PART II

TABLE OF CONTENTS

1.0	General Description	1
1.1	Summary of Features	3
1.1.1	Microcomputer	3
1.1.2	I/O Devices	3
1.1.3	Graphical Video	4
1.1.4	Graphical Sound	4
1.1.5	Resident Software	4
1.1.6	Compatible System Software	5
1.1.7	Functional Block Diagram	6
2.0	Microcomputer	7
2.1	Introduction	7
2.2	Memory Usage	7
2.3	Timer and Interrupt Control	9
2.3.1	Interrupt Control	9
2.4	Master RAM Memory	10
2.4.1	Programmable RAM Select Logic	11
2.4.2	Master RAM Configurations vs Graphical Capability	11
2.5	ROM and Static RAM Memory	12
2.5.1	Simplified memory map	13
3.0	Programmable Video Graphics Generator	14
3.1	Introduction	14
3.2	Screen Data Format	16
3.2.1	Control Word Format	16
3.2.1.1	High Address Byte (Mode byte)	16
3.2.1.2	Low Address Byte (Colour type) + colon TABLE	18
3.2.2	Data Mode	19
3.2.2.1	Four Colour Mode	19
3.2.2.2	Sixteen Colour Mode	19
3.2.2.3	Character Mode	20
3.2.2.4	Unit Colour Mode	27
3.3	Video Interface	27

4.0	Programmable Graphic Sound Generator	28
4.1	Introduction	28
4.2	Programmable Oscillators	28
4.2.1	Frequency Selection	28
4.2.2	Volume Control	29
4.3	Random Noise Generator	29
4.4	Frequency Mixing	29
4.5	Frequency Calculator formula	30
5.0	Input-Output Section	30
5.1	Introduction	30
5.2	Game Paddle Interface	31
5.3	Audio Cassette Interface	32
5.4	Stereo Interface	32
5.5	Scientific Math Peripheral	33
5.6	ASCII Keyboard	34
5.6.1	Keyboard Layout	34
5.6.2	Keyboard scan logic	35
5.7	DCE-BUS	36
5.7.1	DCE-BUS Pinout	37
5.8	RS 232 Interface	38
5.9	I/O Device Address Allocation Reference	39
5.9.1	Master Control Device Addresses	39
5.9.2	Discrete I/O Device addresses	40
5.9.3	Serial I/O, Timer & Interrupt Control Addresses	42
6.0	Resident System Software Guide	44
6.1	Introduction	44
6.2	Resident DAI BASIC	45
6.2.1.	Alphabetic Index of DAI BASIC Statements	45
6.2.1.1	BASIC Commands	45
6.2.1.2	BASIC Functions	46
6.2.1.3	Arithmetic and Logical Operators	47

6.2.2	Format Rules and Constraints	47
6.2.2.1	Variables and Numbers	47
6.2.2.2	Strings	49
6.2.2.3	Operators	50
6.2.2.4	Statements	50
6.2.2.5	Expressions	51
6.2.3	Error Reporting	52
6.2.3.1	Error Report Format	52
6.2.3.2	Error Message Dictionary	52
6.2.4	Interacting with DAI BASIC	56
6.2.4.1	Facilities of the character screen	56
6.2.4.2	Input of Programs and Data	57
6.2.4.3	Amending and Running of Programs	57
6.2.4.4	Merging of BASIC Programs	59
6.2.4.5	Merging of BASIC and Machine Language Programs	59
6.2.5	User Control Statements	61
6.2.5.1	EDIT	61
6.2.5.2	IMP	61
6.2.5.3	LIST	62
6.2.5.4	NEW	63
6.2.5.5	RUN	63
6.2.6	Program Control Statements	64
6.2.6.1	END	64
6.2.6.2	FOR.....NEXT	64
6.2.6.3	GOSUB	65
6.2.6.4	GOTO	66
6.2.6.5	IF.....GOTO	66
6.2.6.6	IF.....THEN	66
6.2.6.7	ON.....GOSUB	66
6.2.6.8	ON.....GOTO	66
6.2.6.9	RETURN	67
6.2.6.10	STOP	68
6.2.6.11	WAIT	68

6.2.7	Physical Machine Access Statements	69
6.2.7.1	CALLM	69
6.2.7.2	INP	69
6.2.7.3	OUT	69
6.2.7.4	PDL	70
6.2.7.5	PEEK	70
6.2.7.6	POKE	70
6.2.7.7	UT	71
6.2.8	BASIC System Data & I/O Statements	71
6.2.8.1	DATA	71
6.2.8.2	GETC	72
6.2.8.3	INPUT	72
6.2.8.4	PRINT	73
6.2.8.5	READ	73
6.2.8.6	RESTORE	73
6.2.9	Cassette and Disc I/O Statements	74
6.2.9.1	CHECK	74
6.2.9.2	LOAD	74
6.2.9.3	LOADA	75
6.2.9.4	SAVE	75
6.2.9.5	SAVEA	76
6.2.10	Program Debug and Comment Statements	77
6.2.10.1	CONT	77
6.2.10.2	REM	77
6.2.10.3	STEP	77
6.2.10.4	TRON	78
6.2.10.5	TROFF	78
6.2.11	Array and Variable Statements	78
6.2.11.1	CLEAR	78
6.2.11.2	DIM	78
6.2.11.3	FRE	79
6.2.11.4	LET	79
6.2.11.5	VARPTR	79

6.2.12	Graphics and Display Statements	80
6.2.12.1	MODE	80
6.2.12.2	COLORG	82
6.2.12.3	COLORT	82
6.2.12.4	Drawing Facilities	82
6.2.12.4.1	DOT	83
6.2.12.4.2	DRAW	83
6.2.12.4.3	FILL	83
6.2.12.5	ANIMATED DRAWING FACILITY	84
6.2.12.6	XMAX	87
6.2.12.7	YMAX	87
6.2.12.8	SCRN	87
6.2.12.9	CURSOR	87
6.2.12.10	CURX	88
6.2.12.11	CURY	88
6.2.13	Graphical Sound Statements	89
6.2.13.1	Programmable Sound Facility	89
6.2.13.2	SOUND	92
6.2.13.3	ENVELOPE	92
6.2.13.4	NOISE	93
6.2.13.5	FREQ	93
6.2.13.6	SYNTHESING VOCAL SOUND	94
6.2.13.6.1	TALK	94
6.2.14	Arithmetic and String Functions	95
6.2.14.1	ABS	96
6.2.14.2	ACOS	96
6.2.14.3	ALOG	96
6.2.14.4	ASC	96
6.2.14.5	ASIN	96
6.2.14.6	ATN	96
6.2.14.7	CHR$	97
6.2.14.8	CCS	97
6.2.14.9	EXP	97
6.2.14.10	FRAC	97

6.2.14.11 HEX$ 97
6.2.14.12 INT 98
6.2.14.13 LEFT$ 98
6.2.14.14 LEN 98
6.2.14.15 LOG 98
6.2.14.16 LOGT 98
6.2.14.17 MID$ 99
6.2.14.18 PI 99
6.2.14.19 RIGHT$ 99
6.2.14.20 RND 99
6.2.14.21 SGN 100
6.2.14.22 SIN 100
6.2.14.23 SPC 100
6.2.14.24 SQR 100
6.2.14.25 STR$ 100
6.2.14.26 TAB 101
6.2.14.27 TAN 101
6.2.14.28 VAL 101
6.2.15 Arithmetic and Logical Operators 102

7.0 Machine Language Utility 106
7.1 Introduction 106
7.2 User Interface 106
7.3 Utility Commands 108
7.3.1 LOOK 109
7.3.2 DISPLAY 110
7.3.3 GO 111
7.3.4 FILL 111
7.3.5 SUBSTITUTE 112
7.3.6 MOVE 113
7.3.7 EXAMINE 113
7.3.8 EXAMINE REGISTERS 114
7.3.9 VECTOR EXAMINE 114
7.3.10 VECTOR EXAMINE BYTES 115
7.3.11 READ 115
7.3.12 WRITE 116

CONVERSION TABLE 118
USEFUL POKES 120
PROGRAM EXAMPLES 120
ON TAPE "ACTIVATE" 122
TO ACTIVATE CASSETTE OR FLOPPY 122
SOFTWARE PROTECTION 123
HOW TO GET RESTARTED IF ACCIDENTAL
RESET DURING PROGRAM KEYING OR AT END
OF PROGRAM 124
SAVING AND RELOADING A DRAWING 124
USEFUL ADDRESSES & DATA 126
PROGRAMS EXAMPLES 139
to 181

1.

GENERAL DESCRIPTION

The DAI Personal Computer is designed to provide the maximum capability that can economically be provided to an individual. The design is realised such that programs are loaded from a low cost audio cassette or a floppy disc. The results of program execution are output to the user via an antenna connector for PAL, SECAM or NTSC standard television receiver. The Graphical Sound Generation also outputs two tracks of separated sound for left and right stereo connections, and the sound channel of the television.

The resources of the DAI Personal Computer are partitioned into four segments; the Microcomputer Section, Programmable Graphical Video Section, the Sound Generator Section and the I/O Section. To optimise usage of components within the design, considerable overlay of logic usage exists within the system. Figure 1 is a logical block diagram of the DAI Personal Computer.

The resident software is comprised of six major modules, Basic Interpreter, Math Package, Screen Driver Module, Keyboard Scan + Encode Routine, the Machine Language Utility and the General House-keeping Module.

The Basic Interpreter incorporates most of the features found in other Personal Computers as well as special statements to control the video graphics and sound generator and interface with the Machine Language Utility as well as assist with generation and editing of source programs. In order to obtain the minimum possible execution time the design of the Basic System is such that it functions as a quasi-interpreter. When the user types in his source program it is compressed and encoded into a special "run-time" code so that the Execution Routine has the smallest possible amount of work left to do.

The Math Package is broken into an Integer Math Module and a Floating Point Math Module. The integer module performs only basic operations as +, -, multiply etc., while the Floating Point Math Module provides these plus transcendental functions.
Integer variables are calculated to nine digit resolution and floating point variables to 6 digit resolution. The Math Package handles floating point numbers in the range $\pm 10^{-18}$ to $\pm 10^{18}$, and zero. When the Scientific Math option is inserted into its socket the Math Package automatically uses it for calculations instead of the software calculation modules.

The Screen Driver Module is responsible for arranging the data in memory to give a correct picture in all modes. It also handles the changing of screen colours, the drawing facilities (DOT, FILL, DRAW) and other screen-related facilities.
The Keyboard of the DAI Personal Computer is a simple matrix of 56 keys connected in an 8 x 7 matrix. The Keyboard Scan + Encode Routine scans the keyboard at fixed time intervals, detects key depressions and encodes a specific key according to a look-up table. Since the keyboard of the DAI Personal Computer has been constructed in this fashion it is possible to provide DAI Personal Computers with other configurations and codes. The keyboard driver software provides for a 3 key rollover mechanism.

The Machine Language Utility is a complete set of keyboard and subroutine callable functions that permit and assist the generation, loading, de-bugging, and execution of machine language programs and subroutines. The control subroutines and housekeeping subroutines of this module allow direct interface between BASIC programs and machine language program and subroutines. An unlimited number of machine language subroutines may be called by a BASIC program.
The General Housekeeping Module is a set of routines that are shared by other modules, providing for instance, the control of memory bank switching. This allows the 8080A microprocessor to operate with 72K bytes of memory instead of the 64K normally.

1. 1 Summary of features

1. 1. 1 Microcomputer

8080A microprocessor running at 2MHz.
8K, 12K, 32K, 36K, 48K RAM memory configurations
24K PROM/ROM capability (software bank switched)
Memory mapped I/O
AMD 9511 math chip support logic
Hardware random number generator
Stack overflow detect logic.

1. 1. 2 I/O Devices

ASCII Keyboard
PAL/SECAM/NTSC/VIDEO TV connection via antenna input (color and B/W)
Sound channel audio modulated on TV signal.
Dual low cost Audio cassette input and output with stop/start control.
Stereo hi-fi output channels
Left and Right game paddle inputs (6 controls)
Interface bus (DAI's DCE-BUS) to:
- floppy disk controller
- printer controller
- standard interface cards (DAI's RWC family)
  - IEEE bus adaptor
  - communication interconnections
  - control connection
  - prom programming
  - special interfaces
  - analog input and output

RS232 Interface
- Programmable baud rates
- Terminal or modem function

1.1.3

Graphical Video

Character screen mode (66 characters x 24 lines normally 11/22/44/66 characters + 13 to 32 lines possible)

16 colors or grey scales

Multiple resolution graphics modes (software selectable)

- 65 x 88
- 130 x 176
- 260 x 352

(Intermixed mode screens of lines of characters and graphics are possible).

True"square" graphics.

1.1.4

Graphical Sound

3 independently programmable frequencies

1 programmable noise generator

Amplitude and frequency software selectable

- smooth music
- random frequencies
- enveloped sound
- vocal sound generator

1.1.5

Resident Software

Extended Highspeed BASIC interpreter

Full floating point scientific math commands.

Hardware scientific functions automatically used if math module present.

Graphical video commands

- full graphic plotting
- arbitrary line specification
- arbitrary dot placement
- filling of arbitrary rectangles

Graphical sound commands

- predetermined volume envelope specification
- individual specification of frequency

- individual specification of volume
- individual specification of tremolo
- individual specification of glissando

Machine Language Utility.

1.1.6

Compatible System Software

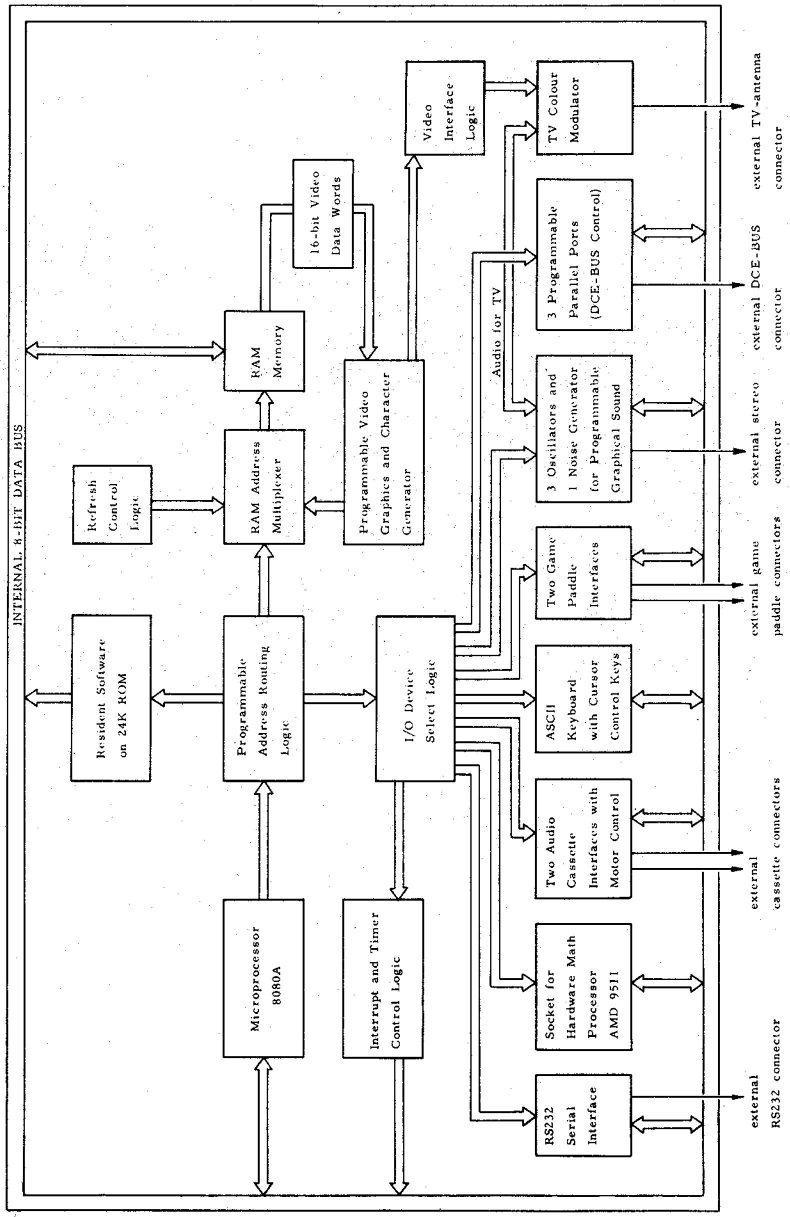
DAI Assembler

8080A Standard software support

FORTRAN Compiler support

MDS/Intellec non-disc software support.

1.1.7
Functional Block Diagram



INTERNAL 8-BIT DATA BUS
Refresh Control Logic
RAM Address Multiplexer
RAM Memory
16-bit Video Data Words
Programmable Video Graphics and Character Generator
Video Interface Logic
TV Colour Modulator
3 Programmable Parallel Ports (DCE-BUS Control)
Audio for TV
3 Oscillators and 1 Noise Generator for Programmable Graphical Sound
Two Game Paddle Interfaces
ASCII Keyboard with Cursor Control Keys
Two Audio Cassette Interfaces with Motor Control
Socket for Hardware Math Processor AMD 9511
RS232 Serial Interface
Resident Software on 24K ROM
Programmable Address Routing Logic
I/O Device Select Logic
Microprocessor 8080A
Interrupt and Timer Control Logic
external TV-antenna connector
external DCE-BUS connector
external stereo connector
external game paddle connectors
external cassette connectors
external RS232 connector

2.0
MICROCOMPUTER

2.1
Introduction

The DAI Personal Computer's processor section is designed around the 8080A Microprocessor. The design is based upon the popular and economical high performance DCE microcomputer architecture. The microcomputer section consists of the microprocessor and timing circuitry; the ROM and Static RAM memory; Interrupt Control and Interval Timer logic; and the Master RAM memory. The Master Ram memory consists of a dynamic memory that is configurable from 8K bytes up to 48K bytes.

2.2
Memory Usage

The DAI Personal Computer's memory space is organised on the basis of memory mapped input-output which allocates normal memory addresses to all I/O operations alongside the RAM and ROM memory addresses that are required for normal system operation.
In the following descriptions the address space is described in terms of hexadecimal numbers where the available range of 64 kilobytes is represented by the address range 0000 to FFFF. Switched banks represent a duplication of addresses.

0000	-	003F	INTERRUPT VECTOR
0040			CONTROL OUTPUT IMAGE
0041	-	0061	UTILITY WORK AREA
0062	-	0071	UTILITY INTERRUPT VECTOR.
0077	-	00CF	SCREEN VARIABLES
00D0	-	00FF	MATH WORK AREA

0100 - 02EB	BASIC VARIABLES
02EC TO TOP OF RAM (VARIABLE BOUNDARIES)	HEAP(STRINGS + ARRAYS)
	PROGRAM (COMPILED BASIC)
	SYMBOL TABLE
	NOT USED RAM
	SCREEN DISPLAY
F800 - F8FF	uC STACK

The following two byte variables are maintained by the system. Addresses are stored on low order byte, high order byte (8080A)

Address (Hex)	Variable
∅29B	START OF HEAP
∅29D	SIZE OF HEAP
∅29F	START OF PROGRAM BUFFER
∅2A1	END PROGRAM BUFFER AND START SYMBOL TABLE
∅2A3	END SYMBOL TABLE
∅2A5	BOTTOM OF SCREEN RAM AREA

2.3 Timer and Interrupt Control

The DAI Personal Computer has 5 interval Timers programmable from 64 us to 16 ms, 2 external interrupts and 2 serial I/O interrupts. These are priority encoded with a masking system and allow an automatic or polled interrupt system to be used.

2.3.1 Interrupt Control

The 8 interrupt vector addresses provided by the 8080 are assigned the following functions:

Vector Address (Hex)	Allocated function
00	Timer 1
08	Timer 2
10	External interrupt
18	Timer 3
20	Receive buffer full
28	Transmit buffer empty
30	Timer 4
38	Timer 5/auxiliary interrupt

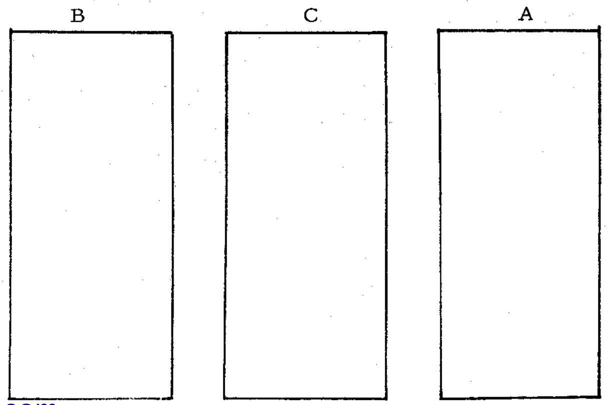
The external interrupt is connected to a signal which indicates that the address range F000 to F7FF has been accessed. This condition normally indicates a "stack overflow" condition.
The auxiliary interrupt is connected to a page signal from the TV picture logic. This provides a convenient 20 ms clock for timing purposes.
More complex features of this part of the logic are beyond the scope of this manual, and anyone needing such information should refer to the DAI publication "DCE MICROCOMPUTER SYSTEMS DESIGNER'S HANDBOOK".
The programming advice given on the TICC is valid also for Personal Computer systems. The access to the keyboard is also via the same logic, using the associated parallel input and output ports.

2.4 Master RAM Memory

The Master RAM memory is divided into three separate memory banks, called A, B, C. With one restriction each RAM memory may contain 4K or 16K dynamic RAM chips or they may be left empty. This yields a total RAM availability from 8K to 48K bytes.

The addressing of the dynamic RAM is controlled by a single PROM programmed to correspond to the physically present RAM configuration. The exchange of this chip and changing of a switch is the only operation, other than replacement of RAM chips, that is necessary to implement a configuration change.
The RAM memory is seen by the program as a continuous block of memory starting at (hex) address 0000 up to a maximum address which for 48K is BFFF.

The first RAM bank, (if present) starts at address 0000 and is available for program use only and may not contain display data. The remaining two banks which must both be present are arranged for 16 bit (two-byte) wide access by the display controller. Bank B contributes the low-order bits, and bank C the high-order bits of the 16 bit word. For processor access even-address bytes are in bank B and odd-address bytes are in bank C, e.g. : if bank A is 4K and occupies addresses 0000 to 0FFF then address 1000 is in bank B, address 1001 is in bank C etc. to the end of the Master RAM.



2.4.1 Programmable RAM select Logic

For each RAM configuration of the DAI Personal Computer it is necessary to define the address decoding. This is achieved using a single factory programmable ROM. These are supplied for each defined RAM configuration.

RAM configuration	Banks B+C address	Bank A
8K	0000 - 1FFF	not used
12K	1000 - 2FFF	0000 - 0FFF
32K	0000 - 7FFF	not used
36K	1000 - 8FFF	0000 - 0FFF
48K	4000 - BFFF	0 - 3FFF

No other aspect of the machine is altered by changes to the RAM configuration.

2.4.2 Master RAM Configurations VS Graphical Capability

Master RAM Configuration	Graphical Resolution	Display Color Modes	Required Picture Space	Available Prog. and Work space	Notes
8K	65 x 88	4 16	1.5K	6.5K	
	130 x 176	4 16	5.8K	2.2K	
12K	65 x 88	4 16	1.5K	10.5K	
	130 x 176	4 16	5.8K	6.2K	
32K	65 x 88	4 16	1.5K	30.5K	
	130 x 176	4 16	5.8K	26.2K	
	260 x 352	4 16	22.8K	9.2K	
36K	65 x 88	4 16	1.5K	34K	
	130 x 176	4 16	5.8K	30K	
	260 x 352	4 16	22.8K	13K	
	240 x 528	4 16	32K	4K	

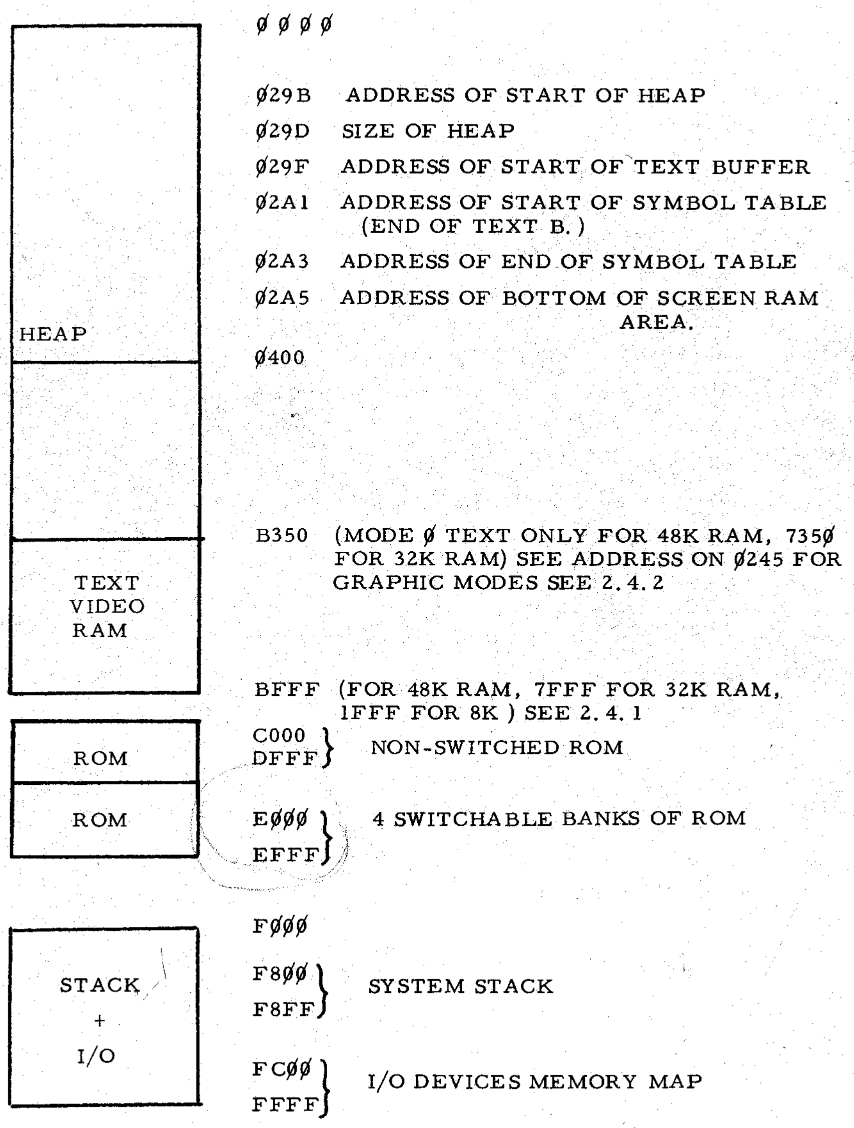
48K	65 x 88	4	16	1.5K	46.0K	
	130 x 176	4	16	5.8K	42.0K	
	260 x 352	4	16	22.8K	25.0K	
	240 x 528	4	16	32 K	16.0K	non-square

The above are examples of the RAM requirement for possible all-graphics screen configurations. Actual usage will be affected by the screen driver package used.

2.5 ROM and Static RAM Memory

The system software resides in mask programmed ROM'S starting at address C000 and extending to EFFF. Addresses C000 through DFFF are continuous program space while addresses E000 through EFFF have four switchable BANKS of program space. Total program ROM space is therefore 24K bytes. In the address range F800 to F8FF a bank of static RAM is included for use by the 8080A stack, and for a vector of jump instructions that allow the emulation of an MDS system.

2.5.1 Simplified memory map (48K RAM P.C.).



ØØØØ

Ø29B ADDRESS OF START OF HEAP
Ø29D SIZE OF HEAP
Ø29F ADDRESS OF START OF TEXT BUFFER
Ø2A1 ADDRESS OF START OF SYMBOL TABLE (END OF TEXT B.)
Ø2A3 ADDRESS OF END OF SYMBOL TABLE
Ø2A5 ADDRESS OF BOTTOM OF SCREEN RAM AREA.
Ø400

B350 (MODE Ø TEXT ONLY FOR 48K RAM, 735Ø FOR 32K RAM) SEE ADDRESS ON Ø245 FOR GRAPHIC MODES SEE 2.4.2

BFFF (FOR 48K RAM, 7FFF FOR 32K RAM, 1FFF FOR 8K) SEE 2.4.1
C000 } DFFF } NON-SWITCHED ROM
EØØØ } EFFF } 4 SWITCHABLE BANKS OF ROM
FØØØ
F8ØØ } F8FF } SYSTEM STACK
FCØØ } FFFF } I/O DEVICES MEMORY MAP

3.0 PROGRAMMABLE GRAPHICS GENERATOR

3.1 Introduction

The programmable video graphics + character system makes use of a scheme of variable length data to give efficient use of memory when creating pictures.
A few definitions are necessary before further examination of the scheme.

A "Scan" is:
One traverse of the screen by the electron beam drawing the picture. (there are 625 in a European television picture).

A "Line" is:
A number of scans all of which are controlled by the same information in the RAM.

A "Mode" is:
One of the different ways information may be displayed on the screen. For instance, in "character mode" bytes in memory are shown as characters on the screen, in "4 colour graphics" mode, bytes describe the colour of blobs on the screen.

A "Blob" is:
The smallest area on the screen whose color can be set (The physical size of a blob is different in different screen modes).

A "Field" is:
A set of 8 blobs whose colour is controlled by a pair of bytes from memory.

The picture is defined by a number of lines, one after another down the screen. Each line is independent of all others and may be in any of the possible modes.
At the start of each line two bytes are taken from memory which define the mode for that line, and may update the colour RAM two bytes. These are called respectively the Control and Colour Control bytes. The rest of each line is colour or character information, and the number of bytes used for it is a characteristic of the particular mode. (see example programs).
The screen can operate at a number of different definitions horizontally (e. g. blobs/scan). In the highest definition graphics mode there are 352 visible blobs across the screen. The two lower definitions have respectively 1/2 and 1/4 of this number. There are about 520 scans visible on a "625 line" television, and the screen hardware can only draw (at minimum) 2 scans per line, due to the interlacing. This gives a maximum definition of 260 by 352 which is close to the 3:4 ratio of the screen sides. Thus circles come out round !
Characters are fitted onto this grid by using 8 columns of blobs per character, the dot positions being defined for each character by a ROM. This allows 44 characters per line maximum (or 22/11 in lower definition modes).
→ A fourth horizontal definition provides for a "high density" character mode with 66 characters/line.
A total of 16 different colours, including white and black can be displayed by the system. Whenever a 4 bit code is used to describe a colour, it selects from this range of possibilities. In some modes (characters + or four colour graphics) a set of 4 of these colours (not necessarily distinct) are loaded into a set of "colour registers". Any 2 bit code describing a colour selects an entry from these registers.
Vertical definition is set by a 4 bit field in the control byte. In graphics modes this simply allows repetition of the information to fill any even number at scans from 2 to 32. In character mode it defines the number of scans occupied by each line of characters; thus the vertical spacing on the screen can be changed to allow anything between an 8 x 7 (the sensible minimum) and 8 x 16 character matrix, giving between 35 and

15 lines of characters on the screen.

Arrangement of information in memory

The first byte of information for the screen is located at the top of an 8K or 32K block of memory. Successive bytes follow at descending addresses. The screen takes memory and displays a picture on the screen accordingly until the whole screen has been filled. It then starts again at the first byte.

3.2 Screen Data Format

At the beginning of the data for each line, two bytes of data represent the lines control word. The control word defines the raster scan depth of the line, the horizontal graphical resolution of the line and selects the display mode of that particular line. Subsequent to this control word a number of data w ords are stored that represent the colour of pixels,or definition and colour of characters according to the selected display mode.

3.2.1 Control Word Format

3.2.1.1 High Address Byte (Mode byte)

```
Bits 7,6              5,4            3,2,1,0
  |                    |                |______Line Repeat Count
  |                    |
  |                    |______Resolution Control
  |
  |______Display Mode Control
```

Line Repeat Count

The line repeat count controls the number of horizontal raster scans for which the same data will be displayed. Since interlace of the TV scan is ignored a minimum of two raster scans correspond to a line repeat count of zero. Thereafter, each additional repeat adds two scans to the line. The maximum programmable depth of any horizontal display segment is thus 32 scans. (European TV sets will show approximately 520 scans total for a full picture).

Resolution Control

The resolution control bits allow selection of one of four different horizontal definitions for display of data on the TV screen for each individual line.

Code (Bit 5, Bit 4)	Definition (pixels per screen width)
00	88 (Low definition graphics)
01	176 (Medium definition graphics)
10	352 (High definition graphics)
11	528 (Text with 66 characters per line) (Screendriver uses 60 characters for text). (Could be used for a very high definition graphics mode).

Mode Control

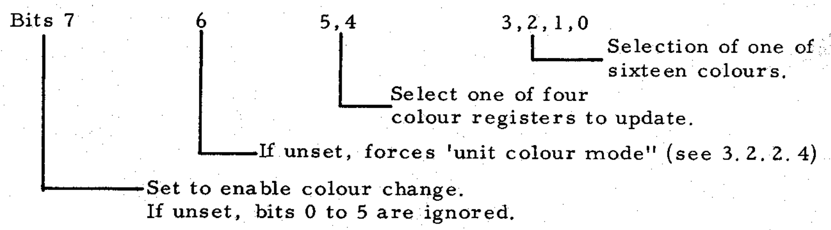
The mode control bits determine how data will be used to generate the picture for that particular segment.

Code (Bit 7, Bit 6)	Display mode
00	Four colour graphics
01	Four colour characters
10	Sixteen colour graphics
11	Sixteen colour characters

3.2.1.2

Low Address Byte (Colour type)

The Low Address control byte is used to store colours into a set of 4 "colour registers" for the four colour mode. Any one of the four colours in the registers can be changed at the beginning of any line of display data. Only the colours in these registers can be displayed in any 4 colour mode. The four colours are freely selectable from the sixteen colours defined in Colour Select Table.


Bits 7
6
5,4
3,2,1,0
Selection of one of sixteen colours.
Select one of four colour registers to update.
If unset, forces 'unit colour mode" (see 3.2.2.4)
Set to enable colour change. If unset, bits 0 to 5 are ignored.

Code	Code
0	Black
1	Dark blue
2	Purple Red
3	Red
4	Purple Brown
5	Emerand Green
6	Kakhi Brown
7	Mustard Brown
8	Grey
9	Middle Blue
10	Orange
11	Pink
12	Light Blue
13	Light Green
14	Light Yellow
15	White

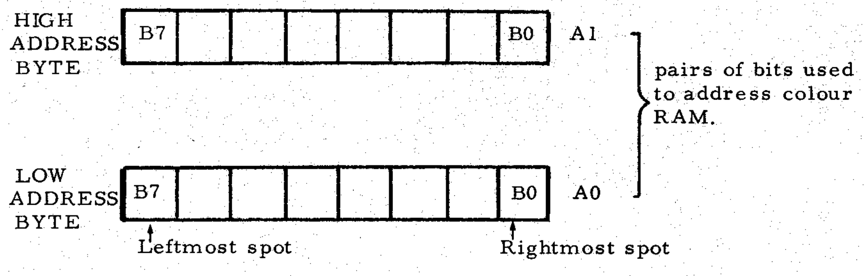
3.2.2

Data Mode

3.2.2.1

Four Colour Mode

In this mode only two bits of data are required to define the colour of a pixel. These data bits are obtained in parallel from the upper and lower bytes of each data word using the high order bits first.
The 2 bytes in a field are considered as 8 pairs of bits. Each pair sets the colour for one spot.



The 2 bits for each spot select one of the four colours which have been loaded into the colour RAM by previous Colour Control bytes. So on any line 4 colours are available. On the next line any one of these may be changed for another, and so on.

3.2.2.2

Sixteen Colour Mode

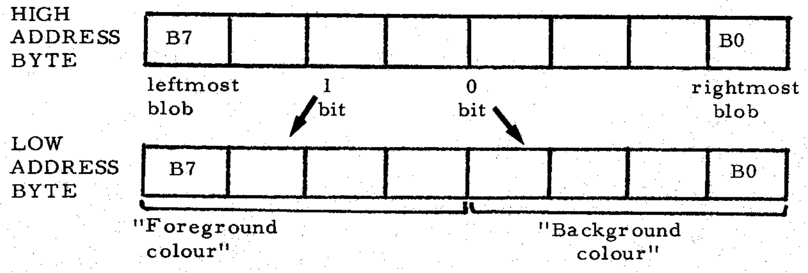
This graphics mode is designed to allow multi-colour high definition pictures in half the memory requirement of other systems.
The basic organization is that the low address byte selects two of the sixteen possible colours.

Bits 0 - 3	"Background" colour.
Bits 4 - 7	" Foreground" colour.

The high address byte than defines by each successive bit whether a colour blob should be foreground or background.

NB

The two bytes in the field serve different purposes, one being used to define two available colours for use in the field, and the other to choose one of these for each spot.



The bit for each spot can select either the "foreground" or the "back-ground" colour. However, what these colours are is totally independent of the preceding or following fields. So any line may use any and all of the total 16 colours. The contents of the colour RAM are irrelevant in this mode.

One additional feature is added to eliminate restrictions of the scheme. After each eight bit field of colour the background is extended into a new area, even if a new background colour is specified, until the new foreground is first used. It is therefore possible to create a required picture by suitable combination of foreground and background.

3.2.2.3

Character Mode

In this mode, characters are generated using a character generator ROM in conjunction with the four colour registers or using any 2 colours for each in the 16 colour character mode.

The usual character matrix is 6 x 9 bits out of a possible 8 x 16. Therefore the line repeat count should be at least eleven, to guarantee full character display plus line spacing.

Four colour characters are produced on the screen in a way similar to the four colour graphics mode, but with the character ASCIV data replacing the high address data byte used for four colours. The result is that characters are displayed using colours from the four colour registers. The data from the character generator ROM control the lower address bit and bits from the low-address byte determine the other. This allows characters on a single horizontal display segment to be in one of two colour combinations of character/background, or even with a vertical striped pattern controlled by the low address byte. However, note that as compared with four colour mode information (but not the low-address byte) is subject to a one character position delay before appearing on the screen.
In character mode the height of the characters is a set number of horizontal scans. The character width is determined by the definition selection in the control byte. A definition of 352 yields 44 characters per line, 528 hields the normal 66 characters per line. Other definitions are possible and they yield wide characters, useful as large capitals in applications such as the power-on message. However, this feature is not supported by the resident BASIC.

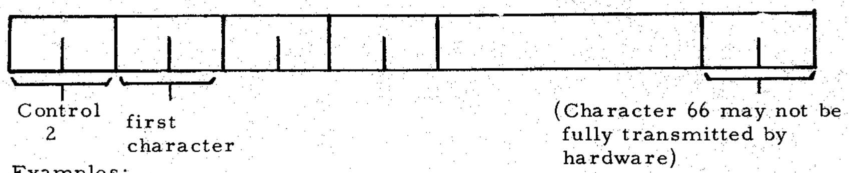
Special characters:

CR	Terminates a line of characters and positions the cursor at the first position of next line. If necessary, the screen is "rolled up" to make room.
FF	Fills the character area with spaces and positions the cursor at the start of the tope line on the screen.
BS	If the current line has some characters on it, then the cursor is moved back to the previous position and the character there is replaced by a space.

- A line of characters on the screen can be extended up to 4 screen widths. Continuations are indented a few characters, and a letter "C" is displayed in the first position of these lines.
- When a third continuation line is full any character except CR, FF and BS is ignored.
- Attempts to backspace past the beginning of the line are ignored.
- If the screen is in "all graphic mode" and character output is necessary then a mode change will be to an appropriate mode including a character area. First the corresponding "split" mode will be tried e.g. if the screen is in mode 1, then mode 1A. If in mode 1 a program claims all free memory (e.g. by using "CLEAR") then mode 1A, which requires more memory than mode 1, will not be possible and the default is to mode 0. In this case the program is deleted by an automatic "NEW" command.

CHANGING LINE BACKGROUND OR LETTERS COLOR ON ONE LINE

Line 1 Control byte is located at address XFEF and line 1 Color Control byte address at XFEE (X being 1 for 8K machine, 7 for 32K machine, B for 48K machine). The first character byte of line 1 is located at line 1 Control byte address minus 2, and the character Colour Control byte at line 1 Control byte address minus 3. Each of the 66 positions of the screen is located at line Control byte - (2 * position of character on the line) for the character and at line Colour Control byte - (2 * position of character on the line) for the Colour Control byte of the character. Remember that there are 66 character positions on the screen but that the first and last three characters are kept blank for the margins. Therefore the Control byte for the next line is located at Control byte of previous line (i.e. XFEF) less 134 bytes (#86. So if the Control byte of line 1 is a BFEF, the Control byte of line 2 will be at #BFEF - # 86 = #BF69.



Examples:

Control Byte Line 1	#BFEF	
Control Byte Line 5	#BFEF - (# 86 * 5)	= #BDD7
Colour Control byte Line 5		= #BDD6
Character N° 6 on Line 5	# BDD7 - 6 * 2	= #BDCC
Colour Character 6 of Line 5		= #BDCB

(see VIDEO RAM TABLE and examples 1 and 2)

Use the POKE in your program for changing line background, letter colour, or letter, and Utility 3 for checking the location you intend to POKE (when you return to BASIC the colour changes you made in Utility mode are erased if you enter MODE 1, RETURN, MODE 0.

Example

```
   COLORT 8 0 5 10
   POKE #BA2D,#DA    (Will change colour of letter from black 0 to
                      colour 10 on line 12)
   POKE#BA2D,#C3     (Will change background from 8 to 3)
```

The locations from #x350 to #x35F and #xFF0 to #xFFF
x = 1 FOR 8K RAM, x = 2 FOR 12K, x = 7 FOR 32K, x = B FOR 48K
control the screen background and foreground colours

Example
COLORT 0 15 7 8

```
     00 00 B8 3F 00 00 A7 3F 00 00 9F 3F 00 00 80 3F


     00 00 B8 36 00 00 A7 36 00 00 9F 36 00 00 80 36
```

*POKE#735A,#90:POKE#7FFA,#90:POKE#735E,#80:POKE#7FFE,#80

You will see the screen black and the letters black
the # numbers 90 and 80 can be replaced by any # number
from #90 to #9F and #80 to # 8f

Changing colour of background and text

Example 1

```
10    MODE 0
15    REM START AT #BEE2 for 48K, #7EE2 for 32K, #2EE2 for 12K, #1EE2 for 8
20    COLORT 3 0 5 15
25    FOR A%=1 TO 23:PRINT A%,:FOR B=0.0 TO 40.0:PRINT "+";:NEXT:PRINT :NEX
30    REM YOU FIND IN    LINE  1 - 2 TEXT COLOUR  0 BACKGROUND  8
35    POKE #BEE2,#CF:REM LINE  3 - 7 TEXT COLOUR  0 BACKGROUND 15
40    POKE #BC44,#DF:REM LINE  8 - 9             15            15
50    POKE #BB38,#D8:REM LINE 10      (no text)   3            15
60    POKE #BAB2,#D0:REM LINE 11 -12              0            15
70    POKE #B9A6,#DF:REM LINE 13 -14 (no text)   15            15
80    POKE #B89A,#D5:REM LINE 15                  5            15
90    POKE #B814,#D0:REM LINE 16                  0            15
92    POKE #B78E,#DF:REM LINE 17 -18 (no text)   15            15
93    POKE #B682,#C6:REM LINE 19 -21             15             6
94    POKE #B4F0,#C8:REM LINE 22 -24             15             8
95    GOTO 95
```

Example 2

```
10    E%=#FF
20    COLORT 8 0 0 8
25    REM START AT #BEE2 for 48K, #7EE2 for 32K, #2EE2 for 12K, #1EE2 for 8
30    B%=#BFEF
40    FOR A%=1 TO 23
50    D%=B%-3
60    FOR C%=0 TO 65
70    POKE D%,E%
80    D%=D%-2:NEXT
90    B%=B%-#86:NEXT
93    E%= INOT E% IAND #FF
95    GOTO 30
```

VIDEO RAM TABLE

Line N°	Start Address of Line (in Hex)	Line Colour Control byte Address (in Hex)
1	XFEF	XFEE
2	XF69	XF68
3	XEE3	XEE2
4	XE5D	XE5C
5	XDD7	XDD6
6	XD51	XD5Ø
7	XCCB	XCC4
8	XC45	XC44
9	XBBF	XBBE
10	XB39	XB38
11	XAB3	XAB2
12	XA2D	XA2C
13	X9A7	X9A6
14	X921	X920
15	X89B	X89A
16	X815	X814
17	X78F	X78E
18	X7Ø9	X7Ø8
19	X683	X682
20	X5FD	X5FC
21	X577	X576
22	X4F1	X4FØ
23	X46B	X46A
14	X3E5	X3E4

X = 1 FOR 8K MACHINE, X = 2 FOR 12K, X = 7 FOR 32K, X = B FOR 48K

3.2.2.4

Unit colour mode

This mode is available for space saving during uniform scans of the picture. A horizontal band of constant colour (or repeated pattern) can be drawn using only one control word and one data word. The data for this mode should be in high speed format.

Using this mode a full screen of data need be no more than 40 bytes of ram.

3.3

Video Interface

The television interface is realized such that a separate adaptor module plugs into the fundamental logic to realize normal Black and White interface, standard colour modules of PAL, SECAM or NTSC and video monitors. Other video interfaces are easily realizable by construction of an adaptor that plugs into the video interface connector of the DAI personal computer.

4.0
PROGRAMMABLE GRAPHICAL SOUND GENERATOR

4.1
Introduction

The sound generator of the DAI Personal Computer has considerable flexibility because every frequency is generated by digital oscillators that yield precise results. Additional random noise generation and digital volume controls complete the system.

4.2
Programmable Oscillators

The Programmable Graphical Sound Generator is realised via three independent programmable oscillators and a random noise generator. Each oscillator is connected as an I/O device to the microprocessor and is programmable to any frequency within the range 30 HZ to 1MHZ. Obviously the higher frequencies are not interesting for audio work but since the three oscillators are added together before modulation of the audio channel of the TV interesting effects can be obtained by beating together various possibilities.
The programmable oscillators are used for sound generation and game paddle interfaces.

4.2.1
Frequency Selection

In order to program a frequency into one of the channels a 16 bit number must be sent to one of the following addresses:

Oscillator Channel	Device Address
1	FC00 or F001
2	FC02 or F003
3	FC04 or F005

Prior to sending a frequency to a channel, address FC∅6 must be loaded with the following 8-bit data words:

1	36 Hex
2	76 Hex
3	B6 Hex

The 16 bit frequency data is sent as two 8-bit transfers to the specified address sending least significant byte first.

4.2.2
Volume Control

The amplitude of the oscillator output as well as that of the noise generator is digitally controllable by writing a control word to the address specified in I/O device allocation section.

4.3
Random Noise

A noise generator circuit is included within the sound generation circuitry. The purpose of this device is to simulate as near as possible white noise for the purpose of complex sound generation and to provide a time random sequence for random number generation. Random events generated by this circuit provide the basis for information input on an I/O port to generate a true random number.

4.4
Frequency Mixing

All sound channels as well as the output of the noise generator are added together before modulation of the audio channel. Channels 1 and 2 and 2 and 3 are added together for left and right stereo output. For the stereo configuration noise is inserted in Channels 1 and 3.

4.5 Frequency Calculator Formula

To output a frequency of nHz from a given oscillator, program it with an integer equal to 2×10^6 divided by n. A special BASIC function (FREQ.) performs this calculation when required.

5.0 INPUT-OUTPUT SECTION

5.1 Introduction

All input-output of the DAI Personal Computer is arranged on a memory mapped basis. I/O is thus directly accessible to BASIC programs, however care is necessary to avoid conflict with the BASIC interpreter activity when using POKE commands.

5.2. Game Paddle Interface

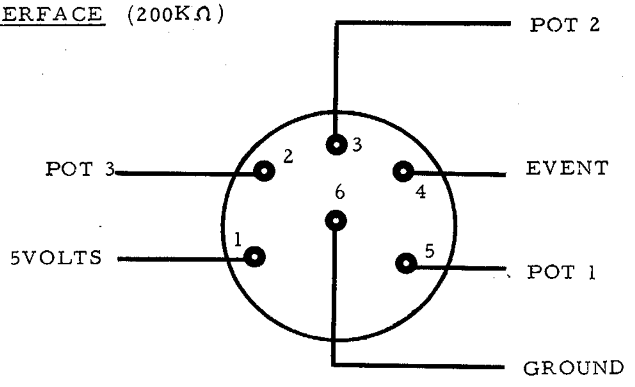
The Personal Computer is equipped with circuitry required to connect two game paddles as input devices. Each paddle contains three variable resistors whose positions are read as values and one on-off event (single contact switch).

The position of any paddle resistor is found by putting its binary address onto the 3 bits in port FD06. Then channel 0 of the sound generator is put into a mode such that it operates as a counter. The read of the positions is triggered by reading location FD01. The value is read out and mapped onto an 8 bit range for a result.

DIN PLUG CONNECTIONS FOR DAI PERSONAL COMPUTER

(6 PINS DIN PLUG 240° VIEWED FROM INSIDE OF THE PLUG OR TO THE COMPUTER PLUG)

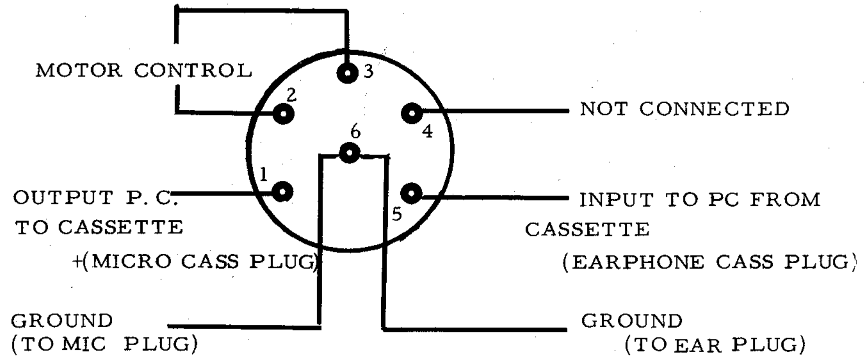
PADDLE INTERFACE (200KΩ)



5. 3
Audio Cassette Interface

The Personal Computer of DAI contains the entire logic and interface circuits needed to connect a low cost audio cassette for the input and output of data and programs.
The Personal Computer input from the cassette should be made via the crystal ear phone outlet or the external speaker outlet. In these cassettes that have no such outputs simply connect the speaker wires to the Personal Computer input.

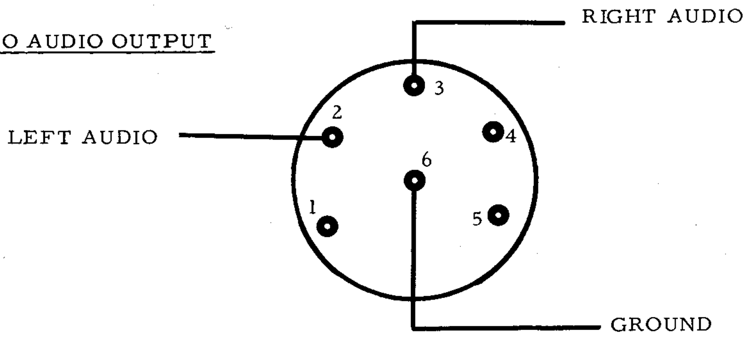
DIN PLUG CONNECTIONS FOR DAI PERSONAL COMPUTER
(6 PINS DIN PLUG 240° VIEWED FROM INSIDE OF THE PLUG OR TO THE COMPUTER PLUG)
CASSETTE RECORDER INTERFACE



5. 4
Stereo Output

The DAI Personal Computer Graphical sound Generator is connectable to the left and right channels of a stereo set. Channels 0 and 1 and channels 2 and 3 are summed to make the left and right channel respectively.

STEREO AUDIO OUTPUT



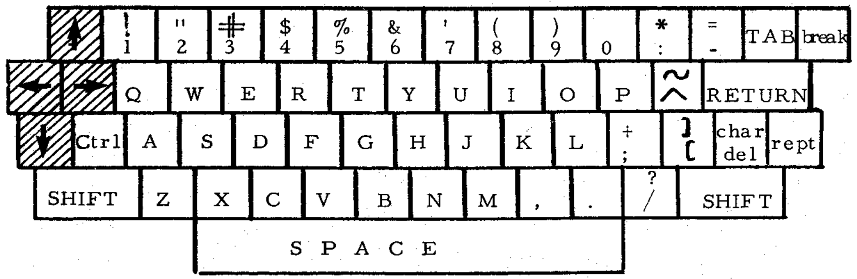
5. 5
Scientific Math Peripheral

As an option for high speed calculations the logic of the DAI Personal Computer supports the Scientific Math Chip of Advanced Micro Devices (9511).
The device is addressed at locations FB00 (data) and FB02 (command and status). The "PAUSE" signal is correctly used to make the CPU wait for data. Note that the SHLD and LHLD instructions are not usable with this device for double byte transfers.

5. 6 ASCII Keyboard

The ASCII keyboard is scanned as a matrix of switches. Encoding, debouncing and roll-over are realized via a software routine.

5. 6. 1 Keyboard Layout



The keys are assigned to rows and columns.

ROWS

Output lines (FF07)

	0	1	2	3	4	5	6
0	0	8	re-turn	H	P	X	↑
1	1	9	A	I	Q	Y	↓
2	2	:	B	J	R	Z	←
3	3	;	C	K	S	[	→
4	4	,	D	L	T	^	Tab
5	5	-	E	M	U	space bar	ctrl
6	6	.	F	N	V	rept	break
7	7	/	G	O	W	char del	shift

COLUMNS

Input lines (FF01)

5. 6. 2 Keyboard Scan Logic

The Personal Computer contains a software keyboard scan and encoder. This can be used by other programs which may use the standard key encoding tables, or supply their own.
All keys are scanned periodically, and action is taken when a key is noticed to have been newly pressed. Alternatively, if the repeat key is pressed, then periodically all currently pressed down keys are acted on. The repeat speed is fixed.

The actual code for the key is obtained from a table. The "shift" system selects which of two possible tables to use. By setting a flag byte the keyboard handler can be made to scan only for the "BREAK" key which obviously takes less time.

On initialisation the alphabetic keys (A - Z) give capital letters if unshifted, and small when shifted. Pressing the "CTRL" key inverts this arrangement to give a "type-writer-like" effect. Successive uses invert each time.

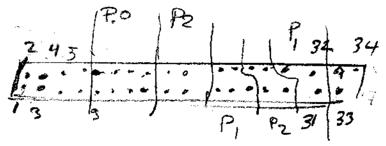
The standard codes returned by each key: see decimal/characters table end of this book.

5.7 DCE-BUS

The DAI Personal Computer provides the possibility of external connection by flat cable of a DCE standard bus. The provided logic drives the bus exactly as a standard DCE Processor with the same addressing and characteristics including reset and interrupt lines. ⁎ The DCE bus can be connected directly to external equipment.
Included in the Personal Computer are routines to communicate with DAI Real-World-Cards. Note that the interface to these routines is different from that in some other DAI software.
Example routines follow in 6.2.15 third page. Note that the internal logic of the routine is subject to changes. Only the interface is guaranteed.

EXAMPLE OF ROUTINE TO DRIVE A PARALLEL PRINTER THROUGH DCE-BUS

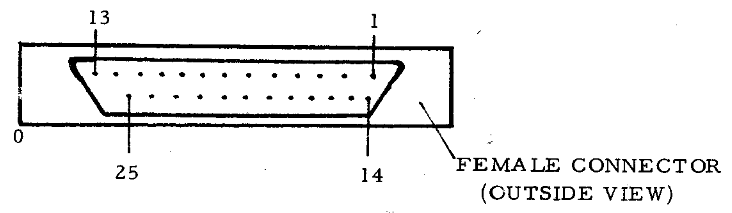
```
10   CLEAR 1000 : REM MUST BE SET FOR YOUR PROGRAM
20   DIM PRI (10)
30   INPUT "TYPE J IF YOU WANT A PRINT" ; A$ : PRINT
40   IF A$<> "J" GOTO 100
50   FOR X = #400 TO 419
55   READ C
60   POKE X,C
65   NEXT X
70   POKE # FEØ3,# AC
75   POKE #2DD, # C3
80   POKE # 2DE, #ØØ
85   POKE #2DF, # 4
90   DATA 229,213,197,17,2,254,6,16,33,1,254
95   DATA 119,43,54,Ø,54,1,26,160,194,11,4,193,209,225,201
100  PRINT CHR$ (12)
110  IF A$<> "J" GOTO 200
120  IF A$ = "J" THEN POKE #131,3 : REM OUTPUT TO DCE-BUS
                                                     ONLY
```



5.7.1 DCE-BUS Pinout

SIGNAL NAME	DESCRIPTION		pin on real-world card	pin on personal comp. card.
P0B0	General Interface PORT 0	Bit 0	24	16
P0B1	data bus	Bit 1	26	14
P0B2		Bit 2		12
P0B3		Bit 3	28	10
P0B4		Bit 4	29	9
P0B5		Bit 5	27	11
P0B6		Bit 6	25	13
P0B7		Bit 7	23	15
P1B0	General Interface PORT 1	Bit 0	12	30
P1B1	CARD SELECT	Bit 1	10	31
P1B2		Bit 2	8	32
P1B3		Bit 3	7	25
P1B4		Bit 4	9	24
P1B5	INTERNAL CARD	Bit 5	11	23
P1B6	ADDRESSING	Bit 6	13	22
P1B7	BUS EXPAND	Bit 7	15	21
P2B0	General Interface PORT 2	Bit 0	18	26
P2B1	WRITE	Bit 1	17	27
P2B2	READ	Bit 2	16	28
P2B3		Bit 3	14	29
P2B4		Bit 4	19	20
P2B5		Bit 5	20	19
P2B6		Bit 6	21	18
P2B7		Bit 7	22	17
EXINTR+	External Interrupt		4	6
IN7+	Parallel input Bit 7(aux. interrupt)		3	5
EXRESET	External Reset (Ground for Reset)		5	7
+12V	+12V DC		2	2
+5V	+5V DC		1	1
-5V	-5V DC		6	3
INTR	INTERRUPT PIN 14 OF CPU 8080		-	33
IN7+			-	34
NOT CONNECTED			-	8
ground				4

PERSONAL COMPUTER RS-232 CONNECTOR:



PIN	FUNCTION	
1	GND	
2	SERIAL OUT	OUTPUT DATA FROM P. C.
3	SERIAL IN	INPUT DATA TO P. C.
4	DATA TERMINAL RDY	INPUT READY HIGH (5V), NOT READY LOW (ØV)
5	+12V ✱	
6	+12V ✱	
7	GND	
8	+12V ✱	
9 ↓ 25	N. C.	

Note: This connector is wired as fo
a terminal and signals to pins 2 and
3 may have to be swapped if it is to
send data to a terminal/printer.

✱ 12V THROUGH 220Ω 1/4W.

5.8
RS232 Interface

The Personal Computer has an RS232 compatible interface giving a serial input line, serial output line and a status line to halt output (DTR). These are available on a CCITT standard connector at the rear of the machine. The DTR signal allows synchronisation of the output with a printer. If unused, then output will be unimpeded.
Interrupts to locations 20 and 28 can be set up for receive and transmit ready. The BASIC interpreter however uses the locations for other purposes.

5.9
I/O Device Address (Allocation Reference)

5.9.1
Master Control Device Address (Hex)

F900 - F9FF	Spare	
FA00 - FAFF	Spare	
FB00/1	Data	Scientific Math Chip
FB02/3	Command	Scientific Math Chip
FC00/1	Channel 0	Graphical Sound Generator
FC02/3	Channel 1	Graphical Sound Generator
FC04/5	Channel 2	Graphical Sound Generator
FC06/7	Command	Graphical Sound Generator
FDXX	See 5.8.2	
FE00/1/2	I/O ports 0/1/2	DCE-BUS
FE03	Command port	DCE-BUS
FFXX	See 5.9.3	

5.9.2
Discrete I/O Device Address (Hex)

ADDRESS	NOTES	IN/OUT	BIT ALLOCATION	
FD00	1	IN	0	-
			1	-
			2	Page Signal
			3	Serial output ready
			4	Right paddle button (1 = closed)
			5	Left paddle button (1 = closed)
			6	Random data
			7	Cassette input
FD01	3	IN		Single pulse used to trigger paddle timer circuit.
FD04	2	OUT	0, 1, 2, 3	Volume, oscillator Channel 1
			4, 5, 6, 7	Volume, oscillator Channel 2
FD05	2	OUT	0, 1, 2, 3	Volume, oscillator Channel 3
			4, 5, 6, 7	Volume, random noise

Cont. . . . .

ADDRESS	NOTE	IN/OUT	BIT ALLOCATION	
FD06	3	OUT	0	Cassette data out
			0, 1, 2	Paddle channel select code
			3	Paddle enable bit
			4	Cassette motor 1 control (0 = run)
			5	Cassette motor 2 control (0 = run)
			6,7	ROM bank switch

Notes:

1 User may read from or write to any of these addresses at will. No harm can result.

2 Reading from these locations does nothing.
Writing to them will modify the appropriate volume settings, but if the BASIC system accesses the channel the effect may be lost, as it has an internal memory of its own last set value.

3 These locations should not be written into.

5.9.3 Serial I/O, timer & interrupt control

The detail given here is sufficient to allow use of the serial I/O. All these facilities are given by one LSI component, and the BASIC interpreter uses many of the facilities itself. So care must be taken not to disturb the normal running of the system.

ADDRESS	NOTE	FUNCTION
FF00	1	Serial input buffer Contains the last character received on the RS232 interface.
FF01	1	Keyboard input port Bottom 7 bits are data input from the keyboard. Bit 7 is the IN7 line from the DCE-BUS and is attached to the page blanking signal for the TV.
FF02	2	Interrupt address register
FF03	1	Status register Bit allocations: 7,6,5 Not useful 4 Transmit buffer empty Set if RS232 output ready to accept another character. 3 Receive buffer loaded Set if a character has been received 2 Overrun Set if a character has been received but not taken by the CPU. 1 Frame error Set by a "BREAK" on RS232 input
FF04	2	Command register
FF05	3	RS232 Communications rate register

Send (Hex)	for	
1/81	110 baud	2/1 stop bits
2/82	150 "	"
4/84	300 "	"

Send (Hex)	for	
8/88	1200 "	"
10/90	2400 "	"
20/A0	4800 "	"
40/C0	9600 "	"

Underlined is usual one to use.
Other combinations not useful

ADDRESS	NOTE	FUNCTION
FF06	3	Serial output Write byte to this location to send it on RS232 output. Use only when address FF∅3 bit 4 HIGH
FF07	4	Keyboard output port Data output to scan keyboard. Not useful to user.
FF08	2	Interrupt Mask register
FF09 FF0A FF0B FF0C FF0D	2	Timer addresses

Notes:

1 May be read but not written to by user

2 Should not be accessed by user

3 May be written but not read by user

4 May not be read, writing is harmless and useless ! System keyboard scanner will overwrite user data.