

Notes on the back story of this letter:

This is my second letter to **John McGeachie**, of the Australian **PPC Melbourne Chapter**, who previously had most kindly replied to my first batch of a zillion questions in a very comprehensive, handwritten multi-page letter where he strived to answer them all as best he could.

After a several-month hiatus and having joined **PPC** as he suggested in his reply, I took the opportunity to contact him again and show him my progression from "*utterly clueless newbie*" to "*accomplished knowledgeable member*" (or something like that), though that wouldn't save him from still more questions and some private comments.

Among the matters I discuss in this 2nd letter, there's a report on my attempt to create a **CP** (*Clear all Programs*) synthetics-based routine for the **PPC ROM**, including my alpha-level code for him to examine and tell me some way to make it work reliably. I also mention a **RAM test** routine, pseudo-random *seeds* generation, ask how to access microcode in order to create and call machine-language (*microcode*) routines (including the possibility of modifying microcode in **ROM***,) and the uses of some odd functions displayed by the *Flag 30 Catalogs* (**eGOBEEP**, **\$T+N IA**, etc.)

I then include my own opinions about some things I didn't like about **PPC** (asking for his comments), most of which were quite private and thus have been edited out, though not all. Besides, I include** my article about the powers of the synthetic pair **STO/RCL b**, describing many extremely useful applications (easily breaking **PRIVATE**, for instance,) but which **Mr. Nelson** (as usual) hadn't published, and one of my **HP-34C** programs which, again, went unpublished as well.

Finally, I offered exchanging **HP-41C** programs with him or any other interested **PPC MC** members, and mentioned my **Othello** playing program for the **HP-41C**, which I had sent to **PPC** for publication several months ago but, as I say in the letter, "*I don't believe they will ever publish it*".

Which proved prophetic because matter of fact, **Mr. Nelson** never published it.

Valentin Albiillo, 31-10-2021

* I was still so naïve that I considered the possibility of modifying **ROM** contents, which is blatantly impossible but, as I mention in the letter, made plausible by someone at Corvallis giving a friend a binary program for the **HP-85** computer that allowed him to change the original **BASIC** English messages to their Spanish equivalents. I took this as evidence for being able to change the **HP-85 BASIC** in **ROM**, and it would take me a year or two to get to know the **HP-85** system internals in detail and realize that translating the built-in English messages to Spanish required just *answering* a certain *system poll*, not modifying the **ROM** ! But to someone inexperienced with the **HP-85** and its polls, that "miraculous" translation seemed evidence enough.

** You can find all those materials included with the letters I sent to Richard Nelson for publication in **PPC**.

J. E. (John) McGeachie
Philosophy Department
Monash University
Clayton, Victoria
AUSTRALIA, 3168

Valentin Albillo (4747)
Padre Rubio, 61 - 2º C
Madrid 29
SPAIN

Dear John:

How are you? Do you recall me? I wrote you a letter last year asking about 67 NNN's and the like. Your kind reply was very useful to me, indeed, and I joined the PPC Club, following your suggestions. Since then, I have contributed several articles, programs and routines to the Club, some of them have been published in the May and June issues.

Well, when I received your letter, I was quite pleased with the idea of having gained a friend (a little far from home, but a friend anyway), so I want to share some ideas and comments with you, hoping you would give me some advice:

- a) I'm most interested about the PPC Custom ROM. I've already contributed some routines to be included in it. Since I've gained a good knowledge about synthetic programming (thanks to Wickes and your KA program), nothing dealing with the status registers and functions can frighten me. The last thing I've submitted to the PPC is a routine to clear all assignments at once, without disturbing program or registers. It clears from 0 to 70 registers in about 9 seconds (I didn't mean "registers" but "assignments"). It should fill a real need, I hope. Now, I'm interested in the following routine:

--name, CP

--purpose: to clear all programs in RAM at once

It will be useful: if you have, say, 4 programs in RAM and want to clear them all to make room, without causing MEMORY LOST, you must search their names in the catalog, then CLP (alpha) name (alpha), wait for PACKING to execute, then repeat the procedure four times. It is easier to simply XEQ "CP", and there you are.

Now, I have been trying to program that routine. Here is my last attempt:

LBL "CP"	"1-7"	16	X() c	x	
RCL c	0	/	RTN	+	-it is 55 lines, slightly
CLA	X() N	INT	LBL 01	END	over 112 bytes
X() M	STO M	193	text 3		
"1-1234"	RDN	-	RCL M		text 1 is F7C0002D00000000
X() N	XEQ 01	text 1	FRC		text 2 is F501690C00BF
"1-12"	X() Y	RCL M	10		text 3 is "append" 0000000001
RCL N	STO M	text 2	x		
CLA	RDN	RCL M	RCL M		, this is F67F0000000001
STO M	XEQ 01	X() c	INT		
"1-123456"	256	X() Y	1		
CIX	x	STO IND Z	x		
X() N	+	X() Y	16		

- the program is based on the article of Charles Close (3878) in V7N3P8-9. He says: (about inserting ENDS via bug 2) "... its 3rd byte is 2D, the created END with the permanent coding will become the new permanent END, and ALL instructions between it and the old permanent END will be lost after packing the file".

So, it seemed very simple: just insert and .END. (CO/00/2D) at the very top of program memory, then PACK, and all programs cleared. This is what the listing above tries to do. It search the address of the curtain between program and data, then stores a permanent END (CO/00/2D) in the top of program memory, then stops for the user to PACK. However, for unknown reasons, it does not work. It inserts the final .END. but after PACKING, all programs are lost, including the just created final .END., and GTO.. does not restore it as it was supposed to do. Several times, it is possible to regain the final .END., but something remains in program memory: LBL 10, LBL 14. Why?

I don't know what is wrong. The routine is supposed to be included in the ROM, so the routine does not clear itself. But, if executed in RAM, the routine can alter itself, and perhaps this causes the trouble, though I do not believe.

If you find a cure, or a different routine altogether, please forward it to the ROM progress column as soon as possible. It will be very useful if included in the ROM.

Some other ideas I'm trying to program for the ROM are: a RAM test. This seems impossible. To test the RAM it is necessary to store something into all registers, then recall to see if it is the same that was stored, while preserving the original contents of the register. But the normalization of non-stack registers makes this impossible. Any ideas? Random seed is possible? I tried to program a random seed generator, that used a combination of values in the status registers to arrive at a normal, numerical

value, but it didn't seem much random. It often converged to some value if called repeatedly.

Now, my interest are shifting to other themes. For instance: can microcode routines be accessed ? I know that ROMs have certain addresses to allow access from the 41c. Further, Bill Wickes has found a way to access directly ROMs (see V7N5P55). Now, some ROMs have both user level and microcode routines: the printer ROM has PRPLOT written in user level, together with REGPLOT, in microcode. If PRPLOT may be directly accessed, is the same possible with REGPLOT ? Further, can the 41c ROMs be accessed directly ? If they can, should we be able to use machine language routines written in microcode as subroutines ? Should we be able to modify something in the ROMs ? You must be thinking something as "ROMs cannot be modified, they are Read Only Memories". Yes, but perhaps some routines are not in the ROMs but in magnetic memories. I'll explain: a friend of mine, who works at HP, made contact with some folk at Corvallis. That folk gave him some binary programs for the HP-85, together with instructions and details. Using that programs, my friend was capable of accessing the BASIC language of the 85, and could make changes. For instance, the messages prompting for error are now given in Spanish, rather than English. You can't imagine the expression in the faces of all HP people who saw this changes !. Since then, I've tried to do the same for the 41c, or, at least, if changes can't be done, to access microcode routines.

Also related to this are the odd functions displayed by the flag 30 catalogs: I have found the real function of several, but to no purpose, as they seem to be many possible functions for each name, and many possible names for each function. The real problem is: how to generate them ? Your program, KA, generates quite a lot, but how - about the rest ? Are they useful, in general ? Do you know what the PRIVATE __ or the OO REG __ functions perform ? Another feat: everybody knows that ~~eg~~ BEEP __ generates XROM for the printer functions. Very alike, \$T+N IA ____, generates all XEQ local, including A,B,...,M,N,...,X,Y,...,d,e, if the prompt is filled with numbers. Now the question: can a similar function we found that generates all STO's and RCL's ? Imagine something as OO REG _____. If you fill the prompt, STO 00, STO 01,... then STO e are generated. Surely nothing can be more useful. A single assignment will do the same as a whole status card full of assignments. Then, the STO M, etc, will be almost directly executable functions. I'm still looking for it.

Well, that's enough stuff for a single letter. Now, I want to share some comments with you about PPC. Please, keep them confidential, they are only comments, not criticisms, and I would want to know your opinion about them. I am a little dissatisfied about

Examples:

It is too much, don't you think ?

He is always asking for 67/97 material to publish it. He always says that the reason for 67/97 programs not - appearing in the Journal is simply that member's don't send them in. That's not true: although I have no 67/97, but a 41c, I've sent no less than 4 programs for the 67. They do arrive at PPC, but no one has ever been published. The June issue has not a single 67 program, neither mine nor someone's else. It seems that

Finally, and more important,

Nothing more to say. Included are several programs submitted to PPC many

months ago. Also included is an article about the powers of STO b, RCL b, which also shows the easiest method I know of to break PRIVATE. Despite the incredible usefulness of the techniques provided by this article, it remains unpublished, too. Also included is one of 5 programs I sent for the 34c. Although Richard asked for them sometime ago, and guaranteed its publication, they weren't. It seems that has the exclusive for 34c programs.

Anyway, I can't be very much angry. PPC has published about 5 full pages from me in the last two issues; but I sent about 50 pages, not just 5. You may say, if they published your 50 pages, no one can be included in the issue but you. Certainly, right, but, why waste instead of, say "Systems of first order differential equations" for the 41c, specially when P. Fraundorf (1025) ask in V7N5P12 for such a program? I feel dissapointed, indeed.

Please, forgive me for all these comments. I've exceeded myself quite a lot. It's only that I have no one here to comment all those things, and the letters I sent to Nelson about this were never answered.

Nothing more, I hope you will be so kind as to answer me once more. Any comments will be greatly appreciated.

Sincerely

P.D: If any friend of you (or yourself) is interested in exchanging 41 c programs (specially math, games, utility routines), please, ask him to write me. I have many, many programs (quite good, though I say so myself). By the way, are you interested in an OTHELLO playing program for the 41c? It was sent to PPC for its publication, but I don't believe they will ever publish it. It includes full graphic capabilities, plays quite well, and is short (3 cards) and fast (30 moves in 25 minutes).