Network programs in BASIC
(Kipper BASIC/BASIC on Bails)

ShadowM
ECCC
September 18, 2010
some background

✓ ip65 is a TCP/IP stack for the Commodore 64, created by Per Olofsson ("Magervalp") and currently maintained by Jonno Downes

✓ Jonno wrote netboot65, which provides cartridge images that include tftp, gopher and telnet clients, as well as network booting

✓ geolink (IRC client for GEOS) also based on ip65
why aren't there more programs?

✔ there are several networking libraries available for RR-Net compatible network cards, but...

✔ network coding in ML is notoriously tedious and error prone; the number of people willing and able to spend the time on this is small

✔ there have not been high level language bindings for C= network stacks...
but now...

✔ Jonno has just created two different runtimes which provide BASIC language bindings for the ip65 network stack

✔ the runtimes work like any other BASIC extension, and provide additional keywords for writing networked programs

✔ extremely easy to use, good for prototyping as well as complete applications
why two runtimes?

✔ memory constraints - not all functionality would fit in a single runtime

✔ Kipper BASIC is for general networking (socket programming, TFTP, netcat, &c.)

✔ BASIC on Bails is for writing web apps ('64 acts as a server)

✔ both have the ability to autoload a program after initialization (see STARTUP in docs)
Keywords used by both runtimes

✔ MAC (set low bytes of MAC address)

✔ MYIP, NETMASK, GATEWAY, DNS
  (for manually setting network parameters)

✔ DHCP (auto configure network settings)

✔ IPCFG (show network settings)

✔ PING
Kipper BASIC keywords

✔ TCPCONNECT (open TCP/IP connection)
✔ TCPLISTEN (open server socket and block)
✔ POLL (poll network for received packets)
✔ TCPSEND (send data over network)
✔ TCPBLAT (send file over the network)
✔ TCPCLOSE (close connection)
more Kipper BASIC keywords

✓ TFTP (set TFTP server for TFPUT, TFGET)
✓ TFGET/TFPUT (send/receive file via TFTP)
✓ NETCAT (echo keyboard input to network)
Kipper BASIC reserved variables

✔ IN$ - holds received data from polling (up to 255 bytes)

✔ CO% - 1 if connection is open, 0 otherwise (to detect dropped connection)

✔ ER% - last network error number (see documentation for details)
loading Kipper BASIC

**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.
LOAD"KIPPERBAS.PRG",8
SEARCHING FOR KIPPERBAS.PRG
LOADING
READY.
RUN
**** KIPPER BASIC 1.0 ****
READY.
example: remote shell client

REMOTE SHELL
CONFIGURING NETWORK WITH DHCP...
INTERFACE : RR-NET
MAC ADDRESS : 00:80:10:19:53:64
IP ADDRESS : 192.168.1.100
NETMASK : 255.255.255.0
GATEWAY : 192.168.1.99
DNS SERVER : 192.168.1.99
DHCP SERVER : 192.168.1.99
TFTP SERVER : 255.255.255.255
HOSTNAME OR IP? 192.168.1.101
PORT? 1953
CONNECTING TO 192.168.1.101/1953
CONNECTED
}$

DISK KB-20100822
1 START KIPPERBASE PRG
40 KIPPERBASE.PRG PRG
3 GOPHERD PRG
5 RSHD PRG
1 GOPHERMAP.TXT SEQ
3 ADDRESSES.TXT PRG
5 RSH PRG
593 BLOCKS FREE.
sample code (client)

```
10 PRINT "REMOTE SHELL"
12 PRINT "CONFIGURING NETWORK WITH DHCP.."
14 DHCP
16 IPCFG
20 INPUT "HOSTNAME OR IP"; HNS$
22 IF LEN(HNS$) <> 0 THEN 30
24 PRINT "YOU MUST ENTER A HOSTNAME."; GOT020

30 INPUT "PORT"; PNS$
32 PRINT "YOU MUST ENTER A PORT NUMBER."; GOT030
33 NU = 1
34 FOR I = 1 TO LEN(PNS$)
35 C$ = MID$(PNS$, I, 1)
36 IF C$ <> "0" OR C$ = "9" THEN NU = 0
37 NEXT
38 IF NU = 0 THEN PRINT "PORT NUMBER MUST BE NUMERIC."; GOT030
40 PRINT "CONNECTING TO "; HNS$; "/"; PNS$
42 TCPCONNECT HNS$, VAL(PNS$)
```
example: remote shell server

REMOTE SHELL SERVER
CONFIGURING NETWORK WITH DHCP...
INTERFACE : RR-NET
MAC ADDRESS : 00:80:10:19:53:65
IP ADDRESS : 192.168.1.101
NETMASK : 255.255.255.0
GATEWAY : 192.168.1.99
DNS SERVER : 192.168.1.99
DHCP SERVER : 192.168.1.99
TFTP SERVER : 255.255.255.255
LISTENING ON PORT 1953...
CONNECT!
'$
'CO:RSH.BAK=RSH'
'SO:RSH.BAK'
DISCONNECTED!
LISTENING ON PORT 1953...
Sample code (server)

```
10 PRINT"REMOTE SHELL SERVER"
15 CL$=CHR$(13)+CHR$(10)
20 PRINT"CONFIGURING NETWORK WITH DHCP..
30 DHCP
40 IPCFG
50 PRINT"LISTENING ON PORT 1953..."
60 TCPLISTEN 1953
70 PRINT"CONNECT!"
80 POLL
90 IF CO%=0 THEN PRINT"DISCONNECTED!":GO TO 50
95 GETG$:IFG$="""THEN100
96 IFG$="0""THENPRINT"CANCELED!":CLOSE1:TCPCLOSE:END
100 IF LEN(IN$)=0 THEN 80
110 CM$=""
111 FOR I=1TOLEN(IN$)
112 C$=MID$(IN$,I,1)
113 IF ASC(C$)<=13 THEN I=255:GOTO116
114 IF ASC(C$)>=97 AND ASC(C$)<=122 THEN C$=CHR$(ASC(C$)AND 223)
   BREAK
   READY.
```
BASIC on Bails keywords (setup)

✔ HTTPD <port, default_line_no>
  (starts web server with default callback)

✔ HOOK <path, line_no>
  (specify callback for HTTP path, e.g. HOOK "/HELLO",1000)
  can also access path via reserved variable PA$

✔ TYPE <mime_type>
  (change MIME type)

✔ STATUS <http_status>
  (change HTTP status, e.g. "403 forbidden")
BASIC on Bails keywords (data)

✓ ! <output_string>
  (send output to client)

✓ XSEND <filename>
  (send file to client, e.g. “FAVICON.ICO”)

✓ YIELD
  (complete HTTP response and wait for next callback)

Request parameters are translated to BASIC variables, but only the first letter is significant.
loading BASIC on Bails

```
JIFFYDOS V6.01 (C)1989 CMD
C-64 BASIC V2  38911 BASIC BYTES FREE
READY.
@$
0 "KIPXPLORER" 00 2A
38 "BAILS.PRG" PRG
15 "FAVICON.ICO" SEQ
 2 "HELLO.BAS" PRG
608 BLOCKS FREE.
READY.
LOAD"BAILS.PRG",8
SEARCHING FOR BAILS.PRG
LOADING
READY.
RUN
### BASIC ON BAILS ###
READY.
```
web server output

listening on 192.168.1.101:80
connection from 192.168.1.200:33186
path: /
listening on 192.168.1.101:80
connection from 192.168.1.200:33187
path: /select
listening on 192.168.1.101:80
connection from 192.168.1.200:33188
path: /
listening on 192.168.1.101:80
BASIC on Bails, baby!

ECCC/VCFMW 2010 PRESENTATION SCHEDULE

(This server is running BASIC on Bails on a Commodore 64!)

Select presentation to view details:

-----
10:30 XUM1541
11:00 CBM-COMMAND
12:00 ATARI PROJECTS
13:30 KIPPER BASIC
14:00 CRIMSON TWILIGHT
14:30 STEREINSID

Select
source code (startup)

```html
<html>
<body style="background-color:#b0c4de; margin:24px">
<h2>eccc/vcfmw 2010<br>presentation schedule</h2>
<p>(this server is running "basic"+nb$+"on"+nb$+"bails on a commodore 64!")</p>
</body>
</html>
```
source code (form)

```html

<form name=s1 action=select>
  <select name=p size=7 style=font-family:monospace>
    <option>----- altair workshop</option>
    <option>10:30 xum1541</option>
    <option>11:00 cbm-command</option>
    <option>12:00 atari projects</option>
    <option>13:30 kipper basic</option>
    <option>14:00 crimson twilight</option>
    <option>14:30 stereoinsid</option>
  </select>
  <br>
  <input type=submit value='select'>
</form>

yield ready.
```
source code (dispatch)

```html
<body style=background-color:#b0c4de;margin:24px;>

```
Implications

✔ Now anyone can code network programs
✔ Even if you are considering a larger project written in ML, it's good for prototyping
✔ IT'S FUN! When is the last time you had fun writing a program in Commodore BASIC?
resources

✔ not yet released, but can be checked out using Subversion

✔ repo contains a D64 image ready to go (no building necessary)

http://sourceforge.net/projects/netboot65/develop

✔ I've also started a page on my site:

http://www.lyonlabs.org/commodore/kb-bob/index.html