Don Lancaster
Synergetics

Lowercase for Your Apple II (Part 1)

Expand the usefulness of your Apple with this inexpensive addition.

With a few modifications and some new software, you can plug a TVT 6-5/8 lowercase module A into an Apple II.

These simple changes turn your Apple II into a combined uppercase and lowercase computer and can cost you as little as $10. Your new lowercase ability frees your Apple to do word processing, text editing and typesetting, generate mailing lists, write form letters and so on. The modifications require two extra integrated circuits added to the breadboard area already on the Apple. If you like, you can get by with only add-on wires and no foil cuts.

The change-only-the-character-generator approach doesn't tie up or restrict blocks of ROM, RAM or graphics display memory. What we are about to show you is also totally invisible—your Apple II stays an uppercase machine till you specifically ask for some lowercase output. Software does the switchover at any time, and the regular Apple II keyboard is used for both uppercase and lowercase.

There are two minor limitations to this conversion. If you still want to be able to reverse video, you may have to add a changeover switch, which gives you a choice of reverse video or lowercase. You'll also find that lowercase characters will be flashed more attractively with software rather than hardware. The method we'll show you should work on many other terminals and computers, if they use a new style 2513 character generator and have a full 8-bit-wide display memory.

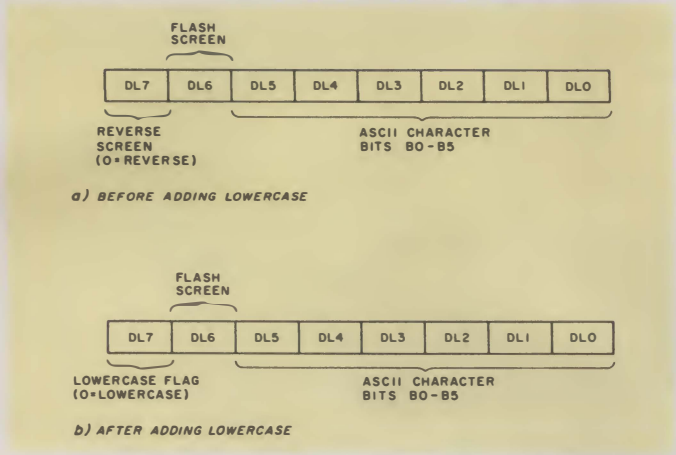

Fig. 1. Bit definitions of Apple II character DL bus.

Some Details

Just adding lowercase to any old computer or terminal sounds simple enough. Plug in an uppercase and lowercase combined character generator, and you are home free, right?

Well, not really. First and foremost, you have to want to do something useful with your new lowercase characters. While they are nice to look at for displays and some games, unless you have a printer or other output that needs and uses lowercase, you really haven't gained very much. So if you want the new characters, make sure you have some way to get them out of the machine. An important rule is to *make sure you have some use for lowercase before you go to the trouble of providing it.*

An obvious problem that immediately crops up involves the keyboard and its encoder electronics. The Apple II has an uppercase-only keyboard. An old National chip was used for the encoder. This chip is strictly uppercase only, compared to the usual 2376 with its choice of coding options. The Apple keycaps, particularly those on the M and the P, will also limit how you can use the existing keyboard. And there are no spare keys to speak of.

We'll show you how to use software to trick the existing Apple II keyboard into giving us lowercase when and where we want it. The software secret is to use the Escape key as a shift lock for lowercase. More on this later.

Another problem is created by the firmware in the Apple II. The operating systems and monitor are needed for machine language, for the mini-assembler, for Integer BASIC and for AppleSoft.

All four of these languages demand uppercase only, and the firmware is happy to provide it. In fact, most of the sequences go to a lot of trouble to make sure that everything is uppercase. Put in lowercase, and the sequences will convert it back for you. Even the winking cursor forces an uppercase-only output. So, even if you force-feed your Apple from a new lowercase keyboard, the internal firmware will try to change it all back to uppercase anyway.

The way around all this is to use some new software that bypasses the firmware when and if we need lowercase. This is a key to full alphanumerics. We have to make sure that everything we do stays fully invisible and appears to be uppercase only, unless we specifically call for the new characters.

Our modifications meet these goals:

- The existing keyboard is used

DL7	DL6	Screen
0	0	Black characters, white background
0	1	Flashing character, black background
1	0	White character, black background
1	1	White character, black background

Table 1.

DL7	DL6	Screen
0	0	White lowercase characters
0	1	Flashing characters
1	0	White uppercase characters
1	1	White uppercase characters

Table 2.

without any changes.

- Apple hardware changes consist of two new ICs in the breadboard area and a plug-in module. No foil cuts are needed.
- Lowercase is completely invisible until it is called with software.
- No hi-res graphics or large blocks of ROM and RAM are tied up.

Fig. 1 shows us the old and the new bit assignments for the Apple II display memory, or DL bus. The lower six bits are used for the ASCII character code, arranged in the usual order. The next bit is DL6. It's used to flash the screen. Screen flashing is most often used for the cursor, but it is also convenient for alarm or error messages.

The final bit is DL7. It was originally used to reverse the screen display. This gives you black characters on a white background, and is normally used for emphasis.

Lines DL6 and DL7 are not independent. You cannot flash a white screen; you can only flash a black screen. The truth table for these two lines before modification is shown in Table 1. If it weren't for the interaction between these two bits, some capital letters would always flash with the existing Apple II firmware.

The obvious thing to do is make DL7 equal to the seventh ASCII line needed for your new character generator. But there doesn't seem to be any reasonable way to do this and still have invisible operation when you *don't* want lowercase. Instead, we use DL7 as a lowercase flag. If DL7 is a 0 *and* if DL6 is a 0, then we want lowercase out of our character generator. Otherwise, we want everything to stay just the way it was. Table 2 shows our truth table after modification.

Once again, the reason we do this in a nonobvious and seemingly complicated way is to keep compatibility with everything that is already working in your Apple II.

The hardware modifications involved are simple and inexpensive, but you should not attempt them if you aren't adept at adding wires to a printed circuit board, reading socket pins and so on. There are three things involved in the hardware changes:

- The character generator is replaced with one that also generates lowercase.
- A new integrated circuit gate is added to decode lowercase for the character generator.
- A new integrated circuit gate is added to prevent lowercase characters from appearing as black on white.

The first change is done using a TVT 6-5/8 module A. This consists of an $8 uppercase and lowercase MCM6674 character generator mounted on a small adapter card that plugs into the existing 2513 character generator socket. The second two changes involve 15¢ integrated circuits added on new sockets in the Apple breadboard area. One direct IC-to-IC wire is used to eliminate the need for any foil cuts.

The schematic of the lowercase modification for the Apple II is shown in Fig. 2. Character generator A5 is unplugged and replaced with a TVT module A that carries a new uppercase and lowercase MCM6674P character generator. A new wire is routed to pin 23 of A5 that carries the new seventh ASCII bit A6 needed for the dual-case operation.

The logic rules for this new lead tell us to make A6 the complement of A5 for uppercase, numerals and punctuation, but to make A6 a 1 for lowercase. This lowercase condition happens when DL6 and DL7 are both zeros.

A new 74LS02 quad NOR gate integrated circuit is put in the breadboard area at A11 to do this A6 logic conversion for us. The gate outputs a 1 if DL6 and DL7 are both 0, and otherwise outputs the complement of DL5. The reasons behind this logic are apparent if you study the ASCII coding involved.

If we simply changed the character generator and added a quad NOR gate, we would get invisible normal operation and lowercase when we called for it. The only hassle involved is that the lowercase would appear as reverse video, with black characters on a white background. To beat this final problem, we add a second integrated circuit in the breadboard area A13. A13 outputs a signal for us that is low only when the flashing condition of DL6 = 1 and DL7 = 0 takes place. Otherwise, a 1 is output that forces the normal white-on-black screen display.

Note that the original DL7 connection going to pin 6 of B13 has to somehow be broken. This can be done by cutting foil, but a safer and more reversible way is to bend pin 6 of B13 out of its socket and make a direct topside wire connection.

There is one final detail we must attend to in the modification for lowercase. The Apple II still applies unused negative voltages to pins 1 and 12 of the character generator. This probably dates back to the days when some 2513s needed these supply voltages, or else is a hedge should a different part be needed. At any rate, *an unmodified TVT module A will short out the power supplies if it is plugged into an unmodified Apple II.* Fig. 3 shows us several ways out of this bind. Anything that keeps a short off the −5

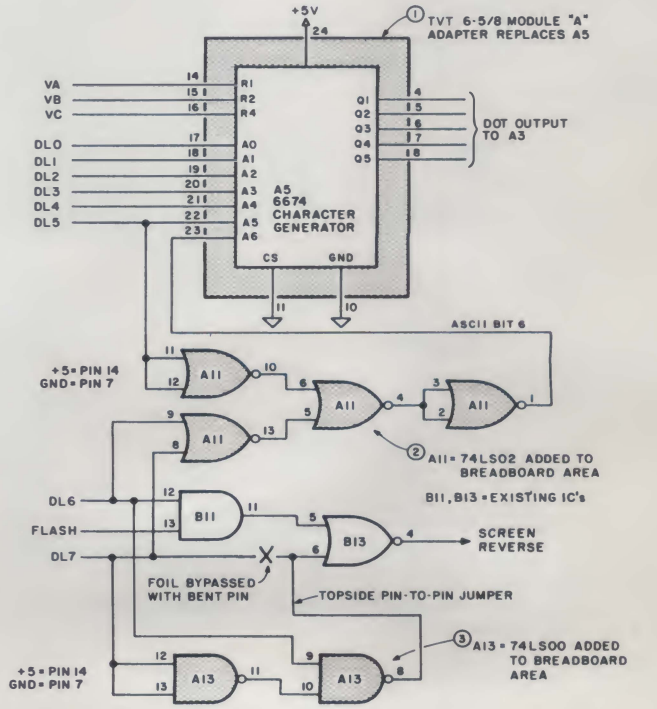

Fig. 2. Schematic of Apple II lowercase modifications.

- Leave pins 1 and 12 off of module A during assembly
  or
- Bend pins 1 and 12 of module A up and out of the road
  or
- Cut pins 1 and 12 of module A flush with its circuit board
  or
- Use a PC layout for module A that floats pins 1 and 12
  or
- Cut the foil on the dead-end supply lines going to pins 1 and 12 of character generator A5 on your Apple II.

Fig. 3. Do not plug an unmodified TVT Module A into an unmodified Apple II! The Apple II provides live but unused negative supply voltages to pins 1 and 12 of its character generator. Here are several routes to Module A compatibility.

and −12 lines will work.

Hardware Changes

As with just about anything in the new computer world, both hardware and software are involved. If you make only the hardware changes we are about to look at, your Apple II will still behave just as it did before, with the only exception being the loss of screen reversal. To actually use lowercase, we have to add new software as well.

Our new software will be in the form of short integer BASIC programs and sequences. Once you decide what you really want to do with your lowercase Apple, you can use these sequences as they are, integrate them into your working programs or convert them up to Applesoft or down to machine language. We'll be giving you more than enough software to get you started.

Fig. 4 repeats the details of the TVT module A from the *Cheap Video Cookbook* (Sams 21524). We have changed the callouts around to match the Apple's and have eliminated pins 1 and 12 from the module to eliminate the supply-shorting problem.

Module A Assembly

Carefully inspect the circuit board for opens, solder bridges, etc. Try tinning one of the runs on the board. If there is any problem with easy solder adhesion, carefully clean all the areas to be soldered with an ordinary pink eraser. Avoid handling the board, as it will make soldering more difficult.

Set your PC board bare side up with the notch in the upper left-hand corner. Insert a 0.1 uF disk ceramic capacitor in the two middle, left-most holes. Solder the capacitor in place. Clip and save the excess leads.

Use one of the leads left over from the previous step to provide a jumper in the two middle, right-most holes.

Use the other remaining lead to provide a jumper immediately to the left of the one you just installed. Solder both jumpers in place.

Add an 18-pin integrated circuit to the remaining middle holes. If the socket has orientation marks or notches, point these toward the capacitor.

Shorten one of the 12-pin strips so it is only 10 pins long. Center this strip above the socket. The long end of the pins and the spacer go on the bare side; the short pin side goes to the foil. Solder in place after making sure that the strip is flat and that one hole remains unused at each end of the strip.

Add a 12-pin strip to the remaining 12 holes at the bottom. Be sure this strip is flat before soldering and that it points the same direction as the previous strip.

Carefully study the foil side in Fig. 4 and add the following four wire-pencil connections to the foil side:

IC pin 12 to module pin 4
IC pin 13 to module pin 5
IC pin 15 to module pin 7
IC pin 16 to module pin 8

Note: Be sure you understand the pin numbering before you start. On the foil side, the connector pins run *counterclockwise*. The 18-pin IC socket pins run *clockwise*. The end jumper and capacitor holes are *not* counted. There are no module pins at locations 1 and 12.

Check the previous step. Your four connections should form a cross within a cross that reverses the sequence of five side-by-side pad pairs.

Insert an MCM6674P character generator into the module, *putting the notch at the capacitor end*. You may have to gently force the pins on the IC slightly together by rotating the IC against a tabletop or bench.

Store your completed module in protective foam at all times.

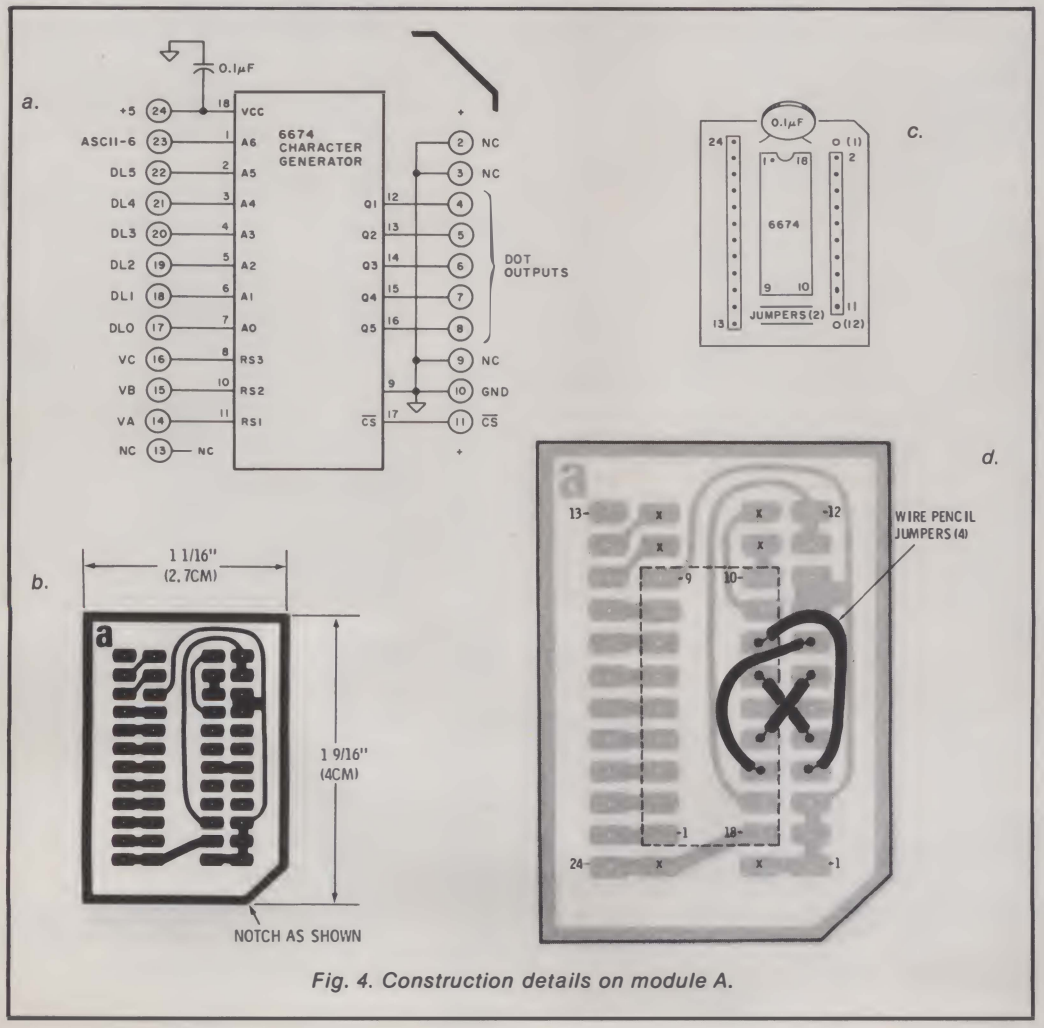

Fig. 4. Construction details on module A.

1 TVT module A lowercase plug-in with floating pins 1 and 12 (Fig. 4).
2 14-pin integrated circuit sockets
1 74LS02 quad low power Schottky TTL NOR gate
1 74LS00 quad low power Schottky TTL NAND gate
1 Length of #24 solid, insulated wire, around two feet long.
1 Length of electronic solder suitable for PC board use, around two feet long.

Phillips Screwdriver
1/4 inch nutdriver (optional)
Needle-nose pliers
Diagonal cutting pliers
Wire stripper
Small soldering iron

Fig. 5. Parts and tools list to add lowercase to an Apple II.

This completes assembly of your module A.

Lowercase Modification

Fig. 5 provides a list of the tools and parts you will need for your Apple II modifications. If you know how to solder on a printed circuit board and are familiar with PC socket numbering, the changes should be inexpensive and easy to do. If you aren't into this sort of thing, or if chopping and channeling a $1000 computer is against your religious convictions, have somebody else do the work for you.

Turn your Apple II off and remove the power supply. Remove all video cables and cassette cables.

Lift the lid off the Apple II by pulling sharply up at the left of rear center and then at the right of rear center to unsnap the Hedlok fasteners. Set the lid aside.

Carefully unplug any remaining rf modulator cables, game paddles or other I/O connections, and any plug-in cards, making careful note of where they go and how they are oriented.

Place the Apple II upside down on a bench that is protected with a rug or a foam pad.

Remove the four semi-recessed Phillips head screws at the bottom front (Fig. 6). Set them aside in a safe place.

Remove only the six outermost Phillips head screws from the bottom. There should be two at the extreme left, two at the extreme right and two at the extreme rear. Set these screws aside. *Do not remove any other screws!* Often the outside six screws are a slightly different color than the others (Fig. 6).

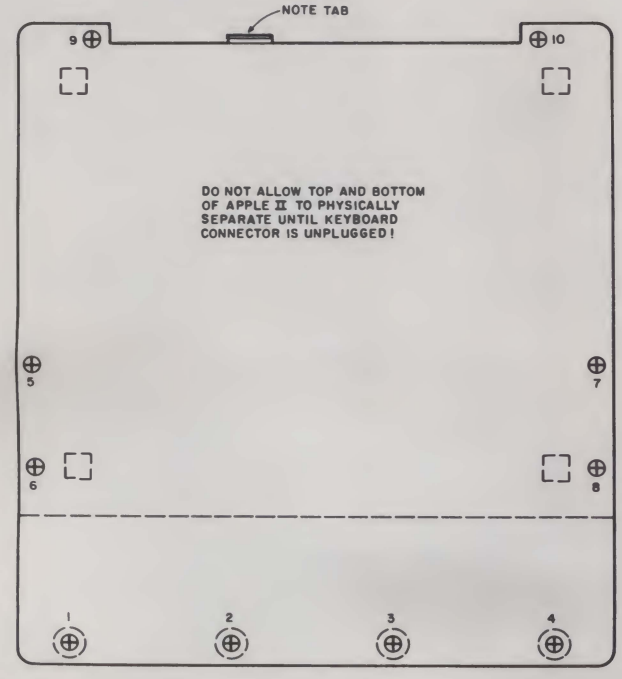

Fig. 6. To disassemble your Apple II, remove only the screws shown here.

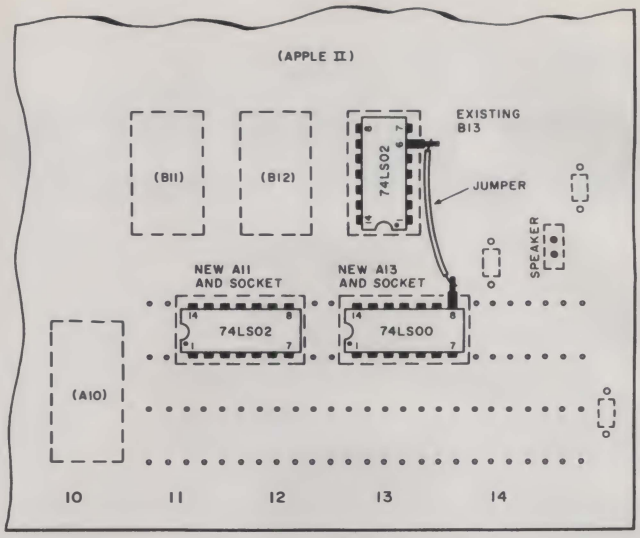

Fig. 7. Top view of lowercase modifications. Jumper shown eliminates the need for foil cut.

While you are carefully holding the top and bottom of the computer tightly together, turn the computer over so it is right-side up.

Gently lift up the front of the computer only far enough that you can see inside. Note the keyboard connector that plugs into location A7. Gently remove this connector from the main computer board end.

Check the rear of the main circuit board by the video jack. If an rf modulator or something else is plugged into the four-pin connector at K14, carefully remove it.

At this point there should be nothing preventing you from removing the top of the case. Remove the cover and set it aside. Note: The pins on the keyboard connector and the unprotected speaker cone are easily damaged. Be gentle!

Note how the integrated circuits are numbered by column and alphabetized by row. Verify that (1) there is a 2513 character generator at A5; (2) there is a breadboard area at A11 through A14; (3) all integrated circuits have code notches and dots that line up pin 1 with white dots on the circuit board.

Unplug the power supply connector. Pry gently against the plastic clips on either end of the socket to release them.

Remove the 6-32 nut and washer in the center of the main computer board near F8.

Unplug the speaker connector.

Note that there are six white nylon board supports. Be sure to note the one at J9.

Gently squeeze the support at A1 with your needle-nose pliers till the barb releases the board. Lift the board up only far enough to free it from the barb.

Release the other barbs, one at a time, starting with A14, followed by J9, K14, K9 and, finally, K1.

Remove the circuit board from the computer. Set all the computer parts aside except for the circuit board.

Study Fig. 7. Add a 14-pin integrated circuit socket to A11, so that it straddles the uppermost breadboard row, starts in the third hole from the left (two holes show at socket left) and has any notches or dots oriented to the left. Tack the IC socket in place at pins 1 and 8. Then remelt these pins while pushing down on the socket to make sure the socket is solidly seated. Solder all 14 pins from the foil side.

Skip two holes and add a second 14-pin integrated circuit socket immediately to the right of the first one. It should also straddle the upper two rows and should have seven holes visible on the right and two holes visible between the sockets.

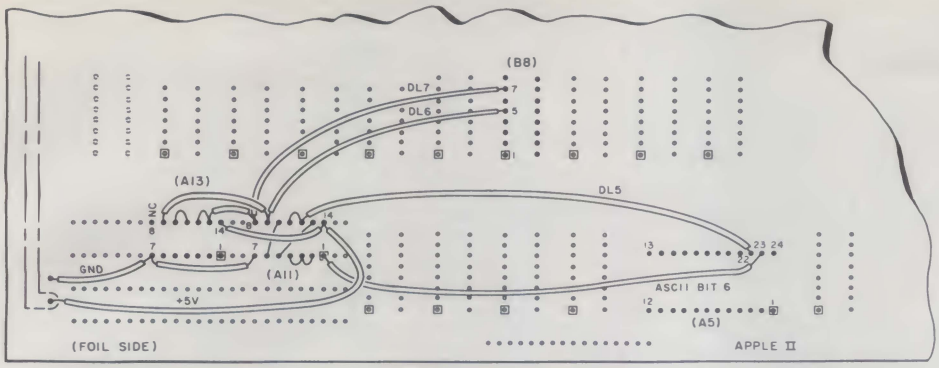

Fig. 8. Bottom view of lowercase modifications.

Plug a 74LS02 into the leftmost socket at A11, making sure the code dot and notch go to the left as shown.

Take a 74LS00 and carefully bend pin 8 so it sticks straight out. Now plug this 74LS00 into location A13, making sure the code dot and notch go to the left as shown.

Carefully remove the 74LS02 in socket B13. Then bend pin 6 of this integrated circuit straight out. Replace this integrated circuit in its socket, making sure the code notch and dot point down toward you, just like all the others in that row.

Prepare a 1-1/4 inch (32 mm) wire by stripping 1/8 inch (3 mm) off each end. This should be a solid wire, preferably #24.

Solder this wire between the two "flying" pins, pin 8 of A13 and pin 6 of B13.

Turn the board upside down and provide the following connections, each time picking a reasonable length of wire and stripping 1/8 inch (3 mm) from each end. When soldering to existing pads, butt the wire against the pad after tinning it. Do not place the wire beside the pad where it can contact the next pad over. *Note that integrated circuit pins count clockwise from the foil side.* See Fig. 8.

Ground wire 7/A11 to 7/A13 to ground at green capacitor A14. *Do not connect to the wide foil.* Connect only to the capacitor lead.

+5 supply wire 14/A11 to 14/A13 to +5 at green capacitor A14. *Do connect to wide foil.*

ASCII bit 6 output wire 23/A5 to 1/A11.

Short bare jumper 2/A11 to 3/A11 to 4/A11.

Short bare jumper 5/A11 to 13/A11.

Short bare jumper 6/A11 to 10/A11.

DL5 input wire from 22/A5 to 11/A11 and 12/A11.

DL6 from 5/B8 to 9/A11 to 9/A13. Be very careful finding 5/B8. Note the square foil pad on all pin 1s of the integrated circuits.

DL7 from 7/B8 to 8/A11 to 13/A13 and 14/A13.

Short bare jumper 10/A13 to 11/A13.

Inspect all the previous connections for possible shorts against adjacent pins.

Remove the character generator A5 from the computer and store it in protective foam. If you have no other foam, use the other side of the foam holding module A.

Plug module A into A5 so that the notched corner is located at A4. See Fig. 9.

Vigorously shake the board to make sure no wire ends remain on the board. This completes the actual modifications.

Gently place the board back onto the nylon supports on the computer bottom. Press down till each barb grabs its portion of the circuit board.

Replace the 6-32 washer and nut in the center of the board.

Plug the power supply connector and the speaker back into their respective sockets.

Set the top back onto the computer.

Gently lift the top and plug in the keyboard connector at location A7, KEYBOARD. Make sure that pin 1 aligns with the white dot, that no pins are bent and that no pins stick out either end of the socket. Check the keyboard end of this ribbon cable to make sure it is also firmly seated.

Reconnect the rf modulator to the 4-pin VIDEO connector if you have one.

While you are firmly holding the top and bottom of your computer together, carefully turn it upside down onto the rug or foam pad on your bench.

There is a metal hook at the back of the computer. Make sure this hook goes into its matching slot in the plastic top (Fig. 6).

Replace the rear-most two Phillips screws. Do not tighten completely. Note: These are flathead screws without washers.

Replace the center-front two Phillips screws. Do not tighten completely. Note: These are binder screws with lock washers.

Replace the remaining two binder head screws at the front.

Replace the remaining four flathead screws, two on each side.

Tighten all screws.

Replace the game paddles, rf output leads, I/O cards and I/O connectors, exactly as you found them.

Replace the cover. Tuck the front end under the top of the computer and then carefully align the cover. Then press firmly down with the heel of your hand, first at left rear, then at right rear, until the Hedlok fasteners snap into place.

Replace the video and cassette connectors.

This completes the modification of your Apple II to lowercase.

Initial Checkout

Check your modification to make sure it is working:

Switch the computer to off and then plug it in.

Very briefly switch the computer on and then back off again. The power supply should click only once, and the POWER light should come on. If the power supply continuously clicks or if the POWER light doesn't come on, you have a short somewhere. Backtrack to find out where.

Now switch the computer on only long enough to press the RESET key. The speaker should beep. If the speaker does not beep, stop to find out why.

Check out your display with an integer BASIC program of some sort. You should have a completely normal display, all uppercase and white on a black background. Some of the punctuation may be slightly different, such as larger periods and commas than before.

Load and run the integer BASIC lowercase test program (see next page). All the letters should appear as lowercase on the lower line, repeating over and over again. Numerals and punctuation should appear normally. As this is a simple test program

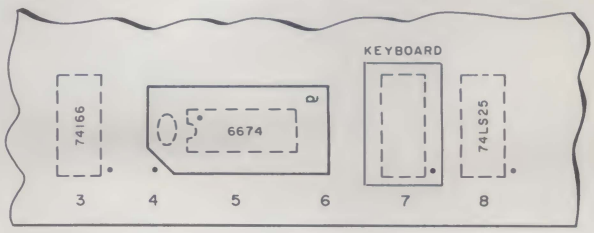

Fig. 9. Correct positioning of module A.

used for debugging, don't worry about the missing cursor or the lack of scrolling.

Type a CTRL A. You should get a capital letter A. Type a CRTL B. You should get a capital B.

Type a CTRL C. What happens? Why?

This completes your checkout. Should you have problems, isolate the trouble to the likely area. For instance, if you can't light the POWER lamp or if the power supply overload relay continuously clicks, look for shorts caused by not-floating pins 1 and 12 of module A, solder blobs or two-pad shorts or integrated circuits plugged in wrong. Note that an *unconnected* power supply will also continuously click.

Your module A generates the characters for you. It receives its lowercase control signal from A11. The screen reversal inhibiting is done by A13. Should you run into trouble, isolate the problem to the source.

If you want to get back to uppercase only, just put the old character generator back, remove A11, A13 and B13 and then put the new or straightened-out 74LS02 back in slot B13. If you are an old pro at PC work, you can put the topside wire on the bottom by cutting the foil going to pin 6 of B13. This is not recommended till after you have debugged your lowercase.

In part 2, we will examine the software development that calls for the lowercase when it is needed. We will also consider further hardware modifications, including adding a switch to give you a choice of reverse screen or lowercase.■

```
100  FOR CURS = 2000 TO 2039
110       CHAR = PEEK ( – 16384): IF CHAR<127 THEN 110
120       POKE ( – 16368),0
130       IF CHAR>192 THEN CHAR = CHAR – 160
140       POKE CURS,CHAR
150       NEXT CURS
160  GOTO 100
```

A lowercase test program that puts lowercase characters on the bottom display line. Numerals and punctuation appear normally. Use this program for hardware checkout. CTRL-C restores normal BASIC operation.