41CL Flash Updates

Every effort has been made to ensure the accuracy of the information contained herein. If you find errors or inconsistencies please bring them to our attention.

Notice:

Acknowledgements:

Introduction

Since the original release of the 41CL board the number of software images available has more than doubled. As a result, depending on when you first acquired your board, you may be missing quite a few of the latest images. There have also been a number of revisions to the basic 41CL software, as well as new 41CL-specific software, that provide significant advances in the functionality available to a user.

If your 41CL calculator has the optional serial port connector it is fairly easy to add new images or new versions of images to the Flash memory. This document will go through the basic steps to update Flash memory using the serial port.

41CL Board Versions

There have been three different board revisions of the 41CL board, called V2, V3 and V4.(1) V2 boards have half the Flash memory and half the RAM of V3 and V4 boards. The primary difference between a V3 board and a V4 board is that V4 boards contain a connection that allows the serial port to be enabled under software control. This feature is only important when doing 41CL-to-41CL serial transfers, so as far as this document is concerned V3 and V4 boards are identical.

Before attempting to update the Flash memory, it is important to know which board version you have. A quick and dirty way to determine this is to read a location in memory that is outside the range of valid addresses for a V2 board. If the board version is V2, the data returned will be read from a valid V2 location, because the memory addresses wrap around, and will be different from the data returned with a V3 or V4 board. You can do the following to determine your board version:

"100000-FFFF"	This is the address-data format in the ALPHA register required for the PEEK operation.
XEQ "YPEEK"	Reads location 0x100000. The display will return the data read from memory in place of the "FFFF" that was in the ALPHA register. If the data returned is "0201" then you have a V2 board. Any other value means V3 or V4.(2)

The definitive method of determining the type of Flash on the 41CL board requires a special command to query the Flash memory for the built-in "device code." This method will be covered later in this document.

41CL Memory Map

The standard memory maps for the 41CL Flash memory are available on the 41CL website. Before attempting to update your Flash memory you should print out the current memory map and decide exactly which images you plan to update or add. These memory maps contain the image .rom file name, the 41CL mnemonic (used for the PLUG commands), the XROM number, the name of the image, the CRC value for the image, and the revision date, if any. The revision date can be used as a first check for images that need to be updated, and the CRC value can be used as a final check if the image loaded into your machine is current. The CRC values are also useful for verifying that a downloaded image has been received without error.

V2 boards	http://www.systemyde.com/pdf/mem_ref_v2.pdf
V3/V4 boards	http://www.systemyde.com/pdf/mem_ref.pdf

Working with the Flash memory is a multi-step process, and having a printed copy of the memory map makes it much easier to keep track of what you are doing and where you are in the process. Before starting the update process make sure that you have copies of all of the .rom files that you intend to update and an executable copy of the CLWriter software on your PC.

Flash Memory Details

Flash memory is erased by sectors, which are usually 64K bytes in size (32K words, or eight pages) for the 41CL. But either the top or the bottom sector of the Flash memory is special, in that it is erased in pieces smaller than the entire sector. You will need to take this into account when working with the top or bottom sector.

For V2 boards, the special sector is broken into four pieces: 8K words (two pages), 4K words (one page), 4K words (one page) and 16K words (four pages). The table below shows the page addresses for these two cases.

Version	8K	4K	4K	16K
V2 "top"	0FE-0FF	0FD	0FC	0F8-0FB
V2 "bottom"	000-001	002	003	004-007

For V3 and V4 boards either the bottom or the top sector is broken into eight identical pieces of 4K words (one page). The table below shows the page addresses for these two cases.

Version	**4K**	**4K**	**4K**	**4K**	**4K**	**4K**	**4K**	**4K**
V3/V4 "top"	**1FF**	**1FE**	**1FD**	**1FC**	**1FB**	**1FA**	**1F9**	**1F8**
V3/V4 "bottom"	**000**	**001**	**002**	**003**	**004**	**005**	**006**	**007**

The *41CL Flash Functions* image contains a function that issues the special control sequence that causes the Flash to return the "device code," which uniquely identifies the organization of the Flash. The *41CL Flash Functions* are not needed very often, so they are not loaded into the 41CL Flash by default. Use the sequence below to load the *41CL Flash Functions* into RAM. Any legal RAM address can be used, and any Port can be used, but this example uses Page 831 in RAM and the upper half of Port 1. Starting from the MEMORY CLEAR state:

XEQ "MMUCLR"	Make sure that the MMU registers are cleared. The MEMORY CLEAR condition does not affect the MMU registers in memory.
XEQ "TURBO50"	The 50x Turbo mode is required for the 41CL to be able to keep up with a continuous stream of data from the PC as the new images are downloaded.
XEQ "SERINI"	Initialize the serial port. This command sets the serial port to Async mode and clears any error conditions.
XEQ "BAUD48"	Set the baud rate to 4800. **Don't forget that the selected baud rate is lost when the 41CL is turned off.**
"831000-0FFF"	Address/length specifying page 831 and a block length of 4k words (8k bytes).
XEQ "YIMP"	Import the data. On the PC, using COM3, we need to do: "CLWriter flash-1a.rom COM3 4800"
"831-RAM"	Direct address specification for Plug command
XEQ "PLUG1U"	Plug the RAM copy of the *41CL Memory Functions* into the upper half of Port 1.
XEQ "MMUEN"	Enable the MMU to access the downloaded functions.
XEQ "FLASH?"	Fetch the device code for the Flash.

The function FLASH? in the *41CL Flash Functions* returns the four-digit hexadecimal device code to the display and the ALPHA register. The table below shows the device ID values for the various Flash devices that have been used in the 41CL.

Version	device code	Flash device
V2 "top"	**22C4**	**M29W160ET**
V2 "bottom"	**2249**	**M29W160EB**
V3/V4 "top"	**2256**	**M29W320ET**
V3/V4 "bottom"	**2257**	**M29W320EB**

The *41CL Flash Functions* image contains two other functions that provide information useful when updating the Flash:

XEQ "IDBVER?"	Fetch the date encoded in the Image Database (IMDB).
XEQ "FDBVER?"	Fetch the date encoded in the Flash Database (FLDB).

Both of these functions return the date in MM/DD/YYYY format to the display and the ALPHA register. Note that early versions of both the IMDB and the FLDB did not contain date information, and if the date information is not present or is corrupt a "DATA ERROR" message will be returned to the display.

Flash Memory Cautions

Writing and erasing Flash memory requires that the 41CL circuit board draw more current than during normal operation, and Flash memories are not tolerant of power disruptions during write and erase operations.

Do not start any Flash write or erase if the BATT annunciator is on. It is best to have new batteries in the calculator when writing or erasing Flash memory. In addition, make sure that the 41CL circuit board is making good contact with the keyboard PC board through the flexible connectors. If there is any play between the case halves make sure to repair it before attempting a write or erase of Flash. The extra current required by a Flash write or erase operation may cause the connector to flex sufficiently to disrupt the power to the 41CL board, which in turn may damage the Flash memory. Never remove the battery during a Flash write or erase operation.

During a Flash memory write or erase, no other accesses of the Flash memory are allowed. This means that the code doing the erase or write must be resident in RAM to work. The Flash Erase (YFERASE) and Flash Write (YFWR) functions in the *41CL*

Extra Functions check for this, and return with an error message if they are not running in RAM. To use these functions you must copy the entire *41CL Extra Functions* image to RAM and then program the MMU to use this RAM copy of these functions.

The functions YFERASE and YFWR are also available in the *41CL Extreme Functions*, and these versions automatically copy the erase or write code to a dedicated area of RAM for execution, which makes them much easier to use. In addition, the *41CL Memory Functions* provide several advanced functions which make the updating process even easier. In what follows, we will first download the *41CL Memory Functions* to RAM so that we can use these advanced functions. Then we'll go through the process of updating the *41CL Extreme Functions* and *41CL Library Functions* in Flash.

Before Starting

For what follows, we will be using specific locations in RAM to hold copies of the images to write to Flash, and the image of the *41CL Extra Functions* used during Flash updates. We will also use specific locations in RAM to hold the Operating System (OS) for the special case of updating the sector of Flash that is normally protected against writes or erasure. There is nothing special about these choices of RAM locations, and you are free to choose differently, but using the RAM addresses here will help to reduce the possibility of error.

The eight RAM pages starting at address 0x810000 (pages 810 through 817) will be used to hold the data to be written to a Flash sector. New images will be transferred directly into this region at the appropriate page before the Flash update. The RAM page at address 0x8300000 (page 830) will be used to hold the *41CL Extra Functions* until we can switch over to using the *41CL Extreme Functions*.

You should never erase a Flash sector that is currently being used by the 41CL, so it is strongly recommended that you start the update procedure from the MEMORY LOST state. The previously mentioned test that prevents erasing the OS sector protects the default *41CL Extra Functions*, so you will always be able to recover from an erase error, but it is best to avoid the complications that arise from such an error.

At this point it is appropriate to connect the 41CL to the PC via the serial port. **Make sure the PC is on before connecting the serial port, and do not let the PC go to sleep, or power it down, with the serial port connected to the 41CL. The resulting transients can damage the 41CL.** Remember that the 41CL should be off when inserting the plug into the serial connector, because transients from the insertion often cause the 41CL power supply to glitch and reset the 41CL. I use the CLWriter program, available on the 41CL website, for all of the downloads. Although the 41CL is capable of 9600 baud I find that 4800 baud is more reliable, so that is what we will use here.

Updating the *41CL Memory Functions*

The *41CL Memory Functions* will be updated first because this simplifies the remainder of the update process. Starting from the MEMORY CLEAR state:

XEQ "MMUCLR"	Make sure that the MMU registers are cleared. The MEMORY CLEAR condition does not affect the MMU registers in memory.
XEQ "TURBO50"	The 50x Turbo mode is required for the 41CL to be able to keep up with a continuous stream of data from the PC as the new images are downloaded.
XEQ "SERINI"	Initialize the serial port. This command sets the serial port to Async mode and clears any error conditions.
XEQ "BAUD48"	Set the baud rate to 4800. **Don't forget that the selected baud rate is lost when the 41CL is turned off.**
"007>830"	Source and destination information for the memory copy.
XEQ "YMCPY"	Copy the default *41CL Extra Functions* to RAM.
"830-RAM"	Direct address specification for Plug command
XEQ "PLUG1L"	RAM copy of the *41CL Extra Functions* is plugged into the lower half of Port 1.
XEQ "MMUEN"	Enable the MMU so that the RAM copy of the *41CL Extra Functions* is active.

Next the latest *41CL Memory Functions* image needs to be downloaded to the 41CL. It is easiest to enter the commands on both the 41CL and the PC without hitting ALPHA (on the 41CL) or "Enter" (on the PC), and then pressing ALPHA on the 41CL to start the import and then hitting "Enter" on the PC to start the download. If the download doesn't start soon enough the 41CL will time out, requiring you to enter the commands again. If the baud rates on the two machines don't match you will probably get an "OVERFLOW" message on the 41CL. Clearing this condition requires that you initialize the serial port and set the baud rate to 4800 again.

"831000-0FFF"	Address/length specifying page 831 and a block length of 4k words (8k bytes).
XEQ "YIMP"	Import the data. On the PC, using COM3, we need to do: "CLWriter yfnf-1d.rom COM3 4800"

If your 41CL already has the latest version of the *41CL Memory Functions* you can skip the download and instead just copy the image to RAM at page 831. Once the image is in RAM we need to plug the image into a Port. Since the image is in RAM, we need to specify the memory address directly for the Plug function.

"831-RAM"	Direct address specification for Plug command
XEQ "PLUG1U"	Plug the RAM copy of the *41CL Memory Functions* into the upper half of Port 1.

If you downloaded the latest copy of the *41CL Memory Functions* to RAM, now is a good time to update the image in Flash. This image is located in the sector starting with page 160 for a V3 or V4 board, and page 0A8 for a V2 board. If you are updating a V2 board, change the Flash page addresses appropriately in the sequence below.

"160>810"	FLASH pages 160-167 to RAM pages 810-817.
XEQ "YMCPY8"	Copy eight pages (one sector). This copy will take a while, but the progress is shown in the display during the transfer.
"831>817"	RAM page 831 to RAM page 817.
XEQ "YMCPY"	Copy the new *41CL Memory Functions* image in page 831 to the correct spot in the RAM version of the sector to update.
"160000"	Specify the sector to erase (pages 160-167).
XEQ "YFERASE"	Perform the erase.
"810>160"	RAM pages 810-817 to FLASH pages 160-167.
XEQ "YFWR8"	Write the eight pages to Flash. This will take a while, but the progress is shown in the display during the transfer. Note that his function skips the page write if the page is empty (first word of 0xFFFF.)

It's that simple! The reason that we update the *41CL Memory Functions* first is because this is where the YMCPY8 and YFWR8 functions reside. Without these functions every page must be moved individually, which is both tedious and error-prone.

At this point is is useful to update the sector that contains the *41CL Image Database* and the *41CL Flash Database*. This is because the *41CL Flash Database* can be used to verify that any further updates are correct. These two images are located identically for all board versions, and the procedure is very similar to that just used for the *41CL Memory Functions*. Note that the image file names are different for the V2 board versions. Download fldb_v2.rom and imdb_v2.rom if you are updating a V2 board.

"0D8>810"	Flash pages 0D8-0DF to RAM pages 810-817.
XEQ "YMCPY8"	Copy eight pages.
"816000-0FFF"	Full 4k words at page 816.
XEQ "YIMP"	Import the data. On the PC, using COM3, we need to do: "CLWriter fldb.rom COM3 4800"
"817000-0FFF"	Full 4k words at page 817.
XEQ "YIMP"	Import the data. On the PC, using COM3, we need to do: "CLWriter imdb.rom COM3 4800"
"0D8000"	Specify the sector to erase (pages 0D8-0DF).
XEQ "YFERASE"	Perform the erase.
"810>0D8"	RAM pages 810-817 to Flash pages 0D8-0DF.
XEQ "YFWR8"	Write eight pages.

Now that the *41CL Flash Database* is present, we can use it to verify other images. First verify that the *41CL Image Database* is correct:

"0DF"	Flash page address to verify.
XEQ "FLDB?"	Use the *41CL Flash Database* to verify that the page is correct. The function should return "0DF MATCH" to indicate that the CRC values match.

Next we can go back and verify that the sector containing the updated *41CL Memory Functions* is all correct. As before, substitute the correct page addresses (0A8-0AF) in the case of a V2 board.

"160>167"	Pages 160-167
XEQ "FLDB?"	Verify the specified pages. This function takes a while, and will report either "MATCH" or DIFF" for each page. The 41CL will beep when the last page has been verified.

If any page is reported as "DIFF" the image in that page is either out-of-date or contains some kind of error according to the *41CL Flash Database*. If this is the case, go through the update procedure again, updating the out-of-date or errored image.

Note that you can always use the YCRC function (in the *41CL Extra Functions Plus* or *41CL Extreme Functions*) to check the CRC value of a page immediately after download. Since errors are rare, I prefer to do the check automatically using the FLDB? function after the Flash sector has been updated.

Updating the remainder of Flash

At this point you have downloaded two or three images to your 41CL and updated two sectors in Flash memory. The *41CL Memory Functions* and the *41CL Flash Database* make the Flash update process relatively painless. The steps to update a sector are the same for every sector except the protected OS sector:

First, copy the entire Flash sector to RAM. You can skip this step if you are adding all new images to a sector.

Second, download the new images to RAM. This is where it helps to always use the same RAM addresses for updates, and I always write the RAM page addresses on the memory map next to the Flash page addresses so that I don't make a mistake with image placement.

Third, erase the Flash sector. Always double-check that you have entered the correct sector address before executing the YFERASE command, because the command is not reversible. If you're working on the top sector and it is segmented you'll need to erase each subsector individually.

Fourth, write the eight pages of RAM to the Flash.

Finally, using the FLDB? function, verify that the images are correct in Flash.

I recommend that you update everything except for the OS sector to become familiar with the process before even attempting to update the OS sector. The next page has a handy table that I use when updating a single sector.

I use a printed copy of the Memory Reference when updating the entire Flash, taking care to remember the RAM addresses that correspond to each Flash address in a sector. The "Revision Date" column corresponds to the "Needs Update" column and I use the "YCRC value" column for the "Checked" information.

RAM Page	Flash Page	Contents	Needs Update	Checked
810				
811				
812				
813				
814				
815				
816				
817				

"_ _ _>810"	Flash pages to RAM pages 810-817.
XEQ "YMCPY8"	Copy eight pages.

"81_000-0FFF"	Full 4k words at RAM page.	Repeat these two steps for each page that needs updating.
XEQ "YIMP"	Import the data. Use CLWriter on the PC with the correct .rom file.	

"_ _ _000"	Specify the sector to erase.
XEQ "YFERASE"	Perform the erase.
"810>_ _ _"	RAM pages 810-817 to Flash.
XEQ "YFWR8"	Write eight pages.
"_ _ _>_ _ _"	Specify the sector to verify. The starting page address for a sector will end with either "0" or "8" and the ending page address for a sector will end with either "7" or "F"
XEQ "FLDB?"	Verify that the entire sector is now up-to-date.

STOP!

What follows is only for updating the sector of Flash that contains the Operating System.

If you have updated the remainder of Flash, there is really no need to continue, because the machine is perfectly usable without updating the OS sector.

If you mess up the update of the OS Sector your machine will be a brick. Do not proceed unless you are sure you know what you are doing.

Updating the OS sector of Flash

Before attemping to update the OS sector always make sure that you have the latest versions of the *41CL Extreme Functions* and *41CL Memory Functions* installed in Flash. What follows requires the latest versions.

The eight RAM pages starting at address 0x820000 (pages 820-827) will be used to hold the executing copy of the first sector of Flash (where the 41C Operating System resides) during the update of the OS sector. The two RAM pages starting at address 0x830000 (pages 830 and 831) will be used to hold the *41CL Extreme Functions* and the *41CL Memory Functions* when updating the OS sector. These images contain checks that prevent writing or erasing the OS sector, and these checks must be patched before the OS sector can be written or erased.

The OS sector update process is complicated by the fact that in some versions of the CL board the first sector of Flash is broken into multiple pieces that can be erased individually. Refer back to the "Flash Memory Details" section to determine the structure of the OS sector.

Once you have determined whether or not the OS sector is segmented, you can proceed with setting up for the OS update process. If the OS sector is segmented the procedure will be simpler, because only the pages that you want to update need to be erased. First we need to plug in RAM copies of the *41CL Extreme Functions* and the *41CL Memory Functions*. Starting from the MEMORY CLEAR state:

XEQ "MMUCLR"	Make sure the MMU is initialized.
XEQ "TURBO50"	Set 50x Turbo mode.
XEQ "SERINI"	Initialize the serial port.
XEQ "BAUD48"	Set the baud rate to 4800.
"00A>830"	Flash page 00A to RAM page 830.
XEQ "YMCPY"	Copy the *41CL Extreme Functions* to RAM.
"830-RAM"	Direct address specification for Plug command
XEQ "PLUG1L"	RAM copy of the *41CL Extreme Functions* is plugged into the lower half of Port 1.
"167>831"	Flash page 167 to RAM page 831. For a V2 board the Flash address should be "0AF."

XEQ "YMCPY"	Copy the *41CL Memory Functions* to RAM.
"831-RAM"	Direct address specification for Plug command
XEQ "PLUG1U"	RAM copy of the *41CL Memory Functions* is plugged into the upper half of Port 1.
XEQ "MMUEN"	Enable the MMU so that the RAM copy of the *41CL Extreme Functions* and *41CL Memory Functions* are active.

Next we need to copy the OS sector to RAM. First we copy the sector to the normal RAM area (starting at page 810) where we assemble new images. Theoretically we could also execute from this copy of the software during the update process, but it is safer to execute from a separate copy, so we also copy the OS sector to RAM starting at page 820.

"000>810"	Flash pages 000-007 to RAM pages 810-817.
XEQ "YMCPY8"	Copy the OS sector to RAM. This is the copy of the OS sector that will be updated.
"000>820"	Flash pages 000-007 to RAM pages 820-827.
XEQ "YMCPY8"	Copy the OS sector to RAM again. This is the copy of the OS sector that will be executed during the Flash update.

After these copy operations are done, we need to activate the RAM version of the OS via the MMU. Because we are plugging in RAM each page needs to be plugged in individually. The MAPEN function then activates the MMU translation for pages 0 through 3 and 5. But first we need to make sure that the MMU for the relevent pages are not locked. This can happen because the MMU registers for pages 0 through 3 are not affected by the MMUCLR command.

XEQ "UNLOCK"	Then enter "0" at the prompt.
XEQ "UNLOCK"	Then enter "1" at the prompt.
XEQ "UNLOCK"	Then enter "2" at the prompt.
XEQ "UNLOCK"	Then enter "3" at the prompt.
"-820 0"	Address and page.

XEQ "PPLUG"	Plug RAM page 820 into Page 0. This is the YFNX version of the Plug function.
"-821 1"	Address and page.
XEQ "PPLUG"	Plug RAM page 821 into Page 1.
"-822 2"	Address and page.
XEQ "PPLUG"	Plug RAM page 822 into Page 2.
"-823 3"	Address and page.
XEQ "PPLUG"	Plug RAM page 823 into Page 3.
"-826 5"	Address and page.
XEQ "PPLUG"	Plug RAM page 826 into page 5. Note that this might seem wrong, but this is where the Time Module image is located.
XEQ "MAPEN"	Enable the MMU translation for Pages 0-3 and 5. **Don't forget that like the baud rate, this special MMU enable does not persist if the 41CL is turned off.**

At this point you should go ahead and download the images you want to update into the RAM assembly area in pages 810-817. For example, to update the *41CL Extra Functions* to the latest version:

"817000-0FFF"	Full 4k words at page 817.
XEQ "YIMP"	Import the data. On the PC, using COM3, we need to do: "CLWriter yfnz-4e.rom COM3 4800"

Because the OS sector is critical, the downloaded images should always be verified before they are written to the Flash. Use the YCRC command on each individual page and compare the results with the 41CL Memory Map. For example, to verify the *41CL Extra Functions*:

"817"	RAM page 817.
XEQ "YCRC"	Compute the CRC. The CRC result must be manually compared to the 41CL Memory Map because the FLDB? function only knows about the correct CRC for pages that are in Flash memory.

Once you have verified that all of the images you plan to update are correct, it's time to patch the *41CL Extreme Functions* and *41CL Memory Functions* so that they are able to operate on the OS sector. This involves changing one location in the RAM version each of these images.

"830B14-0000"	Address and data for the *41CL Extreme Functions* patch.
XEQ "YPOKE"	Remove the OS sector check for YFERASE and YFWR. These two functions share a single check subroutine.
"83164C-0000"	Address and data for the *41CL Memory Functions* patch.
XEQ "YPOKE"	Remove the OS sector check for YFWR8.

Up to this point everything we have done is reversable. We are executing the 41C operating system out of RAM, the *41CL Extreme Functions* and *41CL Memory functions* are in RAM and have been patched to allow operating on the OS sector, and the updated images to write to the OS sector are in RAM and have been verified correct. The next few steps must be completed in their entirety, so read through them before starting so you know exactly what needs to be done.

The sequence to update the OS sector depends on the organization of the Flash, which we determined back at the beginning of this process. For a board with a first sector that is the full 64K only a few functions are required:

"000000"	Specify the sector to erase.
XEQ "YFERASE"	Perform the erase.
"810>000"	Need to write the updated sector data in RAM at page 810 back to the Flash.
XEQ "YFWR8"	Write eight pages.
"000>007"	Specify the sector to verify
XEQ "FLDB?"	Verify that the OS sector is now up-to-date.

For a board with a segmented first sector the process is almost exactly the same, except that each subsector must be individually erased and programmed. It is safest to only erase and program those subsectors that actually need updating. **Be very careful when working with a Flash that doesn't have uniform sizes for the subsectors. Only one YFERASE should be performed per subsector, or you will erase the pages that you just wrote.**

"00_000" **XEQ "YFERASE"**	Subsector to erase. Perform the erase.	**Perform this step only once per subsector!**
"81_>00_" **XEQ "YFWR"**	Source and destination Write the page.	Repeat these steps for each page that needs updating.

"000>007"	Specify the sector to verify
XEQ "FLDB?"	Verify that the OS sector is now up-to-date.

Once all the pages are reported as up-to-date, we're almost done! But we are still executing the OS out of RAM, and the *41CL Extreme Functions* and *41CL Memory Functions* in RAM are dangerous to keep around because they have been patched. So, we need to do the following to get the 41CL back to a benign state:

XEQ "MAPDIS"	Disable the OS mapping.
XEQ "MMUDIS"	Disable the MMU.
XEQ "MMUCLR"	Clear all of the MMU entries. This command does not clear the MMU entries for Pages 0 through 3, but that's okay because we will no longer be mapping the OS.
"830000-0000"	Address and data to erase the patched *41CL Extreme Functions*, just to be safe.
XEQ "YMCLR"	Do the erase.
"831000-0000"	Address and data to erase the patched *41CL Memory Functions*, just to be safe.
XEQ "YMCLR"	Do the erase.

Your 41CL is now completely updated and ready to go.

Notes

(1) There were a pair of V1 boards, but they were only used for Alpha testing.
(2) It is theoretically possible that locations 0x000000 and 0x100000 hold the same data in a V3 or V4 board, but with the standard memory map this is not the case.