DYNAMIC CONCEPTS, INC.
One Columbia, Suite 100
Aliso Viejo, CA 92656
(714) 448-8600 or (800) 822-2742

UniBasic Port as Device Driver

24 June 1994

The IRIS Port as a Device Driver (PDn), has been implemented as a user call under uniBasic. All five functions, **Open, Close, Read, Write** and **Print** are available. These functions are used to Read or Write to another serial port.

The PDn routine uses a circular buffer to capture data from the port. The default size for this circular buffer is the same as the controlling port's **INPUTSIZE** environment variable. The circular buffer size can be changed explicitely by setting a new environment variable, **PDNBUFSIZ**E, to a value different than INPUTSIZE.

The uniBasic PDn driver works with both interactive and non-interactive ports. However, it is the user's responsibility to assure the port being used is disabled and the port is not in use by a non-PDn process prior to accessing, such as uucp or cu.

CALL $DEVOPEN

The PDn port must be opened prior to any other operations. The syntax for opening a port is:

CALL $DEVOPEN, *chan.num*, "*$PDn*"

chan.num is a pseudo device channel number. This channel number is different from the uniBasic file channel number. The pseudo channel number must be between 0 and 99 inclusive.

$PDn has the same syntax as IRIS. For example, $PD2 will open $PD2 and $PD3 will open $PD3. An equivalent environment variable without the leading dollar sign ($) must define the associated UNIX port. This environment variable should be set in user's .profile file. An example of PDn port configuration follows:

```
PD4=/dev/tty004; export PD4
```

Failure to set the appropriate environment variable results in uniBasic error 70, "Filename does not exist".

The PDn open routine creates a **LCK..ttyxxx** file to indicate a busy port. This lock file is created in /tmp directory. Users can change the default /tmp directory by setting **LOCKDIR** environment variable. The user's process ID is written in the lockfile.

CALL $DEVCLOSE

Unlike uniBasic, the user must explicitly call the close routine. The syntax to close a pseudo channel is:

```
CALL $DEVCLOSE,{chan.num}
```

chan.num is an optional pseudo device channel number used during OPEN. All psuedo channels are closed if channel number is omitted.

The PDn close routine removes the lock file in addition to closing the port. If the CALL $DEVCLOSE is not issued, the lock files will not be removed when uniBasic is terminated or a uniBasic process is killed, including "kill -15 PID".

CALL $DEVREAD

The syntax to read a port is:

```
CALL $DEVREAD, chan.num, rec #, offset #, time-out, str.var
```

chan.num	pseudo device channel number
rec #	record number, must be set to -1
offset #	byte offset number, must be set to -1
time-out	time-out value, must be set to -1
str.var	string variable to store the contents of data coming from *chn.num*

rec #, offset #, and *time-out* must all be set to negative one (-1), as IRIS does not allow these parameters to be defined.

CALL $DEVREAD supports the various modes that may be set using CALL $DEVPRINT, such as : echo on (IOBE), echo off (IOEE), binary input (IOBI), activate on control characters (IOBC), input len and input time out. Only string variables are allowed since numeric variables are not supported for PDn driver under IRIS.

Like IRIS, a Basic error 95 is returned if a read request cannot be satisfied in the allocated time.

Input character translations, similar to uniBasic Input handling, are also done during a READ. However, Cursor tracking and hot-key swapping are not available. A hot-key swap character is disregarded in the event it is read.

CALL $DEVPRINT

The syntax to print to a port is:

CALL $DEVPRINT, *chan.num*, *rec #*, *offset #*, *time-out*, *str.var*

chan.num	pseudo device channel number
rec #	record number, must be set to -1
offset #	byte offset, set to either -1 or 225
time-out	time out, must be set to -1
str.var	any string variable to be sent to specified *chan.num*

rec #, *offset #* and *time-out* are usually set to negative one (-1), as IRIS does not allow these parameters to be defined. An exception to this rule for PDn under unBasic is while printing a terminal mnemonic. The *offset #* should be set to 225 to indicate CRT mnemonics. For example,

```
CALL $DEVPRINT,1,-1,225,-1,'IOCI'
```

CALL $DEVPRINT does not allow PRINT USING or TABS. Only string variables are allowed, although IRIS allowed numeric data to be printed.

CALL $DEVWRITE

The $DEVWRITE call has two separate modes of operation. One for actual writing to a port, and another for setting various parameters on a port.

The syntax to WRITE to a port is:

chan.num	pseudo device channel number
rec #	Must always be set to -1
offset #	Must always be set to -1
time-out	Time-out must always be set to -1
str.var	Any string variable to be sent to specified *chan.num*

Record number, byte offset and time-out must all be set to negative one (-1) while writing a string variable, as IRIS does not allow these parameters to be defined.

A numeric variable cannot be written to a port.

The syntax to change different modes of a port are as follows:

CALL $DEVWRITE, *chn.num*, *rec#*, *mode*, *time-out*, *num.var*

chan.num	Psuedo device channel number

rec # Must be set to 0 when setting various port characteristics
mode Acceptable modes are 0 thru 7 inclusive, as desribed below
time-out Must always be set to -1 when setting various port characteristics
num.var Value to set based on *mode*

The following are the various *modes* acceptable for use with CALL $DEVWRITE:

MODE 0: Set input time-out value in tenths of a second (in this case 1 second). Example:

```
CALL $DEVWRITE,1,0,0,-1,10 ! set input time-out to 1 second
```

MODE 1 : Set a specific input length. Example:

```
CALL $DEVWRITE,1,0,1,-1,2 ! set input length to 2 bytes
```

MODE 2 : IRIS allows polling mode, but is not implemented for uniBasic.

```
CALL $DEVWRITE,1,0,2,-1,xx ! not supported
```

MODE 3 : Set port baud rate. Example:

```
CALL $DEVWRITE,1,0,3,-1,9600 ! set port to 9600 baud
```

MODE 4 : Set port word length. Example

```
CALL $DEVWRITE,1,0,4,-1,8 ! set word length of port to 8
```

MODE 5 : Set no parity, odd parity or even parity. A zero equals no parity, a one equeals odd parity, and a 2 equals even parity. Example:

```
CALL $DEVWRITE,1,0,5,-1,2 ! set parity bit to even
```

MODE 6 : Set to one or two stop bits. A one will set it to one stop bit, anything else will set it to 2 stop bits. Example:

```
CALL $DEVWRITE,1,0,6,-1,1 ! set port to 1 stop bits
```

MODE 7 : Set modem control for a port. A non-zero sets modem control, i.e. clear CLOCAL and set HUPCL. A zero sets CLOCAL and clears HUPCL. Example:

```
CALL $DEVWRITE,1,0,7,-1,1 ! set modem control hupcl
```