

Do-It-Yourself RPN Calculator Hardware

By Richard Ottosen

Some history

I have been complaining that HP no longer makes good RPN calculators. The last good one really meeting this description is the no-longer-available HP-32SII. I view the HP-33S, "Cell phone calculator", as a sad joke.

I am not alone in this thinking. One friend of mine actually went to eBay to get a backup calculator in case his quit working or was lost. I have talked to Eric Smith, as well, about the excessive cost of replacing my HP-41C if I had to do it on eBay.

Eric was engaged in simulating the operation of the RPN calculators from HP's classic era. He was getting very knowledgeable of them in order to get his simulations to run the actual HP calculator code. His method was to write a simulation of the HP microcode in a high level language running under Linux on a PC.

It occurred to me that if Eric could make a simulator that behaved precisely like the real thing on a PC then maybe the same could be done on a hand-held platform. I considered getting him to port his simulator to one of the palm sized computers but I could not figure out how to make a touch screen feel like a real HP calculator.

You can see how desperate I was. Finally, I decided that custom hardware would have to be done. Fortunately, I convinced Eric that he should create a version of his simulator for the custom hardware. This marks the beginning of the Do-It-Yourself RPN (DIY-RPN) calculator.

The feel of an HP calculator

An important requirement is the "feel" of the calculator. A touch screen, such as on a PDA, has no tactile feedback so it can never feel like a good calculator with real keys that go click. There were some fanciful ideas such as molding or machining an overlay for a PDA that would have the key graphics and -- more importantly -- the feel of clicking keys.

I will sidetrack for a moment here to mention how I use an HP calculator. I hold it in my left hand and use my thumb to press the keys. This leaves my right hand free to write notes. This makes the width of the calculator case very critical. If it is too wide then I cannot reach all the keys. The keys must also have a very distinct snap and click to let my thumb know the press worked as expected. My HP-25 met these requirements and my HP-41C does as well.

Another important part of the operation of a calculator is that it has many keys dedicated to common functions. This would not be possible on a touch screen or a small alphanumeric keyboard. The specialized keys are part of what makes a calculator easier to use than a computer for quick calculations.

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**

Which calculator?

I have had several HP calculators over the years -- the HP-25, HP-33E, HP-41C and HP-32SII. The HP-41C is much my favorite although I like the feel of the smaller HP-25 case.

HP-25 Pluses	HP-25 Minuses
Small case feels good in my hand	
Relatively easy to make	
Fewer buttons needed	Needs 2 shift keys
no complex I/O	
	LED display uses a lot of power
Can do LCD instead of LED	Must do LCD instead of LED to conserve power

HP-41C Pluses	HP-41C Minuses
Fit my hand, barely	A little bit big for my hand
Lots of built-in functions	
Only one shift key needed	Needs more buttons
	Needs I/O to be complete HP-41C
	Harder to make than HP-25
Low power needs	
Already uses LCD	LCD module different than segment LCD

Since the HP-25 is much simpler than the HP-41C and I had always liked the shape of the HP-25 I figured that it would be a good place to start. I chose to build hardware similar to the HP-25 for the first try. I did add some features, somewhat like the HP41C, to the prototype.

The Do-It-Yourself RPN calculator

I wanted to use standard push-button switches so that a custom keyboard would not be needed. I started by choosing switches that had what I thought was a good feel. Also, I wanted a single printed circuit board for the switches as well as the rest of the circuitry.

This required that the parts had to be surface mounted -- the switches on one side of the board and the rest of the components on the other side. This all had to fit onto a circuit board that would fit into an HP-25 size case.

One difficulty was the display. I did not want to do an LED display since LED's consume a huge amount of power by today's standards. Instead I wanted to use a standard off-the-shelf LCD character display module.

Unfortunately, all the usual modules were too wide for a handheld calculator case. I have big hands but these modules would have made the case too wide for me to use.

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**

The LCD display

LCD character modules are pretty standard. They all use the same connector pinout and control scheme. A common 16-character-wide display is 80mm wide by 36mm tall by 8.8mm thick. Other similar displays go up in size from there.

Starting with 80mm (3.15") for the LCD and adding two 2mm (0.08") thicknesses for the case walls makes the case about 84mm (3.3") wide. This is about 15mm (0.6") wider than the HP-25 case, which is about 70mm (2.7") at its widest point.

This will never do. Fortunately, Lumex makes a display that is only 66mm wide. The Lumex display is two lines by 16 characters. The HP-25 only needs one line of digits. One line of display is wasted. No problem: the display is cheap -- less than \$6 in singles.

So, 66mm + 4mm = 70mm (2.8"). This is a bit more than the HP-25 size but still tolerable. Now that we know how wide the display is, we also know how wide the PCB can be since there is nothing to be gained by making the PCB narrower than the display.

The display is one of only two components that are not surface mounted. (The other is the LED). Since no components are mounted under the display there is no reason to make the PCB extend under the display. This makes the PCB shorter by nearly the height of the display.

In other words, the PCB is just a little bit longer than the keyboard portion of the PCB and the same width as the display. This ended up as 65mm (2.5") by 100mm (3.9")

To keep the mechanical assembly simple, the push-button switches and the display are on one side of the PCB with the rest of the components on the other side. This way, only one board is needed.

PCB

I had the boards made by a local company -- Advanced Circuits:

<http://www.4pcb.com/index.htm> and <https://www.barebonespcb.com/!BB1.asp>

All the work was done "untouched by human hands". The files were created on my PC using schematic capture and PCB layout programs. The files were sent over the Internet for a quote and fabrication.

I used the "Bare Bones Proto PCB" service to reduce the cost of the prototypes. Double sided boards without solder mask or silkscreen are shipped in one day. This puts them on my porch two days after the files (and my credit card number) are submitted. The cost is about 65 cents a square inch plus \$60 for setup and shipping.

The 2-sided board without a solder mask does, however, create a couple problems. The first is that the layout of the circuit board is almost like two one-sided boards. This makes routing the traces a pain. The second difficulty is in the ground plane around the switching power circuitry. The surface mount parts are hard to solder without bridging. In fact I had to use a stereo microscope to inspect for solder bridges.

Overlay

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**

The calculator uses standard switches chosen to match the height of the LCD display. This makes the top of the calculator flat so an overlay can be made with just one cutout for the display.

The front panel was done by printing an overlay on a photo printer. I could have used artwork, but instead I started with a picture of an HP-25 calculator and "PhotoShopped" it to change the slide switches to push buttons.

It was surprising how well the buttons felt considering the simplicity of the construction. I expected that the overlay would wear out very quickly since it is just a picture done on a printer. The prototype has not been used enough to know how long the overlay will last, but it shows no signs of wear through the testing stage of the project.

The rest of the calculator circuitry

The processor chosen for the DIY-RPN is the PIC18F452 in a 44-pin PLCC package.

Why this PIC? The PIC processors are cheap, readily available and both Eric and I have development tools for using PIC's. I also had the PIC18F452 in stock from a previous project, and it had enough pins and functions to do the job. The reason for the PLCC package is that it can be socketed. Parts often need to be removed from a prototype PCB because they get damaged or traces must be cut beneath them. I hate to unsolder a surface mount parts from a prototype PCB.

The PIC18F452 has a PWM output for generating an adjustable contrast voltage for the LCD module. In the prototype large value caps are used in a charge pump circuit to get the contrast voltage for the LCD module. These big caps make the adjustment of the contrast sluggish. It should be fixed but it does not prevent the calculator from being usable.

Other features of the DIY-RPN are an IR LED for talking to HP peripherals, a serial port, a second oscillator with a clock crystal and a connector for an MMC flash memory module. None of these are part of the original HP-25 but were added with the thought that someday we might want to do a DIY-RPN calculator more like the HP-41C.

The LED (a visible one right now) and serial port are also useful for debugging the calculator hardware and software.

The serial port uses the PIC I/O port pins and a few resistors. No RS-232 level shifter IC is used and, the connector for the serial port is a 1/8" phone jack. The MMC connector has not been used for anything yet.

The calculator is powered by 3-volt batteries. Two AAA batteries would be nice but they would make the case really thick. Instead, I chose to put two lithium coin cell connectors on the PCB. The battery life isn't as good with the coin cells as the AAA's but the case is a lot thinner.

The power supply consists of two step-up switching regulators. One of the switchers is used to generate the +5 volts for the PIC and display. The second switcher creates +3 volts for the MMC flash memory module.

The goal is to have a battery life of one year made up of five one-hour days per week. The lithium cells may not quite meet this requirement.

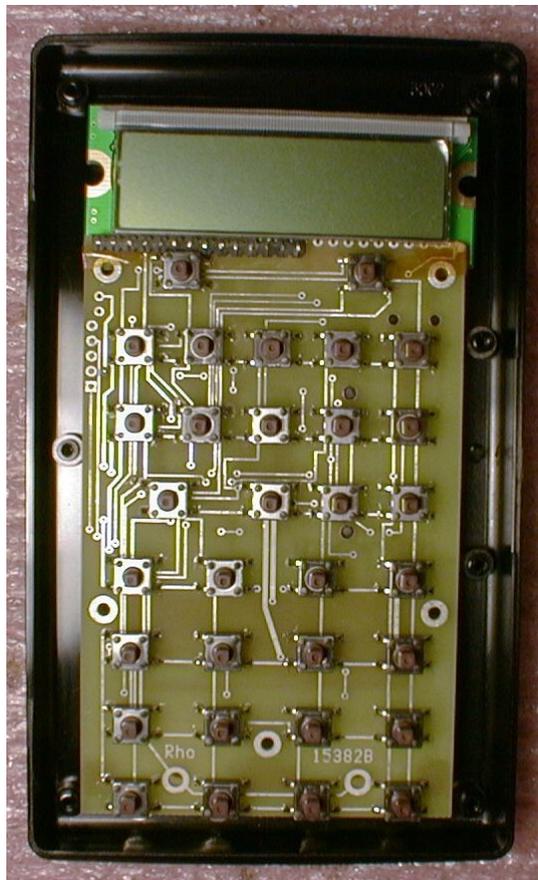


Illustration 1

*Switch side of DIY-RPN calculator PCB assembly
(in bottom half of case).*

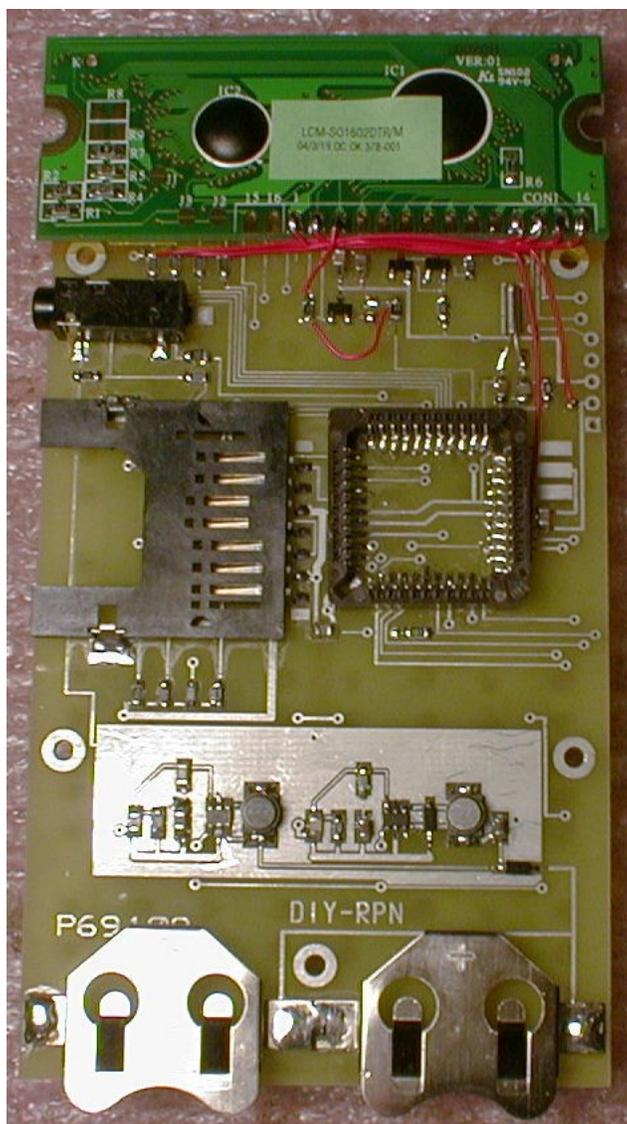


Illustration 2

Component side of DIY-RPN calculator PCB assembly.

The DIY-RPN case

I looked around and could not find an off-the-shelf case that was big enough to hold the PCB, display and batteries. I looked at the sizes of a very large number of cases and either the case was too big, too small, too thick or too thin. It was very frustrating.

Finally, I found a case that sort of worked. It was only good enough for the prototypes but that is enough for now. It is a clamshell case similar to the case of the HP-28C. This case, a Serpac V-15, is no longer made but I found a few surplus. I took the two halves of the case apart at the hinge and used each of the halves as one DIY-RPN case. The hinge area is a problem because it

creates large gaps along the edges of the calculator top.

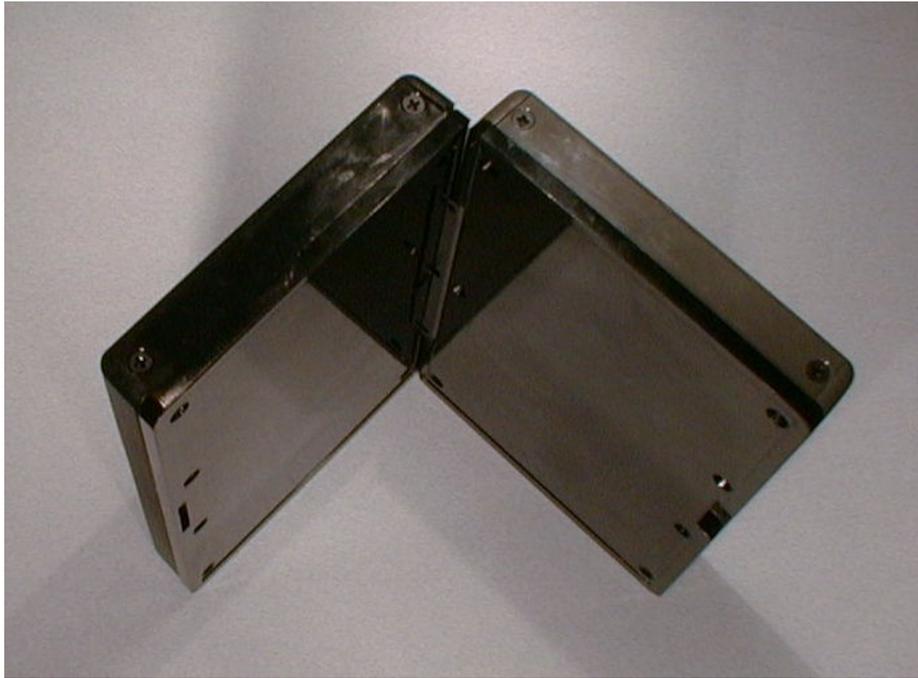


Illustration 3

Serpac V-15 case before disassembly.

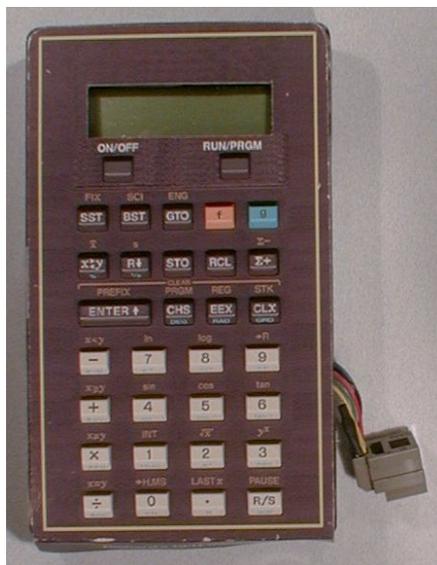


Illustration 4

DIY_RPN in Serpac case with programming cable sticking out of the unused hinge opening.

DIY-RPN software

Software was written to run the keyboard, display and to start testing the microcode simulation. The LED and serial port were tested for later use in debugging. The MMC was only slightly tested.

Start of DIY-RPN2

I was getting ready to upgrade the DIY-RPN design when Eric said something to the effect of: "The HP-25 simulation has a bug and doesn't work, why don't we do an HP-41C instead since its simulation works?"

After a moment's thought I said something like "Sure, I like the HP-41C and it's a better calculator as well. I'll do that".

A lot had been learned from the first DIY-RPN calculator prototype and I tried to fix and improve a number of things.

These included:

- Fix a big oops on first PCB to put the display on the correct side of the PCB.
- Only one power supply voltage to simplify interfacing between PIC and accessories.
- Accept either 5-volt or 3-volt LCD modules. Lumex has a 3-volt only module that does not need a negative contrast voltage.
- Better LCD contrast control circuit to reduce power use and smaller caps to reduce the contrast adjustment time.
- Real RS-232 driver and receiver. Needed because 3-volt power is not enough for "TTL level" trick.

Use a better connector to support SD flash memory cards as well as the MMC.
Put SD, LED and serial connectors on the top edge of the PCB to make them user accessible.
Add a serial EEPROM to store programs.
Add a piezoelectric transducer and amplifier to get sound.
Change the key layout to match the HP-41C keyboard.
Bigger battery holders to accept higher capacity lithium batteries.

Display

The DIY-RPN2 uses the same display as the DIY-RPN but the display is operated on 3-volt power with a negative contrast voltage. The negative voltage is generated by a capacitor charge pump inverter. An op-amp and a filter take the PWM output from the PIC and convert it to a DC voltage with an offset matched to the LCD module's contrast control range.

Using the 5-volt display on 3 volts with a negative contrast is not within the specifications of the display. What is not guaranteed is that the display controller will work properly at only 3 volts. In a display that uses a packaged controller chip you would just look at the part number to see if it is one that works at 3 volts. The display uses a chip-on-board with an epoxy blob over it -- there is no part number to read.

I used e-mail to ask the manufacturer if it was kosher to run the 5-volt display on 3 volts with a negative contrast and did not get an answer. Instead, I was told that I should use the version that is designed for 3-volt only operation. This sounded great since the charge pump circuitry would not be needed. In addition, the 3-volt displays are specified to use less power than 5-volt ones.

I got samples of the 3-volt display, since they are only available in quantities of 500 or more. For an unknown reason, the actual 3-volt display didn't work very well. Its contrast was poor even when the contrast voltage was at ground, and the power savings did not seem to be as much as I expected. I have not had the time to pursue this matter more fully.

I/O for the HP-41C

A real HP-41C can accept up to four plug-in modules. These include RAM memory modules, time module, card reader, infrared printer module, barcode reader, HP-IL to RS-232 interface and many others. Many software application packs were available in ROM modules.

The DIY-RPN2 uses the SD flash memory card to substitute for both the card reader and ROM modules.

The built-in RS-232 may be usable in place of HP-IL to RS-232 interface modules that were made.

The IR LED should be useful for talking to an infrared printer.

Power supply

There are two switching power supplies in the DIY-RPN. One for +5 volts to the PIC and display, and one for +3 volts to the SD. The dual voltages of the DIY-RPN were awkward since all signals from the +5 volt PIC to the +3 volt accessories had to be done by changing the PIC's port pins back and forth between inputs and outputs to act like open collector gates.

A bug was found when trying to switch off accessory power in HP-41C. The switcher would not shut off even though the specification sheet said it would. In fact, the spec. sheet was deceiving. When I discussed this with Eric, he pointed out that the switcher wasn't needed at all! Instead, it could be replaced by an FET switch.

A pair of diodes are used to protect the calculator and batteries if one or both of the batteries are installed reversed.

Batteries

I used larger lithium coin cells in the DIY-RPN2 to increase battery life. This was possible because the board is slightly wider than the one in the original DIY-RPN.

Again, the battery life goal is to get one year of five 1-hour days a week from a pair of batteries. I would have used CR2450 batteries, but Digi-Key does not sell holders for them so I settled for CR2332's. The PCB layout is the same for both battery sizes so it will be easy to fix later when I get the thicker holders from another distributor.

Having the two cells in parallel allows changing the batteries one at a time without losing the memory contents. A large backup capacitor is not needed, but the calculator must be off if both batteries are changed at once.

Printed Circuit Board

The PCB was made longer than the DIY-RPN to get room for the LED and the connectors for the SD card and RS-232 serial. This does not add to the length of the calculator since the board is extended under the display.

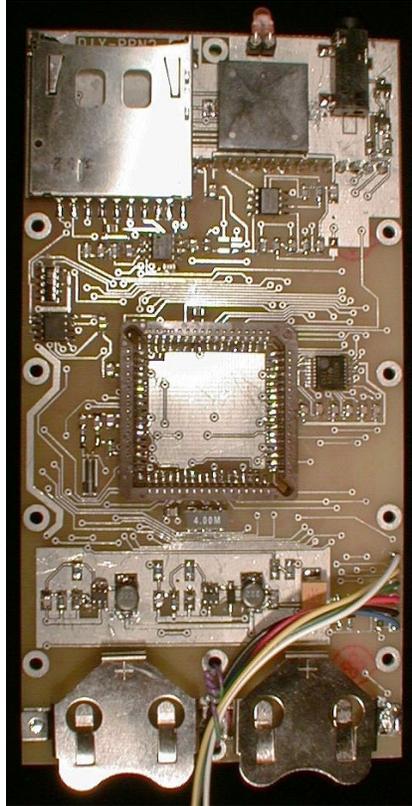


Illustration 5
Component side of DIY-RPN2.

Other circuitry changes

The processor was changed to a PIC18LF6680 in a 68-pin PLCC package. This PIC has the extra I/O pins for the added features and a lot more memory.

The watch-crystal oscillator circuit will now be used since some versions of the HP-41C have timekeeping functions.

An RS-232 transceiver IC was added to get true RS-232 voltage levels since only 3 volts is available from the power supply. The transceiver has power control pins to extend the life of the batteries.

Not only is there a Piezo buzzer for making noise, but it is driven from a low pass filtered and amplified PWM output. This will (in theory) enable the calculator to do more than beeps and clicks. The amplifier has a shutdown pin to conserve battery power when no sounds are required.

An internal serial EEPROM is now available for nonvolatile storage of programs.

The external SD flash memory card can be used to replace the HP-41C card reader in some uses. Even the smallest and cheapest SD is big enough for calculator use.

RAM was added to the DIY-RPN2 after the PCB was made. It is needed to match versions of the

HP-41C that have the maximum amount of built-in memory. The added memory chip is a FRAM, which has the same pinout as serial EEPROM. It was piggybacked on top of the serial EEPROM with the enable lead fly-wired.

Keyboard and overlay

The HP-41C could hold custom overlays that clipped in place. Some software application packs came with an overlay having keys matching the special needs of the program. HP sold blank overlays that let a user label his keys to match his own software.

The overlay in the DIY-RPN2 is exchangeable. When the battery compartment lid is removed, the overlay can be slid in and out. This makes it possible to have custom overlays similar to the ones of the HP-41C.

Exchangeable overlays could be made for more than one model of calculator. Maybe, more than one calculator may be possible in the same case with a special key sequence to select the model of calculator. The overlay would be switched to match the selected calculator. As an example, an overlay for the HP-25 would hide one row of unneeded HP-41C buttons.

To protect the printed surface from wear, I applied a transparent adhesive sheet. This does not seem to degrade the feel of the switches significantly.

The overlay is smooth and flat so you can barely feel the switch before pressing it. For my one-handed method of using the calculator this is a significant problem. One solution was to put small holes through the cover, centered on the buttons. This gives just enough tactile feel to find the button without adding to the height of the overlay.

Custom DIY-RPN2 case

The difficulty finding a case for the DIY-RPN was mentioned earlier. For the DIY-RPN2 I decided to make a custom case.

The case is a lot harder than the circuit design in many respects but without a proper case you cannot really have a calculator. With this in mind, I looked at many different ways of making a few custom cases.

emachineshop.com (<http://www.emachineshop.com/>) was extremely valuable in learning what was involved in many different fabricating processes. Their downloadable CAD software is like having a machinist looking over your shoulder. The software handles machining (milling), sheet metal punching and bending, pressure forming of plastic sheets, laser cutting and much more.

In addition, the software checks for ease of manufacture, helps with material selection, and gives quotes on costs of material, labor and shipping.

There are a number of paths to a getting a calculator case including: modifying an off-the-shelf case, injection molding plastic, rapid prototyping, pressure forming plastic sheets, punching and bending sheet metal, and machining metal or plastic.

How can a case be made?

Since I had such a problem finding a properly sized and shaped case for the DIY-RPN, I did not pursue modifying an existing case. Because of the cost, I dismissed rapid prototyping and injection molding for a small number of cases.

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**

I did look at pressure forming plastic sheet but determined that it was not very accurate and needed secondary machining to get the parts shaped for use.

CNC machining is the best way to make low volume, high quality and complex shapes suitable for a calculator case. The biggest problem is cost. The case could have cost 200 to 300 dollars for the two halves. I cannot justify this high cost for my prototypes. I did do some designing of the parts using the emachineshop.com CAD program and in general liked the results. If I had more time and money, I think that I could make a suitable case this way.

The first attempt at making a case for the DIY-RPN calculator would have to be a cardboard mockup of a bent metal case. I sent this to Eric in March of 2004 to pique his interest in doing software for the (as yet, non-existent) calculator hardware.

See Appendix A for how to build your own paper DIY-RPN calculator.

A bent-metal case can be made from either aluminum or stainless steel. Aluminum is light, inexpensive and can be finished in many colors by painting or anodizing. Stainless steel has the advantage that no finishing is needed other than wire brushing to remove scratches and burrs.

One advantage of punching and bending metal over machining is that the cost is lower. In addition, the processes are simple enough that simple tools can be used to hand make prototypes. The biggest drawback to a sheet metal case is that there are no easy ways to mount the circuit board and hold the case together.

Rapid prototyping is a method of making a mechanical part by building up layers one at a time until the part is complete. The layers are usually only a few thousandths of an inch thick so it can take quite some time to make a part. This makes the parts fairly expensive.

Laser cutting

A different way to make a case would be to use a stack of laser cut parts. This would be similar to rapid prototyping but with very thick layers. Laser cutting is not expensive -- about \$60 an hour for setting up and running the laser.

A laser works best cutting plastics such as acrylic and ABS. Some plastics such polycarbonate (Lexan) and PVC cannot be laser-cut. Polycarbonate is hard to cut with a laser. PVC decomposes into hydrochloric acid when cut, and the acid ruins the optics and mechanics in the laser cutting machine.

Laser machines can also etch the surface of the plastic. This allows labels and other marking on the plastic. The etch can also be deep into the plastic to create mechanical features. Because etching, especially deep etching, takes a lot of time the cost of etching can add a lot to the cost of a part. Another added cost of etching is that more setup time is needed to adjust the power of the laser to get the desired depth of etch.

Laser cutting a DIY-RPN case

I realized that I could do a poor-man's rapid prototyping by laser cutting many layers of plastic and then gluing the layers together. My layers are 1/16 inch thick so they do not give as smooth a shape as a real rapid prototyping machine would do. Making both the top and bottom of the case this way does give enough accuracy that they fit together tightly. One limitation is that a wedge

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**

shaped case cannot be made by gluing equal thickness onto each other. I was willing to live with that since a test case could be made for a fraction of the cost of the other methods I had looked into.

Acrylic, sometimes called by the brand names of Plexiglas and Lucite, is a plastic that is commonly used for products that are laser-cut and etched. Acrylic is available in a wide range of colors such as black, white, clear and even fluorescent red and green. I wanted to see into the finished case to verify the fit of the layers and electrical parts so I chose clear acrylic for the first case. One disadvantage of acrylic is that it is brittle and might crack if dropped.

Future cases may use ABS instead of acrylic. ABS is cheaper than acrylic -- about \$0.60 a square versus about \$3.25, but it is only available in a few colors: black, white and natural (beige?). A "Hair cell" texture on one side is standard and smooth on both sides may not be readily available in small quantities. ABS is a rugged material and can better withstand abuse like dropping than acrylic.

See Professional Plastics at www.professionalplastics.com for more information on pricing and availability of different plastics.

I started the laser-cut case design using the emachineshop.com software. This let me choose a type and thickness of plastic that was readily available and laser cuttable. I drew several layers and looked at them using the software's 3D rendering to get a feel for how the individual layers would look. It is a shame that the software would not let me stack the layers to see what the final results would be. Before the design can be rendered, the emachineshop.com software makes sure that the part is makable. After the design is rendered, a quote can be obtained. Then, the software may make suggestions on how to reduce the cost of the part.

One difficulty of the design is that every layer is different. That means every layer must be hand tweaked to get all of its functions right. Since there are 14 layers in the case design there are 14 pieces that must be designed.

After emachineshop.com said that the design could be made, I exported the design to CorelDRAW. CorelDRAW is commonly used for laser etching and cutting. I then converted the design so the laser machine could make the parts. Part of this involved separating the etches and cuts into their own layers in the drawing. The laser uses vectors to cut and raster scanning to etch so the two different processes must be done separately.

After the conversion to CorelDRAW, I wanted to import the design back into emachineshop.com to let it verify that the design could still be made. I was really upset when I could not do this because CorelDRAW handles objects, such as circles, as individual vectors making them too big for emachineshop.com to handle. This is unfortunate since the emachineshop.com program does design error checking for the different tools it handles, including the laser.

Gluing the layers

The parts must be precisely aligned when they are glued together or the outside of the case will be overly rough and the circuit board might not fit into the case. The finished case is held together by #2-56 screws. The screws pass through the bottom half of the case into tapped holes in the top half of the case. These same holes are used as indexing holes by inserting screws into them during gluing.

There are two sizes of holes for the screws -- the larger size is for the through hole and the smaller size is for the tapped hole. It worked out that the tap-hole size for a #2 screw is the same

size as a through hole for a #1 screw. This allowed me to use #2 screws for the index pins in the bottom half of the case and #1 screws for the index pins in the top half of the case.

There are some places where the battery lid overlaps the bottom layers of the case. These are slots and tabs that align the lid as it slides into place. These overlaps can't be glued together; otherwise the lid would never slide. To keep from fusing the overlaps, I put a shallow etch in the overlapping areas to make room for wax paper that kept the tabs from being glued to the surrounding layers.

I glued the layers using a water thin glue (WELD-ON 3) that would flow between the layers while they were stacked and aligned by the indexing screws. Some cutting and re-gluing was required to get the case to work even though I had made cardboard cutouts to test the design before laser cutting the plastic pieces.

The first case I made assembled fairly quickly, but the second case did not go together as well and took longer to glue. This extra time exposed me to too much of the glue fumes. After gluing the second case I noticed that my far distance eye focus was not working well for a while. This was apparently caused by the solvent in the glue. From now on I will use better ventilation and try not to repeat this experiment.

Finishing the case

Once the glue set there were a few more steps to finishing the case. The holes had to be tapped for the screws that hold the case halves together and attach the circuit board to the top half of the case.

The very top layer has a deep etch to make the groove for the overlay and does not have any indexing holes in it because it covers these holes. This layer was glued in place after the holes were tapped.

The battery compartment lid is still attached to the bottom half of the case. Making the lid part of the case while gluing guaranteed that it would have precise alignment with the rest of the case. Now that the case is glued together, the thin connection holding the lid to the rest of the case can be cut. The separation was done with a razor saw (a saw with a blade only a few thousandths of an inch thick).

The lid slides between layers of the top half of the case to make a tight fit. The top layer of the lid would have been glued in place so it could not slide if it had been in place while the top half of the case was glued. This piece was removed before any gluing was done and can now be glued to the lid to finish it.

A tab sticks up from the lid to stop the overlay from sliding out while the lid is in place. This small rectangle is glued into a slot of the lid.

The last step making the case is to add a #1 screw to make a mechanical detent for holding the battery lid closed. The screw is in a tab on the lid that overlaps a hole in the bottom layer of the case. When the lid is closed, the screw head snaps into this hole. There was no way to make this bump part of the plastic since it is a small floating piece.

After adding some rubber feet to the bottom of the case I was done. (These feet fit into deeply etched areas on the bottom layer of the case to keep them from sliding off).

When the case was done, there was some roughness to the sides caused by imperfect alignment

of the layers. I thought about sanding or filing to smooth out the roughness but decided that it was not worth the effort.

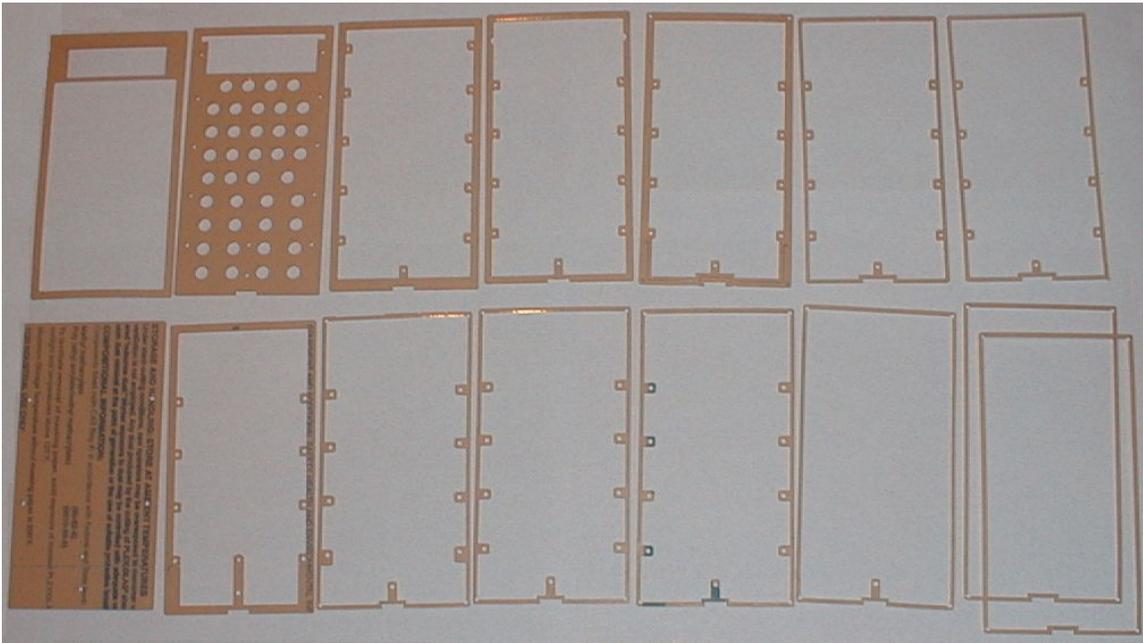


Illustration 6

The "kit" of DIY-RPN2 case parts with protective paper still on the pieces.

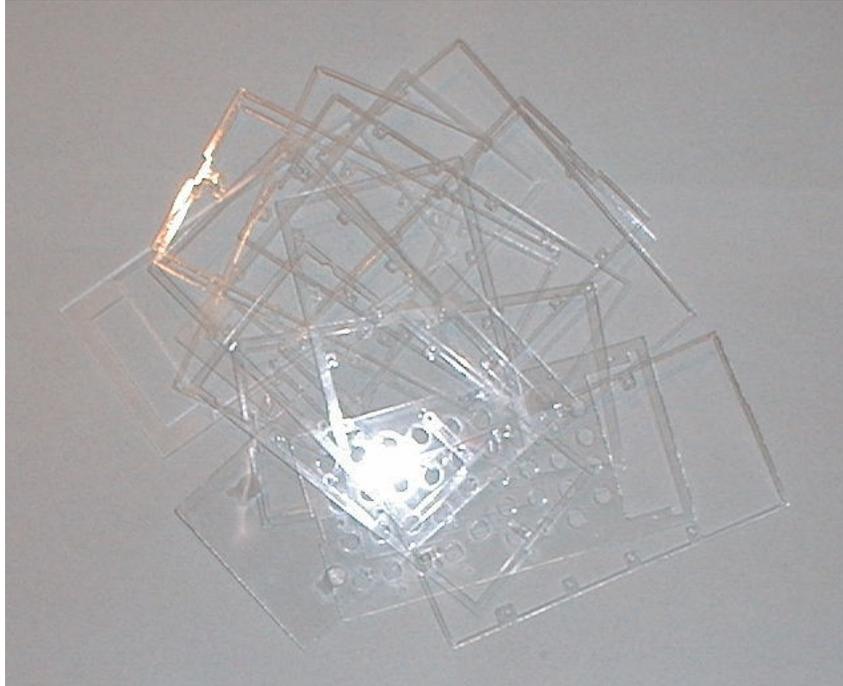


Illustration 7

Pile of DIY-RPN2 case parts.

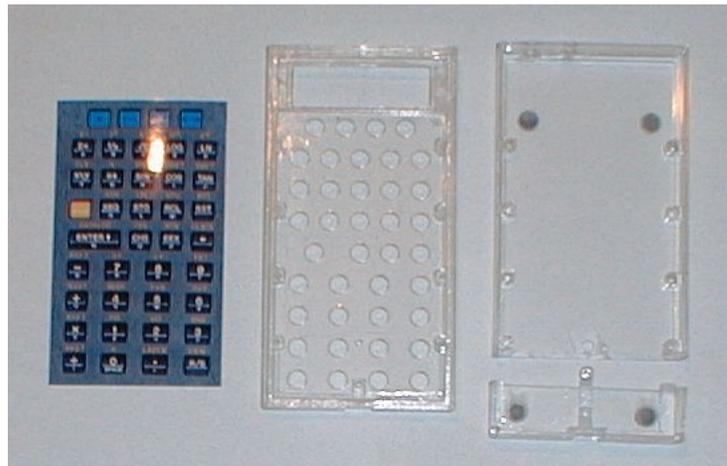


Illustration 8

DIY-RPN2 overlay, top and bottom and battery lid.

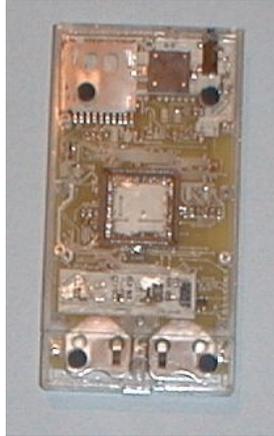


Illustration 9

*Bottom of DIY-RPN2
case with PCB.*

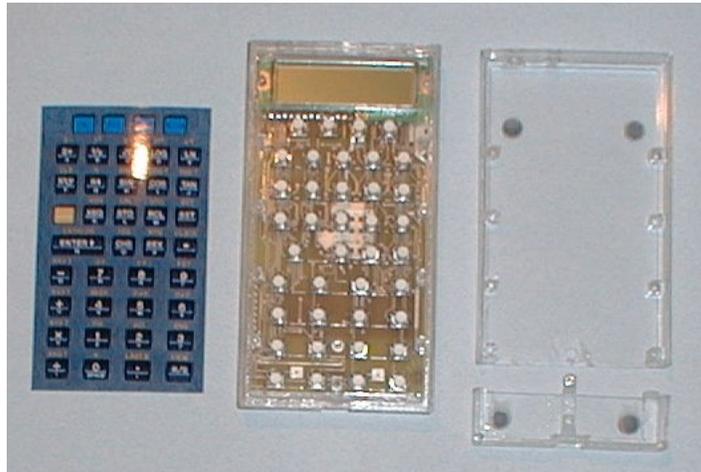


Illustration 10

DIY-RPN2 overlay, top with button caps installed and bottom.

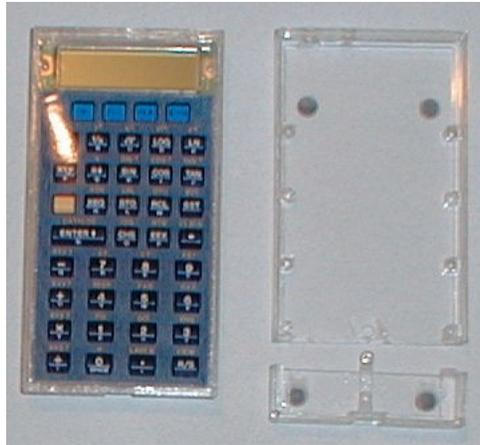


Illustration 11

DIY-RPN2 with overlay installed and bottom.



Illustration 12 Front of DIY-RPN2.



Illustration 13
DIY-RPN2 from above.

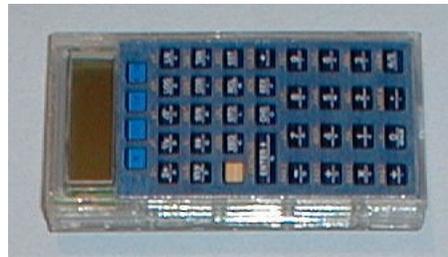


Illustration 14
Side of DIY-RPN2.

Features of each assembly

Here are how the two prototypes compare:

	DIY-RPN	DIY-RPN2
HP model	HP-25	HP-41C
Keys	32	39
Overlay	Fixed	Changeable
Size	81mm x 140mm x 17mm 3.2" x 5.5" x 0.65"	71mm x 140mm x 23mm 2.8" x 5.5" x 0.9"
Weight	140 g (5 oz)	163 g (5.75 oz)
Case	Modified Serpac V-15	Custom, laser cut
Processor	PIC18F452	PIC18LF6680

	DIY-RPN	DIY-RPN2
Display	2 line by 16 character	2 line by 16 character
RS-232 serial	Yes, TTL level *1	Yes
External memory	MMC *1	SD
Internal memory	No	256KB serial EEPROM and 8KB serial FRAM
LED	Yes *1	Yes
Sound	No	Yes, amplified PWM
Clock crystal	Yes	Yes
Batteries	Two CR2032 *1	Two CR2332 (CR2450 in future)
Battery life	Unknown (maybe 1 year)	6 months (1 year)

*1: No opening in case

Acknowledgments

Extreme thanks to John Doran and his company, Epilog Laser.

<http://www.epiloglaser.com/>

John has been very supportive of the DIY-RPN project with his time and the use of a laser engraver. In addition, John was instrumental in some of the design concepts that made the case work. He suggested using deep etching to make a groove so that the overlay could slide in and out.

Of course, Eric Smith has been the software architect that made the project conceivable.

Loren Blaney proofread this paper greatly improving it.

Many members of the 6502 Group have added ideas and helped at one time or another.

<http://www.6502group.org/>

**HHC
2005**

September 17 & 18, 2005
Chicago, Illinois, USA

**HHC
2005**