NutIP version 1A
TCP/IP stack for HP-41

User's Manual

Khanh-Dang Nguyen Thu Lam

March 22, 2009

Contents

Section 1

Introduction

1.1 Foreword

NutIP is a free minimal TCP/IP stack for HP-41C/CV/CX calculators. It implements encoding and decoding TCP/IP packets and connects your HP-41 to the Internet, allowing it to exchange data with any Internet host. From a user perspective, *NutIP* implements the following features:

- a simple DNS resolver (FINDIP);
- a web server (WWWD) which can serve user programs in a readable form.

NutIP also implements response to ICMP echo requests packets (the so-called “pings”).

By now, only IP version 4 is implemented. *NutIP* does not support IP fragmentation; if IP fragments are received, they are discarded.

NutIP is released under a BSD license (see annexe A). Source code in MCODE assembly language is provided.

1.2 Historical background

The idea of the *NutIP* project arose naturally. This year, 2009, is the 30th anniversary of the HP-41, first announced on July 16, 1979. Even though the first packet switching network was realised in the 1960s – it was called ARPANET –, the Internet reached its current form around 1980. The DoD standard Internet Protocol specification was published in January 1980[1], obseleted by the Internet Protocol specification published in September 1981[2].

The idea of the World Wide Web appeared much later, in March 1989[3]. That is, 2009 is not only the 30th anniversary of the HP-41, but it is also the 20th anniversary of the WWW which is 10 years younger. I thought it would be a nice idea to celebrate these anniversaries by programming a web server that would run on a HP-41. There it is!

1.3 How *NutIP* works

1.3.1 The big picture

NutIP is provided as a ROM. Thus, you need some specific hardware to load a ROM (e.g. MLDL, Clonix, NoVRAM). You also need the HPIL ROM module and a HPIL/RS232 peripheral (HP-82164A). The poor man's solution is to use an HP-41 emulator that could simulate the HPIL serial peripheral (see paragraph 1.3.3).

```
,-----------.         ,----------.         ,----------.
|           |----<----|          |         |      |   |         (
| nutIP ROM |         |          |         |      |   |         (
|           |         |          |----<----| SLIP |   |---<-----(
| HP-IL ROM |         |          |---->----|driver|   |--->-----(
|   module  |         |          |         |      |   |         (
|           |---->----|          |         |      |   |         (
`-----------'         `----------'         `----------'

    HP-41     (HP-IL)  HP-82164A  (RS-232C)  Computer         Internet
```

This sketch only gives a picture of the hardware level. Every byte that goes through the wires are part of a TCP/IP packet. TCP/IP data stream between HP-41 and RS-232C use the SLIP protocol; another low-level protocol (depending on your configuration) is used between the Computer and the Internet.

Between:	and:	Protocol stack
HP-41	HP-82164A	TCP over IP over SLIP over HP-IL
HP-82164A	Computer	TCP over IP over SLIP over RS-232C
Computer	Internet	TCP over IP over *foo*

[1] http://tools.ietf.org/html/rfc760
[2] http://tools.ietf.org/html/rfc791
[3] http://info.cern.ch/www20/

In the table above, *foo* stands for any protocol(s). What this or these protocols are depends on your configuration. This could be WiFi, Ethernet, PPP, ATM, or whatever.

The point is, everything, included the HP-41, use or transport the TCP and IP protocols.

1.3.2 Without the HP-IL/RS-232 interface

Actually, the HP-82164A peripheral, or any equivalent, is not needed if you have a HP-IL ISA card in your computer. The functionnal equivalent of the serial interface is in the HP-IL card's driver, or so.

```
,-----------.         ,----------.
|           |----<----|      |   |         (
| nutIP ROM |         |      |   |         (
|           |         | SLIP |   |---<-----(
| HP-IL ROM |         |driver|   |--->-----(
|   module  |         |      |   |         (
|           |---->----|      |   |         (
`-----------'         `----------'


    HP-41     (HP-IL)   Computer          Internet
```

Using *NutIP* with such a configuration is not documented.

1.3.3 *NutIP* can also run on emulators

If you want to give *NutIP* a try but you haven't got a HP-41 (with the additionnal HP-IL ROM module) or a HP-82164A peripheral, you can still use a software that could emulate both, e.g Nonpareil or Emu41.

On decent systems

Under most UNIX systems, the Emulator program can output data directly to the SLIP driver, using the pseudo-terminal interface (see the figure below). The Emulator has to write the emulated HP-82164A's RS-232C output to a device file like `/dev/ttya0`; the SLIP driver has to read data from `/dev/ptya0`.

```
,-----------.       ,---------.         ,----------.
|           |---<---|         |         |      |   |         (
| nutIP ROM |       |         |         |      |   |         (
|           |       |         |----<----| SLIP |   |---<-----(
| HP-IL ROM |       |         |---->----|driver|   |--->-----(
|   module  |       |      ttya0       ptya0   |   |         (
|           |--->---|         |         |      |   |         (
`___________'       `_________'         `__________'


    HP-41    (HP-IL) HP-82164A                              Internet
\______________________________/
       Running on Emulator
\___________________________________________________/
                 Running on Computer
```

On less decent systems

If your system doesn't implement a feature like the pseudo-terminal interface, you may use another computer (see the figure below). Computer 1 is running the emulator and writes HP-IL output to the serial device (e.g. COM1, using DOS notations). Computer 2 is running the SLIP driver on its serial device.

```
,-----------.       ,---------.         ,----------.
|           |---<---|         |         |      |   |         (
| nutIP ROM |       |         |         |      |   |         (
|           |       |         |----<----| SLIP |   |---<-----(
| HP-IL ROM |       |         |---->----|driver|   |--->-----(
|   module  |       |       COM1       COM1    |   |         (
|           |--->---|         |         |      |   |         (
`___________'       `_________'         `__________'


    HP-41    (HP-IL) HP-82164A  (RS-232)                    Internet
\______________________________/
       Running on Emulator
\______________________________/       \____________/
      Running on Computer 1         Running on Computer 2
```

Section 2

Installation

2.1 Installing the ROM

NutIP must be used at a fixed page number. The default ROM files run at page 8. (If you need to install it at another page, you could use the source and just reassemble the whole ROM.)

NutIP requires the PIL ROM module.

If any of the two above conditions are not fulfilled, running any command from the *NutIP* ROM will raise the Non-Existent error.

2.2 Installing *NutIP* on emulators

If you did not intend to use *NutIP* on emulators, you may want to skip this paragraph.

Operation with Nonpareil and Emu41 are described below.

2.2.1 Nonpareil

Nonpareil[1] is a high-fidelity simulator for calculators, originally written for UNIX systems. It supports the HP-41 series. The "official" 0.79 release by Eric Smith won't run the PIL ROM module, unless you install a patch. Installation instructions and documentation of the patch are available at `http://pagesperso-orange.fr/kdntl/hp41/`

[1] Nonpareil is free software (GPL version 2). Documentation and download at `http://nonpareil.brouhaha.com/`.

nonpareil-patch-doc.html.

Write in your $HOME/.nonpareil/41CV.nal:

```
MOD hpil.mod            # loads the HP-IL ROM module
HPIL hpil.conf          # loads the configuration file
MOD nutip1a.mod         # loads the nutIP ROM

MEMLOST 44              # nutIP needs at least 44 user registers
```

Write in your $HOME/.nonpareil/hpil.conf:

```
serial tty=/dev/ttya0
```

Then, start nonpareil with the following command at your shell prompt:

```
$ nonpareil 41cv
```

2.2.2 Emu41

Emu41[2] is an emulator of the HP-41C and HP-IL system. Emu41 is written for DOS systems; you can run it on Linux systems thanks to programs such as Dosemu (see below).

Write in your EMU41.INI:

```
[MODULES]
0  12 ROM41CX.BIN
7   4 HPIL1H.BIN
8   4 NUTIP1A.BIN

[DEVICES]
SERIAL1
```

Now, at your DOS prompt, go to your Emu41 installation directory and run Emu41:

```
D:\> CD EMU41
D:\EMU41> EMU41.EXE
```

Please refer to Emu41's documentation.

In the emulated HP-41, type the following command:

XEQ ALPHA SIZE ALPHA 044

[2] Emu41 is a freeware. Documentation and download at http://www.jeffcalc.hp41.eu/emu41/.

Running Emu41 on Dosemu

If you need Dosemu to run Emu41 (e.g you are using Linux), you may want to write the following line in your dosemu's configuration file (e.g `$HOME/.dosemurc` on UNIX systems):

```
$_com1 = "/dev/ttya0"
```

Please refer to Dosemu's documentation[3].

2.3 Configuring SLIP

You need to set a SLIP stream on a computer and bind an IP interface to it. *NutIP* uses the basic SLIP protocol, not CSLIP (compressed SLIP).

The SLIP stream is itself bound to either a real serial line or a pseudo-serial line. If relevant, the real serial line must be configured to have the same settings as those of the HP-82164A (which default settings are 9600 bauds, 8 data bits, no parity, one stop bit[4]).

2.3.1 On UNIX systems

Run the following commands, as root:

```
# slattach -p slip /dev/ptya0 &
# ifconfig sl0 192.168.0.1 pointopoint 192.168.0.41
```

The slattach command binds the `/dev/ptya0` serial device to a network interface called `sl0` (default name for a SLIP interface under Linux, may differ on other UNIX systems). The ifconfig command sets the local IP address of this new interface to 192.168.0.1 and set the remote (HP-41) IP address to 192.168.0.41.

Of course, replace `/dev/ptya0` by your serial or pseudo-serial line device file (e.g. `/dev/ttyS*`, `/dev/cua*` or any other `/dev/pty*`). You could also choose different IP addresses.

Note that the slattach command as shown above is the one found on Linux systems. Syntax may differ significantly on other UNIX.

[3] See `/usr/local/share/doc/dosemu/` (or wherever the Dosemu's documentation is installed on your system). The man pages dosemu(1) and dosemu.bin(1) may be useful too.

[4] See *HP-IL/RS-232-C Interface – Owner's Manual*, Hewlett-Packard, 1983.

You may also want to route IP packets between your new sl0 SLIP interface and other network interfaces. For example, on Linux systems, the following commands may be useful:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# iptables -A FORWARD -i sl0 -o eth0 -j ACCEPT
# iptables -A FORWARD -i eth0 -o sl0 -j ACCEPT
```

Please refer to the adequate documentation for further details[5].

2.3.2 On other systems

Please refer to your system's documentation.

[5] On Linux systems, the relevant man pages are slattach(8), ifconfig(8), iptables(8), pty(7).

Section 3

Operation

3.1 Selecting the right serial peripheral

It is advised to have the HP-82164A as the only peripheral on the HP-IL loop. If so, there is nothing to do, the HP-82164A is automatically selected.

If there are many devices on the loop, you have to select which one the HP-41 should use. Say the HP-82164A is the fifth device on the loop. Then, you may type the following commands on your HP-41[1]:

5 XEQ ALPHA SELECT ALPHA

Note You can find the device's ID with the FINDID command (from the PIL ROM module). The following program first searches for the serial peripheral and then selects it; if the HP82164 is not found, the SELECT command will raise the “ADR ERR” error message.

```
LBL "FOO"
"HP82164"
FINDID
SELECT
RTN
```

[1] See *HP-IL Module, Owner's Manual*, Hewlett-Packard, 1981.

3.2 *NutIP* needs some free registers

NutIP uses a 44 register-long buffer located in the main memory. The 44 last user registers are used. For example, say you have 60 user registers in your main memory. This could be achieved by the command XEQ ALPHA SIZE ALPHA 060. Then, the registers 16, 17, . . . , 59 will be used by TCP/IP operation (60 minus 44 equals 16). If the user registers area is smaller than the 44 required buffers, TPC/IP operations will fail with the NO ROOM error.

NutIP also needs two more free registers to store its IP address and the DNS server used for name resolution. These two addresses are stored in a hidden place called I/O buffer, so that they cannot be erased accidentally by the user. The I/O buffer is automatically created by the IPCFG and DNSCFG commands (see the next paragraphs).

3.3 Setting the IP address (IPCFG)

If no IP address was previously specified, all *NutIP* commands will assume 0.0.0.0. As this IP address is a forbidden one, you won't get *NutIP* applications working properly.

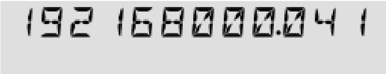
IPCFG can be run either in program mode or in interactive mode. In both cases, the IP address must be entered in decimal form, with no point, each part of the address being three chars long, left padded with zeros if necessary. For example, 192.168.0.41 must be entered as the following sequence: 192168000041.

Interactive mode

When running in interactive mode, IPCFG will prompt you for the IP address. The cursor is shown as the decimal point. When you press a numeric key, the digit at the left of the cursor is replaced by your entry and the cursor moves to the next right digit. The backarrow key and the decimal point key respectively move the cursor to the left and to the right. Press any other key to exit the prompt; IPCFG will then save its address.

For example, type the following keys to set the IP address to 192.168.0.41:

XEQ ALPHA IPCFG ALPHA
192168000041
R/S


```
192 168000.04 1
```

In running mode

When called from within a program, IPCFG takes its argument from the ALPHA register.

For example, here is a small program setting the IP address to 192.168.0.41:

```
LBL "MYCFG"
"192168000041"
IPCFG
RTN
```

Possible errors

Message	Meaning
DATA ERROR	The IP address is invalid. (One of the four parts of the IP address is greater than 255.)
NONEXISTENT	Either the PIL ROM module is not installed or *NutIP* is installed at the wrong page. Please install the PIL ROM module or install *NutIP* in the right slot.
NO ROOM	There is not enough space left in memory to create a I/O buffer. IPCFG needs two free registers for its buffer.

3.4 Domain name resolution

3.4.1 Configuring DNS (DNSCFG)

To use name resolution, you first need to tell *NutIP* the IP address of the DNS server. This is what the DNSCFG is for. DNSCFG reads the ALPHA register for the IP address and then stores it into its dedicated I/O buffer. DNSCFG works exactly like IPCFG does.

DNSCFG can be run either in program mode or in interactive mode. In both cases, the IP address must be entered in decimal form, with no point, each part of the address being three chars long, left padded with zeros if necessary. For example, 10.9.0.1 must be entered as the following sequence: 010009000001.

Interactive mode

When running in interactive mode, IPCFG will prompt you for the IP address. The cursor is shown as the decimal point. When you press a numeric key, the digit at the left of the cursor is replaced by your entry and the cursor moves to the next right digit. The backarrow key and the decimal point key respectively move the cursor to the left and to the right. Press any other key to exit the prompt; IPCFG will then save its address.

In running mode

When called from within a program, IPCFG takes its argument from the ALPHA register.

Here is a small program setting the IP address to 192.168.0.41 and the DNS server IP address to 10.9.0.1:

```
LBL "MYCFG"
"192168000041"
IPCFG
"010009000001"
DNSCFG
RTN
```

Possible errors

Message	Meaning
DATA ERROR	The IP address is invalid. (One of the four parts of the IP address is greater than 255.)
NONEXISTENT	Either the PIL ROM module is not installed or *NutIP* is installed at the wrong page. Please install the PIL ROM module or install *NutIP* in the right slot.
NO ROOM	There is not enough space left in memory to create a I/O buffer. IPCFG needs two free registers for its buffer.

3.4.2 DNS lookup (FINDIP)

Given a host name, the FINDIP command can find its IP address by asking it to the DNS server. The DNS server must be configured by the DNSCFG command (see previous paragraph).

Because of the limited strings handling capability of the HP-41, the FINDIP command only work in programs. The host name is coded as a sequence of strings, two consecutive strings being implicitly separated by a period. For example, the name bar.example.org is coded as the following name sequence: `"bar" "example" "org"`.

The host name must be entered after the FINDIP instruction. Either an instruction other than a string or a null string terminates the name sequence.

The result is then written in the ALPHA register.

```
LBL "FINDBAR"
"010009000001"
DNSCFG
FINDIP
"BAR"
"EXAMPLE"
"ORG"
AVIEW
RTN
```

The program above sets 10.9.0.1 as the DNS server. It then asks the IP address of bar.example.org and displays it.

Notes

- Host names are case insensitive.
- Each part of the name sequence cannot be more than 15 chars long.
- Execution of the program continues after the last string of the name sequence.

Possible errors

Message	Meaning
ADR ERR	The DNS server answered with an error code. This probably means the specified name does not resolv to an IP address.
NONEXISTENT	Either the PIL ROM module is not installed or *NutIP* is installed at the wrong page. Please install the PIL ROM module or install *NutIP* in the right slot.
NO ROOM	The specified name sequence is too long. Only names shorter than 109 characters are allowed.
TRANSMIT ERR	No response from the DNS server. The issue is either at HPIL level or at a higher level (TCP/IP or DNS).

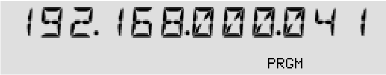
3.5 The web server (WWWD)

NutIP provides a small webserver called WWWD (stands for World Wide Web Daemon). When requested by a distant user, e.g. via a web navigator software, WWWD can serve the program listing of any user program installed in the HP-41. WWWD can serve an index page with a link to all installed user programs as well.

Starting WWWD

Just run the WWWD command:

XEQ ALPHA WWWD ALPHA



```
192.168.000.04 1
          PRGM
```

The LCD shows the PRGM annunciator and the IP address of the web server. The TCP port is 80 and cannot be configured.

You can access the web server from any computer on the network. If you assigned a public IP address to *NutIP*, you can even access your HP-41 from any computer connected to the Internet.

In the following, your HP-41 is assumed to be configured to the IP address 192.168.0.41.

WWWD's display

When operating, WWWD displays various informations on the LCD screen. The IP address of *NutIP* is shown, as well as the following annunciators:

Annunciator	Meaning
PRGM	WWWD is running (always on).
0	WWWD is processing data.
1	WWWD is building a TCP retransmission packet and sending it.
3	A TCP packet is being received and processed.
4	A ICMP packet is being received and processed.

Exiting WWWD

Just press the ← key while WWWD is running. If WWWD doesn't exit when the key is released, it may be busy; press the key a little longer. In any case, pressing it for three seconds must suffice.

WWWD answers to pings

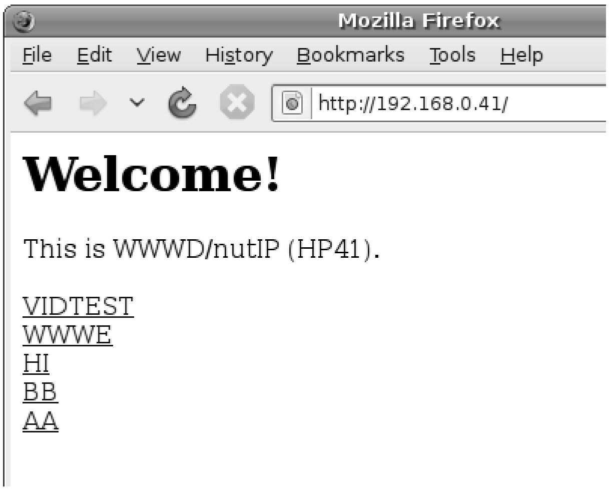
You can first try to ping your HP-41. Open a command line prompt and type the following command (just type the first line without the dollar sign):

```
$ ping -i 4 192.168.0.41
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=64 time=1232 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=64 time=1224 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=64 time=1228 ms
64 bytes from 192.168.0.41: icmp_seq=4 ttl=64 time=1222 ms
^C
--- 192.168.0.41 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 12011ms
rtt min/avg/max/mdev = 1222.768/1227.206/1232.525/3.888 ms
$
```

The `-i 4` option tells the program to send a ping every 4 seconds.

WWWD answers to HTTP requests

To use the web server, just type the following URL in your web navigator: `http://192.168.0.41/`. You should get a welcome page with the index of all user programs installed on your HP-41.

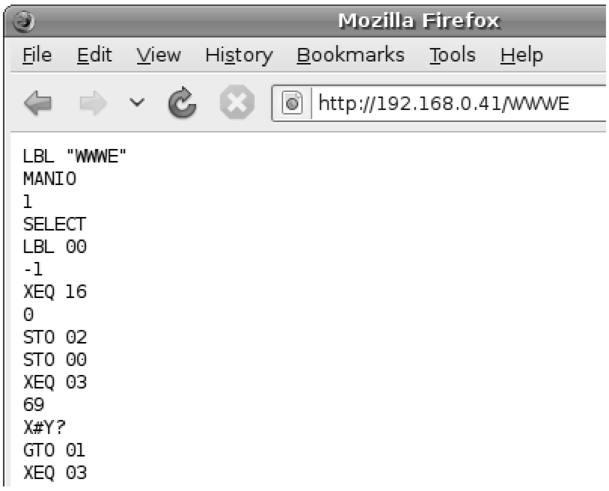


Welcome!

This is WWWD/nutIP (HP41).

VIDTEST
WWWE
HI
BB
AA

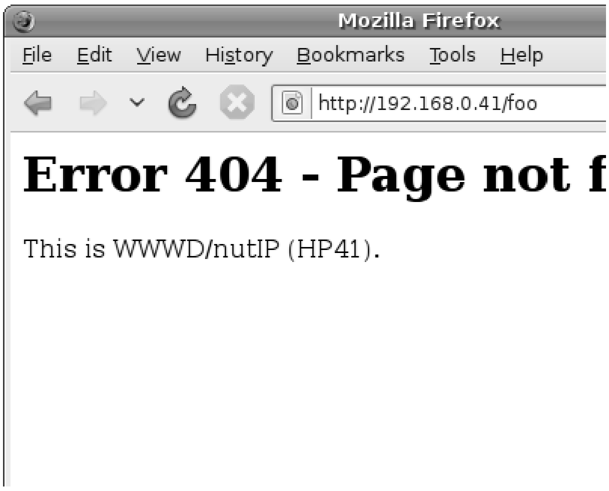
You can request a program to be displayed by appending the program's name to the main URL. For example, say you have the program WWWE in your HP-41's memory. Type the following URL in your web navigator: `http://192.168.0.41/WWWE`. Note that case must be preserved.
Another way, simplier, is to click on the program's link in the welcome page displayed by your web navigator.



http://192.168.0.41/WWWE

```
LBL "WWWE"
MANIO
1
SELECT
LBL 00
-1
XEQ 16
0
STO 02
STO 00
XEQ 03
69
X#Y?
GTO 01
XEQ 03
```

If WWWD cannot find the requested program, it answers the well-known 404 error.



http://192.168.0.41/foo

Error 404 - Page not f

This is WWWD/nutIP (HP41).

If WWWD got a bad HTTP request, it answers the 400 error.

Possible errors

Message	Meaning
NONEXISTENT	Either the PIL ROM module is not installed or *NutIP* is installed at the wrong page. Please install the PIL ROM module or install *NutIP* in the right slot.
NO ROOM	There is not enough space left in memory to create a buffer. Please free some memory.
TRANSMIT ERR	A HPIL command timed out. The serial peripheral may be misconfigured or shut down.

Appendix A

License

Appendix B

Assembling the ROM from the source

B.1 Toolchain and required build tools

Basically, you need a standard POSIX environnement (Linux, *BSD, Mac OSX, Solaris, etc.) with dosemu.

You need:

- A41.EXE, the MCODE assembler from the SDK41 package. This is an assembler written by Warren Furlow, under the GPL license. A41 was developped for DOS, so it won't run natively on a POSIX environnement. Fortunately, one solution, among others, is:
- Dosemu, a DOS emulator. It will enable us to execute A41.EXE.
- An ANSI C compiler is needed to compile L41-posix which is provided in the `tools/l41-posix/` subdirectory. L41-posix is a port of L41.EXE from the SDK41 package. L41.EXE also runs under dosemu, but it has a bug that prevented nutIP to be linked properly. A C compiler is also needed to compile a small utility needed to convert .rom files to .mod files.
- M4, the standard UNIX macro processor. Needed because I use sugar syntax in the source files.
- make, /bin/sh, grep, sed, awk, and probably some others.

I tested the assembling and linking process under x86 architecture with:

- A41.EXE says it is version 3.01 (`http://www.hp41.org/`),
- dosemu 1.4.0.0 (`http://dosemu.sourceforge.net/`),

- GNU gcc 4.3.2,
- GNU m4 1.4.11,
- GNU make 3.81, GNU grep 2.5.3, GNU sed 4.1.5, GNU awk 3.1.6.

Any version should do the job because I tried not to use non-POSIX features of the GNU tools. However, m4 may be critical and lots of things may be different if you are not using a different version. I didn't use any M4 advanced features though.

B.2 Building *NutIP*

Changing the page

Change the `PAGE` variable in the `tools/l41pp` script. For example, the following will for relocate the ROM to page A (hexadecimal, decimal is 10).

```
#!/bin/sh

PAGE=A

sed 's/{NUTIP_PAGE}/'$PAGE'/g'
```

Assembling

If your system fits all the previously described requirements, building *NutIP* is as simple as typing the command:

```
$ make
```

(Don't type the `$` character and hit the Return key...)

This should create many files in the build/ directory. You may be interested by either `build/nutip1a.rom` or `build/nutip1a.mod`. The first file is for use with Emu41, whereas the second one could be used with Nonpareil.

Appendix C

Development details

C.1 Some editor tools

If you are using the vim editor, you could take a look at the `tools/mcode.vim` file. This adds syntax highlighting for MCODE sources.

Typing `make tags` in `src/` creates the `tags` file, which could be used by vim or emacs to access subroutines. See the ctags(1) man page.