together with the hugely increased storage memory it transforms your extended HP-25E, HP-29E vintage LED calculator into a top model calculator for the 21st century.

And as one of the last achievements I finally found a way to make the “Print X” and “Paper Advance” functions programmable in the printable versions of some calculators (see the printer section for more information).

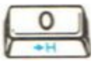
1.) Revision and Serial Number

If you press  at power up you will get displayed the revision and serial number of your ACT.

The Firmware Revision Number is shown at the left as **"r 1.00"** for example, on the right appears the serial number, **"Sr 00001"** in this case. This display is shown until you press any key.

The power up keys decribed here are not longer implemented in revision 1.18 and higher

2.) Standby Mode

Standby mode is toggled by pressing 

at power up.

It cannot be compared to modern standby modes, where you can leave the batteries in your machine for years before you have to replace them, because the calculator needs only microamps to run a LCD display. The old LED technique draws much more current from the batteries, normally the Woodstock calculators need between 120 to 180 mA at 2.5 Volts, depending on how many digits are shown. With modern 2500 mA accumulator cells, this will be enough for 10-15 hours. But the standby mode helps you to get one third more battery lifetime, just by reducing the number of displayed segments after one minute without keyboard activity. There will be shown only one decimal point beginning at the left and moving slowly to the right side of the display, signaling that the calculator is still switched on and waiting for input. As soon as you press a button, the display will show its previous contents regardless which button you choose.

3.) HP Logo

You can show or hide this feature by pressing the button 

at power up.

It is probably not a feature, but if you switch on the calculator, its name „**HP-25**" or „**HP-21**" will show up until you press any key. This is not the original behaviour of a woodstock, it is a new gimmick. You can show this to your friends as evidence, that you really have a new ACT chip inside, because the original ACT always came up with „**0.00**" in the display.

4.) External/Internal ROM

The new ACT executes the original HP ROM code of your calculator for doing its calculations, but it has two ways to do so: reading the original external ROM chips and executing what they read, or using a copy of the same ROM code, which is integrated in the *"new ACTs"* internal ROM.

If you hold the 

key pressed while switching on the calculator, it will change between external ROM and internal ROM mode. You can decide, whether you want to execute from fast internal or from original external ROM whenever you switch on the calculator. If one of these modes doesn't work, switch on again to try the other mode.

There are two reasons, why the calculator should be run from internal ROM inside the new ACT:

- If the original ACT was damaged, the external ROM could be damaged as well and not be able to output its ROM Code.

- The time to read the external ROM could slow down the calculator.

Indeed these two reasons are given and therefore we recommend to run the ACT from internal ROM, it is running much faster (four times faster since r1.04, more than 8 times faster since r1.18), because it doesn't need to read each instruction serially from the original HP-ROM, and you don't need the external ROM chips.

But there are two other reasons, that could point to use the external ROM.

- The external ROM could contain a different software than the ACTs internal ROM and you want to execute this version.

- You want to use the same new ACT in different machines and there is only room for the ROM code of one machine inside the ACT.

The *"new ACT"* chip contains only one preprogrammed original HP ROM Code, either for HP-21, HP-22 or HP-25 etc. whatever you ordered. If you decide to use an ACT with HP-25 Internal ROM in an HP-21 machine, you will have switch to "external ROM" execution, to run the HP-21 calculator.

5.) Slow/Fast Speed

Press and hold the 

key at power up to toggle between slow and fast speed mode.

Since the r1.04 ACT version, the heart of the emulator has been undergone a major optimizing process. Actually it is written in pure assembler code. You will benefit from this optimizing by a very fast execution speed. If you want to run your program with full speed select the fast speed mode when running from "Internal ROM". There is no indication, which mode is selected, you will see the difference when making your first trigonometric calculation just by watching the impressive speed difference.

Since r1.18 the new ACT runs always in Fast Speed mode when Internal ROM is seletced

6.) Continuous Memory

If you switched off an original HP-25, its program and register contents were gone and you had to type it in again the next time. When Hewlett-Packard introduced the HP-25C, the program was stored in a CMOS RAM, named *„Continuous Memory“ and retained its contents*. Now, the new ACT introduces *„Continuous Memory“* for the HP-25, because programs and registers are saved into Flash memory. Even after switching off and removing the batteries, the program will remain stored. So, with the ACT replacement you transform your HP-25 automatically into an HP-25C.

Both your actually loaded program and your eight registers were saved in “Continuous Memory”. Since HP-25E Version 1.09 also the four stack registers and display mode, trigonometric mode and the actual program step will be saved and the last visible number will show up directly after power on.[1]

Please note, that the behaviour will be slightly different, because *“Continuous Memory”* will be stored in Flash Memory and the original CMOS RAM will not be used. Flash Memory will be updated, whenever the PRGM/RUN 

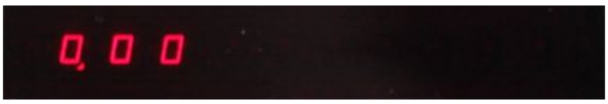
switch is moved, either back to RUN mode or to PRGM mode. If you switch off the calculator before you saved the memory, the last few entries will be lost. The advantage is, that memory is kept even when batteries are removed. If you want to be sure, that the registers and your program will be saved in *“Continuous Memory”*, just switch the PRGM/RUN switch to the other side before you switch off your calculator 

.

7.) Original mode

If you are a purist and don't like the new features, you will love the *“Original mode”*. Whenever you don't want to see the additional features, which were not part of your original calculator, like sleep mode or showing a logo at start up, you can switch your calculator to this mode, which is as close as possible to the HP-21, HP-25 or HP-29C as they work originally. Just start your calculator while holding the 

key pressed. This will switch off every features, which are built into the *“new ACT”*. If you start up the next time with the same key pressed, the features will be again at your service. In original Mode the display always starts up with **0.00** left justified, as you know.



No rule without exception: *“Continuous Memory”* will be active all the time, also in an ordinary HP-25 in original mode. But it will be nearly invisible and so helpful, that nobody wants to miss it.

Original Mode is not longer supported since r1.18

[1]Prior to HP-25E V1.09 display mode and the four stack registers are not saved and will be always reset to FIX 2 and 0.00 like the HP-25C did after switching on, however the HP-29C will save also the X register and the actual display format in “Continuous Memory”.

8.) Stopwatch

Just press 

in RUN mode[1] and a stopwatch with hours at the left and 1/100 seconds at the right will appear.

It is comparable to the HP-55 stopwatch function, or more accurately, it is functioning like the HP-45 hidden stopwatch. You can Start/Stop the timer, and reset it to zero. When the stopwatch is active, all calculator keyboard entries are bypassed and interpreted by the stopwatch.

Start/Stop/Reset

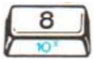
Start and Stop functions are placed on the 

button. Reset to zero is naturally done by the 

Button after the stopwatch has been stopped. But you can reset the time while it's counting, it will immediately continue counting from zero then. If you switch back to calculator mode[2] with another 

, while the stopwatch is running, it is still counting in the background and you can switch back whenever you want and read the elapsed time. Probably it tells you how much time you have needed to solve your actual mathematical problem?

Stopwatch at power up

If you want to see the stopwatch directly after switching on your calculator, set the *"Show Stopwatch"* flag as described in "Functions", or press button 

at power up to toggle this flag.

Stopwatch calibration

The initial accuracy of the stopwatch is not too good, because it runs without quartz. That is the reason, why I mentioned, it would be more comparable to the HP-45 hidden stopwatch that didn't have a quartz crystal inside, other than the original HP-55, that was indeed delivered with an accurate crystal. Not to speak of the famous HP-01, which was very accurately individually calibrated and could even do calculations with measured times. That is beyond the features, which the new ACT can offer.

But there is a calibration procedure for the stopwatch, that you can initiate, if you take only five minutes of your time, that increases the accuracy up to only 1 second in 8 hours[3]. You need another stopwatch as reference and all you have to do is starting both stopwatches at the same time. But don't use the 

button for starting, use the calibration button 

instead. Then let both timers count for five minutes. As soon as the reference stopwatch reaches exactly five minutes, then press 

again.

The ACT stopwatch should stop and display "**00-05-00.00**". From now on, the stopwatch is accurate, as accurately as you pressed the keys. This procedure must be done only once, because the adjustment is stored in Flash memory. However if you need very accurate time measurement and the ambient temperature has changed, then you can repeat it whenever you need. If you press the calibration button and the time is not near the five minutes mark, the calibration is not done and the time will not be set to "**00-05-00.00**".

Stopwatch calibration is not longer needed since r1.18

Two stopwatch timers

There are not only one, but two stopwatch timers present. With the R/S button you can start and stop them individually. Always the currently visible timer will be affected. If you use the CHS button instead, both timers will be started or stopped simultaneously.

You can switch between both timers with the x≷y key. The two timers can be easily distinguished by their separator digits. Timer A shows a minus sign, timer B shows a decimal point sign between hours and minutes.

Storing and Recalling Lap Times

You can store up to eight lap times with STO 0 to 7 . Recalling these values is possible with RCL 0 to 7 .

The stored lap times can be read in calculator mode and you can do calculations with them. The storage format is the same as used for the ->HMS function. Digits before the decimal point are hours, then followed by minutes and seconds and 1/100 seconds.

Chess Clock

There is also a Chess Clock (Game Clock) integrated as part of the stopwatch. It can be used when you press the ENTER ↑ key. Whenever you press ENTER, you change between the two players, each player must press the ENTER key when he has finished his move. Then the next players time is shown. Only the visible time is counting, the invisible time is stopped. To reset both timers to zero you must press CLx two times, one for each timer when it is displayed. If you then start the timer, it will start the currently visible clock.

To distinguish between both players, the chess clock displays a minus sign as separator between the digits for player A and decimal points for player B, so you always know, which time is running. There is no maximum limit time, you must decide by yourself, when the maximum time for one player is up. The Chess clock and the Stopwatch always count upwards.

[1] g NOP is still available in HP-25 PRGM mode. HP-21 is always in RUN mode and uses g DSP button.

[2] also switching the PRGM/RUN button returns to calculator mode

[3] this is equivalent to 1/100 second in 5 minutes

9.) More Program Continuous Memory

If 49 Steps are not enough, and of course memory will never be enough, now you can keep your whole library of HP-25 programs inside your calculator. This is possible, because the new ACT offers 490 additional program steps for fast and easy access, and as described later another 4900 steps in the program library!

The first ten programs are accessible with only three keystrokes. The longest possible program is still 49 steps, but if you like to solve another problem you don't have to type it in, but just get it from where you stored it earlier in your *"Program Continuous Memory"*.

To store the actual 49 steps in one of the ten "*Fast access*" programs, switch to PRGM mode, and just type f STO followed by a number from 0 to 9 .

To recall the program and replace the actual 49 program steps just type f RCL followed by the same number.

When you store your program its number and its checksum will be displayed. Just press any key to quit the message.

10.) More Storage Continuous Memory

Originally the HP-25 has eight storage registers, accessed by STO 0 - 7 and RCL 0 - 7. Now with the "*new ACT*", you have added ten times more registers.

When you are in RUN mode PRGM RUN ,just press f STO followed by a number from 0 to 9, and the eight actual registers were stored into one of the ten new register sets.

If you type f RCL followed by a number from 0 to 9, the actual registers were overwritten by the contents of the previously stored register set. Together you have now 88 registers, all of them saved in Flash *"Continuous Memory"*.

The HP-29 ACT has even more registers, because each set contains 16 registers with a total of 176 registers.

11.) Merging Programs

If the available program steps (49 for HP-25, 98 for HP-29) are not sufficient for a large program, you can divide the programs in parts and store them in different program locations and load and run them sequentially. This merging technique was already used in the HP-67 calculator for larger programs, by inserting magnetic cards with the following instructions and loading the next partial program, taking the results from the previous program as input.

This can be easily done now with the huge amount program storages. Also some register sets can be dedicated to those programs, which need preloaded register contents when started. All you have to do is keeping an actual list of your programs and load the appropriate numbers and register sets when needed.

12.) Show Mantissa

This nice feature shows all digits of a number without decimal comma. We all know this feature from later calculators, but the Woodstock models HP-21 and HP-25 didn't have it. It is invoked by the common key sequence 

and will show all 10-digits available, as long as you hold down the key.

Especially when large numbers with exponent display or scientific notation are used, the mantissa shows the actual number with its full precision and you can write down its digits.

Additionally if printing is enabled, the X register will be printed in its actual display format.

13.) Show Program Checksum

The same key sequence pressed in PRGM mode shows the actual loaded program number and its checksum. This is a convenient way to make certain, which program is actually loaded.

This information is still also available when invoking Function 4, which can be invoked also from RUN mode.

14.) Paper Advance and Available Memory

When printing is enabled the sequence 

will print an empty print line for separation of results. You can call “Paper Advance” several times to move the paper roll forward before tearing the paper with your last calculation.

Since r 1.12 this key sequence also shows the free register and program memory space for two seconds or as long as you hold down the Enter key. This information was previously shown by Function 5 and is now replaced by a much more detailed memory overview (see chapter Functions).

15.) Repeat Function

There are six keys, that have an automatic repeat function when hold down.

The  key can be hold down to show the X and Y registers, and the  button displays all four stack registers with only one keystroke.

In PRGM mode, using the  and  buttons, steps forward or backward in the program. Just keep it pressed and the program steps will increment or decrement automatically, you do not need to press the buttons many times.

Since version 1.12 also the  and  keys have auto repeat in RUN mode. If you fill the stack with a constant and hold down these keys you can calculate the continuous sum or the growth of a value.

16.) Show XY

As described above when you press and hold down the  button, it will be executed twice and you end up with the same X value, that was displayed before you pressed the button.

17.) Show Stack

Three of the four stack registers X, Y, Z, T are always invisible, only the X register is shown (nearly) all the time, but you were able to show them automatically by holding down the  button, which cycles through all four stack registers. This can be useful, if your program has stopped with four result values, using up the complete stack. You could say there are more important things in the world, and I agree, but it's a “nice to have”. This Repeat function can be switched on/off by the corresponding flags described later.

18) Show Registers

You can show all eight registers sequentially by the keystrokes   . The registers, beginning from 0, ending with 7 (or more in case of the HP-29), were shown one after the other for one second in the X register. First the register number is shown on the right, then the value is displayed.

Unfortunately the stack registers will be overwritten by this action, because I preferred to show the registers without having too much display flicker, which would be seen, if some stack drop actions had to be inserted before each Recall.

19.) Show Statistical Registers

The statistical registers are a subset of the registers above. When   are pressed, the registers 3-7 are shown. They contain statistical sums like described in the HP-25 manual.

20.) Show Program steps/ Print Program

When you are in PRGM mode and press g SST all 49 program steps will appear one after the other and you will have both hands free for writing some notes.

The sequence will be stopped by any keystroke or by switching back to RUN mode. If the last step is reached or all following lines are empty, it will stop automatically (prior to V1.04, it never stopped and started from 00 again). So you can check and recapitulate what you have written without pressing the SST button 49 times.

If printing is enabled, simultaneously each program step is printed together with its mnemonic to the HP82240B printer. Thus you can have a printout of your complete program if you start from the first step or only a partial program if you start from any line number.

Automatically stepping backward in your program is not implemented, because you can use the BST key and its repeat function for this purpose and the corresponding g BST sequence is occupied for the more important undo function described later.

21.) Single Step Animate

Surely not the same as showing program steps is executing program steps. This is done with g SST in RUN mode .

Beginning from the current program step, the program will be executed one step at a time, first showing the program step for one second, then executing it. It's the same as pressing SST for one second and then release it. Any GTO instruction will be executed and jumps to the program step that it's pointing to.

22.) Show Annunciators

There was always a lack of knowing which trigonometric mode is active at the moment and whether a prefix key is pending. This is accomplished now by annunciators in the display.

You activate annunciators by g GTO 0 3

Trigonometric mode

There are three trigonometric modes, DEG, RAD, GRD. But you couldn't see which of them are currently selected. Now you can activate the annunciator display (see Functions, Flags). If RAD trigonometric mode is selected, you will always see a decimal point at the far right. If GRD is selected, the decimal point will be shown at the second digit from the right. The most convenient DEG mode, where a full circle is measured in 360 Degrees, doesn't show any dot, so most of the time you will see the usual display with a number and its regular decimal point. This annunciator is shown only in RUN mode.

Prefix key

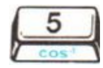
The second annunciator will be shown at the far left of the display. It shows whether a prefix button was pressed and waits for completion of its key sequence. All prefixes were combined in one dot at the leftmost digit, where the minus sign is shown.

This annunciator is shown also in PRGM mode, because it could be useful to know whether the prefix is still active with the new extended instructions, which don't show a prefix code in the actual program step.

23.) Right justified entry

The reason why the engineers of the original HP-25 decided to enter numbers always from left to the right was pure economy. It saved valuable program space, because after each new digit entry, the previous digits would have had to be shifted to the left, which needed some extra steps. In modern calculators we are often used to enter the numbers from right to the left, because the least significant digits are always at the far right, like when writing numbers on paper.

Can you imagine entering numbers on a HP-25 from right to the left?

Now the *“new ACT”* has the power to do that. Numbers were entered from right to the left and displayed right justified. To activate/deactivate this mode you must toggle the “Right justification Flag” by either pressing button  at power on, or setting this Flag with the above combination as described in the “Functions” chapter.

If you activated right justification, you will see any numbers on the right side. At the first moment this is very unusual for a HP-25 user and nearly unbelievable. How can this be done with an emulator, that does run only the original HP-25 firmware? I admit, that this was the last feature that was included in the extended features of the ACT firmware, because it was not easy to implement[1].

[1]When running a program and showing results with f PAUSE or single stepping through the program the right justification is temporarily not activated.

24.) Inserting program steps

I hope you really will like the following two features, which the HP-25 was missing since forty years:

Inserting and deleting program steps without typing in the following steps again.

Moving all steps upward to make space for a new entry is easily done in PRGM mode by the sequence 

.

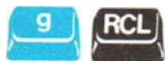
All program steps, following the actual visible line, are moved upward by one and you can just type in your new program step at the following location. For example, if program step 15 is actually shown, then steps 15-48 are moved to 16-49, step 49 is always lost. Steps 1 to 14 stay where they are. Step 15 is copied into 16 and exists double now, but it will be overwritten with your next keystroke. In other words, if you see the instruction that you want to be kept in the display, then you are positioned right to insert the new instruction. If your line points to step 00 or 01, all steps are moved upwards.

But what happens with the destination addresses of GTO instructions? Do they point now to a wrong line number, or will they be adjusted? Yes, they will be adjusted, that means every destination greater equal to 15 in our example will be incremented automatically by one. With this method you don't need labels for moving a program. If a GTO 20 instruction was pointing to step 20, regardless whether it was located below or above of step 15, it will be changed to GTO 21. Additionally, if a GTO instruction was pointing to step 49 it will be replaced by a GOTO 00 instruction and should require your attention. Also you have to check a GTO that had pointed exactly to the actual program line. GTO 00 instructions were never changed, because they are used to stop the program.

The new ACT does not offer an insert mode, where *any* sequential keystrokes automatically move the following steps upward, this would have been much more complicated to implement, and does not really give more freedom while entering program code. The main advantage is achieved, when program steps can be moved at all.

25.) Deleting program steps

Now lets have a look at deleting program steps.

You can delete the visible program step in PRGM mode very easily with 

.

All lines, following the actual program step, will be moved downward by one and the actual line will be overwritten by the following step. Hereby, step 49 will automatically be replaced by a GTO 00 instruction. And, of course, all GTO instructions, that point to an address higher than the actual visible step, will be checked for correction. If a GTO instruction was pointing exactly to the deleted line, it will stay pointing to the same line, but this should cause you to think whether there must be done a correction in your program.

If you just inserted a program step somewhere and you delete it at the same line, the program will be the same as before, except when step 49 was moved out of the memory.

This features should help you to enjoy program entry even more with your new *“Extended ACT Version”* for HP-25.

26.) GTO in PRGM mode

In RUN mode PRGM RUN it is easy to jump to any program step by typing GTO followed by its step number. This is convenient, because you could hold several small programs in memory and the line number represents something like a label.

In PRGM mode there is no way to jump to another program location, the calculator assumes you want to enter a jump instruction instead.

However sometimes you want to advance the program line to a line number some far steps away from your current location to see what you have programmed there or looking for a place to enter a new part of the program. You could use and hold down the SST button and wait until you arrive at the desired location. This could last more than ten seconds. There is a short cut now:

To avoid programming the GTO instruction, you follow it by the • key.

For example: to jump to the last program step just press GTO • 4 9 and voila!

27.) PRGM/RUN switch buttons

It may want to switch between RUN and PRGM mode without using the switch by just pressing keys instead.

f g switches to RUN mode, g f switches to PRGM mode.

Whenever you use the real switch PRGM RUN it will behave as you are accustomed to.

"Continuous Memory" will be saved whenever you use either the real switch or the switch buttons.

28.) Undo arithmetic operation

You don't have to use LastX and perform the inverse operation to get the same stack result as before, just execute the Undo sequence g BST. This key sequence in RUN mode restores the stack registers as they were before the last arithmetic operation, it will place all four values back into the registers X, Y, Z, T

If you accidentally entered 4 ENTER↑ 5 − and you wanted to press + instead, just press g BST + and you will see the corrected result.

29.) Undo Program Step

When entered in PRGM mode, the same keys g BST will restore your last entered program step, if you occasionally have overwrote it. This can happen very easily by nearly any button and you don't know what the program step contained, because always the invisible next step will be overwritten when entering a new step.

There is only one undo step possible. If you want to undo a sequence of steps you could have stored the complete program in one of the available program memories before editing.

30.) Hexadecimal conversion

Very useful for programmers is the conversion of decimal to hexadecimal numbers, or sometimes also octal numbers. There is a base conversion program in the HP-25 program collection number 07, but you will not find it very convenient loading this program just for converting some number and replacing what you previously had loaded, even when you can store and recall it in one of the free program spaces, instead of typing it in again as in the original HP-25 calculator. No, you want to see the hexadecimal number at once and at any time by just clicking two buttons.

This is done by clicking f R/S in RUN mode. It shows the hexadecimal converted number of the current X register value. With these keystrokes you started also the “Hex entry mode”. You can enter any hexadecimal number now by using the numbers and the six “Alpha keys”, A -F:

0 - 9 A B C D E F

To clear entry and start with zero use the CLx key. When you leave “Hex entry mode” by typing f R/S again, your number will be converted to decimal and you can do any math with it in calculator mode.

31.) Octal conversion

In “Hex Entry mode” you can change to “Octal Entry mode” and convert your number from hexadecimal to octal representation any time by the CHS key. This will not clear your input, you can just go on entering octal numbers 0-7 now, all other numbers and alpha keys are locked. It is only another representation of the same number. However, if you leave “Hex entry mode” or “Octal entry mode” and return later from calculator mode, your next entry will start from zero.

Only 32-bit signed or unsigned integer numbers in the range from +-2147483647 or 0-4294967295 will be converted correctly. Larger numbers will produce overflow and display a partial result. The hexadecimal result after converting from decimal are 8-digits numbers ranging from “**00000000**” to “**FFFFFFFF**”, the 11-digit octal values range from

"**00000000000**" to "**37777777777**". If you convert a number from hexadecimal to decimal it will be always treated as unsigned integer and you will get a positive result, i.e.. if you convert -1 (0xFFFFFFFF) you will get 4294967295 when you convert back to decimal.

32.) Binary conversion

If you expect now that I will describe, how your HP-25 can display ones **1** and zeros **0,** I must disappoint you. I know that there are some calculators, which display the binary representation of decimal integers, the most prominent and probably the first was the HP-16C. But they don't have enough digits to display the whole number. For 32-bit integers as we used in hexadecimal and octal mode, the HP-25 would need 32 digits instead of 12, every approach to display binary numbers must show partial results and shift to the right and to the left and show some annunciators for the currently invisible digits, in other words - it's a mess.

No, I added this paragraph because all what you must know as a real programmer for "seeing" the binary ones and zeros is: how to count from 0 to 15. And that is very easy. You need to play with only four bits.

0000 = 0	**0100 = 4**	**1000 = 8**	**1100 = C**
0001 = 1	**0101 = 5**	**1001 = 9**	**1101 = D**
0010 = 2	**0110 = 6**	**1010 = A**	**1110 = E**
0011 = 3	**0111 = 7**	**1011 = B**	**1111 = F**

A binary up counter toggles always the lowest bit and whenever it's falling from 1 to 0, the next bit to the left toggles and so on. That's all. Very easy to learn: 0 = all zeros and F= all ones, 1 = 1, E = 1110 F minus 1, 8=1000, if you need 9, add 1 to 8 and so on, all the patterns are easy to remember. And you never need a calculator to convert them or to enter 32 ones and zeros to get a decimal number. Enter binary numbers as hexadecimal digits, that is much faster and the most ingenius abbreviating ever invented for binary numbers.

If you only know these sixteen 4-bit patterns then you can convert every hexadecimal number of any size to binary and back, by replacing each hexadecimal digit 0-F with its 4-bit pattern of ones and zeros.

Example : **1A60EF** hex **= 0001 1010 0110 0000 1110 1111** binary

The octal representation of a binary number is written very similar. Just replace not four but only three binary digits with the numbers from 0 to 7.

1A60EF hex **= 06460357** oct **= 000 110 100 110 000 011 101 111** binary

For converting this binary number to decimal enter its hexadecimal representation and press 

which shows the converted result 1728751.00 for the above example.

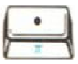
 

= 

Octal numbers were also invented as an abbreviation for binary numbers, but they are much less elegant, because each digit represents only 3 bits and they cannot be easily converted to hexadecimal. Their advantage however is, that they don't need numbers above seven and can be represented in a numerical only calculator.

Now finally you can use the HP-25 as your daily programmers calculator!

33.) PI and e

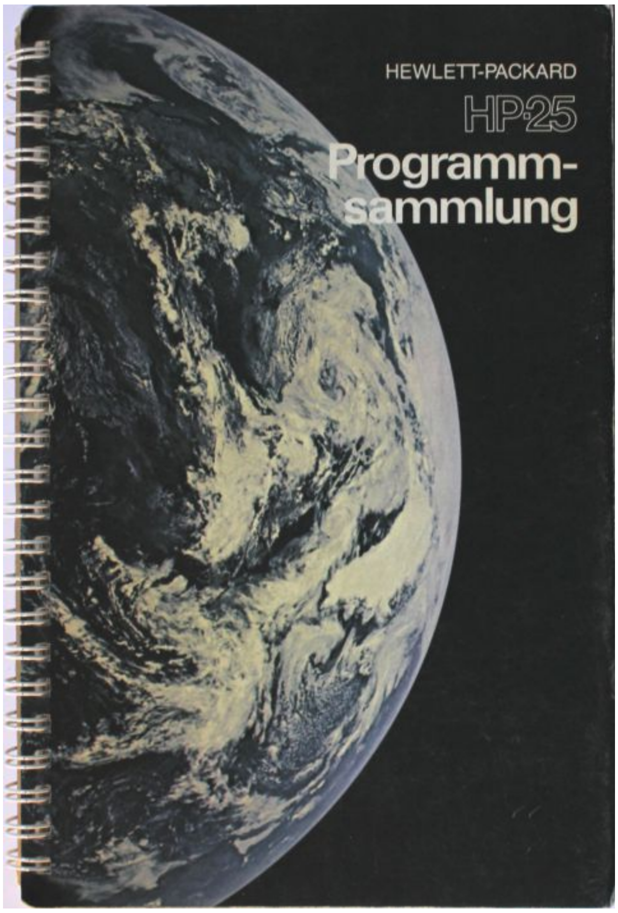
If you type   , you will recall the constant **π**, which is normally shown as **3.14.**

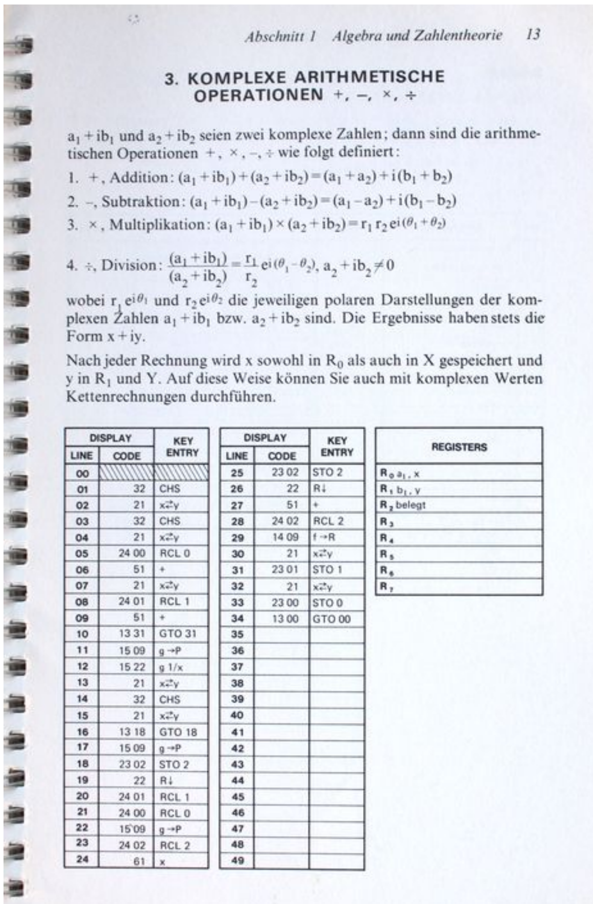
If you activate *"Symbolic display"*, the HP-25 ACT will show the letters "**PI**" instead. Whenever the result of a calculation or your entry represents exactly the number **π** or the Euler number **e** then it will be shown as symbolic name. The Euler number is shown as letter "**E**".

With this another small helpful feature the list of new extended features ends. There were more features possible, but most of what *I could do,* I have done. No one ever before expected the HP-25 to become a calculator with more than it was in 1977. But now it's got its first major update. I hope you will enjoy the improvements.

ROM Program collection

Imagine how sensationally it would have been, if the HP-25, when it came into the market, could have had not only one program space with 49 steps, but the huge amount of hundred times more; equivalent to 100 magnetic cards, if the HP-25 had even had a card reader. This program space would have been enough for any programs, that users could have dreamed of at that time. It would have made the HP-25 a severe competitor to the HP-65, because it could have stored these programs inside without needing to load them by a card reader.





Abschnitt 1 Algebra und Zahlentheorie 13

3. KOMPLEXE ARITHMETISCHE OPERATIONEN +, −, ×, ÷

$a_1 + ib_1$ und $a_2 + ib_2$ seien zwei komplexe Zahlen; dann sind die arithmetischen Operationen +, ×, −, ÷ wie folgt definiert:

1. +, Addition: $(a_1 + ib_1) + (a_2 + ib_2) = (a_1 + a_2) + i(b_1 + b_2)$
2. −, Subtraktion: $(a_1 + ib_1) - (a_2 + ib_2) = (a_1 - a_2) + i(b_1 - b_2)$
3. ×, Multiplikation: $(a_1 + ib_1) \times (a_2 + ib_2) = r_1 r_2 e^{i(\theta_1 + \theta_2)}$
4. ÷, Division: $\frac{(a_1 + ib_1)}{(a_2 + ib_2)} = \frac{r_1}{r_2} e^{i(\theta_1 - \theta_2)}$, $a_2 + ib_2 \neq 0$

wobei $r_1 e^{i\theta_1}$ und $r_2 e^{i\theta_2}$ die jeweiligen polaren Darstellungen der komplexen Zahlen $a_1 + ib_1$ bzw. $a_2 + ib_2$ sind. Die Ergebnisse haben stets die Form $x + iy$.

Nach jeder Rechnung wird x sowohl in R_0 als auch in X gespeichert und y in R_1 und Y. Auf diese Weise können Sie auch mit komplexen Werten Kettenrechnungen durchführen.

DISPLAY		KEY ENTRY	DISPLAY		KEY ENTRY
LINE	CODE		LINE	CODE	
00			25	23 02	STO 2
01	32	CHS	26	22	R↓
02	21	x⇄y	27	51	+
03	32	CHS	28	24 02	RCL 2
04	21	x⇄y	29	14 09	f →R
05	24 00	RCL 0	30	21	x⇄y
06	51	+	31	23 01	STO 1
07	21	x⇄y	32	21	x⇄y
08	24 01	RCL 1	33	23 00	STO 0
09	51	+	34	13 00	GTO 00
10	13 31	GTO 31	35		
11	15 09	g →P	36		
12	15 22	g 1/x	37		
13	21	x⇄y	38		
14	32	CHS	39		
15	21	x⇄y	40		
16	13 18	GTO 18	41		
17	15 09	g →P	42		
18	23 02	STO 2	43		
19	22	R↓	44		
20	24 01	RCL 1	45		
21	24 00	RCL 0	46		
22	15 09	g →P	47		
23	24 02	RCL 2	48		
24	61	×	49		

REGISTERS
R_0 a_1, x
R_1 b_1, y
R_2 belegt
R_3
R_4
R_5
R_6
R_7

The “Extended ACT Version” provides space for 100 additional programs, 54 of them are preloaded from the original HP-25 program collection. The program collection contains programs from different applications, like geometry, financial, navigation and so on. You can load them just by typing their number like 03 for the above example of “Complex Arithmetic Operations”[1].

If you don't own the HP-25 program collection with the description of each program, a scan of both language versions, together with a huge amount of other HP manuals, can be ordered from the HP Museum MoHPC web site.

[1] Note that the program collection is copied from the german edition “HP-25 Programmsammlung”, which is neatly numbered but for some reason, which keeps a secret of the editors, was rearranged and some programs differ in numbering by one.

1.) Load program from Library

Just type f RCL • in PRGM Mode, followed by a number from 00 to 99.

The program with that number is loaded into your 49 step program memory.

Program numbers 01 to 54 contain the complete original HP-25 program collection, that was part of the package, when buying the HP-25.

Program number 00 is not part of the HP-25 program collection, it is the „Battleships“ game shown later in this manual.

Program numbers from 00 to 54 are write protected, you cannot change their contents and you will never lose them. However programs 55 to 99 can be used for your own. You can store whatever you like here. You have the incredible amount of 100 programs, together with the 10 programs which are directly accessed without the dot button and the actual program memory, there are incredible 5439 program steps at your service.

Whenever you want to check whether the right program is loaded, you can show its checksum. See chapter “Functions” for displaying the checksum of the currently loaded program.

2.) Store program to Library

As you assume, the key sequence for storing a program in the library is f STO • followed by a number from 55 to 99. You will get displayed the program number and the checksum of the stored program

If you enter a number from 01 to 54 your program will not be stored, because the programs in this area are write-protected and you get displayed “**Error**” in the display.

3.) Store program to free location

If you don't know which location in your “Program library” or in your “Program storage” is currently free, you don't have to find a free location. Just store to program number 00 with

f STO • 0 0 and your friendly calculator will automatically find a free location for you. It displays the program number and check sum, so you can see where it was stored. However if there was already an exactly identically program stored somewhere, it will not store the same program to another location, but displays just the number of the identically program and its check sum. If you want to store the same program twice you must explicitly choose a number and use the above “Store program to Library” key sequence.

If you want to store an empty program with check sum 0000, it won't be stored if another empty program space was found anywhere in your program storage[1].

If there were no more programs free, which should not occur so easily, then you get displayed the “Show available Memory” message instead, as described later in “Functions”.

1 If a program was automatically stored at location 00 (HP-29E only), you can clear this location by writing an empty program to it.

4.) Load programs dynamically

Indeed it was hard and needed a long development time to achieve this highly desired feature. But ultimately your HP-25E and HP-29E[1] can dynamically load any of the available user programs from within a running program without interrupting the program flow. This feature finally overcomes the firmware limit of 49 program steps (HP-29 limit is 98 steps). The technique is also called “Program overlaying”.

The single program length is still 49 program steps, but you can add a “Load Program” instruction anywhere in your code, while new 49 steps will be loaded from your library and program execution resumes from step 01. You can choose to load any random program number or just load the next or the previous program in sequence. There are three programmable instructions, which give you freedom to run a very large program with theoretically more than 5000 steps without intervention from outside. GTO instructions are still limited to within the actual 49 steps. If you need to jump elsewhere, you have to load the correct program part first.

To implement the program overlay technique, the HP-25E repurposes the programmable instructions ENG 7, ENG 8, ENG 9, which normally sets the engineering display format. Now they are used for loading new program code and do not change the actual display format. It is not easily possible to add new programmable instructions to the HP-25 calculator, therefore I could only use existing instructions. But nobody needs the original instructions because in engineering mode never more than 5 digits can be shown, the exponent always occupies the least significant digits. Luckily it is no loss to reuse the virtually useless ENG 7 to 9 instructions.

If you execute the  instruction, ENG 9 coded as **14 13 09,** either by SST or within a running program, the program number indexed by the contents of register 2 is loaded. If the program is running it will keep running and executes the freshly loaded program, beginning from step 01, until a R/S instruction is encountered as usual. The dynamically loaded program can contain another load instruction and so on. You can overlay as many programs as you need.

Register 2 was chosen, because the higher registers 3-7 are reserved for statistical functions and dynamic program loading should not interfere with them.

The number in register 2 must be within the range 0 to 109 (the fractional part is not used), whereas numbers below 100 load the library programs, and numbers above equals 100 load the 10 “Fast Access” user programs. Practically the first 55 programs are not usable because the HP-25 application program collection are single programs and not prepared for overlays.[2]

Typically you need four program steps to load a program, setting the register 2 value and executing the ENG 9 instruction.

7
0
STO 2
ENG 9 // load program 70 from the library

There is no need to save program steps any more, but there still is a way to do so.

If you stored your overlay programs in ascending order you can have an advantage to execute just f GTO 7 ENG 7, coded as **14 13 07,** and the next program from your library (following the program you loaded by ENG 9), will be loaded. You don't need to set register 2 in this case, it is free for other purposes. You only need one program step to load your next 49 steps overlay.

If you are using this command and you never executed the ENG 9 command since you switched on your calculator, user program 0 (number 100) will be loaded, then 1 and so on. After user program 9 (number 109) the program number will resume from 00.

The third instruction in the set f GTO 8 ENG 8, coded as **14 13 08,** is used for loading the previous program, which could be useful if you want to jump back to the start of a big loop, which is spread over two or more overlays, located in a previous program part. If your recently loaded program was 00, user program 9 (which is represented by the highest number 109) is loaded.[3]

Any program can also be loaded in RUN mode just by entering one of the above ENG instructions manually. For example a normal sequence for loading program 59 (and setting the program numer to 59) in RUN mode is:

5 9 STO 2 f GTO 9.

In contrast to thís command the normal “Load Program” command f RCL • 5 9 will not change the actual program number, which is used by subsequent f GTO 7 or f GTO 8 instructions.

With the g GTO 4 function you can display the actually loaded program number and its checksum.

This unique feature makes your HP-25E an even more powerful red LED calculator for the 21st century.

1 since Version 1.10

2 HP-25E with GPS module has a total of 80 (instead of 110) programs. In this version programs 0 to 69 represent the 70 library programs and 70 to 79 the ten user programs 0-9. For having more flexibility only the first application programs 00-19 are write protected. You can overwrite programs 20 to 54 by your own programs if you decide so. To reestablish an application program, which you have overwritten, you have to type it in again from the original book.

3 If executed without previous ENG 9, program 98 is loaded.

ROM Constant Collection

There is another library, the “ROM Constant Collection”, which gives you access to some useful predefined constants. Additionally to the mathematical constant π (the only constant of the original HP-25), there are now twenty more astronomical, physical and mathematical constants ready to recall.

The key sequence for recalling a constant in RUN mode is  followed by a number from 00 to 19. Immediately after recalling, the constant is displayed in the X register, ready to use.[1]

List of predefined constants:

number	name	value	description
00	c	2.997924580×10^{8} m/s	speed of light
01	G	6.6738400×10^{-11}	Newtonian constant of gravitation.
02	N_A	6.022141290×10^{23}	Avogadro's number
03	a_0	$5{,}291772109\times10^{-11}$ m	Bohr radius
04	e	$1.602176565\times10^{-19}$ C	Electron charge
05	g	$9{,}8066500$ m/s^2	Earth acceleration
06	h	$6{,}626069500\times10^{-34}$ Js	Planck constant
07	m_n	$1.674927351\times10^{-27}$ kg	Neutron mass
08	m_p	$1.672621777\times10^{-27}$ kg	Proton mass
09	m_e	9.10938291×10^{-31} kg	Electron mass
10	eE	2,718281828	Euler constant e
11	k	1.3806488×10^{-23}	Boltzmann constant
12	α	$7.2973525698\times10^{-3}$	Fine-structure constant
13	T_0	273.15 Kelvin	0°C, standard temperature
14	ly	$9.4607304725808\ 10^{15}$ m	light year
15	au	1.4960×10^{11} m	astronomical unit
16	K_J	4.83597870×10^{14}	Josephson constant
17	m_u	$1.660538921\times10^{-27}$	atomic mass unit
18	F	9.64853365×10^{4}	Farady constant
19	M_E	5.9736×10^{24} kg	Earth mass

[1] The previous X register contents will be overwritten, there is no stack lift when recalling a constant.

Not enough to get twenty predefined constants always from memory, you can use another eighty constants, that you can define by your own. Together there are now hundred full precision constants permanently stored in your HP-25.

The remaining eighty constants contain the initial value zero. You can store any number of your own just by entering the constant in the X register and typing 

followed by its number. The recall command for your own constants is the same as for the predefined constants, just use a number from 20 to 99 instead.

Functions

The new ACT has a set of function keys. These are not mathematical function keys, but some additional routines, which display some useful information. Functions were called by the key sequence:

followed by a number from to or the decimal point key.

Because the HP-21 doesn't have a key, the sequence for functions on the HP-21 is followed by a number from to or key.

All Function menus either show some text or you can edit something within. The menu will stay there as long as you want and waits for keyboard entry. You can leave and close all functions and return to calculator mode when you press the key.

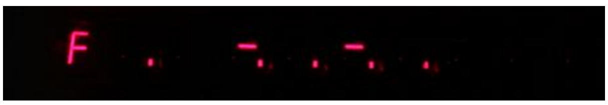
1.) Show Revision and Serial Number

If you press you will get the Revision and Serial Number display.

On the left the Firmware Revision Number is shown as **"r 1.00"** for example. Then the serial number **"Sr nnnnn"** is shown. Each ACT will have its own unique serial number counting up beginning from number 00004, because there were 3 prototypes made. If a customer wants to show the 10-digit serial number, which is printed on the back of his calculator, it can be programmed before purchase. In this case, the calculators 10-digit serial number like **"1810S20234"** is shown automatically after the revision number.

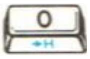
2.) Show Flags

There are up to 20 flags displayed, which were represented by ten dots and ten minus signs preceded by the letter **"F"** at the far left[1] . The dots from left to right show flags 0-9, the minus signs show flags 10-19

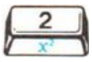


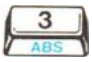
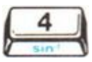
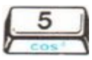
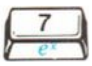
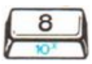
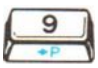
[1] *HP-21 shows H for High instead, because letter F is not available. Flags 10-15 are present since V1.08 or higher, Flags 16-19 cannot be set, they have no function and are reserved for future use.*

The flags have the following meaning:

-  “Sleep mode” This flag will activate or deactivate sleep mode. Sleep mode will be entered after one minute without key entry and saves up to 80 mA.

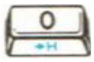
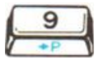
-  “HP Logo” This flag activates the calculator logo message at start up.

-  “Battery check” Switching off this flag can be used to suppress HPs low battery indicator: inverting all decimals points. If your calculator has a battery detection, that shows too often low battery, then you can switch off “Battery check”. However you will not be warned, when the battery is going to get empty.

-  “Show annunciators” Shows annunciators for current trigonometric mode and prefix keys as already described earlier in *Extended Features.*
-  “Repeat keys” Activates auto repeat for SST BST and some other keys like XY and ROLL.
-  “Right aligned display” When activated, shows the numbers right aligned in the display. Other than in normal mode, when all digits start from the left, the number is entered from the right. Only numbers with exponent or with many decimals need the whole display and start from the left.
-  “Symbolic display” When activated the mathematical constants “**PI**” and “**E**” were shown by its name instead by its value.
-  “Hexadecimal Alpha Mode” With this flag you can decide whether you want to get displayed Alpha characters either as **A 8. 2. d E F** or dotted numbers **0. 1. 2. 3. 4. 5**. for representing the hexadecimal letters A to F.
-  “Show Stopwatch” When this flag is set the Stopwatch will appear directly after switching on your calculator.
-  “Welcome” When activated shows your personal “Welcome” message at power up.
-  “Infrared Print Mode” Toggle Infrared printing mode. When set printing is enabled and after power up “**HP-25E Ir**” is shown if “HP Logo” is enabled.
-  “Trace mode” When printing is enabled, this flag switches on Trace mode, where all of your manual calculations are printed on paper.
-  “Internal ROM” When this flag is set, firmware is executed from internal ROM. If you switch off this flag at runtime it will probably cause a crash if external and internal ROM are different.

-  “Original mode” When activated the calculator behaves like the original HP-25 and no additional features are available. When you try to enter the Flags menu again it will not work. You need to press  at power up to deactivate original mode.

-  “Fast mode” When activated executes the firmware in fast mode, which is up to 4 times the original speed. This feature can be active only when “Internal ROM” is selected.
-  “Flicker supression” Since r 1.12 when this flag is set shows a running program without the characteristic original flickering display and calculated intermediate results remain stable until the next value is shown.[6]

All sixteen Flags can be toggled by pressing its above mentioned keys  to  and the keys of the second keyboard row from left to right and . The new state of the corresponding flag will be reflected in the display.

Most of the flags can also be toggled by pressing their key while power up. However for backward compatibility “Internal ROM” will be toggled by  key, “Original mode” will be toggled by  key and “Fast mode” will be toggled by  key. “Trace Mode” cannot be toggled at power up.

The Flags display mode will be quit and the flags will be saved when the  or any other key not mentioned above is pressed.

3.) Show HP Logo

This feature gives you the possibility to show, which calculator you are currently running and which version of ACT is inside. Of course, you will know that before. But you could like to see the name of the calculator displayed, or you want to show the different versions of internal and external ROM. If your ACT contains different external and internal ROM versions then you will see the currently used one. The current ROM code is compared with the internal signature of the original HP ROM for detecting the actual calculator. You can even toggle between internal and external ROM by the  key at power up and the display will show you perhaps another HP calculator, if you have installed a HP-25 ACT in a HP-21 machine for example. If a HP-21 ROM code is detected, it will show "**HP-21**", otherwise "**HP-22**", "**HP-25**" or others, if available. If you run the *"Extended ACT Version"* the calculators name is followed be the letter "**E**"

Note that it is not possible to show the letter "**H**" with the decoder hardware. The segment driver chip of the Woodstock calculators does not offer to display this letter. Nevertheless you can see it on the display. Do you know how this is done?

4.) Show ROM Code

This is surely not the normal job of any calculator. It is an additional feature just for curiosity or for fun. The external or internal ROM code is shown as octal representation when you enter:

On the left the ROM address is shown as letter "**A**" followed by either four octal or three hexadecimal digits, which is a 12-bit ROM address, the maximum an ACT chip can address without bank switching. On the right side the 10-bit ACT opcode is shown. The address is preceded by the letter "**E**" if external ROM is shown. Because opcodes are 10-bits wide its leftmost octal digit is either 0 or 1. An octal digit represents three binary bits. Each number from 0 to 7 represents a 3-bit binary group from 000 to 111. The image below shows the 12-bit **E**xternal **A**ddress 0000 0100 1111 and the 10-bit opcode 01 1000 1110 in **H**exadecimal representation. The following example opcode is 1/2048 part of the original HP-25 ROM

With the  and  buttons you can show the next or previous address[3].

Holding down the buttons will move fast forward or backward. If you have enough time you can visit any of the 2048 (3777 octal) program cells of the two ROM chips of your HP-25 and show their contents, contemplating it. When you look at them, they will be used to be executed at the same time to display themselves.

You can choose to display address and opcode as 3-digit hexadecimal numbers instead of the octal representation. Toggle between the two modes with the  button. If hexadecimal numbers are shown the letter "**H**", otherwise "**o**", will be seen on the far right. Hexadecimal numbers contain normally the letters "**A**"-"**F**" for decimal values 10 to 15 , but not all of them can be displayed, if you deactivate "Hexadecimal Alpha display" (see Functions Flags) they were shown as "**0.**" to "**5.**" instead. The decimal point adds the value 10 to the number.

Or you want to to see external ROM instead of internal ROM. Toggle between the two with the  button [1]. If external ROM is shown the letter "**E**" precedes the address:

[1] This does not affect the ROM that is actually used for emulating.

[3] HP-21 uses + and - button instead

5.) Show ROM Checksum

When you call this function, two checksums were calculated and displayed, the external ROM checksum on the left, marked by the letter "**E**", and the internal checksum on the right.

Both checksums are 4 digit 12-bit octal numbers.

Normally if you run the HP-25E ACT on an HP-25 calculator they should be identical. If they are not identical, the preprogrammed version, which was delivered with the ACT differs from the external HP-ROM firmware. Either because the ROM is not readable or you used another hardware board as your calculator.[1]

[1] Since V1.04 and later the previously described feature of programming internal ROM from external ROM is not longer supported.

6.) Show program checksum

If you call this function it shows the number of the currently loaded program and its checksum. The number, preceded by the letter **P**, is either a single digit from **0 - 9** if its loaded from the “Program Memory”, or like in the picture a two digit number from **00 - 99** if it is loaded from the “Program Library”. The checksum is shown as a 4 digit octal number. This can be useful, if you want to compare it with your own recordings.

An empty program has checksum **0000** and even if it matches with another empty program somewhere in your storage, will display **P - -**. If your program doesn't match with any of the stored programs only the letter **P** is shown without number.[2]

2 Since V1.12 this function is also available by the shorter sequence f Enter in PRGM mode.

7.) Show available Memory/ Register and Program usage

This function informs you about the amount of free program memory and register memory storage. There are a maximum of 55 programs available for your own self-written programs, 10 directly accessible in the *“Program Memory”* and 45 in the *“Program Library”*, each up to 49 steps. Hence the memory usage can show you **“P-00”** to **“P-55”**. If a programs contains at least one program step, it is considered as used. If **“P-00”** is shown, all programs are occupied.

There are a maximum of 88 registers available. The number of free registers are shown right of the free programs. If all programs and registers are free you will be informed by **“P-55 88 FrEE”** in the display [5].

Since r 1.12 the HP-25E has got a greatly improved version for showing the available memory, it gives you detailled information about your programs and registers. You can see which of them are free and which are occupied at one glance.

When you use the  or  key you will see numbers and dots like in the following image. They represent occupied registers or programs.

In the above example register set 0 “**r0.**” is displayed. The digits show the used registers which were previously saved by 

and indicate, that registers 0,3,6,7 of this register set are used, hence 1,2,4,5 are free and contain value zero. The additional dots show the registers 8-15 if available depending on the calculator. Registers 8-10 are free and 11-15 (the statistical registers of HP-29) are occupied in this example.

With the 

button the next reigister set will be displayed, up to 10 register sets (80/160 registers) are available.

But you will also need a quick overview for your programs. You can easily show the program usage by pressing the 

button. You will see a program set with up to ten numbers.

If “**Pr.**” is shown, the numbers show the occupied fast access programs 0-9, which were saved by 

followed by a number. In the above example Program 1 and 3 are used, 2 and 4 to 9 are free. Each line beginning with “**Pn.**” where n is a number from 0-9, represents 10 programs of your available up to 100 HP-25 library programs. In the above example “**P5.**”, the programs 50 to 59 are shown; 50,51,53,55 and 59 are occupied, the remaining programs are empty and free to use.

Program sets “**P0.**” and “**P1.**” are never shown because they are always occupied with the preloaded HP-25 program collection and cannot be cleared. Since r 1.12 programs 20-55 are not longer write protected, you can overwrite their contents by your own programs if you ever should run out of free program space.

With this improved function you get a quick overview of your register and program usage and you can find free locations for your next program in one second.

8.) Operating time and battery time

There are two timer counters, that are counting the operating time in hours and minutes, when you are using your calculator. Whenever you switch on, the operating time is counting up. The first timer value can be up to 99999 hours and will be stored in Continuous Memory. The second operating time counter is thought for any purpose like measuring how long the battery life time was since the last full charge or how long you used your calculator in one week. Its range is up to 99 hours. This second counter is resettable by the  key.

9.) Flash write Cycles

Every time, when your program or registers have changed, the „*Continuous Memory*“ is updated. The data is written into Flash memory. Each Flash memory has a limited amount of guaranteed write cycles. For transparency the total number of actually performed write cycles can be shown here with this function. It will be incremented whenever you switch the RUN/PRGM switch and your registers or program has changed since the last write.

There is a special area of high endurance flash memory used in the hardware of your ACT for the „*Continuous Memory*“. It guarantees at least 100000 write cycles, before writing could become unreliable. You will not get to this limit within years and the limit can be exceeded normally several times beyond the guaranteed value, before it really fails to write.

The “Program Library” and your “Constants collection” are written into a normal Flash memory area which is much bigger, but offers only 10000 guaranteed write cycles. But a single write cycle into this area occurs only when you store a program or registers or constant with their dedicated keystrokes. And you can write each program or register location separately 10000 times. I'm sure there will be a lifetime necessary for going beyond the limit of write cycles.

10.) LED Test

After invoking the LED Test function by number 8, all segments of all digits must be on, showing the digits **“8.8.8.8.8.8.8.8.8.8.8.8.”**, including their decimal points. If the same segment of all digits is not showing up, this is a contact problem with the display connector, you have to open the calculator and wiggle the display connector a little bit. Then, normally it will work again. If there is missing only one segment missing of only *one* digit, this is a more severe problem inside the segment module, and cannot be repaired easily.

11.) Welcome Message

With this function you will be given the possibility to "personalize" your vintage HP-25 calculator for yourself and let him display a "Welcome" message on the display, whenever you switch it on. You may create any 12-digit symbol, that is possible to display by the hardware and, believe it or not, even some more characters which are ***NOT*** possible to display by the hardware. I could not build a complete alphanumerical calculator for you, because HP's BCD encoded seven-segment driver chip allows only five characters and 10 numbers to display. But at least I made exactly half of the alphabet available from its limited seven-segment display capabilities, with thirteen characters from twenty-six in the alphabet. How is this done?

First some introduction to the HP-25 segment driver: It is a chip, that receives a 4-bit code for each digit and two additional bits for showing the minus sign and the decimal point. With 4-bit however you can display only 16 symbols, one is reserved for blank and 10 are reserved for the numbers. The remaining five Symbols are:

r F o P E This is not to be much good.

But there are some more characters, when you allow to use numbers 5 for S , 6 for G and even 8 for B, 0 for O and 1 for I (like India) or l (lower case Lima). Normally they cannot be mistaken inside a word as numbers and you could for instance create a Welcome message like " **HEllO** " with letters E , two 1s and zero. But hold, where is the "**H**"? It is not in the character set of the hardware and we cannot change the driver chip.

You cannot control each LED segment separately by the ACT and circumvent the display driver chip, it is constructed only for the purpose to show numbers and to use as less transistors as possible and only to show three additional characters which are thought to compose the word **"Error"** on the display. These three letters "**E**" "**r**" and "**o**" needed to be used, but what about the other two?

Interestingly the first calculator of the woodstocks, the HP-21s, segment driver could show the letter "**H**" instead of "**F**" as shown above, its character set was:

r H o P E

Either the engineers were recalling for hope or the initials of Hewlett and Packard were coded for this purpose in the calculator. They were invisibly carved into the silicon, but they were never used. The HP-21 and HP-25 and all other woodstock calculators don't use them in their firmware. The "new ACT" is the first chip, that releases the letters H and P from the original segment drivers.

Later when the HP-67/HP-97 and HP-29C were introduced, also the letter "**P**" was sacrificed and replaced by "**C**", to display "**Crd**" for using the Card reader. The times for being forced to change the transistors of a chip to get one additional displayable character and removing another for it are now long ago.

Now we have lost the letter "**H**" in the HP-25 and got "**F**" instead. And we cannot show the letter "**H**"!

Can we? Yes, we can! The new HP-25 gives you automatically three more characters by toggling between two available characters every display cycle. Thus the letter "**H**" is created by toggling r and 4, the letter "**A**" is created by toggling F and 1, and lower case "**d**" is miraculously created by o and 1.

Bingo!

Using this trick, there are now 15 alpha characters available, from which you can compose your "Welcome" message in your vintage calculator, that was made years before the real alpha numericals like the HP-41C came into and replaced the old ones.

Your HP-25 is a three-quarter alphanumerical calculator now, fully numerical and half alpha.

When you enter the function you will be shown the actual Welcome message. If you just want to show it, press  to leave the menu. If you want to edit the message press . The leftmost digit starts blinking now. You can just type your message by using the number keys and the additional alpha keys. You can advance the cursor by  or go back one digit with  .

Add a Minus sign by  or toggle the decimal point with  .

If you are finished with your composition you save it in *"Continuous Memory"* by pressing again, if you don't want to save it, just press the  key and you will leave this function menu.

The additional alpha keys are:

A B[1] C[2] D E F[3] H P R o O I,L Z S G

Space Minus Dot

[1] The letter B is the same as letter 8 with a dot

[2] The letter C is only available in HP-29C hardware.

[3] The letter F is not available in HP-21, it was first introduced with the HP-25 to show the "**OF**" message, then vanished again in the HP-29C display driver.

[5] Since HP-25E r 1.12 90 programs are available

[6] Before 1.12 flag15 was used as "HP-21 hardware" flag. Since r 1.12 you can toggle HP-21 by pressing the DIV key at power up. The PBC keyboard layout of HP-21 compared to other woodstock calculators is different. Normally the boardtype will be detected automatically at power up. In rare cases, where HP-21 hardware is present and the external ROM chips are not readable, this flag must be set manually to tell the program, that the different hardware layout of the HP-21 is present, otherwise the keys will be have a wrong order.

Multitasking

Could you ever imagine to perform multitasking on your HP-25E/HP-29E? If your answer is Yes or No, no matter, anyway this becomes true with firmware 1.12.

You can run your program in the background and invoke every function and use the stopwatch without interrupting the program.

Normally when you have started your program, every keystroke would stop it immediately. The HP-25E now has changed this original behaviour and only the 

key will stop a running program. You can invoke any of the additional key sequences of the HP-25E without interrupting the program. As long as you don't press the right key your program will continue even when results are shown with the PAUSE instruction. If you invoke the stopwatch you can measure the progam run time, either you can show the register usage or edit the welcome message and so on.

If you have started both, a program and the stopwatch, you still can change to another function, the stopwatch and the program are running in the background. At least you can do up to three tasks simultaneously.

I admit, this is not real multitasking, but let's call it true tripletasking.

Flicker Supression

This feature offers a new dimension of glancing at your calculator. Whenever a program is started the HP-25 showed the characteristic flickering display, that indicated a running program. This could be charming, but can also be a pain, when you are waiting for periodic results with the PAUSE instruction, which are shown only for one second; the result vanishes too early and your eyes were filled with the flickering segments and you try to remember what you just have seen.

Since version 1.12 you have the choice of getting your result displayed steadily until the next result is calculated. There is no flickering display any more, only a small dot is blinking at the right, indicating that the program is still running. During a PAUSE instruction the dot stops blinking for one second, then starts again.

This is a true improvement and recommendation for using the HP-25E 1.12 version. I apologize for not having built this feature already into the first newACT versions. At that time I just didn't have the idea. Now it was relatively easy to implement. But at least every ACT from the beginning can be updated. The bootloader has proven stable and it is waiting for improvements not for the first time .

To have some fun with flicker supression you can calculate the factorials when typing in the following small program. Just store the first natural number 1 into register 1 and start.

 Program Checksum = 1610 

LINE	CODE	KEY ENTRY	COMMENTS
01	01	1	
02	23 00	STO 0	
03	01	1	
04	23 51 01	STO + 1	
05	24 01	RCL 1	
06	24 00	RCL 0	
07	61	x	
08	14 74	PAUSE	
09	13 02	GTO 02	

Or try this version without using registers. Program Checksum = 3345 

LINE	CODE	KEY ENTRY	COMMENTS
01	14 34	CL STK	
02	01	1	
03	21	x<>y	
04	01	1	
05	51	+	
06	61	x	
07	14 74	PAUSE	
08	14 73	LASTx	
09	13 04	GTO 04	

Another program calculates the prime numbers. I wrote it in 1976, when I got my HP-25.

LINE	CODE	KEY ENTRY	COMMENTS
01	14 33	CLREG	
02	06	6	
03	71	/	
04	14 01	INT	
05	06	6	
06	61	x	
07	01	1	
08	51	+	

09	23 00	STO 0	
10	05	5	
11	23 01	STO 1	
12	24 01	RCL 1	
13	15 02	x^2	
14	24 00	RCL 0	
15	14 41	x<y	
16	13 37	GTO 37	
17	24 01	RCL 1	
18	71	/	
19	15 01	FRAC	
20	15 71	x=0	
21	13 38	GTO 38	
22	02	2	
23	23 51 01	STO + 1	
24	24 01	RCL 1	
25	15 02	x^2	
26	24 00	RCL 0	
27	14 41	x<y	
28	13 37	GTO 37	
29	24 01	RCL 1	
30	71	/	
31	15 01	FRAC	
32	15 71	x=0	
33	13 38	GTO 38	
34	04	4	
35	23 51 01	STO + 1	
36	13 12	GTO 12	
37	14 74	PAUSE	
38	02	2	
39	23 51 00	STO + 0	
40	24 02	RCL 2	
41	15 71	x=0	
42	13 45	GTO 45	
43	23 41 02	STO - 2	
44	13 10	GTO 10	
45	02	2	
46	23 02	STO 2	
47	23 51 00	STO + 0	
48	13 10	GTO 10	

Program Checksum = 3405 

Printing with HP82240B Infrared Printer

Since firmware version 1.04 the new ACT has the unique capability to connect itself to the famous HP82240B Infrared Printer[1] and will be able to print its programs and calculation results on paper.

The HP82240B Printer

The necessary infrared transmitting diode can be easily connected to the ACT circuit, just by adding one wire and connecting the other wire to battery plus. Then your woodstock calculator will miraculously become a wireless printing calculator like the top model calculators of later generations.

First, printing of a program on paper is most useful for archiving, but it shows also the program steps as plain text and will be much better readable than the cryptic program code shown in the display. Also seeing all program lines listed together one below the other gives you better overview than reading a single line row by row from the display. Second and no less important is tracing your manual calculations on printer, at which you can document all operations you invoked and their results. Both these features are fully implemented in the **"HP-25E Ir"** Infrared version.

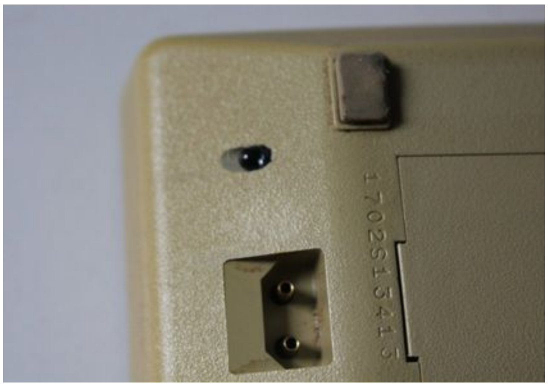
If you ordered the Infrared version of the "new ACT", the circuit will be already equipped with an additional transistor and two resistors and you will get the infrared diode with two cables attached to it, so you just have to drill a hole and to connect the cables to the ACT.

If you already own an HP-25E and you want to update to the Infrared version you can modify it by yourself, this is easily done, you just add a small SMD transistor and two resistors as described in the next chapter.

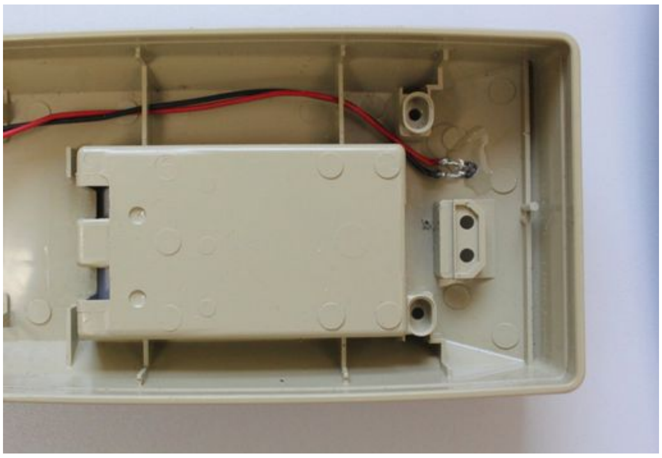
1.) Installing the IR Diode

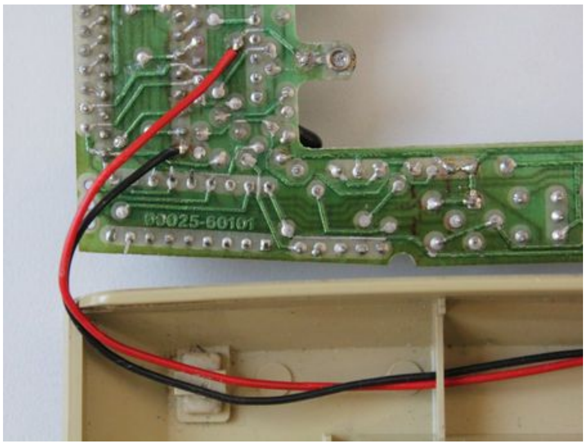
Preparing your calculator for printing needs you to mount the Infrared LED. You can use any 5 mm or 3 mm diode with 940 nm wavelength like Vishay TSAL4400.

A logical spot for minimal changes to your case would be to place the diode inside the charger slot as shown in the image left, but you cannot use the charger any more. You just drill two little 0.8 mm holes into the case with 2,5 mm distance horizontally. Then you cut the pins to about 5 mm length, plug them through the holes and solder a red wire (anode) and black wire (cathode) of 10 cm to each on them inside. The anode will be recognized by the the slightly longer pin. You can fix the diode pins internally with some two-component adhesive if you like.



However I would recommend to chose the place shown in the following images with the smaller 3 mm LED. It is relatively easy to drill the slanted 3 mm hole in the case. Start with a 1-2 mm drill from above, then expand to the final 3 mm, drilling horizontally. It will allow you to still use the charger.





Internally solder the black wire to ACT pin 22 and red wire to battery plus.

If you managed soldering the two wires right you can try your first print line with

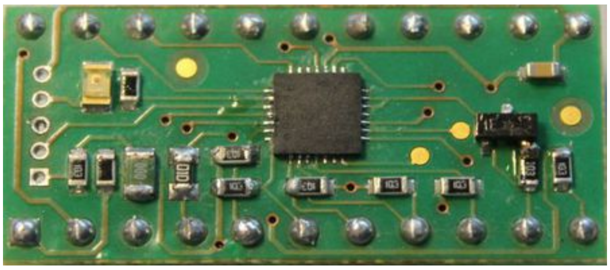
2.) Modifying the ACT for printing

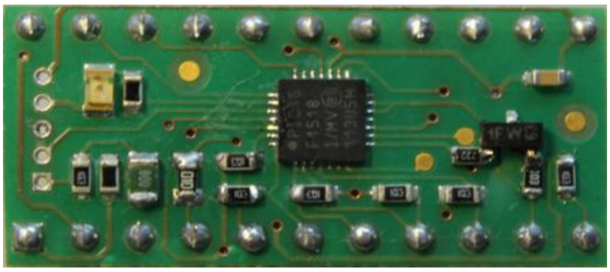
The initial ACT layout did not contain the Infrared transistor amplifier, the idea of adding a print functionality was not yet in mind. To modify the standard ACT circuit by yourself you need to add a NPN transistor BC847 or equivalent and two 0603 resistors 2,2 kΩ and 33Ω. Use the images below as a guideline. You need a suitable soldering iron with a small tip and tweezers to place the very small components at the right place.

After having done the tricky removing of the defective original ACT chip, this procedure should be rather easy for you.

1.) Solder pin 3 of the BC847 transistor to the 10k resistor at pin 10 of the ACT as shown.

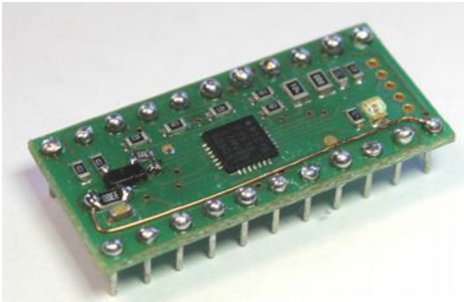
2.) Solder the 2.2 kΩ resistor at the left to pin 1 of the transistor.





3.) Use a small wire to connect the 2.2 kΩ resistor to the gold pad above the resistor.

4.) Solder a 33 Ω resistor to pin 3 of the transistor. Connect the 33 Ω resistor to the unused pin 22 by enameled copper wire. You can use the free pad right of the resistor. Using pin 22 and connect the black wire to this pin on the calculators PCB bottom side, instead of soldering it to the ACT top, allows you to remove the ACT any time from the socket without having the black wire fixed to it.



The diode forward current is limited by the 33 Ω resistor. With the above mentioned Vishay IR diode and a battery voltage of 2,5 Volt this should be enough for the usual printing distance of 50 cm up to 1 m.

Your printer interface is ready!

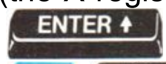
3.) Enable Print Mode

There is no hardware slider switch for selecting the print mode like in the HP-97. The infrared printing capability will be activated by holding down the 

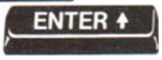
button while switching power on or setting this flag in the flags menu. The display will show **"HP-25E Ir"** then and printing is enabled. The "Ir" mode will be toggled by each power up with the XY button. But you can let the infrared capability activated all the time if you like. When there is no printer listening nearby, the only difference you will notice, will be that "Show mantissa" blanks the display for about 150 ms while printing, and automatically listing the program steps with "Show program" will do equally at each step. Only you should deactivate the "Trace mode" in this case, otherwise the blank display times will happen on nearly every keystroke and might slow down your calculations. The additional current consumption while sending via infrared diode will not significantly reduce battery hours, because the infrared pulses are short and your calculator is a LED calculator anyway. Thus even for normal operation I would recommend you to use the "Ir" mode. Then you just have to switch on your printer and put it in front of your calculator, when you want to start printing.

4.) Print X

The current display (the X register) will be printed in RUN mode, when you invoke "Show mantissa" with 



. Because the HP-21 doesn't have an 

button, Print X will be performed by  

. This is a convenient keystroke for this purpose. The printer will show the currently displayed number in its actual display format, while the display shows the mantissa of the number with all significant digits[1].

5.) Paper advance

If you want to advance the paper by one empty line without pressing the according button on the printer you can achieve this with 

. This key sequence was used before for clearing the prefix key and of course still keeps this function.

Because the HP-21 doesn't have a 

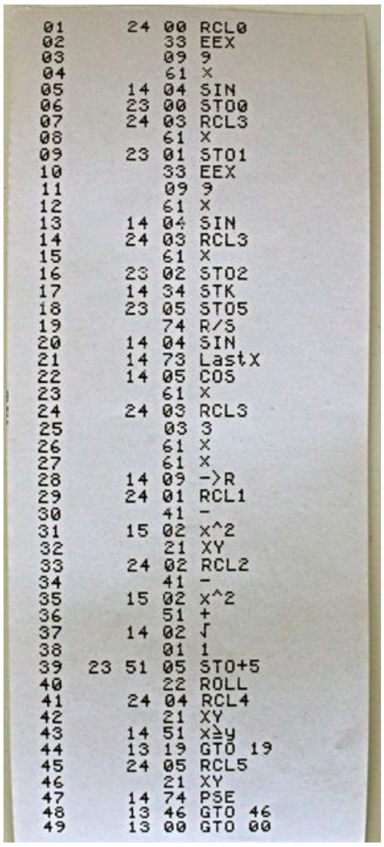
button, Paper advance will be performed by

[1].

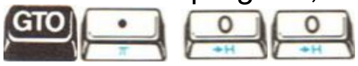
[1] Since HP-25E/HP-29E Version 1.06 Print X has become a programmable instruction. When Infrared mode is activated and running your program, each programmed PAUSE command will print the actual X register exactly as displayed in the calculator. HP-67E V1.02 will print the X register when -x- is executed and will print the stack registers when PRT STK is executed. HP-25E since V1.07 will print an empty line if a programmed NOP is executed. HP-67E since V1.03 will Paper Advance if SPACE is executed

6.) Print program



```
01       24 00 RCL0
02          33 EEX
03          09 9
04          61 x
05       14 04 SIN
06       23 00 STO0
07       24 03 RCL3
08          61 x
09       23 01 STO1
10          33 EEX
11          09 9
12          61 x
13       14 04 SIN
14       24 03 RCL3
15          61 x
16       23 02 STO2
17       14 34 STK
18       23 05 STO5
19          74 R/S
20       14 04 SIN
21       14 73 LastX
22       14 05 COS
23          61 x
24       24 03 RCL3
25          03 3
26          61 x
27          61 x
28       14 09 ->R
29       24 01 RCL1
30          41 -
31       15 02 x^2
32          21 XY
33       24 02 RCL2
34          41 -
35       15 02 x^2
36          51 +
37       14 02 √
38          01 1
39    23 51 05 STO+5
40          22 ROLL
41       24 04 RCL4
42          21 XY
43       14 51 x≥y
44       13 19 GTO 19
45       24 05 RCL5
46          21 XY
47       14 74 PSE
48       13 46 GTO 46
49       13 00 GTO 00
```

If you like to archive your actually loaded program on paper, just press 

in PRGM mode, as you are already used to for showing your program. Beginning from the actual program step, all steps are printed line by line on paper until the last program step is encountered or until you press any button to stop. If you want to print the whole program, start from the first program step with 

. If your program doesn't use all available steps, the printer will automatically stop after the last used program step. Thus you will know that no further instructions follow. Also you don't need to wait unnecessary time and don't waste your paper.

Each print line shows the line number and program code, which is also shown in the display, followed by the mnemonic of each instruction. You will not longer need to decode the row/column program code to know which instruction is contained in the particular line, you just read it.

On the right you see a printout of the “battleships” HP-25 program, which is described elsewhere in this manual.

7.) Trace mode

In “Trace mode”, the printer will show all operations, which you are performing in RUN mode, and you will get a printed documentation of all your manual calculations.

“Trace mode” will be enabled/disabled by 

, toggling the “Trace Mode” flag.

When “Trace mode” is enabled, every single number operation like sin, cos etc. will print one line with the mnemonic of the operation followed by the result of the operation, thus they will produce two print lines. The 

key and 

are treated also as operators in this context and show their “results” when they are pressed.

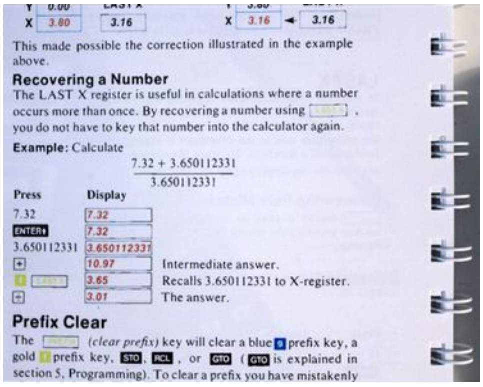
The basic arithmetic two number operators 

show the second operand before the operation, they will also produce two print lines. To show the final result use the Print X function[2].

Some less important operations will not be traced, like changing display format or invoking STO and RCL. The HP-21 has some mnemonics, that are unique to this calculator: M-, M+, Mx, M/ and CLR. They were also traced, when you use the HP-21E calculator.

And last mentioned, I'm afraid that it was not possible to trace a running program. At least the Print X command has become programmable since Version 1.06.

In the image below you see the trace output of the example for "recovering a number" in the HP-25 users manual.



X 3.80 3.16 X 3.16 ◄ 3.16

This made possible the correction illustrated in the example above.

Recovering a Number

The LAST X register is useful in calculations where a number occurs more than once. By recovering a number using [LAST X], you do not have to key that number into the calculator again.

Example: Calculate

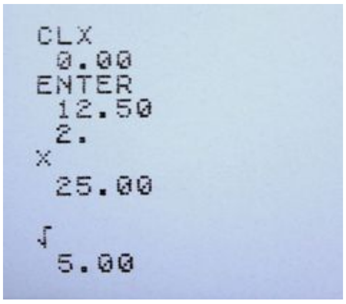
$$\frac{7.32 + 3.650112331}{3.650112331}$$

Press	Display	
7.32	7.32	
ENTER↑	7.32	
3.650112331	3.650112331	
+	10.97	Intermediate answer.
f LAST X	3.65	Recalls 3.650112331 to X-register.
÷	3.01	The answer.

Prefix Clear

The [PREFIX] (*clear prefix*) key will clear a blue g prefix key, a gold f prefix key, STO, RCL, or GTO (GTO is explained in section 5, Programming). To clear a prefix you have mistakenly

Another simple example of traced arithmetic calculation will be shown below:



CLX shows 0.00 as result. Unlike you probably might expect, the number 12.50, which was entered before the ENTER key, appears after the ENTER line. This is because ENTER is treated like any another operator and shows its "result" after the operation[3]. Pressing "x" however shows the operand "2" and the result. The square root is again a single number operation showing the mnemonic and the result. If the result of the last operation produces an **"Error"** like division by zero, the text "Error" will be printed.

If you just enter a number as operand followed by a single number operator you will not see the operand, only the mnemonic and the result is printed. In this case, if you want to print the operand, use the **Print X** function before. If the number was already shown as a result of a previous operation, like the 25.00 in the example above, it is already printed and you don't need to do the explicite **Print X** call.

Alongside the HP10, HP-19C and HP-97 calculators with integrated printer, the "HP-25E Ir" is now placed among these famous printing LED calculators.

However It is in the hall of fame as the only red LED calculator ever made, which prints to the HP82240B printer.

There are some minor differences between the woodstock models concerning printing, which will be described here.

"HP-21E Ir"

As the HP-21 is a non programmable calculator, it cannot print programs, but it is capable of printing the X register, doing paper advance and tracing your calculations in "Trace mode", thus makes it a good office desk printing RPN calculator for tax calculation as well as for good scientific engineering math.

"HP-25E Ir"

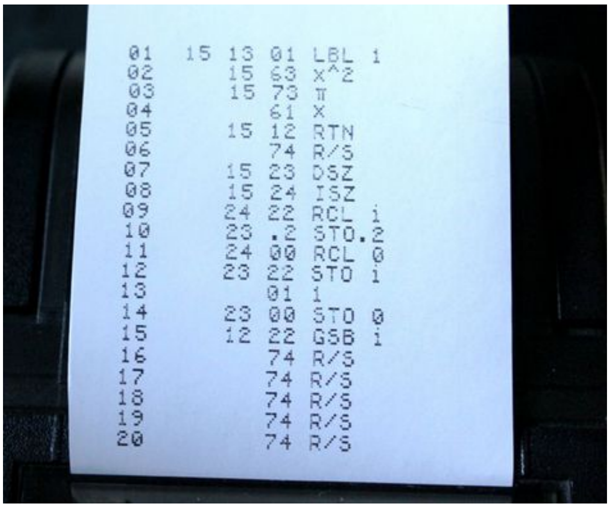
The Infrared Version of the HP-25E has everything that you need for printing, The programs can be printed with up to 49 steps and "Trace mode" is fully implemented. With its up to 110 program storage memories you will find more than one opportunity for switching on the printer and enjoy to get the program steps printed on paper.

"HP-29E Ir"

The HP-29E is also capable of printing to the HP82240B printer. It has some additional mnemonics compared to the HP-25E.

As the new ACT for HP-29C has a different firmware anyway, it contains already the new mnemonics. Showing the program will print up to 98 program steps. The image in the right shows a printout with some typical HP-29C mnemonics like LBL, STO i, RTN etc.

Of course, tracing is available and makes the "HP-29E Ir" the most powerful scientific programmable IR printing LED calculator.



"HP-22 Ir" and "HP-27 Ir"

There was no memory left to implement also the mnemonics of the financial models. Therefore these models will not have the "Trace mode" nor are they programmable. But you will be able to perform at least the **Print X** function and tracing the basic arithmetic and some scientific operations.

[1] The HP82240A printer can be used as well, the only difference will be that two characters, "x" mean and arrow down within "R↓", are not available and thus will be omitted.

[2] Prior to V1.08 the trace mode always showed the final result and used up three paper lines.

[3] This simplified the implementation of the "Trace mode". For further simplicity some less important keys will not be traced, like changing display format or invoking STO and RCL.

Navigating with GPS

This new feature catapults your vintage calculator ultimately from midst of the last century right into the modern space age, where hundreds of satellites orbit our planet, containing highly sophisticated technology. If you like your vintage calculator to communicate with them and you also are an enthusiast of the modern space age, you definitely will become nervous when reading this chapter.

As we all know, GPS (Global Positioning System) has changed the world. Before GPS, ocean sailors and airplane pilots needed to calculate their actual position with the help of sextant readings followed by difficult calculations and a lot of tables, only two or three decades ago. Now they can read the actual position right from the display.

It is interesting to mention here, that the need for a precise calculation of airplane positions lead to the invention of the CORDIC algorithm, which straight on pioneered the invention of the HP-35. *"CORDIC was conceived in 1956 by* Jack E. Volder *at the aeroelectronics department of* Convair *out of necessity to replace the* analog resolver *in the* B-58 *bomber's* navigation *computer by a more accurate ... digital solution. "* [1]. This algorithm was a very effective digital instrument for the calculation of sine, cosine, logarithms, exponential and many more functions. It was offered to Hewlett Packard and gave the starting shot to the development of the HP-35, the first ever HP pocket calculator. It is also an integral part of the HP-25 firmware. And now the circle closes, the HP-25E can receive signals from the modern successors, the most advanced navigation satellites, building the Global Positioning System

Never before a HP calculator could be used for showing the geographical position from thin air. If maintaining a sextant the right way, a pocket calculator could have been indeed helpful to calculate the position by opening the "The Nautical Almanac" an the "Sight Reduction Tables" or by using magnetic cards with nautical programs. But I'm rather sure, that you don't have a sextant, and even in case you have one and also own the navigation pack for HP-41C, you still won't be able to get your position displayed with your calculator every second while sailing or sitting in your car seat.

Now this miracle has become true. You just add the ACT GPS module to your HP-25E calculator. The GPS module with integrated patch antenna is only a 15x15x5 mm wide cube and fits perfectly into the case of any Woodstock calculator. You need to fix only five wires and a small voltage regulator, and your HP-25 will accompany you during walking, driving or flying whenever you are in movement, showing your actual position. More of, it will show also your actual compass heading and speed above ground, number of satellites, and of course, the accurate time with atomic clock precision. Even if you don't have a satellite connection, usually within a concrete building, your RTC (Real Time Clock) runs offline and gives you the actual time. And it allows you to perform a warm start whenever you switch on the calculator, finding the satellites much faster than the first time. This can be achieved, because the battery backup power is always connected to the GPS module consuming only minimal current.

GPS Menu

You activate/deactivate the GPS Menu by pressing g STO in RUN mode.
The “HP-29 GPS” Version uses double prefix g g STO.

There are eight different data values, which can be selected and displayed by pressing its associated number key:

0 UTC Universal Time Coordinated

1 Latitude degrees from 0° to 90° N or S

2 Longitude degrees from 0° to 180° E or W

3 Speed knots or km/h

4 Heading 0° to 360° True North

5 Altitude in meters

6 Dilution of Horizontal Precision in meters

7 Number of satellites found

Each value can be distinguished from each other by its own display format or its label. Due to a lucky coincidence the very limited alpha character set of the HP-25 display driver hardware contains the right letters to show proper labels for all these values. The character set includes the letters H, P, A, E, F and d and together with the numbers 5 and 6, which represent the similar looking letters S and G, they interact perfectly together to build the right words like „**SPEEd**“, „**HEAd**“ and so on.[2]

Next value

Alternatively to using the keys 0 - 7, you can step through these values by pressing ENTER ↑ .Press this key repeatedly and you will be shown all the above values one by one.

UTC (Universal Time Coordinated)

After you entered GPS mode, before you step through different menus, you first encounter the UTC (Universal Time Coordinated). It is displayed as "**HH.MM.SS**" at the right and is updated every second. If „**GPS**" is indicated at the left, the time is directly received from the currently available satellites and will be displayed immediately after receiving each data stream. It also indicates that the position data is valid. If you don't have a satellite fix, that is, if „**GPS**" is not shown, the time displayed is received from the RTC (Real Time Clock) and usually shows a very precise (+-20 ppm) time set from the last fix. If you remove the battery, normally the RTC will start at 0:00:00 and waits for the next fix to get the precise time again.

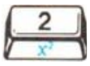
Actual position (Latitude Longitude)

If you step forward with 

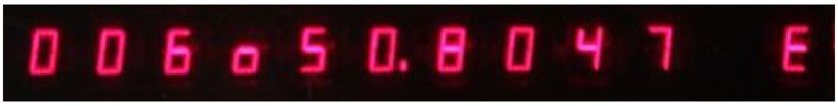
to the next value or press 

, you will see the latitude in degrees from 0° to 90°. On the right you could see the letter "**S**" if you are located in the southern hemisphere. If the rightmost digit is not shown, this indicates N for North.[3]

Pressing 

again or 

shows your actual longitude in degrees from 0° to 180°. If you are located East of Greenwich, you will see the letter "**E**" at the right, otherwise, if it is missing, it indicates West.[3]



If you get displayed just 0° S, 0° E, you did not yet receive a position and have to wait for a satellite fix.

Latitude will be shown as 2 digits degrees, followed by degree symbol, 2 digits minutes and 4 decimal places seconds. Longitude has 3 digits degrees, followed by the degree symbol and also minutes and 4 decimal places seconds.[4]

Speed

The actual speed is calculated by subsequent positions and hence is always speed above ground. If you are not moving, the speed is nevertheless not exactly zero, because of the inaccuracy of position measurement, which is normally between 1 to 3 meters.

If you press the 

button, you can toggle between knots and km/h, knots are the seamanlike expression for nautical miles per hour. As a nautical mile is 1,852 km, the relation between both is a fixed formula:

km/h = 1,852 * knots

knots is about a little more than half km/h

To distinguish between both values, knots will be shown with a minus sign after the label „**SPEEd**“.

Heading

Heading is also calculated from subsequent positions as a value from 0° to 360° true north. It is even more difficult, yet impossible, to calculate the Heading, when you are at halt. But as soon as you are moving you get your true compass course displayed right of the label „**HEAd**“.

Altitude

Especially useful when driving in mountains or flying in an airplane, altitude in meters above sea level can be obtained from the GPS module. However its precision is less accurate than the horizontal position. Usually it can vary between +- 20 meters. Its label shows only the two letters „**Al**“ for altitude.

Horizontal Dilution of Precision

Each position measurement has an accuracy, which depends on the number of satellites available. The more satellites, the more accurate is your position. The GPS module can show the actual accuracy in terms of meters and centimeters. The label shows “**HdoP**”

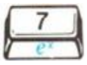
Number of Satellites

A minimum of three satellites is necessary for getting a position. A highly sophisticated correlation procedure is necessary to get contact to the very weak signals from the sky. It needs one or two minutes, the so called „Time To First Fix“ TTFF, to get the signals right. You need to be outside of your building for receiving the data under free sky. Sometimes you can get a connection also inside a building when you are in the topmost floor. Inside a car, there will be no problem. If you have good conditions you will get displayed up to 7-8 satellites. The display shows “**SA**” followed by a two digit number, indicating you see the number of satellites.

Signal Strength

There is one little feature left to tell: When UTC is displayed and you do not yet have a stable satellite connection, then on the left side you will see the no of satellites, which have already established a connection, represented as one to three „**ooo**“ characters. If you don‘t see any „**o**“, then no satellite is yet found.

Auto Display

If you like to observe all values continuously without pressing any button, then use the  key to start the automatic display. It steps through all eight values, showing each value for two seconds and repeat from begin endlessly until you press the  button again or any key from  -  or .

Show Position

The GPS position consists of two values. To show the complete position you need to select Latitude and Longitude. If you want to keep informed about your position, but don't want to switch between Latitude and Longitude, then press the  key. It will start to display both values alternating every two seconds. Press any other key to stop the automatic position display.

Enable/Disable GPS module

When developing the ACT GPS Version, I noticed, that it would of course be fine to have the GPS data present all the time, but the power consumption of 20 mA @ 3.3V which leads to 40 mA from 2,5V batteries would also be present all the time. This is about one quarter of the normal power consumption. If you use the main function of a pocket calculator, which is of course doing maths, then you would not like to run your batteries empty by the unused GPS in the background. Therefore the ACT GPS kit contains a separate step up converter, which supplies the GPS module. The step up converter has three purposes, first it generates 3.3V from the batteries to supply the GPS module, second it releases the HP-25 relatively weak power supply from supplying the GPS module, and no less important it can be deactivated.

When you switch on the calculator, the GPS module is deactivated by default. If you invoke the GPS menu by   it will get activated and TTFF is started. If you leave the GPS menu, switching back to calculator mode GPS will remain activated. If there was no satellite fix yet, it will continue to search satellites and trying to get the fix.

If you don't want to receive actual position data, the you can switch off the GPS module by the  key. After 2 seconds the text „**GPS OFF**“ will appear, because the data stream from he GPS module ceased. Now there is no measurable additional current consumption and you can use your calculator for hours and hours as usual. If you want to enable the module, press  again and the values will reappear.

If you switch off the module, it is still connected to the battery, but doesn‘t need much power, only microamps. In this state the RTC is still running and gives a crystal accurate UTC time, instead of atomic clock accurate GPS time, and it preserves the last satellite positions, which makes the next Time To First Fix much shorter if the module or the calculator is switched on again.

NMEA Data

The GPS module sends serial data in NMEA format to the ACT. The ACT receives "GPGGA" and "GPVTG" messages and extracts their data. The NMEA data format is well documented and explained i.e. here:

http://aprs.gids.nl/nmea/ (english)
http://www.kowoma.de/gps/zusatzerklaerungen/nmea.htm (german)

Store data to register

It was a main goal of the development, that the GPS values should not just be displayed, but are also available for calculations. For this purpose each of the data values can be stored in any of the eight HP-25 registers as a number. Just press followed by - , like you are used in calculator mode, and the actual displayed value will be stored in the dedicated register. There are three different data formats: Position data, Time data, and numbers like Speed or No of satellites.

Position data will be stored as positive or negative number with degrees before the decimal point and minutes and seconds as decimal places behind. Latitudes located in the southern hemisphere will be stored as negative numbers, as well as Longitudes of the eastern hemisphere. This is exactly the data format used by the navigation programs in chapter 4 of the original "HP-25 Applications Programs" manual for calculating Loxodrome and Orthodrome etc. The format takes advantage of the and function, which translates hours and minutes easily into decimal hour format.

If you store the actual **UTC** into a register, it will be converted to a number with hours before the decimal point and minutes and seconds as 4 decimal places. Like for position data you can use the function to convert it to a decimal representation of the actual time, which is more convenient for calculating.

Also the other values **Speed, Heading, Altitude, DoHP and No of Satellites** can be stored in the register of your choice. No of Satellites will be of course always an integer number without fraction part.

If you switch back to calculator mode, you can recall the stored values and make calculations with them either manually or in your programs.

Store dynamic data to register

Now we can go one step further. The ultimate feature for calculating with GPS data is to have the actual values all the time updated in your registers without having to store them manually. This is implemented by using - . If you press this sequence the actual displayed value will be continually stored and updated every second in the chosen register. You can decide which value you want to attach to which register, and you can decide how many registers you will have left free for other purposes.

Stop dynamic data storage

If you do not longer need automatic storage, because you need the registers for other purposes, you can stop any data transfer wit CLx in GPS mode. You can stop a single specific dynamic register transfer by a manual store instruction with STO 0 - 7 to the same register. Then dynamic storage will stop for this register.

Program Examples

As RCL is a programmable instruction you can recall the time or your actual position within a program and this gives you all possibilities to derive other data.

Show Time Zone

Assume you attach UTC to register 0 with f STO 0. then you can display your local time by a small program, which converts UTC let's say to "Pacific Time Zone" of your home in Vancouver, which is 8 hours behind Greenwich. You can adjust to any other Time Zone by replacing this value.

LINE	CODE	KEY ENTRY	COMMENTS
01	14 11 04	f FIX 4	show HH.MMSS
02	24 00	RCL 0	get UTC
03	08	8	
04	41	-	subtract 8 hours
05	15 51	g x>=0	underflow ?
06	13 10	GTO 10	no
07	02	2	yes
08	04	4	
09	51	+	add 24 hours
10	14 74	f PAUSE	display Pacific Time Zone
11	13 02	GTO 02	

Program Checksum = 3126 g GTO 4

Show remaining distance to your destination.

Assume you attach Latitude to register 0 with f STO 0 and Longitude to 1 using f STO 1. Then you can switch to calculator mode and whenever you recall one of the registers with RCL 0 or RCL 1, you will see the actual position in the display. And of course you can access the values by any program.

Store your destination position as latitude and longitude into register 2 and 3, then the following program can easily calculate the distance from your actual position to your destination and show it in the display while you are driving in your car.

A suitable formula for calculating the distance between two points on a sphere like the earth uses the so called law of cosines. R is the radius of the earth[6].

Formula

$d = \mathrm{acos}(\sin \varphi 1 \cdot \sin \varphi 2 + \cos \varphi 1 \cdot \cos \varphi 2 \cdot \cos \Delta\lambda) \cdot R$

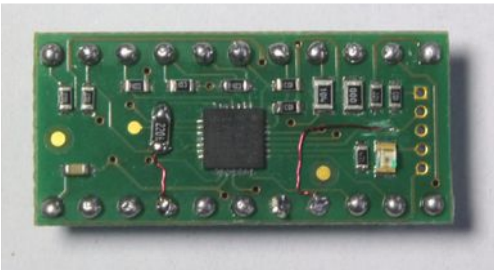
LINE	CODE	KEY ENTRY	COMMENTS
01	15 32	g DEG	Set DEG trignometric mode
02	24 00	RCL 0	Get actual Latitude φ1
03	15 00	->H	Convert to decimal Hours
04	14 05	f COS	cos φ1
05	14 73	f LastX	
06	14 04	f SIN	sin φ1
07	24 02	RCL 2	Get destination Latitude φ2
08	15 00	-> H	
09	14 04	f SIN	sin φ2
10	61	x	
11	21	x<>y	cos φ1
12	24 02	RCL 2	Get destination Latitude φ2
13	15 00	->H	
14	14 05	f COS	cos φ2
15	61	x	
16	24 01	RCL 1	Get Actual Longitude λ1
17	15 00	-> H	
18	24 03	RCL 3	Get destination Longitude λ2
19	15 00	-> H	
20	41	-	Δλ
21	14 05	f COS	
22	61	x	
23	51	+	
24	15 33	g RAD	Set Radians trigonometric mode
25	15 05	g COS^{-1}	acos
26	06	6	Earth radius R = 6371 km
27	03	3	
28	07	7	
29	01	1	
30	61	x	
31	14 74	f PAUSE	Show distance d in km
32	13 01	GTO 01	

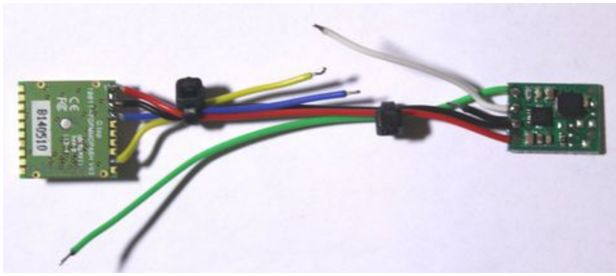
Program Checksum = 2500 

The ACT GPS Kit

If you ordered an ACT GPS kit, a lot of preparations are already done: the ACT is programmed with the GPS Version, two small wires are added to previously unused pins and a small SMD resistor is placed somewhere, and colored cables of different lengths are attached to the GPS module and its power converter, which makes it easier for you to place all the parts into the calculator. But you still have to solder these wires to the right places to get the module connected. You will get advice here and the following images will show you exactly what to do.

First I will explain how to integrate the already prepared ACT GPS kit, then you will get a full description how to build everything from scratch without having ordered the ACT GPS kit. This saves you money, but you have to do plenty of things by yourself. Read about both possibilities and decide later, which is the best approach for you.



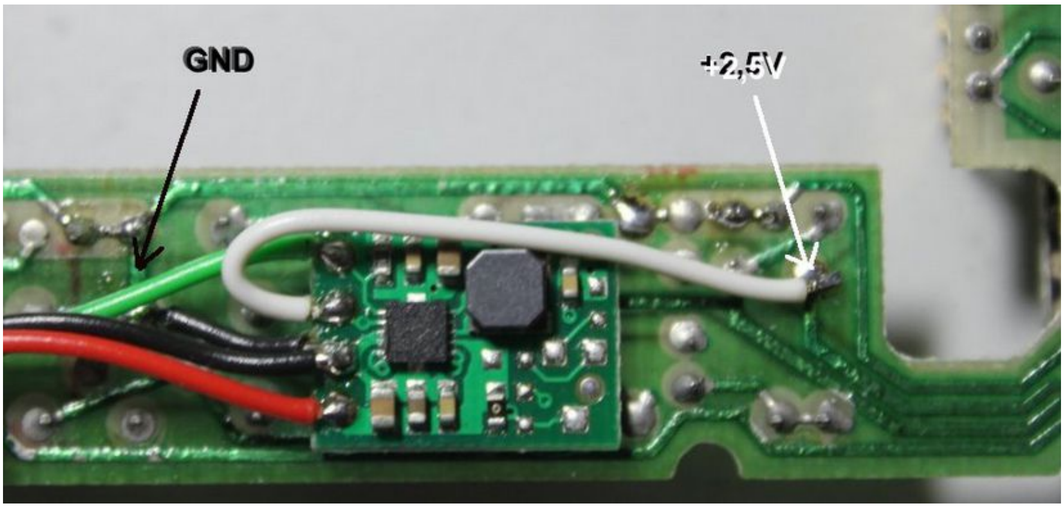


The ACT GPS kit consists of three parts: the new ACT, the GPS module and a 3.3 Volt step up converter.

How to connect the GPS Module?

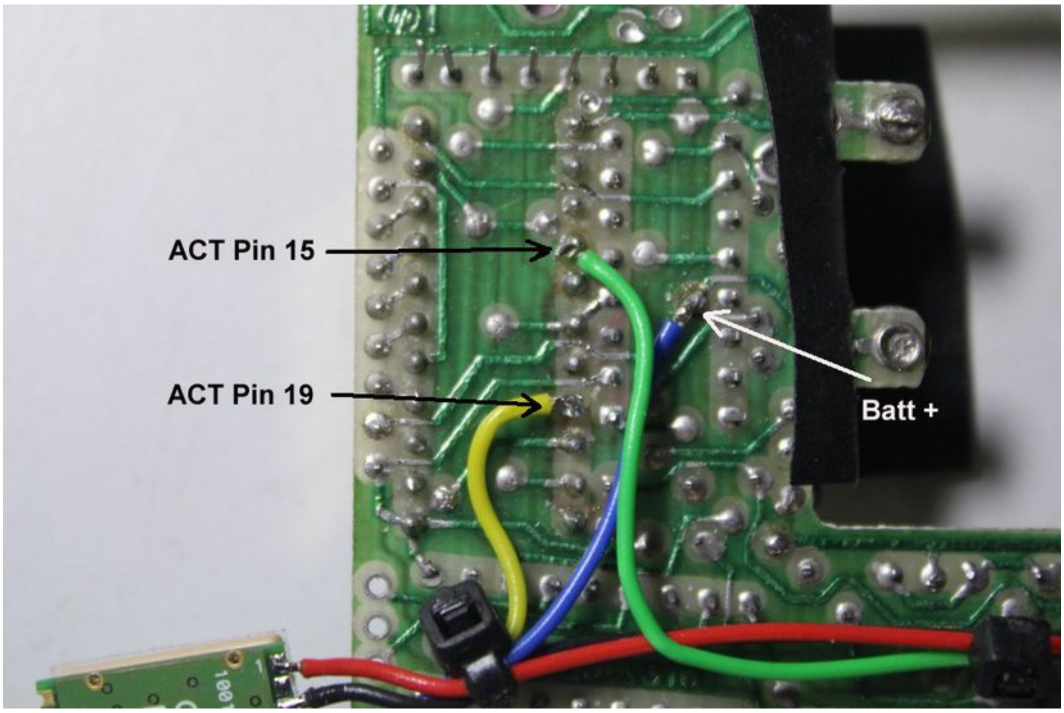
The ACT GPS version must replace the original ACT like any new ACT as described in chapter “Replacement” about the ACT repair kit. Unsolder the old and plug in the new ACT into the low profile sockets, no additional wires are to be soldered to the ACT. You can switch on the calculator and it should show “**0.00**” in the display.

Next, place the step up converter as shown and solder the short black wire to GND and the white wire to +2.5V, which is connected by the ON/OFF switch to battery plus.

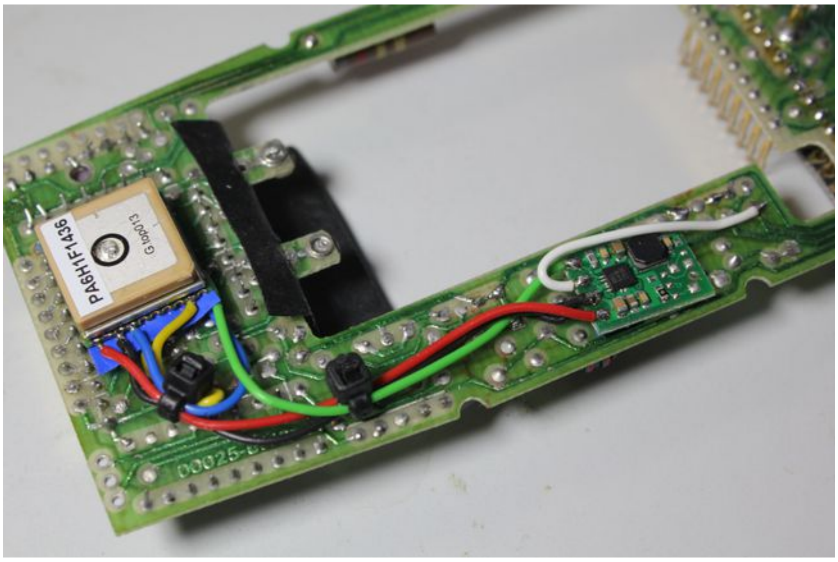


Finally solder the green wire to ACT pin 15. The black cables are connected to battery minus, but there is no need to place them there, we take a better pad nearby.

Now we have to add the GPS module. As preparation you must shorten all the pins of the RAM chip, located right of the ACT on the bottom side of the HP-25 main PCB (Printed Circuit Board). These pins are 2-3 mm too long. Take a small diagonal pliers and cut them. Don‘t try to shorten the pins of the low profile socket of the new ACT, they will get destroyed.



The GPS module is already connected with the step up converter by a red and black wire. There are two wires left, a yellow and a blue. Solder the blue wire to Battery Plus (Batt +) and the yellow wire to Pin 19 of the ACT as shown in the image.

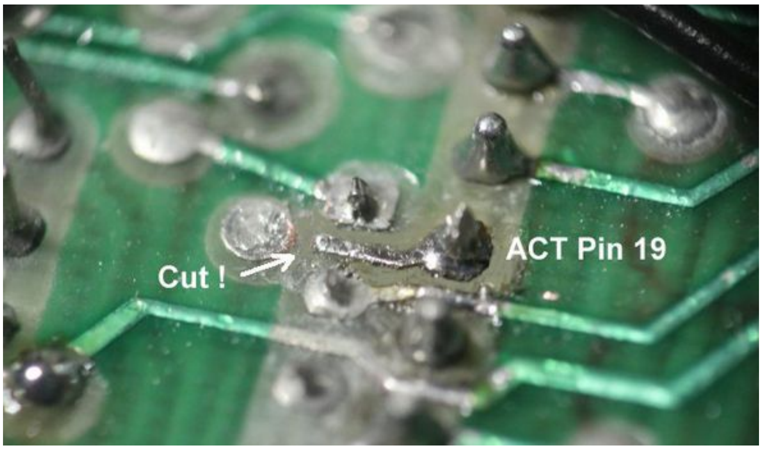


Now you can place the GPS module with the ceramic antenna upside down, as shown in the image, directly below the ACT. This gets you the best sensibility. Don't forget to isolate the GPS module by scotch tape. [5]

The modified ACT for GPS can also be equipped with the modifications for Ir printing. Ir printing uses different pins thus they both don't influence each other.

Attention!

Some later HP-25C boards have pin 19 in use. If you see a copper trace leading away from pin 19, you have to cut it. Take a small cutter and isolate pin 19 from the trace. It leads to the RAM chips, which are no longer needed by the new ACT. If you ever will reinsert the original ACT don‘t forget to reestablish this connection.

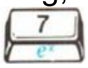


If you like you can optionally fix the GPS module and the converter by a drop of transparent silicone or other kind of adhesive to the PCB, but normally the wires alone will keep the two modules at their place without fixing them separately, at least after having carefully closed the case they will be fixed.

The small step up converter circuit acts as a separate power supply for the GPS module and can be deactivated by the green wire if not in use. It is primarily intended to release the relatively weak internal HP-25 power supply from also supplying the GPS module. Having tried this, I discovered, that the brightness of the display was changing regularly, when the GPS module was active, because the 6,4 V power supply could not provide sufficient current. Therefore now it gets power directly from the 2.5 Volt batteries via the step up converter. The white wire will be connected to Battery Plus as soon as the calculator is switched on and converts to 3.3V for the GPS module. The green wire is a very useful enable signal, by which the ACT firmware can switch on/off the 3.3V at any time, this way the GPS module can be switched off leaving the module in standby mode without additional current consumption.

Now the time has come to put everything together and fix the screws and switching ON the vintage navigation calculator. Be careful while attaching the keyboard to the housing with the GPS module inside and the converter at the right place, where they are intended to be. If there is too much force needed to stick the keyboard together, then stop your attempt and try again. If all goes well, you should see the text „**HP-25 GPS**“ in the display.

Congratulations!

I‘m sure, the first thing you will immediately want to try, is press   and step outside your building, waiting one and a half minute for your first satellite fix. Try the buttons  to . They show you the yet empty values until you get your fix. If you however see „**GPS OFF**“ in the display, then something has gone wrong. The GPS module doesn‘t answer. In this case you have to review your soldering, perhaps you forgot one wire. See the description of the GPS menu for having an excursion into the GPS data, which you can see in your display.

Do it yourself

If you are used to solder some very small parts and wires, you can make your own „**HP-25 GPS**“ calculator for the price of the parts. First you have to order the GPS module and the step up converter from a store.

The GPS module was chosen by its small size and high quality. It is called „FGPMMOPA6H GPS Standalone Module“ and uses the Media Tek MTK3339 chipset, manufactured by Global Top Technology and is available at

https://www.adafruit.com/products/790 (US)

http://www.exp-tech.de/ultimate-gps-module-66-channel-w-10-hz-updates-mtk3339-chipset. (german distributor)

This is an excerpt of the datasheet:

„The FGPMMOPA6H utilizes the MediaTek new generation GPS Chipset MT3339 that achieves the industry's highest level of sensitivity (-165dBm) and instant Time-to-First Fix (TTFF) with lowest power consumption for precise GPS signal processing to give the ultra-precise positioning under low receptive, high velocity conditions.“

The step up converter is also available from the same source. It is called „Pololu 3.3V Step-Up Spannungsregler U1V11F3“.

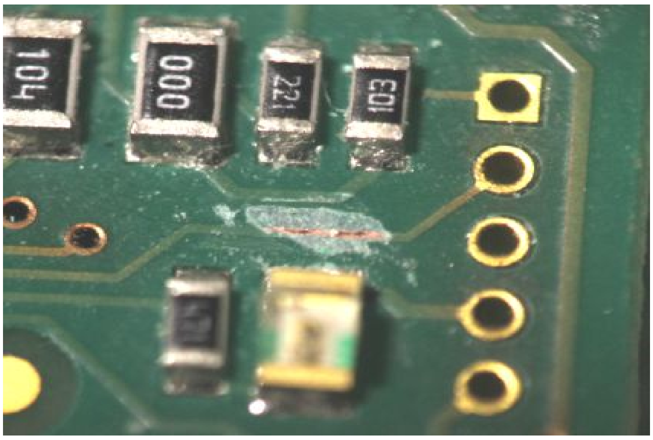
Another excerpt:

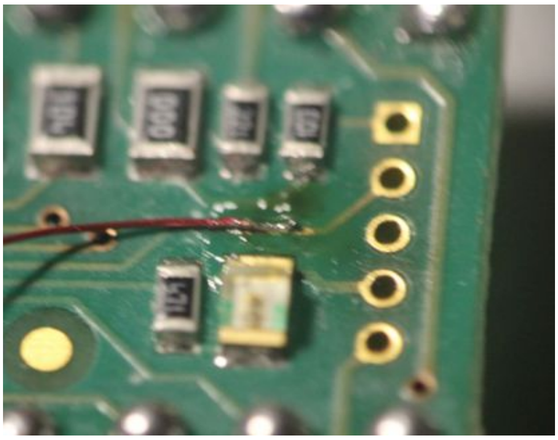
„This 3.3 V boost (step-up) voltage regulator generates higher output voltages from input voltages as low as 0.5 V, and it also automatically switches to a linear down-regulation mode when the input voltage exceeds the output. This makes it great for powering 3.3 V electronics projects from 1 to 3 NiMH, NiCd, or alkaline cells or from a single lithium-ion cell. Additionally, unlike most boost regulators, this unit offers a true shutdown option that turns off power to the load.“

After you got these parts it is easy for you to attach the wires, of course you are not bound to use the same colors or type of wires, which I choose only for giving you a better guidance.

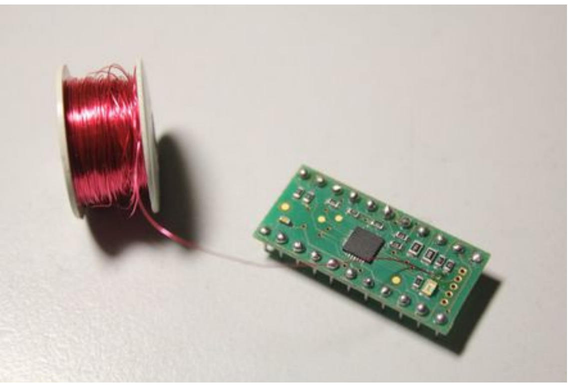
Modifying the ACT

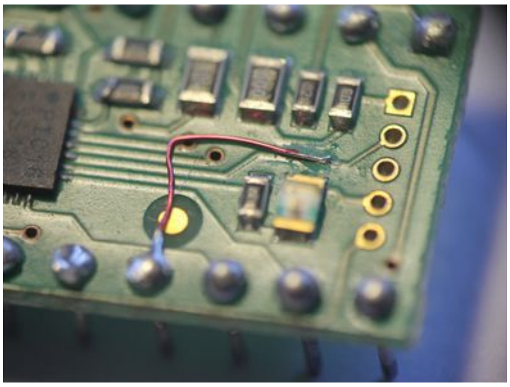
Here comes the tricky part. You must remove carefully the solder resist from a small line, which runs from pin 2 of the 5-pin update connector downwards to the microprocessor. We cannot use the more easy accessible pin 2 itself, because it must be left free for inserting the update connector.



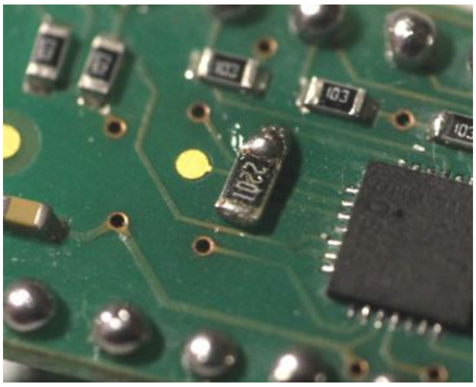


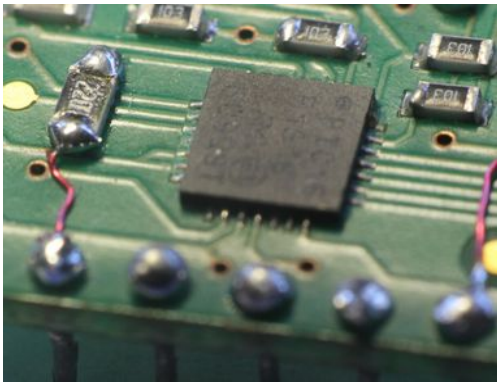
If you removed the solder resist without interrupting the copper, take a 0.8 mm copper wire and attach it carefully with the soldering iron. Then connect the other end of the wire to pin 19 of the ACT.





You have to get a 2,2 kΩ SMD resistor (size 0603 or 0805) and solder one side to the free pad as shown in the image below. Connect the other side to Pin 15 of the ACT.





I recommend the ACT having placed in a socket while soldering to keep the pins in place.

This is your wiring table

Wire	GPS Module	Converter	ACT	Comment
Black	Pin 3	Pin 3	(Pin 12)	Ground
Red	Pin 1	Pin 4	-	VOUT +3,3 Volt
White	-	Pin 2	-	VIN +2,5 Volt
Green	-	Pin 1	Pin 15	/SHDN
Yellow	Pin 9	-	Pin 19	Tx
Blue	Pin 4	-	-	Battery Plus 2.5V

Don't forget to isolate the GPS module antenna side and the converter bottom side by some tape.

Finally download the GPS firmware from www.panamatik.de and use the update kit or your own USB/RS232 TTL converter to reprogram the standard ACT with the GPS firmware as described in the manual chapter „Boot loader“.

If you have done everything right you can be the proud owner of a self made „**HP25 GPS**“ calculator. Try all functions after you get easy satellite contact and enjoy its features.

1 see CORDIC Wikipedia
2 Unfortunately the HP-29C display driver hardware is less cooperative and cannot display the precious letters P, A, and F, which makes no single composition of the labels possible. Although a HP-29E ACT GPS Version is available, it is recommended to insert it into a HP-25 hardware (or HP-21, HP-22), otherwise the labels are not displayed correctly.
3 Letters N and W cannot be displayed by a 7-segment digit.
4 The above coordinates show the location, where this manual was written.
5 I think I observed better sensibility of the module when it is placed this way and less sensitivity when the ceramic side points to the PCB, but I'm not sure and I could be wrong.
6 RCL0 and RCL1 is called only once, to avoid errors during calculation by their changing contents.

Running ACT in different calculators

Each repair kit contains an ACT with preprogrammed *"Internal ROM"* for either HP-21, HP-22, HP-25/HP-25C, HP-27 or HP-29C. The *"Internal ROM"* mode will be enabled by default. If you disable "Internal ROM" mode by holding down the  button while power up, the calculator will automatically become the actual calculator where it is placed in, because it executes the external ROM code. This will work, if the external ROM chip is still alive and readable. There is a good chance that most of the external ROM chips are still in good condition. They are less sensitive to damage than the ACT itself. So, normally you will succeed in running the calculator from external ROM, which is just a little bit slower than the original calculators speed and doesn't matter on a non programmable machine.

However if your external ROM doesn't respond, if your display remains dark or something else happens, in this case you should switch back to *"Internal ROM"* mode. Then you are running the "Internal ROM" firmware of the ACT, regardless of the actual hardware. Even if the "Internal ROM" firmware doesn't match your calculators hardware, i.e. you run HP-25 code on an HP-21 calculator or vice versa, the new ACT can handle the different calculators automatically[1].

Although all "Woodstock" calculators look very similar and have the same number of buttons and the same PRGM/RUN switch, they are different. But nevertheless it is possible to run the ACT with let's say internal HP-25 ROM firmware on an HP-21 or any other model. You just need a vinyl overlay, that contains the HP-25 buttons, and you would have transformed an ordinary HP-21 into an HP-25 programmable machine[1]. Also running internal HP-21 ROM firmware on an HP.25 hardware would be possible, but this is not as good an idea as the other way round. However a small side effect cannot always be avoided when hardware and software don't match: the display driver chips of some models are different and contain different characters, therefore the letters F, C or P will be displayed interchanged sometimes .

More differences can be found in non "Woodstock" calculators or printer calculators, such as the HP-97, HP-19C etc.. It will be a future task to also have replacements for these calculators. At release date of this manual, the "new ACT" has been tested in all "Woodstock" models and the HP-67. Other models which contain the ACT chip are not yet supported. See our website for the latest news.

In the meantime however you can help yourself if you want to repair some of these currently not supported machines, by exchanging the healthy ACT of an HP-21/HP-25 calculator, replace and upgrade it by the "new ACT", and use the original ACT chip for repairing your HP-67 or HP-97. Then you will have both calculators working. I cannot promise, that you can use any original ACT in any machine, but there is a good chance that some of them will work.

[1] The HP-21 keyboard hardware is not compatible with the later models, but ACT V1.04 will automatically correct the incompatibility when running different internal ROM code.

HP-21

„The first of the 20 series was essentially an HP-35 in a smaller, less expensive package.“ This description in the chapter about this calculator in the MoHPC (Museum of HP Calculators) illustrates, that the HP-21 was just a simple non programmable scientific calculator equipped „only“ with the fundamental mathematical functions like sin, cos, tan, ln, log, and square root, besides the basic arithmetic operations. Thus it was a pure RPN calculator with the famous four level stack with only *one* additional storage register.

If the HP-35, as the first of all, was a miracle of programming, using the CORDIC algorithm for calculating trigonometric and exponential functions and square roots with one minimalized algorithm, the HP-21 adopted this firmware and added some improvements like display formatting and storage arithmetic and needed the still incredible small amount of only 2048 words of ROM memory,.

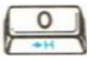
If you take a look to its keyboard, you see only one blue prefix button and no yellow button. The usual yellow printing above most of the buttons of the later models is missing, and it seems, that only half of the keys have a blue prefix function. But the latter is not really true. The digit buttons, which don't show a blue printing, can be prefixed by the DSP button for adjusting the number of decimals in either fixed or scientific display format. Therefore the key sequences for the extended functions must be well thought out.

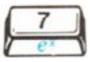
The new ACT is dedicated also to the HP-21. If the HP-21 ROM signature is detected at power up, which can be read either from external or internal ROM, there will be automatically considered the necessary differences in keyboard.interpretations to get access to all the extended functions. However, because the HP-21 is a non programmable machine, the new ACT doesn't change that. When running the HP-21E firmware all extended features, which are related to programming, of course are not available.

Nevertheless what the HP-21E offers you, besides making a defective machine running again, is considerable. It includes the complete stopwatch functionality, decimal to hexadecimal/octal conversion, standby mode, right adjustment, show mantissa, repeat keys, the complete functions menu, and believe it or not, it adds 88 new storage registers which are „Continuous Memory“.

The many new storage registers gives this non-programmable a new class. They can be accessed just by 

or 

followed by a number from 

to 

like in the HP-25. Register 0 is identical with the original HP-21 register. Unlike the HP-25, the storage arithmetic can only performed with register 0, using the dedicated keys M+, M-, Mx, and M/. Originally the STO and RCL buttons were not followed by a number, they were just one button instructions. Now if you press the STO or RCL button, the new ACT retains the keystroke until a number is pressed and thus allows to store the X register into different locations. There are 10 banks of 8 registers each, which can be stored or recalled into the actual 8 registers, like described in a previous chapter.

Because there is only the blue  prefix button some key sequences of the HP-25E must be reviewed for the HP-21E. See the HP-21E key sequence summary in Appendix A for all available functions.

Now, I have to admit, that I didn't tell the whole truth, when I said before, that the new ACT cannot make a programmable calculator from the non-programmable HP-21, it can. If your external HP-21 ROM is still functioning, the new ACT (you need V1.04 or higher) gives you the unique possibility to have two calculators in one. Running from “external ROM” it will be a non programmable HP-21E, running from “internal ROM” it will be a complete programmable HP-25E. The only thing that you need is an overlay for seeing the different button labeling of the HP-25 keyboard.

Anything more to say? Yes, the „HP-21E Ir“ will give you the possibility of printing your results and tracing your operations on the HP82240B Infrared printer, if you have installed the printing ACT version.

HP-22

The non-programmable HP-22 financial calculator is an easy candidate for an ACT replacement. It is fully hardware compatible to the HP-25 and uses the same printed circuit board. The different keyboard layout however needs some remarks.

What on the first sight looks as if only a few buttons are occupied with prefix functions and there should be plenty of them free for the advanced features, turns out to be wrong. All buttons have prefix functions. The number buttons are taken to select the number of fixed point decimals by  followed by any number key.

Together will the existence of only one prefix key this leads to the question how to invoke the extended functions. Even if we remove all the functions of the HP-25E, that are related to programming and are not used in the HP-22, there are still some features like stopwatch or invoking the „Functions“ menu, that should be accessible. What can be done to convert the HP-22 into an HP-22E? The problem is solved when we let us help by the slider switch. If it is in BEGIN position the HP-22E acts as usual, but if it is in END position some prefix functions will be ignored and the keys are used for the extended functions instead.

The available extended functions of the HP-22E are listed in Appendix A

Not available in the HP-22E are: „ROM constant collection“, „ROM program collection“. and „More storage memory“ and of course all functions related to programming.

HP-25 / HP-25C

This is definitely the calculator the new ACT was originally made for. The extended features are designed to make especially this calculator a better one. The key sequences for invoking the new functions are chosen to make use of its available unused buttons.

This manual describes in detail the behaviour of the HP-25E. The chapters about the other models describe only deviations from the main description. Therefore I don't have to write anything else about the HP-25 here, except:

The HP-25 accompanied me from school until today and it was the machine, that told me what programming means. It is still the pocket calculator of my choice. :)

HP-27

The HP-27 is a remarkable calculator. It is the only calculator made by HP in the early days, which combines financial functions and advanced scientific functions in one model. Another remarkable thing is its interior. It looks like a treasure chamber. All chips have gold plated surfaces. The ACT is not a gray plastic chip like in some other models, but resides in a high quality ceramic housing. It would be a shame, when this ACT had become defective and have to be replaced. But if so, at least it can be replaced. The original HP-27 ACT is designed in NMOS silicon, it is the first NMOS chip HP used for its calculators. The power supply unit produces a 12V signal instead of -12V as in the PMOS models. Pin 12 which was the GND pin in the HP-25 ACT is now connected to +12 Volt. And pin 2, which was the -12V supply pin becomes the new GND pin. Therefore the HP-27 needs a modified “new ACT” hardware, these pins must be redirected. But after this small modification the „new ACT“ can be used also as a replacement for a defective HP-27.

Keep in mind, that you cannot mix the two ACTs between the models, you cannot use a normal HP-25 „new ACT“ inside an HP-27 nor use a modified HP-27 “new ACT” in an HP-25 hardware, it will not work.

HP-29C

This is the well known high end programmable calculator of the „Woodstock“ models. The firmware has double the size of the HP-25C to allow additional new functions, the program memory was doubled by the HP engineers to 98 program steps and they provided thirty user registers for storing numbers, instead of only eight in the HP-25 model, thus the HP-29C is an really improved HP-25. Some noteworthy features were added like indirect addressing, labels and subroutines, inserting and deleting program steps and many others.[1]

Unlike the lower models HP-21 and HP-25, which were perfectly running with the „nonpareil“ emulator, the HP-29C was never emulated before. First, its firmware was never extracted or published, and second, it used some instruction sequences, that the „nonpareil“ emulator failed to execute correctly. As a consequence the emulator was hanging in an endless loop, whenever a label search, which is an integral part of nearly every program, had to be performed. As nobody yet got the HP-29C firmware to run by an emulator, Eric Smith was aware of this lack, but for some reason he didn't find anymore time going on with his research to reveal the last secrets of the ACT after all the years that he already had invested. But he freely shared his knowledge and gave me some valuable hints where to search. With the help of a reprogrammed new ACT board it was possible to record the exact program flow of an original HP-29C and comparing it with the differing program flow of the emulated HP-29C firmware. I found the crucial instruction sequences that failed to work and I could eventually correct the wrong loop. The major step forward was done.

Another research case applied to „Continuous Memory“. The continuous memory of the HP-29C is checked at every power up and shows an „Error“ message if it does not contain correct data. I had to find which checksum or byte patterns were expected to avoid the „Error“ message and clearing of the registers and the program. It was not too difficult but needed some time. And after having found the appropriate memory locations and the checksum, finally the first fully functional HP-29C firmware was running with the „new ACT“.

Only a few weeks later the first smartphone HP-29C App was published, containing the ROM firmware from the original calculator, that I had extracted and published .

Because of many new features, which were integrated into the HP-29C nearly all keys were fully used up with the functions of the  and  prefixes. This lead to some difficulty to find appropriate key combinations for making the „extended ACT“ features also available for the HP-29C. Some key combinations like  , which could be used to switch between stopwatch mode and calculator mode in HP-25E, were occupied now in either RUN and PRGM mode. Fortunately some others were not longer used, like inserting or deleting program steps, because they belonged to the new HP-29C functions and were already built in. Also the bigger program size and the firmware needed a lot of adaptions in the ACT firmware to get a HP-29C „extended ACT“ replacement running.

Another problem arose because the display driver chip of the HP-29C was coded without the possibility to show the letters „**P**“ „**F**“ and „**A**“ anymore. Some texts like „**FrEE**“ in the memory usage function and the „**PI**“ symbol could not longer be shown. Accordingly the calculators name „**HP-29C**“ could not be displayed. I choose to show „**HC-29E**“ instead. It can be translated as „*Hp Calculator-29 Extended*“ But all in all these were only minor setbacks, which did not really count. Finally nearly all extended ACT features could be activated within the HP-29C hardware and the emulator is running the original firmware flawlessly.

Disregarding the different display drivers and different RAM and ROM sizes, the HP-29C hardware is identical to the HP-25. It is possible to run the HP-29E on an HP-25 hardware, when using the internal ROM and the internal RAM banks, however a slightly adapted keyboard overlay would be necessary in that case. Because of the different memory requirements however the extended features for storing programs and constants will not be compatible, thus it is not possible to run an HP-29E ACT and switch to an HP-25E with external ROM when it is inserted in an HP-25 hardware.

Extended Features

The key sequences of the extended features differ from the HP-25E, they are summarized in the Appendix A in the HP-29E Key sequence summary. Following are listed the main differences between the HP-25E and its new counterpart: the HP-29E.

ROM Program Collection

There are 30 programs of 98 program steps available. Together with the actual program memory these are 3038 program steps altogether. Ten programs are stored with

f STO 0 - 9, twenty more programs can be stored by

f STO. • 0 0 - 1 9.

All programs are free to use, there is no preprogrammed built in ROM program collection. It is up to you which programs you like to store anywhere in the available program storage. There are many HP-29C programs available in libraries, which are part of the MoHPC Document set.

ROM Constant collection

The complete ROM constant collection of the HP-25E is also available in the HP-29E, the 20 predefined constants are accessible by f RCL • 0 0 - 1 9 in RUN mode. Another twenty memory locations for writing your own user constants are available. They can be set by

f STO • 2 0 - 3 9 and recalled by

f RCL • 2 0 - 3 9.

The full set of 100 constants like in HP-25E had to be reduced, because the larger HP-29C ROM firmware occupied the space for it.

Storage Continuous Memory

There are ten storage sets available with f STO or f RCL 0 - 9 like in the HP-25E, however each set contains sixteen registers instead of only eight. Together these are 190 registers, 102 more than the HP-25E provided. However the indirect registers 16-29 are not part if it and cannot be stored in the register sets.

Welcome message

Because of a different display driver chip, the letters **„A,F,P“** cannot be used, however the letter **„C“** is available now as a new letter. In hexadecimal mode the digits A and F will be shown as **„0.“** and **„5.“**

Symbol display

Symbol display is deactivated, because the letter P in **„PI“** cannot be displayed.

Show Memory

The show Memory functions shows up to 190 free registers (10 sets of 16 registers and 30 normal registers), and up to 30 free programs.

Program Checksum

The program checksum function uses the letters „CH.“ Instead of „Pr.“ in the display.

Insert, delete, goto program step

Insert program step, delete program step and goto program step are already build in functions.

Functions

Because [g] [GTO] is already used for label entry, the ACT function menu will be invoked by double prefix [g] [g] [GTO] instead.

Stopwatch

Because [g] [R/S NOP] is already used for the 1/x operator, the stopwatch is invoked by double prefix [g] [g] [R/S NOP] instead.

Show Registers

Show registers is invoked by double prefix [g] [g] [RCL]. The registers 0-9 will be shown. „Show statistical registers“ is not available.

Show program

Show program is called by [g] [g] [SST] in PRGM mode.

Repeat BST is not available, because there is no dedicated BST button.

Printing

Of course the HP-29E Ir printing version can print its complete 98 steps of program memory on paper and has trace capability. The “Print X” command is programmable using a **PAUSE** instruction. But I didn't find a possibility for a programmable “Paper Advance”, because there was no program code equivalent to NOP available.

[1] A complete description of the features of the HP-29C calculator can be found in the MoHPC website.

HP-21/HP-25 Two calculators in one

When the “new ACT“ came into life, the idea of using “Internal ROM” firmware allowed a repair, even if the “external ROM” chips would be damaged. But still it should be possible to run the firmware from the external ROM if they are still intact. This lead automatically to the possibility of having two different firmware sets and thus having two calculators in one. Because the “new ACT” automatically detects the model from the signature of the ROM, that it is actually reading, it knows at power up, which firmware is running and switches its key sequences and some other details according to it. Because the models differ also in hardware layout, the new ACT has to correct these differences if necessary, when it has detected the hardware by its “external ROM” signature, even when running the “internal ROM” firmware.

If the flash memory of the “new ACT” would have been much larger, it would have been even possible to include all woodstock models firmware into one ACT and switching between them at power up. But I preferred to use the available flash memory for giving the programmable models a huge program storage instead, which is surely the better decision. It remains, that you have at least two calculators in one.

The models HP-21, HP-22, HP-25 can be combined this way. It is up to you which model you choose. The ACT for HP-25E runs in any of these, making them twin calculators, just by switching between internal ROM and external ROM with the  key at power up. Also the other way round could be possible, if you order an ACT with HP-22 internal ROM firmware inside and operating it in an HP-25 hardware. Even the display drivers are identical and the displayed characters are identical. Premise of course is that your external ROM chips are not defective.

Only the HP-27 and HP-29C need a special ACT version, the first because it is hardware modified for replacing an NMOS silicon ACT, the second because its double sized firmware could not be included together with the HP-25E in one chip. Therefore they are not interchangeable with other models.

HP-34E

The HP-34C was the flagship top model calculator of the HP-30 series, the successor of the “Woodstocks”, the so called “Spice” calculators. This model is said to be one of the best pocket calculators ever made, it was the first calculator which offered root finding and numerical integration. In fact, it was the best of its time and was maybe overtaken only by some later models like HP-41C and HP-42S. It is also famous for its error free functionality. There are virtually no bugs known, no bug list available.

But the “Spice” calculators don't contain an ACT chip, HP developed new chips for this series to become more powerful and cheaper in production. There is no 22-pin chip inside any more, therefore the new ACT cannot be used to repair one of these calculators. How is this calculator connected to the “new ACT”?

Well, the firmware of the spice calculators is compatible to the ACT instruction set and can be executed by the new ACT. If the internal ROM of the “new ACT” would contain the HP-34C firmware it could imitate a spice calculator, it could run a spice calculator in a woodstock hardware. Fortunately both series have the same number of buttons, which moreover they are ordered in the same way. Of course, the labels of the buttons are not the same and they have different functions. But if either the buttons were exchanged and/or a vinyl overlay is used, the new ACT could transform a woodstock calculator into the HP-34C or one of the other “Spice” calculators.

Some adjustments had to be made to control the LED display. The “Spice” calculators have only 11 digits instead of 12 and they contain decimals and commas, the “Woodstocks” have no counterpart of commas in their display drivers, the codes had to be translated. Also the key codes had to be be calculated differently and some more details had to be implemented before this adventure got results. But finally it was possible to run the HP-34C firmware on an HP-25 hardware using the “new ACT”.

The mutated calculator is called “**HP-34E**”. E for extended because the HP-34E could add some features even to the best calculator of its time. It adds memory for 18 additional program/register sets of up to 210 steps each. It integrated the “Stopwatch” functionality and, if using the “HP-34E Ir” version, it makes the HP-34C an Infrared printing calculator.

From the complete palette of functions, which you have found in the HP-25E not all of them could be integrated in the HP-34E, One reason was, that as an advanced model it had already build in functions like inserting and deleting program steps or showing mantissa, the other reason was, that there was not enough flash memory space in the ACT to squeeze all functions into it, because the HP-34C firmware was now double the size of the HP-29 which is four times the size of an HP-25. Highest priority was given to store additional programs. At least there was found space for 18 programs in memory, and 18 programs of up to 210 steps sum up to 3990 program steps. As the HP-34C has a dynamically allocation of program steps, consuming some of its registers, while programs grows, also the HP-34E stores programs and registers together. Each program from 0-17 contains a combined program with registers, longer programs have less registers available, shorter programs have more.

Let me notice, that besides printing programs and tracing calculations on the HP82240B Infrared printer and having nearly 4000 program steps, there is another remarkable feature realized: the HP-34E operates four times faster than the original HP-34C if fast mode is activated.

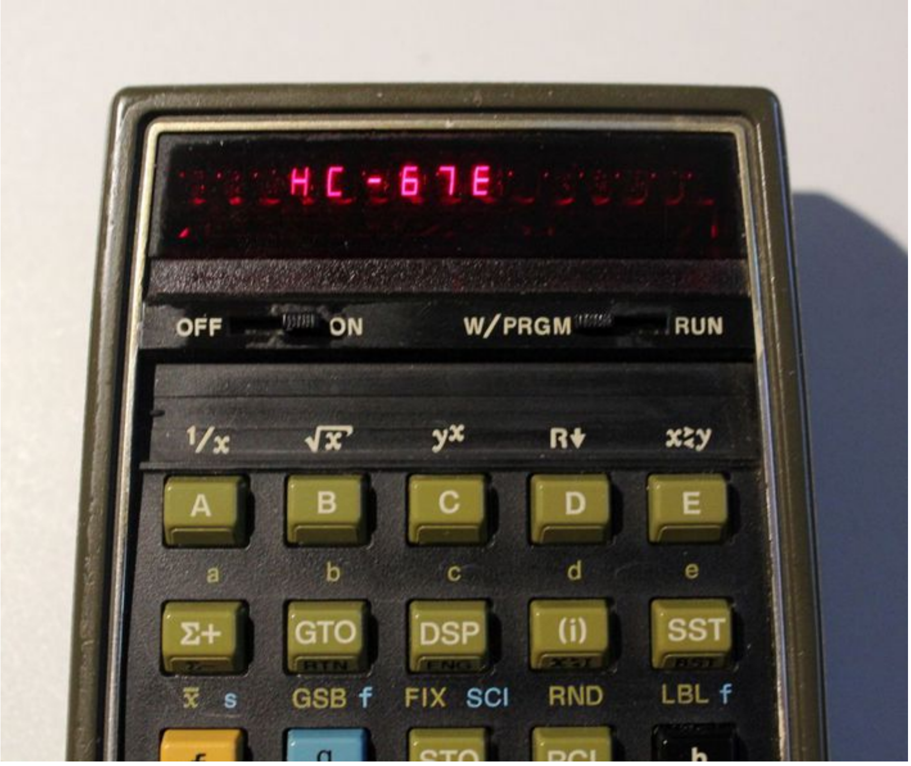
And as everybody knows the HP-34C is a rare and expensive collectible. Now after transplantation the HP-34E ACT into any “Woodstock” hardware you can own an HP-34C, running faster and with printing capability without searching months or years for an not easy affordable working original.

See the “Key sequence summary HP-34E” in Appendix A for available functions.

Infrared printing Version and GPS Version are separate HP-34E Versions, because they could not fit into program space together. “Show ROM”, “Show ROM Checksum”, “Rightadjust”, “Show Memory” and “Constant Collection” are not available in either of the HP-34E versions.

HC-67E

If the HP-34C was the flagship then the HP-67 was the HP aircraft carrier of the early calculator armada. It was the highest end pocket calculator of the third generation and probably the most powerful LED pocket calculator ever made. Its magnetic card reader made it an unlimited programmable calculator. It was capable of storing up to 224 program steps using both sides of a card. Only the desktop twin calculator HP-97 could compete because it had added a printer besides the card reader. But it was not a pocket calculator.

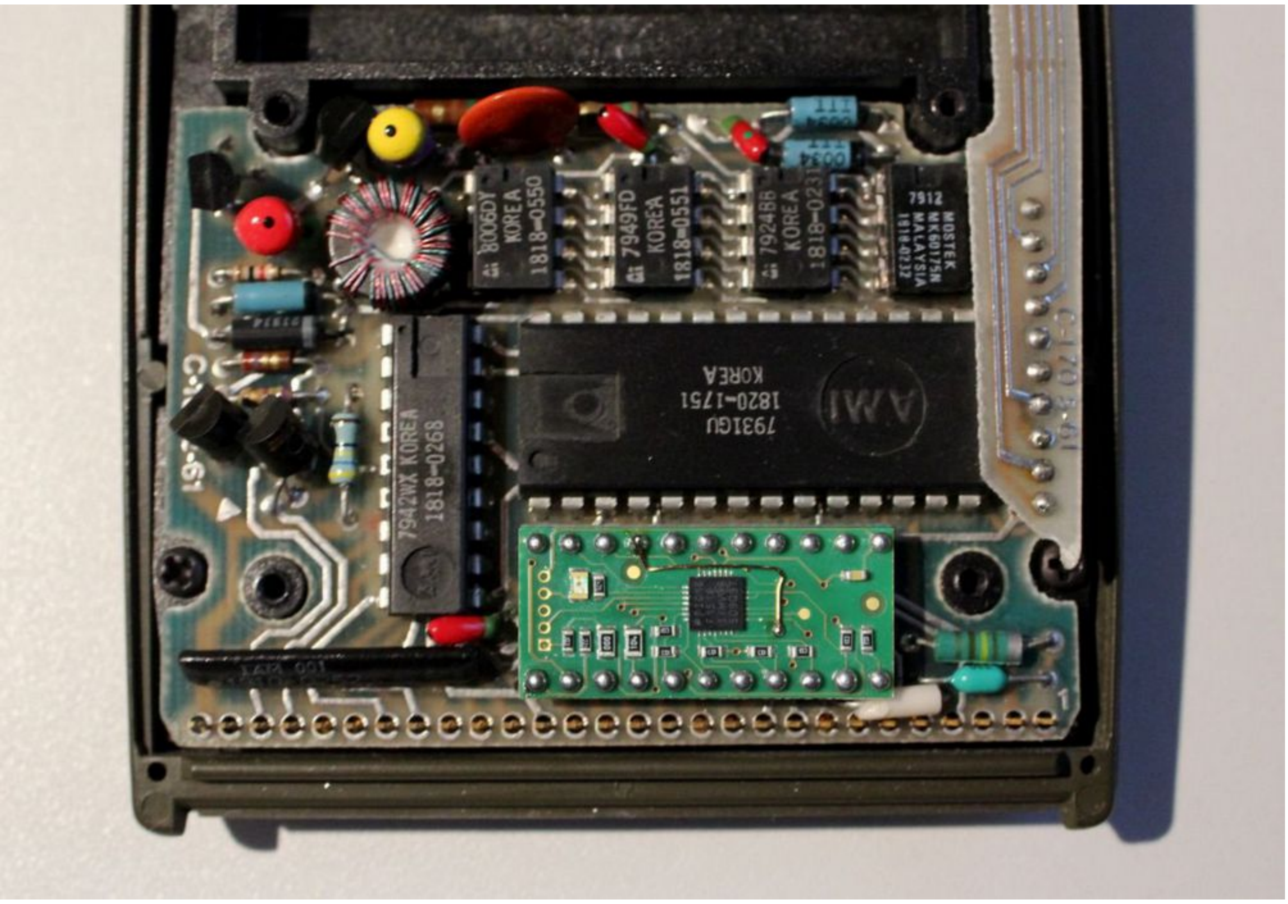


Does the “new ACT” work in an HP-67? Yes, fortunately as you can see in the above image, it does, but with one limit. No, I don't consider the missing letter **“P”** as a limitation, which forces me to show “**HC-67E**” as the calculators name instead of “HP-67E”.

The limit is: it cannot handle the card reader. The interface chip for reading or writing magnetic cards, the so called CRC, does not yet fully communicate with the “new ACT”. But this limit has a solution. Instead of reading and writing cards, the “new ACT” driven HP-67 offers 21 programs of 224 steps each, the equivalent of 21 magnetic cards, altogether more than 4000 program steps, readily stored in ROM, each can be loaded by just a few button clicks instead of inserting magnetic cards. All other functions are identical to the original.

All other functions identical? No, the HP-67E offers an additional Stopwatch, Hexadecimal Conversion and - one more unexpected function: It offers Infrared printing! That makes it unique and thus even more powerful as the HP-97 desktop calculator. And last not least it could run also two times faster than the original.

Other than the HP-34E, which is running only in woodstock calculators, the HP-67E ACT can be plugged into an original HP-67 hardware[1]. The replacement procedure is very similar to the woodstock, you have to open the case, which is described elsewhere, unsolder the ACT and insert the low profile socket with the “new ACT”, programmed with HC-67E firmware.



It is able to read the keyboard and to communicate with the display driver. It cannot yet communicate with the card reader interface. Many or nearly all HP-67 card readers are not working any more, unless they got a repair of their gummy wheel, which typically was pulverized after 35 years. As the card reader repair is not trivial, an alternative is to use the “new ACT” and make the calculator usable again, because it will get “Continuous Memory” instead of the card reader and will not lose its memory after power off.

For accessing the PRGM/RUN switch via the CRC chip, there has to be made a wire connection from the small pad above pin 8 to pin 19, as shown in the image above. (You need HC-67E ACT version 1.03 or higher)

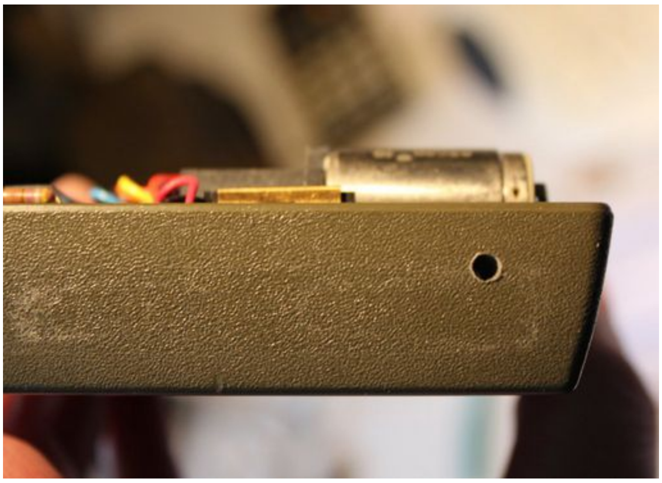
[1] If you inserted the low profile sockets and like to insert the original ACT again, you might have difficulties to close the case, because there is not enough space. You can remove one mm of a plastic strutting of the back cover, then the housing can be closed and screws can be fixed. The “new ACT” should fit without modifications.

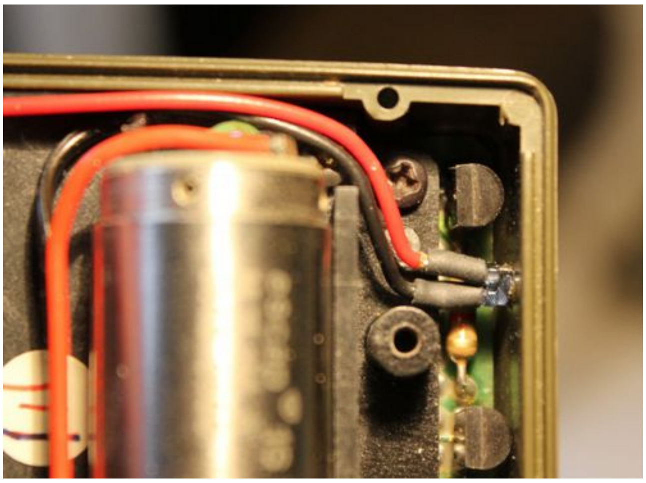
Placing the IR diode inside the HP-67

This chapter shows you how to insert and connect the Infrared diode into an HP-67 calculator for printing on an HP-82240 printer. As you will see it is very easy.

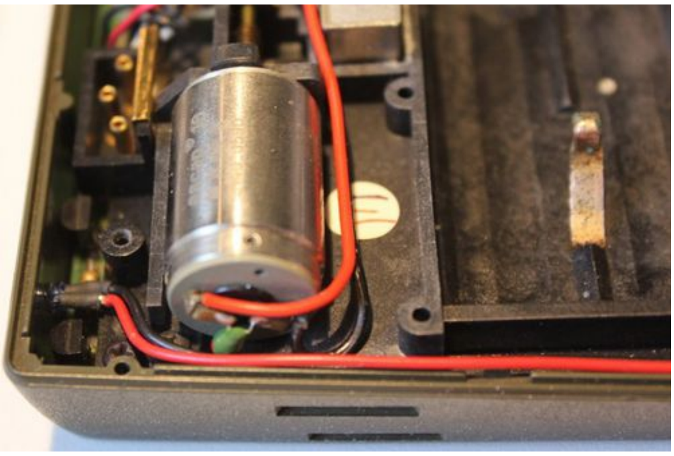
If you own already a modified HP-67E ACT with Ir print interface (that is the additional transistor circuit is installed), you can use it as is for your HP-67. If not, you can add the interface as described in the earlier chapter about the Ir printer or order a newACT for this purpose.

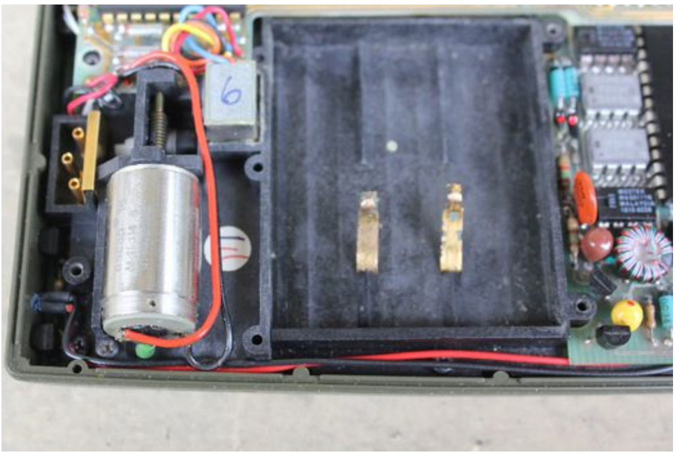
First you have to drill a 3 mm hole into the case at the location shown in the image below. Be careful not to scratch the case with the drill. I recommend to drill a 1-2 mm hole before and then use the final 3mm size drill.



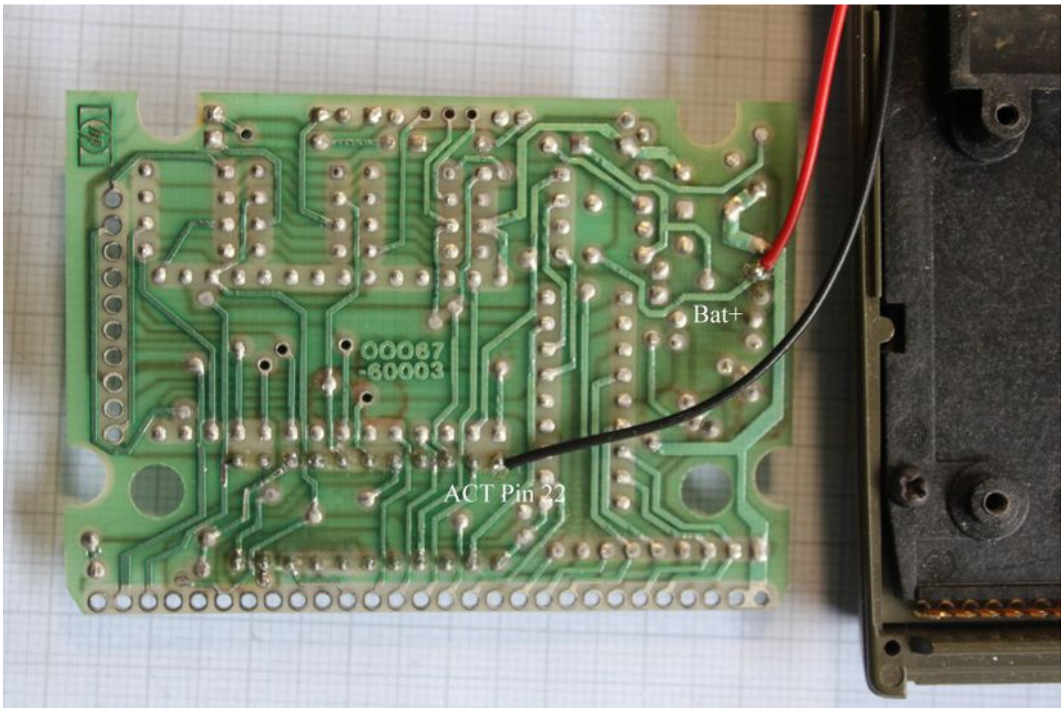


The location chosen in front of the card reader motor is the best suitable space available, it is like it was waiting for the Infrared diode. Place the hole between the left screw of the keyboard plate and the hole where the screw of the back case will be inserted. Attach the red wire to the anode, which is the slightly longer pin, and the black wire to the cathode. Then insert the Infrared diode into the hole. Its orientation can be anyhow, but I recommend the wires to be aligned horizontally like seen above. It is optional to fix them there with some glue. Now the red wire (Anode) and black wire (Cathode) can be lead along the side of the calculators case until they arrive at the mainboard.

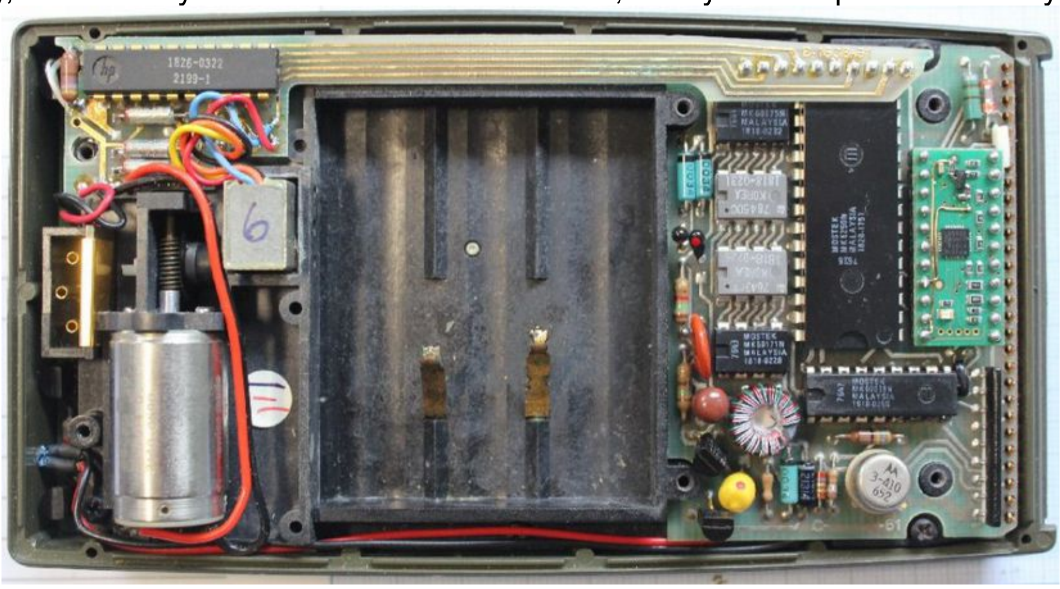




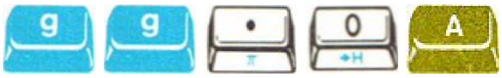
Cut each wire to the right length for soldering them at the two points indicated in the next picture. The red wire should be connected to battery plus, which is accessible by a pin below the power supply part of the main board. The black wire should be connected to the ACT pin 22. Don't forget to check, whether your ACT has a connection from the transistor circuit to this pin, if you built your own interface as described in the Infrared Print chapter. Try whether the mainboard can be placed on the connector prongs without having too much wire left. If the wires are a few mm too long don't cut them further, but you can adjust them by pulling at the case corner towards or over the keyboard screw seen in the picture in the previous page.

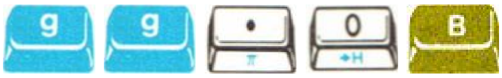


All you have to do now is to place the mainboard at its final place and close the case carefully, check that you dont screw down the wires, if they are not placed correctly.



Now you are ready for printing! You can backup all your HP-67 programs or trace your calculations on thermal paper like with the HP-97. Indeed there is not much difference compared to the desktop HP-97 calculator any more.

Enable print mode by 

Enable trace mode by 

.

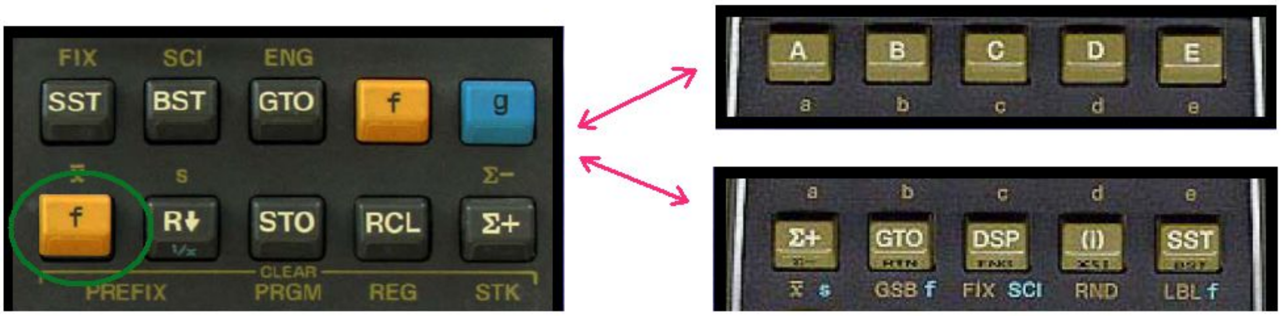
Below you can see the original HP-67 „Moon Lander“ program printed by an HP-67E Ir.

Here is the final result.

HP-67E

If you don't own a real HP-67 then you still can have an HP-67E. There is an ACT version which allows you to run the HP-67 firmware on any woodstock calculator. Like described earlier for the HP-34C the buttons of a HP-25 and its display can be used to show the behaviour of the HP-67 calculator. But as with everything, there had to overcome some obstacles before this goal could be achieved.

First, the HP-67 has one more row of buttons. And these label buttons A-E, a-e are very important. They are function buttons to start individual parts of the program. How can we solve this problem?. The solution is very simple. The first row buttons of the woodstock calculator will take over the first two rows of the HP-67.



If you press any button in the first row normally, the corresponding HP-67 second row button will be executed (buttons Σ+ to SST). But if pressed together with the  =  key (XY on HP-25 keyboard represents the HP-67 f prefix key), this will invoke one of the first row label buttons, and behaves as if button A-E is pressed.

To invoke “A” as one of the upper case labels just press

 + .

To invoke “a” as one of the lower case labels just press prefix  followed by

 + .

To invoke and execute an upper case label A-E repeatedly, for doing sequential calculations many times, just keep the prefix key down and press the first row button as often as you want.

There was never an HP calculator before, which needed to press two buttons simultaneously, but now this is the easy solution for getting five more buttons.

As the  =  prefix button is located in the second row, it can easily be pressed with two fingers of one hand together with a button of the first row of the HP-25 calculator.

Another obstacle did concern the display. The HP-67 has fifteen digits, the woodstocks have only twelve. The solution was not so simple, but effective: the least significant 3 digits of the mantissa will be overwritten by the exponent, if and only if an exponent has to be displayed. This method was invented before by HP, when the HP-21 came into the world as the first woodstock calculator. The “new ACT” analyzes the display contents and decides to rearrange the digits if there is an exponent visible. All previous classic calculators like HP-35 also used to have 15 digits. Now the rightmost digits were used to show dynamically either an exponent or the lower digits of the mantissa, making the program more complicated, but saving 3 digits and thus lowering production costs.

Of course, the HP-67E located in a smaller sized woodstock can display the letter “P” and it supports infrared printing too. And by using a vinyl overlay you could transform any woodstock calculator into the famous HP-67, just by inserting the “new ACT”, programmed with version “**HP-67E**”, making it one of the most powerful if not the best LED pocket calculator ever made.

Image above shows an original HP-67 and HP-67E located in an HP-25 displaying program step 001 LBL 1 in PRGM mode.

See the Appendix A for all available HP-67E/HC-67E functions. The functionality of HP-67E and HC-67E is exactly identical.

HP-35 The Classics

The so called HP Classic series calculators are represented by the HP-35, HP-45, HP-55, HP-70, HP-80, and HP-65, including the first HP pocket calculator ever made, the HP-35.

They do not contain an ACT, it was developed some years later, and their instruction set is not compatible with the ACT. So why can they have something to do with the „new ACT“?

The emulator for the woodstock ACT, based on Eric Smiths “nonpareil”, is not the only emulator the ACT can run. I ported also the classic emulator from “nonpareil” to become part of the “new ACT” and finally could emulate all of the classic calculators on the woodstock hardware platform. In the picture above you can see on the left an emulated HP-35 and on the right the original just after power up showing a single “**0.**” with decimal point.

Again there were the two problems of one more row of buttons and three more digits in the display. They were solved nearly in the same way as in the HP-67E. But as the HP-35 didn't have any prefix key there had to be found another solution than pressing prefix together with one of the first row buttons. The solution was “Long Press”. When pressing the button normally it invokes the second row buttons. When pressed longer than 1/2 second, it invokes the first row buttons.

There are six ACT versions available for download: HP-35E, HP-45E, HP-55E with original stopwatch, and HP-70E HP-80E. Meanwhile also HP-65 ACT is included. All versions just emulate the original calculator without offering additional features or more memory, although at least I made the four stack registers “Continuous Memory” !.

HP-65

Emulating the HP-65, the very first programmable HP calculator, on a Woodstock was a challenge, or at least a lot of work. Perhaps nobody has waited for it, and meanwhile two years have passed since inventing the new ACT, but anyway, with ACT version 1.13 the HP-65 finally can be emulated and run on a Woodstock calculator.

Like the HP-67 it has one more row of buttons, which can be accessed the same was by holding down the f key while pressing A-E.

Because the HP-65 was the very first handheld programmable calculator, it was not yet perfect and memory was very limited. Only 3k ROM code, slightly more than the later basic non programmable scientific HP-21 was projected for even reading magnetic cards and manging 100 program steps. Most of the prefixed functions need two program steps, because the prefix button served as a separate program step, merging program steps into one code line was not so easy with only 3k ROM. And WTH have they thought about not showing the actual program step number? Obviously there was no space for it and it was assumed, that every program, once completely debugged, will never be displayed again to the user. But what if you are the programmer? You have to count every SST keys from 1 to 100 to get track at which line number you actually are. Another missing function is the BST key. There is no way to get back one step. You have to leave PRGM mode and press the RTN key to get to step 00, then switch back to PRGM mode and pressing SST up to 99 times to get to the desired position.

I could not expand these missing features to a better level, I only allowed me to show the actual program step in the display in PRGM mode. And of course, because the Woodstock calculator hardware does not have a card reader, you can store your programs and registers in a library.

If you own the update kit, you can download the HP-65 ACT version 1.13 for trying this famous calculator on your woodstock calculator. It needs some seconds to show **„0.00“** in the display because the power up sequence is longer than normal.

HP-19C

Although the „new ACT“ cannot be used (yet) inside an original HP-19C for repairing this extraordinary machine, you can update your „new ACT“ chip with the HP-19E Ir version (since revision 1.14). If you have installed the infrared printing diode, you will be able to print to the HP82240B printer and you will get exactly the same output on paper as with an original HP-19C calculator.

This model however needed some special thoughts. Although it is the twin calculator of the HP-29C it has a rather different keyboard layout, than any other „Woodstock“ keyboard, and OMG it has one more button if you count them altogether. The PRX button has no counterpart on the woodstock keyboard. All other buttons are identical to the HP-29C although aligned differently. There are three functions to add, which cannot be accessed directly: PRX, PRT STK and SPC. I decided to use the Enter and Sigma button as PRX button replacement. Press

for PRT STK,

for PRT SPC and

for the normal PRX function

Of course these print instructions are also programmable.

But this is not the complete thing. The HP-19C has two sliders with three positions each. The left slider selects between OFF PRGM and RUN mode. This is what we can select when using both sliders on the Woodstock keyboard, so no problem. The right slider however can select between MAN NORM TRACE print modes. How can we simulate this slider? We will use the already available two printflags 10 and 11 to select one of the print modes [1].

Mode	Infrared Print mode Flag 10	Trace mode Flag 11
OFF	0	0
MAN	1	0
NORM	0	1
TRACE	1	1

1 Notice, that NORM hasn't set the print flag 10. This is different to the HP-25E Ir/HP-29E Ir, where flag 10 disables all printing when it is not set.

If you choose OFF, there won't be any printer output, even if you invoke the PRX function. This is different to the original HP-19C, where you cannot disable printing completely. In MAN mode there will be a printout only when pressing a dedicated print function like PRT PRGM or PRX. The difference between NORM and TRACE mode is, that in TRACE mode the result of each calculation is printed automatically, whereas in NORM you have to press the PRX function when you want to see the final result on paper. You can save paper by using NORM mode, because intermediate results are omitted.

Why do we need an HP-29 Ir, when the HP-19E Ir is there? Well, good question, indeed the HP19E is better than the HP-29E Ir. However the HP-19C emulation is not exactly the same as the HP-29C with added printer, it has different row/column codes in PRGM mode, which can be confusing with an HP-29C keyboard because of the different layout, and the FIX SCI and ENG functions are located on different buttons, but everything else is very similar. If you don't mind these differences, you can use the HP-19E as your daily HP-29E calculator with printer, it has all the functions of the HP-29E and has superior printing output, compatible with the HP-19C/29C user manual.

Making the HP-19C possible on a woodstock calculator needed a lot of research and digging into the original HP-19C firmware for implementing the print commands, not to mention that I had to extract the original 5k HP-19C ROM code from an original machine, which was done for the first time. But finally I succeeded. There are two print instructions unique to the HP-19C instruction set, each of them prints the contents of the act_c register. The first command prints up to 15 numbers, which are coded as 4-bit BCD digits. The second print command interprets the act_c register as up to nine 6-bit ASCII characters.

The original PIK chip (Printer Keyboard Interface) is simulated as close as possible by the new ACT. A virtual „Home“ switch, which tells the firmware, when the printer has arrived its home position and the „motor“, which is always printing from left to right, are simulated by internal counters to achieve a timing as if a real motor would be running. During the motor busy time, when it is running from right to left and back, the infrared characters and Line Feed are sent to the HP82240B printer.

HP-01

What can be told about this famous device? I think, it is perhaps the most beautiful wristwatch calculator, which was ever built in the LED era - the legendary HP-01.

When I took my HP-01 last year to a party invitation, I didn‘t need to show it around, it attracted attention to everybody by itself. Nobody wanted to believe me, that it was a nearly 40 year old device. Its impressive keyboard gave it the appearance of a futuristic device. But it was antique. It was furthermore the first and only “Smart Watch” HP ever built, 40 years before this name came into our modern world.

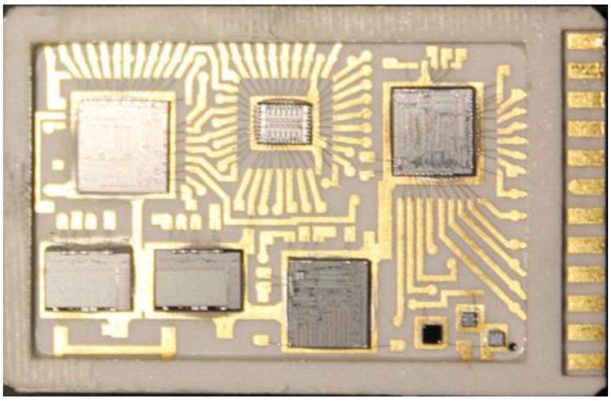
HP-01 stainless steel

As this manual is about the „new ACT“, shouldn‘t you be astonished to read this chapter? No, meanwhile you take everything for granted, don‘t you? The question remains, what links the ACT to the famous watch calculator? Does it contain an ACT? No, there wouldn't even be space for a cut in half ACT in the very small interior of an HP-01. The processor is just a 2x2 mm chip, inaccessible buried deep inside its ceramic hybrid module. There are no pads for sockets or anything else that could be replaced, and its hardware is not at all compatible with the ACT. But you know the answer already. I asked myself: If I had the ROM code and if I had an emulator and if I did understand the hardware, would it be possible to program the new ACT with the HP-01 firmware and run it on a woodstock calculator? Yes, of course! But to achieve this goal I had only to solve three minor problems: getting the ROM code, constructing a new emulator, and simulate the hardware. And finally I got the original HP-01 firmware emulated by the new ACT, running on a woodstock calculator.

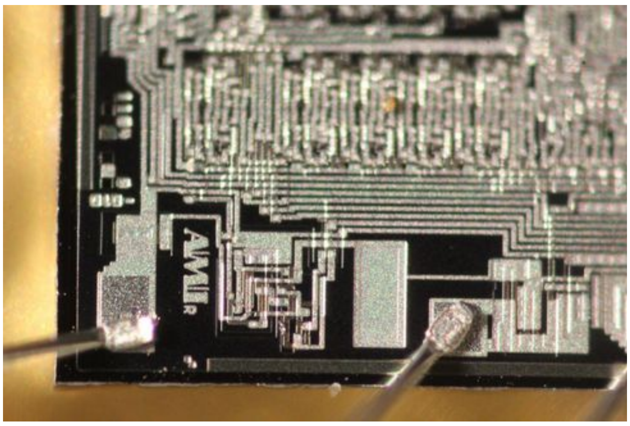
The Technology

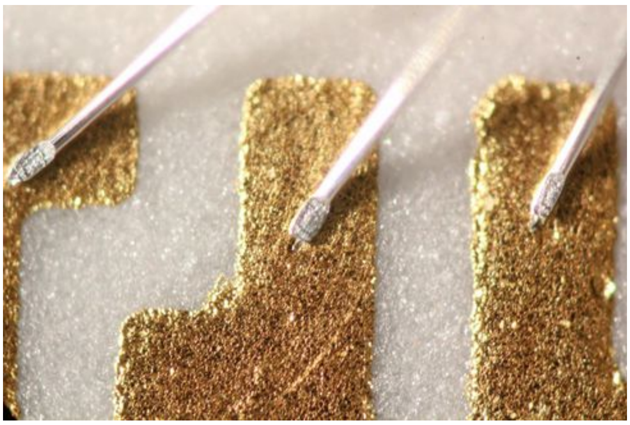
The HP-01 does not belong to the „Classics“, nor to the „Woodstocks“, nor to any other generation of HP calculators. It is a class of its own. And this is true especially in respect to its technology. The hardware is undoubtedly unique, and the firmware does not run under any of the emulators available, it could not be emulated by „nonpareil“, nor by any other emulator.

And even if there was an emulator, the microcode could not just be run. The HP-01 had much more hardware specific functions integrated, which were not part of the firmware. The stopwatch hardware for example will be started and stopped by special instructions forward and backward, and will be displayed even if the processor sleeps. There were some other registers representing the real time clock and date, which count independently of the processor, managing a 200 years range from 1900 to 2100 including leap years, they are spread over different chips, combined in the hybrid module of the HP-01.



I opened the ceramic hybrid module to reveal the six silicon dies.





Images of bond wire connections at the silicon dies and conducting paths.

HP-01 emulator

To build an emulator for the HP-01 firmware from scratch, there needed to be known at least two things: the instruction set, and the instruction flow. Luckily both were documented in the patent specification for the HP-01 in 1977.

The patent specification US4158285 describes the functionality of the HP-01 very precisely. It is a 151 pages document consisting of unbelievable 96 pages hardware schematics, 20 pages functional description, followed by 24 pages ROM code listing and more pages with flow charts and others. I admit, that I could never have built a microcode emulator for the HP-01 without this mighty document. It revealed secrets over secrets of this jewel of engineering, it made the idea of the HP-01 emulator possible.

By reading and comparing the operation codes and instructions of the ROM listing, it became obvious, that it was somehow related to the woodstock ACT, because the same engineers had constructed both. New hardware specific instructions were added, the register size was different, but the fundamental BCD oriented instruction set was quite the same.

After having learned a lot about these early calculators with the making of the new ACT, I was able to understand the instruction set of the HP-01 from the description in the patent. Then I discovered, that it was a not negligible part of the task to emulate also the hardware of the HP-01. Some of the instructions were dedicated to hardware elements, for example meant to copy the actual clock to the display or the clock to the processor and vice versa. By carefully reading the patent specification I got hints, how they could work. For example the single processor instruction „Copy Clock to Display“ does automatically formatting of the time by hardware, it skips the digits, where the colon or dots are shown. And additionally, as the clock register always contains 24h formatted clock data, but the display could show also 12h AM/PM data, it copies only a portion of the time by hardware, the other portion, the hours, were added by firmware and updated every hour by waking up the processor for a short run. There were other hardware elements to copy, like a wakeup of the processor every second, in case the dynamic calculation was started.

When I understood most of these mechanisms little by little, it became clear, that an emulator would need a lot of work, but was principally possible.

But would it be worth to dig into this listing, as the ROM listing was not the original code? It had different assignments of keys and lacks some functions, which were added between the time the patent was published and the date the watch came into the market.

At least if the emulator could run this code it would, if I ever would get it, also run the original HP-01 code. So anyway I decided it should be worth to write an emulator for this code.

I discovered that the ROM code printed there was not executable code. It was the sum of separate compilations of each of the 256 byte ROM pages. To get an executable code, the code had to be copied and linked together like object files and resolved all external references. It was crucial that I was not allowed to make a single typo error, otherwise the code would not run. I used OCR (Optical Character Recognition) software to read the code from the patent and wrote an automatic error correction program, which linked also the pages together. I made as much plausibility tests as possible and cross checking for getting an error free code.

After one week of programming and debugging I had a first HP-01 emulation ready on my PC. Having done the best imagination for all these instructions and having made the software for them, the next step was to check, if it worked. Yes, it did, after many hours of try and error. Then it was a comparably easy step to transfer the code to the new ACT, making it ready for the HP-01 firmware.

LED Display

The beautiful display of the HP-01 has nine digits, the HP-25 has twelve. There should be no problem to place the nine digits somewhere in the middle. Yes, but there was another problem. The HP-01 is the only LED calculator which offered to display a **„:“** colon, it has two dots in each digit, not only one decimal dot like the HP-25. The only way I could see to display something similar to a colon, was to show a combination of minus sign and decimal point. This looked fine and there was no ambiguity resulting of it.

The original HP-01 display normally was switched off all the time, except when you pressed a button, i.e. to show the actual time. After some seconds it vanished again. Obviously this extreme power saving mode is not necessary with a woodstock calculator and its big battery pack, which you can recharge any time. Normally you can leave the display always switched on or press the 1 button if you want to reduce the current consumption by showing only one digit. But you can imitate this behavior and switch on/off the automatic „display off“ function, if you toggle the flag for the standby function with f 0 0 0 or by pressing 0 at power up.

Keyboard mapping

How should I map the HP-01 keys to a woodstock keyboard? Does the number of keys match? The HP-01 had 28 keys, the HP-25 offers 30, at least this was a good starting point.

Of course, most of the functions had to be placed without question to their logical counterparts, like the numbers 0 to 9, decimal point and the operators + - x ÷. Also the C (Clear) key had to become the CLX button. Despite the HP-01 is a non RPN calculator the „=“ key found its counterpart in the ENTER button. The S (Stopwatch) key was logically mapped to the R/S button. The M (Memory) key, which recalls the single storage register into the display, became very natural the RCL key. The ∆ prefix key became the logical equivalent in the f prefix.

Then I decided to place the : colon and / slash key nearby the numbers, because they were used for date and time entry, using the EEX and ∑+ key for them.

D (Date), A(Alarm), T (Time) and R (Recall) were placed in one line on the five top row buttons, R separated only by the f key. And last not least % (Percent) and P (PM/AM) were located at XY and the R↓ buttons, where XY is (coincidentically ?) also the real HP-25 % key.

Later I managed to assign meaningful tasks to the remaining two buttons, by mapping the STO button to the ∆ M function, which stores a number to the memory register with a single keystroke, and the CHS button to ∆ +/-, which made me happy, because you had to press only one button instead of two to negate a number.

Now all 30 HP-25 buttons were logically assigned to the 28 HP-01 buttons. This was the best mapping I could think of - unfortunately a vinyl overlay is not yet available.

HP-01 key	HP-25 key	Function
0-9 .	0 - 9 •	Number entry, decimal point
+ - x /	+ − × ÷	Arithmetic operators
=	ENTER ↑	Equals
∆	f	Prefix key
R	g	Recall function, Stopwatch Reset
D	SST	Date
A	BST	Alarm
T	GTO	Time
S	R/S	Stopwatch
C	CLx	Clear entry
M	RCL	Recall Memory
:	EEX	Time entry
/	Σ+	Date entry
%	x⇄y	Percent Function
p	R↓	PM
∆ M	STO	Store Memory
∆ -	CHS	Change Sign

Keyboard assignment HP-01 HP-25

Extended functions

But there is more: there are extended functions!

The previously described HP-25E function menu is available also in the HP-01 ACT. It is invoked by ∆ 0 followed by a number from 0 to 9. See the description of the function menu in the earlier chapter. You can show the firmware revision and serial number. You can enable standby mode, show the HP logo or operating hours, flash write cycles, invoke the LED test, and you can enter your personal welcome text. Some functions are not available like „show program checksum“, because the HP-01 is a non programmable calculator.

Clock and stopwatch calibration

It cannot be denied, that a pure woodstock calculator doesn‘t contain a real time clock, thus running the HP-01 firmware would show only an approximation of the accurate time. And furthermore, if the calculator is switched off, you will loose your time, because clock and stopwatch will stop counting.

But there are means for improvement.

First the accuracy of the clock and stopwatch can be calibrated. The calibration procedure is the same as previouly described for the HP-25E stopwatch. You start the HP-01 calibration in stopwatch mode „S“ with ∆ 2 and stop after 5 reference minutes again with ∆ 2 key sequence. If the display shows 5:00:00 the calibration was successful. The calibration of the stopwatch must be performed only once, because the calibration value is stored in Continuous Memory. This does not only improve the stopwatch accuracy, but does also calibrate the reference time signal for the clock. The best performance to achieve with this method will be +-1 second per 8 hours.

But there is an even better idea.

HP-01 with GPS module

A previous chapter described the ACT GPS version. The GPS module contains a RTC and it is counting the time even when the calculator is switched off. So if you really want to have a woodstock HP-01 for real use as a watch, then you have to integrate the GPS module into your woodstock calculator. If you have done this, the HP-01 becomes your servant to show the most accurate time you can get on earth, displaying the GPS satellite received UTC with atomic clock precision, it is now more precise than any real HP-01 ever could have been.

The GPS clock will be read into the HP-01 clock register at power up and keeps synchrounous to it, thus shows you always the exact UTC time. If you want to display a different time zone, you can add the time difference to the actual clock as described in the HP-01 manual by T + hours ∆ T.

HP-01 GPS version

The GPS menu can be invoked by ∆ STO. If GPS is available ∆ 1 will update the clock „T“ to UTC. If you lose satellite contact, the time will still be counting but with crystal accuracy. Only the stopwatch will be stopped if you switch off the calculator.

HP-01 scientific

Now I have to offer one more very special feature!

If you program your woodstock calculator with the HP-01 firmware, you will have automatically a scientific calculator like HP-25 and the HP-01 watch working in one machine at the same time.

Running both calculators in one woodstock calculator without reprogramming the ACT, seemed at the first glance a kind of impossibility. There had to be two emulators present inside the ACT chip and two firmwares, each of at least 2k code size. This would not fit into the limited program space. But when we emulate the HP-01 firmware on a real woodstock calculator, the ROM microcode of the HP-01 is programmed in the internal ROM, there is also an external ROM code for this calculator available. Assume you had an HP-25 hardware with intact external ROM chip, then the HP-25 firmware can be read from there and does not consume additional code space. And as the emulators have some similarities, their similarities can be put together, again reducing code space. And by removing part of the huge program library, it was possible to get fitted both into the same ACT.

Finally I realized that it would also be possible to switch between the two calculators without switching off the calculator and keeping their registers, like switching between two multitasking processes. If you switch from one calculator to the other you will see your last calculation results in the display. It took me some time to get this twin combination running, but finally it works.

If you program your ACT with the HP-01 firmware or order this version, you automatically will have the feature of switching to external ROM and this can be an HP-21, HP-22, HP-25 or HP-29, whatever ROM in your calculator will be found.

To switch between internal HP-01 ROM and external ROM press the f key, keep hold down and switch the PRGM/RUN button. Do the same to switch back from any of the mentioned calculators to HP-01 mode. This saves the context of the calculator in Continuous Memory and then switches over to the other calculator. Is it allowed to switch between internal to external ROM any time. If you started the HP-01 stopwatch it will keep running in the background, although you changed to HP-25 mode. Of course, also the clock keeps counting. However if you have a HP-25 program running and switch to HP-01 mode, program execution will be suspended, until you switch back.

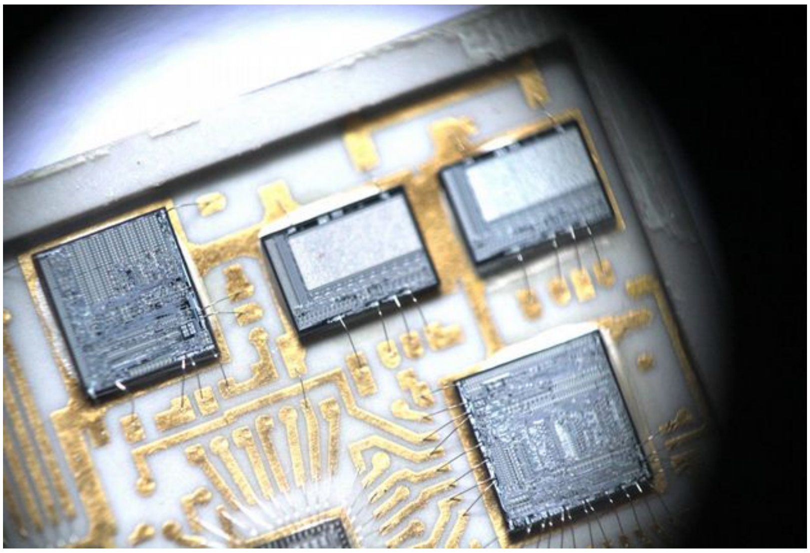
Original HP-01 ROM code

The above mentioned US patent specification contained only the ROM listing of the prototype version of the HP-01. Although it was very helpful to understand the principles, it was clear, that I would need to have the original ROM code. The original ROM code however, delivered in all HP-01 watches to customers between 1977 and 1979, was never published and probably it is stored in a big safe, deep inside the HP archives, written in big letters on it: „Never open!“

HP‘s official website says: „HP discontinued manufacturing the HP-01, its inner workings were destroyed so no one would copy the extraordinarily small package engineering. The HP Archives has a few of the remaining elements.“

Asking HP headquarters for the HP-01 ROM Code would either have been answered very politely negative or answered not at all. So there was no way to get the original ROM code.

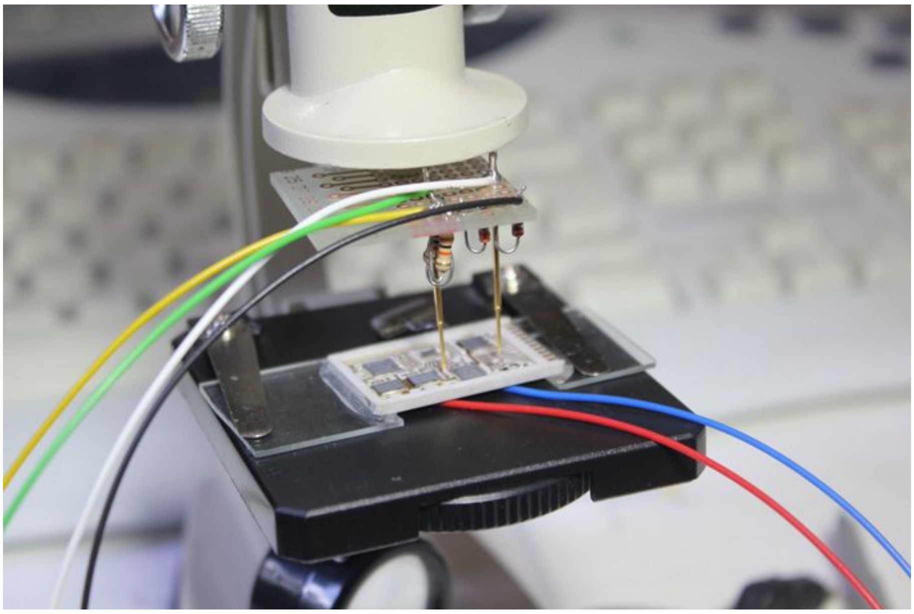
But the ROM code is still there, it is in every working HP-01 watch, partitioned into two ROM chips, deep inside the silicon squares, written in tiny letters on it “Never read!”, which are sealed in the ceramic package, containing also all the other chips connected by tiny bond wires and gold circuit paths.



The two 1k ROMs of the HP-01

Nobody had ever tried to extract the ROM code from there. The obvious reason is the difficulty to get probes attached to the tiny ROMs, and how the probes can read the ROM code actively, but the even higher obstacle are the extraordinary prices each HP-01 achieves as a highly collectible item. And it had to be destroyed to get to the inner parts.

I admit, that I needed more than one year time to think about, until I finally was able to open the sealed ceramic hybrid circuit of my HP-01. Now this was only the first step. How to get near to the contacts and which pins had to be connected and how to read out the ROMs ? Although it was my first intention just to listen with an oscilloscope, I soon had to realize, that eavesdropping the signals of a dead watch was impossible. I tried to see the ROM bits under a microscope, but unlike in HP-35, the HP-01 ROM looked impeccable even under the 40x magnification of a professional Zeiss microscope, it showed nothing that looked like bits. There was no other way than building an interface for the low power circuits, attaching special spring contact probes and writing a spy software, which actively read out the ROMs.



Reading the ROMs

I will not describe how I accidentally ripped off one of the tiny bond wires when attaching a probe and being totally upset, because this could have canceled all my efforts; until I verified under the microscope that the bond wires of the ROMs at least were still intact and I still had a chance to get inside. But now I had to intentionally rip off another bond wire - a rather hard decision. Nor will I describe, how long it needed to build a stable experimental installation for getting the tiny pogo pin probes attached to the ROMs for reading them, using a cheap microscope as an elevator, and how challenging the process of writing the spy software was, but finally I succeeded.

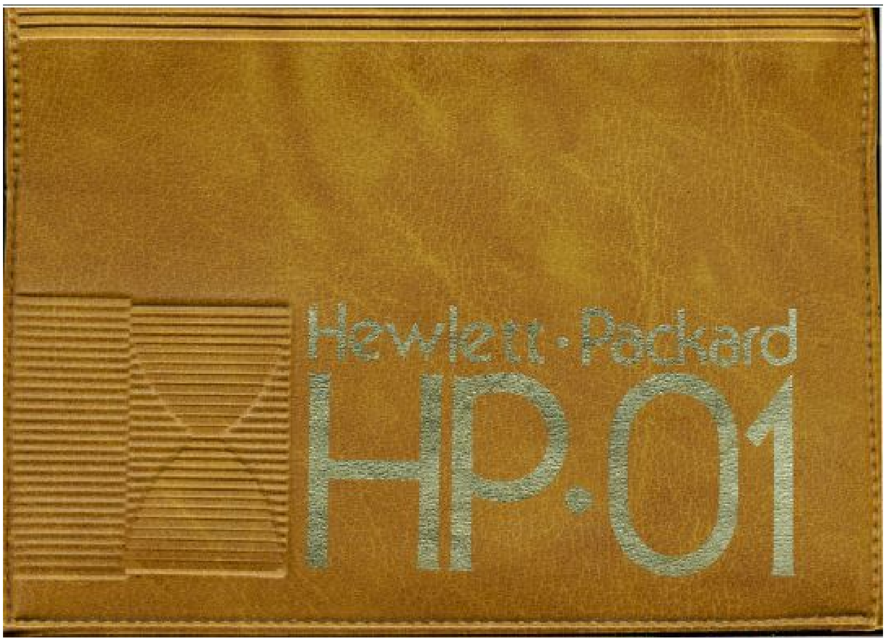
This is the first time the ROM code of the original HP-01 was extracted.

But it would have been too easy, the code was not running at once. I discovered eight new operation codes, which were not used in the documented prototype code and therefore unknown. After some time, I deduced, that they should be direct jump instructions between ROM pages and changed the emulator to what I thought should be right. And bingo, I made a direct hit.

HP-01 Owner‘s Guide

When you will switch on your woodstock HP-01 calculator, **„0.“** will appear in the display. The same will be displayed if you just replaced the batteries of your real HP-01 watch and reset the circuit with the small reset button. It is ready now for exploration of its capabilities.

What you need is the Owner‘s Guide for this unique calculator. It is not included in this manual. To achieve a real leather bound „HP-01 Owner‘s Guide“ you must be very lucky to buy a complete HP-01 watch with all its accessories at an auction.



But it is available as .pdf electronic scan from the „Museum of HP Calculators“ and explains every of its functions in exactly 100 pages.

For a brief introduction, see an overview of the HP-01 functions on the next page.

Key sequence	Function
0-9 .	Numbers and decimal point for number entry
+ - x ÷	The four basic arithmetic operations
%	Percent operator
=	Show result of arithmetic operation
:	Time Entry
/	Date Entry
C	Clear entry to **“0.”**
T	Show actual time
D	Shows actual date
S	Show Stopwatch, press again to start/stop the Stopwatch
R	Lap Time if Stopwatch is running, else reset Stopwatch
A	Show Alarm Time
M	Recall number, time or date, which was stored in memory
∆ M	Store number or time or date in memory
∆ T	Set Time
∆ D	Set Date
∆ A	Set Alarm Time, toggle Alarm active
∆ S	Set Timer for counting backwards
P	Enter PM time
∆ P	Enter AM time
∆ .	Switch between 12h/24h mode if time is displayed
∆ .	Switch between mmddyy and ddmmyy if date is displayed
∆ :	Show day of week
∆ /	Show day of year
∆ +	Set date to 21st century
∆ -	Change sign of number or time
∆ x	Exchange x and y register
∆ ÷	Convert Time to decimal hours
∆ =	Convert decimal hours to Time
T + or - hours ∆ T	Change time zone
D + or - days ∆ D	Adjust date
S x number =	Dynamic calculation
number x percent % =	Calculate percent of number
number + or - percent % =	Calculate net amount and discounts

Continuous Memory

Date and Alarm and the Memory register will be stored in Continuous Memory, and they will reappear when you switch on the calculator the next time. Also the Alarm Active state and the date/time format will be stored and retained. However, if you don‘t use the calculator every day, the date will become invalid, because it can‘t count to the next day if the calculator is not running. And the GPS module doesn‘t deliver the actual date. Normally you just have to add the number of days, the calculator was lying in the drawer, or to enter the actual date directly.

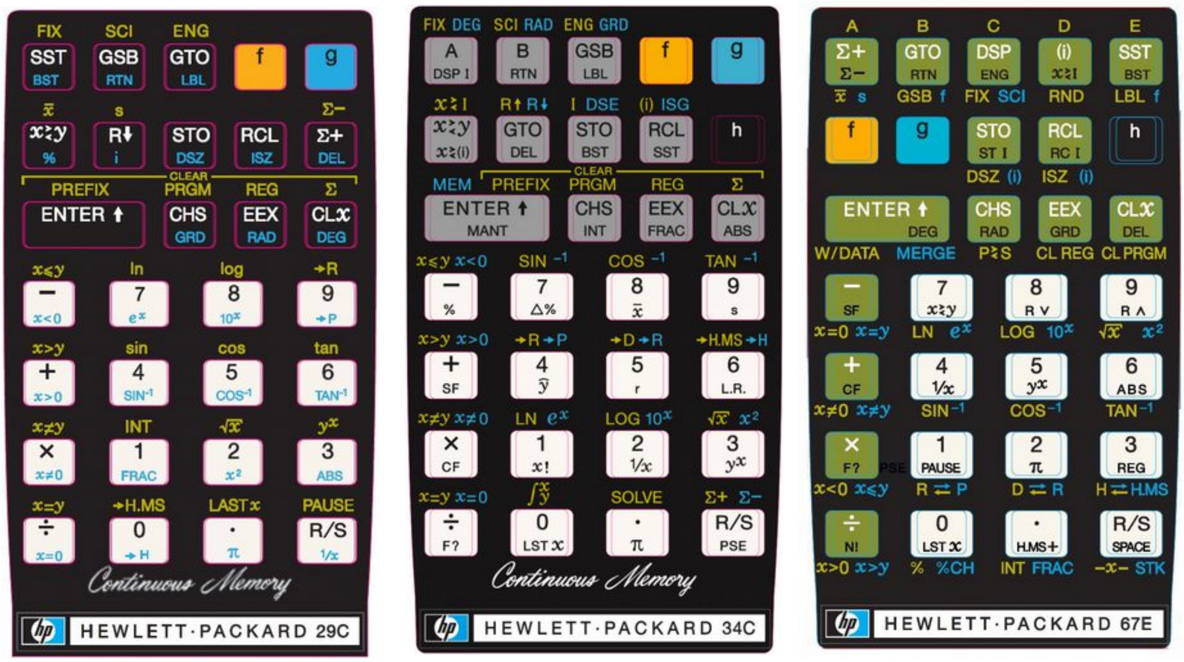
There was a theoretical possibility to show the stopwatch with 8 digits HH:MM:SS.CC when running on a HP-25 hardware, and to show the date with full four digits year like 2016 without much extra work, but I decided to keep the original display format of the HP-01, because it was not the intention to make an upgrade, but to allow a precise emulation of this remarkable early smart watch calculator.

Vinyl Overlays

If you transformed a common woodstock calculator into a higher model like the HP-29C or into another model like HP-34C or HP-67 by programming the HP-34E or HP-67E update, you need an overlay sticker to label each button with its new function. These overlays are available for the HP-29C, HP-34C and HP-67. They are nearly perfect copies of the original labeling. I didn't add any new functions, because they would clearly overload the design and who knows, perhaps you like to use only the original mode.

Normally your donator calculator will be an HP-21 or HP-25 because these are the most common models and therefore more easily to purchase.

To apply an overlay just remove it from its paper and try to still leave the vinyl buttons where they are. Perhaps use a small screw driver or anything similar to get the buttons separated while removing the overlay from its background. Then remove the superfluous frames from the buttons and holes and then place the self adhesive overlay carefully adjusted over the buttons of your calculator. My experience was that it is rather easy to adjust everything before fixing it by pressing firmly. Then you should transfer the button labels calmly one by one, very accurately adjusted, to the buttons. I recommend to bend the buttons by 90 degrees at the edge with your fingers before attaching them.



Three at one go

Having done so, you will now have a more powerful and more handy sized calculator than the original HP-34C or HP-67. This makes is unique. After having attached the vinyl overlay the new model name now is printed on the front side and overlaps the original model name, you cannot see any more, which calculator model is hidden under the new labels, except perhaps by the color of the colored prefix buttons.

I decided not to add the new extended functions to the overlays, because they would overload the design and perhaps you could like to use the original mode, or future software versions could change and add new functions to different buttons. Use the appropriate Quick Guide for invoking the extended functions.

I removed the overlay after some weeks, which was very easy, and nothing happened to the original labels. But there is no experience what happens when you remove the vinyl overlay after months or years. There potentially could be some damage to the original silk screen labels while removing the adhesive glue. Probably it would be helpful to warm up the overlay by a hair dryer before trying to remove it. This is not a problem that affects buttons, because their labels are moulded into the buttons and cannot be damaged. And it also cannot be a problem, when you choose an HP-21 calculator, because it has no printed silk screen labels. But if you use an HP-25 it would be more secure to use an already not perfectly looking unit with already dirty or with partially damaged labels, if you have one in your collection.

PANAMATIK will not take any responsibility for using the yinyl overlays. If you decide to apply them, you agree to do this on your own responsibility, otherwise don't apply them!

But anyway, you most likely will never go back, once you made an HP-34E or a HP-67E calculator from an ordinary HP-21/25.

Boot loader

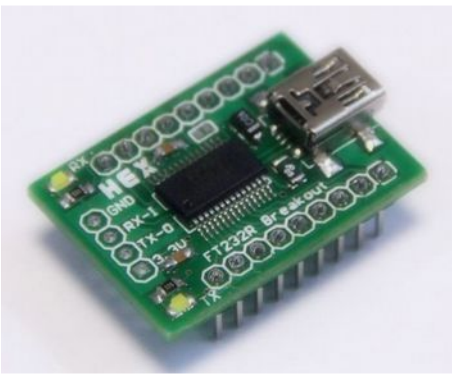
Each “new *ACT”* contains a boot loader. Together with a serial cable you will be able to update your ACT chip with a new firmware version. There could be future versions for this chip, which are able to be a replacement for the HP19C or HP-97 ACTs, however there is no promise, it could be too difficult and I cannot look into the future and there is no time line for it.

1.) Program Update Connector

There is a small row of five through-hole pads, reserved for a connector, that is not assembled. It is used to update the firmware of the ACT, whenever necessary. The connector has 5 pins[1]. The pin-out of the connector is shown in the following table. Pin 1 is marked as a square pad.

Pin	Name	Description
1	/MCLR	Master Clear Low Level Reset Pin
2	RX	Receive Input of ACT 3.3 to 5V
3	TX	Transmit output of ACT 3.3 to 5V
4	VCC +3.3V or +5V	3.3-5V Power supply
5	GND	GND

1 Molex Part number: 53047-0510 PicoBlade connector system, 1,25 mm pitch.



You can use the FTDI FT232RL chip set for Arduino as serial interface or and UM232R module from FTDI.

Connect the following four pins:

GND to ACT GND (Pin 5)

3.3V to ACT VCC (Pin 4)

RX to ACT TX (Pin 3)

TX to ACT RX (Pin 2)

Attention!

Do not connect the ACT directly to a RS232 COM Port with +-12V level. You must use a special adapter for 3,3 to 5V RX/TX signals!

2.) Boot LED

As you might have already recognized, the ACT circuit contains a small red LED. Normally when your calculator is running, it is always switched off to save battery power. However, it comes into action when a firmware update should be performed. When the ACT RX line (Pin 2) is low at power up, it lights up and will be stay switched on as long as ACT RX is held low. After it is released to high level the LED starts flashing slowly, indicating that the boot loader is waiting for data. During data transfer it flashes quickly while receiving data blocks. If transfer stops, it will flash slowly again.

3.) Enter the Boot Loader

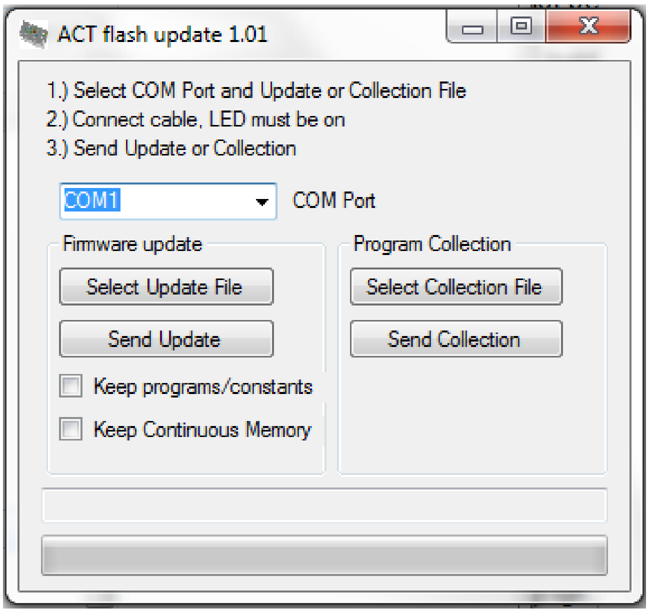
The boot loader is entered by holding ACT RX line low at power up. This can be done by connecting Pin2 (ACT RX) and Pin 5 (GND) or will be done automatically by the ACT Flash Update application. Then the LED will light up, indicating, that the boot loader is entered. If you release the ACT RX line the LED starts flashing slowly, which indicates that data is expected.

If you would switch off now and, without holding ACT RX low, switch on again, the ACT hasn't changed its program, it is still there. But if you run the *actflashupdate.exe* application and upload a new firmware version, the old version will be overwritten.

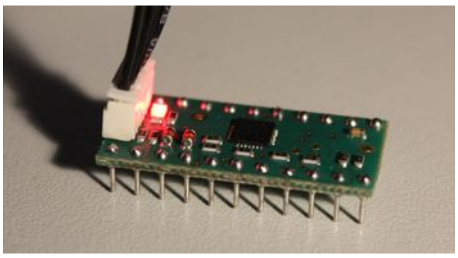
If the update fails half completed, maybe because your cable was disconnected or voltage switched off and the calculator doesn't start up any more, don't worry. The boot loader can never be overwritten and can repeat the upload process until the software transfer is once complete.

4.) The Boot loader application

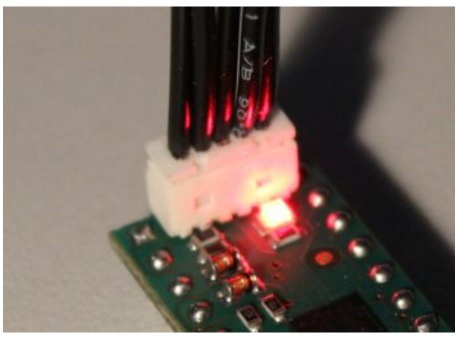
The boot loader application *actflashupdate.exe* is a PC dialog based program, that has buttons, and checkboxes and a combo box for selecting the COM Port. It enables you to select the update file and choose some options for updating your ACT. Furthermore it can send your personal program collection to the ACT.



Before you start *actflashupdate.exe* you should have connected the USB/RS232 converter into the PC, but do not yet connect it to the ACT. When you start and select the COM Port, it pulls the serial TX line low and prompts you to connect the cable now. The cable will automatically power up the ACT and the Boot LED will be on or flashing. If the LED llights up only very short VCC was connected before TX, VCC should be connected at the same time or later. In this case you should connect the ACT again.



Don't solder the connector into the circuit, you will not have enough space to use it inside the calculator. Just plug it into the five pad holes and hold the pins tight against the holes with your fingers, this will give enough connection during the 10 seconds for the update.



Don't update the ACT when it is socketed within the calculator. This will prevent you from damaging when accidentally putting the RS232 adapter cable wrong side in. The image on the left shows you how to plug the connector correctly.

Send update

Now select your Update file by pressing “Select Update File”. The list of available update files with file extension .act can be downloaded from *www.panamatik.de.* Each update file contains the ACT firmware in a special format[1], which the ACT boot loader can receive and understand and flash into its application memory.

Now you can choose to keep programs and constants or keep the “Continuous Memory”, which means keeping only the actual program and actual data register set. However both options are possible only if you update to another version of the same calculator. If you change the calculator type (i.e. from HP-25 to HP29) then keeping programs and “Continuous Memory“ is not allowed, because they have different formats. If you nevertheless check keep programs in this case, your ACT will not work properly.

Now press “Send Update” and the LED will start flashing slowly, the update progress bar will start moving after a few seconds. While updating the LED “flashes” very fast. After programming and verifying it will blink slowly again.

“Send Update” will always preserve your stopwatch calibration, your welcome string, operating time, flash cycles and the user flags, that you actually have set. It also will preserve your serial number[2].

[1] The .act update file format allows to update random areas of flash ROM. With an appropriate update file, it is possible to replace only the internal ROM firmware and leave all your programs untouched.

[2] Only updating from old Versions prior to V1.04 to newer versions will reset the serial number to 00000. Contact PANAMATIK to restore your serial number in this case.

Send program collection

This new feature allows you to keep your entire personal program collection on your PC hard drive or USB stick in a text file, which you can transfer to the ACT instead of typing thousands of program steps in again after you made an update without keeping programs or if you changed the calculator type.

Program collections have a very simple syntax, which is described below. You have to create a personal program collection file for each calculator type once. Program collection files can be made for HP-25E, HP-29E and HP-67E.

If you selected your collection file by “Select Collection File” and no errors were detected, a message shows you how many programs are ready to send. You can connect the cable and press “Send Collection” now.

Before sending a program collection you must ensure, that the ACT is programmed with the same calculator type. *actflashupdate.exe* cannot perform a check, whether the correct firmware is present. If you send a program collection when a wrong calculator firmware is present, the firmware will be overwritten and you have to flash the firmware again with “Send Update”.

Program Collection File Syntax

The program collection file contains only a few keywords and the mnemonics for the program steps, which are described here.

Empty lines or lines preceded by ; (semicolon) are skipped. While using ; you can add comment lines wherever you want. It is allowed to add comments in the same line with program steps.

Each collection file contains programs only for one calculator. It must contain the calculator type "**HP25**", "**HP29**", or "**HP67**" at the beginning before any programs, if no calculator name is encountered it will be assumed HP25.

Each program must be preceded by the keyword "**PROGRAM**" followed by a number from 0-99. However some programs numbers cannot be used, depending on the calculator type.

For HP25 program numbers from 0-9 and 20-99 are allowed, max 49 steps

For HP29 program numbers from 0-9 and 18-20 are allowed, max 98 steps

For HP67 program numbers from 0-9 are allowed, max 224 steps

Programs 0-9 are always stored in fast access program memory which can be recalled by f RCL 0 - 9. Program numbers above 10 however will be stored in the program library and will be recalled by f RCL • 0 0 - 9 9.

The reason why HP25 programs 00-19 and HP-29 programs 00-17 cannot be written to is because they are stored in a special part of memory, which should not be overwritten. This protects also the predefined constants. But as you may have recognized, it is possible to overwrite programs number 20 to 55, which are normally write protected, as they are part of the HP-application programs collection. If you want to restore the original HP- program collection, just send an update without keeping programs/constants

If you use the same program number twice, only the last program will be stored. Unused program numbers will keep the corresponding programs, they won't be cleared

Each HP-25 program from 20-99 is stored together with one constant. Thus you may optionally define a constant, if no constant is given it will be set to 0.00. Use the "**CONSTANT**" keyword for defining the constant, see example for constants. Constants in programs 0-9 are ignored. HP29 and HP67 programs don't support constants for download and they will be ignored too.

The last program step of a program must be followed by "**END**". Program steps are defined by its Mnemonics. Mnemonics are not case sensitive. Unused program steps will be filled by GTO 00 or R/S instructions, depending on the calculator type.

If your program contains more program steps than allowed you will get an error message when loading the collection file. If the Mnemonic is not correct you will get also an error message, which shows you the line the error occurred.

Use the following Mnemonics for entering HP-25 programs

"GTO", "FIX", "SCI", "ENG", "STO", "RCL",

"->H.MS", "INT", "SQRT", "Y^X", "SIN", "COS", "TAN", "LN", "LOG", "->R",

"->H", "FRAC", "x^2", "ABS", "SIN-1", "COS-1", "TAN-1", "e^x", "10^x", "->P",

"0", "1", "2", "3", "4", "5", "6", "7", "8", "9",

"x<y", "x>=y", "x!=y", "x=y", "LASTX", "PAUSE", "CL REG", "CL STK", "/x", "s", "E-"

"x<0", "x>=0", "x!=0", "x=0", "PI", "NOP", "DEG", "RAD", "GRD", "%", "1/x",

"-", "+", "*", ":", ".", "R/S", "ENTER", "CHS", "EEX", "CLX", "x<>y", "R", "E+"

Examples: RCL 7, STO 1, STO + 0, GTO 49, FIX 9,

Use the following Mnemonics for entering HP-29 programs

"R/S", "ENTER", "CHS", "EEX", "CLX", "CL REG", "CL E", "RCL E+",

"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "-", "+", "*", ":",

"->H.MS", "INT", "SQRT", "Y^X", "SIN", "COS", "TAN", "LN", "LOG", "->R", "LastX",

"x<=y", "x>y", "x!=y", “x=y",

"->H", "FRAC", "x^2", "ABS", "SIN-1", "COS-1", "TAN-1", "e^x", "10^x", "->P", "PI" ,"x<0", "x>0", "x!=0", "x=0",

"FIX", "SCI", "ENG", "X<>Y", "R", "E+", "DEG", "/x", "s", "PAUSE", "E-", "RAD",

"%", "1/x", "DSZ", "ISZ", "RTN", "GRD",

"GSB", "GTO", "RCL", "STO", "LBL"

Examples: GSB 0, RCL 1, STO + 1, LBL 1

Use the following syntax for entering HP-67 programs

"R/S", "1/x", "x^2", "SQRT", "%", "E+", "Y^X", "LN", "e^x", "R->P", "SIN", "COS", "TAN", "P->R", "RTN",

"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "ENTER", "CHS", "EEX", ":",

"PAUSE", "N!", "/x", "s", "%CH", "E-", "ABS", "LOG", "10^x", "INT", "SIN-1", "COS-1", "TAN-1", "FRAC", "RND",

"X<>Y", "RDOWN", "CLX", "ENG", "FIX", "-x-", "SCI", "+", "-", "*", "D->R", "R->D", "H->H.MS", "H.HMS->H", "STO(i)", "RCL(i)",

"H.MS+", "SPACE", "STK", "LastX", "W/DATA", "MERGE", "X<>I", "RUP", "PI", "DEG", "RAD", "GRD", "P<>S", "CL REG", "REG",

"x!=y", "x=y", "x>y", "x!=0", "x=0", "x>0", "x<0", "x<=y", "ISZ", "ISZ(i)", "DSZ", "DSZ(i)",

"CF", "SF", "F?", "DSP", "RCL", "RC I" "STO", "ST I" "GSB", "GTO", "LBL",

Examples: RCL 0, STO + 0, GSB A, GTO e, LBL 1, STO + (i), CF 0, SF 3, F? 0 or F0?, DSP 2, STO * 5, STO (i), FIX, SCI

To better understand how you write your program collection file, I will give you two short examples.

Program Collection File example for HP-25

```
; this is a comment line
; start of HP-25 program collection file
HP25

PROGRAM 20       ; simple HP-25 counter example, store to Program location 20
CONSTANT -1.234e2   ; this constant can be accessed by f RCL . 20
1
+
PAUSE    ; show X register
GTO 01

END ; end of program 20

; end of HP-25 program collection file
```

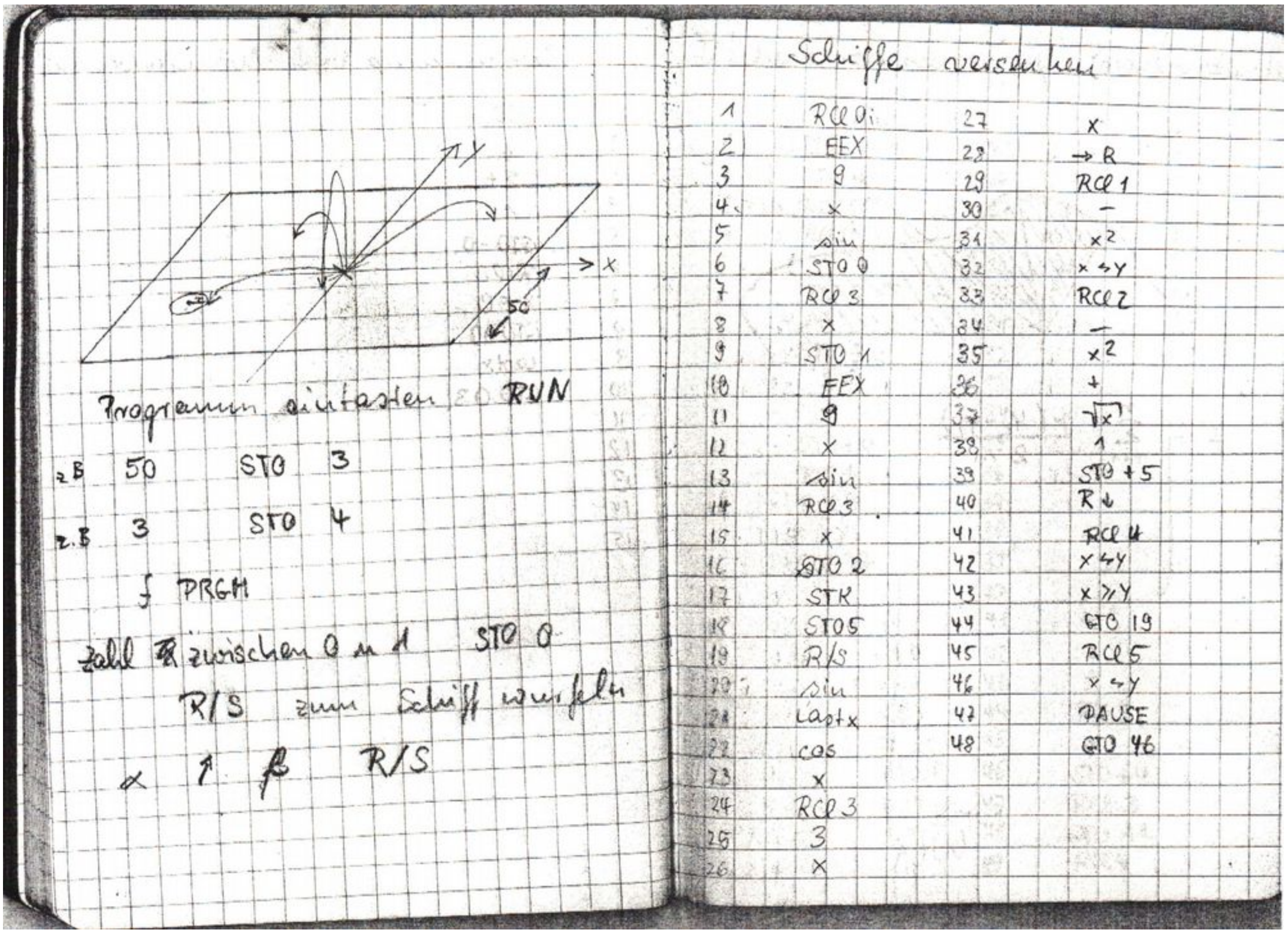
Program Collection File example for HP-29

```
; simple HP-29 counter example, store to Program location 1
HP29

PROGRAM 1
LBL 0
2
STO + 1
RCL 1    ; recall register
PAUSE
GTO 0

END
; end of HP-29 program collection file
```

Battleships for HP-25

I wrote this battleships program for HP-25 in 1978 (Battleships in German: Schiffe versenken)

I scanned it from my program collection booklet, because it needs this little drawing to understand how to fire. I know there is another more sophisticated battleships for HP-25 to find somewhere, but this is my original version and you can play it just for fun.



1.) Load the program from the library with [f] [RCL] [•] [0] [0]

2.) Enter the width of the battlefield in meter to storage register 3 [STO] [3]

3.) Enter the radius, that will severely hit the ship in meter to register 4 [STO] [4]

4.) Switch to RUN Mode, reset the program with [f] Clear PRGM.

5.) Enter a random number from 0 to 1 into storage register 0 [STO] [0]

6.) Press . This will place the ship somewhere in the battlefield, somewhere you don't know.

7.) Now you enter two angles alpha and beta (in Degrees of course, when the calculator is in Degrees mode. Alpha is given from 0-360° and gives the direction where you want to shoot. Beta gives the elevation angle of your artillery gun. With 45 degrees you get the biggest distance, with 90 degree you will fire into the sky above yourself and you will get your own ship, which is in the center of the battle field, but it needs some seconds until you are hit.

8.) Press and repeat from step 7.) as often as you want.

After every shot you will be shown the distance from the ship that you missed, because you can't see it in the mist, but somehow you get information about the miss distance, probably from the ocean wave patterns. You will retry to get closer and closer, because you know the ship cannot move in the mist without getting trouble in the surrounding reefs.

9.) Finally when you severely hit the enemy ship, you will get a flashing display, showing the last hit and the number of times you needed for it. The X and Y coordinates of the ship, that is about to sink, are stored in Registers 1 and 2, you know them exactly now, because the mayday calls reveal them to you.

Enjoy!

Bernhard Emese,

ACT Repair Kit
October 2014 to January 2015, updated 1.09 Dec 2016, updated 1.12 July 2017

Appendix A

1.) Key sequence summary *HP-21E*

Key sequence	Mode	Description
DSP	RUN	Enter stopwatch mode
0	RUN	Enter Hexadecimal entry
STO 0 - 9	RUN	Store register set in Storage Memory 0 to 9
RCL 0 - 9	RUN	Recall register set from Storage Memory 0 to 9
RCL • 9 9	RUN	Recall constant 0-99 from Constant Collection
STO • 9 9	RUN	Store constant 20-99 to Constant Collection
ENTER ↑	RUN	Show mantissa/PrintX
ENTER ↑	RUN	Paper Advance
• 0 - 9	RUN	Enter function 0-9 (see function table)

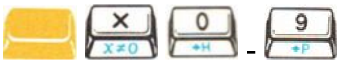
2.) Key sequence summary *HP-22E*

Key sequence	Mode	Description
Σ+	BEGIN/ END	Enter stopwatch mode
0	END	Enter Hexadecimal entry
ENTER ↑	END	Show mantissa/Print X
ENTER ↑	END	Paper Advance
RCL	END	Show registers 0 to 9
• 0 - 9	END	Enter function 0-9 (see function table)

3.) Key sequence summary *HP-25E*

Key sequence	Mode	Description
g R/S	RUN	Enter stopwatch mode
f R/S	RUN	Enter Hexadecimal entry
f STO 0 - 9	RUN	Store register set in Storage Memory 0 to 9
f RCL 0 - 9	RUN	Recall register set from Storage Memory 0 to 9
f STO • 9 9	RUN	Store constant 20-99 to Constant Collection
f RCL • 9 9	RUN	Recall constant 00-99 from Constant Collection
f ENTER↑	RUN	Show mantissa/Print X
f ENTER↑	PRGM	Show program checksum
g ENTER↑	RUN	Show memory/Paper Advance
g RCL	RUN	Show registers 0 to 7
g Σ+	RUN	Show statistical registers 3 to 7
g STO	PRGM	Insert program step
g RCL	PRGM	Delete program step
g SST	PRGM	Show program steps
g SST	RUN	Single Step Debug
g BST	RUN/PRGM	Undo stack registers/Undo program step
GTO • 4 9	PRGM	Goto program step 00 to 49
f g	PRGM	Switch to RUN mode
g f	RUN	Switch to PRGM mode
f STO 0 - 9	PRGM	Store program 0 to 9 to „Program Memory“
f RCL 0 - 9	PRGM	Load program 0 to 9 from „Program Memory“
f STO • 0 0	PRGM	Store program 55 to 99 to „Program Library“
f RCL • 0 0	PRGM	Load program 00 to 99 from „Program Library“
g GTO 0 - 9	RUN/PRGM	Enter function 0-9 (see function table)

4.) Key sequence summary *HP-27E*

Key sequence	Description
	Enter stopwatch mode
	Enter Hexadecimal entry
	Show mantissa/Print X
	Paper Advance
	Enter function 0-9 (see function table)
	Show registers 0-9
	Store register set in Storage Memory 0 to 9
	Recall register set from Storage Memory 0 to 9

5.) Key sequence summary HP-29E

Key sequence	Mode	Description
g g R/S	RUN	Enter stopwatch mode
f R/S	RUN	Enter Hexadecimal entry
f STO 0 - 9	RUN	Store register set in Storage Memory 0 to 9
f RCL 0 - 9	RUN	Recall register set from Storage Memory 0 to 9
f RCL • 0 0	RUN	Recall constant 0-39 from Constant Collection
f STO • 0 0	RUN	Store constant 00-39 to Constant Collection
f ENTER ↑	RUN	Show mantissa/Print X
g ENTER ↑	RUN	Paper Advance
g g RCL	RUN	Show registers 0 to 9
g g SST	PRGM	Show program steps
g g SST	RUN	Single Step Debug
GTO • 9 8	PRGM	Goto program step 00 to 98
f g	PRGM	Switch to RUN mode
g f	RUN	Switch to PRGM mode
f STO 0 - 9	PRGM	Store program 0 to 9 to „Program Memory“
f RCL 0 - 9	PRGM	Load program 0 to 9 from „Program Memory“
f RCL • 0 0	PRGM	Load program 00 to 99 from „Program Library“
f STO • 5 5	PRGM	Store program 55 to 99 to „Program Library“
g g GTO 0 - 9	RUN/ PRGM	Enter function 0-9 (see function summary)

6.) Key sequence summary — *HP-34E*

Key sequence	Mode	Description
g g R/S	RUN/ PRGM	Enter stopwatch mode
g x≷y	RUN	Enter Hexadecimal entry mode
f ENTER↑	RUN	Print X (HP-34E Ir)
g ENTER↑	RUN	Paper Advance (HP-34E Ir)
g EEX	RUN	Show registers 0 to 9
g g RCL	PRGM	Show program steps
g g RCL	RUN	Single Step Debug
f g	PRGM	Switch to RUN mode
g f	RUN	Switch to PRGM mode
f f RCL • 0 0	RUN/ PRGM	Load program/regs 00 to 17
f f STO • 0 0	RUN/ PRGM	Store program/regs 00 to 17
g • 0 - 9	RUN/ PRGM	Enter function 0-9 (see function table)

Repeat keys — *HP-34E*

Key	Mode	Description
x≷y	RUN	Show XY registers
g RCL	PRGM	Show program steps forward in PRGM mode
g STO	PRGM	Show program steps backward in PRGM mode

7.) Key sequence summary *HP-67E*

Key sequence	Mode	Description
g g R/S	RUN/PRGM	Enter stopwatch mode
h CLx	RUN	Enter Hexadecimal entry mode
f f STO 0 - 9	RUN	Store register set in Memory
f f RCL 0 - 9	RUN	Recall register set from Memory
f ENTER↑	RUN	Show Mantissa/Print X (HP-67E Ir)
g ENTER↑	RUN	Paper Advance (HP-67E Ir)
f R/S	RUN/PRGM	-x- Print X (HP-67E Ir)
g R/S	RUN/PRGM	STK Print Stack (HP-67E Ir)
h R/S	RUN/PRGM	SPACE Paper Advance (HP-67E Ir)
g g SST	PRGM	Show program steps
g g SST	RUN	Single Step Debug
f g	PRGM	Switch to RUN mode
g f	RUN	Switch to PRGM mode
f f STO • 0 0	PRGM	Store program 00 to 10 to Library
f f RCL • 0 0	PRGM	Recall program 00 to 10 from Library
f f STO 0 - 9	PRGM	Store program to Memory
f f RCL 0 - 9	PRGM	Recall program from Memory
g g • 0 - 9	RUN/PRGM	Enter function 0-9

Repeat keys *HP-67E*

Key	Description
SST	Show program steps forward in PRGM mode

8.) Key sequence summary *HP-01E*

Key sequence	Description
R/S	S Stopwatch
SST	D Date
BST	A Alarm
GTO	T Time
f	Δ Prefix button
g	R Reset stopwatch
x≷y	% Percent
R↓	p PM
f R↓	a AM
STO	Δ M Store to Memory
RCL	M Recall Memory
CHS	Δ +/- Change Sign
EEX	: Colon Time entry
Σ+	/ Slash Date entry
CLx	C Clear Entry
ENTER ↑	= Show result
0 - 9 •	Number entry
+ − × ÷	Arithmetic operators
f EEX	DW Day of Week
f Σ+	DY Day of Year
f −	+/- Change Sign
f +	21st century
f ×	↔ exchange X Y
f ÷	T→ Time to Decimal
f ENTER ↑	→T Decimal to Time

9.) Special keys *HP-01*

Key	Description
f 0 . 0 - 9	Enter Function Menu
f STO	Enter/Leave GPS mode
f 1	Show UTC (from GPS Module)
f 2	Stopwatch Calibration Start/Stop
f + PRGM RUN	Switch between HP-01 and HP-25 (External ROM)

10.) Function summary

Function	Description
g GTO	Enter Function Menu (HP-29E g g GTO)
.	Show firmware revision and serial number
0	Toggle flags
1	Show calculator logo
2	Show ROM code
3	Show ROM checksum
4	Show program number and checksum
5	Show available Memory
6	Show operating time
7	Show Flash write cycles
8	LED Test
9	Enter Welcome string

11.) Power up keys

Key	Description
.	Show firmware revision and serial number
0 - 9	Toggle Flags 0 to 9
CLx	Toggle between Original mode and Standard or Extended mode
EEX	Toggle between Internal ROM and External ROM firmware
R/S	Toggle between slow and fast execution if internal ROM selected
x≷y	Toggle between Normal mode and Ir printing mode

12.) Stopwatch keys

Key	Description
g R/S	Enter and leave stopwatch mode (HP-29E g g R/S)
R/S	Start and stop stopwatch timer A or B
CHS	Start and stop stopwatch timers A and B simultaneously
CLx	Reset stopwatch timer A or B
x≷y	Toggle between stopwatch timer A and B
ENTER ↑	Game watch toggle between player A player B
EEX	Start and stop calibration
STO 0 - 7	Store lap time 0-7
RCL 0 - 7	Recall lap time 0-7

13.) Repeat keys

Key	Description
R↓	Show stack
x≷y	Show XY registers
SST	Show program steps forward in PRGM mode
BST	Show program steps backward in PRGM mode
+	Show Constant Summation
×	Show growth of value

14.) Print keys

Key	Mode	Description
g ENTER↑	RUN	Paper Advance (HP-21E ENTER↑)
f ENTER↑	RUN	Print X (HP-21E ENTER↑)
g SST	PRGM	Print Program (HP-29E g g SST)
g GTO 0 x≷y	RUN/PRGM	Toggle Print Mode enable
g GTO 0 R↓	RUN/PRGM	Toggle “Trace mode” flag

15.) GPS keys

Key	Mode	Description
g (g) STO	RUN/GPS	Enter/Leave GPS Mode
0 - 7	GPS	Select value to display
ENTER ↑	GPS	Next value to display
R/S	GPS	Automatic value display
EEX	GPS	GPS power On/Off
CLx	GPS	Stop dynamic storage to register 0-7
CHS	GPS	toggle between knots and km/h
x≷y	GPS	Show Position Latitude, Longitude
STO 0 - 7	GPS	Store value to register 0-7
f STO 0 - 7	GPS	Store value to register 0-7 dynamically

16.) PRGM/RUN switch

Switching PRGM RUN in either direction saves the actual program and registers in Flash *"Continuous Memory"* if they have changed since they were saved last time. Also leaves Stopwatch, Hexadecimal Entry and GPS mode.

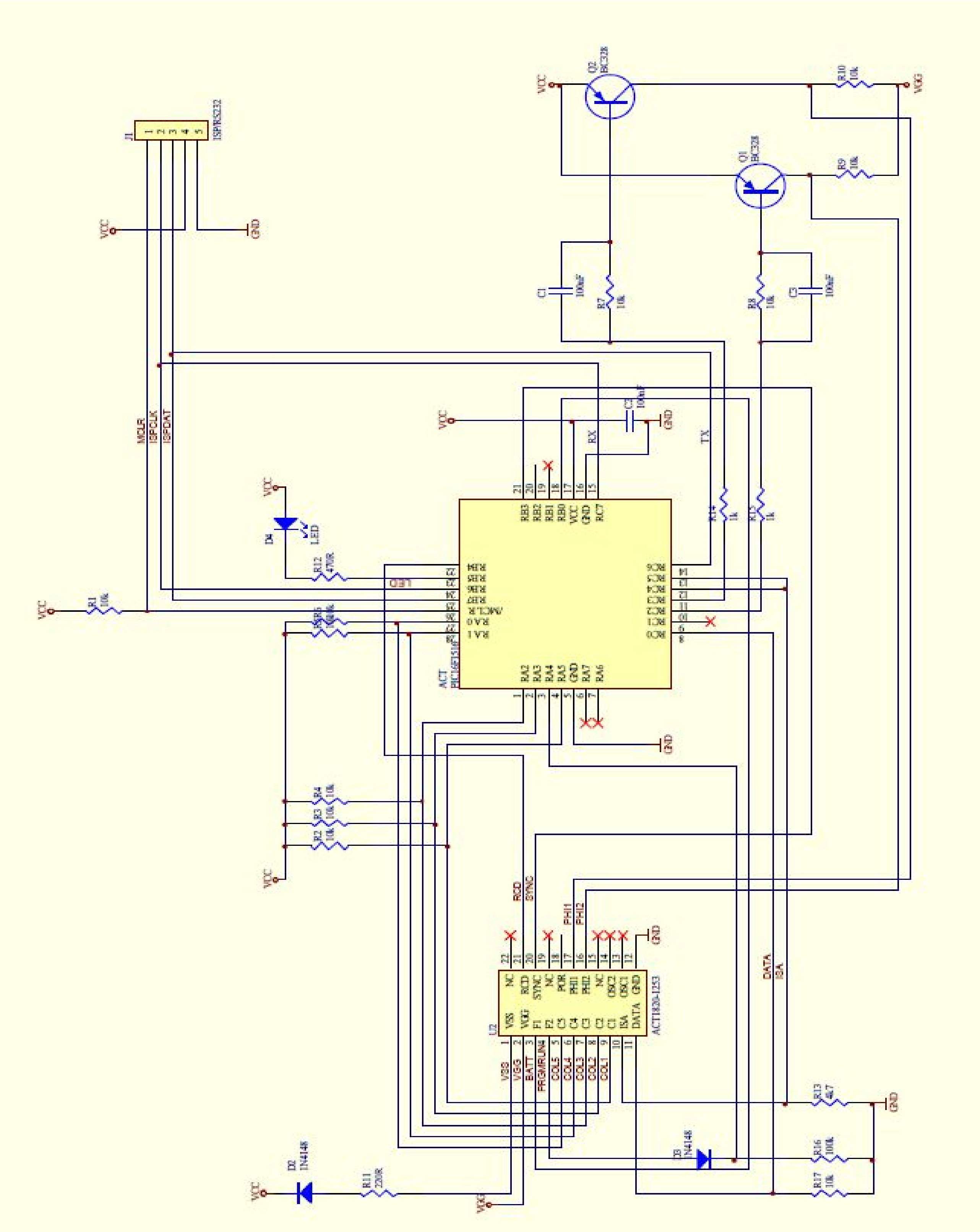
Appendix B

1.) ACT Pinout

Pin	Name	Description	Pin	Name	Description
1	VSS	positive power supply 6,4Volt	22	n.c	
2	VGG	negative power supply -12Volt	21	RCD	Reset Counter Display
3	Batt	Battery Low indicator	20	SYNC	Synchronisation signal
4	PRGM /RUN	Program/Run switch	19	n.c	
5	Col5	Keyboard column 5	18	POR	Power On Reset
6	Col4	Keyboard column 4	17	Phi1	Bus clock one
7	Col3	Keyboard column 3	16	Phi2	Bus clock two
8	Col2	Keyboard column 2	15	n.c	
9	Col1	Keyboard column 1	14	OSC	720 kHz Oscillator
10	ISA	Main Communication line	13	OSC	720 kHz Oscillator
11	DATA	Data Communication line	12	GND	Ground 0 Volt, negative battery

If you measure while the new ACT is running, you should measure the above DC Voltages at Pins 1,2, and 12. You should measure about 3,2 kHz at pin 20 SYNC and 180kHz on either Phi 1 and Phi 2. Pins13/14 are not used. You should measure 720 kHz only with the original ACT on both pins. POR must be low level.

2.) ACT Schematic



3.) Key Codes

The layout of the keyboard in Woodstock calculators is different from model to model. You might find it useful to know which code is generated by which key pressed when the program is running. The ACT contains an instruction, that reads the keyboard and jumps indirectly indexed by the key code into a program location, where the key is processed, without using “if then else” compare instructions. The whole ROM code contains this instruction only once. It is told by some HP engineers, that without the ingenuity of this instruction the HP-25 would not have been possible.

HP-21 Key Codes

0264,0263,0262,0261,0260	1/x	SIN	COS	TAN	g
0104,0103,0102,0101,0100	x<>Y	R↓	e^x	STO	RCL
0324,0322,0321,0320	ENTER	CHS	EEX	CLX	
0144,0143,0142,0141	-	7	8	9	
0244,0243,0242,0241	+	4	5	6	
0164,0163,0162,0161	x	1	2	3	
0224,0223,0222,0221	/	0	.	DSP	

HP-22 Key Codes

0263,0262,0261,0260,0264	n	i	PMT	PV	FV
0103,0102,0101,0100,0104	XY	R↓	STO	RCL	f
0323,0321,0320,0324	ENTER	CHS	%	CLX	
0143,0142,0141,0140	-	7	8	9	
0243,0242,0241,0240	+	4	5	6	
0163,0162,0161,0160	x	1	2	3	
0223,0222,0221,0220	/	0	.	Σ+	

HP-25 Key Codes

0263,0262,0261,0260,0264	SST	BST	GTO	f	g
0103,0102,0101,0100,0104	x<>Y	R↓	STO	RCL	Σ+
0323,0321,0320,0324	ENTER	CHS	EEX	CLX	
0143,0142,0141,0140	-	7	8	9	
0243,0242,0241,0240	+	4	5	6	
0163,0162,0161,0160	x	1	2	3	
0223,0222,0221,0220	/	0	.	R/S	

HP-27 Key Codes

0263,0262,0261,0260,0264	y	x	%	f	g
0103,0102,0101,0100,0104	x<>Y	R↓	STO	RCL	yx
0323,0321,0320,0324	ENTER	CHS	EEX	CLX	
0143,0142,0141,0140	-	7	8	9	
0243,0242,0241,0240	+	4	5	6	
0163,0162,0161,0160	x	1	2	3	
0223,0222,0221,0220	/	0	.	Σ+	

HP-29C Key Codes

```
0263,0262,0261,0260,0264       SST   GSB    GTO    f      g
0103,0102,0101,0100,0104       x<>Y  R↓     STO    RCL    Σ+
0323,0321,0320,0324            ENTER CHS    EEX    CLX
0143,0142,0141,0140            -     7      8      9
0243,0242,0241,0240            +     4      5      6
0163,0162,0161,0160            x     1      2      3
0223,0222,0221,0220            /     0      .      R/S
```