HP41UC

User-Code Utility for HP-41 Calculators

Version 3.00
June 3, 2020

HP41UC – User-Code Utility for HP-41 Calculators

Document History

Date	Version	Change
June 11, 2016	2.00	1) Port v1.30 16-bit DOS to Win32 console app 2) Fix compiler bug on {STO*, STO/, STO+, STO-} IND 3) Add better support for Esc-sequences in strings (see De-compiler Notes) 4) Revise UsersGuide and License documents
June 19, 2016	2.10	1) Fix bug to allow compiling multiple files into a LIF file. 2) Add support for building with GNU tools 3) Build a statically-linked 32-bit x86 ELF executable for Linux.
June 22, 2016	2.20	1) Fix bug in parsing esc-sequence in text strings. 2) Allow conversion of multiple input files if output file is not specified, or if specified output is a LIF file.
June 25, 2016	2.30	1) Fix bug on default extension for input LIF files. 2) Merge conversion routines into a single convert().
Seo 4, 2016	2.31	Fix bug in convert.c – input data type on LIF files.
Sep 8, 2016	2.40	Expand XROM support in decompiler.
Sep 10, 2016	2.41	Bug fixes in file_splitpath, findfile_first and findfile_next.
Sep 11, 2016	2.42	1) Static-linked Win32 app that runs on WinXP and later. 2) Add Card Reader to list of supported XROM's
Sep 13, 2016	2.43	1) Remove non-programmable functions on XROM[30]. 2) Use static for local arrays to minimize stack growth.
Sep 23, 2016	2.44	1) Add support for external XROM files with "/m" option 2) Dump internal XROMs with "/x" option 3) Exclude internal XROMs in compiler with "/x" option 4) Clean-up to use macros instead of numbers: e.g., HP41_ERROR and HP41_OK as return codes.
June 3, 2020	3.00	Skip over Line Feed (0x0A) characters on input DAT files. NOTE: This feature does not require any extra arguments.

HP41UC

Contents

HP41UC

Introduction

This utility provides means for editing and archiving code outside of the HP-41™ calculator, using file transfer and barcode scanning mechanisms. To that end, the utility provides the following functions:

- File Converter
  The file format converter is used to package compiled code with wrappers compliant with various popular file formats. In some cases that may require preceding the code with a "header", appending a "checksum", or adding "filler" (don't care) bytes to satisfy file size requirements.

- Compiler
  The compiler takes code in readable text format, and compiles it into one of the supported file formats.

- De-compiler
  The de-compiler takes compiled code, packaged in any of the supported file formats, and creates a readable text file that can be edited or printed out.

- Barcode Generator
  The barcode generator creates a printable file with raster graphics data for PCL3™, or Postscript™ compatible printers.

For sample descriptions of file transfer methods, see the Shuttle Disk Transfers and Direct Link Transfers sections in this document.

IMPORTANT NOTES

The HP-IL hardware and transfer software described in the Shuttle Disk Transfers and Direct Link Transfers sections is now obsolete. However, the HP41UC use-cases described in those sections may still be applicable while using other HP-IL transfer mechanisms. For an example of a modern HP-IL transfer mechanism, please refer to The PIL-Box, an HP-IL <-> USB Solution.

HP41UC

Brief Overview

Supported File Formats

- LIF [/l]: transfer file for Trans41 and DOSLink
- P41 [/p]: archive file for HP-41 programs on ftp sites
- RAW [/r]: input/output file for V41 and LIFUTILS
- DAT [/d]: input/output file for EMU41 (using INP, OUTP)
- BIN [/b]: output file from 41UCC
- TXT [/t]: program listing (readable text)
- PCL [/h]: barcode printable file (HP PCL3™)
- PS [/s]: barcode printable file (Postscript™)
- WND [/w]: barcode hex-dump file (Wand)

Features

- File Converter
  [LIF, P41, RAW, DAT, BIN] → [BIN, DAT, RAW, P41, LIF]

- Compiler
  [TXT] → [LIF, P41, RAW, DAT, BIN]

- De-compiler
  [LIF, P41, RAW, DAT, BIN] → [TXT]

- Barcode Generator
  [RAW] → [WND, PCL, PS]

HP41UC

LIF Format

File format used by HP-IL mass-storage devices.

Description:
http://www.hp9845.net/9845/projects/hpdir/

Usage:
Optional parameters are surrounded by <>

On file conversions the utility will search the LIF file for a program name in this order:
1) <name>, if specified
2) Output file name (outfile), if specified
3) Input file name (infile)

- LIF file to P41 file
  ```
  /l=infile<.lif> /p<=outfile<.p41>> <name>
  ```

- LIF file to RAW file
  ```
  /l=infile<.lif> /r<=outfile<.raw>> <name> </k>
  ```

  Use /k option to exclude checksum and trailer bytes in RAW file.

- LIF file to BIN file
  ```
  /l=infile<.lif> /b<=outfile<.bin>> <name>
  ```

- LIF file to DAT file (hex digits in ASCII format)
  ```
  /l=infile<.lif> /d<=outfile<.dat>> <name>
  ```

- LIF directory listing
  ```
  /l=infile<.lif> <name>
  ```

  If <name> is specified, list only <name> program from LIF file.

- LIF file to TXT file (De-compiler)
  ```
  /l=infile<.lif> /t<=outfile<.txt>> <name> </a> </n> </x> </m=file1> </m=file2>
  ```

  De-compiler Options:
  ```
  /a - append text using: [ "|-text" ], instead of: [ >"text" ]
  /n - generate line numbers
  /x – exclude all supported XROM Modules (may generate [ XROM mm,ff ])
  /x## – exclude supported XROM module 1..31 (may generate [ XROM mm,ff ])
      See Supported XROM Modules for more details.
  /m=file<.txt> – import functions from XROM modules in specified text file
      See Importing XROM Modules for more details.
  ```

HP41UC

P41 Format

Archiving format popular on FTP sites.

Description:
[32-byte header] + [compiled code] + [1-byte checksum] + [trailer]
- [32-byte header] - single LIF directory entry
- [1-byte checksum] - sum of all [compiled code] bytes
- [trailer] - filler bytes, so that filesize = 32 + multiple 256-byte blocks

Usage:
Optional parameters are surrounded by <>

- P41 file(s) to LIF file
  ```
  /p=infile<.p41> /l<=outfile<.lif>>
  ```

  If output file is specified [infile] can be a directory of P41 files.
  For example: `/p=\p41dir\*.p41 /l=hpil.lif`

- P41 file to RAW file (remove 32-byte header)
  ```
  /p=infile<.p41> /r<=outfile<.raw>> </k>
  ```

  Use /k option to exclude checksum and trailer bytes in RAW file.

- P41 file to BIN file
  ```
  /p=infile<.p41> /b<=outfile<.bin>>
  ```

- P41 file to DAT file (hex digits in ASCII format)
  ```
  /p=infile<.p41> /d<=outfile<.dat>>
  ```

- P41 directory listing (from 32-byte header)
  ```
  /p=infile<.p41>
  ```

- P41 file to TXT file (De-compiler)
  ```
  /p=infile<.p41> /t<=outfile<.txt>> </a> </n> </x> </m=file<.txt>>
  ```

  ```
  De-compiler Options:
  /a - append text using: [ "|-text" ], instead of: [ >"text" ]
  /n - generate line numbers
  /x – exclude all supported XROM Modules (may generate [ XROM mm,ff ])
  /x## – exclude supported XROM module 1..31 (may generate [ XROM mm,ff ])
      See Supported XROM Modules for more details.
  /m=file<.txt> – import functions from XROM modules in specified text file
      See Importing XROM Modules for more details.
  ```

HP41UC

RAW Format

Description:
[compiled code] + [1-byte checksum] + [trailer]
- [1-byte checksum] - sum of all [compiled code] bytes
- [trailer] - filler bytes, so that filesize = multiple 256-byte blocks

Usage:
Optional parameters are surrounded by <>

- RAW file(s) to LIF file
    `/r=infile<.raw> /l<=outfile<.lif>> <name>`

    LIF name = <name> or [infile]

    If output file is specified [infile] can be a directory of RAW files.
    For example: /r=\rawdir*.raw /l=hpil.lif

- RAW file to P41 file
    `/r=infile<.raw> /p<=outfile<.p41>> <name>`

    P41 name = <name> or [infile]

- RAW file to BIN file
    `/r=infile<.raw> /b<=outfile<.bin>>`

- RAW file to DAT file (hex digits in ASCII format)
    `/r=infile<.raw> /d<=outfile<.dat>>`

- RAW file to TXT file (De-compiler)
    `/r=infile<.raw> /t<=outfile<.txt>> </a> </n> </x> </m=file<.txt>>`

    De-compiler Options:
    `/a` - append text using: ["|-text"], instead of: [>"text"]
    `/n` - generate line numbers
    `/x` – exclude all supported XROM Modules (may generate [XROM mm,ff])
    `/x##` – exclude supported XROM module 1..31 (may generate [XROM mm,ff])
        See Supported XROM Modules for more details.
    `/m=file<.txt>` – import functions from XROM modules in specified text file
        See Importing XROM Modules for more details.

HP41UC

BIN Format

Description:

[2-byte header] + [compiled code] + [1-byte checksum] + [trailer]

- [2-byte header] - size of [compiled code] in big-endian format
- [1-byte checksum] - sum of all [compiled code] bytes + [2-byte header]
- [trailer] - filler bytes, so that filesize = multiple 128-byte blocks

Usage:

Optional parameters are surrounded by <>

- BIN file(s) to LIF file
  ```
  /b=infile<.bin> /l<=outfile.lif> <name>
  ```

  LIF name = <name> or [infile]

  If output file is specified [infile] can be a directory of BIN files.
  For example: /b=\bindir*.bin /l=hpil<.lif>

- BIN file to P41 file
  ```
  /b=infile<.bin> /p<=outfile<.p41>> <name>
  ```

  P41 name = <name> or [infile]

- BIN file to RAW file (remove 2-byte header)
  ```
  /b=infile<.bin> /r<=outfile<.raw>> </k>
  ```

  Use /k option to exclude checksum and trailer bytes in RAW file.

- BIN file to DAT file (hex digits in ASCII format)
  ```
  /b=infile<.bin> /d<=outfile<.dat>>
  ```

- BIN file to TXT file (De-compiler)
  ```
  /b=infile<.bin> /t<=outfile<.txt>> </a> </n> </x> </m=file<.txt>>
  ```

  De-compiler Options:
  ```
  /a - append text using: [ "|-text" ], instead of: [ >"text" ]
  /n - generate line numbers
  /x – exclude all supported XROM Modules (may generate [ XROM mm,ff ])
  /x## – exclude supported XROM module 1..31 (may generate [ XROM mm,ff ])
      See Supported XROM Modules for more details.
  /m=file<.txt> – import functions from XROM modules in specified text file
      See Importing XROM Modules for more details.
  ```

HP41UC

DAT Format

Description:
[4-byte header] + [compiled code] + [2-byte checksum]
- [4-byte header] - size of [compiled code] in big-endian format
- [2-byte checksum] - sum of all [compiled code] bytes + [2-byte header]
All bytes are hex digits in ASCII format.

Usage:
Optional parameters are surrounded by <>

- DAT file(s) to LIF file
  ```
  /d=infile<.dat> /l<=outfile<.lif>> <name>
  ```

  LIF name = <name> or [infile]

  If output file is specified [infile] can be a directory of DAT files.
  For example: hp41uc /b=\datdir*.dat /l=hpil.lif

- DAT file to P41 file
  ```
  /d=infile<.dat> /p<=outfile<.p41>> <name>
  ```

  P41 name = <name> or [infile]

- DAT file to RAW file (remove 2-byte header)
  ```
  /d=infile<.dat> /r<=outfile<.raw>> </k>
  ```

  Use /k option to exclude checksum and trailer bytes in RAW file.

- DAT file to BIN file
  ```
  /d=infile<.dat> /b<=outfile<.bin>>
  ```

- DAT file to TXT file (De-compiler)
  ```
  /d=infile<.dat> /t<=outfile<.txt>> </a> </n> </x> </m=file<.txt>>
  ```

  De-compiler Options:
  ```
  /a - append text using: [ "|-text" ], instead of: [ >"text" ]
  /n - generate line numbers
  /x – exclude all supported XROM Modules (may generate [ XROM mm,ff ])
  /x## – exclude supported XROM module 1..31 (may generate [ XROM mm,ff ])
      See Supported XROM Modules for more details.
  /m=file<.txt> – import functions from XROM modules in specified text file
      See Importing XROM Modules for more details.
  ```

HP41UC

TXT Format (Compiler)

Command-line Options:
- /n – ignore line numbers in text file: [100 STO 00]
- /g – global alpha label for ["A..J", "a..e"] if in quotes: [XEQ "A"]
- /x – exclude all supported XROM Modules (may require: [XROM mm,ff])
- /x## – exclude supported XROM module 1..31 (may require: [XROM mm,ff])
  See Supported XROM Modules for more details.
- /m=file<.txt> – import functions from XROM modules in specified text file
  See Importing XROM Modules for more details.

Usage:
Optional parameters are surrounded by <>

- TXT file(s) to LIF file
  /t=infile<.txt> /l<=outfile<.lif>> <name> </n> </g> </x> </m=file<.txt>>

  LIF name = <name> or [infile]

  If output file is specified [infile] can be a directory of TXT files.
  For example: /t=\txtdir*.txt /l=hpil.lif

- TXT file to P41 file
  /t=infile<.txt> /p<=outfile<.p41>> <name> </n> </g> </x> </m=file<.txt>>

  P41 name = <name> or [infile]

- TXT file to RAW file (remove 2-byte header)
  /t=infile<.txt> /r<=outfile<.raw> </k> </n> </g> </x> </m=file<.txt>>

  Use /k option to exclude checksum and trailer bytes in RAW file.

- TXT file to BIN file
  /t=infile<.txt> /b<=outfile<.bin>> </n> </g> </x> </m=file<.txt>>

- TXT file to DAT file (hex digits in ASCII format)
  /t=infile<.txt> /d<=outfile<.dat>> </n> </g> </x> </m=file<.txt>

HP41UC

Supported XROM Modules

The Compiler will recognize and the Decompiler will generate function encodings from this list of XROM modules:

- [17] PLOTTER 1A-
- [18] PLOTTER 2A-
- [22] -HP-IL DEV
- [23] -X MASS 1A
- [23] -X EXT FCN
- [23] -X CTL FNS
- [23] -ADV CTL FN
- [24] -HP-IL DEV
- [25] -EXT FCN 2D
- [25] -CX EXT FCN
- [26] -TIME 2C
- [26] -CX TIME
- [27] -WAND 1F
- [28] -MASS ST 1H
- [28] -CTL FNS
- [29] -PRINTER 2E
- [30] CARD RDR 1E

Command-line Options (if no other options are specified):
- /x – dump the list of all supported XROM modules
- /x## – dump the list of functions from supported XROM module 1..31

For example:
1) To list all supported modules: hp41uc /x
2) To list functions on supported module 17: hp41uc /x17

Importing XROM Modules

XROM modules may be imported from one or more ASCII text files, with each file containing one or more XROM definitions:

- Entries in the text file are single lines of the form: XROM mm,ff
- Comments are allowed preceded by ";" or "#": [# comment]
- Quotes around function names are optional, but required if names contain space characters
- Commas or whitespace can be used as delimiters between arguments
- Functions names are limited to 7 alphanumeric characters
- Module ID's (mm) are limited to numbers 1 to 31 in decimal format
- Function ID's (ff) are limited to numbers 0 to 63 in decimal format

Command-line Option:
    /m=file<.txt> – import functions from XROM modules in specified text file

Sample entries in XROM file:
```
    # MATH 1C
    "MATRIX", 1, 1
    SIMEQ, 1, 2
    VCOL 1 3
    VMAT, 1, 4
    PVT, 1, 5
```

Sample command-line for Compiler (TXT → RAW):
```
/t=infile<.txt> /r<=outfile<.raw>> </n> </g> </x> </m=file1<.txt>> </m=file2<.txt>>
```

Sample command-line for De-compiler (RAW → TXT):
```
/r=infile<.raw> /t<=outfile<.txt>> </a> </n> </x> </m=file1<.txt>> </m=file2<.txt>>
```

NOTE: If the module ID from an imported XROM is the same as one of the supported XROMs, the supported module may be excluded by using the **/x##** option. For example, the Advantage ROM uses the same 22 and 24 ID's as the supported HP-IL Development ROM, so using **/x22 /x24** would exclude the HP-IL Development ROM.

Compiler Rules

- Prefixes are case-insensitive: [rcl 00] same as [RCL 00]
- Postfixes are case-sensitive: [lbl a] different from [lbl A]
- Append text is allowed with: [>"text"], ["|-text"], or ["\-text"],
- Synthetic instructions are allowed: [rcl M], [sto R]
- Synthetic NOP (TEXT0) is allowed using an empty (just quotes) text line: [""]
- Synthetic text is allowed with C-style esc-sequences: ["\x0D\x0A"]
- Comments are allowed preceded by ";" or "#": [; comment]
- Unrecognized XROM module functions must be encoded as: [XROM mm,ff]
- Key assignment for global labels are supported: [lbl "label" Key: 15]
- Multiple instructions per line are allowed if separated by comma+space: [RCL 00, STO 01]

Recognized C-style Escape Sequences within text, in quotes: [" …\Esc…"]

Esc-Sequence	Code/Character	Description
\a	0x07	Alert Bell
\b	0x08	Backspace
\f	0x0C	Form Feed
\n	0x0A	Newline
\r	0x0D	Return
\t	0x09	Horizontal Tab
\v	0x0B	Vertical Tab
\?	?	Question Mark
\'	'	Single Quote
\"	"	Double Quote
\\	\	Backslash
\xhh, \hh	0xhh	h = Hex digit (0-9, A-F, a-f)

NOTE: The Octal ("\ooo") esc-sequence is not supported.

HP41UC

De-compiler Notes

- Empty alpha strings are commented for clarification:

  "" ;F0   (single-byte synthetic NOP)

  GTO "" ;1D F0
  XEQ "" ;1E F0

- Spare instructions generate a comment-line showing the decoded hex codes:

  ;GTO IND __ (SPARE) ;AF,45

- Non-printable ASCII characters (0x00..0x1F, 0x7F..0xFF) within strings generate an esc-sequence and a comment with a ".":

  "Text\x0D\x0A" ; "Text.."

  LBL "abc\x07fg" ; "abc.fg"

  Note: This allows the produced text to be re-compiled using the embedded esc-sequence.

- Global-alpha labels with key assignments generate a comment:

  LBL "label" ; Key: -15   (label assigned to shift-LN key).

- Recognized XROM functions generate a comment:

  ACA ;XROM 29,01

Barcode Generator

References:
HP document 85153-90019: *Creating Your Own HP-41 Barcode*

A HP-41 Bar Code Decoder:
http://www.mh-aerotools.de/hp/bc-reader/BarcodeReader.pdf

Usage:
Optional parameters are surrounded by <>

- RAW file to PCL (HP raster graphics) file
  ```
  /r=infile<.raw> /h<=outfile<.pcl>> <"Program Title">
  ```

  To print the barcode, copy the output file to the printer device; for example:
  ```
  ➢  hp41uc /r=prog.raw /h "Program Title"
  ```

  ```
  On a DOS-compatible PC:
  ➢  copy /b prog.pcl lpt1
  ```

  ```
  On a standard Windows PC:
  ➢  copy /b prog.pcl "\\Computer Name\Printer Name"
  ```
  See Printing Barcode Files on a Windows PC below.

- RAW file to Postscript file
  ```
  /r=infile<.raw> /s<=outfile<.ps>> <"Program Title">
  ```

  To print the barcode, copy the output file to the printer device; for example:
  ```
  ➢  hp41uc /r=prog.raw /s "Program Title"
  ```

  ```
  On a DOS-compatible PC:
  ➢  copy /b prog.ps lpt1
  ```

  ```
  On a standard Windows PC:
  ➢  copy /b prog.ps "\\Computer Name\Printer Name"
  ```
  See Printing Barcode Files on a Windows PC below.

- RAW file to Wand (barcode) file
  ```
  /r=infile<.raw> /w<=outfile<.wnd>> <"Program Title">
  ```

  The output file will be a 16-byte per row hex-dump in readable text format.

Printing Barcode Files on a Windows PC
- Enable "Sharing" on "Printer properties".
- Refer to the Send PRN File to Printer article on *Techni*pages.
- Consider using the freeware PrintFile utility.

HP41UC

The PIL-Box: An HP-IL <-> USB Solution

http://www.jeffcalc.hp41.eu/hpil/index.html

The PIL-Box is an HP-IL/USB translator. It uses NO OBSOLETE HP PARTS such as the HP-IL chip or the custom HP-IL pulse transformer. It aims to be an alternative to the old ISA HP-IL board (HP82973A) and to be compatible with most computers/OS that support USB.

Shuttle Disk Transfers: Legacy Floppy-Disc Solution

Required Hardware

- HP-IL module for the HP-41
- 9114 HP-IL disc drive
- 720K (dual-density) diskettes
  For more reliable results use single-sided, dual-density disks.

Required PC Software

- LIFUTILS
  See References section at the end of this document.

Figure 1. Shuttle Disk Transfers

Notes on LIFUTILS

Since LIFUTILS expects a Volume label in the media, and the HP-41 format function (NEWM) does not create a volume label, it is recommended that disks be formatted on the PC using LIFUTILS. If you have problems reading a disk formatted on the PC try changing the BIOS setting for the drive from 1.44M to 720K.

An alternate solution is to format the disk from the calculator with the 9114 drive. In this case, you must create a volume label for the disk. This issue was addressed in the PPC Journal article: "The Volume Label", V14N2 Feb 1987, by Cristian Rusquelles. For more information, refer to the program listing included in Appendix B of this document.

PC to HP-41 Shuttle Transfer

Let's say you've downloaded some P41 files from the internet, and want to transfer the programs in those files to your calculator using a Shuttle Disk:

1. Remove the 32-byte header from each of the P41 files, and make a note of the LIF directory information in the header (name, size, etc):
   ➢ `hp41uc /p=infile.p41 /r=outfile.raw`

   HP41UC will remove the 32-byte header, and will display the LIF directory information.

2. Transfer each program, to the Shuttle Disk using LIFUTILS. For more information, refer to Dan McDonald's article listed in the References section at the end of this document.

3. Insert the shuttle disk in the 9114 disc drive, and the read the program from the calculator using the READP function. Be sure to match the program name exactly as saved by LIFUTILS, as the name is case-sensitive.

HP-41 to PC Shuttle Transfer

Let's say you have a program on the calculator that you want to edit on a PC, and then transfer back to the calculator using the previous procedure:

1. Calculate the program size in bytes. On a HP-41CX, you can simply use the CATALOG 1 function, watching for the number listed at the "END" indicator of your program. For other machines refer to a discussion on this subject in A Program Byte Counter. This is a well-documented subject covered by many other HP-41 books and articles.

2. Copy the program to the Shuttle Disk using the WRTP function.

3. Insert the Shuttle Disk on the PC diskette drive, and extract the program using LIFUTILS. The output file should be a RAW file with a size of multiple 256-byte blocks. For example, a 56-byte program will be extracted as a 256-byte RAW file.

4. If you're interested in sharing the program, you may convert the file to P41 format:
   ➢ `hp41uc /r=infile.raw /p=outfile.p41 NAME`

   The optional [NAME] parameter will be the name used in creating the 32-byte LIF directory header. If [NAME] is not specified, the input (RAW) filename is used (with all CAPS).

5. To edit the program on the PC, you need to de-compile it:
   ➢ `hp41uc /r=infile.raw /t=outfile.txt`

   If you'd like to print it, you may want to add the [/n] option to generate line numbers:
   ➢ `hp41uc /r=infile.raw /t=outfile.txt /n`

6. Compile the program to a RAW file that can be used by LIFUTILS:
   ➢ `hp41uc /t=infile.txt /r=outfile.raw`

HP41UC will display the LIF information required by LIFUTILS (name, size, etc).

Note: If you added line numbers to your de-compiled file, you must alert the compiler to ignore line numbers by using the [/n] option:
➢ `hp41uc /t=infile.txt /r=outfile.raw /n`

7. To transfer the compiled program to the calculator using LIFUTILS, simply go to step #2 of the previous PC to HP-41 Shuttle Transfer procedure.

Direct Link Transfers: Legacy ISA-bus Solution

Required Hardware

- HP-IL module for the HP-41
- 82973A (or compatible) - HP-IL Card for the PC

Required PC Software

- Trans41, DOSLink, or other HP-IL transfer software.
  See References section at the end of this document.

Note: Contrary to popular belief, a fast PC can be used with an HP-IL card. The requirement for a slow PC, as the original IBM AT, is strictly a software limitation (timing loops that are CPU-speed dependent) with most HP-IL transfer programs, but it is not an issue with Trans41.
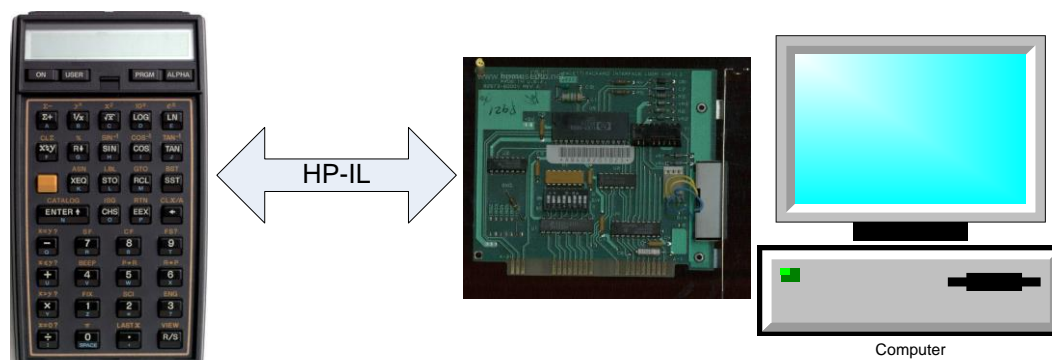

Figure 2. Direct Link Transfers

HP41UC

PC to HP-41 Trans41Transfer

Assuming that you have created a LIF file for your program (e.g., test.lif), you can transfer it to the calculator with the following procedure:

1. Over-write Trans41's transfer file with the created LIF file:
   ➢ `copy test.lif hpside1.pak`

   You may also rename the existing transfer file to save its contents:
   ➢ `rename hpside1.pak backup1.pak`
   ➢ `rename test.lif hpside.pak`

   Or you may simply over-write the transfer file in the compilation step:
   ➢ `hp41uc /t=test.txt /l=hpside1.pak`

2. Establish the HP-IL link in Trans41 using the HPSIDE1.PAK transfer file, and simply download the program using the READP function on the calculator!

PC to HP-41 Trans41 Transfer

Let's say you have a program on the calculator (e.g.,TEST), that you want to edit on your PC, and later transfer back to the calculator.

1. Establish the HP-IL link with Trans41 using the HPSIDE1.PAK transfer file.

2. Upload the program to the PC using WRTP on the calculator. This will copy the program to HPSIDE1.PAK on the PC side.

3. Extract and de-compile the program from HPSIDE1.PAK:
   ➢ `hp41uc /l=hpside1.pak /t=test.txt TEST`

   Note: The program name is case-sensitive, and must match the ALPHA label used with WRTP.

4. Edit test.txt, and compile back it to a LIF file:
   ➢ `hp41uc /t=test /l`

   Note: In this case, HP41UC will create a default output file: test.lif, with a default LIF program name: TEST (input filename, with all CAPS).

Appendix A - 41UCC Compatibility

The following instructions are allowed by the HP41UC compiler for compatibility with 41UCC.

41UCC Format	HP41UC Format
10**X	10^X
APPEND, APPND, APND "Text"	>"Text"
CLSIGMA	CLS
D->R	D-R
E**X	E^X
E**X-1	E^X-1
ENTER^	ENTER
GOTO	GTO
P->R	P-R
R->D	R-D
R->P	R-P
SIGMA+	S+
SIGMA-	S-
SIGMAREG	SREG
STO*, STO/, STO+, STO-	ST*, ST/, ST+, ST-
T "Text String"	"Text String"
X!=0?, X<>0?	X#0?
X!=Y?, X<>Y?	X#Y?
X**2	X^2
Y**X	Y^X

HP41UC

Appendix B - Listing of "WL"

Copied from the PPC Journal article: "The Volume Label", V14N2 Feb 1987 - by Cristian Rusquelles.
Requires an Extended I/O module.

```
01 LBL "WL"
02 >"     "   ; append 6 spaces
03 ASTO 00
04 LBL 00
05 AID
06 16
07 X=Y?
08 GTO 01
09 "NOT A DRIVE"
10 PROMPT
11 GTO 00
12 LBL 01
13 CLRDEV
14 3
15 DEVL
16 2
17 OUTXB
18 6
19 DEVL
20 "D"
21 ARCL 00
22 OUTAE
23 8
24 DEVL
25 END
```

Appendix C – References

Ulisse Quadri
For developing a user-friendly GUI for HP41UC
http://www.ulisse.bs.it/museo/hp41/tutorial.htm

Dan McDonald
For his article: "Ancient Data - Modern Machines:
Handling HP-IL Files on Today's PC", posted at the Museum of HP calculators under:
"HP-IL Files: Where, What, Why, How, etc.".
http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/articles.cgi

Warren Furlow
For his support of the HP-41 (V41 Emulator) and his constructive feedback to this project.
http://www.hp41.org.

LIFUTILS
Open-source project on GitHub: https://github.com/bug400/lifutils
See the Shuttle Disk Transfers section for a description of its use.

Trans41
HP-IL Transfer utility by Eclipse Logic Inc, and Tacit Logic Systems.
See the Direct Link Transfers section for a description of its use.

DOSLink
HP-IL Transfer utility by Southern Software.
Requires the use of a slow PC (AT compatible: 286/16MHz).

41UCC
User-code compiler written by Leslie Brooks & Jim De Arras.
Distributed by Hand Held Products, Inc.

EMU41
http://www.jeffcalc.hp41.eu/emu41/

The PIL-Box, an HP-IL <-> USB solution
http://www.jeffcalc.hp41.eu/hpil/index.html

A Program Byte Counter
"HP-41 Extended Functions Made Easy", by Keith Jarett. Chapter 5, page 97.

Listing of "WL"
PPC Journal article: "The Volume Label", V14N2 Feb 1987, by Cristian Rusquelles.

HP41UC

Trademarks

HP-41 and PCL3 are trademarks of the Hewlett Packard Company.

Postscript is a trademark of Adobe Systems Incorporated.