APPENDIX Q

I/O DISTRIBUTION

Use of multiple input and output devices can be accomodated without the need for specialized PEEKs and POKEs, by using the I/O distribution system which is available under the DOS and BASIC. The following is an illustrative example using the ACIA.

The simplest way to send data to the ACIA is to inform the Disk Operating System (DOS) that the ACIA is to be an output port. The command, responding to the DOS prompt

```
A*
```

is

```
IO ,Ø1
```

This assigns the ACIA as the sole system output port.
The general form of I/O distribution is

IO ,nn to assign input devices only

IO ,mm to assign output devices only

IO ,nn,mm to assign both input and output devices

A blank must be used in the command, as illustrated. Note that these numbers, nn, mm, are in hexadecimal (base 16). Each device number assignment must be a two digit number selected from the following list:

HEX NN INPUT DEVICE CODE	HEX MM OUTPUT DEVICE CODE
ØØ Null	ØØ Null
Ø1 Serial Port (ACIA at FCØØ)	Ø1 Serial Port (ACIA at FCØØ)
Ø2 Keyboard on 54Ø Board	Ø2 Video on 54Ø Board
Ø4 UART on 55Ø Board	Ø4 UART on 55Ø Board
Ø8 Null	Ø8 Line Printer
1Ø Memory	1Ø Memory
2Ø Disk Buffer 1	2Ø Disk Buffer 1
4Ø Disk Buffer 2	4Ø Disk Buffer 2
8Ø 55Ø Board Serial Port	8Ø 55Ø Board Serial Port

Each of the device codes listed is a hexadecimal value corresponding to one bit or device. For example, the ACIA (device Ø1) is given by bits

ØØØØ ØØØ1

and the video board (CRT terminal) is device Ø2, given by bits

ØØØØ ØØ1Ø

Both devices can be used simultaneously by specifying the device with a bit pattern

ØØØØ ØØ11

which is hexadecimal Ø3. Therefore

```
IO ,Ø3
```

Will send data to the CRT terminal and the device on the ACIA port, simultaneously. Multiple output devices may be used (in contrast to only single input devices).

OTHER DEVICES

For other devices, it is probably easier to accept the device handlers built into the BASIC programs. Under BASIC, the devices are numbered sequentially, 1 to 9. This renumbering is distinct from the previous I/O command example. Under BASIC, the devices which are available are

DEVICE NUMBER	INPUT DEVICES	DEVICE NUMBER	OUTPUT DEVICES
1	Serial Port (ACIA)	1	Serial Port (ACIA)
2	Keyboard on 54Ø Board	2	Video on 54Ø Board
3	UART on 55Ø Board	3	UART on 55Ø Board
4	Null	4	Line Printer
5	Memory	5	Memory
6	Disk Buffer 1	6	Disk Buffer 1
7	Disk Buffer 2	7	Disk Buffer 2
8	55Ø Board Serial Port	8	55Ø Board Serial Port
9	Null	9	Null

The DOS I/O command previously discussed remains in effect until it is reset or an error occurs. If an error occurs, the default value is set (start up value). In contrast, the device numbers above can be assigned for each input/output operation as needed. For devices other than those set up by the DOS I/O command, the device assignments immediately above could be used.

For example, to read from the keyboard and write on the printer attached to the ACIA, the following instructions could be used:

```
1Ø INPUT #2,A$ : REM KEYBOARD INPUT
2Ø PRINT #1,A$ : REM TO PRINTER ON ACIA
3Ø LIST #1     : REM AND LIST PROGRAM, TOO
RUN
```

Yielding the input prompt

```
?
```

After typing a message (72 characters or less) and a <RETURN>, the message and the program will be printed on the serial printer.

DISK USE

As an input/output device, disks can be used in a similar manner.

However, prior to using the disk, the user should provide for protecting his buffer areas by running the CHANGE program as

```
RUN"CHANGE"
```

Respond to the terminal width change with

```
NO <RETURN>
```

and respond to a request to change the BASIC and ASSEMBLER use of memory by

```
NO <RETURN>
```

but respond to the work space limit change by

YES <RETURN>

The CHANGE program will ask "how many 8-page buffers before the work space." (Remember each page contains 256 characters.) There are only two valid responses here (1 and 2)

1. if only one file is to be used
2. two files must be open simultaneously

For the example that follows, 1 is sufficient. No additional room is required, so respond

NO <RETURN>

to that question. It is also not necessary to request any room at the top for this example.

The small differences between a disk and other devices are the need to open a disk file by name as

```
DISK OPEN,6, "FILE1"
```

and to close the file when finished by

```
DISK CLOSE,6
```

These last two statements can be used to store a string received from the modem. The input from the modem would be

```
INPUT #1,A$
```

where the string A$ must have as its last character <RETURN>.

Combining these three statements into a program to write a single message on disk

```
10 DISK OPEN,6, "FILE1"   :REM OPENS DISK (W/ONE BUFFER)
20 INPUT #1,A$            :REM LISTENS TO MODEM
30 PRINT #6,A$            :REM ECHOS TO DISK
40 DISK CLOSE,6           :REM CLOSES DISK FILE
50 END
```

Likewise, we could later recover the data by the program

```
10 DISK OPEN,6, "FILE1"
20 INPUT #6,A$
30 PRINT #2,A$
40 DISK CLOSE,6
50 END
```

In this problem, writing and reading was done sequentially. Modifying the program to accept multiple messages requires that they be stored sequentially.

It is possible to inspect the sequential disk file by

```
RUN"SEQLST"
```

which provides a listing of the file when the information requested is given. The computer responds

```
SEQUENTIAL FILE LISTER
TYPE A CONTROL C TO STOP
FILE NAME?
```

Respond with the file name of a sequential file

FILE1

and a listing of the file will be printed. Upon reaching the end of the disk file, the message

```
ERR #D ERROR IN 1ØØ
```

will indicate completion of the listing.

Caution: be aware that when using the SEQLST utility to inspect files which have BASIC pro them, the display will look different than the original text. The reason for this is that the BASIC BASIC source programs in a shorthand, called a tokenized form.

Another popular way is to transfer the disk file (say it was stored on track 39) by the CALL st

```
DISK!"CALL D3ØØ=39,1"
```

which writes the file contents onto the middle of the CRT screen. Note that some apparent garba tionally printed here due to the unused portion of the disk file's being printed, too.

To handle data in a random order, for example extracting the 20th data item from a file, it is not n the 19 prior data items. The use of random data items, also called records, is particularly useful wh large set of data. Such data might be a set of customer accounts, a checking account history, or ev records for given days. In all these cases, the need arises to extract a specific record, without lookin records.

To aid in understanding the handling of random records, visualize a pointer which marks the start GET command moves this pointer at the start of a given record. For example,

```
DISK GET,Ø
```

places the pointer in front of the first record. Similarly,

```
DISK GET,5
```

places the pointer in front of the sixth record. This method makes it easy to locate a record on the d is wasteful of disk storage capability.

Each record uses a large disk area (128 bytes). The value of 128 bytes is preset by the operatin

A random (not sequential) input record may be terminated by the PUT command. This will close ord from further input.

A simple program to write three records on disk file "SCRTCH" and then GET the second recor would be

```
1Ø REM PROGRAM WRITE TEST
2Ø REM OPEN THE DISK FILE SCRTCH
3Ø DISK OPEN, 6, "SCRTCH"
4Ø REM LOOP THREE TIMES TO END OF LOOP
5Ø FOR TIME=1 TO 3
6Ø REM PLACE 128 BYTE RECORDS ON DISK BY
7Ø REM (A) POSITION POINTER WITH A GET COMMAND
8Ø REM (B) PASSING THE MESSAGE TO THE DISK BY PRINT COMMAND
9Ø REM (C) CAUSE THE RECORD TO BE WRITTEN BY PUT COMMAND
1ØØ DISK GET, TIME-1
11Ø INPUT #2,A$ : REM TYPE IN ANY PHRASE FROM KEYBOARD
12Ø PRINT #6,A$ : REM PLACE IN MESSAGE BUFFER
13Ø DISK PUT    : REM TRANSFER MESSAGE BUFFER TO DISK
14Ø NEXT TIME
15Ø REM END OF LOOP
```

```
160 RCRD=2
170 DISK GET,RCRD-1 : REM POINTER AT START OF RECORD 2
180 INPUT #6,A$     : REM READ DISK's SECOND RECORD
190 PRINT #2,A$     : REM AND OUTPUT TO CRT (TERMINAL)
200 DISK CLOSE,6
210 END
```

The use of sequential and random disk files permits simpler control and bookkeeping than the C
or LOAD and PUT commands which were used for earlier file handling. This is one difference betw
dling as compared to file handling.

MORE DEVICES

Memory can also be treated as a device. When accepting data from memory (Random Access M
as the input device, the DOS uses the address found in locations 238A (low address half) hexade
(high address half) hexadecimal to determine what memory region to use. After each input, the
mented by one location. Memory, as an output device, is specified by the contents of 2391 (low ad
adecimal and 2392 (high address half) hexadecimal.

To load the address of memory to be used as an input device into 238A and 238B, and also
memory to be used as an output device into 2391 and 2392, DOS provides the command

```
MEM mmmm,nnnn
```

mmmm is the address of memory to be regarded as an input device (its starting address) and nnnn
memory to be regarded as an output device (its starting address). For example,

```
MEM 5000,5500
```

would load the locations

	LOCATION		
	DEC	**HEX**	**CONTENTS**
Input	9098	238A	00
Address	9099	238B	50
Output	9105	2391	00
Address	9105	2392	55

which establishes memory locations 5000 and up to be used as an input device and locations 5500 a
as an output device. No end of these memory regions is specified, so the user is cautioned in th

Finally, a device called the null device is provided. The null device permits writing programs w
worry about the physical device characteristics. For example, a program could be tested which woul
on the printer; by assigning the null device, no paper would be wasted while the program is chec

INDEX

A

AIS Board 77
AC 68
AC1 68
AC2 68
AC12P 60
AC17P 60
AC21 54
Accessory Interface 61
ACIA 33
ACTL 51
Action Button 43
Advanced
  Features 40
  Topics 67
Alarm Operations 54
Analog
  I/O 61
  To Digital Conversion 62
Appliance Control 50
ASCII
  Code 25
  Example 99
  Table 106
ASM (Assembler) 38, 133
ASM Extensions 121

B

Back Arrow 119, 123
Backpanel Connectors 3
Barber Pole 76
BASIC Commands 38, 81
BASIC Conversions 86
BASIC Errors
  BASIC-in-ROM (Cassette) 85
  BASIC (Disk) 84
BASIC Extensions 65, 120
BASIC Programming 12
Baud Rate 64
Bit Switching and Sensing 60
BEXEC 11, 71, 141
Block 78
Break 5, 6, 41
Breakpoint 121
BUS, 16 Pin I/O 59, 60, 62
Bytes Free 95

C

CA-15 59
CA-20 59
CA-21 60
CA-22 61
CA-23 60
CA-24 61
CA-25 61
Calculator Mode 13
Cassette
  BASIC-in-ROM 14
  Cold Start 5
  Control Shift 8, 35
  Data Recovery 35
  Data Storage 35
  Load 9, 33
  Save 34
Change Utility 10, 94
Character Graphics 109
Character Manipulation 15
Character String 24, 82
Clock, Time of Day 68
Code, Machine Language 31
Cold Start 20
Color Graphics 27
Color, Inverted 29
Color's 10
Color Tuning 76
Combining Programs 139
Computer Setup
  Cassette 4
  Disk 4
Computer Interface to 16 Pin I/O Bus 59
Conditional Statement 15, 24

Connections
Closed Circuit Video ... 2
Pin ... 77
Video ... 2
Control C ... 41, 122
Control E ... 122
Control O ... 119
Control Q ... 119
Control R ... 122
Control S ... 119
Control Shift ... 8
Control T ... 122
Control U ... 122, 123
Control Y ... 122
Conversions
Analog to Digital ... 62
Digital to Analog ... 31, 62
Hexadecimal to Decimal Chart ... 104
Languages ... 38
Copier ... 124
Copy ... 95
Countdown Timer ... 69
Create ... 97

D

D- ... 41
Data Register (PIA) ... 54, 59, 60, 71
Default ... 24
Delete (files) ... 93
Device Numbers ... 120
Devices ... 144, 147
Digital to Analog (D/A) Converter ... 30, 31, 62
Dictionary ... 119
Dimension (DIM) ... 24, 81
Direct Video (modifications) ... 2
Directory
DIR ... 8, 10, 11, 97
Disk Directory Listings ... 11, 97
Format ... 11, 97, 123
DIR SRT (Sorted directory) ... 11, 97
Disk
Allocations ... 79, 80, 120, 123
BASIC Commands ... 6, 81-84
Care of ... 1
Copier ... 124
Copy ... 95
Extensions ... 65, 121
I/O (OPEN, GET, PUT, CLOSE) ... 146
Organization ... 38
Programs ... 7, 36, 37
Read/Write ... 36
Systems ... 10
Track Formatting ... 80, 122
Track Ø Read/Write ... 94
Utilities ... 94
DOS (disk operating system)
Commands ... 38
Errors ... 85

E

EM Command ... 38
Entries Free ... 97
Error ... 60
BASIC-in-ROM ... 85
Disk BASIC ... 84
DOS ... 85
Numbers ... 118
Extended Monitor ... 38, 121
Extensions to Disk BASIC ... 65, 119
Extensions to Assembler ... 121
External Switches ... 54

F

File ... 67
Access ... 10, 33, 139, 142
Create ... 97
Delete ... 93
Indirect ... 139
Source Formatting ... 123
Flag Bit Setting, I/O ... 119
Flow Chart Explanation ... 17
Frequency (piano keyboard) ... 92

G

Generator, Tone ... 30
GET/PUT Overlays ... 80
Graphics ... 108
Greenhouse Example ... 71

H

Head End Carps ... 59, 60
Heterodyning ... 76
Hexadecimal to Decimal
Tables ... 104
Tutor and Conversions ... 99
Home Security ... 52

I

Indirect Files 138
Inverted Video 29
I/O (input/output)
  Analog 61
  Disk 119
  Distribution 143
  Flag Bit Setting 119

J

Joysticks 42

K

Keyboard 40, 92
Keypads 47
Key Usage 41

L

Labeling 13
Languages 31, 38, 67, 124, 133, 135
Left$ 16
List 83, 121
Loops, Explanation of 20
LUN 120

M

M- 41
Machine Code Language 31, 124, 135
Machine Organization 76
Memory
  Allocation 118
  Map 79
  Move 82
  Video Maps 28
MID$ 16
Modem 60, 63
Monitor, Extended 121
Monitor (machine) 124
  Real Time (RTMON) 71
Multiplexer 61, 62

N

NEW 124
Notations 7
Null Device 147

O

Organizations
  Machine 76
  Operating System 38
  Floppy Disk 78
OS-65D User's Guide 116

P

Page 78
Passing Parameters 131
Peripherals 63, 65, 67
PIA Data 54, 55, 60, 71
Piano Keyboard 92
Plot BASIC 67
POKEs and PEEKs 87
Pound Sign 120
Power
  Down 8
  Up 6
PRINT 81
Printer 65
Program Mode 13
Prototyping 61

R

RAM 33, 78
Read/Write
  Cassette 35
  Disk 36
  Track Ø 96
Real Time
  Clock 68
  Control 68
  Monitor (RTMON) 54, 69, 70, 72
Remarks 15
RENAME 93
Return 5, 6, 9, 41
RF Modulator/Standard TV 2
Right$ 16
ROM 14
RUN 8, 9, 13

S

Screen Cleaning Routine (USR-X) 127
SCRTCH 36, 98
Sector 78
Security 52
SEQLST (sequential file lister) 145
Shift (O) 14, 41
Shift (P) 41
SL 41
Source File Format 123
Sound 30
Space Bar 41
Strings 15
Subscripted Variables 21, 24
Summary 62
Switches (external) 54
Syntax (Error) 75
System Organization 38

T

Telephone Interface 63
Terminal Communications 63
Time
  Control 68
  Of Day Clock 68
  Monitor (RTMON) 69
Timer (countdown) 69
Tone Generator 30
Track Formatting 80, 123
Track Ø Read/Write 96, 124
Troubleshooting 75
Tutor, Hexadecimal to Decimal 99

U

Unlock 7, 19, 12
Up Arrow 123
USR(X) Function 67, 126

V

Video
  Connections 2
  Close Circuit 2
  Direct (modifications) 2
  RF modulator 2
  Memory Map 78, 115
  Screen Layout 110

X

XQT
  File Storage 135
  Filnam Command 118

Z

Zero Utility 80

NOTES