
Converting HP-41 Programs to PDF
Presented at HHC September, 2012
by Dr. Jackie F. Woldering

During the summer of 2011, I was cleaning my basement and thinking about how to downsize several collections of documents. Some could be eliminated, others could be scanned, still others would best be kept in original format. I knew how to deal with the books and papers that could be discarded or kept, but for those I might like to scan, I wondered if someone had already done the work. After searching the net, I found that almost all of the documents pertaining to Hewlett-Packard calculators, which I had collected since 1974, had already been scanned by Jake Schwartz, and put together on a single DVD, including 14 years of Richard Nelson's outstanding PPC Calculator Journal.

I was member 2864 of the PPC Users Group, and had kept the PPC Journals since January, 1978 when I joined (with back issues dating to May, 1976 to cover the HP-67/HP-97 calculators), until the bitter end, in July, 1987. In four binders, these took up a large milk crate worth of space, and the wisdom and technical data within those pages were worth keeping, but not necessarily in paper form. I bought Jake Schwartz's DVD, and was pleasantly surprised to find so much information in one disc, including HP-65 Notes, PPC Calculator Journal, PPC Journal, the PPC ROM Manual, CHHU Chronicle, HPX Exchange, HP Journal Articles, HPCC, Australian Newsletters and Conference Proceedings, all dating back to the early days of computing. When I started looking at this great treasure trove of information, however, I saw a minor problem. I still have an HP-41CX with way too many accessories, including the wand, and the pages of the PPC Journal, although scanned at a decent resolution, would not produce wand-readable barcode when printed from the DVD.

My first thought process was focused on how to solve this problem. Pages from the PPC Journals with barcode could be rescanned at higher resolution, but that seemed like a band-aid approach. I tested some of the pages from my original journals, and found that even at 600 DPI, the scans were not as reliable as I wanted. Some of this came from the original barcode not being correct (or perhaps my wand was failing?), but other problems came from the paper not being aligned exactly right while being scanned. Bottom line, this method could work, but would be tedious, require a lot of printing, and would enhance the value of PPC Journal only to HP-41 owners who still had a wand.

I began to look at some alternatives. What would it take to clean up or reproduce the barcode? MS Paint would work, but that would be an impossible task. Scanning the original barcode from the journals, and then finding a way to print that again in a more easily stored and more reliably readable format might be an alternative, but still this was limited to only those PPC Journal programs originally printed in barcode form. I read the 1981 HP book entitled Creating Your Own HP-41 Bar Code and saw what it took to produce the small black bars, including physical specifications, checksum calculations and more.

Once the barcodes were read from the original journals, they would reside in an HP-41CX, but the problem has always been how to move that data from the calculator to some other, more modern medium. I dusted off an HP-IL interface and an HP-82164A HP-IL/RS-232C (serial) interface, and after checking I was able to set the baud rate, number of bits, stop bits, and parity to then talk to the serial interface of a notebook computer. After taking care of the minor difficulty that my laptop computer only had USB, plus a small wiring question, I used a USB to serial interface from the laptop to the HP-82164A. Digging into the laptop, I found that the old Hyper Terminal (hypertrm) was the simplest serial interface based program that had data capture facility, and it was built into Windows. Experimentation with sending alpha strings by using OUTA proved that the link worked, but there was nothing built into the HP-IL Module that would transmit a program to the computer's serial interface.

I considered trying to redirect the target of WRTP or PRP to be the serial interface, but unlike the video interface, which understands PRP to print the named program to the screen, the HP-82164A would not recognize either of those commands. Besides, a program in tape or printer format wouldn't be too useful. As I continued to look into my set of HP-41C tools, I found that the HP-82183A Extended I/O Module had the desired functions to send a program to the serial interface, and to do so in a hexadecimal format that could be easily converted to barcode. The OUTP and INP functions were the key to sending and receiving.

With an HP-41CX, an HP-82153A Barcode Wand, an HP-82160A HP-IL Interface, an Extended I/O Module, an HP-IL Serial Interface, a USB to Serial Interface and a laptop, I was now able to capture barcoded programs and transmit them back and forth between the calculator and the computer. Still, after all this experimentation, I felt that this setup was still limited to bailing out the barcoded programs and saving them each in a hexadecimal character format file. Now I would have to learn how to format and print the barcodes, so it was back to the "Creating Your Own HP-41 Bar Code" Manual for some additional research.

The programs PRGMBR from BYTE Publications is presented on pages 17-31 of the Bar Code Manual, and comprises about 825 lines of BASIC language code. Translating this into something useable in C, C++ or Java would be a lot of work. Another possibility would be to take the format for Program Type Bar Code shown in Figure 1 on page 8 along with the Physical Specifications for size and shape shown in Figure 6 on page 13, the one-byte end-around carry running checksum calculation from page 43, and the flow chart for the header algorithms on page 49 and write my own program, but that would be difficult. The next step seemed like it might take months; I began to question the task.

It was at this point that I began searching the net for the possibility that someone else might have already created some sort of translation program. After several unsuccessful attempts and false starts, I found Warren Furlow's www.hp41.org site, and on it the HP41UC User Code Utility. This program was found by hunting for HP-41C barcode printing, then following a discussion, an explanation, and then a link on a user forum rather than by a direct search. Leo Duran's HP41UC converts between user code formats, compiles and decompiles user code, and, most importantly, generates barcode. This was what I needed!

Leo Duran's program translates between four important formats for our use. The /T or TXT format is the program as one would see it printed on a printer or on a video display, with line numbers and commands. Certain characters, such as sigma, angle sign, and not equal have to be translated to replacement characters, and there are a few anomalies with XROM numbers and such, but for the most part, this is a line numbered program. In fact, the program can be entered without line numbers, translated to another form, and then made to add the line numbers when translated back to TXT format. The /R or RAW format is the program code in a binary form, exactly as it would be loaded into the HP-41C or an emulator such as V41 from Warren Furlow's website. The /D or DAT format is the RAW file with each binary byte translated into a pair of hexadecimal digits. This textual file also contains a leading number of bytes and a trailing checksum, and is an exact duplicate of what is put out by OUTP. Finally, barcode printable /H (Hewlett-Packard PCL) or /S (Postscript) files complete the task by producing great barcodes. I tested both barcode formats, and found that the PCL format produced large, virtually unreadable files, the Postscript option generated files that could be easily understood and altered as necessary to combine code and data into a single, complete, printable unit.

With HP41UC, I realized that HP-41C programs could not only be scanned in, sent to the computer as a DAT file, then stored and reproduced as a PS file to be scanned in later to the HP-41C with a wand, but programs could be typed in directly to a text file (thus avoiding the tedious entry through a calculator keyboard), and translated to RAW, DAT and Postscript barcode files. I now had the facility to enter any HP-41C program ever written and make it useable by a variety of HP-41C/CV/CX configurations as well as by various emulators.

At this point, I gave some thought as to a form in which to store and later display the program code, and to make sure that each of the parts could be used by a maximum number of calculator or emulator configurations. Also, the idea of which programs would make good candidates for this treatment. I went back to the earliest HP-41C programs, in Volume 6, Number 4, July, 1979 with programs for Chebyshev Polynomials, Random Numbers, and the Standard Normal Distribution Curve. Each of these routines was no more than a few lines, and provided a nice test bed. It was not too difficult to type each program into the computer, convert it from TXT to RAW, DAT, and PS files, and then print the output. At first, I entered the programs with line numbers, then later learned that the programs could be entered without line numbers, converted to RAW format using the /K option to keep size to a minimum, then changed back to TXT form using the /N option to add line numbers, thereby reducing typing.

Storing the program as a .txt file, with the runnable program in binary format and the barcode in Postscript was useable, but not convenient, as any adjustments would have to be made to each format separately. One output file could be achieved but that would require combining program code, a directly runnable form of the program, and barcode into one document. The PDF Creator program from www.sourceforge.net was familiar to me, but it expected input in Postscript format. After realizing that the DAT format would be a reasonable substitute for the RAW file, the next step was to learn more about Postscript.

I studied the Postscript files as output by HP41UC, and observed that each page started with a page number, and each line contained location information along with a moveto command, left parenthesis, line content, right parenthesis and a show command. There were lines to end each page and to end the document. 70 text lines could be placed on each 8.5" x 11" letter-sized page starting at the top at vertical position 740.00 through 50.00 at the bottom, with starting horizontal position of 50.00 for half-inch margins all around, and a 10-point Courier (non-proportional) font for properly aligned text columns.

The task of adding the necessary meta-information for Postscript formatting could be performed via a program, but this seemed like another unnecessarily involved task. After giving this some more thought, I realized that an Excel spreadsheet could contain all the necessary information, and it would be much easier to create and use, allowing the combination of one or more program TXT and DAT file pairs, and the ability to vary the number of lines between them. This spreadsheet has the header and trailer information, as well as the line location and left parenthesis in column A, with the closing right parenthesis and show instruction in column C. Column D then combines columns A, B and C. The user simply splits the program into groups of 70 lines each, then cuts and pastes into column B. When all the files have been pasted into column B, the complete set of Postscript lines can be extracted from column D and then pasted into a Postscript barcode file to make a complete printable document.

Other considerations were based upon how to format the DAT file, which is normally stored as one continuous string of hexadecimal characters. I broke that up into 50 column groups and added it to the end of the TXT file, along with a byte count based upon the size of the RAW file. Since each byte of a RAW file is represented by two hexadecimal digits in the DAT file, and there are four hex byte count digits at the front and two for checksum at the back, the total size of the DAT file is $(2 * \text{RAW Byte Count}) + 6$. I added headers to each program, indicating a title, author, source document and any special modules or peripherals required, and various amounts of program comments. The only quirks to be found in the Postscript format included the need to precede any single backslash (\) character or right/left parenthesis with a backslash character, translation of the Sigma character to S in commands or tilde (~) in strings, and conversion of the HP-41 angle sign to \10.

I also made another Excel spreadsheet to help with relocation of barcode lines, but it is more a calculation assistant than an automated translation. It can be used to combine text and barcode on one page and save space overall.

After all of this, the real grunt work began. I spoke to Jake Schwartz in August, 2011 and asked him if he knew if anyone had ever produced text copies of the PPC programs, and mentioned the difficulties with reading the barcodes from scanned copies. He said that he had never heard of anyone having taken on this task, and that it would be a particularly tough one. We spoke again a year later, when I sent him over 550 files covering the first three and a half years of the PPC Journal and PPC Calculator Journal, from Vol. 6, No. 4, July, 1979 to Vol. 9, No. 8, Dec, 1982, all indexed with hyperlinks from four Excel spreadsheets divided by years. Each program is listed with a hyperlink, page number, program name, month and year, author, and PPC member number (for nostalgic reasons). I type quickly and reasonably accurately, and I've given much of my spare time during the past 12 to 15 months to accomplish this task.

Jake has had a chance to look at the files, and has suggested a method of incorporating them into the scans of the PPC journals that would allow quite seamless integration and navigation between similarly formatted PDF journal and program files. Just as the blue boxes surrounding a program or article title will take the user to that page, the indicator of a red box around a section of code will cue the user that there is a PDF document available with the program text, the file in DAT format, and program barcode. Once the user has navigated to that page, he or she may examine the program in a consistent neat format (no more squinting to try and figure out too-light thermal paper images, bad fonts, inconsistent or handwritten code), copy the DAT file from the PDF to a notepad file, eliminate the carriage returns, and then send this DAT file via the aforementioned HP-82164A serial interface or HP-82973A PC HP-IL Interface and HP-IL Module, or perhaps a USB PIL-Box to a real HP-41C. The DAT file with carriage returns removed can also be converted easily via Leo Duran's HP41UC to a RAW file for the V41 or other emulator, or you can print out pages containing barcode and then scan them with a barcode wand. One might also use an HP9114A/B 3.5" diskette drive and the "Shuttle Disk" LIF file method outlined in the README file accompanying the HP41UC program.

I have included a small program called PRINT, based on a program contained in the HP-82164A User's Guide with this document, both as a sample of output and to enter or scan to the HP-41C calculator for serial interface transfer. Also attached is an MS-Paint version of the Wand Alpha Character Table by Jake Schwartz, from PPCJ V7N6P23 as an example of an updated barcode table.

Jake said that he was asked "Why not just scan the PPC journals using OCR software and have a text-based document with index." Truthfully, I skipped this step entirely, and have typed in just about every one of these programs by hand, with the exception of a few which were scanned in from barcode, then transferred via the serial interface to my laptop and then checked, adjusted, ancillary files created, those files combined and printed to a PDF document. As we both discovered, the journals were printed in so many different fonts and almost every page had two columns of text, so OCR would not work well.

At this point, I have finished up Volume 10, Number 4, May, 1983 with 33 out of 75 editions of PPC Journal and PPC Calculator Journal completed. At this rate, I might be able to complete this task in another two years or so, depending upon available spare time and other commitments, unless there are volunteers willing to learn and help with the program entry and other work. It is not a glamorous job, but it does make many years of Journal programs available for use by all. I expect, when all the journals have been done, there will be approximately 1500 separate line entries from 75 PPC Journals. Copies of what has been completed so far has been included on the HHUC 2012 Conference flash drive indexed by Excel spreadsheets covering 1979 to 1983. If anyone discovers any errors, please email me at jackie@cis.csuohio.edu.

To conclude, this has been a bit of a tedious job, but one that has provided much learning about HP-41C regular and synthetic code, module and peripheral techniques, spreadsheets, Postscript, barcodes and various internal workings, and I hope that the results will make the HP-41C a more useful tool for all.

HP-41C RS-232C Interface Setup from HP82164A Manual
Uses HP82183A Ext. I/O, 82160A HP-IL, 82164A Serial

```
01 LBL "PRINT" ;Set up HP82164A RS-232 Interface
02 XROM 28,27 ;AUTOIO
03 "HP82164" ;Interface identity
04 XROM 28,28 ;FINDID
05 XROM 28,39 ;SELECT
06 XROM 28,38 ;REMOTE
07 "SBA" ;2400 Baud
08 XROM 28,35 ;OUTA
09 "P4" ;No parity
10 XROM 28,35 ;OUTA
11 "C2" ;Communication mode: Receive
12 XROM 28,35 ;OUTA
13 "SL7" ;No flow control
14 XROM 28,35 ;OUTA
15 XROM 23,34 ;NOTREM
16 LBL 01
17 "DATA?" ;Prompt for some characters
18 AON
19 PROMPT
20 AOFF
21 XROM 23,09 ;ALENGIO
22 X=0?
23 GTO 02
24 68 ;Decimal value of ASCII "D"
25 XROM 23,14 ;XTOAL
26 RDN
27 XROM 23,38 ;OUTAN - Send contents of Alpha (except leading "D")
28 GTO 01
29 LBL 02 ;Send Form Feed
30 12 ;Form Feed character
31 XROM 23,39 ;OUTXB
32 END
```

0051C000F6005052494E54A71BF748503832313634A71CA727
A726F3534238A723F25034A723F24332A723F3534C37A723A5
E202F5444154413F8C8E8BA5C967B3001618A5CE75A5E6B2B3
031112A5E7C20B0D6C

81 BYTES

HP-41C RS-232C Interface Setup Program from HP-82164A Owner's Manual

Program Registers Needed: 12

Row 1 (1 - 3)



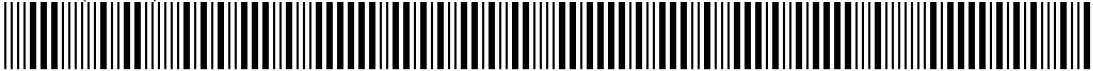
Row 2 (3 - 7)



Row 3 (7 - 11)



Row 4 (12 - 17)



Row 5 (17 - 24)



Row 6 (24 - 31)



Row 7 (32)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
	0 — ♦	1 天 ×	2 ☒ ×	3 ☒ ←	4 人 α	5 天 β	6 丁 Γ	7 ☒ ↓	8 ☒ Δ	9 ☒ σ	10 ☒ ♦	11 ☒ ↗	12 ☒ μ	13 ☒ ∠	14 ☒ ↖	15 ☒ ⚡
1																
	16 ☒ Θ	17 ☒ Ω	18 ☒ δ	19 ☒ ρ	20 ☒ α	21 ☒ ß	22 ☒ ä	23 ☒ Ö	24 ☒ ö	25 ☒ Ü	26 ☒ ü	27 ☒ Æ	28 ☒ œ	29 ☒ ≠	30 ☒ £	31 ☒ ⚡
2																
	32 (space)	33 !	34 " "	35 # #	36 \$ \$	37 % %	38 & &	39 ' '	40 < <	41 > >	42 * *	43 + +	44 , ,	45 - -	46 . .	47 / /
3																
	48 ☒ 0	49 ☒ 1	50 ☒ 2	51 ☒ 3	52 ☒ 4	53 ☒ 5	54 ☒ 6	55 ☒ 7	56 ☒ 8	57 ☒ 9	58 ☒ :	59 ☒ ;	60 ☒ <	61 ☒ =	62 ☒ >	63 ☒ ?
4																
	64 ☒ @	65 ☒ A	66 ☒ B	67 ☒ C	68 ☒ D	69 ☒ E	70 ☒ F	71 ☒ G	72 ☒ H	73 ☒ I	74 ☒ J	75 ☒ K	76 ☒ L	77 ☒ M	78 ☒ N	79 ☒ O
5																
	80 ☒ P	81 ☒ Q	82 ☒ R	83 ☒ S	84 ☒ T	85 ☒ U	86 ☒ V	87 ☒ W	88 ☒ X	89 ☒ Y	90 ☒ Z	91 ☒ [92 ☒ \	93 ☒]	94 ☒ ↑	95 ☒ _
6																
	96 ☒ T	97 ☒ a	98 ☒ b	99 ☒ c	100 ☒ d	101 ☒ e	102 ☒ f	103 ☒ g	104 ☒ h	105 ☒ i	106 ☒ j	107 ☒ k	108 ☒ l	109 ☒ m	110 ☒ n	111 ☒ o
7																
	112 ☒ p	113 ☒ q	114 ☒ r	115 ☒ s	116 ☒ t	117 ☒ u	118 ☒ v	119 ☒ w	120 ☒ x	121 ☒ y	122 ☒ z	123 ☒ π	124 ☒	125 ☒ →	126 ☒ Σ	127 ☒ f

Table 2. Top half of the hex table (see V6 N5 P22) including paper keyboard alpha character bar codes.

