41CL Clone Functions

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:

“HP-41C”, “HP-41CV”, “HP-41CX” and “HP” are registered trademarks of Hewlett-Packard, Inc. All uses of these terms in this document are to be construed as adjectives, whether or not the noun “calculator”, “CPU” or “device” are actually present.

Acknowledgements:

Sylvain Côté, Ángel Martin and Robert Prosperi served as beta-testers for this software.

Table of Contents

Introduction

The *41CL Clone Functions* allow one 41CL to act as the host computer during the update of a second 41CL. Obviously the 41CL running the *41CL Clone Functions* should be fully up-to-date, and you will need a null modem adapter to connect the two serial cables, but otherwise using a 41CL as the host computer is just as easy as using a computer running the *clupdate* software.

This manual will not attempt to expain the operation of the *41CL Update Functions*, so you should be familiar with those functions before trying to use a 41CL as the host computer for an update.

The *41CL Clone Functions* are independent of any other software, and can be plugged into a port on the 41CL on an as-needed basis. Any port (6 through F) can be used for the *41CL Clone Functions*.

The *41CL Update Functions* contains a function (**FLASH?**) that reports the size of the Flash memory on the 41CL calculator. Unfortunately, because of space constraints, this function returns the "device code," which must then be cross-referenced to a table in the manual for the *41CL Update Functions*. As an alternative the *41CL Clone Functions* contains an equivalent function that returns the Flash size and organization directly, so no look-up table is required.

Getting the *41CL Clone Functions* on your 41CL

The *41CL Clone Functions* are new in 2017, so the vast majority of 41CL machines do not have this software pre-loaded in Flash memory. This means that you will need to download this software to your 41CL before you can actually use it to update another 41CL. This section will detail how to do this. Note that if your 41CL is up-to-date the *41CL Clone Functions* will already be installed in the Flash Memory.

What follows requires the use of the serial port on the 41CL. If you did not install the serial connector you will need to find another way to update the Flash memory, because the *41CL Clone Functions* only use the 41CL serial port.

Starting from the MEMORY LOST state, do the follwing:

XEQ "MMUCLR"
Make sure that the MMU registers are cleared. The MEMORY CLEAR condition does not affect the MMU registers in memory.

"YFNZ"
Select the default *41CL Extra Functions* for the PLUG command.

XEQ "PLUG1L"
Plug the *41CL Extra Functions* into the lower half of Port 1. This doesn't take effect yet because the MMU is still disabled.

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.

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. Although the 41CL serial port is capable of 9600 baud, at this rate an inter-character gap is required, even with 50x Turbo mode. So 4800 baud is safer. **Don't forget that the selected baud rate is lost when the 41CL is turned off.**

"830000-0FFF"
Address/length specifying physical address 0x83000 (page 0x830) and a block length of 4k words (8k bytes).

XEQ "YIMP"
Import the data. See below for the commands on the host side. Start the serial transfer from the PC immediately after issuing this command. See below for examples.

"830-RAM"	Direct address specification for Plug command
XEQ "PLUG1U"	Plug the RAM copy of the *41CL Clone Functions* into the upper half of Port 1.
XEQ "MMUEN"	Enable the MMU.
CAT 2	Verify that both the *41CL Extra Functions* and the *41CL Clone Functions* are both plugged into the 41CL.

On the PC (using COM3 in these examples) you can either use the *CLWriter* software or the *clupdate* software. Note that you'll need to specify the current version of the *clupdate* software as well as the *41CL Clone Functions*. The examples below are current as of the date of this document.

The *CLWriter* software is a little less typing: "CLWriter UPHST-1A.ROM COM3 4800"

Using the *clupdate* software: "java -jar clupdate-1.0.0.jar --upload UPHST-1A.ROM COM3 4800"

Note on Batteries

Running the protocol state machine in the *41CL Clone Functions* requires that the batteries be in good shape, because the 41CL runs continuously in 50x Turbo mode while waiting for messages from the 41CL being updated. It is safest to install new batteries in the 41CL before starting an update session, particularly if you are checking and updating the entire Flash memory. Even if you are only updating a portion of the Flash memory, the serial port itself draws an extra 5mA from the battery while the protocol state machine is running.

The protocol state machine in *41CL Clone Functions* is invoked via the **CLONE** (*Clone 41CL*) function. This function can be aborted from the keyboard, and each time the keyboard is checked the battery voltage is also checked. If a low battery condition is detected the function is automatically aborted and the ***BAT*** indicator is activated in the display. If the ***BAT*** indicator is already on when this function is executed, the function will also abort, before executing.

Operation Details

During an update the 41CL being updated completely controls the update process, so for brevity this machine will be referred to as the **master** 41CL. The *41CL Clone Functions* implement the protocol state machine required to communicate with the *41CL Update Functions* running on the master 41CL. The machine running the *41CL Clone Functions* will be referred to as the **slave** 41CL.

The master 41CL usually requests a copy of the Flash Database (FLDB, stored in Flash memory at page address 0x0DE), from the slave 41CL and uses this information to create the Correlated Flash Database (CFLDB.) The CFLDB is returned to the slave 41CL and is used by the master 41CL to decide which pages to request from the slave 41CL.

The FLDB contains the two words of the expected CRC value for each page in Flash memory. Each page of Flash memory is allocated four words in the FLDB. By default the first two words are used to hold the CRC value, and the remaining two words both contain 0xFFFF, as shown in the table below.

FLDB entry word	default contents
1	CRC LSWord
2	CRC MSWord
3	0xFFFF
4	0xFFFF

The CFLDB is stored in RAM at page address 0x806 on both the master 41CL and the slave 41CL. The CFLDB replaces the 0xFFFF words with values that mark pages as up-to-date, or not up-to-date. The default 0xFFFF words identify a page as unverified. The various cases are shown in the table below.

CFLDB words 3 & 4	meaning
0xFFFFFFFF	default, unverified
0x00000000	verified up-to-date
any other value	needs updating

The CFLDB provides a way for both the software running on the master 41CL and the slave 41CL to know which pages in Flash memory need updating, and can be created on either machine.

The CFLDB is normally created automatically by the master 41CL using the **FLCHK?** function, or manually using the **CDBINV** function for each invalid page. It is also possible to create the CFLDB on the slave 41CL and then import it on the master 41CL for use during the update. This type of operation is only for advanced users and will be covered later in this manual.

As mentioned reviously, the *41CL Clone Functions* use a specific page in RAM to hold the CFLDB and imported pages, as shown in the table below. Make sure that you are not using this RAM page before executing any *41CL Clone Functions*, because any pre-existing data in this page will be lost.

page	Usage	
0x806	Correlated Flash Database	
0x810	page 0 of a Flash memory sector	Sector Buffer
0x811	page 1 of a Flash memory sector	
0x812	page 2 of a Flash memory sector	
0x813	page 3 of a Flash memory sector	
0x814	page 4 of a Flash memory sector	
0x815	page 5 of a Flash memory sector	
0x816	page 6 of a Flash memory sector	
0x817	page 7 of a Flash memory sector	

The CFLDB only resides in RAM, and should not be written to Flash memory in place of the regular FLDB. The *41CL Clone Functions* distinguish between the FLDB and the CFLDB, and will always use the correct version.

Although it is not part of the normal 41CL update process, the protocol state machine will accept pages transferred from the master 41CL. The table below shows the destination address within the sector buffer for the various source address possibilities.

source address	destination page
0xXX0 or 0xXX8	0x810
0xXX1 or 0xXX9	0x811
0xXX2 or 0xXXA	0x812
0xXX3 or 0xXXB	0x813
0xXX4 or 0xXXC	0x814
0xXX5 or 0xXXD	0x815
0xXX6 or 0xXXE	0x816
0xXX7 or 0xXXF	0x817

Two of the functions in the *41CL Clone Functions* beep to signal progress or completion. These beeps can be disabled by clearing Flag 26. The table below lists the various cases:

function	tone	meaning
CLONE	3	protocol error low battery error exit
	9	function complete
CDBCNT? **CLONE**	3	displaying "bad" count
	7	displaying "ok" count

Clone Functions

There are just four functions in the *41CL Clone Functions*, and in most cases you will only use one of these functions. Advanced users who are doing semi-automatic or manual updates might find the other three functions useful.

CLONE

Executing **CLONE** (*Run Update Protocol State Machine*) is equivalent to running the *clupdate* progam with the --*update* option. While the protocol state machine is running the slave 41CL responds to the master 41CL according to the table below. This function is not programmable.

master 41CL Byte(s) sent	Response	Action by slave 41CL
"A" (0x41)	**"B" (0x42)**	**Open Communication channel**
"C" (0x43)	**CFLDB page**	**Send 1Mx16 CFLDB**
"E" (0x45)	**CFLDB page**	**Send 2Mx16 CFLDB**
"G" (0x47)	**CFLDB page**	**Send 4Mx16 CFLDB**
"K" (0x4B)	**"L" (0x4C)**	**Receive 1Mx16 CFLDB**
"M" (0x4D)	**"N" (0x4E)**	**Receive 2Mx16 CFLDB**
"O" (0x4F)	**"P" (0x50)**	**Receive 4Mx16 CFLDB**
"Q" (0x51)	**"R" (0x52)**	**reserved**
"S" (0x53) + page no.	**page**	**Send specified page**
"U" (0x55) + page no.	**"V" (0x56)**	**Receive specified page**
"W" (0x57)	**"X" (0x58)**	**Close Communication Channel**
"Y" (0x59)	**none**	**stay awake**

The **CLONE** function will only wait for messages from the master 41CL for ten minutes. If no message is received from the master 41CL in this time the function aborts and signals completion with a tone 3 beep. This time limit is reset each time a message is received from the master 41CL.

This function continues to completion unless halted with the **R/S** key. The function can be stopped and restarted using the **R/S** key. The function is cancelled using the backspace key (while stopped), but this means that the update of the master 41CL will not complete.

The one difference between using the 41CL as a host computer and using an actual PC running the *clupdate* program is that the 41CL only has one FLDB available (the one in Flash memory at page address 0x0DE.) This means that there are limitations on updating between different versions of the 41CL. The table below shows the various combinations of master 41CL and slave 41CL. The FLDB is currently fully compatible across V3, V4 and V5 boards versions, although this may change in the future.

master 41CL running FLCHK?	**slave 41CL running CLONE**	**comments**
V2	V2	Allowed, but difficult in practice, because there is no way to easily turn on the serial ports
	V3 V4 V5	***VER ERR*** error message
V3	V2	***VER ERR*** error message
	V3	Allowed, but difficult in practice, because there is no way to easily turn on the serial ports
	V4 V5	Normal operation
V4	V2	***VER ERR*** error message
	V3	Must execute **SERON** on master 41CL to turn on the serial ports on both machines
	V4 V5	Normal operation
V5	V2	***VER ERR*** error message
	V3	Must execute **SERON** on master 41CL to turn on the serial ports on both machines
	V4	Works, but only the lower half of the Flash memory on the master 41CL can be updated
	V5	Normal Operation

Just as the display in the master 41CL reports the current state of the update process, the display on the slave 41CL reports the current state of the protocol state machine. Unlike the master 41CL, the slave 41CL does not beep to report progress, although it does beep to signal an error, completion, or when displaying the counts from the CFLDB. The table below shows a typical update session.

master 41CL	slave 41CL	comments
	WAITING	After executing **CLONE** on slave 41CL, the protocol state machine is waiting to open the communication channel. Any message other than the "Open Communication Channel" byte will cause a Tone 3 beep.
	READY	After executing **CMOPEN** on master 41CL the communication channel is open.
IMP 806	***EXP 0DE***	After executing **FLCHK?** on master 41CL, the CFLDB is transferred from the slave 41CL to the master 41CL.
CRC nnn	***READY***	The display on slave 41CL blinks after each CRC calculation or CRC compare is complete.
EXP 806	***IMP 806***	The CFLDB is returned to the slave 41CL.
	nnn OK	The number of up-to-date pages is reported on the slave 41CL. Tone 7 beep.
	nnn BAD	The number of out-of-date pages is reported on the slave 41CL. Tone 3 beep.
	READY	The slave 41CL is waiting.
CPY nnn	***READY***	After executing **FLUPD** on master 41CL, the update process begins.
IMP 81x	***EXP yyy***	The destination address (in the sector buffer) is reported on the master 41CL, while the source address in Flash memory is reported on the slave 41CL.
ERS nnn	***READY***	The slave 41CL is waiting while the master 41CL continues the update process.
WR nnn	***READY***	
	READY	Once the update is complete the slave 41CL is sill waiting, because it has no way of knowing that the update is complete.
		After executing **CMCLOSE** on master 41CL the communication channel is closed and the **CLONE** function terminates with a Tone 9 beep.

CDBCNT?

Executing **CDBCNT?** (*Count Entries in Correlated Flash Database*) counts the number of pages in the CFLDB that are marked as up-to-date and out-of-date. Unverified pages are not counted.

This function first displays the number of up-to-date pages, along with ***OK***, followed by the number of out-of-date pages, along with ***BAD***. This is identical to what the **CLONE** function reports upon receipt of the CFLDB from the Master 41CL. Each status is displayed for 10 seconds, but this time can be shortened by pressing **R/S**. A Tone 7 beep signals display of the up-to-date count, and a Tone 3 beep signals display of the out-of-date count.

This function is really only useful when creating the CFLDB on the slave 41CL, because these statistics are automatically reported by the **CLONE** function.

USECDB

Executing **USECDB** (*Use Correlated Flash Database*) selects the Correlated Flash Database (in RAM page 0x806) as the source of the CFLDB sent by the protocol state machine to the master 41CL.

Before issuing this command the CFLDB must be created. The easiest way to do this is to use the **FDB2CDB** function in the *41CL Update Functions*, along with the **PGINV** function to mark pages needing updating.

This option is useful if you want the slave 41CL to supply a pre-configured CFLDB, rather than creating the CFLDB on the master 41CL.

USEFDB

Executing **USEFDB** (*Use Flash Database*) enables the protocol state machine to use the normal FLDB in Flash as the source of the CFLDB it sends to the master 41CL.

This is normal operation and the default selection after the ***MEMORY LOST*** condition or when the calculator is turned on.

Flash ID Function

As mentioned reviously, the *41CL Clone Functions* contains a function that returns the Flash size and organization directly.

FSIZE?

Executing **FSIZE?** (*Read Flash Size/Organization*) issues the special control sequence that causes the Flash device to return the "device code," which uniquely identifies the organization of the Flash. This four-digit hexadecimal device code is then translated into the actual size and organization information, which is written to the ALPHA register, and in Run mode this information is also returned in the display.

The table below shows the **FSIZE?** message for the various Flash devices that have been used in the 41CL.

Version	device code	Flash device	**FSIZE?** message
V2 "top"	0x22C4	M29W160ET	***2MB TOP***
V2 "bottom"	0x2249	M29W160EB	***2MB BOT***
V3/V4 "top"	0x2256	M29W320ET	***4MB TOP***
V3/V4 "bottom"	0x2257	M29W320EB	***4MB BOT***
V5 "top"	0x227E	M29W640GT S29GL064NT	***8MB TOP***

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.

FSIZE?	8K	4K	4K	16K
2MB TOP	0x0FE-0x0FF	0x0FD	0x0FC	0x0F8-0x0FB
2MB BOT	0x000-0x001	0x002	0x003	0x004-0x007

For V3 and V4 boards either the bottom or the top sector is broken into eight identical pieces of 4K words (one page). For V5 boards only devices with the top sector segmented have been used. The table below shows the page addresses for these cases.

FSIZE?	4K	4K	4K	4K	4K	4K	4K	4K
4MB TOP	0x1FF	0x1FE	0x1FD	0x1FC	0x1FB	0x1FA	0x1F9	0x1F8
4MB BOT	0x000	0x001	0x002	0x003	0x004	0x005	0x006	0x007
8MB TOP	0x3FF	0x3FE	0x3FD	0x3FC	0x3FB	0x3FA	0x3F9	0x3F8

Error Conditions

With only five functions in the *41CL Clone Functions* there are not many possible error conditions. The table below lists all possible error messages returned by the *41CL Clon Functions*, along with the meaning of the message.

Error Message	Function	Meaning
CDB ERR	**CDBCNT?**	Invalid CFLDB
COM ERR	**CLONE**	Communication channel error
ID ERR	**FSIZE?**	Unrecognized device code
OVR ERR	**CLONE**	Receiver overrun
TIMEOUT	**CLONE**	No message from master 41CL in 10 minutes
VER ERR	**CLONE**	Invalid board version between master 41CL and slave 41CL

Internal Details

The table below shows the XROM numbers for functions in the *41CL Clone Functions*.

Function	XROM Number
CLONE	XROM 31,02
CDBCNT	XROM 31,03
USECDB	XROM 31,04
USEFDB	XROM 31,05
FSIZE?	XROM 31,06

In addition to the page in RAM used for the CFLDB, the *41CL Clone Functions* use dedicated locations in RAM to store information required for proper operation. These locations are identical to those used by the *41CL Update Functions*. The table below lists these locations and their use.

address	Usage	meaning
0x804031	Communications channel status	0xFFFF = open
0x804032	CFLDB valid	0xFFFF = valid
0x804039	FLDB/CFLDB Download Control	0xFFFF = CFLDB

Some of these memory locations are initialized by 41C polling points, according to the table below:

Polling Point	address	Usage	init value
Memory Lost	0x804032	CFLDB valid	0x0000
Memory Lost or Calculator On	0x804031	Communications channel status	0x0000
	0x804039	FLDB/CFLDB Download Control	0x0000

Revision History

09/27/2017	Original issue.
09/27/2017	First of the innumerable, inevitable corrections.
09/27/2017	Second of the innumerable, inevitable corrections.
10/01/2017	Added Tone stuff.
07/22/2019	Added FSIZE? function. Bumped revision level to -2A.
12/06/2019	Modified format for double-sided printing.