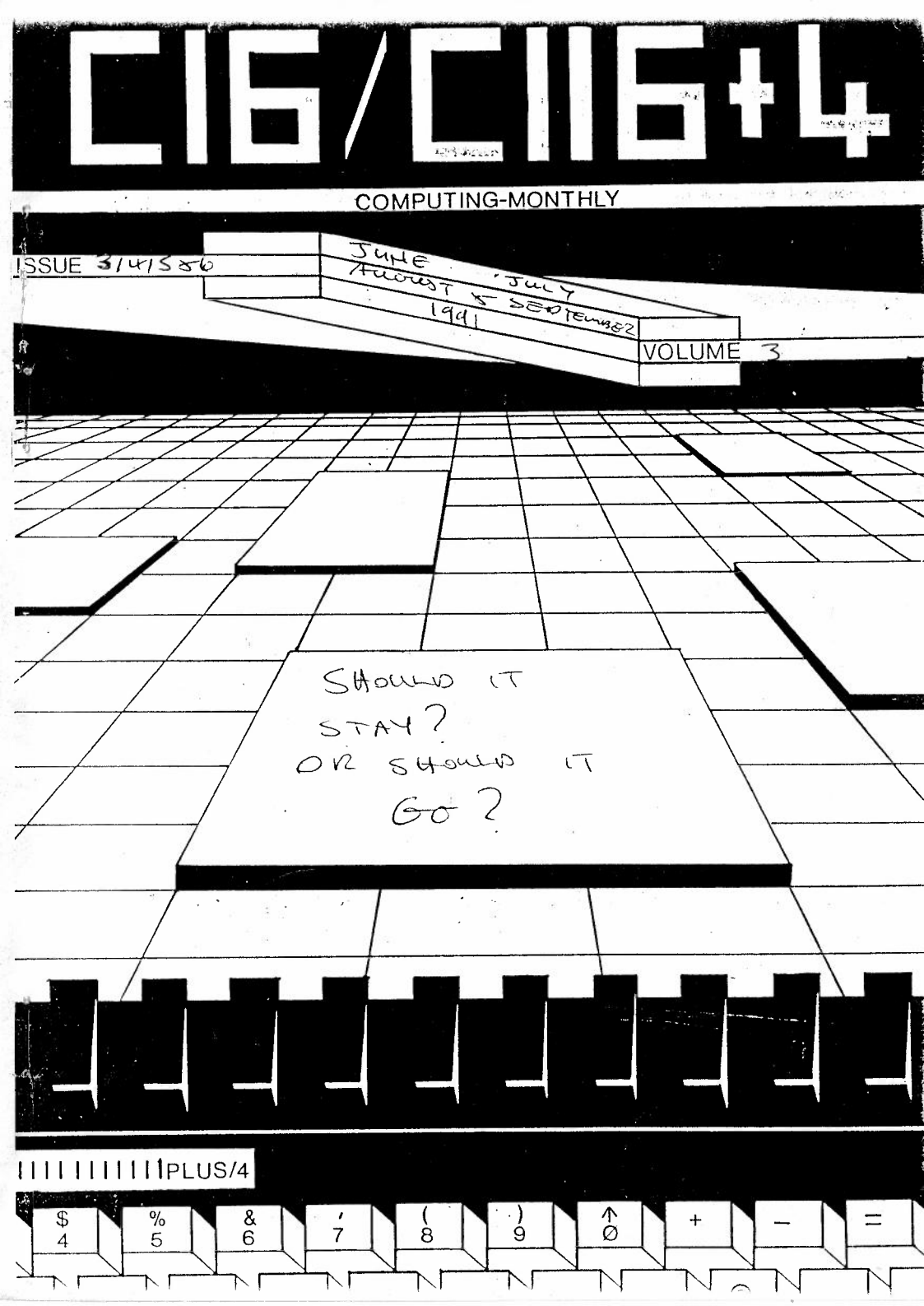
C16/C116+4

COMPUTING-MONTHLY

ISSUE 3/4/5 86

JUNE / JULY
AUGUST & SEPTEMBER
1991

VOLUME 3



Ed's Bits

Hello
And welcome to the June July August & September Issues, sorry for the delay,, AGAIN!

Pages 2 to 26, include all you could ever want to know about the hardware of the +4.

Pages 27 to 36, some progs from Andy Brett, how gave me the above!

Pages 37 to 48, program con't from Peter Crack.

Pages 49 to 57 Letters.

Pages 58 to 63 reviews.

Pages 64 Letter.

Pages 65 Review.

Pages 66 to 85 RS232 info for use by anybody that knows how to use it!!

Pages 86 Order form for Aug/Sept/Oct & Nov 1991 Mags.

You will see that pages 50 to 53 have some german info. This is for 2 german disk magazines for the +4 which I get by exhange each time there due. I suggest that it is an investment to subscribe to them (even if you can't speak german, like me, because the programs that are given away on the disk are of a very high standard, I just wish I could produce the same quality. I understand that the few german members that are members of this group saw C16/C116/+4 Bi Monthly Mentioned in each of these mags, so if you would like to see good new games & demos & useful utils then write to the people at the addresses given, they are more than willing to help, and would like to here from you.

Well I cut it short, take care everybody, bye!!!

Roy Robinson.

COMMODORE

C PLUS 4

CONTENTS

PLUS 4 PRODUCT SPECIFICATION

MEMORY

64K RAM. 60K RAM User accessible for BASIC programs.

ROM

32K ROM Standard (includes operating system and BASIC interpreter) with 32K additional ROM containing the built-in productivity software.

MICROPROCESSOR

7501 Microprocessor — .89 or 1.76 MHz clock.

DISPLAY

40 Columns x 25 lines of text.

COLORS

128 Colors (16 colors; 8 luminance levels).

CHARACTERS

Upper & lower case letters, numerals and symbols. Reverse and flashing charachters. All PET graphic characters.

DISPLAY MODES

Text characters. High resolution graphics. Split screen text/high resolution graphics. Muiticolor graphics.

RESOLUTION

320 x 200 Pixels

SOUND

2 Tone generators or 1 Tone and 1 white noise generator.

VOLUME

8 Volume levels

KEYBOARD

Full size typewriter style design

KEYS

67 Keys total. 4 Cursor control keys. 4 Programmed (reprogrammable) function keys (up to 8 user defined functions possible). Color control keys. HELP key. Upper and lower case character set. Graphics character set.

INPUTS/OUTPUTS

PLUS/4 MODEM (User) port. Serial port. ROM cartridge and parallel disk drive port. 2 Joystick ports. C1531 Cassette drive interface port. RF Output-channel 3 or 4. Video output-composite/ chrominance/luminance. Audio input/output. Power supply input.

PLUS 4 PRODUCT SPECIFICATION (Continued)

FEATURES

Built-in extended BASIC 3.5 — over 75 commands. Built-in Machine Language monitor — over 12 commands. Built-in graphics and sound commands. Screen window capability. Reset button (Warm start). Built-in integrated productivity software.

PERIPHERALS

C1551 Fast Disk drive, C1531 Datasette, MPS 802 Dot matrix printer, MPS 803 Dot matrix printer, DPS 1101 Daisy wheel printer, C1802 color monitor.

OTHER PERIPHERALS

C1541 Disk drive, MPS 801 Dot matrix printer, C1702 color monitor.

PLUS 4 OVERVIEW

The Plus 4 system is based on the 7501 microprocessor, an HMOS version of the 6510. Video processing is achieved by the 7360 TED chip. 64K bytes of dynamic RAM are accomplished by 8 (64K x 1) I.C.'s. (See page). The system program is contained in 2 (16K x 8) ROMs. The system supports up to 128K x 8 of ROM banked in 16K sections. By software control, through the 7360, ROM can be completely banked out and RAM banked in for a true 64K of RAM (minus 256 byte pages), allowing 60,671 bytes available for BASIC.

Keyboard and joystick scanning are accomplished by outputting the row data on the data bus while addressing a particular register in the TED chip. This will in turn cause the TED chip to latch the column information.

A standard serial port supports serial bus peripherals such as the 1541 disk drive and the various serial printers. A cassette port is provided and the expansion port supports ROM cartrdiges. TTL serial ASCII is intended to drive an RS-232 adapter.

PARTS LIST
PLUS/4

TOP CASE ASSY

Top Case	C 251453-01
Keyboard, 67 Key, KKR-I	C 251501-01
Nameplate	C 251655-01
Shield Clip, R	C 251855-01
Shield Clip, F	C 251856-01

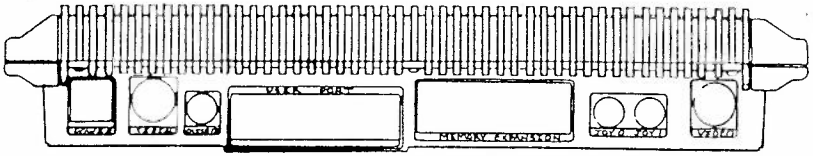
BOTTOM CASE ASSY

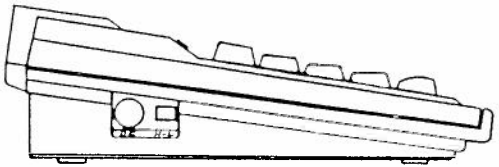
Bottom Case	C 251454-01
Foot, Self-Adhesive	C 950157-04
Paper Shield	C 310156-01
Shield Chip	C 310199-01
Shield Plate	C 310197-01
Insulation Sheet	C 310198-01

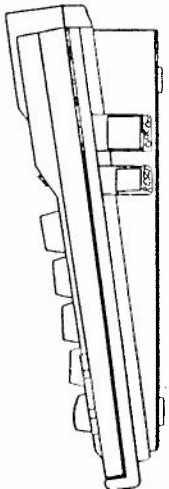
ACCESSORIES

Users Manual	C 310196-01
Power Supply	C 310157
RF Cable	C 326189-02
Switch Box	C 904778-01

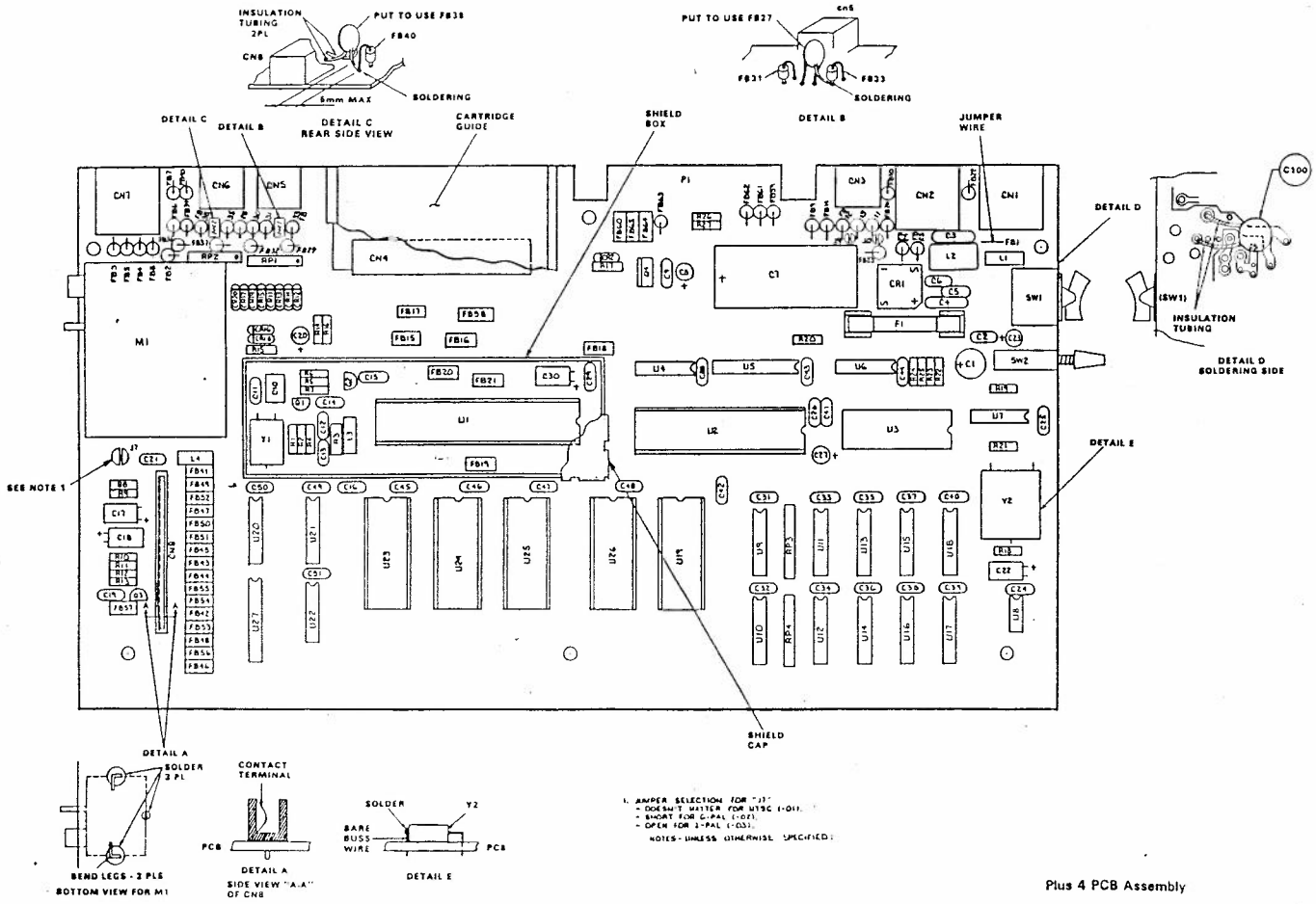
C — Commodore Stock Part







Plus 4 Casework Identification



Plus 4 PCB Assembly

PARTS LIST — PLUS/4 PCB ASSEMBLY #310163-01

PLEASE NOTE: Commodore part numbers are provided for reference only and do not indicate the availability of parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Approved cross-references for TTL chips, Transistors, etc. will be available in manual form through the Service Department in November of 1984. Unique or non-standard parts will be stocked by Commodore and are indicated on the parts list by a "C".

INTEGRATED CIRCUITS		
U1	7360 VLSI, Text Dislay (TED)	
	Sub:	C 251535-01
	8360	C 251535-02
U2	7501 Custom Microprocessor	C 251536-01
U3	6551A (Synertek)	901895-02
U4	74LS08	901521-03
U5	6529B Single Port Interface	C 251640-03
U6	74LS04	901521-02
U7	7406	901522-06
U8	555	901523-01
U9-10	74LS257	901521-57
U11-18	4164-2 D-RAM	901505-01
U19	7700-010 PLA	C 251641-02
U20	74LS139	901521-18
U21	74LS175	901521-34
U22	74LS27	901521-22
U23	23128 ROM TED Basic	C 318006-01
U24	23128 ROM TED Kernal	C 318005-04
U25	23128 FUNCTION ROM, 3+1 LOW	C 317053-01
U26	23128 FUNCTION ROM, 3+HIGH	C 317054-01
U27	6529B Single Port Interface	C 251640-03

TRANSISTORS		
Q1-Q3	2SC 1815	902693-01
Q4	2SD 880	902694-01 Sub:
	Tip 29A	902653-01 Sub:
	2SD 1266	902694-04

DIODES		
CR1	Bridge Rectifiers S2VB10 Sindengen	215026-01 Sub:
	Bridge Recitifers DBA20B Sanyo	251026-02 Sub:

DIODES (Continued)		
CR1 (cont.)	Bridge Rectifiers DBA20C Sanyo	251026-03
CR2	Diode, Zener RD 6.8 EB	900927-01
CR11-20	Diode, IN 914 Sub:	900850-16
	Diode, IN 4148 Taping	251819-21 Sub:
	Diode, IN 4148	900850-01

RESISTORS — All values are in ohms-1/4 W 5% unless noted otherwise.

R1	4.7K	R14	240
R2	10K	R15	250
R3	470K	R16	100K
R4	220K	R17	1.5K
R5	18K	R18	47K
R6	1.5K	R19	100K
R7	470K	R20	3K
R8	100K	R21	1K
R9	1K	R22	1K
R10	1K	R23	1K
R11	12K	R24	1K
R12	10K	R25	1K
R13	1K		

RESISTOR PACK		
RP1, 2	3.3K, 6 PIN	902441-29
RP3, 4	68, 8 PIN 4 ISOLATED	326149-06

CAPACITORS			
C1	Elect	0.1 μF 25V	900100-40
C2	Ceramic	0.1 μF 25V	251075-06
C3	Film	0.22 μF 100V	900150-11
C4	Film	0.22 μF 100V	900150-11
C5-C6	Ceramic	0.22 μF 50V	900022-01
C7	Elect	2200 μF 16V	900101-33
C8	Elect	10 μF 16V	900100-25
C9	Ceramic	0.1 μF 25V	251075-06
C10	Trimmer	40 pF	251029-02
C11	Ceramic	22 pF 50V	251070-14

PLEASE NOTE I BELIEVE U24 SHOULD READ C 318004_05

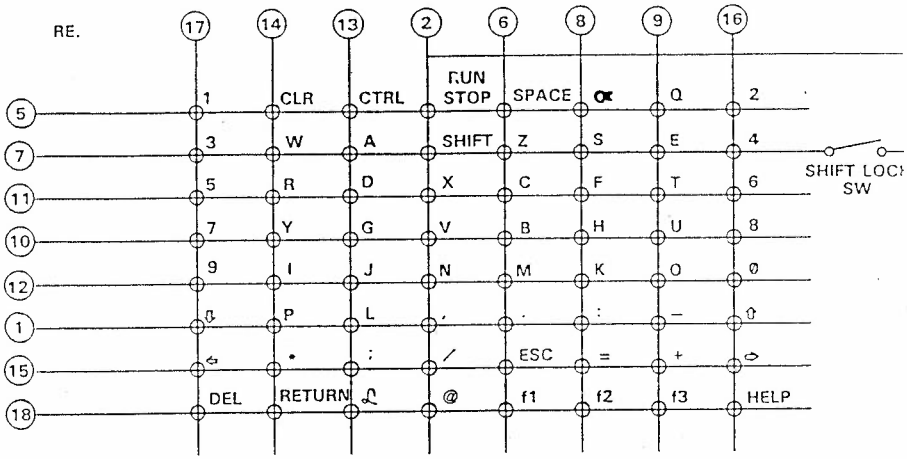
PARTS LIST — PLUS/4
PCB ASSEMBLY #310163-01 (Continued)

CAPACITORS (Continued)		
C12	Ceramic 220 pF 50V	Sub: 251071-26 Sub:
	Ceramic 220 pF 50V	900463-08
C13	Ceramic 150 pF 50V	251071-24 Sub:
	Ceramic 150 pF 50V	900462-41
C14	Ceramic 0.1 µF 25V	251075-01
C15-C16	Ceramic 0.1 µF 25V	251075-06
C17-C18	Ceramic 10 µF 16V	900100-25
C19	Ceramic 0.01 µF 25V	251075-01
C20	Elect 10 µF 16V	900100-25
C21	Ceramic 0.1 µF 25V	251075-06
C22	Elect 10 µF 16V	900100-25
C23	Elect 1 µF 16V	900100-16
C24-C26	Ceramic 0.1 µF 25V	251075-06
C27	Elect 10 µF 16V	900100-25
C28-C29	Ceramic 0.1 µF 25V	251075-06
C30	Elect 10 µF 16V	900100-25
C31-C32	Ceramic 0.1 µF 25V	251075-06
C33-C40	Ceramic 0.22 µF 25V	251075-07 Sub:
	Ceramic 0.22 µF 50V	900022-01
C41-C51	Ceramic 0.1 µF 50V	900020-06 Sub:
	Ceramic 0.1 µF 50V	9000010-01
C100	Ceramic 0.1 µF 50V	900010-20

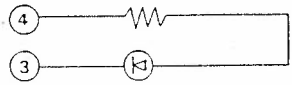
MISCELLANEOUS		
Y1	Crystal 14.31818 MHZ	251081-01 Sub:
	Crystal 14.31818 MHZ	251081-02
Y2	Crystal 1.8432 MHZ	900555-02
SW1	Switch, Rocker (PC Mount)	C 251587-01
SW2	Switch, Push Button	C 251260-01
M1	RF Modulator	C 251844-01 Sub:
	RF Modulator	251311-01

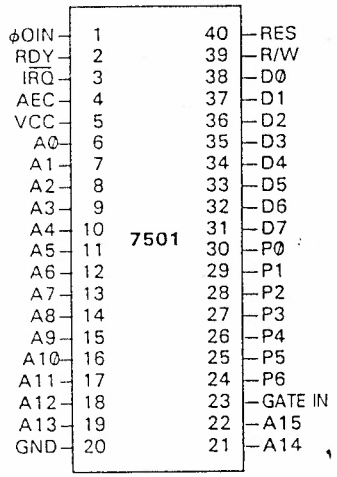
MISCELLANEOUS (Continued)		
FB2-14	Ferrite bead	325563-01
FB15-21	Ferrite bead	903025-01
FB22-26, FB28-38, FB40	Ferrite bead	325563-01
FB41-58	Ferrite bead	903025-01
FB59	Ferrite bead	325563-01
FB60	Ferrite bead	903025-01
FB61-63	Ferrite bead	325563-01
FB64-65	Ferrite bead	903025-01
EM1,2	EMI Filter	251842-01
CN1	Connector 4 PIN (power supply)	C 251614-01
CN2	Connector 6 PIN DIN (serial bus)	C 903361-01
CN3	Connector 7 PIN MINI DIN (cassette)	C 251616-01
CN4	Connector 50 PIN Female Edge (exoab)	C 251630-01
CN5-6	Connector 8 PIN MINI DIN (joy 1 & 2)	C 251259-01
CN7	Connector 8 PIN DIN (audio/video)	325573-01
CN8	Connector 18 PIN (keyboard)	C 251841-01
L1	Noise Filter	251264-01
L2	Line Filter	906127-01
	Sub:	251701-01
	Sub:	
L3,L3	Coil Inductor 1.2 uHpt	901152-01
	Sub:	325570-01
F1	Fuse 250V 1.5A	903556-18
	Fuse Clip	906102-01
	Cartridge Guide	310171-01
	Shield Box	C 310172-01
	Shield Cap	C 310173-01

C — Commodore Stock Part

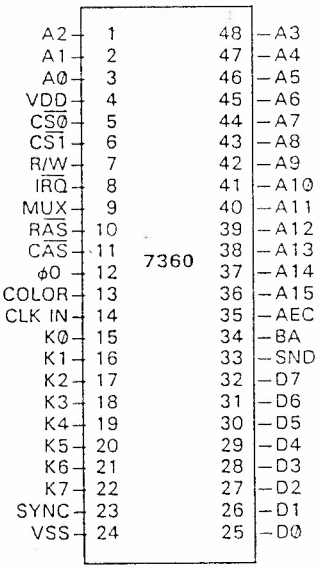


KEYBOARD MATRIX

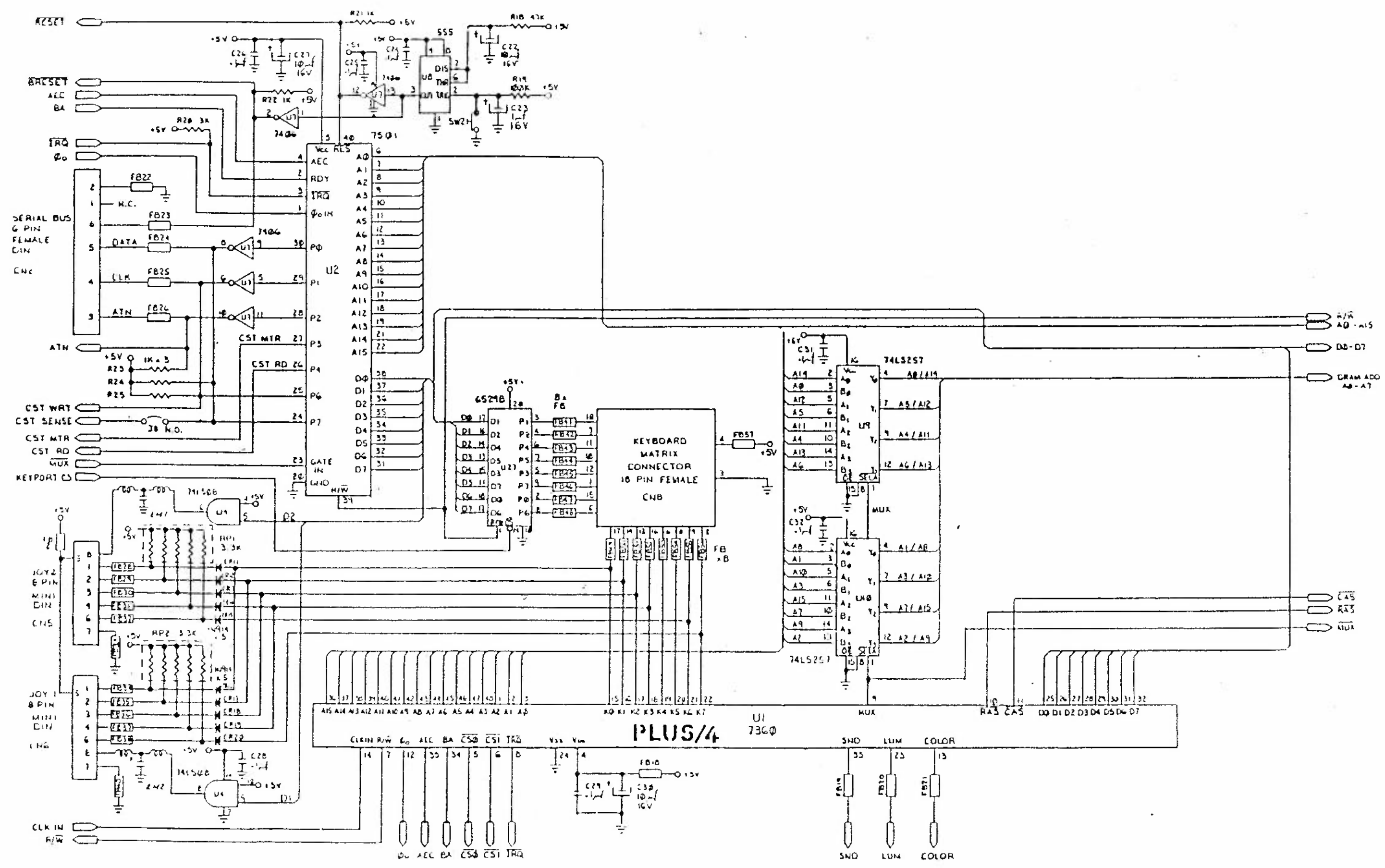




PIN ASSIGNMENT
U2-251536-01
CUSTOM MICROPROCESSOR



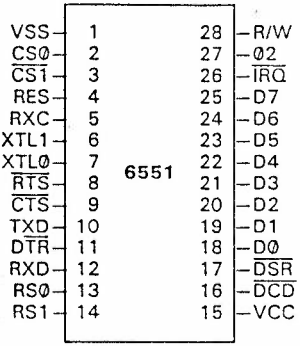
PIN ASSIGNMENT
U1-251535-01
VLSI, TEXT DISPLAY
(TED)



PLUS/4
U1 7360
U2 7501
555
6529B
74LS257
KEYBOARD MATRIX CONNECTOR 18 PIN FEMALE CN8
SERIAL BUS 6 PIN FEMALE DIN CN4
JOY2 8 PIN MINI DIN CN5
JOY1 8 PIN MINI DIN CN6
RESET
BRESET
AEC
BA
IRQ
CST WRT
CST SENSE
CST MTR
CST RD
MUX
KEYPORT CS
ATN
CLK IN
R/W
R/W A0-A15
D0-D7
DRAM ADD A0-A7
CAS
RAS
SND
LUM
COLOR

M_1 SCHEMATIC ON PAGE 11

PIN CONFIGURATION



U3-901895-02
ACIA

SYNERTEK	SYP6551A	2 MHZ

TRANSMIT/RECIEVE CHARACTERISTICS

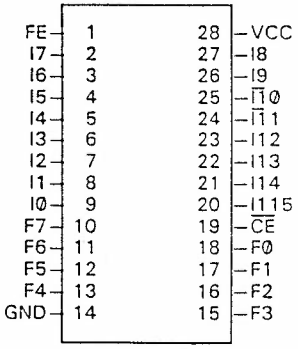
CHARACTERTISTICS	SYM	−02 2 MHZ		UNIT
		MIN	MAX	
TRANSMIT/RECEIVE CLOCK RATE	t_{CCY}	400	—	ns
TRANSMIT/RECEIVE CLOCK HIGH TIME	t_{CH}	175	—	ns
TRANSMIT/RECEIVE CLOCK LOW TIME	t_{CL}	175	—	ns
XTL1 TO TXD PROPAGATION DELAY	t_{DD}	—	500	ns
RTS PROPAGATION DELAY	t_{DLY}	—	500	ns
IRQ PROPAGATION DELAY (CLEAR)	t_{IRQ}	—	500	ns

(tr, tf = 10 to 30 ns)

*The Baud Rate with External Clocking is:

$$\text{BAUD RATE} \quad \frac{1}{16 \times T_{CCY}}$$

PIN CONFIGURATION

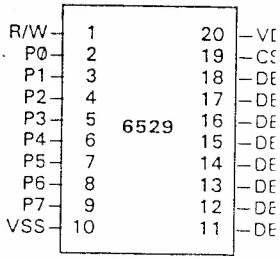


U19-251641-02
PLA

CS	R/W	D0-D7
L	L	DATA BUS TO PORT
L	H	PORT TO DATA BUS
H	X	ISOLATION

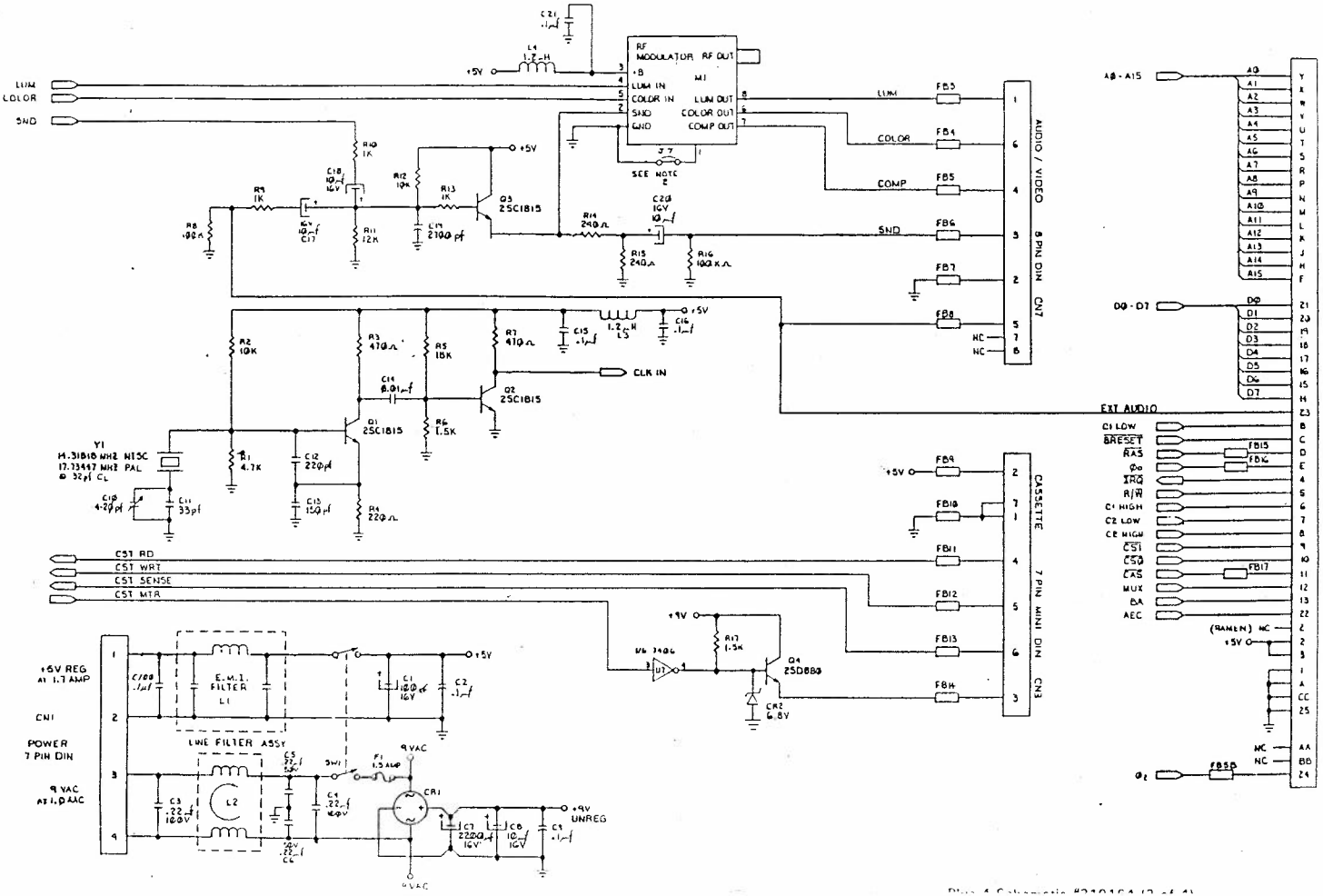
L = LOW LEVEL
H = HIGH LEVEL
X = IRREVELANT

PIN CONFIGURATION



MOS	6529B	3 MHz

U5/U27-251640-03
SINGLE PORT INTERFACE



Plus 4 Schematic #310164 (2 of 4)

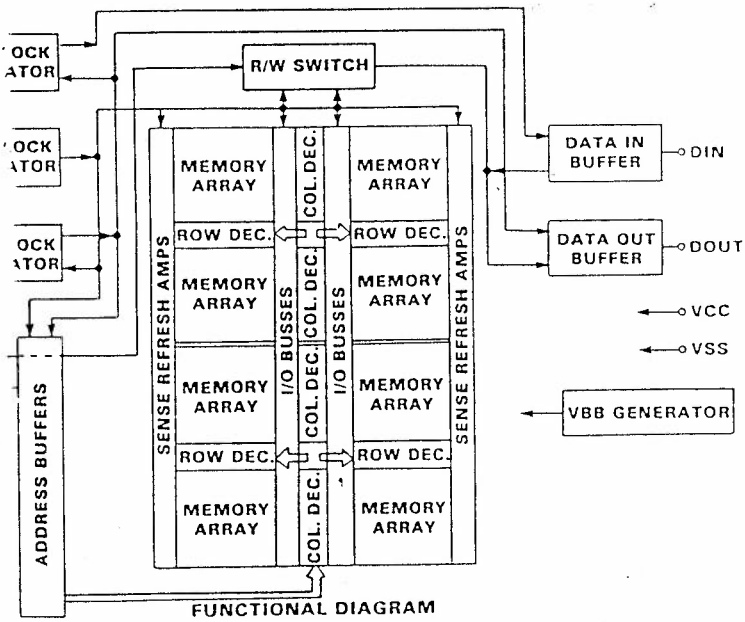
PINOUTS ON PAGE 8

N
RATION

16	VSS
15	CAS
14	DOUT
13	A6
12	A3
11	A4
10	A5
9	A7

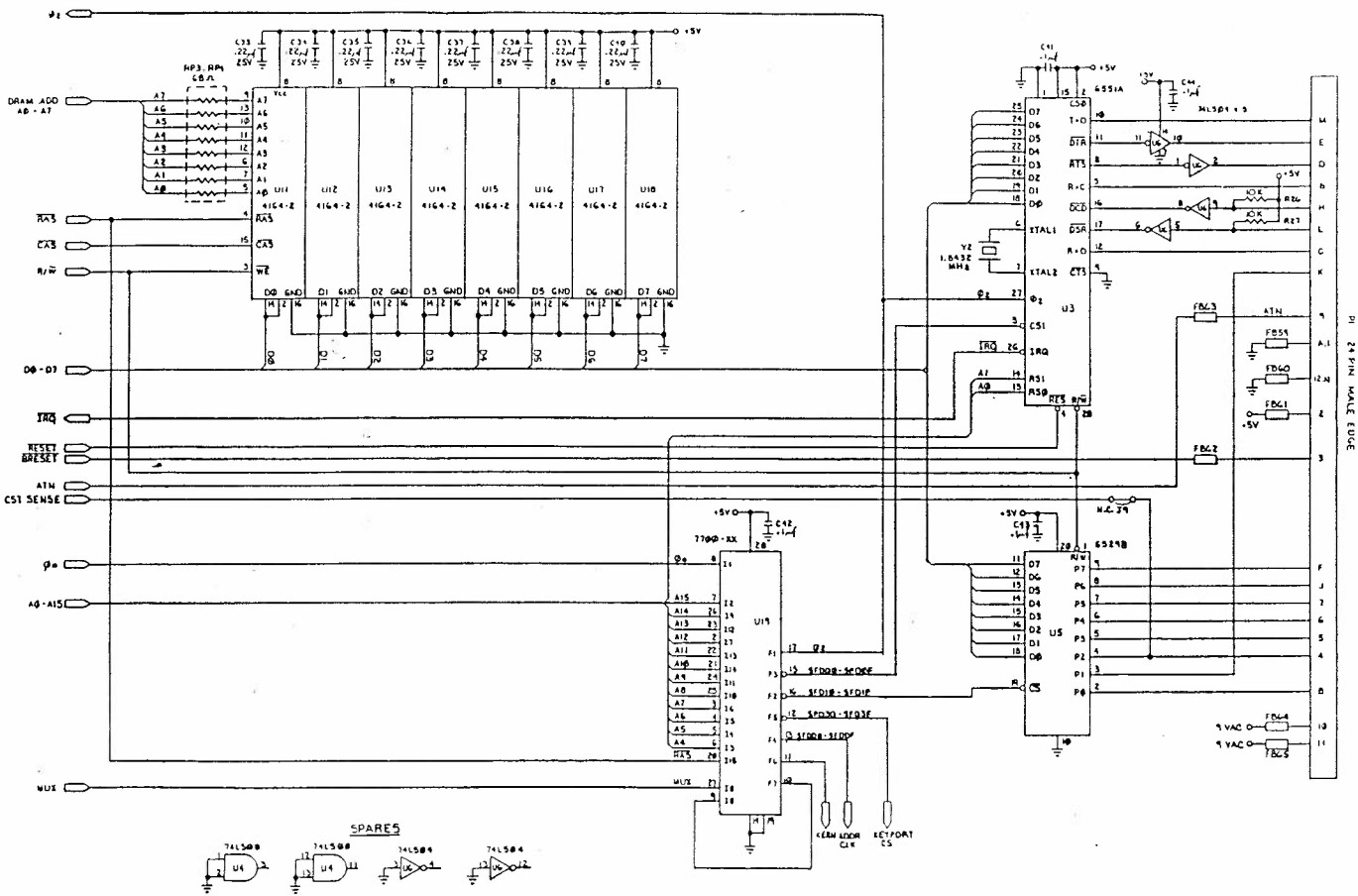
18
MIC RAM
5-01

COMMODORE PART NUMBER	APPROVED SOURCE OF SUPPLY	VENDOR PART NUMBER	ACCESS TIME (ns)	CYCLES (ns)	POWER	
					ACTIVE (MW)	STAN (MAX)
901505-01	HITACHI	HM4864-3	200	335	330	20
901505-01	NEC	μPD4164-2	200	375	250	28
901505-01	MITSUBISHI	M5K416NS-20	200	330	275	28
901505-01	MOSTEK	MK4564N-20	200	345	300	22
901505-01	OKI	MSM3764-20	200	330	248	23
901605-01	MICRON TECHNOLOGY	MT4264-3	200	385	300	30
901505-01	HITACHI	HM4864P-3	200	335	330	20
901505-01	MATSUSHITA (PANASONIC)	MN4164P-20	200	330	275	27
901505-01	SIEMENS	HYB4164-3	200	330	150	20
901505-01	SHARP	LH2164-Z1	200	330	248	28
901505-01	HITACHI	HM4864AP-3	200	330	242	20
901505-01	TOSHIBA	TMM4164AP-20	200	330	275	22

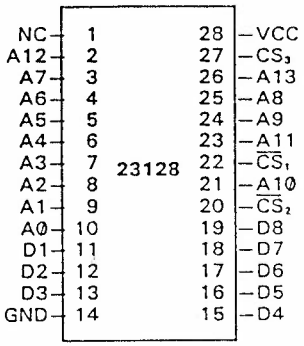


FUNCTIONAL DIAGRAM

16

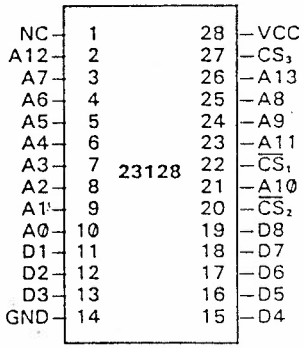

SPARES
74LS00
74LS04
Plus 4 Schematic #310164 (3 of 4)

PIN CONFIGURATION

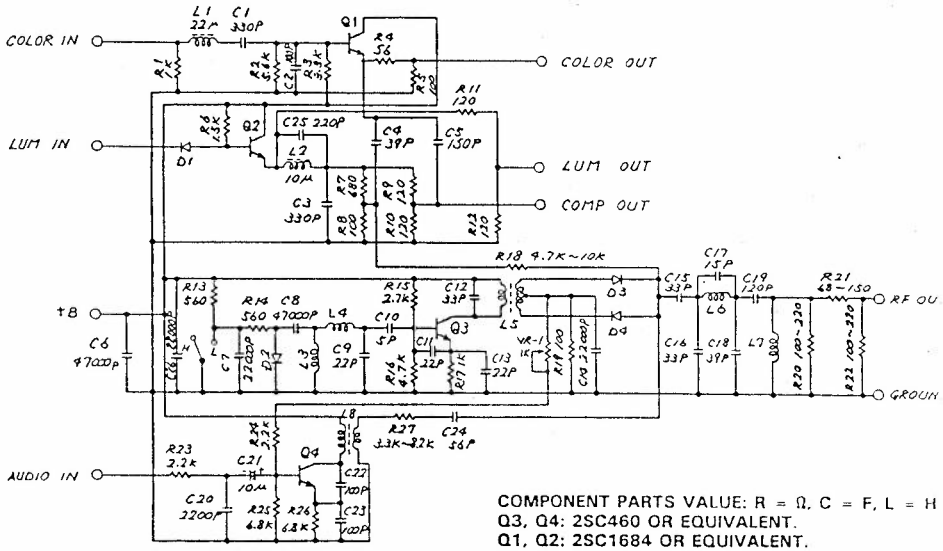


U23-318006-01
ROM — BASIC

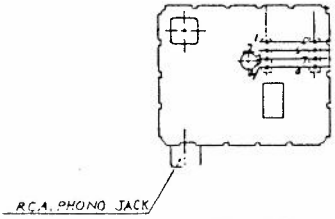
PIN CONFIGURATION



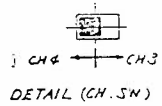
U24-318005-04
ROM — KERNAL



COMPONENT PARTS VALUE: R = Ω, C = F, L = H
Q3, Q4: 2SC460 OR EQUIVALENT.
Q1, Q2: 2SC1684 OR EQUIVALENT.
D3, D4: 1SS198 OR EQUIVALENT.
D2 : 1SS110 OR EQUIVALENT.
D1 : 1SS119 OR EQUIVALENT.



TOP VIEW

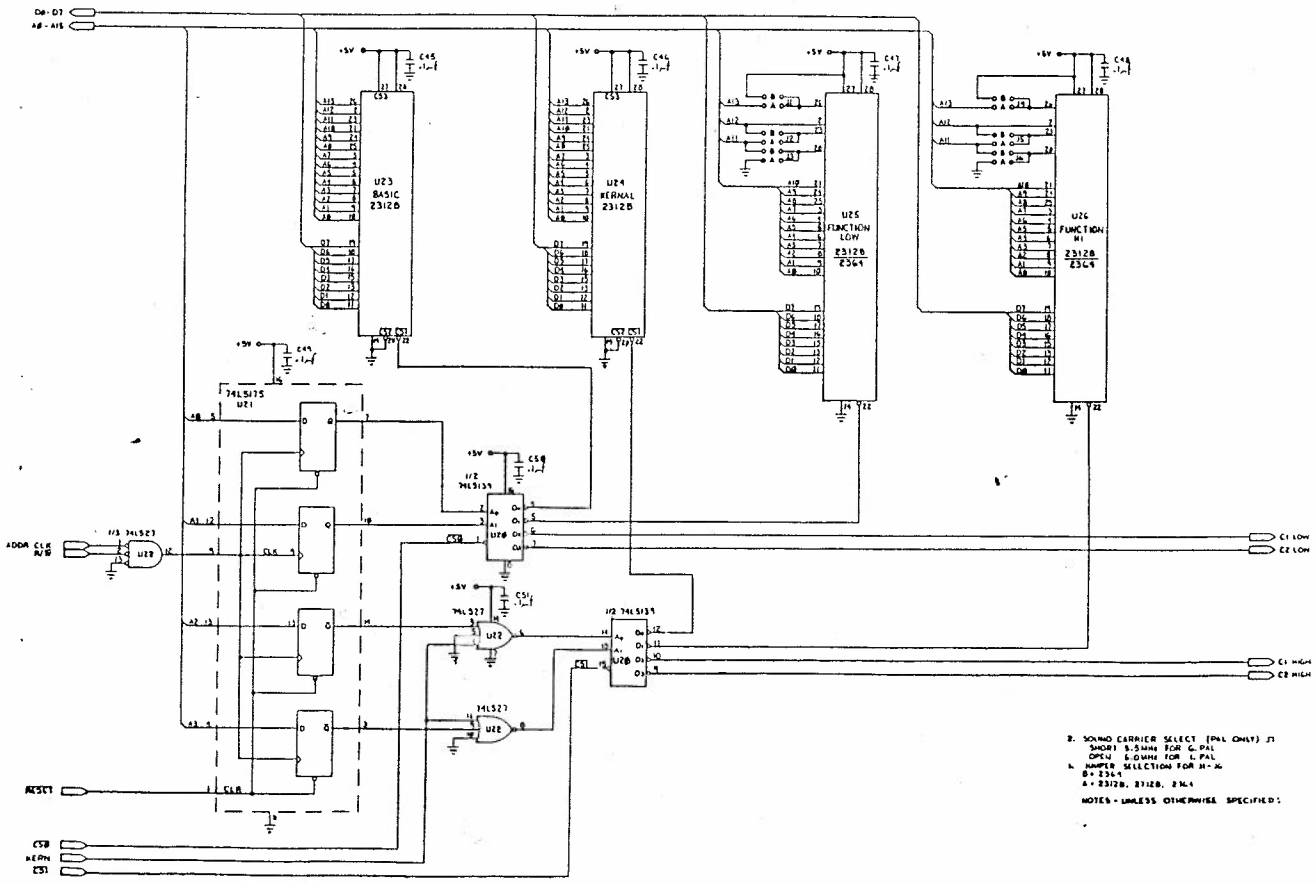


DETAIL (CH. SW)

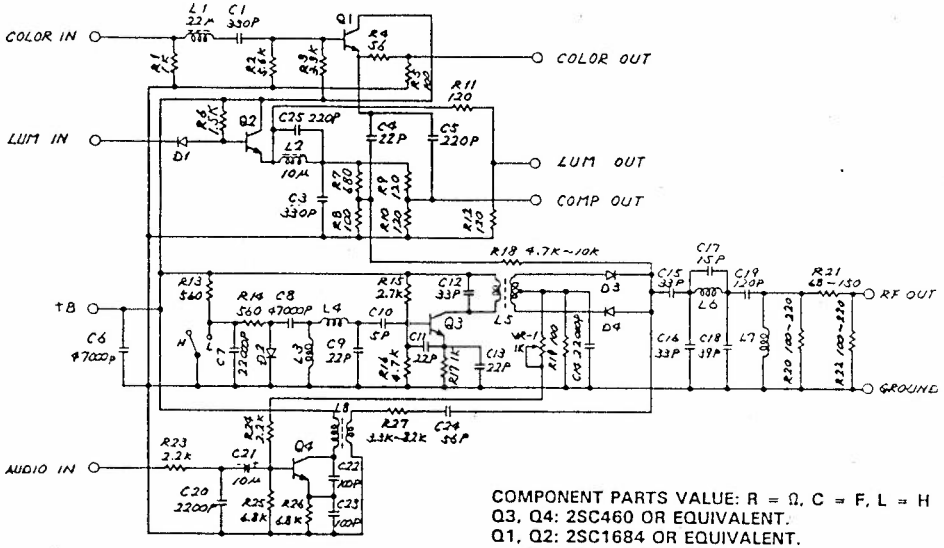
REAR VIEW

NO.	TERMINALS
1	N.C.
2	AUDIO SIG. INPUT
3	+8 (+5V)
4	SYNC + LUM. SIG. INPUT
5	COLOR SIG. INPUT
6	COLOR SIG. OUTPUT
7	COMPO. SIG. OUTPUT
8	SYNC + LUM. SIG. OUTPUT
9	RF OUTPUT
10	CHANNEL SELECT SW.

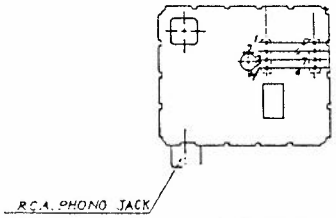
RF Modulator Layout and Schematic #251844



Plus 4 Schematic #310164 (4 of 4)

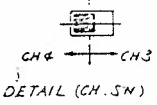


COMPONENT PARTS VALUE: R = Ω, C = F, L = H
Q3, Q4: 2SC460 OR EQUIVALENT.
Q1, Q2: 2SC1684 OR EQUIVALENT.
D3, D4: 1SS198 OR EQUIVALENT.
D2 : 1SS110 OR EQUIVALENT.
D1 : 1SS119 OR EQUIVALENT.



TOP VIEW

REAR VIEW



NO.	TERMINALS
1	N.C.
2	AUDIO SIG. INPUT
3	+B (+5V)
4	SYNC + LUM. SIG. INPUT
5	COLOR SIG. INPUT
6	COLOR SIG. OUTPUT
7	COMPO. SIG. OUTPUT
8	SYNC + LUM. SIG. OUTPUT
9	RF OUTPUT
10	CHANNEL SELECT SW.

RF Modulator Layout and Schematic #251311

Issue 7, 1984 : Computer 4

commodore TECHTOPICS

Model: C16, PLUS 4

LINE DEFINITIONS

A0 to A15	Address Bit 0 to 15
AEC	Address Enable Control
ATN	Attention
BA	Bus Available
BRESET	Buffered System Reset
C1 HIGH, C1 LOW	External Cartridge Chip Select
C2 HIGH, C2 LOW	"
CAS	Dynamic RAM Column Address Strobe
CLK IN	Master Clock (Single Phase, 14.31818 MHz)
COLOR	Chroma Output
COMP	Composite Chroma and Luma
CE	Chip Enable
CS	Chip Select
CS0	Low ROM Chip Select
CS1	High ROM Chip Select
CST MTR	Cassette Motor Control
CST RD	Cassette Read
CST SENSE	Cassette Sensor
CST WRT	Cassette Write
CTS	Clear To Send
DB0 to DB7	Data Bit 0 to 7
DCD	Data Carrier Detect
DRAM	Dynamic RAM
DRAM ADD	Dynamic RAM Address
DSR	Data Set Ready
DTR	Data Terminal Ready
EXT AUDIO	External Audio Input
GATE IN	R/W GATE
IRQ	Interupt Request
K0 to K7	Keyboard Latch 0 to 7
KERN	Kernal ROM Control Line
LUM	Composite Sync and Luminence
MUX	Address Multiplex Control
P0 to P7	Port Bit 0 to 7
RAS	Dynamic RAM Row Address Strobe
RESET	System Reset
RxC	Receive Clock
RxD	Receive Data
R/W	Read/Write Line
RTS	Request To Send
SND	Sound Line
TED	Text Display
TxD	Transmit Data
ϕ 0	System Clock (Varies between 1 and 2 MHz)
ϕ 2	Artificial ϕ 2, Address Valid Rising Edge, Data Valid Falling Edge

Issue 6, 1984 : Computer 3

commodore TECHTOPICS

Model: PLUS 4

Troubleshooting Aides

NOTE: Visual inspection is critical in this unit!
The upright position of many of the components used on the board can create problems. It is possible for them to be shorted to the shield or to each other. Make sure they are evenly spaced and do not contact the shield.

Areas of the PCB particularly vulnerable to this problem are:

- Bottom right corner - caps
- Bottom center - J1-J6
- Bottom left - Q3 shorted to FB57
- Top left - ferrite beads
- Center - Twisted caps (just outside of RF can)

1) NO VIDEO - Absolutely no video on screen

A) Check for 5 volts

O.K. ↓

If not: 1) Check fuse
2) Check for twisted or bent caps (5 V. short to ground)
3) Check L1

B) Check for oscillation at pin 14 of U1

O.K. ↓

If not: 1) Check for good connection at pin 14 of U1
2) Check for good connection at R1 thru R7

C) Check for LUM signal at pin 23 of U1, at pin 4 of the modulator, at pin 8 of the modulator, at FB3 and at pin1 of CN7

O.K. ↓

If not: 1) Check for LUM signal shorted to ground
2) Check for open traces
3) Check modulator

D) Check for reset

O.K. ↓

E) Check for control signals:

Signal	I.C.	Pin	Signal	I.C.	Pin
AEC	U1	35	R/W	U1	7
AEC	U2	4	0	U1	12
CAS	U1	11	MUX	U1	9
CS1	U1	6	IRQ	U1	8
CS1	U20	15	RDY	U2	2
CS0	U1	5	BA	U1	34
CS0	U20	1	RAS	U1	10

2) BAD VIDEO - Scrolling lines on screen - Random blocks on screen - Blurred display

A) Check J1-J6 for shorts to ground or each other
O.K.
↓
B) Check reset for correct operation
O.K.
↓
C) Check U1 for proper operation
O.K. If not: 1) Check socket for good solder
↓ 2) Check for bad U1
D) Check U2 for proper operation
O.K. If not: 1) Check socket
↓ 2) Check for bad U2
E) Check RAM data lines for correct amplitude
O.K. If not: 1) Check for hot surface of RAM
↓ 2) Jump out RAM to verify
F) Check multiplexers U9, U10 - signals at RP3 and RP4 should be similar in frequency and amplitude
O.K. If not: 1) Suspect U9 or U10
↓
G) Check ROM for chip select signal at pin 22 of U23 and U24
O.K. If not: 1) Check for signal generation
↓ at U20
H) Check that all ROM addresses are present and correct amplitude
O.K. If not: 1) Trace problem address A0-A15
↓
I) Check U19, U23, U24 by replacement with known good

3) NO POWER
A) Verify voltage +5 and +9 volts
1) Check for shorts to ground
2) Check switch
3) Check power supply

4) BAD BASIC - Random characters on screen - Random colors - Power-up message is missing
A) Check Basic ROM U23
B) Check B thru I above (Bad Video)

5) NO COLOR or BAD COLOR

A) Check U1 pin 14 for 14.31818 MHz with frequency counter
O.K. ↓ If not: 1) Check solder joints of CT1 and adjust for correct frequency
2) Check crystal, Q1 and Q2
3) Check clock circuit for opens or shorts

B) Check U1 pin 13 for Color Out signal.
O.K. ↓ If not: 1) Swap U1 w/known good

C) Check modulator M1 pin 5 for Color In signal and pin 6 for Color Out signal
O.K. ↓ If not: 1) Check M1 operation

D) Check FB4 and CN7 pin 6 to see if color signal is present.
1) Check for shorts

6) NO SOUND or BAD SOUND

A) Check U1 pin 33 for SND signal
O.K. ↓ If not: 1) Check socket for open circuit
2) Swap U1 w/known good

B) Check audio circuit for short to ground or loss of signal.
O.K. ↓ If not: 1) Check Q3 - Be sure emitter and base are not shorted to 5 V.

C) Check modulator M1 pin 2 for SND signal
1) Adjust I.F. can (top right of modulator) for clean, loud volume
2) M1 pin 2 to ground should read approximately 480 ohms
3) Check M1 for component failure

7) SERIAL FAILURES

A) Check FB23-26 for shorts to shield or each other
B) Check U7, U2 and CN2

8) KEYBOARD FAILURES

A) Check pins on ribbon cable for good connection
O.K.
↓
B) Check for shorts - CN5, CN6, FB's, Diodes
O.K.
↓
C) Check chip select to U27 and the I.C. U27
O.K.
↓
D) Check U1 for proper operation
O.K. If not: 1) Check socket
↓ 2) Check for bad U1

9) FAILURES IN SOFTWARE MODE - All units should be checked for proper operation, when any repairs are necessary.
To Check: 1) Press 'F1' on keyboard
2) Press 'Return' to enter Word Processing mode
3) Press 'Commodore' key and 'C' key at the same time
4) Type 'tc' and press 'Return' to enter Spreadsheet
5) Press 'Commodore' key and 'C' key again
6) Type 'tw' to return to Word Processing mode

Watch for video or loading problems, then:

A) Check jumpers at J1-J6 for correct connection
O.K.
↓
B) Check U1, U2, U25, U26

Andy Brett
72. St. Neots Road,
Eaton Ford,
St. Neots,
Huntingdon,
Cambs.
PE19 3BD.

0480 213504

22 July 1991

Dear Roy,

I read Nick Ritchie's article about the Nufont program in the April/May issue, and I've written to him with a little information. For anyone else that is interested here's a rundown.

I had problems with the program when it was first published, and sent a disk with my non working 'Nufont' to Your Commodore. I recived back a corrected version, plus the descender set from the C64 'New Characters on the MPS 801/3' program. Also on the disk were two WP files 'Demo1' and 'Demo2' these are a slightly different version of the Nufont Creating Characters article in Your Commodore December 1988.

One piece of information was missed out altogether.

Quote from the article:

'List 1. is published in the Serious Users Guide (1987) as a basic program. (New Characters on the MSP 801/3) DESCENDER SET.)'

Then comes a line that was missed out:

'This can be used if required. Line 110 is changed from SA=64512 to SA=28672'

Using this information I typed in the program for the Italic Set (changing line 110 to SA=28672) and saved it. By loading 'Nufont',8,1 and then loading and running the 'Italic' program followed by SYS30000, I had a working program.

To make things easier I loaded as before, then went into the monitor and saved the complete Nufont Italic program. All I have to do now is load 'Nufont Italic',8,1

I hope this is of some help to anyone who had trouble getting the other fonts working.

* * * * * * *

I enclose two programs, one prints out a diary for a personal organiser. In order to get a double sided page it prints both sides side by side, you then cut out and fold in half so that the printing is on the outside. It is designed to print one week per page on 12in fan fold paper.

The second program prints out a daily dated year planner and a undated monthly year planner for a personal organiser. Because both printouts are wider than normal paper, they are printed down the paper. The day dated planner is about 13in long, making it print over the page perforations, and too long for a single sheet of paper.

If anyone wants a copy of these programs without the typing, or a copy of 'Nufont','Nufont Italic', or my version of 'Word Pro' with five different descenders, send me a disk or tape and S.A.E. and I'll send you a copy.

Andy

DIARY FOR PERSONAL ORGANISER

```
100 POKE 59468,12
110 CLR:DIM M$(26)
120 REM THIS PROGRAM WAS WRITTEN BY J.R ROSSUM
130 REM SUGGESTED BY 'JULIAN' IN THE HP BASIC LIBRARY
140 REM DIARY PRINTOUT WAS ADDED BY A.J.BRETT 1991
150 P=0:PRINT"■■■■■■■■■■■■PROGRAM TO GENERATE CALENDER FOR"
160 PRINT"■          FOR ANY MONTH "
170 PRINT"■      SINCE THE BIRTH OF CHRIST"
180 PRINT"■ OR PRINTOUT A PERSONAL ORGANISER DIARY    AFTER 1900 WITH 1 WEEK TO
2 PAGES"
190 PRINT:PRINT" TYPE ALL FOUR DIGITS OF DESIRED YEAR";PRINT:PRINT,:INPUTG:PRINT
200 PRINT:PRINT"DO YOU WANT TO PRINT A WHOLE YEAR? (Y/N)"
210 GETK$:IFK$<>"N"ANDK$<>"Y"THEN210
220 IFK$="Y"THENM=1:K$="P":P=1:GOTO280
230 PRINT:PRINT,:INPUT"MONTH(1-12)";M
240 PRINT:PRINT,"SCREEN OR PRINTER?"
250 GETK$:IFK$<>"S"ANDK$<>"P"THEN250
260 T0=TI:IFK$="P"THEN280
270 IF G<1900 THEN 360
280 SCNCLR
290 D=(G-1900)*365
300 FOR Q=1901 TO G-1
310 IF Q/4=INT(Q/4) THEN D=D+1
320 NEXT
330 D=D+2
340 IFK$="P"THEN710
350 GOTO 440
360 PRINT "■■■■■■■■■■■■■YOU WILL HAVE TO WAIT A BIT FOR THAT ONE"
370 D=G*365
380 FOR Q=1 TO G-1
390 IF Q/400=INT(Q/400) THEN 410
400 IF Q/100=INT(Q/100)THEN 420
410 IF Q/4=INT(Q/4) THEN D=D+1
420 NEXT
430 D=D+1
440 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■JULIAN CALENDAR"
450 FOR J=1 TO 24:READ M$(J):NEXT
460 FOR J=1TO7:READ D$(J):NEXT
470 PRINT M$(2*M-1)TAB(35)G
480 PRINT:PRINT:FOR J=1 TO 7:PRINT TAB(6*(J)-6)D$(J);" ";:NEXT
490 IF G/400=INT(G/400)THEN 510
500 IF G/100=INT(G/100) THEN 520
510 IF G/4=INT(G/4) THEN M$(4)="29"
520 FOR J=2 TO 2*M-2 STEP 2:S1=S1+VAL(M$(J)):NEXT
530 IF M=1 THEN S1=0
540 S=S1+D-7*INT((S1+D)/7)
550 V=VAL(M$(2*M))
560 S=S+7:IF S>7 THEN S=S-7
570 T=1+6*(S-1):U=5
580 PRINT:PRINTTAB(T-1);
590 IF T=37 THEN 610
600 FORD=1TOV:GOTO620
610 PRINT 1:FOR D=2TOV
620 PRINT D;SPC(U-LEN(STR$(D)));
630 IF POS(0)>34 THEN U=3
640 IF POS(0)<34 THEN U=5
650 NEXT
```

```
660 PRINT:PRINT:PRINT:PRINT TAB(10)"TIME="INT((TI-T0)/36)/100 "MINUTES"
670 PRINT:INPUT "ANOTHER";K$
680 IF LEFT$(K$,1)="Y" THEN 110
690 IF LEFT$(K$,1)="N" THEN END
700 GOTO 670
710 IFG<1900THENPRINT"[SHIFT-*]  SCREEN ONLY WILL NOT PRINT A DIARY",,"BEFORE 19
00"
720 IFG<1900THENFORJ=1TO3000:NEXT:GOTO150
730 PRINT"[SHIFT-*]IS YOUR PRINTER TURNED ON AND THE PAPER SET NEAR TO THE TOP OF
 PAGE? (Y)"
740 GETKEYK$:IFK$<>"Y"THEN150
750 IFP=1THENPRINT"[SHIFT-*]  PRINTING A DIARY FOR THE YEAR";G:GOTO770
760 PRINT"[SHIFT-*]    PRINTING A DIARY FOR ONE MONTH",,"IN THE YEAR";G
770 OPEN4,4:CMD4:FORJ=1TO37:L$=L$+"─":NEXT:L$=L$+";":FORJ=1TO37:L$=L$+"─":NEXT
780 F$="└":FORJ=1TO73:F$=F$+"─":NEXT:F$=F$+"┘"
790 S$="┌":FORJ=1TO73:S$=S$+"─":NEXT:S$=S$+"┐":FORJ=1TO6:B$=B$+" ":NEXT
800 FORJ=1TO24:READM$(J):NEXT:M$(25)=M$(1):M$(26)=M$(2):RESTORE1490
810 FOR J=1TO8:READ D$(J):NEXT:T=28-LEN(M$(2*M-1))
820 GOSUB1240:PRINT:PRINTSPC(T)M$(2*M-1);" ";G;B$;M$(2*M-1);" ";G
830 IF G/400=INT(G/400)THEN 850
840 IF G/100=INT(G/100) THEN 860
850 IF G/4=INT(G/4) THEN M$(4)="29"
860 FOR J=2 TO 2*M-2 STEP 2:S1=S1+VAL(M$(J)):NEXT
870 IF M=1 THEN S1=0
880 S=S1+D-7*INT((S1+D)/7):IFS=0THENS=7
890 V=VAL(M$(2*M)):C=1:D=0:X=0:D1=1:D2=9-S
900 DOUNTILC=>V:D=D+1:IFX=1THENC=C+1
910 TD=31-LEN(D$(D+4)):PRINTL$:IFX=0ANDD=STHENX=1
920 IFX=0THENGOSUB1270:IFD=4THENC=D1+3:GOSUB1250:PRINTL$:GOSUB1220:GOSUB1320:X=1
:D=0:GOTO1030
930 IFX=0THEN1020
940 IFD<>4THEN970
950 PRINTSPC(TD)D$(8);"          ";:IFC<10THENPRINT" ";
960 PRINTC;D$(D):GOSUB1250:PRINTL$:GOSUB1220:GOSUB1320:GOTO1030
970 PRINTSPC(TD)D$(D+4);:IFC<13THENPRINT" ";
980 IFC<5THENPRINT"    ";:ELSEPRINTC-3;
990 PRINT"     ";
1000 IFC<10THENPRINT" ";
1010 PRINTC;D$(D)
1020 GOSUB1250
1030 LOOP:D1=1
1040 IFC<>VTHEN1090
1050 C=C-2:IFD=3THEN1080
1060 FORJ=D+1TO3:TD=31-LEN(D$(J+4)):PRINTL$:PRINTSPC(TD)D$(J+4);" ";C;"     "D1;
D$(J)
1070 GOSUB1250:C=C+1:D1=D1+1:NEXT
1080 PRINTL$:PRINTSPC(26)D$(8);"          "D1;D$(4):GOTO1140
1090 C=C-2:D1=3-(V-C):IFD=3THEN1130
1100 FORJ=1TO3:TD=31-LEN(D$(J+4)):PRINTL$:IFC=V+1THENC=1
1110 PRINTSPC(TD)D$(J+4);:IFC<10THENPRINT" ";
1120 PRINTC;"      ";D1;D$(J):GOSUB1250:C=C+1:D1=D1+1:NEXT
1130 PRINTL$:PRINTSPC(26)D$(8);"          "D1;D$(4)
1140 IFP<>1THEN1170
1150 C=D1:M=M+1:V=VAL(M$(2*M)):IFM=13THEN1170:ELSEGOSUB1250:PRINTL$:GOSUB1220:GO
SUB1320
1160 GOTO900
1170 GOSUB1250:PRINTL$:PRINTF$:PRINT#4:CLOSE4
```

DIARY FOR PERSONAL ORGANISER

```
1180 PRINT"■■■■■■■■■■ FOLD DOWN CENTRE DOTS SO THAT THE ■      PRINTING IS ON THE
OUTSIDE,";
1190 PRINT" CUTOUT TO■   THE EDGE MARKS, GLUE AROUND EDGES AND"
1200 PRINT"■ PUNCH BINDER HOLES."
1210 PRINT"■■■■■■■■■■DO YOU WISH ANOTHER (Y)":GETKEYK$:IFK$="Y"THEN110:ELSE END
1220 PRINTF$:REM BOTTOM EDGE OF PAGE
1230 FORJ=1TO31:PRINT:NEXT:REM PAGE SPACING*ALTER THIS TO CHANGE SPACING BETWEEN
 PAGES
1240 PRINTS$:RETURN:REM TOP EDGE OF PAGE
1250 FORQ=1TO7:PRINT:NEXT:RETURN
1260 REM FIRST WEEK PRINT ROUTINE
1270 IFS<5THENPRINTSPC(TD)D$(D+4);"              ";D$(D):RETURN
1280 PRINTSPC(TD)D$(D+4);:IFD=4THENPRINT"     ";:GOTO1300
1290 IFD+4<5THENPRINT"     ";:ELSEPRINT" ";D1;:D1=D1+1
1300 PRINT"      ";D2;D$(D):D2=D2+1:RETURN
1310 REM PRINT MONTH HEADINGS
1320 T=28-(LEN(M$(2*M-1)))
1330 IFC=VTHENT=28-LEN(M$(2*M+1)):GOTO1380
1340 IFC=V-3THEN1440
1350 IFC>V-3THENT=27-LEN(M$(2*M+1))-LEN(M$(2*M-1)):GOTO1400
1360 IFC>V-7THEN1420
1370 PRINT:PRINTSPC(T)M$(2*M-1);" ";G;B$;M$(2*M-1);" ";G:D=0:C=C+3:RETURN
1380 IFM=12THENG=G+1
1390 PRINT:PRINTSPC(T)M$(2*M+1);" ";G;B$;M$(2*M+1);" ";G:D=0:C=C+3:RETURN
1400 PRINT:PRINTSPC(T)M$(2*M-1);" ";M$(2*M+1);" ";:IFM=12THENG=G+1
1410 PRINTG;B$;M$(2*M+1);" ";G:D=0:D1=8-(V-C):C=C+3:RETURN
1420 PRINT:PRINTSPC(T)M$(2*M-1);" ";G;B$;M$(2*M-1);" ";M$(2*M+1);" ";:IFM=12THEN
G=G+1
1430 PRINTG:D=0:D1=1:C=C+3:RETURN
1440 PRINT:PRINTSPC(T)M$(2*M-1);" ";G;B$;M$(2*M+1);" ";:IFM=12THENG=G+1
1450 PRINTG:D=0:D1=1:C=C+3:RETURN
1460 DATA JANUARY,31,FEBRUARY,28,MARCH,31,APRIL,30,MAY,31,JUNE,30,JULY,31
1470 DATA AUGUST,31,SEPTEMBER,30,OCTOBER,31,NOVEMBER,30,DECEMBER,31
1480 DATA SUN,MON,TUE,WED,THU,FRI,SAT
1490 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,NOTES
```

YEAR PLANNER

```
0 REM ****** YEAR PLANNER FOR PERSONAL ORGIANISER ******
100 DIMC$(44),M(12),S(12),S1(12)
110 G$=CHR$(8):H$=CHR$(15)
120 RESTORE:PRINT"                    PERSONAL    "
130 PRINT"                ORGANISER   ":PRINT"               YEAR   PLANNER "
140 INPUT"    ENTER YEAR BETWEEN 1900 & 2500";Y$:Y=VAL(Y$)
150 IFY<1900ORY>2500THENPRINT"            NOT ACCEPTABLE":FORJ=1TO2000:NEXT:GOTO1
20
160 PRINT"   DO YOU WANT A DAY DATED YEAR PLANNER,
170 PRINT"    OR A UNDATED MONTHLY YEAR PLANNER?"
180 PRINT"               PRESS D OR M"
190 GETKEYK$:IFK$="D"THEN800
200 IFK$="M"THEN220
210 GOTO190
220 PRINT"                 PLEASE WAIT - READING DATA"
230 FORC=0TO38
240 FORB=1TO12:READA
250 C$(C)=C$(C)+CHR$(A)
260 NEXTB,C
270 FORB=1TO6:READA
280 C$(39)=C$(39)+CHR$(A)
290 NEXTB
300 FORC=40TO42
310 FORB=1TO12:READA
320 C$(C)=C$(C)+CHR$(A):NEXTB,C
330 C$(43)=CHR$(255)
340 L$=C$(37):FORJ=1TO8:L$=L$+C$(36):NEXT
350 L1$=L$+C$(36)+C$(39):C$=C$(39)
360 S$=C$(37):FORJ=1TO9:S$=S$+C$(38):NEXT:T$=C$+C$(38)+C$
370 PRINT"          IS THIS THE CORRECT YEAR?  ";Y;" "
380 PRINT" IS THE PRINTER READY?":PRINT" IS THE PAPER SET TO TOP OF PAGE?":PRINT
"               Y OR N
390 GETK$:IFK$="Y"THEN420
400 IFK$="N"THEN120
410 GOTO390
420 READB,C,D
430 PRINT"          NOW PRINTING MONTHLY YEAR PLANNER FOR":PRINT"
  ";Y
440 OPEN4,4:PRINT#4,H$;"┌";:FORJ=1TO63:PRINT#4,"¯";:NEXT:PRINT#4,"┐ "G$:PRINT#4,
H$"| "SPC(63)" | "G$
450 PRINT#4,H$"|        +        +         +                        +        +         +        |
| ";G$
460 PRINT#4,H$"| "SPC(63)" | "G$:PRINT#4,H$"└";:FORJ=1TO63:PRINT#4,"_";:NEXT:PRINT
#4,"_┘ "G$
470 PRINT#4,G$;L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$
480 FORJ=1TO4:A=VAL(MID$(Y$,J,1)):READB,C,D
490 PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);C$;C$(A);C$:NEXT
500 FORJ=1TO4:READB,C,D:PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$:NEXT
510 GOSUB730:GOSUB720
520 FORJ=1TO8:READB,C,D
530 PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$:NEXT
540 A=13:GOSUB700:GOSUB720
550 FORJ=1TO8:READB,C,D
560 PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$:NEXT
570 A=13:GOSUB700:GOSUB720
580 READB,C,D:PRINT#4,G$;L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$
590 FORJ=1TO4:A=VAL(MID$(Y$,J,1)):READB,C,D
600 PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);C$;C$(A);C$:NEXT
```

```
610 FORJ=1TO3:READB,C,D:PRINT#4,L$;C$(B);C$;L$;C$(C);C$;L$;C$(D);T$:NEXT
620 GOSUB730
630 PRINT#4,H$;"└";:FORJ=1TO64:PRINT#4,"─";:NEXT:PRINT#4,"┘"
640 PRINT"■■■■■■■■■ CUT OUT & FOLD TWICE TO FORM THREE
650 PRINT"     EQUAL SECTIONS EXCLUDING THE                    LEFT MARGIN■"
660 PRINT"  FIRST FOLD FORWARDS TO FORM A CREASE      BETWEEN THE R & Y  IN FEBR
UARY,"
670 PRINT"  THEN BACKWARDS TO LINE UP THE R/H END        OF THE PAGE WITH THE CR
EASE
680 PRINT"■      PUNCH HOLES IN LEFT MARGIN"
690 PRINT"■■■■          ANY KEY TO CONTINUE":GETKEYK$:GOTO1190
700 FORK=1TOA:FORJ=1TO2:PRINT#4,L$;C$(36);C$;:NEXT:PRINT#4,L$;C$(36);C$;
710 PRINT#4,C$(38);C$:NEXTK:RETURN
720 FORK=1TO2:FORJ=1TO2:PRINT#4,S$;C$;:NEXT:PRINT#4,S$;T$:NEXTK:RETURN
730 PRINT#4,L1$;L1$;L1$;C$(33);C$(37):PRINT#4,L1$;L1$;L1$;C$(14);C$(37)
740 PRINT#4,L1$;L1$;L1$;C$(10);C$(37):PRINT#4,L1$;L1$;L1$;C$(27);C$(37)
750 PRINT#4,L1$;L1$;L1$;C$(38);C$(37):PRINT#4,L1$;L1$;L1$;C$(25);C$(37)
760 PRINT#4,L1$;L1$;L1$;C$(21);C$(37):PRINT#4,L1$;L1$;L1$;C$(10);C$(37)
770 PRINT#4,L1$;L1$;L1$;C$(23);C$(37):PRINT#4,L1$;L1$;L1$;C$(23);C$(37)
780 PRINT#4,L1$;L1$;L1$;C$(14);C$(37):PRINT#4,L1$;L1$;L1$;C$(27);C$(37)
790 RETURN
800 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■PLEASE WAIT - READING DATA"
810 D=(Y-1900)*365:FORJ=1901TOY-1:IFJ/4=INT(J/4)THEND=D+1
820 NEXT:D=D+2
830 J=2000:FORC=0TO38:RESTOREJ:FORB=1TO10:READA:C$(C)=C$(C)+CHR$(A):NEXTB:J=J+10
:NEXTC
840 RESTOREJ:FORC=1TO5:READA:C$(39)=C$(39)+CHR$(A):NEXT:J=J+10
850 FORC=40TO42:RESTOREJ:FORB=1TO10:READA:C$(C)=C$(C)+CHR$(A):NEXTB:J=J+10:NEXTC
860 RESTORE2480:FORJ=1TO12:READA:M(J)=A:NEXTJ
870 IFY/400=INT(Y/400)THEN890
880 IFY/100=INT(Y/100)THEN900
890 IFY/4=INT(Y/4)THENM(2)=29
900 A=2:FORJ=1TO11:S1(A)=S1(A-1)+M(J):A=A+1:NEXT:S1(1)=0
910 FORJ=1TO12:S(J)=S1(J)+D-7*INT((S1(J)+D)/7):S(J)=S(J)+7:IFS(J)>7THENS(J)=S(J)
-7
920 NEXT:C$(43)=CHR$(255):PRINT"■■■■■■■■■■■■■IS THIS THE CORRECT YEAR? ■";Y;"■"
930 PRINT"■IS THE PRINTER READY?":PRINT"■IS THE PAPER SET TO TOP OF PAGE?":PRINT
"■■■■■■■■■■■■■■Y OR N
940 GETK$:IFK$="Y"THEN970
950 IFK$="N"THEN120
960 GOTO940
970 OPEN4,4:B=1
980 D$=C$(28)+C$(15)+C$(29)+C$(32)+C$(29)+C$(22)+C$(28)+C$(43)
990 D1$=C$(10)+C$(27)+C$(17)+C$(14)+C$(30)+C$(24)+C$(30)+C$(43)
1000 D2$=C$(29)+C$(18)+C$(30)+C$(13)+C$(14)+C$(23)+C$(23)+C$(43)
1010 FORJ=1TO7:T$=T$+C$(42):NEXT:T$=T$+C$(43):FORJ=1TO7:L$=L$+C$(36):NEXT:L$=L$+
C$(43)
1020 PRINT"■■■■■■■■■■■■ NOW PRINTING YEAR PLANNER FOR";Y
1030 PRINT#4,H$;"┌";:FORJ=1TO64:PRINT#4,"─";:NEXT:PRINT#4,"┐"G$:PRINT#4,H$"│ "SPC
(64)" │"G$
1040 PRINT#4,H$"│         +         +         +                   +         +
  │";G$
1050 PRINT#4,H$"│ "SPC(64)" │"G$:PRINT#4,H$"└";:FORJ=1TO64:PRINT#4,"─";:NEXT:PRINT
#4,"┘"G$
```

YEAR PLANNER

```
1060 A=VAL(LEFT$(Y$,1)):GOSUB1250:A=VAL(MID$(Y$,2,1)):GOSUB1270:A=VAL(MID$(Y$,3,
1)):GOSUB1280
1070 A=VAL(MID$(Y$,4,1)):GOSUB1290
1080 GOSUB1300:FORC=2TO12:READA:GOSUB1290:B=C:GOSUB1300:NEXT
1090 A=VAL(LEFT$(Y$,1)):GOSUB1290
1100 A=VAL(MID$(Y$,2,1)):GOSUB1250:A=VAL(MID$(Y$,3,1)):GOSUB1270:A=VAL(MID$(Y$,4
,1)):GOSUB1280
1110 PRINT#4,H$"L";:FORJ=1TO65:PRINT#4,"_";:NEXT:PRINT#4,"I "G$
1120 PRINT"■■■■■■■■■ CUT OUT & FOLD THREE TIMES TO FORM FOUR
1130 PRINT"     EQUAL SECTIONS EXCLUDING THE                    LEFT MARGIN■"
1140 PRINT" FIRST FOLD FORWARDS ON THE LINE BETWEEN  JUNE & JULY, THEN FOLD THE
RIGHTHAND
1150 PRINT"END BACKWARDS TO MEET THE FIRST FOLDLINE THEN FOLD BACKWARDS SO THAT
THE FIRST
1160 PRINT"FOLD IS BEHIND THE LINE ON THE R/H SIDE "TAB(10);"OF THE LEFT MARGIN"
1170 PRINT"■      PUNCH HOLES IN LEFT MARGIN"
1180 PRINT"■■■■          ANY KEY TO CONTINUE":GETKEYK$
1190 PRINT#4,H$:CLOSE4
1200 PRINT"■■■■■■■■■■■■■■■ DO YOU WANT TO RUN THE PROGRAM AGAIN?":PRINTTAB(16)"■■Y/N
"
1210 GETKEYK$:IFK$="Y"THENPRINT"■":CLR:RUN
1220 IFK$="N"THEN1240
1230 GOTO1210
1240 PRINT"■";TAB(16)"■■BYE":END
1250 PRINT#4,G$;C$(22);C$(28);C$(43);D$;D$;D$;D$;D$;C$(36);C$(A);C$(43):RETURN
1260 PRINT#4,G$;C$(22);C$(28);C$(43);D$;D$;D$;D$;D$;C$(36);C$(A);C$(43)
1270 PRINT#4,G$;C$(24);C$(30);C$(43);D1$;D1$;D1$;D1$;D1$;C$(36);C$(A);C$(43):RET
URN
1280 PRINT#4,G$;C$(23);C$(23);C$(43);D2$;D2$;D2$;D2$;D2$;C$(36);C$(A);C$(43):RET
URN
1290 PRINT#4,C$(42);C$(42);C$(43);T$;T$;T$;T$;T$;C$(42);C$(A);C$(43):RETURN
1300 F=M(B)-1+S(B):D1=3
1310 D=M(B):FORJ=37TO1STEP-1:D1=D1-1:IFD1<1THENPRINT#4,C$(43);:D1=7
1320 IFF<JORD<1THENPRINT#4,C$(41);:NEXT:GOTO1380
1330 IFD>29THENPRINT#4,C$(3);:D=D-1:NEXT
1340 IFD>19THENPRINT#4,C$(2);:D=D-1:NEXT
1350 IFD>9THENPRINT#4,C$(1);:D=D-1:NEXT
1360 IFD>0THENPRINT#4,C$(36);:D=D-1
1370 NEXT
1380 READA,E:PRINT#4,C$(43);C$(A);C$(E);C$(43):D1=3
1390 M$=STR$(M(B)):D=M(B):D2=VAL(RIGHT$(M$,1))
1400 FORJ=37TO1STEP-1:D1=D1-1:IFD1<1THENPRINT#4,C$(43);:D1=7
1410 IFF<JORD<1THENPRINT#4,C$(41);:NEXT:GOTO1440
1420 PRINT#4,C$(D2);:D=D-1:D2=D2-1:IFD2<0THEND2=9
1430 NEXT
1440 READA,E:PRINT#4,C$(43);C$(A);C$(E);C$(43)
1450 FORJ=1TO7:READA,E:PRINT#4,C$(36);C$(36)C$(43);L$;L$;L$;L$;L$;C$(A);C$(E);C$
(43):NEXT
1460 RETURN
2000 DATA255,128,188,230,238,246,230,188,128,128,128,128
2010 DATA255,128,254,152,152,156,152,152,128,128,128,128
2020 DATA255,128,254,134,188,224,230,188,128,128,128,128
2030 DATA255,128,188,230,224,184,176,254,128,128,128,128
2040 DATA255,128,176,254,178,180,184,176,128,128,128,128
2050 DATA255,128,188,230,224,190,134,254,128,128,128,128
```

```
2060 DATA255,128,188,230,230,190,134,252,128,128,128,128
2070 DATA255,128,140,140,152,176,230,254,128,128,128,128
2080 DATA255,128,188,230,230,188,230,188,128,128,128,128
2090 DATA255,128,190,224,252,230,230,188,128,128,128,128
2100 DATA255,128,230,254,230,230,188,152,128,128,128,128
2110 DATA255,128,190,230,230,190,230,190,128,128,128,128
2120 DATA255,128,188,230,134,134,230,188,128,128,128,128
2130 DATA255,128,190,182,230,230,182,158,128,128,128,128
2140 DATA255,128,254,134,134,158,134,254,128,128,128,128
2150 DATA255,128,134,134,134,158,134,254,128,128,128,128
2160 DATA255,128,188,230,246,134,230,188,128,128,128,128
2170 DATA255,128,230,230,230,254,230,230,128,128,128,128
2180 DATA255,128,188,152,152,152,152,188,128,128,128,128
2190 DATA255,128,156,182,176,176,176,248,128,128,128,128
2200 DATA255,128,230,182,158,158,182,230,128,128,128,128
2210 DATA255,128,190,134,134,134,134,134,128,128,128,128
2220 DATA255,128,226,226,234,254,246,226,128,128,128,128
2230 DATA255,128,230,246,254,254,238,230,128,128,128,128
2240 DATA255,128,188,230,230,230,230,188,128,128,128,128
2250 DATA255,128,134,134,190,230,230,190,128,128,128,128
2260 DATA255,128,176,188,230,230,230,188,128,128,128,128
2270 DATA255,128,230,182,190,230,230,190,128,128,128,128
2280 DATA255,128,188,230,224,188,134,252,128,128,128,128
2290 DATA255,128,152,152,152,152,152,254,128,128,128,128
2300 DATA255,128,188,230,230,230,230,230,128,128,128,128
2310 DATA255,128,152,164,230,230,230,230,128,128,128,128
2320 DATA255,128,226,254,254,234,226,226,128,128,128,128
2330 DATA255,128,152,152,152,188,230,230,128,128,128,128
2340 DATA255,128,230,230,188,152,188,230,128,128,128,128
2350 DATA255,128,254,134,140,152,176,254,128,128,128,128
2360 DATA255,128,128,128,128,128,128,128,128,128,128,128
2370 DATA255,255,128,128,128,128,128,128,128,128,128,128
2380 DATA128,128,128,128,128,128,128,128,128,128,128,128
2390 DATA255,255,128,128,128,128
2400 DATA192,192,192,192,192,192,192,192,192,192,192,192
2410 DATA170,213,170,213,170,213,170,213,170,213,170,213
2420 DATA255,136,136,136,136,136,136,136,136,136,136,136
2430 DATA28,22,19,14,10,10,25,33,23,29,36,30,14,36,10,22,36,27,11,36,33,14,36,36
,27,36,36
2440 DATA24,19,15,12,30,14,29,23,11,24,14,27,11,36,30,14,36,10,27,36,27,36,36,33
2450 DATA23,19,22,24,30,10,31,21,27,14,33,12,22,36,17,11,36,36,14,36,36,27,36,36
2460 DATA13,10,10,14,30,25,12,16,27,14,30,18,22,28,21,11,29,36,14,36,36,27,36,36
2470 REM:
2480 DATA31,28,31,30,31,30,31,31,30,31,30,31
2490 DATA36,36,19,36,10,33,23,14,30,10,10,27,27,36,33,25,36,21,10
2500 DATA36,23,15,23,14,14,11,27,27,36,30,36,10,36,27,36,33,36,36
2510 DATA36,36,36,36,22,36,10,36,27,36,12,36,17,36,36,36,36,36,36
2520 DATA36,36,36,36,10,36,25,36,27,36,18,36,21,36,36,36,36,36,36
2530 DATA36,36,36,36,36,36,22,36,10,36,33,36,36,36,36,36,36,36,36
2540 DATA36,36,36,36,19,36,30,36,23,36,14,36,36,36,36,36,36,36,36
2550 DATA36,36,36,36,19,36,30,36,21,36,33,36,36,36,36,36,36,36,36
2560 DATA36,36,10,36,30,36,16,36,30,36,28,36,29,36,36,36,36,36,36
2570 DATA28,36,14,36,25,36,29,36,14,36,22,36,11,36,14,36,27,36,36
2580 DATA36,36,24,36,12,36,29,36,24,36,11,36,14,36,27,36,36,36,36
2590 DATA36,36,23,36,24,36,31,36,14,36,22,36,11,33,14,14,27,10,27
2600 DATA36,36,13,25,14,21,12,10,14,23,22,23,11,14,14,27,27,36,36
```

1992 YEAR PLANNER

	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEM
SUN			1								1
MON			2			1					2
TUE			3			2			1		3
WED	1		4	1		3	1		2		4
THU	2		5	2		4	2		3	1	5
FRI	3		6	3	1	5	3		4	2	6
SAT	4	1	7	4	2	6	4	1	5	3	7
SUN	5	2	8	5	3	7	5	2	6	4	8
MON	6	3	9	6	4	8	6	3	7	5	9
TUE	7	4	10	7	5	9	7	4	8	6	10
WED	8	5	11	8	6	10	8	5	9	7	11
THU	9	6	12	9	7	11	9	6	10	8	12
FRI	10	7	13	10	8	12	10	7	11	9	13
SAT	11	8	14	11	9	13	11	8	12	10	14
SUN	12	9	15	12	10	14	12	9	13	11	15
MON	13	10	16	13	11	15	13	10	14	12	16
TUE	14	11	17	14	12	16	14	11	15	13	17
WED	15	12	18	15	13	17	15	12	16	14	18
THU	16	13	19	16	14	18	16	13	17	15	19
FRI	17	14	20	17	15	19	17	14	18	16	20
SAT	18	15	21	18	16	20	18	15	19	17	21
SUN	19	16	22	19	17	21	19	16	20	18	22
MON	20	17	23	20	18	22	20	17	21	19	23
TUE	21	18	24	21	19	23	21	18	22	20	24
WED	22	19	25	22	20	24	22	19	23	21	25
THU	23	20	26	23	21	25	23	20	24	22	26
FRI	24	21	27	24	22	26	24	21	25	23	27
SAT	25	22	28	25	23	27	25	22	26	24	28
SUN	26	23	29	26	24	28	26	23	27	25	29
MON	27	24	30	27	25	29	27	24	28	26	30
TUE	28	25	31	28	26	30	28	25	29	27	
WED	29	26		29	27		29	26	30	28	
THU	30	27		30	28		30	27		29	
FRI	31	28			29		31	28		30	
SAT		29			30			29		31	
SUN					31			30			
MON								31			

1992 YEAR PLANNER

JANUARY	FEBRUARY	MARCH	APRIL

MAY	JUNE	JULY	AUGUST

SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER

1992 YEAR PLANNER

JANUARY	FEBRUARY	MARCH	APRIL

MAY	JUNE	JULY	AUGUST

SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER

MON	
SUN	
SAT	
FRI	
THU	
WED	
TUE	
MON	
SUN	
SAT	
FRI	
THU	
WED	
TUE	
MON	
SUN	
SAT	
FRI	
THU	
WED	
TUE	
MON	
SUN	
SAT	
FRI	
THU	
WED	
TUE	
MON	
SUN	
EMBER	
NBER 1992	

```
**************************************************************************
**************************** LUNAR LANDER ********************************
****************************     part 3    ********************************
**************************** by PETER CRACK ******************************
**************************************************************************
* Hello again. this month I will not tire you with long explanations of each
* routine. but just give you the outline of what each one will do.
* so here goes. As always first load in the programme so far (all parts).
4A00-4A0A Transfer sprite pointers 'X'=start of transfer point 'Y'=number of
*         pointers counting down to zero.
4A0B-4A15 Same as above.
4A16-4A28 Save sprite angle and save sprite page number.
4A29-4A36 Get sprite page number and angle save them in $460D and $D1.
4A36-4A45 Load sprite data into working area.
4A47-4A5E Set sprite number get this sprites data and rotation pointer. if
*         it is equal do no more else move sprite one pixel point to the
*         right until $D7 is counted down to zero. this lines up the new
*         sprite with the one on the screen.
4A60-4CD0 This routine is for the largest sprite and makes all the checks as
*         required.
4A60-4AA9 This routine is done when last set of screens first entered. save
*         screen number just left and gosub set new screen. work out the new
*         sprites position in the new screen and print it reset raster
*         offset from middle of sprite in $4500 (print sprite routine).
4AAC-4CD0 Start of main loop 3 (this is the loop for the largest sprite).
4AAC-4AD9 Print rocket sprite. check $E8 is engine running no branch to
*         $4ADB else load 'Y' reg. with flame sprite start point and
*         increase flame sprite swap number. check if it is odd if not
*         branch to $4AC2 else load #$70 into 'Y' reg. (this is the start
*         point of the second flame sprite and by swapping one with the
*         other each time a flicker is produced). Set correct angle and set
*         $E5 to #$01 (flame sprite). get pointers and print sprite.
4ADB      If $E5 is zero then gosub remove flame sprite.
4ADB-4AEA Check hight of sprite if it has moved below the stars on the
*         screen then check collision pointer $4604 and if this is not equal
*         jump to crash routine.
4AED-4B0F This part checks all the values which have to be correct to make
*         a good landing. first has the rocket reached the level of a
*         landing site (all sites are #$97 pixels down the screen). then
*         check if it is upright (it cannot land on its side). if yes then
*         is it still rotating?. is it moving horizontally?. and lastly
*         its vertical speed. first the hundreds column then the tens
*         anything less then ten can be coped with by the landing gear.
*         if it passes all these tests then branch to $4B12 else JMP$4BC1.
4B1D-4B35 This part puts the bump into this landing place so preventing
*         it from being used again. remember the rocket must land on a flat
*         place. get screen page number and store it into the routine.
*         load the correct values and store them into this screens data page
4B37-4B3F Set this landing places number in the message string. and increase
*         the number of good landings by one.
4B42-4B50 Print message.
4B53-4B63 Gosub compress the four fuel values into two $5C and $5D add #$05
*         to $5C (the hundreds and thousands value) thus adding 500 fuel
*         units to total. this is done in decimal to prevent non numeric
*         values appearing in the total and gosub return them to registers
4B66-4B75 Check number of good landings if less than 9 then branch to $4B8C
*         else print end message.
4B78-4B89 Get key input and check value . N=jump to $41DB and end game.
*         Y=jump to $5000 and start again. reseting the stack just in case.
4B89-4B97 Short delay so that messages can be read.
4B99-4B8E Gosub reset pointers. reset raster offset and reset information
*         list after a good landing ready to start again with another rocket
*         and jump to start of first screen.
4BC1-4C31 Having found that the rocket has not landed we continue the checks
*         on its position this routine checks to see if it is moving of the
************************************ CONTINUED ****************************
```

```
*****************************************************************************
*         we check the vertical position register and the to see if the word
*         'UP' is still in place at the end of the vertical speed indicator.
*         if no then branch to $43C4 else clear screen save present screen
*         pointers and gosub print next screen. get sprite angle and mix in
*         sprite two data page number. clear $60 and reset raster offset.
*         gosub get sprite pointers. calculate the position required on new
*         screen to make the sprite look as if it has just risen from the
*         second screen landing area ($60 and $61 are the pointers for
*         across the screen and $62 the one for down). and jump to the start
*         of loop two.
4C34-4C37 $4609 is the high byte of the position across the screen of the
*         sprites pointers so if this is zero then the rocket is not at the
*         right hand side so branch to $4C7D else.....
4C39-4C7A The rocket is near the righ hand side so check the low byte
*         against the limit $4703. no? then branch to $47CD . now check if
*         it is intending to continue moving right no? then again branch to
*         $47CD else. increase screen counter and see if we are on the last
*         one if yes jump to $4DB0 and print message else. set up next
*         screen calculate the rockets new position. print new screen and
*         print rocket then jump back to start of loop three.
4C7D-4CBC As above but for moving off the left hand edge of the screen.
4CBF-4CD3 If none of the above apply then get joystick return. reset speed
*         and fuel pointers and jumpback to start of loop three.
4D00-4D79 This routine prints a new screen. set draw pointers to left side
*         of screen. and draw a line linking up 88 dots using the basic
*         'DRAW' command. now put 48 dots at the top of the screen using
*         the same routine.
4D7A-4D88 Get and set new sprite pointers 'X'=end of pointer area in $4800
*         'Y'=number of pointers to fetch the data area for the second
*         routine is $4900.
4D89-4D9E Random number generator $A70E is the basic routine and the number
*         limits are placed into $4D92 and $4D96 for comparison.
44DA-4DAD This routine clears the under sprite data areas for both the
*         current rocket and flame sprite this prevents spurious data being
*         transferred from one screen to the next.
4DB0-4DCA This routine isused when a rocket moves off the the last screen
*         either left or right and in doing so leaves your sector of control
*         so.... reset stack pointer to clear any RTS addresses. clear the
*         screen and print a message then jump to short delay at $4B8C.
4DE0-4DFC Reset pointers after a good landing or moving off the top of one
*         of the screen two or three groups.
4E00-4E57 This is one of the second group of screens base line the values
*         down the screen. low byte the high byte is always zero.
4E58-4EAF This is the values for how far across the screen. low byte the
*         high byte. no matter which screen is being printed is always
*         taken from $5300 to $5357. think of it as a join the dots picture
*         the values represent the position of the dots and the computer
*         just draws one line to join them all up.
4EB0-4EFF These locations do represent some important information but at the
*         moment all I can tell you is that the first three are the page
*         numbers of the second group of screen base lines.
4F00-4F17 Begin the check for position in the first screen and the second
*         group of screens. this part is for moving up.
4F19-4F2F The rocket has just moved off the top of screen one. so print
*         message and jump to delay. the two PLA commands at the start are
*         to enable the programme to leave a subroutine without an RTS
*         something which cannot be done in BASIC (I think).
4F32-4F74 The rocket has just left a second level screen and moved to the
*         first level screen. so. calculate its new position in screen one
*         and jump to start of loop one.
4F77-4FCA Check for moving off the righ side of the screen. if on screen
*         one then jump to $4BD0 and print message else if on one of the
*         screen two group check to see if it is the last on the right if
*         yes then again jump to $4BD0 else move to the next screen on the
*         right or if all else fails return from subroutine.
***************************************** CONTINUED **********
```

```
*************************************************************************
4FCB-4FF7 As above but for moving off the left hand edge of the screen
5000-50E9 This at last is the start point of the programme.
5000-5020 Make sure that all landing places are cleared.
5022-5039 Set out of fuel flag to one (zero=no fuel). set number of landings
*         to zero. gosub set a few more pointers. and set raster offset to
*         prevent flicker for the first two sizes of sprites.
503E-504F Set sound and volume.
5052-5054 Set high res screen. in TEDMON chip.
5057-507E Set initial values for information lines.
5081-508C Set initial characters for infomation lines.
509F-50A9 Clear under sprite data areas.
50AB-50CD Set $07F8 so that $8000 to $FFFF can be seen. set screen and
*         border colours to black. set character and foreground colour. set
*         aside 10K (this is so BASIC routines can be used). set colour
*         source to one (foreground). set HI-RES (BASIC). set screen size
*         again for use of BASIC routines.
50CF-50D2 Gosub clear screen and print first game screen. gosub get first
*         sprite data.
50D5-50E6 And print sprite.
50E7-50E9 Set first screen number.
50EC-5166 This is the loop for the first and smallest sprite.
50EC-50EF Gosub move and print sprite. gosub check if it has moved off the
*         screen.
50F2-5100 Check to see if it has moved below the stars. if yes check to see
*         if it has crashed into something on the screen. ($5BB0 is crash
*         routine.
5103-514C Check to see if it is near one of the nine landing places these
*         are in three groups of three so it is almost impossible to crash
*         on screen one. but it can be done.
5156-5166 Gosub get joystick return. gosub update information lines and
*         update volume. and jump back to the start of loop one.
C000-CDFF Yet more data this is for the largest sprites flames. in each page
*         data from location $$00 to $$6F is for flame one and $$70 to $$BF
*         is for flame two $$C0 to $$FF is blank and is printed when the
*         rockets engine is switched off thus clearing the flame sprite.
*         and thats it for this month but there is more to come.
********************************** PETER CRACK ****************************
*************************************************************************
*
```

```
*************************************************************************
HELLO ROY.
*        This is for the letters page.
* First off a big thanks to NICK RICHIE for the information he has sent to
* me. following on from this as he sugested. could anybody out there send in
* to the magazine some short sound routines in BASIC or M/C which I could
* use in the programme. Now as it is an adventure set in a haunted house
* with a few ghosts. creaky doors. eerie winds and such could I have a few
* souds along those lines. As far as NICK being the oldest member (in age
* that is). I am sure you are not. Computing is not just for the young.
* this is all very serious and grown up stuff. At least thatis what I tell
* my family as I drag them around yet another computer store.
*                    all the best till next month
*
*
*                        PETER CRACK.
```

```
. 4A00  BD 00 46 LDA $4600,X
. 4A03  99 00 46 STA $4600,Y
. 4A06  CA       DEX
. 4A07  88       DEY
. 4A08  10 F6    BPL $4A00
. 4A0A  60       RTS
. 4A0B  BD 00 49 LDA $4900.X
. 4A0E  99 00 47 STA $4700.Y
. 4A11  CA       DEX
. 4A12  88       DEY
. 4A13  10 F6    BPL $4A0B
. 4A15  EA       NOP
. 4A16  AD 0D 46 LDA $460D
. 4A19  48       PHA
. 4A1A  29 0F    AND #$0F
. 4A1C  8D E7 53 STA $53E7
. 4A1F  68       PLA
. 4A20  29 F0    AND #$F0
. 4A22  8D F6 53 STA $53F6
. 4A25  20 29 4A JSR $4A29
. 4A28  60       RTS
. 4A29  AD E7 53 LDA $53E7
. 4A2C  85 E5    STA $E5
. 4A2E  0D F6 53 ORA $53F6
. 4A31  8D 0D 46 STA $460D
. 4A34  85 D1    STA $D1
. 4A36  78       SEI
. 4A37  8D 3F FF STA $FF3F
. 4A3A  A0 01    LDY #$01
. 4A3C  84 D0    STY $D0
. 4A3E  88       DEY
. 4A3F  B1 D0    LDA ($D0).Y
. 4A41  99 00 81 STA $8100.Y
. 4A44  C8       INY
. 4A45  D0 F8    BNE $4A3F
. 4A47  A9 00    LDA #$00
. 4A49  20 F0 41 JSR $41F0
. 4A4C  A4 D7    LDY $D7
. 4A4E  F0 0B    BEQ $4A5B
. 4A50  98       TYA
. 4A51  48       PHA
. 4A52  20 13 44 JSR $4413
. 4A55  68       PLA
. 4A56  A8       TAY
. 4A57  88       DEY
. 4A58  D0 F6    BNE $4A50
. 4A5A  8D 3E FF STA $FF3E
. 4A5D  58       CLI
. 4A5E  60       RTS
. 4A5F  EA       NOP
. 4A60  8E D1 67 STX $67D1
. 4A63  20 54 43 JSR $4354
. 4A66  A9 01    LDA #$01
. 4A68  8D D2 6B STA $6BD2
. 4A6B  AE D2 67 LDX $67D2
. 4A6E  AD 0A 46 LDA $460A
. 4A71  38       SEC
. 4A72  FD D8 67 SBC $67D8.X
. 4A75  10 13    BPL $4A8A
. 4A77  49 FF    EOR #$FF
. 4A79  18       CLC
. 4A7A  69 01    ADC #$01
. 4A7C  0A       ASL
. 4A7D  0A       ASL
. 4A7E  85 61    STA $61
. 4A80  A9 A0    LDA #$A0
. 4A82  38       SEC
. 4A83  E5 61    SBC $61
. 4A85  85 61    STA $61
. 4A87  38       SEC
. 4A88  B0 07    BCS $4A91
. 4A8A  0A       ASL
. 4A8B  0A       ASL
. 4A8C  18       CLC
. 4A8D  69 A0    ADC #$A0
. 4A8F  85 61    STA $61
. 4A91  20 8F 43 JSR $438F
. 4A94  A9 00    LDA #$00
. 4A96  85 60    STA $60
. 4A98  A9 10    LDA #$10
. 4A9A  85 62    STA $62
. 4A9C  A9 68    LDA #$68
. 4A9E  8D 0F 41 STA $410F
. 4AA1  8D 30 41 STA $4130
. 4AA4  20 65 42 JSR $4265
. 4AA7  A9 00    LDA #$00
. 4AA9  8D 04 45 STA $4504
. 4AAC  20 07 40 JSR $4007
. 4AAF  A5 E8    LDA $E8
. 4AB1  C9 01    CMP #$01
. 4AB3  D0 26    BNE $4ADB
. 4AB5  A0 00    LDY #$00
. 4AB7  EE F4 53 INC $53F4
. 4ABA  AD F4 53 LDA $53F4
. 4ABD  4A       LSR
. 4ABE  90 02    BCC $4AC2
. 4AC0  A0 70    LDY #$70
. 4AC2  98       TYA
. 4AC3  48       PHA
. 4AC4  A9 C0    LDA #$C0
. 4AC6  0D E7 53 ORA $53E7
. 4AC9  8D 46 43 STA $4346
. 4ACC  A9 01    LDA #$01
. 4ACE  85 E5    STA $E5
. 4AD0  20 F0 41 JSR $41F0
. 4AD3  68       PLA
. 4AD4  A8       TAY
. 4AD5  20 FC 40 JSR $40FC
. 4AD8  38       SEC
. 4AD9  B0 03    BCS $4ADE
. 4ADB  20 A8 43 JSR $43A8
. 4ADE  AD 0B 46 LDA $460B
. 4AE1  C9 38    CMP #$38
. 4AE3  90 2A    BCC $4B0F
. 4AE5  AD 04 46 LDA $4604
. 4AE8  F0 03    BEQ $4AED
. 4AEA  4C B0 5B JMP $5BB0
. 4AED  AD 0B 46 LDA $460B
. 4AF0  C9 97    CMP #$97
. 4AF2  D0 1B    BNE $4B0F
. 4AF4  AD E7 53 LDA $53E7
. 4AF7  D0 16    BNE $4B0F
. 4AF9  AD BA 53 LDA $53BA
. 4AFC  CD DE 53 CMP $53DE
. 4AFF  D0 0E    BNE $4B0F
. 4B01  AD CE 53 LDA $53CE
. 4B04  C9 30    CMP #$30
. 4B06  D0 07    BNE $4B0F
. 4B08  AD CF 53 LDA $53CF
. 4B0B  C9 30    CMP #$30
. 4B0D  F0 03    BEQ $4B12
. 4B0F  4C C1 4B JMP $4BC1
```

40

```
. 4B12  78       SEI
. 4B13  8D 3F FF STA $FF3F
. 4B16  20 A8 43 JSR $43A8
. 4B19  8D 3E FF STA $FF3E
. 4B1C  58       CLI
. 4B1D  AD 19 4D LDA $4D19
. 4B20  8D 33 4B STA $4B33
. 4B23  29 0F    AND #$0F
. 4B25  AA       TAX
. 4B26  BD E0 67 LDA $67E0.X
. 4B29  8D 32 4B STA $4B32
. 4B2C  A2 02    LDX #$02
. 4B2E  BD E4 67 LDA $67E4.X
. 4B31  9D 22 68 STA $6822.X
. 4B34  CA       DEX
. 4B35  10 F7    BPL $4B2E
. 4B37  AD D0 67 LDA $67D0
. 4B3A  09 30    ORA #$30
. 4B3C  8D D3 69 STA $69D3
. 4B3F  EE F7 69 INC $69F7
. 4B42  A9 69    LDA #$69
. 4B44  85 23    STA $23
. 4B46  A9 B0    LDA #$B0
. 4B48  85 22    STA $22
. 4B4A  A9 48    LDA #$48
. 4B4C  A0 01    LDY #$01
. 4B4E  A2 01    LDX #$01
. 4B50  20 E2 41 JSR $41E2
. 4B53  A0 0F    LDY #$0F
. 4B55  20 00 56 JSR $5600
. 4B58  F8       SED
. 4B59  A5 5C    LDA $5C
. 4B5B  18       CLC
. 4B5C  69 05    ADC #$05
. 4B5E  85 5C    STA $5C
. 4B60  D8       CLD
. 4B61  A0 0F    LDY #$0F
. 4B63  20 30 56 JSR $5630
. 4B66  AD F7 69 LDA $69F7
. 4B69  C9 39    CMP #$39
. 4B6B  90 1F    BCC $4B8C
. 4B6D  E6 23    INC $23
. 4B6F  A9 18    LDA #$18
. 4B71  A0 08    LDY #$08
. 4B73  A2 04    LDX #$04
. 4B75  20 E2 41 JSR $41E2
. 4B78  20 E4 FF JSR $FFE4
. 4B7B  C9 4E    CMP #$4E
. 4B7D  D0 03    BNE $4B82
. 4B7F  4C DB 41 JMP $41DB
. 4B82  C9 59    CMP #$59
. 4B84  D0 F2    BNE $4B78
. 4B86  A2 F0    LDX #$F0
. 4B88  9A       TXS
. 4B89  4C 00 50 JMP $5000
. 4B8C  A2 FF    LDX #$FF
. 4B8E  A0 00    LDY #$00
. 4B90  20 11 E3 JSR $E311
. 4B93  88       DEY
. 4B94  D0 FA    BNE $4B90
. 4B96  CA       DEX
. 4B97  D0 F7    BNE $4B90
. 4B99  20 E0 4D JSR $4DE0
. 4B9C  A9 34    LDA #$34
. 4B9E  8D 04 45 STA $4504
. 4BA1  A2 33    LDX #$33
. 4BA3  8E CF 53 STX $53CF
. 4BA6  CA       DEX
. 4BA7  8E DB 53 STX $53DB
. 4BAA  A2 44    LDX #$44
. 4BAC  8E D2 53 STX $53D2
. 4BAF  A2 49    LDX #$49
. 4BB1  8E DF 53 STX $53DF
. 4BB4  A9 52    LDA #$52
. 4BB6  8D DE 53 STA $53DE
. 4BB9  A9 4F    LDA #$4F
. 4BBB  8D D3 53 STA $53D3
. 4BBE  4C CF 50 JMP $50CF
. 4BC1  AD 06 47 LDA $4706
. 4BC4  CD 0B 46 CMP $460B
. 4BC7  90 6B    BCC $4C34
. 4BC9  AD D2 53 LDA $53D2
. 4BCC  C9 55    CMP #$55
. 4BCE  D0 64    BNE $4C34
. 4BD0  20 67 C5 JSR $C567
. 4BD3  AE B7 4E LDX $4EB7
. 4BD6  20 A0 4D JSR $4DA0
. 4BD9  BD B0 4E LDA $4EB0.X
. 4BDC  8D 19 4D STA $4D19
. 4BDF  8D 25 4D STA $4D25
. 4BE2  BD B3 4E LDA $4EB3.X
. 4BE5  20 05 4D JSR $4D05
. 4BE8  AD E7 53 LDA $53E7
. 4BEB  09 B0    ORA #$B0
. 4BED  48       PHA
. 4BEE  A9 00    LDA #$00
. 4BF0  85 60    STA $60
. 4BF2  A9 34    LDA #$34
. 4BF4  8D 04 45 STA $4504
. 4BF7  A2 8F    LDX #$8F
. 4BF9  A0 1F    LDY #$1F
. 4BFB  20 00 4A JSR $4A00
. 4BFE  A2 47    LDX #$47
. 4C00  A0 0F    LDY #$0F
. 4C02  68       PLA
. 4C03  8D 0D 46 STA $460D
. 4C06  20 0B 4A JSR $4A0B
. 4C09  A9 A0    LDA #$A0
. 4C0B  85 61    STA $61
. 4C0D  A9 70    LDA #$70
. 4C0F  85 62    STA $62
. 4C11  EE 79 42 INC $4279
. 4C14  EE 98 42 INC $4298
. 4C17  20 74 42 JSR $4274
. 4C1A  CE 79 42 DEC $4279
. 4C1D  CE 98 42 DEC $4298
. 4C20  8D 3E FF STA $FF3E
. 4C23  58       CLI
. 4C24  AE D0 67 LDX $67D0
. 4C27  BD AF 68 LDA $68AF.X
. 4C2A  85 62    STA $62
. 4C2C  BD BF 68 LDA $68BF.X
. 4C2F  85 61    STA $61
. 4C31  4C D5 51 JMP $51D5
. 4C34  AD 09 46 LDA $4609
. 4C37  F0 44    BEQ $4C7D
. 4C39  AD 0A 46 LDA $460A
. 4C3C  CD 03 47 CMP $4703
. 4C3F  D0 3C    BNE $4C7D
. 4C41  AD DE 53 LDA $53DE
. 4C44  C9 52    CMP #$52
. 4C46  D0 35    BNE $4C7D
```

```
. 4C48  EE D0 67  INC $67D0
. 4C4B  AE D0 67  LDX $67D0
. 4C4E  E0 0A     CPX #$0A
. 4C50  D0 03     BNE $4C55
. 4C52  4C B0 4D  JMP $4DB0
. 4C55  BD AF 6B  LDA $6BAF.X
. 4C58  8D B7 4E  STA $4EB7
. 4C5B  BD BF 6B  LDA $6BBF.X
. 4C5E  8D D1 67  STA $67D1
. 4C61  20 54 43  JSR $4354
. 4C64  A9 00     LDA #$00
. 4C66  85 60     STA $60
. 4C68  A9 1C     LDA #$1C
. 4C6A  85 61     STA $61
. 4C6C  AD 0B 46  LDA $460B
. 4C6F  38        SEC
. 4C70  E9 20     SBC #$20
. 4C72  85 62     STA $62
. 4C74  20 8F 43  JSR $438F
. 4C77  20 65 42  JSR $4265
. 4C7A  4C AC 4A  JMP $4AAC
. 4C7D  AD 0A 46  LDA $460A
. 4C80  CD 05 47  CMP $4705
. 4C83  D0 3A     BNE $4CBF
. 4C85  AD DE 53  LDA $53DE
. 4C88  C9 4C     CMP #$4C
. 4C8A  D0 33     BNE $4CBF
. 4C8C  CE D0 67  DEC $67D0
. 4C8F  AE D0 67  LDX $67D0
. 4C92  D0 03     BNE $4C97
. 4C94  4C B0 4D  JMP $4DB0
. 4C97  BD AF 6B  LDA $6BAF.X
. 4C9A  8D B7 4E  STA $4EB7
. 4C9D  BD BF 6B  LDA $6BBF.X
. 4CA0  8D D1 67  STA $67D1
. 4CA3  20 54 43  JSR $4354
. 4CA6  A9 00     LDA #$00
. 4CA8  85 60     STA $60
. 4CAA  A9 F4     LDA #$F4
. 4CAC  85 61     STA $61
. 4CAE  AD 0B 46  LDA $460B
. 4CB1  38        SEC
. 4CB2  E9 20     SBC #$20
. 4CB4  85 62     STA $62
. 4CB6  20 8F 43  JSR $438F
. 4CB9  20 65 42  JSR $4265
. 4CBC  4C AC 4A  JMP $4AAC
. 4CBF  20 58 54  JSR $5458
. 4CC2  20 A0 42  JSR $42A0
. 4CC5  A9 99     LDA #$99
. 4CC7  38        SEC
. 4CC8  ED 0B 46  SBC $460B
. 4CCB  4A        LSR
. 4CCC  4A        LSR
. 4CCD  8D D4 6B  STA $6BD4
. 4CD0  4C AC 4A  JMP $4AAC
. 4CD3  EA        NOP

. 4D00  20 67 C5  JSR $C567
. 4D03  A9 B0     LDA #$B0
. 4D05  8D AF 02  STA $02AF
. 4D08  A9 00     LDA #$00
. 4D0A  8D AE 02  STA $02AE
. 4D0D  8D B0 02  STA $02B0
. 4D10  8D AD 02  STA $02AD
. 4D13  A2 00     LDX #$00
. 4D15  8A        TXA
. 4D16  48        PHA
. 4D17  BD 58 68  LDA $6858.X
. 4D1A  8D B1 02  STA $02B1
. 4D1D  BD 00 53  LDA $5300.X
. 4D20  8D B2 02  STA $02B2
. 4D23  BD 00 68  LDA $6800.X
. 4D26  8D B3 02  STA $02B3
. 4D29  A9 00     LDA #$00
. 4D2B  8D B4 02  STA $02B4
. 4D2E  20 DA C0  JSR $C0DA
. 4D31  68        PLA
. 4D32  AA        TAX
. 4D33  E8        INX
. 4D34  E0 58     CPX #$58
. 4D36  D0 DD     BNE $4D15
. 4D38  A2 30     LDX #$30
. 4D3A  8A        TXA
. 4D3B  48        PHA
. 4D3C  A9 00     LDA #$00
. 4D3E  8D 96 4D  STA $4D96
. 4D41  8D B4 02  STA $02B4
. 4D44  8D B0 02  STA $02B0
. 4D47  A9 02     LDA #$02
. 4D49  8D 92 4D  STA $4D92
. 4D4C  20 89 4D  JSR $4D89
. 4D4F  8D B2 02  STA $02B2
. 4D52  8D AE 02  STA $02AE
. 4D55  A9 FF     LDA #$FF
. 4D57  8D 92 4D  STA $4D92
. 4D5A  20 89 4D  JSR $4D89
. 4D5D  8D B1 02  STA $02B1
. 4D60  8D AD 02  STA $02AD
. 4D63  A9 24     LDA #$24
. 4D65  8D 92 4D  STA $4D92
. 4D68  20 89 4D  JSR $4D89
. 4D6B  8D B3 02  STA $02B3
. 4D6E  8D AF 02  STA $02AF
. 4D71  20 DA C0  JSR $C0DA
. 4D74  68        PLA
. 4D75  AA        TAX
. 4D76  CA        DEX
. 4D77  10 C1     BPL $4D3A
. 4D79  60        RTS
. 4D7A  A2 1F     LDX #$1F
. 4D7C  A0 1F     LDY #$1F
. 4D7E  20 00 4A  JSR $4A00
. 4D81  A2 0F     LDX #$0F
. 4D83  A0 0F     LDY #$0F
. 4D85  20 0B 4A  JSR $4A0B
. 4D88  60        RTS
. 4D89  20 0E A7  JSR $A70E
. 4D8C  A0 04     LDY #$04
. 4D8E  B9 03 05  LDA $0503.Y
. 4D91  C9 24     CMP #$24
. 4D93  B0 05     BCS $4D9A
. 4D95  C9 00     CMP #$00
. 4D97  90 01     BCC $4D9A
. 4D99  60        RTS
. 4D9A  88        DEY
. 4D9B  10 F1     BPL $4D8E
. 4D9D  38        SEC
. 4D9E  B0 E9     BCS $4D89
. 4DA0  A0 00     LDY #$00
. 4DA2  98        TYA
. 4DA3  99 00 80  STA $8000.Y
```

```
. 4DA6  99 00 82 STA $8200.Y
. 4DA9  C8       INY
. 4DAA  D0 F7    BNE $4DA3
. 4DAC  60       RTS
. 4DAD  00       BRK
. 4DAE  00       BRK
. 4DAF  00       BRK
. 4DB0  A2 F8    LDX #$F8
. 4DB2  9A       TXS
. 4DB3  20 67 C5 JSR $C567
. 4DB6  A9 6B    LDA #$6B
. 4DB8  85 23    STA $23
. 4DBA  A9 DB    LDA #$DB
. 4DBC  85 22    STA $22
. 4DBE  A9 27    LDA #$27
. 4DC0  A2 04    LDX #$04
. 4DC2  A0 00    LDY #$00
. 4DC4  20 E2 41 JSR $41E2
. 4DC7  4C 8C 4B JMP $4B8C
. 4DCA  EA       NOP

. 4DE0  A9 53    LDA #$53
. 4DE2  8D 19 4D STA $4D19
. 4DE5  A9 54    LDA #$54
. 4DE7  8D 25 4D STA $4D25
. 4DEA  EA       NOP
. 4DEB  EA       NOP
. 4DEC  EA       NOP
. 4DED  EA       NOP
. 4DEE  EA       NOP
. 4DEF  A9 49    LDA #$49
. 4DF1  8D 0F 41 STA $410F
. 4DF4  8D 30 41 STA $4130
. 4DF7  A9 10    LDA #$10
. 4DF9  8D D8 4E STA $4ED8
. 4DFC  60       RTS
. 4DFD  EA       NOP
. 4DFE  EA       NOP
. 4DFF  EA       NOP

>4E00 A0 B8 C2 BD BD C0 C3 C4 : 8B==@CD
>4E08 C2 C0 B9 B4 AC A9 A2 A4 :B@94.)"$
>4E10 A8 B3 C0 C4 BE B1 AA A2 :(3@D>1*"
>4E18 A6 A4 AE A0 B2 B4 B0 AC :&$. 240.
>4E20 A0 B3 A8 A0 9C 90 80 84 : 3( ....
>4E28 94 B0 B3 AC B0 A9 A7 B0 :.03.0)'0
>4E30 A8 A8 A6 A8 A4 A1 A0 A9 :((&($! )
>4E38 A8 A4 98 94 90 96 97 9A :($......
>4E40 9D A1 A4 A2 A0 AA 8D A0 :.!$" *.
>4E48 90 90 94 98 9A A1 73 71 :.....!sq
>4E50 7D 91 7D 74 78 87 85 80 :}.}tx...
>4E58 02 06 0C 0E 24 26 28 30 :....$&(0
>4E60 31 33 39 3A 45 49 4A 4E :139:EIJN
>4E68 50 53 56 58 5C 5E 62 64 :PSVX\^bd
>4E70 68 6A 6C 70 74 76 78 7A :hjlptvxz
>4E78 7C 80 83 85 88 93 95 98 :|.......
>4E80 9A 9D A0 A3 A5 A6 A7 A8 :.. #%&'(
>4E88 AC C4 C6 C7 C9 CA CB CC :.DFGIJKL
>4E90 CD CF D1 D4 D7 D8 DA DE :MOQTWXZ^
>4E98 E2 E3 E6 E8 EC EE F1 F5 :bcfhlnqu
>4EA0 01 19 1A 1B 1C 1D 1E 1F :........
>4EA8 20 22 24 26 28 2A 30 38 : "$&(*08
>4EB0 4E 5A 5B 90 80 40 FF 00 :NZ[..@.
>4EB8 20 20 20 78 4C A0 FF FF :   xL
>4EC0 28 60 4C A8 98 48 FF FF :(`L(.H
>4EC8 A0 90 60 90 78 90 FF FF : .`.x.

>4ED0 D0 B0 70 FF FF FF FF FF :P0p
>4ED8 10 68 E0 00 50 C8 06 0A :.h`.PH..
>4EE0 1B 17 15 0B 00 F2 EE E8 :.....rnh
>4EE8 EE F6 00 0E 14 17 FF FF :nv....
>4EF0 00 0E 14 1C 1D 1D 14 00 :........
>4EF8 92 9D 9D 9C 94 8A FF FF :......

. 4F00  A9 00    LDA #$00
. 4F02  8D D2 6B STA $6BD2
. 4F05  AD 06 47 LDA $4706
. 4F08  CD 0B 46 CMP $460B
. 4F0B  90 6A    BCC $4F77
. 4F0D  AD D2 53 LDA $53D2
. 4F10  C9 55    CMP #$55
. 4F12  D0 63    BNE $4F77
. 4F14  AD B7 4E LDA $4EB7
. 4F17  10 19    BPL $4F32
. 4F19  68       PLA
. 4F1A  68       PLA
. 4F1B  20 67 C5 JSR $C567
. 4F1E  A9 6C    LDA #$6C
. 4F20  85 23    STA $23
. 4F22  A9 D8    LDA #$D8
. 4F24  85 22    STA $22
. 4F26  A9 27    LDA #$27
. 4F28  A2 04    LDX #$04
. 4F2A  A0 00    LDY #$00
. 4F2C  20 E2 41 JSR $41E2
. 4F2F  4C 8C 4B JMP $4B8C
. 4F32  20 E0 4D JSR $4DE0
. 4F35  68       PLA
. 4F36  68       PLA
. 4F37  AE B7 4E LDX $4EB7
. 4F3A  A0 01    LDY #$01
. 4F3C  4E 09 46 LSR $4609
. 4F3F  6E 0A 46 ROR $460A
. 4F42  88       DEY
. 4F43  10 F7    BPL $4F3C
. 4F45  AD 0A 46 LDA $460A
. 4F48  18       CLC
. 4F49  7D DB 4E ADC $4EDB.X
. 4F4C  90 03    BCC $4F51
. 4F4E  EE 09 46 INC $4609
. 4F51  48       PHA
. 4F52  AD 09 46 LDA $4609
. 4F55  48       PHA
. 4F56  AD E7 53 LDA $53E7
. 4F59  48       PHA
. 4F5A  20 00 4D JSR $4D00
. 4F5D  20 7A 4D JSR $4D7A
. 4F60  68       PLA
. 4F61  8D E7 53 STA $53E7
. 4F64  20 29 4A JSR $4A29
. 4F67  68       PLA
. 4F68  85 60    STA $60
. 4F6A  68       PLA
. 4F6B  85 61    STA $61
. 4F6D  A9 28    LDA #$28
. 4F6F  85 62    STA $62
. 4F71  20 65 42 JSR $4265
. 4F74  4C E7 50 JMP $50E7
. 4F77  AD 09 46 LDA $4609
. 4F7A  F0 4F    BEQ $4FCB
. 4F7C  AD 0A 46 LDA $460A
. 4F7F  CD 03 47 CMP $4703
. 4F82  D0 46    BNE $4FCA
```

```
. 4F84  AD DE 53 LDA $53DE
. 4F87  C9 52    CMP #$52
. 4F89  D0 3F    BNE $4FCA
. 4F8B  AD B7 4E LDA $4EB7
. 4F8E  10 03    BPL $4F93
. 4F90  4C B0 4D JMP $4DB0
. 4F93  C9 02    CMP #$02
. 4F95  F0 F9    BEQ $4F90
. 4F97  EE B7 4E INC $4EB7
. 4F9A  20 67 C5 JSR $C567
. 4F9D  AE B7 4E LDX $4EB7
. 4FA0  68       PLA
. 4FA1  68       PLA
. 4FA2  20 A0 4D JSR $4DA0
. 4FA5  BD B0 4E LDA $4EB0.X
. 4FA8  8D 19 4D STA $4D19
. 4FAB  8D 25 4D STA $4D25
. 4FAE  BD B3 4E LDA $4EB3.X
. 4FB1  20 05 4D JSR $4D05
. 4FB4  AD E7 53 LDA $53E7
. 4FB7  09 B0    ORA #$B0
. 4FB9  48       PHA
. 4FBA  A9 00    LDA #$00
. 4FBC  85 60    STA $60
. 4FBE  A9 01    LDA #$01
. 4FC0  85 61    STA $61
. 4FC2  A9 68    LDA #$68
. 4FC4  8D D8 4E STA $4ED8
. 4FC7  4C B2 51 JMP $51B2
. 4FCA  60       RTS
. 4FCB  AD 0A 46 LDA $460A
. 4FCE  CD 05 47 CMP $4705
. 4FD1  D0 F7    BNE $4FCA
. 4FD3  AD DE 53 LDA $53DE
. 4FD6  C9 4C    CMP #$4C
. 4FD8  D0 F0    BNE $4FCA
. 4FDA  AD B7 4E LDA $4EB7
. 4FDD  10 03    BPL $4FE2
. 4FDF  4C B0 4D JMP $4DB0
. 4FE2  F0 FB    BEQ $4FDF
. 4FE4  CE B7 4E DEC $4EB7
. 4FE7  20 67 C5 JSR $C567
. 4FEA  AE B7 4E LDX $4EB7
. 4FED  68       PLA
. 4FEE  68       PLA
. 4FEF  A9 68    LDA #$68
. 4FF1  8D D8 4E STA $4ED8
. 4FF4  4C 71 51 JMP $5171
. 4FF7  EA       NOP
. 4FF8  EA       NOP
. 4FF9  00       BRK
. 4FFA  00       BRK
. 4FFB  00       BRK
. 4FFC  00       BRK
. 4FFD  00       BRK
. 4FFE  00       BRK
. 4FFF  00       BRK
. 5000  A2 02    LDX #$02
. 5002  A9 A8    LDA #$A8
. 5004  9D 1A 67 STA $671A.X
. 5007  9D 22 68 STA $6822.X
. 500A  9D 2A 69 STA $692A.X
. 500D  9D 1A 6A STA $6A1A.X
. 5010  9D 2A 6B STA $6B2A.X
. 5013  9D 3A 6C STA $6C3A.X
. 5016  9D 1A 6D STA $6D1A.X
. 5019  9D 12 6E STA $6E12.X
. 501C  9D 2A 6F STA $6F2A.X
. 501F  CA       DEX
. 5020  10 E2    BPL $5004
. 5022  A9 01    LDA #$01
. 5024  8D F0 67 STA $67F0
. 5027  A9 30    LDA #$30
. 5029  8D F7 69 STA $69F7
. 502C  EA       NOP
. 502D  EA       NOP
. 502E  EA       NOP
. 502F  EA       NOP
. 5030  EA       NOP
. 5031  20 E0 4D JSR $4DE0
. 5034  A9 34    LDA #$34
. 5036  8D 04 45 STA $4504
. 5039  EA       NOP
. 503A  EA       NOP
. 503B  EA       NOP
. 503C  EA       NOP
. 503D  EA       NOP
. 503E  A9 A0    LDA #$A0
. 5040  8D 0E FF STA $FF0E
. 5043  A9 40    LDA #$40
. 5045  8D DF 4E STA $4EDF
. 5048  A9 7F    LDA #$7F
. 504A  8D 10 FF STA $FF10
. 504D  A9 02    LDA #$02
. 504F  8D 11 FF STA $FF11
. 5052  A9 CF    LDA #$CF
. 5054  8D 12 FF STA $FF12
. 5057  A9 30    LDA #$30
. 5059  A2 2A    LDX #$2A
. 505B  A0 32    LDY #$32
. 505D  8E BA 53 STX $53BA
. 5060  8E BB 53 STX $53BB
. 5063  8D B8 53 STA $53B8
. 5066  8D B9 53 STA $53B9
. 5069  8D CE 53 STA $53CE
. 506C  8D D0 53 STA $53D0
. 506F  8D DA 53 STA $53DA
. 5072  8D DC 53 STA $53DC
. 5075  8C C3 53 STY $53C3
. 5078  8C C2 53 STY $53C2
. 507B  8C C4 53 STY $53C4
. 507E  8C C5 53 STY $53C5
. 5081  A9 44    LDA #$44
. 5083  8D D2 53 STA $53D2
. 5086  A9 52    LDA #$52
. 5088  8D DE 53 STA $53DE
. 508B  A9 49    LDA #$49
. 508D  8D DF 53 STA $53DF
. 5090  A9 33    LDA #$33
. 5092  8D CF 53 STA $53CF
. 5095  A9 32    LDA #$32
. 5097  8D DB 53 STA $53DB
. 509A  A9 4F    LDA #$4F
. 509C  8D D3 53 STA $53D3
. 509F  A0 00    LDY #$00
. 50A1  98       TYA
. 50A2  99 00 80 STA $8000.Y
. 50A5  99 00 82 STA $8200.Y
. 50A8  C8       INY
. 50A9  D0 F7    BNE $50A2
. 50AB  A9 80    LDA #$80
. 50AD  8D F8 07 STA $07F8
```

```
. 50B0  8D 15 FF STA $FF15
. 50B3  8D 19 FF STA $FF19
. 50B6  A9 71    LDA #$71
. 50B8  8D 3B 05 STA $053B
. 50BB  85 86    STA $86
. 50BD  A9 01    LDA #$01
. 50BF  85 75    STA $75
. 50C1  85 84    STA $84
. 50C3  A9 20    LDA #$20
. 50C5  85 83    STA $83
. 50C7  A9 28    LDA #$28
. 50C9  85 87    STA $87
. 50CB  A9 19    LDA #$19
. 50CD  85 88    STA $88
. 50CF  20 00 4D JSR $4D00
. 50D2  20 7A 4D JSR $4D7A
. 50D5  78       SEI
. 50D6  8D 3F FF STA $FF3F
. 50D9  A9 00    LDA #$00
. 50DB  85 E5    STA $E5
. 50DD  20 F0 41 JSR $41F0
. 50E0  20 32 45 JSR $4532
. 50E3  8D 3E FF STA $FF3E
. 50E6  58       CLI
. 50E7  A9 FF    LDA #$FF
. 50E9  8D B7 4E STA $4EB7
. 50EC  20 00 40 JSR $4000
. 50EF  20 00 4F JSR $4F00
. 50F2  AD 0B 46 LDA $460B
. 50F5  C9 30    CMP #$30
. 50F7  B0 02    BCS $50FB
. 50F9  90 5B    BCC $5156
. 50FB  AD 04 46 LDA $4604
. 50FE  F0 03    BEQ $5103
. 5100  4C B0 5B JMP $5BB0
. 5103  AD 09 46 LDA $4609
. 5106  F0 02    BEQ $510A
. 5108  D0 45    BNE $514F
. 510A  AD 0A 46 LDA $460A
. 510D  C9 60    CMP #$60
. 510F  B0 0E    BCS $511F
. 5111  AD 0B 46 LDA $460B
. 5114  C9 A0    CMP #$A0
. 5116  B0 02    BCS $511A
. 5118  90 3C    BCC $5156
. 511A  A2 00    LDX #$00
. 511C  4C 68 51 JMP $5168
. 511F  C9 6B    CMP #$6B
. 5121  B0 02    BCS $5125
. 5123  90 31    BCC $5156
. 5125  C9 BC    CMP #$BC
. 5127  B0 0E    BCS $5137
. 5129  AD 0B 46 LDA $460B
. 512C  C9 58    CMP #$58
. 512E  B0 02    BCS $5132
. 5130  90 24    BCC $5156
. 5132  A2 01    LDX #$01
. 5134  4C 68 51 JMP $5168
. 5137  C9 E4    CMP #$E4
. 5139  B0 02    BCS $513D
. 513B  90 19    BCC $5156
. 513D  C9 F5    CMP #$F5
. 513F  B0 07    BCS $5148
. 5141  AD 0B 46 LDA $460B
. 5144  C9 37    CMP #$37
. 5146  B0 02    BCS $514A
. 5148  90 0C    BCC $5156
. 514A  A2 02    LDX #$02
. 514C  4C 68 51 JMP $5168
. 514F  AD 0B 46 LDA $460B
. 5152  C9 74    CMP #$74
. 5154  B0 F4    BCS $514A
. 5156  20 58 54 JSR $5458
. 5159  20 A0 42 JSR $42A0
. 515C  AD 11 FF LDA $FF11
. 515F  29 F0    AND #$F0
. 5161  09 02    ORA #$02
. 5163  8D 11 FF STA $FF11
. 5166  D0 84    BNE $50EC
```

```
>C000 00 00 3C 00 00 00 18 00 :..<.....
>C008 00 00 81 00 00 02 24 40 :......$@
>C010 00 00 3C 00 00 04 18 20 :..<.... 
>C018 00 00 00 00 00 00 00 00 :........
>C020 00 00 00 00 00 04 00 20 :....... 
>C028 00 00 00 00 00 00 24 00 :......$.
>C030 00 00 3C 00 00 02 18 40 :..<....@
>C038 00 00 00 00 00 00 00 00 :........
>C040 00 00 00 00 00 00 81 00 :........
>C048 00 00 00 00 00 00 24 00 :......$.
>C050 00 00 3C 00 00 00 18 00 :..<.....
>C058 00 00 00 00 00 00 00 00 :........
>C060 00 00 00 00 00 00 00 00 :........
>C068 00 00 00 00 00 00 00 00 :........
>C070 00 00 24 00 00 00 3C 00 :..$...<.
>C078 00 00 18 00 00 00 00 00 :........
>C080 00 00 00 00 00 00 00 00 :........
>C088 00 00 00 00 00 00 24 00 :......$.
>C090 00 00 3C 00 00 00 18 00 :..<.....
>C098 00 00 00 00 00 00 00 00 :........
>C0A0 00 00 00 00 00 00 00 00 :........
>C0A8 00 00 00 00 00 00 24 00 :......$.
>C0B0 00 00 3C 00 00 00 18 00 :..<.....
>C0B8 00 00 00 00 00 00 00 00 :........
>C0C0 00 00 00 00 00 00 00 00 :........
>C0C8 00 00 00 00 00 00 00 00 :........
>C0D0 00 00 00 00 00 00 00 00 :........
>C0D8 00 00 00 00 00 00 00 00 :........
>C0E0 00 00 00 00 00 00 00 00 :........
>C0E8 00 00 00 00 00 00 00 00 :........
>C0F0 00 00 00 00 00 00 00 00 :........
>C0F8 00 00 00 00 00 00 00 00 :........
>C100 00 00 80 00 00 03 A8 00 :.......(.
>C108 00 07 02 00 00 02 40 00 :......@.
>C110 00 00 E0 00 00 09 C0 00 :..`...@.
>C118 00 00 80 80 00 10 00 00 :........
>C120 00 00 00 00 00 10 00 00 :........
>C128 00 00 0C 40 00 00 2C 00 :...@....
>C130 00 04 3C 00 00 00 08 00 :..<.....
>C138 00 00 00 40 00 00 00 00 :...@....
>C140 00 01 00 00 00 00 00 80 :........
>C148 00 00 10 C0 00 00 05 C0 :...@...@
>C150 00 00 01 C0 00 00 00 00 :...@....
>C158 00 00 00 00 00 00 00 00 :........
>C160 00 00 00 00 00 00 00 00 :........
>C168 00 00 00 00 00 00 00 00 :........
>C170 00 00 80 00 00 00 80 00 :........
>C178 00 05 80 00 00 07 80 00 :........
>C180 00 01 00 00 00 00 00 00 :........
>C188 00 00 00 00 00 00 10 00 :........
>C190 00 00 90 00 00 00 F0 00 :.......D.
>C198 00 00 20 00 00 00 00 00 :.. .....
```

```
>C1A0 00 00 00 00 00 00 00 00 :........
>C1A8 00 00 01 00 00 00 0B 00 :........
>C1B0 00 00 0F 00 00 00 02 00 :........
>C1B8 00 00 00 00 00 00 00 00 :........
>C1C0 00 00 00 00 00 00 00 00 :........
>C1C8 00 00 00 00 00 00 00 00 :........
>C1D0 00 00 00 00 00 00 00 00 :........
>C1D8 00 00 00 00 00 00 00 00 :........
>C1E0 00 00 00 00 00 00 00 00 :........
>C1E8 00 00 00 00 00 00 00 00 :........
>C1F0 00 00 00 00 00 00 00 00 :........
>C1F8 25 0F 07 03 08 03 03 B0 :%......0
>C200 00 00 00 00 00 09 20 00 :...... .
>C208 00 18 00 00 00 38 04 00 :.....8..
>C210 00 72 00 00 00 03 00 00 :.r......
>C218 00 4F 00 00 00 06 01 00 :.O......
>C220 00 00 00 00 00 40 18 00 :.....@..
>C228 00 00 08 00 00 00 58 40 :......X@
>C230 00 20 78 00 00 00 00 00 :. x.....
>C238 00 08 00 00 00 00 00 60 :.......`
>C240 00 00 80 60 00 00 09 E0 :...`...`
>C248 00 00 01 C0 00 00 00 00 :...@....
>C250 00 00 00 00 00 00 00 00 :........
>C258 00 00 00 00 00 00 00 00 :........
>C260 00 00 00 00 00 00 00 00 :........
>C268 00 00 00 00 00 00 00 00 :........
>C270 00 10 00 00 00 18 00 00 :........
>C278 00 30 00 00 00 F0 00 00 :.0...p..
>C280 00 40 00 00 00 01 80 00 :.@......
>C288 00 00 80 00 00 05 80 00 :........
>C290 00 07 80 00 00 00 00 00 :........
>C298 00 00 00 00 00 00 06 00 :........
>C2A0 00 00 02 00 00 00 16 00 :........
>C2A8 00 00 1E 00 00 00 00 00 :........
>C2B0 00 00 00 00 00 00 00 00 :........
>C2B8 00 00 00 00 00 00 00 00 :........
>C2C0 00 00 00 00 00 00 00 00 :........
>C2C8 00 00 00 00 00 00 00 00 :........
>C2D0 00 00 00 00 00 00 00 00 :........
>C2D8 00 00 00 00 00 00 00 00 :........
>C2E0 00 00 00 00 00 00 00 00 :........
>C2E8 00 00 00 00 00 00 00 00 :........
>C2F0 00 00 00 00 00 00 00 00 :........
>C2F8 25 0F 07 03 08 03 03 B0 :%......0
>C300 00 00 00 00 00 00 00 00 :........
>C308 00 00 00 00 00 01 40 00 :......@.
>C310 00 04 08 00 00 20 00 00 :..... ..
>C318 00 24 00 00 00 76 01 00 :.$...v..
>C320 00 66 00 00 00 06 00 00 :.f......
>C328 00 0C 18 20 00 20 0C 00 :... . ..
>C330 00 00 0C 00 00 00 34 30 :......40
>C338 00 20 00 10 00 00 00 18 :. ......
>C340 00 08 00 70 00 01 01 20 :...p...
>C348 00 00 10 00 00 00 00 00 :........
>C350 00 00 00 00 00 00 00 00 :........
>C358 00 00 00 00 00 00 00 00 :........
>C360 00 00 00 00 00 00 00 00 :........
>C368 00 00 00 00 00 00 00 00 :........
>C370 00 00 00 00 00 00 00 00 :........
>C378 00 00 00 00 00 00 00 00 :........
>C380 00 00 00 00 00 60 00 00 :.....`..
>C388 00 30 00 00 00 30 00 00 :.0...0..
>C390 00 E1 80 00 00 00 C0 00 :.a....@.
>C398 00 01 C0 00 00 03 82 00 :..@.....
>C3A0 00 00 03 00 00 00 03 00 :........
>C3A8 00 00 07 00 00 00 0F 00 :........
>C3B0 00 00 00 00 00 00 00 00 :........
>C3B8 00 00 00 00 00 00 00 00 :........
>C3C0 00 00 00 00 00 00 00 00 :........
>C3C8 00 00 00 00 00 00 00 00 :........
>C3D0 00 00 00 00 00 00 00 00 :........
>C3D8 00 00 00 00 00 00 00 00 :........
>C3E0 00 00 00 00 00 00 00 00 :........
>C3E8 00 00 00 00 00 00 00 00 :........
>C3F0 00 00 00 00 00 00 00 00 :........
>C3F8 25 0F 07 03 08 03 03 B0 :%......0
>C400 00 00 00 00 00 00 00 00 :........
>C408 00 00 00 00 00 00 00 00 :........
>C410 00 00 00 00 00 00 04 40 00 :.......@.
>C418 00 10 04 00 00 00 00 00 :........
>C420 00 20 00 40 00 00 00 00 :. .@....
>C428 00 98 18 18 00 CC 0C 0C :.....L..
>C430 00 CC 0C 0C 00 98 18 18 :.L......
>C438 00 00 00 00 00 20 00 40 :..... .@
>C440 00 00 00 00 00 10 04 00 :........
>C448 00 04 40 00 00 00 00 00 :..@.....
>C450 00 00 00 00 00 00 00 00 :........
>C458 00 00 00 00 00 00 00 00 :........
>C460 00 00 00 00 00 00 00 00 :........
>C468 00 00 00 00 00 00 00 00 :........
>C470 00 00 00 00 00 00 00 00 :........
>C478 00 00 00 00 00 00 00 00 :........
>C480 00 00 00 00 00 00 00 00 :........
>C488 00 00 00 00 00 00 00 00 :........
>C490 00 00 00 00 00 00 00 00 :........
>C498 00 C1 81 80 00 60 C0 C0 :.A...`@@
>C4A0 00 60 C0 C0 00 C1 81 80 :.`@@.A..
>C4A8 00 00 00 00 00 00 00 00 :........
>C4B0 00 00 00 00 00 00 00 00 :........
>C4B8 00 00 00 00 00 00 00 00 :........
>C4C0 00 00 00 00 00 00 00 00 :........
>C4C8 00 00 00 00 00 00 00 00 :........
>C4D0 00 00 00 00 00 00 00 00 :........
>C4D8 00 00 00 00 00 00 00 00 :........
>C4E0 00 00 00 00 00 00 00 00 :........
>C4E8 00 00 00 00 00 00 00 00 :........
>C4F0 00 00 00 00 00 00 00 00 :........
>C4F8 00 00 00 00 00 00 00 00 :........
>C500 00 00 00 00 00 00 00 00 :........
>C508 00 00 00 00 00 00 00 00 :........
>C510 00 00 00 00 00 00 00 00 :........
>C518 00 00 22 38 00 02 00 78 :.."8...x
>C520 00 00 00 18 00 20 00 00 :..... ..
>C528 00 00 38 10 00 40 3C 00 :..8..@<.
>C530 00 00 08 20 00 48 18 00 :... .H..
>C538 00 1C 00 00 00 CE 00 00 :.....N..
>C540 00 64 00 80 00 70 00 00 :.d...p..
>C548 00 20 08 00 00 04 00 00 :. ......
>C550 00 01 40 00 00 00 00 00 :..@.....
>C558 00 00 00 00 00 00 00 00 :........
>C560 00 00 00 00 00 00 00 00 :........
>C568 00 00 00 00 00 00 00 00 :........
>C570 00 00 00 00 00 00 00 00 :........
>C578 00 00 00 00 00 00 00 00 :........
>C580 00 00 00 00 00 00 00 00 :........
>C588 00 00 00 00 00 00 00 00 :........
>C590 00 00 03 80 00 00 01 C0 :........@
>C598 00 00 00 00 80 00 00 01 80 :........
>C5A0 00 01 C0 00 00 00 60 00 :..@...`.
>C5A8 00 00 40 00 00 F0 C0 00 :..@..p@.
>C5B0 00 38 00 00 00 10 00 00 :.8......
>C5B8 00 30 00 00 00 00 00 00 :.0......
```

46

```
>C5C0 00 00 00 00 00 00 00 00 :........
>C5C8 00 00 00 00 00 00 00 00 :........
>C5D0 00 00 00 00 00 00 00 00 :........
>C5D8 00 00 00 00 00 00 00 00 :........
>C5E0 00 00 00 00 00 00 00 00 :........
>C5E8 00 00 00 00 00 00 00 00 :........
>C5F0 00 00 00 00 00 00 00 00 :........
>C5F8 25 0F 07 03 08 03 03 B1 :%.......1
>C600 00 00 00 00 00 00 00 00 :........
>C608 00 00 00 00 00 00 00 00 :........
>C610 00 00 00 00 00 00 01 C0 :........@
>C618 00 00 11 E0 00 00 00 60 :...'...'
>C620 00 01 00 60 00 00 00 00 :...'....
>C628 00 10 00 00 00 00 78 40 :......x@
>C630 00 00 58 00 00 40 08 00 :..X..@..
>C638 00 00 18 00 00 00 00 80 :........
>C640 00 46 00 00 00 0F 00 00 :.F......
>C648 00 03 00 00 00 72 02 00 :.....r..
>C650 00 38 00 00 00 18 08 00 :.8......
>C658 00 0A 40 00 00 00 00 00 :..@.....
>C660 00 00 00 00 00 00 00 00 :........
>C668 00 00 00 00 00 00 00 00 :........
>C670 00 00 00 00 00 00 00 00 :........
>C678 00 00 00 00 00 00 00 00 :........
>C680 00 00 00 00 00 00 00 00 :........
>C688 00 00 00 00 00 00 00 00 :........
>C690 00 00 00 00 00 00 1E 00 :........
>C698 00 00 16 00 00 00 02 00 :........
>C6A0 00 00 06 00 00 00 00 00 :........
>C6A8 00 00 00 00 00 07 80 00 :........
>C6B0 00 05 80 00 00 00 80 00 :........
>C6B8 00 01 80 00 00 40 00 00 :.....@..
>C6C0 00 F0 00 00 00 30 00 00 :.p...0..
>C6C8 00 18 00 00 00 10 00 00 :........
>C6D0 00 00 00 00 00 00 00 00 :........
>C6D8 00 00 00 00 00 00 00 00 :........
>C6E0 00 00 00 00 00 00 00 00 :........
>C6E8 00 00 00 00 00 00 00 00 :........
>C6F0 00 00 00 00 00 00 00 00 :........
>C6F8 25 0F 07 03 08 03 03 B1 :%.......1
>C700 00 00 00 00 00 00 00 00 :........
>C708 00 00 18 00 00 00 3C 00 :......<.
>C710 00 00 24 00 00 00 00 00 :..$.....
>C718 00 00 81 00 00 00 00 00 :........
>C720 00 00 00 00 00 00 00 00 :........
>C728 00 02 18 40 00 00 3C 00 :...@..<.
>C730 00 00 24 00 00 00 00 00 :..$.....
>C738 00 04 00 20 00 00 00 00 :... ....
>C740 00 00 00 00 00 00 00 00 :........
>C748 00 04 18 20 00 00 3C 00 :... ..<.
>C750 00 02 24 40 00 00 81 00 :..$@....
>C758 00 00 18 00 00 00 3C 00 :......<.
>C760 00 00 00 00 00 00 00 00 :........
>C768 00 00 00 00 00 00 00 00 :........
>C770 00 00 00 00 00 00 00 00 :........
>C778 00 00 00 00 00 00 00 00 :........
>C780 00 00 00 00 00 00 00 00 :........
>C788 00 00 18 00 00 00 3C 00 :......<.
>C790 00 00 24 00 00 00 00 00 :..$.....
>C798 00 00 00 00 00 00 00 00 :........
>C7A0 00 00 00 00 00 00 00 00 :........
>C7A8 00 00 18 00 00 00 3C 00 :......<.
>C7B0 00 00 24 00 00 00 00 00 :..$.....
>C7B8 00 00 00 00 00 00 00 00 :........
>C7C0 00 00 00 00 00 00 18 00 :........
>C7C8 00 00 3C 00 00 00 24 00 :..<...$.

>C7D0 00 00 00 00 00 00 00 00 :........
>C7D8 00 00 00 00 00 00 00 00 :........
>C7E0 00 00 00 00 00 00 00 00 :........
>C7E8 00 00 00 00 00 00 00 00 :........
>C7F0 00 00 00 00 00 00 00 00 :........
>C7F8 00 00 00 00 00 00 00 00 :........
>C800 00 00 00 00 00 00 00 00 :........
>C808 00 00 00 00 00 00 00 00 :........
>C810 00 00 00 00 00 00 03 80 :........
>C818 00 07 90 00 00 06 01 00 :........
>C820 00 06 00 00 00 00 00 10 :........
>C828 00 00 00 00 00 00 1E 04 :........
>C830 00 02 1A 00 00 00 10 00 :........
>C838 00 00 18 02 00 00 00 00 :........
>C840 00 00 80 60 00 00 00 F2 :...'...r
>C848 00 00 00 C0 00 00 00 4E :...@...N
>C850 00 00 20 1C 00 00 00 18 :.. .....
>C858 00 00 04 90 00 00 00 00 :........
>C860 00 00 00 00 00 00 00 00 :........
>C868 00 00 00 00 00 00 00 00 :........
>C870 00 00 00 00 00 00 00 00 :........
>C878 00 00 00 00 00 00 00 00 :........
>C880 00 00 00 00 00 00 00 00 :........
>C888 00 00 00 00 00 00 00 00 :........
>C890 00 00 00 00 00 00 78 00 :......x.
>C898 00 00 68 00 00 00 40 00 :..h...@.
>C8A0 00 00 60 00 00 00 00 00 :..'.....
>C8A8 00 00 00 00 00 00 01 E0 :.......'
>C8B0 00 00 01 A0 00 00 01 00 :... ....
>C8B8 00 00 01 80 00 00 00 02 :........
>C8C0 00 00 00 0F 00 00 00 0C :........
>C8C8 00 00 00 18 00 00 00 08 :........
>C8D0 00 00 00 00 00 00 00 00 :........
>C8D8 00 00 00 00 00 00 00 00 :........
>C8E0 00 00 00 00 00 00 00 00 :........
>C8E8 00 00 00 00 00 00 00 00 :........
>C8F0 00 00 00 00 00 00 00 00 :........
>C8F8 25 0F 07 03 08 03 03 B2 :%.......2
>C900 00 00 00 00 00 00 00 00 :........
>C908 00 00 00 00 00 00 00 00 :........
>C910 00 00 00 00 00 00 08 00 :........
>C918 00 04 80 80 00 0E 00 10 :........
>C920 00 18 00 00 00 08 00 04 :........
>C928 00 0C 2C 00 00 00 30 00 :......0.
>C930 00 00 30 04 00 04 18 30 :..0....0
>C938 00 00 00 60 00 00 00 66 :...'...f
>C940 00 00 80 6E 00 00 00 24 :...n...$
>C948 00 00 00 04 00 00 10 20 :.......
>C950 00 00 02 80 00 00 00 00 :........
>C958 00 00 00 00 00 00 00 00 :........
>C960 00 00 00 00 00 00 00 00 :........
>C968 00 00 00 00 00 00 00 00 :........
>C970 00 00 00 00 00 00 00 00 :........
>C978 00 00 00 00 00 00 00 00 :........
>C980 00 00 00 00 00 00 00 00 :........
>C988 00 00 00 00 00 00 00 00 :........
>C990 00 00 F0 00 00 00 E0 00 :..p...'.
>C998 00 00 C0 00 00 00 C0 00 :..@...@.
>C9A0 00 00 41 C0 00 00 03 80 :..A@....
>C9A8 00 00 03 00 00 00 01 87 :........
>C9B0 00 00 00 0C 00 00 00 0C :........
>C9B8 00 00 00 06 00 00 00 00 :........
>C9C0 00 00 00 00 00 00 00 00 :........
>C9C8 00 00 00 00 00 00 00 00 :........
>C9D0 00 00 00 00 00 00 00 00 :........
>C9D8 00 00 00 00 00 00 00 00 :........
```

```
>C9E0 00 00 00 00 00 00 00 00 :........
>C9E8 00 00 00 00 00 00 00 00 :........
>C9F0 00 00 00 00 00 00 00 00 :........
>C9F8 25 0F 07 03 08 03 03 B1 :%.......1
>CA00 00 00 00 00 00 00 00 00 :........
>CA08 00 00 00 00 00 00 00 00 :........
>CA10 00 00 00 00 00 00 02 20 :.......
>CA18 00 00 20 08 00 00 00 00 :.. .....
>CA20 00 02 00 04 00 00 00 00 :........
>CA28 00 18 18 19 00 30 30 33 :.....003
>CA30 00 30 30 33 00 18 18 19 :.003....
>CA38 00 00 00 00 00 02 00 04 :........
>CA40 00 00 00 00 00 00 20 08 :...... .
>CA48 00 00 02 20 00 00 00 00 :... ....
>CA50 00 00 00 00 00 00 00 00 :........
>CA58 00 00 00 00 00 00 00 00 :........
>CA60 00 00 00 00 00 00 00 00 :........
>CA68 00 00 00 00 00 00 00 00 :........
>CA70 00 00 00 00 00 00 00 00 :........
>CA78 00 00 00 00 00 00 00 00 :........
>CA80 00 00 00 00 00 00 00 00 :........
>CA88 00 00 00 00 00 00 00 00 :........
>CA90 00 00 00 00 00 00 00 00 :........
>CA98 00 01 81 83 00 03 03 06 :........
>CAA0 00 03 03 06 00 01 81 83 :........
>CAA8 00 00 00 00 00 00 00 00 :........
>CAB0 00 00 00 00 00 00 00 00 :........
>CAB8 00 00 00 00 00 00 00 00 :........
>CAC0 00 00 00 00 00 00 00 00 :........
>CAC8 00 00 00 00 00 00 00 00 :........
>CAD0 00 00 00 00 00 00 00 00 :........
>CAD8 00 00 00 00 00 00 00 00 :........
>CAE0 00 00 00 00 00 00 00 00 :........
>CAE8 00 00 00 00 00 00 00 00 :........
>CAF0 00 00 00 00 00 00 00 00 :........
>CAF8 00 00 00 00 00 00 00 00 :........
>CB00 00 00 00 00 00 00 00 00 :........
>CB08 00 00 00 00 00 00 02 80 :........
>CB10 00 00 00 20 00 00 10 04 :... ....
>CB18 00 00 00 0E 00 01 00 26 :.......&
>CB20 00 00 00 73 00 00 00 38 :...s...8
>CB28 00 00 18 12 00 04 10 00 :........
>CB30 00 00 3C 02 00 08 1C 00 :..<.....
>CB38 00 00 00 04 00 18 00 00 :........
>CB40 00 1E 00 40 00 1C 44 00 :...@..D.
>CB48 00 00 00 00 00 00 00 00 :........
>CB50 00 00 00 00 00 00 00 00 :........
>CB58 00 00 00 00 00 00 00 00 :........
>CB60 00 00 00 00 00 00 00 00 :........
>CB68 00 00 00 00 00 00 00 00 :........
>CB70 00 00 00 00 00 00 00 00 :........
>CB78 00 00 00 00 00 00 00 00 :........
>CB80 00 00 00 00 00 00 00 0C :........
>CB88 00 00 00 08 00 00 00 1C :........
>CB90 00 00 03 0F 00 00 02 00 :........
>CB98 00 00 06 00 00 00 03 80 :........
>CBA0 00 01 80 00 00 01 00 00 :........
>CBA8 00 03 80 00 00 01 C0 00 :.......@.
>CBB0 00 00 00 00 00 00 00 00 :........
>CBB8 00 00 00 00 00 00 00 00 :........
>CBC0 00 00 00 00 00 00 00 00 :........
>CBC8 00 00 00 00 00 00 00 00 :........
>CBD0 00 00 00 00 00 00 00 00 :........
>CBD8 00 00 00 00 00 00 00 00 :........
>CBE0 00 00 00 00 00 00 00 00 :........
>CBE8 00 00 00 00 00 00 00 00 :........
>CBF0 00 00 00 00 00 00 00 00 :........
>CBF8 25 0F 07 03 08 03 03 B2 :%.......2
>CC00 00 00 00 00 00 00 02 50 :.......P
>CC08 00 00 00 10 18 00 00 00 1C :........
>CC10 00 00 00 40 4E 00 00 00 C0 :...@N...@
>CC18 00 00 00 F0 00 00 00 62 :.......b
>CC20 00 01 00 00 00 00 18 00 :........
>CC28 00 00 10 02 00 00 1A 00 :........
>CC30 00 02 1E 00 00 00 00 08 :........
>CC38 00 00 00 00 00 06 00 80 :........
>CC40 00 06 00 00 00 07 88 00 :........
>CC48 00 03 80 00 00 00 00 00 :........
>CC50 00 00 00 00 00 00 00 00 :........
>CC58 00 00 00 00 00 00 00 00 :........
>CC60 00 00 00 00 00 00 00 00 :........
>CC68 00 00 00 00 00 00 00 00 :........
>CC70 00 00 00 08 00 00 00 18 :........
>CC78 00 00 00 0C 00 00 00 0F :........
>CC80 00 00 00 02 00 00 01 80 :........
>CC88 00 00 01 00 00 00 01 A0 :........
>CC90 00 00 01 E0 00 00 00 00 :.......`....
>CC98 00 00 00 00 00 00 60 00 :......`.
>CCA0 00 00 40 00 00 00 68 00 :..@...h.
>CCA8 00 00 78 00 00 00 00 00 :..x.....
>CCB0 00 00 00 00 00 00 00 00 :........
>CCB8 00 00 00 00 00 00 00 00 :........
>CCC0 00 00 00 00 00 00 00 00 :........
>CCC8 00 00 00 00 00 00 00 00 :........
>CCD0 00 00 00 00 00 00 00 00 :........
>CCD8 00 00 00 00 00 00 00 00 :........
>CCE0 00 00 00 00 00 00 00 00 :........
>CCE8 00 00 00 00 00 00 00 00 :........
>CCF0 00 00 00 00 00 00 00 00 :........
>CCF8 25 0F 07 03 08 03 03 B3 :%.......3
>CD00 00 00 00 00 00 00 01 80 :........
>CD08 00 00 49 E0 00 00 00 80 :..I`....
>CD10 00 01 04 00 00 00 07 D0 :........P
>CD18 00 00 03 80 00 02 00 08 :........
>CD20 00 00 00 00 00 00 00 08 :........
>CD28 00 00 20 00 00 04 24 00 :.. ...$.
>CD30 00 00 1C 10 00 00 38 00 :......8.
>CD38 00 00 00 00 00 02 00 80 :........
>CD40 00 00 00 00 00 01 00 00 :........
>CD48 00 03 24 00 00 01 E0 00 :..$...`.
>CD50 00 00 80 00 00 00 00 00 :........
>CD58 00 00 00 00 00 00 00 00 :........
>CD60 00 00 00 00 00 00 00 00 :........
>CD68 00 00 00 00 00 00 00 00 :........
>CD70 00 00 01 00 00 00 01 20 :.......
>CD78 00 00 01 E0 00 00 00 C0 :...`...@
>CD80 00 00 00 00 00 00 00 00 :........
>CD88 00 00 08 00 00 00 0D 00 :........
>CD90 00 00 0F 00 00 00 06 00 :........
>CD98 00 00 00 00 00 00 00 00 :........
>CDA0 00 00 80 00 00 00 C0 00 :......@.
>CDA8 00 00 F8 00 00 00 F0 00 :..x...p.
>CDB0 00 00 00 00 00 00 00 00 :........
>CDB8 00 00 00 00 00 00 00 00 :........
>CDC0 00 00 00 00 00 00 00 00 :........
>CDC8 00 00 00 00 00 00 00 00 :........
>CDD0 00 00 00 00 00 00 00 00 :........
>CDD8 00 00 00 00 00 00 00 00 :........
>CDE0 00 00 00 00 00 00 00 00 :........
>CDE8 00 00 00 00 00 00 00 00 :........
```

COMMODORE C16

Plus/4 Koerier

REDAKTIE: RONALD & GERARD DE BRUIN - UITGEVER: "RONSOFT HOLLAND UNLIMITED" - ADRES: HYACINTHSTRAAT 8 32 61 XD OUD-BEYERLAND

Oud-Beyerland, 19 May 1991.

HI THERE ROY!

Here at last another note from the other side of the water.
From my dad I heard you tried to reach me on the telephone yesterday. It was a pity that I wasn't at home at the time.
Thanx a lot to you and Andy for mentioning RONSOFT in the Mag (we're very honoured indeed!).
Yes, it's really true about GEOS and PAOS, being converted for the Plus/4! Even Printfox and Pagefox are converted!
Also quite a lot of famous games, such as Bard's Tale III, Tass Times, Borrowed Time and even Battle Chess (!) are on Plus/4, thanks to "PIGMY" in Hungary who converted them as DD 1541-versions, and "CEEKAY" in Germany, who made these conversions suitable for 1551-drives.
Some other converted games: Elite, Tetris, Barbarian 1 & 2, Freddie Hardest, The Goonies, Soccer Director, Blue Angel, Puzzle Shuffle, Atomix, Spy vs Spy 3, Karateka and (from ZX Spectrum) Head over Heels.
In Hungary several (former) crackers are coding new games themselves too now. (and really very good stuff too!)
At this moment Hungary is to be the leading Plus/4-country in the whole world. It still has a lively Plus/4-scene and the standard of coding is very high, as you could see with Digital Ball. (The sound in it is an actual conversion from Amiga 500).
However, like in Germany, in Hungary a lot of Plus/4 freaks are changing to Amiga now. A big difference with Germany is that in Hungary also a "second Plus/4 generation" is rising, as not everyone can afford to buy an Amiga. (Over there the average monthly wages are £ 70.--, while Amigas cost about £ 360.--).
At the moment we've several contacts in Germany and Hungary (RONSOFT is even in some 'greeting-lists' in scroll-texts of some demos and intros), and from time to time we get in some of the latest games and demos.
Games, demos or utilities (how about a 90 seconds self-formatting disk-copier for 1541, or 9 seconds format!) there is plenty to be had.
There are also programs for sprite & sound conversion from C64 to Plus/4 and for creating, changing and printing graphics.
There are even programs, which allow you to create your own demos and intros very easily, without any knowledge of coding.
Anyway, some stuff will be on its way to you soon, so you can see for yourself and tell the other freax about it in the mag.
Signing with many Plus/4-greetings, and wishing you Good Byte from Holland,

Ronald & Gerard de Bruin,
RONSOFT HOLLAND UNLIMITED

P.S.: Enclosed £ 5.- (subscription for April, May, June & July).

V.Berzins
193 Gorsemoor Rd.
Heath Hayes
Cannock
WS12 5HR

16.7.91.

Dear Roy.

I just received Your magazine, thanks. I enclose couple adverts from Germany for Plus/4 with english translations. A short disk directory tip for the magazine, translated from one of the disks sent to me by Mr De Bruin.

The reason I write is, that I am sending You a cheque, value £ 5.00 as payment for Mr.R. De Bruin's magazine subscription continuation.

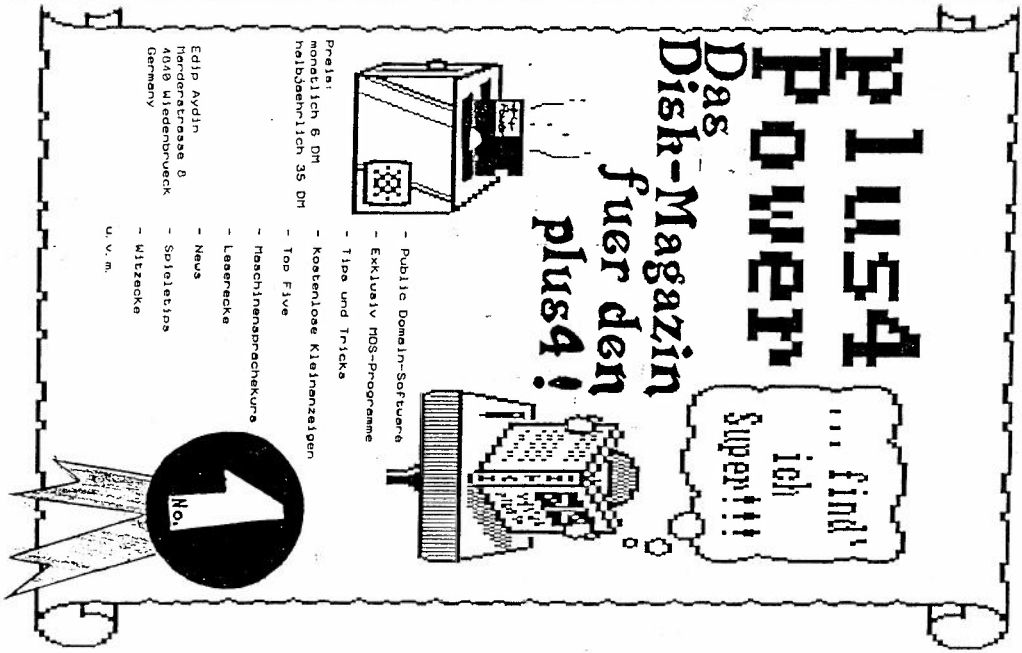
Your Bill

DISC DIRECTORY TIP.

How to save (:) behind the program name in the Directory, so that by LOADing from Directory it is only necessary to place Cursor over the number of blocks in the file, press F2 (DLOAD) and RETURN. To do this, save the file like this: DSAVE''File name (SHIFT SPACE) :. After the SHIFT SPACE do not forget the colon [:]. Directory example:

```
 2  ''DISC TIP'':       PRG
```

50



PLUS/4 POWER
...I find it superb!!!
THE DISK-MAGAZIN FOR PLUS/4!

Price:	-Public Domain-Software
Monthly 6 DM	-Exclusive MOS-Programs
Halfyearly 35 DM	-Tips and Tricks
	-Free small adds
	-Top five
	-Machine language course
	-Reader corner
Edip Aydin	-News
Marderstrasse 8	-Game tips
4840 Wiedenbrueck	-Joke corner
Germany	e.c.t.

Für jeden Plus/4-Benutzer stellt sich die Frage,wo man Informationen über und zu diesem Computersystem bekommen kann. Die meisten Zeitungen berichten gar nicht mehr über den Plus/4, andere nur sehr selten.
Um dieses Informationsdefizit abzubauen gibt es seit dem 1.November 1988 ein neues Informationsorgan für alle Plus/4-User:

PLUVI, das Plus/4-Magazin auf Diskette

PLUVI erscheint alle 6 Wochen.
Auf jeder (zweiseitigen!) Diskette gibt es Informationen, Berichte, Tips & Tricks, Tests kommerziell vertriebener Hard- und Software und natürlich auch einige gute Programme. Diese befinden sich sofort auf Diskette, ganz ohne das bei Zeitschriften übliche Abtippen!
Bei den Programmen handelt es sich um ausgesuchte Public Domain Software aus Ungarn, USA und Deutschland, Eigenentwicklungen und Konvertierungen vom C64. Für jeden ist etwas dabei: Spiele, Anwenderprogramme, Utilities, ...
Alle Programme sind natürlich ausführlich dokumentiert!

Und das alles gibt es sehr preiswert:
PLUVI kostet nur 10,- DM !!!
Im Abo gibt es PLUVI noch billiger: 3 Ausgaben kosten nur 24.-DM, 6 Ausgaben gibt es schon für 45,-DM und 12 Ausgaben für 84,-DM, pro Ausgabe also für 7,-DM !!!

Aus Kosten- und Zeitgründen ist nur Vorkasse möglich: bar, Verrechnungsscheck, Überweisung

Volker Huppert
Ferlingsweg 15
4150 Krefeld 1
Kto.Nr.: 1587 58-435, BLZ 360 100 43, Postgiroamt Essen

PLUVI

THE PLUS/4 MAGAZINE ON DISKETTE.

Every Plus/4 user asks himself the question, wher one can obtain information suitable for this computer Most magazines report absolutely nothing abou Plus/4, few others only seldom.

To overcome this information deficit, there is sinc 1. November 1988 a new information organ for al Plus/4 users.

PLUVI, the Plus/4 magazin on disk.

PLUVI is published every 6 weeks.

On every (doublesided!) disk are info, reports, tip & tricks, tests of commercial Hard- and Software, an naturaly also some good programs. You will find thes on the disk, without the usually necessary typin required by printed magazines!

These programs consist of choosen Public Domai Software from Hungary, USA and Germany, our ow developement and conversions from C64. There ar something for everybody: games, practical programs utilities,...

All programs are cleary explained!

And all this are quite inexpensive:

PLUVI cost only DM 10.-!!!

With subscription PLUVI are still cheaper: 3 issue cost only DM 24-,6 issues for oonly DM 45.- and 1 issues for DM 84.-, t.i. DM 7.- per issue!!!

Because of expense and use of time, pre-payment onl is acceptable: cash, crossed cheque or money transfer

Volker Huppert
Ferlingsweg 15
4150 Krefeld 1
Germany
Kto.Nr.:1587 58-435, BLZ 360 100 43,
Postgiroamt Essen.

64 Westwood cres,
Winton,Eccles,
Manchester,
M308DZ.

Dear Roy/readers,

There are a few things I wish to say.The first is a request .Could anyone who has a copy of 'Catacombs' or 'Bukarroo Banzai'and has either completed it/them ,or has achieved anything in them,however small,I would be most grateful to you if you get in touch.Now,by my opening you might be forgiven for your Pyrrhonism,at my next statement.The thing is that I have quite a lot of adventures for the c16/+4 ,and the few I havn't completed I've just tired with for the moment.This means,however un-believable it may seem,I shoul be able to help you on your adventuring problems.I'll list the adventures I have.However if game you are having problems with isn't listed don't be dismayed I may still be able to help you.What you could then do is either,send me a description of the game ,a description of some of the situations you have been in(including the one you are stuck in)and a list of commands you have already tried, I may then be able to supply you with a few more commands.Alternaatively,a system which would be more beneficial to us both,send me a copy of the game(or the original)and I'll have a tamper with it.That way I'll be able to see what style the game has been written in ,and have a poke around in the listings and be able tto supply you with the information you require much quicker,not only making you happy,but also me because I'd have the chance to play an adventure new to me.I promise to send all games back,with solutions,and anything I uncover I'll send to Roy for him to publish,as making induvidual replies would probably cripple my bank account.

Now just a few other things,a search for software and a manual for the MPS801 Printer.I would be most grateful if anyone could send me copies of the following:

'ACE 2'(for the Plus4)'
'Tir Na Nog''(again for the Plus4)
Any of the'Zork'adventures(for the Plus4)
Any adventure I havn't got(on c16 or Plus4)

By the way I havn't a disk drive so games would have to be on tape.

Now for a list of those adventures I already own:
'Spiderman,Pirate Adventure,Strange Odyssey,Voodoo Castle,Buckaroo Banzai,Catacombs,Classic Adventure,Circus,The Tower,The Monster Returns,Time Machine,Escape from Pulsar7,Treasure Hunter,Salvage,Ten Little Indians,Mansion Adventure1,LoneSurvivor,Twin kingdom Valley,Feasability Experiment,Golden Baton,Waxwoorks and lastly Arrow of Death Part One.'

Hope to hear from someone soon,
yours faithfully,

Neville Ó Cleirigh
Neville O Cleirigh.
(Neville Clarke)

5-2

LOAD IN THE FOLLOWING,RUN IT,TYPE NEW.TYPESYS1536.LOAD FINDERS KEEPERS A NORMAL.

?

```
10 FORI=1536TO1578:READA:POKEI,A:S=S+A:N
EXT:IFS<>74368THEN PRINT"DATA ERROR"
20 DATA120,169,13,141,20,3,169,6,141,21,
3,88,96,162,0,189
30 DATA29,6,157,93,1,232,224,14,208,245,
76,14,206,169,234,141,81,47,141,82
40 DATA47,141,83,47,76,202,31
```

LOST FOR WORDS IN AN ADVENTURE? SIMPLY LOAD IN THE ADVENTURE YOU'RE STUCK IN,RESET,ENTER TTHE FOLLOWING POKES,LOAD IN THE FOLLOWING LISTING, RUN IT,THEN SIT BACK AND READ. "P"WILL PAUSE,"R"WILL RESTART.

```
POKE52,18:POKE56,18

10 FORI=1536TO1578:READA:POKEI,A:S=S+A:N
EXT:IFS<>74368THEN PRINT"DATA ERROR"
20 DATA120,169,13,141,20,3,169,6,141,21,
3,88,96,162,0,189
30 DATA29,6,157,93,1,232,224,14,208,245,
76,14,206,169,234,141,81,47,141,82
40 DATA47,141,83,47,76,202,31
```

Dear Roy
I enclose a small Pattern Prog taken from a CBM PET which I was browsing thru in the local library. I think the color command works as I have only a B/W TV (SOMETHING I ADDED).

Yours
Peter Appleby, NOTTS.

```
5 SCNCLR
10 N=10
20 DIM P(N)
25 FOR I=1 TO N
30 READ P(I)
35 NEXT I
40 DATA 32,46,58,45,124,95,127,251,120,160
45 FOR R=1 TO 25
50 FOR C=1 TO 40
55 H=(R*C)^(1/3)
60 I=INT(H)
65 Q=3072+40*(R-1)+C-1
70 POKE Q,P(I)
75 X=INT(16*RND(1))+1
80 Y=INT(08*RND(1))
90 COLOR 1,X,Y
100 NEXT C
110 NEXT R
120 GOTO 120
```

==========

Dear Roy
After typing in GRAPHIC EDITOR (AUG/SEPT 90) P15. The Prog will not run properly, obviously I've checked/re-checked line by line to no avail.
The line causing the problem is:
1380 CHAR ,D%,0,"[8SPC]",1:GSHAPE S1$,DT%,0 ? NO GRAPHICS AREA IN 1380.
Any Ideas, please, so I'll recheck again.
Thanks for any help.

Peter Appleby, NOTTS

Well Peter Regarding Fidelity PSU, it will not work without alterations, and the tape decks are about £25-£35 now, its a bit expensive, so if you want I'll buy the +4 off you (phone me about it, 0964-534611). Disk drives, well I'm always looking out for them, but if anyone has a 1541/1551 for sale write to Peter Appleby, 18 Abbey Rd, Newstead, NOTTS, NG15 0BL. Peter Phone me on the Question of 'TESTING' a computer, I'm not sure what you mean. Regarding the Graphic Editor problem, I'm at a loss, and I think you'll have to ring Rob Marshall the Author of Graphic Editor, or send me the tape with the program and let me go through it. Rob's Tel No is: 0622-871066, I'm sure he would like to hear from you.

Dear Roy, The following C64 program was in YC April 1989 (p.48) all I've done is altered LN 60075. I've typed it in my C16 and it works OK - one day I'll possess a PRINTER!
Yours
Peter Appleby, NOTTS.

```
1  Z$="THE MAIN TITLE CAN BE 40 CHARACTERS LONG"
2  Z=10
3  Z$(1)= "FIRST MENU ENTRY"
4  Z$(2)= "THE NEXT MENU ENTRY"
5  Z$(3)= "THE THIRD MENU ENTRY"
6  Z$(4)= "MENU OPTION 4"
7  Z$(5)= "NUMERO 5"
8  Z$(6)= "SIXTH"
9  Z$(7)= "OPTION 7 USE THE MAXIMUM 39 CHARACTERS"
10 Z$(8)= "THE     EIGHTH     MENU     ENTRY"
11 Z$(9)= "YET ANOTHER OPTION"
12 Z$(10)="AT LAST! NUMBER 10"
13 GOSUB 60050
14 PRINT "[CLR]":IF Z=7 THEN PRINT "ONLY GREEDY PEOPLE SELECT OPTION"Z:GOTO
16
15 PRINT "AH! OPTION NUMBER"Z"[LEFT]. A WISE CHOICE."
16 END
60050 Z2=LEN(Z$):IF Z=0 OR Z2>40 THEN STOP
60055 PRINT "[CLR]":ZZ=(40-Z2)/2:PRINT SPC(ZZ)Z$;:IF Z2<40 THEN PRINT
60060 PRINT SPC(ZZ);:FOR Z1=1 TO LEN(Z$):PRINT "[SHIFT, E]";:NEXT:IF Z2<40
THEN PRINT
60065 PRINT "[HOME][CD22][COMMODORE, P40]";
60070 PRINT "[RVSON] HIGHLIGHT ANY OPTION WITH THE CSRS KEY ";
60075 PRINT "      THEN USE THE <RETURN> KEY TO SELECT IT";:POKE 4071,160
60080 ZZ%=7-Z/2:ZZ$="[CR10]"
60085 PRINT "[HOME]":FOR Z1=0 TO ZZ%:PRINT "[CD]";:NEXT
60090 FOR Z1=1 TO Z:Z$="":Z1=LEN(Z$(Z1)):IF Z2>7 THEN STOP
60095 IF Z2/2=INT(Z2/2=INT(Z2/2) THEN Z2=Z2+1
60100 IF Z2<39 THEN Z$=LEFT$(ZZ$(41-Z2)/2)
60105 FOR Z3=Z2 TO 1 STEP -2
60110 PRINT "CU"Z$,LEFT$(Z2$,Z3$);
60115 PRINT MID$(Z$(Z1),Z3/2+.5,Z2-Z3+1):NEXT
60120 IF Z1<>Z THEN PRINT:PRINT
60125 Z$(Z1)=Z$+Z$(Z1):NEXT
60130 ZZ$="[HOME][CD21]"
60135 Z1=1:PRINT "[RVSON]LEFT$(ZZ$,ZZ%+2*Z1)Z$(1)
60140 GET Z1$:IF Z1$="[CU]" THEN GOSUB 60160
60145 IF Z1$="[CD]" THEN GOSUB 60180
60150 IF Z1$=CHR$(13) THEN Z=Z1:RETURN
60155 GOTO 60140
60160 PRINT LEFT$(ZZ$,Z2%+2*Z1)Z$(Z1)
60165 Z1=Z1-1:IF Z1=0 THE Z1=Z
60170 PRINT "[RVSON]"LEFT$(ZZ$,ZZ%+2*Z1)Z$(Z1)
60175 RETURN
60180 PRINT LEFT$(ZZ$,ZZ%+2*Z1)Z$(Z1)
60185 Z1=Z1+1:IF Z1=Z+1 THEN Z1=1
60190 PRINT "[RVSON]"LEFT$(ZZ$,ZZ%+2*Z1)Z$(Z1)
60195 RETURN
```

56

Dear Roy,
On this new +4 of mine, I put the CHESS tape in typed LOAD "",1,1 <RETURN>. Would not run, so I went into MONITOR and tried other variations, still no joy. So I exchanged for the C16, typed LOAD "",1,1 <RETURN>, instant success, but when I type SYS65418 as on P.22 of the start no's (JAN/FEB 1991) no start; but SYS8192 did ok well such is life!(C16)

On the subject of start numbers, in the back of this manual I accquired, someone's wrote BANDITS @ ZERO SYS5540, TWIN KINGDOM VALLEY SYS12288, but I can't verify if they are correct.

I've retyped the CRIBBAGE prog in, it runs a little but then it says ?BAD SUBSCRIPT ERROR IN 3270, I've re-checked the listings but I can't see anything wrong, I'll re-check, but is there a way to find out where the fault is. I typed ?PEEK(63)+PEEK(64)*256 but it was OK.

This +4 of mine has a definite fault in it, I'm sure, I typed in LUNAR LANDER $4000 - $4119 SAVED it, all OK, is said; but when I typed VERIFY "LUNAR LANDER",01 (MONITOR) nothing happened, so what do you reckon; is there a fault or have I not done something right! - again.

Thanks for any help
Peter Appleby, NOTTS.

CHESS:
Do LOAD "",1,1 then when its loaded do POKE 65299,17:RUN, if this still does'nt work then let me know. As for Tape to Disk backup, I run through the sequence again in the October/November Mag. As for the SYS65418 on page 22 of the Jan/Feb '91 ish, I could'nt find it!! Cribbage, I've enclosed a tape with the working program on, ENJOY!!! Don't know anything about the PEEK(63)..... can anybody help?? Lunar Lander, It should work, but Peter give Peter Crack a ring on 081-367-3152 after 7pm, and he'll be pleased to go through any problems regarding his programs. If known of the above succeed, then send the +4 to me and I'll check it over for you, FREE of charge, but you'll have to pay P&P both ways.

==========

Dear Roy
Could you please let me know what the command : POINT 2, is on the C16/+4.

Thanks for any help
Peter Appleby, NOTTS.

Peter I'm afraid I don't know, but if any other reader does will the please write to Peter at, 18 ABBEY RD, NEWSTEAD, NOTTS, NG15 OBL.

NB:- Peter is after a PSU & Tape-deck for the +4 if any one has one To spare and would like to sell them to him, or knows of anybody who can sell him one (relatively cheaply) then please write to him.

Game Review

Game Reviewed - Daily Thompson's Start Events
Publisher - Ocean
Price - £2.99 (I bought if for £7.95 in 1985, ED)
Reviewer - Martin Sulivan (Ex Member)

As you can see this game is published by Ocean, this is rare for a C16-Plus/4 game, but back to the game...

Daley Thompsons Star Events is a game much like 'STREET OLYMPICS' (thats because street olympics was ripped using DTSE, ED), you have to control Daley though 7 events, which are...

On Side 1 - 100 Metres, 110 Metre Hurdles, 400 Metres, 200 Metre Hurdles

On Side 2 - Long Jump, Shot Pot & Javelin

For all of the events you need to continually move the joystick from left to right, which is very tiring (does'nt do the J/Stick much good either, ED), and frustrating if you don't do very well and then not reach the qualifying times.

Scoring
On screen scoring shows the number of attempts made, points scored, world records, qualifying times and distances.

All in all this is a disapointing game (I thought it was good for its time & still is, ED), which I expected to be quite good. Saying all that, it does have good graphics & music.

Ratings as a %

GRAPHICS	97%
SOUND	80%
PLAYABILTY	80%
VFM	49%
OVERALL	80%

Game Review

Game Reviewed - Auf Wiedersehen Monty
Publisher - Gremlin Graphics
Price - £2.99
Reviewer - Martin Sullivan (Ex Member)

Auf Weidersehen Monty is p;qart of a collection of 'Monty' games, including 'Monty On the Run'. In this adventure, Monty travels around Europe to find freedom, picking up all of the travellers cheques as he goes, in order to raise enough money to buy the Greek Island of Montos where he will be safe.

The game has good graphics, excellent sound, with an on-running music tune and excellent playability, with hours of fun and frustration.

Definetely one of my best games!!! And worth £2.99

Ratings as %

Graphics - 95%
Sound - 95%
Playability - 95%
VFM - 95%
OVERALL - 95%

Game Review

Title: TIR NA NOG
Publisher: PD - Germany
Requirments: 64K (ie, +4 or C16+64K)
Reviewer: Andy Tang (London)

The majority of the C16/+4 games that are available are 'arcade' games, theres a few 'adventures' and a handful of 'arcade adventures'. The so called combination of an adventure (the qualities of depth and puzzles that test your brains) with arcade action (attractive graphics & sonics, as well as addictive gameplay).

So far I can only think of two really 'outstanding' arcade adventures available for the +4, Mercenary & Tir Na Nog. I would'nt be surprised if any other members disagrees with my choice: we all have different opionions!

What makes Tir Na Nog special is its depth, its one of those games you have to map or you'll lose your bearings pretty fast. Its also fairly unique as you can view the game from different perspectives (ie, front, back and two side views)- the reason for this is to make the game three dimensional whilst still retaining the 'side view' graphics. Initially thus may be confusing, but the longer you play the game; the more you get use to it. It is vital to learn how the different views work if the game is to be played properly.

Usually when I write reviews I include information about the aim of the game, its scenery and the plot, but for this game I can't, since I don't have the instructions. What I think you have to do is to successfully complete some kind of quest (or quests) by collecting objects from different locations and manipulating them.

So far in my travels I have accquired many objects ranging from scrolls. keys to a very useful mace! I've also met some strange characters that inhabit the land like the forest king and the rather nasty ape-like creature who likes to run you over - I've tried everything to defeat thus beast, from bashing it on the head with a mace (timing is crucial here) to offering it a scroll - but it always leaves me steam rolled!

The character you control in the game has a very funny walk - nicely drawn with splendid animation, in fact the animation throughout the game is excellent.

The game also has many backgrounds with different types representing different locations; ie, nice mountain ranges in the plains to detailed trees in the forest. The game features many buildings which you can enter, some are locked so finding keys is essential, inside some are useful objects which are all nicely drawn. Colour is weel used in the game.

With most of the machines memory devoted to graphics and game-depth, its not really surprising that thers a shortage of sound FX, but that does'nt matter much as the graphics do more than enough to compensate!

My only complaints with this game are no instructions and the lack of an option to save the game like the one used in mercenary (or maybe its because I have'nt discovered it yet!)

Tir Na Nog is a truely good arcade adventure and whats more its a PD Game Thats 100% value.

Well done to the programmer(s) of this game.

Another point is that Tir Na Nog is just an example of how good PD games can be - I would but it even if it were a full price commercial product!

I feel that the game appeals more to the adevnturers than the arcade freaks - but I still fully recommend Tir Na Nog to ALL +4 owners.

Ratings:

Graphics	10
Sound	4
Playability	9
Value	10
OVERALL	10

Future Thoughts

The quality of arcade adventures (as well as other types of games) will get better and better, since in most European Countries the +4 is favoured over the C16 - Many games are in 64K format (ie, more sophisticated). With games like 'ELITE' and 'TIR NA NOG' already available I think all us +4 (& expanded C16) owners have something good to look forward to...

Tir Na Nog

B = Move Left
N = Move Right
H = Rotate VIEW Clockwise
G = Rotate VIEW Anticlockwise
R = Drop Item
E = Take Item
SPACE = Use item (ie, swing mace)

*NB: These controls are repeated on other keys, which I shall leave for you to discover (find the layout that suites you best!!)

Publisher Profile
NOVAGEN

I got this idea from an old club I used to belong to, each article will feature a publisher of C16 or +4 software - what they have done anmd what they are currently doing (if any!!!).

I decided to start with Novagen because they have only produced two games and its late now, what more can I say apart from.....

Novagen is one of those companies who jumped on the C16/+4 bandwagon back in 1985/86 when these machines were at their peak in terms of popularity and userbase (I read somewhere that in 1986 the C16/+4 command around 10% of the total UK Game sales! - Imagine that: One in evry ten games sold were for the C16/+4), once on ythe bandwagon they produced two games, then like many a company, jumped off and headed straight to the 16-bit Market, whjere they produced Mercenary 2: Sword or Damocles. Sadley this sequel to Mercenary won't be available for the +4 or any other 8 Bit - 'IT JUST WON'T FIT' commented a Novagen Employee. There are two Novagen games, both require 64k Memory.

MERCENARY

A classic Arcade Adventure with elements of flight simulation, fast and smooth 3D Vector graphics which are very realistic in terms of perspective and different angled views. Very original and unique, there is alot of freedom as you can explore a whole planet. Many funny situations and plenty of puzzles to keep the player guessing. Probably the most professional game available for the +4.

Voted Arcade Adventure of 1986 by many magazines.

Overall Rating: 10
Price £9.95

THE SECOND CITY

The Second City is a dataset for Mercenary (ie, you need the original Mercenary to play it (no you don't, I cracked it in 1988, and you can use it with a copy, ED)). It basically is mercenary revisited - nut a much harder version of the original (to be honest I get absolutely nowhere in this game). Same graphics - different colours: Pink and Red; A good addition to the classic original, but lost marks for being to hard to complete!

Overall Rating: 7
Price £5.95

Both games are available on one purchase for £14.95.

Solutions For Mercenary

There are 2 ways (as far as I know!)

1) The hard way is to collect a series of objects in a particular order, key objects are the NOVADRIVE, SPIDERS WED (which is a skeleton key) and the KITCHEN SINK. Then fly to location ##-## and find the teleporter that transports you to loc: 03-15 where an intersteller ship is stored. What I have just written is only a rough outline to this solution because I can't fully remember what I did - Sorry!

2) Alternatively you can try this, collect the 'Antenna' and 'Metal Detector', go and earn yourself one million credits (ie, sell objects, help one side by destroying the other sides buildings - this is where the metal detector is usefeul: the colour code tells you which building belongs to who). Once you have the million; go to the comminications room where you can hire a ship from 'HERTZ', Easy?

Heres Some Tips To Help You On Your Travels On Targ:

1) The cheese is a super fast ship!
2) To collect the 'spiders web' you need the 'kitchen sink' which is in the Palyar's Colony Craft (Loc:08-08 Altitude:64997) The Sink enables you to pick up and carry most objects (including you ship and more importantly the spiders web) regardless of their mass or weight.
3) One of my favourite tactics is to sell as many objects as I can too the Mechanoids; then change sides and kidnapp the Mech's leader and sell him to the Palyars.
4) Finally choose carefully which side you sell objects too, usually one side offers substantially more credits than the other!

Final Words

I hope more members write articles for this magazine - lets keep it alive! Everyones welcome to write Publisher Profile Articles - I'm Sure Roy would gladly print any (yep sure I don't mind, ED); nomatter how long or short they are.

I think this is the only club in the UK which C16/+4 users can express their thoughts & ideas, share each others knowledge and with the new PD software from overseas the clubs prospects look better already (Well the prospects have slackened abit, but I'm going to have a re-think of how the club is structured and run, ED).

Andy Tang, LONDON.

Dear Roy
I thought I'd write and submit a few articles for publication. I think I should also say if any of your members are stuck in the adventures listed below I might be able to help them. I might send some tips along with my next letter anyway, but if you'd like to list the following, people count contact me; either through the magazine or direct to me.

Yours
Neville Clark, 64 Westwood Cres, Eccles, Manchester, M30 8DZ.

Adventures
Ten Little Indians
Classic Adventure
Spiderman
Feasibility Experiment
Salvage
Lone Survivor
Waxworks
Circus
Escape From Pulsar 7
Mansion Adv 1
Arrow Of Death
Time Machine
Treasure Hunter
Golden Baton

Some Pokes
Exorcist

To stop ghosts killing, reset & run/stop into monitor, type:
M1ACB Put 18 into first line, then start game with:
G3B00

Blagger

POKE 10468,96 for infinite energy
POKE 13652,234
POKE 13653,234
POKE 13654,234 for infinite lives

Big Mac

Load the reset run/stop, type:
M23BA and put BD into the first line, for infinite lives, the start game with:
G1B58

Jet Set Willy

POKE 14884,96 for no nasties
POKE 10874,234
POKE 10875,234
POKE 10876,234 for infinite lives
SYS 10752 starts game

Review

Product: Games Designer
Reviewer: Neville Clark, Manchester

It was some years ago when I bought this game and have not, I will admit, used it that often. However just recently a fellow +4 user recommended to me that I should begin to use it again.

Though I have not fully mastered the controls yet, it does look like quite an entertaining and rewarding buy. What it involves is you designing and creating games for you to play and save.

With the lack of new software on the market I can hardly complain about the restrictions of this game. Though you can only construct one screen, I am sure as my friend has said that with a bit of fiddling about in the machine code could allow extra screens to be produced.

Don't let that put you off though, if you know nothing about programming. You need not have any skills whatsoever to be able to use this program.

As I have said, with the lack of new games on the market nowadays you can't really lose with this (Hmmmm?, ED). If you see it, buy it and if you can't get the hang of using it, then the three games pre-programmed should keep you entertained.

From [illegible], Holland.

N.J. Ritchie
176 Albert Avenue,
Prestwich,
Manchester.
M25 8HF
4/5/91

Dear Roy,

Please find enclosed the information on RS232 that I promised. I have got all the copies of "Your Commodore and "C.C.I" going back to July '86 so if there are any other articles you would like please let me know. In return I would be very grateful if you could send me a copy of the instructions for "Botticelli" as soon as possible as there are a few things I can't work out, such as how to put text on the screen etc. Also could you possibly do me a sketch of where to solder the wires on the PLUS4 cicuit board to provide sound to an external speaker? Perhaps that would be useful for other members so maybe it should go in the magazine?

On the subject of RS232, it seems there is a bug in the software. Commodore didn't do toowell debugging the PLUS4 software did they? I have read that there is one in the spreadsheet when you get to 255 files but I can't get mine to work at all. When I write the first file and save it to disk it somes back as complete garbage and I can't move on to the next file. I have got two PLUS4's and the same thing happens with both. Do you know if this is a common problem or have I just been unlucky?

Well, I must close now so I can have another go at "Botticelli". Best of luck with the RS232.

Yours

Nick

P.S. I photocopied the other article mentioned from January 87 but I think I must have put it in the envelope with the ones I sent to Peter Crack on adventures. I'll send you a copy as soon as I get back to work.

Nick

RS232 On The Plus/4

19,200 baud on the Plus/4? A Beeb listing in a Commodore mag? Are we kidding you? We certainly aren't. . .

By R C Hemes

Surprising though it may seem, the Plus/4 is a lot faster at communicating using RS232 than the 64. This is due to the 64 using a software simulation to copy the action of the 6551 ACIA IC which would normally be used for RS232. Indeed, the 64 is so slow that it often misses characters even at the leisurely 300 baud.

The Plus/4 on the other hand, has a 6551 which handled all the RS232 signals except CTS which is dealt with by a 6529. Also, there is no need to connect S-into pin B (as suggested in 'Interfacing with the RS232', *Your Commodore*, January 1987, p 82-90).

The Plus/4 software has the ability to use XON/XOFF flow control with the user's choice of XON/XOFF characters. These are normally CTL/S for XOFF and CTL/Q for XON. This facility is used to stop the remote device from transmitting more data when the receiving device has no more room in its input buffer, and to restart the remote device when the receiving device has removed enough characters from its buffer.

The Plus/4 has a dedicated 64-byte RS232 user receive buffer located at $03F7, to $0436, and a one byte system transmit buffer at $S07CF which is used to hold the current XON/OFF character.

If the ACIA is configured, the interrupt handler will call two subroutines to handle RS232 interrupts. The first routine:

```
  5 REM BBC PROGRAM TO TEST PLUS/4 TO BBC RS232 AT 19200 BAUD
 10 *FX7,8
 20 *FX8,8
 30 *FX3,5
 40 *FX2,1
 50 FOR Y=0TO7
 60 A$=INKEY$(0)
 70 IF A$<>CHR$(19) THEN100
 80 A$=INKEY$(0)
 90 IF A$<>CHR$(17) THEN80
100 PRINT STR$(Y);
110 NEXTY
120 GOTO 50
```

```
  5 REM PLUS/4 PROGRAM TO TEST PLUS/4 TO BBC RS232 AT 19200 BAUD
 10 OPEN2,2,0,CHR$(31)+CHR$(5)
 20 POKEDEC("FC"),17:POKEDEC("FD"),19
 30 GET£2,A$
 40 IFA$<>"" THENPRINTA$;
 50 GOTO 20
```

Listings 1 and 2: programs for the BBC B (top) and the Plus/4 to show transmission from the BBC to the Plus/4 at 19,200 baud.

Checks for a remote device initiated XON/XOFF sequence and handles it appropriately;
Checks user receive buffer is full, and if so then ignores this received character and returns;
If there is room for eight characters, then it sets various flags, and stores XOFF character from $FD into the system transmit buffer and puts the received character in the user input buffer.

The second routine called will:
Check ACIA transmit buffer empty and return if not;
Check for CTS signal low, and return if so;
Check input buffer full flag, and send either XOFF or XON as appropriate.

It is the job of the RS232 portion of the CHRIN routine to reset the user buffer-full and remote-paused flags and initiate sending of the XON character.

Note that the XON/XOFF protocol is used only if the user has POKEd the XON/XOFF characters into $FC and $FD respectively; if these two locations are zero then characters received when the buffer is full are ignored.

And The Proof. . .

I have tested the Plus/4 connected to a BBC Model B, running at 19,200 baud, with the BBC sending the Plus/4 at full speed, and over a one hour period not a single character was lost by the Plus/4. The two programs for the Plus/4 and the BBC are shown in Listing 1 and 2.

Unfortunately, there is a bug in the Plus/4 ROM RS232 routines, which causes the Plus/4 to crash immediately after it has transmitted the first XON resume character, and the rest of this article describes how to fix this bug.

First we need to make our own copy of the Kernal ROM, and then patch the incorrect code. This is not so easy. . . Listing 3 is a listing of a Basic Program and a machine code program which copy the ROM down into RAM, cause the new version of the Kernal to be executed, and inhibit switching back to ROM. The top of memory pointers are also reset to $7FFF, and the bad code in the RAM copy is fixed. The Basic program must be typed in Exactly as shown, with No additional spaces, etc, as the machine code program is immediately above it at $4163.

A disassembly of the RS232 code in the Kernal is included in figure 3.

```
1 POKE51,0:POKE53,0:POKE55,0:POKE52,128:POKE54,128:POKE56,128:CLR

2 SYS4163

3 NEW
```

Listing 3A: Basic program to reset top of memory pointers and call machine code program to copy the Kernal and Basic from ROM to RAM, and change all references to $FF3E to $FF3F.

Listing 3B: Machine code program to copy ROM to RAM and inhibit switching to ROM.

```
                        ORG     $1043
1043 78                 SEI
1044 8D 3E FF           STA     $FF3E     ;SWITCH TO ROM
1047 A0 00              LDY     #$00
1049 B9 00 80   LP1     LDA     $8000,Y   ;COPY PART 1 OF ROM TO RAM
104C 99 00 80           STA     $8000,Y   ;$8000 TO $FCFF
104F C8                 INY
1050 D0 F7              BNE     LP1

1052 EE 4B 10           INC     LP1+2

1055 EE 4E 10           INC     LP1+5

1058 AD 4E 10           LDA     LP1+5

105B C9 FD              CMP     #$FD

105D D0 EA              BNE     LP1

105F A0 40              LDY     #$40

1061 B9 00 FF   LP2     LDA     $FF00,Y   ;COPY PART 2 OF ROM TO RAM

1064 99 00 FF           STA     $FF00,Y   ;$FF40 TO $FFFF

1067 C8                 INY

1068 D0 F7              BNE     LP2
```

```
106A A9 3F          LDA     £<$FF3F   ;INHIBIT ALL ROM SWITCHING
106C 8D 82 04       STA     $0482
106F 8D 9E 04       STA     $049E
1072 8D AC 04       STA     $04AC
1075 8D B7 04       STA     $04B7
1078 8D C2 04       STA     $04C2
107B 8D CD 04       STA     $04CD
107E 8D D8 04       STA     $04D8
1081 8D E3 04       STA     $04E3
1084 8D E1 07       STA     $07E1
1087 8D 32 81       STA     $8132
108A 8D 4E 81       STA     $814E
108D 8D BB CF       STA     $CFBB
1090 8D F7 FF       STA     $FFF7
1093 8D 3F FF       STA     $FF3F     ;SWITCH TO RAM
1096 A9 4C          LDA     £$4C      ;PUT JMP $CEC5 AT $EB1B
1098 8D 1B EB       STA     $EB1B
109B A9 C5          LDA     £<$CEC5
109D 8D 1C EB       STA     $EB1B+1
10A0 A9 CE          LDA     £>$CEC5
10A2 8D 1D EB       STA     $EB1B+2
10A5 A2 06          LDX     £$06      ;COPY CODE TO PATCH AREA
10A7 BD B2 10  LP3  LDA     PCH,X
10AA 9D C5 CE       STA     $CEC5,X
10AD CA             DEX
10AE 10 F7          BPL     LP3
10B0 58             CLI
10B1 60             RTS
10B2 8D CF 07  PCH  STA     $07CF     ;SAVE CH IN SYSTEM O/P BUFFER
10B5 68             PLA               ;PULL NEXT USER INPUT CHAR
10B6 4C 1E EB       JMP     $EB1E     ;RETURN TO RS232 RCUTINE
```

E9

Listing 4: The Plus/4 RS232 routines.

```
        TRANSMIT RS232 CHARACTER, ENTERED FROM IRQ ROUTINE AT $CE28
EA5B AD D4 07 SEA5B   LDA $07D4       ;GET ACIA STATUS
EA5E 29 10            AND £$10        ;TX DATA REG FLAG
EA60 F0 32            BEQ BEA94       ;->TX DATA REG NOT EMPTY
EA62 AD 10 FD         LDA $FD10       ;6529B
EA65 29 02            AND £$02        ;CTS PRESENT
EA67 F0 2B            BEQ BEA94       ;->NO, DO NOT TRANSMIT
EA69 A2 00            LDX £$00
EA6B 2C D0 07         BIT $07D0       ;SYSTEM INPUT BUFFER FULL?
EA6E 10 09            BPL BEA79       ; -> NO
EA70 AD CF 07         LDA $07CF       ;YES, GET CHAR FROM SYS BUF
EA73 8E D0 07         STX $07D0       ;CLEAR SYSTEM BUFFER FULL FLG
EA76 4C 89 EA         JMP JEA89
EA79 2C CE 07 BEA79   BIT $07CE       ;USR INPUT BUFFER FULL?
EA7C 10 16            BPL BEA94       ; -> NO
EA7E 2C D6 07         BIT $07D6       ;YES, LOCAL PALSE SET?
EA81 30 11            BMI BEA94       ; -> YES
EA83 AD CD 07         LDA $07CD       ;NO,GET USER CHAR TO SEND
EA86 8E CE 07         STX $07CE       ;CLEAR USER BUFFER FULL FLAG
EA89 8D 00 FD JEA89   STA $FD00       ;ACIA TRANSMIT REG
EA8C AD D4 07         LDA $07D4       ;GET ACIA STATUS REG
EA8F 29 EF            AND £$EF        ;SAVE ONLY TRANSMIT DATA REG
EA91 8D D4 07         STA $07D4       ; STATUS BIT
EA94 60       BEA94   RTS
        CHECK FOR RECEIVED RS232 CHARACTER ENTERED FROM IRQ AT $CE25
EA95 AD D4 07 SEA95   LDA $07D4       ;GET ACIA STATUS REG
EA98 29 08            AND £$C8        ;RECEIVE DATA REG FLAG
EA9A F0 54            BEQ BEAF0       ;->RX DATA REG NOT FULL
EA9C AD D4 07         LDA $07D4       ;GET RX DATA REG FLAG
EA9F 29 F7            AND £$F7        ;SAVE ONLY RX DATA REG
```

```
EAA1 8D D4 07          STA $07D4      ; STATUS BIT
EAA4 AD 00 FD          LDA $FD00      ;GET ACIA RCV REGISTER
EAA7 F0 19             BEQ BEAC2      ; -> NO DATA OR NULL
EAA9 8D D5 07          STA $07D5      ;SAVE THE CHAR
EAAC C5 FC             CMP $FC        ;IS IT XON CHARACTER?
EAAE D0 07             BNE BEAB7      ; -> NO
     RECEIVED XON REMOTE END WANTS US TO RESTART TRANSMISSION
EAB0 A9 00             LDA #$00       ;YES, CLEAR
EAB2 8D D6 07          STA $07D6      ;  LOCAL PAUSE FLAG
EAB5 F0 39             BEQ BEAF0      ;    AND RETURN
EAB7 C5 FD    BEAB7    CMP $FD        ;IS IT XOFF CHARACTER?
EAB9 D0 07             BNE BEAC2      ; -> NO
     RECEIVED XOFF CHARACTER, REMOTE END WANTS US TO STOP TRANSMISSION
EABB A9 FF             LDA #$FF       ;YES, SET LOCAL
EABD 8D D6 07          STA $07D6      ;  PAUSE FLAG
EAC0 D0 2E             BNE BEAF0      ;    AND RETURN
EAC2 AD D3 07 BEAC2    LDA $07D3      ;NCHARS IN USR INPUT BUFFER
EAC5 C9 3F             CMP #$3F       ;FULL?
EAC7 F0 27             BEQ BEAF0      ; -> YES
EAC9 C9 38             CMP #$38       ;NO, AT THRESHOLD FOR XOFF?
EACB D0 0F             BNE BEADC      ; -> NO
EACD A5 FD             LDA $FD        ;YES, GET XOFF CHAR
EACF F0 08             BEQ BEADC      ; -> NO XOFF CHAR DEFINED
EAD1 8D CF 07          STA $07CF      ;PUT IN SYSTEM O/P BUFFER
EAD4 A9 FF             LDA #$FF
EAD6 8D D0 07          STA $07D0      ;SET SYSTEM BUFFER FULL FLG
EAD9 8D D7 07          STA $07D7      ;SET REMOTE PAUSE FLAG
EADC AE D1 07 BEADC    LDX $07D1      ;GET USER I/P BUFFER INDEX
EADF E8                INX            ;ADD ONE
EAE0 8A                TXA
EAE1 29 3F             AND #$3F       ;MAKE INDEX MOD 64 AND
```

71

```
EAE3 8D D1 07          STA $07D1       ; STORE NEW INDEX
EAE6 AA                TAX             ;GET INDEX IN .X
EAE7 AD D5 07          LDA $07D5       ;GET RECEIVED CHAR
EAEA 9D F7 03          STA $03F7,X     ; AND STORE IN USER I/P BUF
EAED EE D3 07          INC $07D3       ;NCHARS IN USR I/P BUFFER
EAF0 60        BEAF0   RTS
     CONTINUATION OF CHRIN FOR RS232 DEVICE
EAF1 AD D3 07 SEAF1    LDA $07D3       ;NCHARS IN USR I/P BUFFER
EAF4 F0 34             BEQ BEB2A       ; -> EMPTY
EAF6 08                PHP             ;SAVE INHIBIT STATUS
EAF7 78                SEI
EAF8 AE D2 07          LDX $07D2       ;I/P Q FETCH INDEX
EAFB E8                INX             ;ADD ONE
EAFC 8A                TXA
EAFD 29 3F             AND #$3F        ;MAKE MOD 64
EAFF 8D D2 07          STA $07D2       ; AND RESTORE IT
EB02 28                PLP             ;RECOVER INHIBIT BIT
EB03 AA                TAX             ;I/P Q FETCH INDEX
EB04 BD F7 03          LDA $03F7,X     ;GET NEXT CHAR FROM I/P BUF
EB07 48                PHA             ; ONTO STACK
EB08 CE D3 07          DEC $07D3       ;DECREMENT NCHARS IN BUF
EB0B AD D3 07          LDA $07D3       ;GET NCHARS IN I/P BUF
EB0E C9 08             CMP #$08
EB10 D0 19             BNE BEB2B       ; -> NCHARS <> 8
EB12 2C D7 07          BIT $07D7       ;IS REMOTE PAUSE FLAG SET?
EB15 10 14             BPL BEB2B       ; -> NO
EB17 A5 FC             LDA $FC         ;IS XON CHAR DEFINED?
EB19 F0 10             BEQ BEB2B       ; -> NO
EB1B 8D CF 07          STA $07CF       ;YES, PUT IT IN SYS O/F BUF
     HERE IS THE BUG. THE NEXT CHARACTER TO BE RETURNED TO THE USER
     IS THE TOP ENTRY ON THE STACK. A PLA SHOLD BE DONE HERE.
```

```
EB1E 38                 SEC
EB1F 6E 00 07           ROR $07D0       ;RESET SYS BUFFER FULL FLAG
EB22 4E D7 07           LSR $07D7       ; AND REMOTE PAUSE FLAG
EB25 2C D8 07 EB25      BIT $07D8       ;ACIA PRESENT?
EB28 10 0B              BPL BEB35       ; -> NO
EB2A 48         BEB2A   PHA             ;YES, SAVE NEXT CHAR TO SEND
EB2B AD D4 07 BEB2B     LDA $07D4       ;GET ACIA STATUS REG
EB2E 29 4F              AND #%C1001111  ; SAVE DSR
EB30 49 40              EOR #%01000000  ;  AND INVERT IT
EB32 85 90              STA $90         ;   STORE IN KERNAL ST WORD
EB34 68                 PLA             ;RECOVER CHAR READ
EB35 18         BEB35   CLC             ;FLAG NO ERROR
EB36 60                 RTS

     SETUP USER CHARACTER TO TRANSMIT

EB37 2C CE 07 BEB37     BIT $07CE       ;USER XMIT BUFFER FULL?
EB3A 30 FB              BMI BEB37       ;->YES, WAIT FOR IT TO EMPTY
EB3C 8D CD 07           STA $07CD       ;NO, STORE USER O/P CHARACTER
EB3F 38                 SEC             ; IN USER XMIT BUFFER, SET
EB40 6E CE 07           ROR $07CE       ;  USER XMIT BUFFER FULL FLAG
EB43 4C 2A EB           JMP BEB2A

     INITIALIZE RS232 CONSTANTS AND ACIA

EB46 A9 00      SEB46   LDA #$00        ;CLEAR ALL
EB48 A2 0B              LDX #$0B        ; RS232
EB4A 9D CD 07 BEB4A     STA $07CD,X     ;  VARIABLES
EB4D CA                 DEX
EB4E 10 FA              BPL BEB4A
EB50 8D 01 FD           STA $FD01       ;RESET ACIA
EB53 85 FC              STA $FC         ;CLEAR XON CHARACTER
EB55 85 FD              STA $FD         ;CLEAR XOFF CHARACTER
EB57 60                 RTS
```

7 SEAMY ROAD,
GOOLE,
NORTH HUMBERSIDE
DN14 6TA.

30-3-1991.

DEAR ROY,

SORRY FOR THE DELAY. I'VE FOUND THE PROGRAM, (I HAD TO CONDUCT AN EXTENSIVE SEARCH THROUGH MY BEDROOM TO FIND IT THOUGH!) AS YOU CAN SEE IT'S CALLED "BUS ROUTE 64" I DON'T KNOW IF IT'S ANY USE TO YOU BUT HAVE A LOOK ANYWAY.

HOW'S THE MAG GOING? AT A GUESS, NOT TOO GOOD (?) I'M BACK WRITING PROGRAMS SO HOPEFULLY I CAN HELP

SORRY I HAVEN'T WROTE BEFORE, BUT I HAVE HAD A LOT OF THINGS ON MY MIND. EVEN NOW I HAVE REVISION FOR MY EXAMS. AS I SAID, ONE OF THE BIGGEST PROBLEMS WAS SUBSCRIBING ANNUALLY BECAUSE I DIDN'T HAVE TO WRITE TO YOU EVERY MONTH. THEREFORE, THIS YEAR I WILL PAY MONTHLY.

ANYWAY, BYE FOR NOW

Bus Route 64

Connect two Commodore 64 computers through their serial ports or link in to the C16 and Plus 4

by Simon Clarke

There are many ways of connecting one Commodore computer to another but most methods require custom-made cables and complex software. This routine simply uses the serial bus which already has a cheap and easily available connector in the form of the disk drive/printer cable.

The serial bus has two lines which are capable of input and output on both computers: the CLocK and DATa lines. There is also the ATN (attention) line that is used to interrupt external devices, but unfortunately it cannot accept a data signal. There is a line designated SRQ IN (serial request in) on the C64 but this has been deleted on the C16 and Plus/4 microcomputers.

The SRQ IN line allows external devices to interrupt the C64 and can only be used as an input. The CLK and DAT lines are so called because of their use in the Commodore Kernal ROM during communication with serial bus devices. It is the CLK line that governs which bits are valid on the DAT line and in a sense it 'clocks' the bits going out.

Recently a few dual-player games have appeared on the market where two Commodore computers are connected together and a game is loaded into both. This routine has possibilities in such environments, requiring very little modification to the actual code.

Two copies must be made to get the routine up and running, one per computer, with a slight modification at the beginning of the routine 'INTVAR' in each version. Where it has LDA #XX, the XX must be 00 in one and 01 in the other. This is in order to condition TLKFLG (TaLK FLaG) so that one computer has the initial talk priority. This is toggled after every press of the return key.

When converting the routine to work with other Commodore computers which have a serial bus, the following table of register/bit numbers may be useful

```
10                      ;SERIAL BUS COMMUNICATIONS
20                      .ORG    $0801
30 COLOUR               =$0286
40                      ;VIDEO INTERFACE CHIP II REGISTERS
50 VIC                  =$D000
60 EXTCOL               =VIC+$20
70 BGCOL0               =VIC+$21
80                      ;COMPLEX INTERFACE ADAPTOR #2 REGISTERS
90 CI2                  =$DD00
100 CI2PRA              =CI2+$00
110 CI2DDRA             =CI2+$02
120                     ;KERNAL ROM ROUTINES
130 CHROUT              =$FFD2
140 GETIN               =$FFE4
150                     ;BASIC HEADER
160                     .WORD   EOB
170                     .WORD   1987        ;LINE NUMBER
180                     .BYTE   $9E         ;SYS TOKEN
190                     .BYTE   '2061'
200                     .BYTE   $00         ;SIGNALS END OF BASIC LINE
210 EOB                 .BYTE   $00,$00     ;SIGNALS END OF BASIC TEXT
220 SERCOM              JSR INTVID          ;INITIALISE VIDEO
230                     JSR LINREL          ;RELEASE SERIAL BUS LINES
240                     JSR INTVAR          ;INITIALISE VARIABLES
250 CHKFLG              LDA TLKFLG
260                     BNE TALK            ;BRANCH TO 'TALK OR 'LISTEN
270                     JMP LISTEN
280 TALK                JSR GETIN
290                     CMP #$00
300                     BEQ TALK            ;WAIT FOR CHAR FROM KEYBOARD
310                     STA BYTE
320                     PHA                 ;SAVE CHAR ON STACK
330                     LDX #$0D            ;LIGHT GREEN TEXT
340                     STX COLOUR
350                     JSR CHROUT          ;PRINT CHARACTER TO SCREEN
360                     SEI                 ;THEN SEND ON SERIAL BUS
370                     JSR CLKLO           ;SEND ATTENTION
380 WDATLO              JSR GETBIT
390                     BCS WDATLO          ;WAIT FOR ACKNOWLEDGE
400                     LDX #$08
410 NXTBIT              JSR CLKLO           ;MAKE DATA NOT VALID
420                     ASL BYTE            ;BIT TO SEND IN .C
430                     BCS MIDAT
440                     JSR DATLO           ;SEND ZERO BIT
450                     JMP DATVAL
460 MIDAT               JSR DATHI           ;SEND ONE BIT
470 DATVAL              JSR CLKHI           ;MAKE DATA VALID
480 WCLKLO              JSR GETBIT
490                     BMI WCLKLO
500 WCLKHI              JSR GETBIT
510                     BPL WCLKHI          ;WAIT FOR BIT ACKNOWLEDGE
520                     DEX
530                     BNE NXTBIT
540                     JSR DATHI           ;RELEASE DATA LINE
550                     CLI                 ;ALLOW MASKABLE INTERRUPTS
560                     PLA
570                     CMP #$0D            ;IF 'RETURN' KEY THEN LISTEN
580                     BNE CHKFLG
590                     DEC TLKFLG
```

COMPUTER	CLK-IN	CLK-OUT	DAT-IN	DAT-OUT
C16, Plus/4	$0001/6	$0001/1	$0001/7	$0001/0
VIC 20	$911F/0	$912C/3	$911F/1	$912C/7
C64/128	$DD00/6	$DD00/4	$DD00/7	$DD00/5


```
PROGRAM: BUS ROUTE 64

E3  10 BL=26    :LN=50    :SA=3814
    4
C0  20 FOR L=0 TO BL:CX=0:FOR D=
    0 TO 15:READ A:CX=CX+A
82  25 POKES3280,A:POKE SA+L*16+
    D,A:NEXT D
A5  30 READ A:IF A><CX THENPRINT
    "ERROR IN LINE";LN+(L*10):ST
    OP
1D  40 NEXT L:SYS38400
5D  50 DATA 11,8,195,7,158,50,48
    ,54,49,0,0,0,32,113,8,32,765

ED  60 DATA 173,8,32,107,8,173,2
    53,8,208,3,76,181,8,32,228,2
    55,1753
EE  70 DATA 201,0,240,249,141,25
    2,8,72,162,13,142,134,2,32,2
    10,255,2113
11  80 DATA 120,32,145,8,32,163,
    8,176,251,162,8,32,145,8,14,
    252,1556
5F  90 DATA 8,176,6,32,127,8,76,
    77,8,32,136,8,32,154,8,32,92
    0
82  100 DATA 163,8,48,251,32,163
    ,8,16,251,202,208,223,32,136
    ,8,88,1837
06  110 DATA 104,201,13,208,176,
    206,253,8,240,171,169,1,141,
    253,8,96,2248
55  120 DATA 169,147,32,210,255,
    169,0,141,32,208,141,33,208,
    96,173,0,2014
EA  130 DATA 221,9,32,141,0,221,
    96,173,0,221,41,223,141,0,22
    1,96,1836
CE  140 DATA 173,0,221,9,16,141,
    0,221,96,173,0,221,41,239,14
    1,0,1692
6A  150 DATA 221,96,173,0,221,20
    5,0,221,208,248,10,96,173,0,
    221,41,2134
3A  160 DATA 207,141,0,221,120,3
    2,163,8,48,251,32,127,8,32,2
    46,8,1644
96  170 DATA 32,136,8,162,8,32,1
    63,8,16,251,46,252,8,32,145,
    8,1307
FC  180 DATA 32,246,8,32,154,8,3
    2,246,8,202,208,233,88,169,1
    0,141,1817
AB  190 DATA 134,2,173,252,8,32,
    210,255,173,252,8,201,13,208
    ,197,238,2356
AB  200 DATA 253,8,76,22,8,160,1
    28,136,208,253,96,0,0,0,0,0,
    1348
DF  210 DATA 169,0,133,250,169,1
    49,133,251,169,1,133,174,133
    ,193,169,8,2234
A5  220 DATA 133,175,133,194,169
    ,0,133,252,169,150,133,253,1
    60,0,177,250,2481
59  230 DATA 145,174,230,250,208
    ,2,230,251,230,174,208,2,230
    ,175,165,250,2924
```

```
                BEQ CHKFLG
INTVAR          LDA #$01
                STA TLKFLG      ;0-LISTENER / 1-TALKER
                RTS
                ;
INTVID          LDA #$93
                JSR CHROUT      ;CLEAR SCREEN
                LDA #$00
                STA EXTCOL
                STA BGCOL0      ;BLACK SCREEN AND BORDER
                RTS
                ;
DATLO           LDA CI2PRA
                ORA #$20
                STA CI2PRA      ;MAKE SERIAL LINE DATA LOW
                RTS
                ;
DATHI           LDA CI2PRA
                AND #$DF
                STA CI2PRA      ;MAKE SERIAL DATA LINE HI
                RTS
                ;
CLKLO           LDA CI2PRA
                ORA #$10
                STA CI2PRA      ;MAKE SERIAL CLOCK LINE LOW
                RTS
                ;
CLKHI           LDA CI2PRA
                AND #$EF
                STA CI2PRA      ;MAKE SERIAL CLOCK LINE HIGH
                RTS
                ;
GETBIT          LDA CI2PRA
                CMP CI2PRA      ;AWAIT STABLE CLOCK & DATA LINES
                BNE GETBIT
                ASL A           ;GET DATA IN .C & CLOCK IN .N
                RTS
                ;
LINREL          LDA CI2PRA
                AND #$CF
                STA CI2PRA      ;RELEASE DATA & CLOCK (BOTH HI)
LISTEN          SEI
WLOCLK          JSR GETBIT
                BMI WLOCLK      ;WAIT FOR ATTENTION
                JSR DATLO       ;SEND ACKNOWLEDGE
                JSR DELAY
                JSR DATHI
                LDX #$08
WHICLK          JSR GETBIT
                BPL WHICLK      ;WAIT FOR DATA VALID
                ROL BYTE        ;GET BIT
                JSR CLKLO       ;THEN ACKNOWLEDGE IT
                JSR DELAY
                JSR CLKHI       ;RELEASE CLOCK LINE
                JSR DELAY
                DEX
                BNE WHICLK      ;DO EIGHT BITS
                CLI
                LDA #$0A        ;LIGHT RED TEXT
                STA COLOUR
                LDA BYTE
                JSR CHROUT      ;PRINT CHAR TO SCREEN
                LDA BYTE
                CMP #$0D
                BNE LISTEN      ;IF RETURN KEY THEN TALK
                INC TLKFLG
                JMP CHKFLG
                ;
DELAY           LDY #$80
WAIT            DEY
                BNE WAIT
                RTS
                ;
BYTE            .BYTE   $00
TLKFLG          .BYTE   $00
ENDCOM          .END
```

```
40  240 DATA 197,252,208,234,165
    ,251,197,253,208,228,169,155
    ,133,187,169,150,3156
7F  250 DATA 133,188,169,12,133,
    183,169,0,133,185,160,0,185,
    107,150,240,2147
B1  260 DATA 6,32,210,255,200,20
    8,245,32,207,255,240,251,201
    ,49,240,4,2635
2E  270 DATA 201,56,48,230,41,15
    ,133,186,76,234,245,147,17,1
    7,73,78,1797
B9  280 DATA 80,85,84,32,68,69,8
    6,73,67,69,32,78,85,77,66,69
    ,1120
43  290 DATA 82,13,17,67,65,83,6
    1,49,32,47,32,68,73,83,75,61
    ,908
AA  300 DATA 32,56,32,79,82,32,5
    7,58,45,32,0,66,85,83,32,82,
    853
92  310 DATA 79,85,84,69,32,54,5
    2,0,0,0,0,255,255,255,255,0,
    1475
```

Interfacing with the RS232

Many people switch off when they hear the mention of the RS232 interface and related subjects. This article is aimed at clearing up some confusion and doubt.

By Steve Carrie

Some (if not most) of you will have heard by now of that favourite of all computer industry subjects, the RS232 communications standard. People have been known to go weak at the knees and hide when it is mentioned. Every day, it causes problems for computer engineers connecting up equipment such as printers and modems. It has even given rise to a lucrative business of building the so-called "break-out boxes" for monitoring and "fixing" RS232 lines.

If it is supposed to be an industry standard, why does it cause so much hassle? Every computer manufacturer has different ideas on how to build a computer: which devices to use, how big the screen should be, etc. It would seem that this train of thought also includes the RS232 standard.

Now, before I go on, let me say that the idea behind RS232 is great. A standard interface for connecting different types of hardware (including computers) together. RS232 is really useful in the field of telecommunications. Modems are usually connected to a computer via an RS232 link. You can connect two computers together and transfer programs between them (a technique often called "porting").

Unfortunately, things are not this simple. With different manufacturers having different ideas on how to implement the standard (Commodore is no exception) a great deal of confusion can arise when two pieces of hardware are to be connected together. Most of this confusion surrounds the way in which the control lines of an RS232 should be used. Thankfully, this does not concern us in this article since we will be using only a basic RS232 interface.

Basic RS232

As you may know, RS232 uses the serial method of data transmission. Information is sent bit by bit along a single wire to a receiving machine. Since data may flow in both directions, two wires plus a common return are required to make a basic RS232 communication line. The lines are usually connected to equipment by a 25-way D-type connector. Data leaving a computer exits via pin two and incoming data enters via pin three. The Ov return is connected to pin seven. This gives the "3-line" RS232 interface. There is no control over the flow of the data in either direction unless handled by the software (more on this later). An RS232 interface using more than these three lines is known as an "X-line" interface.

It is best to use a three or four core cable with a shield (we will see why in a moment) rather than separate wires.

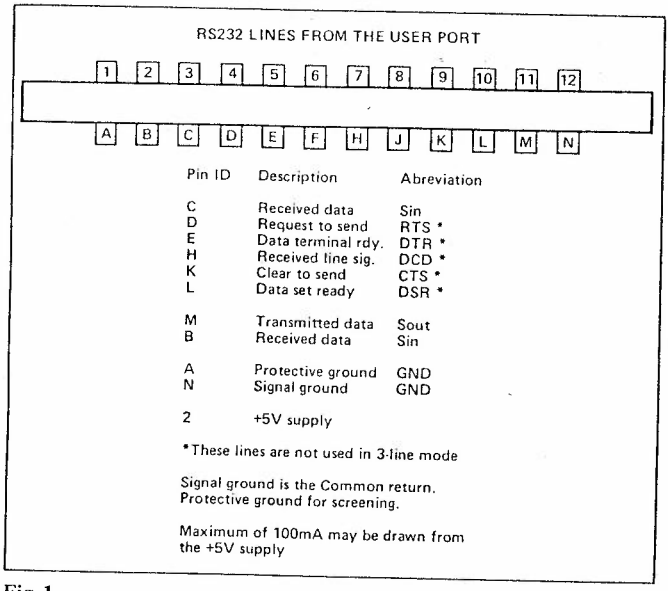
For the purposes of this article, this is all we require. "Ah yes" you say. "that's all very well but my Commodore doesn't have a 25-way d-type connector let alone any RS232 interface". True it doesn't have a 25-way connector but it does have an RS232 interface. You mean you didn't know?

Commodore RS232

Since the time of the Vic 20, Commodore gas "programmed" in a limited form of RS232 port. I say programmed because the hardware device normally associated with RS232 communications, the Universal Asynchronous Receiver/Transmitter (UART), is not present in the circuitry of the Vic 20, C64, 64C, C128 and C128D. The omission of the C16 and Plus/4 is intended. The C16 cannot handle RS232 comms (we shall see why in a moment). The Plus/4 on the other hand is omitted for a different reason. This machine DOES have a UART.

RS232 signals appear at the user port. This is exactly the reason why the C16 cannot handle RS232; it does not have a user port (a strange omission by Commodore. Anyone know why?). All the conversion is handled by the Operating System (Kernel). Thus

3-LINE RS21NTERFACE	CONNECTIONS (25-way D-connector)	
pin	function	
2	Transmitted data	Sout
3	Received data	Sin
7	Common (Ov)	Gnd



Pin ID	Description	Abreviation
C	Received data	Sin
D	Request to send	RTS *
E	Data terminal rdy.	DTR *
H	Received line sig.	DCD *
K	Clear to send	CTS *
L	Data set ready	DSR *
M	Transmitted data	Sout
B	Received data	Sin
A	Protective ground	GND
N	Signal ground	GND
2	+5V supply	

*These lines are not used in 3-line mode

Signal ground is the Common return.
Protective ground for screening.

Maximum of 100mA may be drawn from the +5V supply

Fig 1

(except in the case of the Plus/4) the user port coupled with the Kernel becomes almost the equivalent of the 6551 UART (the 6551 is the device that the software is supposed to emulate).

In the Vic the user port is controlled by a Versatile Interface Adaptor (VIA). In the other machines a Complex Interface Adaptor (CIA) is used (not a lot of difference as far as this article is concerned). Now before some of you run and hide at the mention of VIAs and CIAs, let me say that (thankfully) we do not have to program these devices directly. The Kernel handles all of the RS232 associated programming.

Figure 1 shows the RS232 associated connections to this port looking from the rear of the machine. Note that the user port uses a 0.156" edge connector (available from Maplin: order number BK74R). The top row of terminals are identified with numbers while the bottom row terminals are identified with letters. This means that it is very easy to connect the socket the wrong way up which could have disastrous consequences for your computer. The correct way is with the letters to the bottom as shown.

Connecting Up

The terminals labelled Sout (transmitted data) and Sin (received data) and GND (Ov common) are the three lines we need. Note however that there are TWO terminals labelled Sin. The reason for this lies with the way in which the RS232 on these machines works. One of the Sin lines is for data, the other is a flag or trigger input. Because the Commodore RS232 relies on Non-Maskable Interrupts (NMI) or interrupt request (IRQ) in the case of the Plus/4, there must be some form of detection to let the system know when data is being received.

When data is being received over the Sin line, the voltage level on this pin changes rapidly, causing interrupts to occur. The system software collects the data on the Sin line, placing it in an area of memory called the Receive Buffer. There is also a Transmit buffer for outgoing data. Thus the reception and transmission of data is basically transparent to the user. However, this method has its problems as we will see. The practical upshot of all this is that BOTH Sin terminals must be connected to the Sin line. Since they are next to one another, this is no great problem.

Also note the connection called PROTECTIVE GROUND. This terminal should be connected to the shield of your cable if you suspect any outside electrical interference of causing data errors.

While on the subject of making connections, you will have to use a soldering iron. Now don't run and hide (again). If you are not too sure about handling one, get an electronics hobbyist friend to do it for you.

It should go without saying that YOU SHOULD NEVER MAKE CONNECTIONS TO THE SOCKET WHILE IT IS CONNECTED TO THE COMPUTER. Always disconnect the socket BEFORE making or changing any connections and NEVER connect the socket to a line machine. ALWAYS switch off the power BEFORE plugging or unplugging!

OK. So you've connected your socket up and plugged it in the correct way... what now? If you only want to communicate with another Commodore (Vic, C64, Plus/4 or C128) then there is no great problem. The only thing to watch is that you must connect the Sout of one machine to the Sin of the other as in Figure 2 in order for them to exchange data (a bit obvious really!).

Commodore connections

If you don't intend connecting your machine to anything other than another Commodore, you can skip the next bit all together and get on with the programming. If you have a friend with one of the machines mentioned, you will be able to type messages to one another; handy if you live next door and you have a long piece of 3-core cable – your own mini-network! (Not recommended if you live across the street or several houses away!) How far you will be able to keep the machines apart depends very much on the operating conditions. Electrical interference may cause errors and there will come a point where the line is just too long. It's best to experiment with what you've got.

Non-Commodore Connections

This is where things get tricky. Connecting to anything other than a similarly equipped Commodore has its

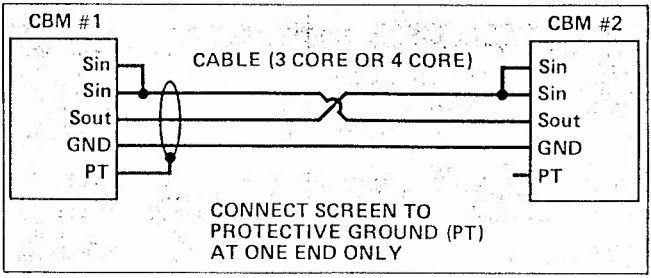


Fig 2

problems. This is because normal RS232 requires logic levels -12v (logic 1) and +12v (logic 0) as opposed to the Commodore RS232 logic levels of 0v (logic 0) and +5v (logic 1) (No that isn't a mis-print. RS232 logic is opposite voltage level-wise to your Commodore). However this problem can be overcome quite easily. There are various logic devices available specifically for this purpose. Among these are the RS423 line driver 3691 and RS423 line receiver 88LS120N. These devices convert between TTL logic levels and RS423 logic levels. RS423 is another (stricter) standard which is compatible with RS232 (at least to a certain extent). The beauty of using these devices is that they will operate from a +5v/-5v supply. This means that the user port's +5v output (pin 2: see Figure 1.) may be used for the +5v supply and all that is required is a -5v supply. These are the devices used by the BBC micro for it's RS423 port. Figure 3 shows the circuit required.

I can hear the groans of discontent. "I can't do that! I've never built an electronic circuit before!". If you do have an electronics hobbyist friend, try bribing him/her into building it for you.

I will not go into detail over the construction of the interface. I will assume that if you are building this, you know what you are doing. The 74LS00 device is used here as a logic inverter. I suggest that you use i.c. sockets so as not to subject the devices to heat which could damage them.

Power Supplies

If you plan to use the mains (careful!) power supply circuit shown in Figure 4, I suggest that you use both the +5v and -5v supply circuits so as not to rely too much on the already overworked computer power supply. It goes without saying that no-one should use the mains unless they do know EXACTLY what they are doing.

The alternative battery supply circuit in figure 5 assumes that you are using the +5v supply from the computer. This circuit may also be used with one of those pocket calculator mains adaptors (like the Spectrum power supply) since the circuit regulates the voltage to -5v. The diodes shown prevent damage to the circuit if the battery/adaptor is incorrectly connected. Remember to disconnect the battery when you are not using the interface. This has the advantage of not requiring you to build a mains power supply.

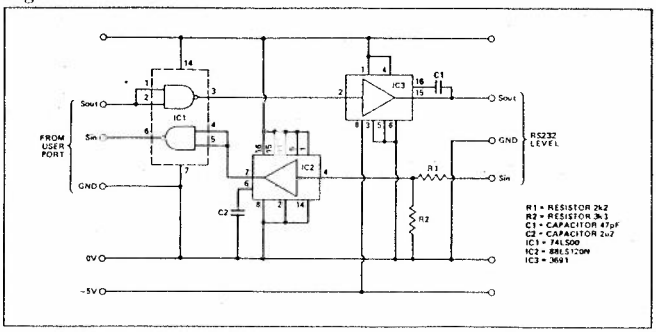
Remember that you cannot connect an ordinary Commodore (without this interface) to one using this circuit. Both machines must be similarly equipped. Using this interface, you will be able to connect your Commodore to other computers such as the BBC and any suitably equipped IBM compatible. I have connected my C128 to my Televideo TS1605 (IBM compatible running MSDOS) and transferred files between them without any problems. This should also work with the Commodore PC10 and PC20 IBM compatibles. The only thing to watch for here is that most IBM compatibles need to have pins four and five (RTS and CTS) connected together as shown in Figure 6. The BBC will also require a similar connection. Because of RS232 differences between systems, I cannot say that this will work with every machine, but it will work with most. I have also connected my 128 up to a mainframe DECSYSTEM 2050 running at 1200 baud with absolute success.

Programming Commodore RS232

OK so you've got this far. You've linked your Commodore to another machine. What now?

Using the RS232 interface on a Commodore is much the same as using a printer or a disk drive. You must OPEN a logical file and use PRINT# and GET# to send and receive data. Note that you should NOT use INPUT# since there is a possibility that the system might hang by attempting to get input when it isn't there. Also be aware that on the Vic 20 and C64, the RS232 receive and transmit buffers are created at the top of Basic memory when the OPEN command is executed. (The buffers are permanently defined on the C128 and, I think, the Plus/4). This has the effect of wiping out any variables previously defined. (It performs a CLR). Therefore, you should make the OPEN RS232 statement the first in any program. Another, more serious problem occurs if your Basic program is very large. Then there is a chance that OPENing an RS232 channel may destroy the end

Fig 3



of your program. So Beware!

The format for the OPEN statement is shown below:

OPEN lfn,dev,sec,CHR$ (control register)+CHR$ (command register)

where:
lfn = logical file number
dev = device (2 for RS232)
sec = secondary address (usually 0)
<control register>= see below
<command register>= see below

The RS232 interface is device number two. In order to operate correctly, the system requires you to tell it how fast you wish to transfer data (baud rate) and the format of the data number of stop bits, number of data bits, parity). The control register tells the system the baud rate, stop bits and data bits as in Table 2.

bit	7	6	5	4	3	2	1	0
	S	D	D	R	B	B	B	B

S – stop bits
0 – one
1 – two

D – data bits
00 – 8
01 – 7
10 – 6
11 – 5

B – baud rate
R – Rclock

BBBB	
0001	50
0010	75
0011	110
0100	134.5
0101	150
0110	300
0111	600
1000	1200 (C64 max)
1001	1800
1010	2400 (max)

R (plus 4 only)
0 – external clock
1 – internal clock
(keep to 1)

Plus/4 capable of up to 19200 baud

Table 2

2400 baud is the maximum speed that is available on the Commodore RS232 on the C128. The C64 will allow up to 1200 so this is the maximum baud rate that I will use in the examples. If however you have any problems, try selecting 600 baud instead.

The command register defines other interface parameters as follows

bit	7	6	5	4	3	2	1	0
	P	P	P	D	T	T	R	H

P – Parity
T – Transmit control
D – duplex
R – receive control
H – handshake

PPP	H (non PLUS4)	H (PLUS4)
000 Disabled	0 Full	0 3-line
001 Odd	1 Half	1 X-line
011 Even		
101 Mark (1)		
111 Space (0)		

TT (Plus/4 only)	R (Plus/4 only)
00 IRQ Disabled, RTS=1, TX off	0 IRQ on
01 IRQ Enabled, RTS=0, TX On	1 IRQ off
10 IRQ Disabled, RTS=0, TX On	
11 IRQ Disabled, RTS=0, BRK	

Table 3

Plus/4 users should refer to pages 207-211 of the user manual for further details on their machine's RS232 interface. This machine can handle transfer rates of up to 19200 baud.

Handshake determines how the interface will operate. We will be using 3-line. X-line is where you are using control lines as well as the data lines. This makes things a bit complex so we will stick to 3-line.

Duplex should be set to Full. This determines how the receive and transmit will behave.

Parity is a kind of error check. When data is received, the system checks it to see if it agrees with the parity. If not, the parity error bit in the variable ST (status) is set indicating some sort of error. These bits allow you to set the type of parity check required. Of course, both machines should be set to the same parity. In most cases, parity is not used and error checking is done in a different way (more on this later).

This may seem a little complex but it isn't really. Lets suppose that we want to open an RS232 channel to run at 1200 baud, eight data bits, one stop bit, no parity. The OPEN statement would be:

```
OPEN 2,2,0,CHR$(24)+CHR$(0)
(non Plus/4)
OPEN 2,2,0,CHR$(24)+CHR$(5)
(Plus/4)
```

and that is that! Simple, eh? In fact, if you stick to this particular format of eight data bits and no parity, you can't really go wrong. Just change the baud rate to suit.

Incidentally, the Commodore 64 programmers reference Guide tells you that the command register character is NOT required. It's probably safer to leave it in. My C128 sometimes won't work without it!

Using PRINT#AND GET# we can write a fairly simple terminal program running at 1200 baud. Running this on two machines (assuming its two Commodores) you will be able to type in a message on one keyboard and see it appear on the other machines display as well as your own. Figure 7 shows the program while Figure 8 shows a similar program for an IBM compatible machine running PCDOS or MSDOS and GWBASIC.

Figure 7 Commodore Basic mini terminal program

```
10 OPEN 2,2,0,CHR$(24)+CHR$(0)
11 REM PLUS4 USERS USE OPEN
2,2,0,CHR$(24)+CHR$(5)
20 PRINT "[CLS]"
30 GET#2,A$
40 IF A$<>"" THEN PRINT A$;
50 GET B$
60 IF B$<>"" THEN PRINT
B$;:PRINT#2,B$;
70 GOTO 30
```

Figure 8 GWBASIC mini terminal program

```
10 OPEN "COM1:1200,N,8,1" AS #1
20 CLS
30 IF LOC(1)=0 THEN 50
40 A$=INPUT$(1,#1):PRINT A$;
```

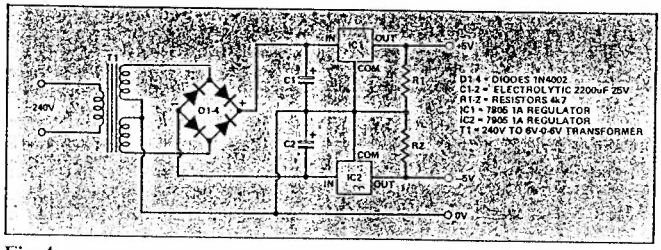


Fig 4

```
50 B$=INKEY$
60 IF B$<>"" THEN PRINT
B$::PRINT#1,B$;
70 GOTO 30
```

Both programs check first for a character from the RS232 port. If one is found, it is output to the screen. If not, the program checks for a keyboard input. If found, the character is sent to the screen and also to the RS232.

Another interesting experiment when using an IBM compatible machine is to make the following change to the mini terminal program on the Commodore:

```
60 IF B$<>"" THEN PRINT£2,B$;
```

and run it. Now type the following on the IBM while in DOS:

```
MODE COM1:1200,N,8,1
CTTY COM1
```

(MODE is external so disk containing the command must be in the default drive).

This causes the IBM to think that its input/output device is the RS232 port instead of the console. This has the very odd effect of making the Commodore control the IBM! (although the character sets are different and some odd graphics characters may be printed). To return control to the IBM console, you must type CTTY CON on the Commodore.

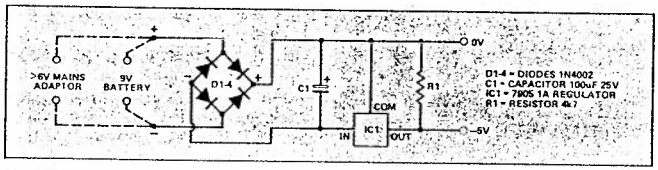
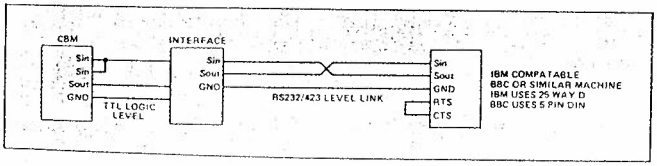


Fig 5

Obviously, not many of you have access to an IBM, but this experiment does show up one very important problem with the Commodore's 3-line RS232 interface. If you take control of the IBM using the CTTY COM1 command again and type DIR which is the MSDOS command to display the directory, the first few lines are printed normally and then suddenly, there is a whole lot of garbage. If you slow down the baud rates to say 300, the problem may disappear. Why does this happen? Remember what I said about the Receive buffer? Well what happens is that data is received transparently over the RS232 lines and placed in the buffer. The buffer fills up quicker than Basic can empty it with the result that the buffer overflows and data is lost. Buffer overflow can be detectd by examining the ST status variable, but Basic is just too slow and you may still loose data.

Fig 6



Solutions

There are a few ways of solving this problem:

1. Use a slower baud rate to let Basic keep up.
2. Devise some form of software data flow control.
3. Connect control lines up and use an x-line interface.
4. Use machine code to process RS232 data.

Option 1 is not a good idea. Anyone who has used an RS 232 link at 300 baud will know why (yawn!).

Option 2 is better but requires programming which we will touch on later when we discuss file transfer.

Option 3 is not really practical here.

Option 4 is probably the best solution in this case. Data can be removed from the buffer much faster using machine code and this will be fine for the speeds at which we will be working (up to 1200 baud).

A very short machine code program can be written to handle the RS232 interface. The routine that follows is given in the form of a Basic loader. Change the variable AD to any free area in your computer's memory (the tape buffer is a good place). This program is for a C128 but it should work on any of the Commodore machines discussed.

```
10 OPEN 2,2,0,CHR$(24)+CHR$(0)
11 REM PLUS4 USERS: OPEN
2,2,0,CHR$(24)+CHR$(5)
20 AD=2816:CA=AD
30 READ BY
40 IF BY =-1 THEN POKE
AD,BY:AD= AD+1:GOTO 30
50 :
60 PRINT CHR$(14):REM SWITCH
TO LOWERCASE'
70 SYS CA
80 :
90 :
100 DATA 162,2,32,198,255,32,228,255,
168
110 DATA 32,204,255,152,240,3,32,210,
255
120 DATA 32,228,255,240,233,72,
162,2,32
130 DATA 201,255,104,32,210,255,
32,204
140 DATA 255,56,176,217,-1
```

The only way out of this program is a RUNSTOP/RESTORE. Table 3 is the disassembled code with comments (addresses may differ from your version.

```
OBOO LDX#$02      ; Make channel 2 the current input device
OBO2 JSR $FFC6
OBO5 JSR $FFE4    ; Scan the RS232 buffer for 1 character
OBO8 TAY          ; Save data
OBO9 JSR $FFCC    ; Clear the channel
OBOC TYA          ; Restore the date
OBOD BEQ $OB12    ; If data is a null (a zero byte) then skip
OBOF JSE $FFD2    ; Output valid character
OB12 JSR $FFE4    ; Check for keyboard input
OB15 BEQ $OBOO    ; If none then recheck RS232
OB17 PHA; Save data
OB18 LDX #$02     ; Make channel 2 the current output device
OB1A JSR $FFC9
OB1D PLA          ; Restore data
OB1E JSR #FFD2    ; Send byte to RS232
OB21 JSR $FFCC    ; Clear channel
OB24 SEC          ; Force loop to check RS232 input
OB25 BCS $OBOO
```

Table 3

The program uses the kernal jumptable calls so it should be valid for all of the machines.

After the machine code has been placed in memory, the RS232 channel is opened with a file number of 2 (the channel number. If you change this you must change the two LDX instructions in the code) and the machine code routine called. Now the RS232 receive buffer does not get a chance to fill up so no data is lost. The OPEN statement could be replaced by the code in Table 4.

```
LDA #$02          ; Channel 2
LDX #$02          ; Device 2
LDY #$00          ; Secondary Addr. 0
JSR $FFBA         ; Set logical file
LDA £$01          ; 2 chars in filename
LDX # NAME        ; Pointer to filename
LDY # NAME        ; Pointer to filename
JSR $FFBD         ; Set filename
JSR $FFCO         ; Open RS232
Rest of code
.NAME:#B:24,0 ; Control and command register values (PLUS 4—
.NAME:#B:24,5) (#B: Is the byte directive of my C64 and C128
assemblers)
```

Table 4

Character set translation

All this should be fine for Commodore to Commodore communications. It will also work with Commodore to IBM/BBC etc. comms but may yield strange results, the reason being that the Commodores do not use standard ASCII codes. They use what is sometimes called PETSCII, the Commodore PET being the first to use it. The most noticeable effect of this is that certain characters will appear on the Commodore's screen as graphics characters. The solution to this is to insert some form of translation table or routine into the code. Since data flows two ways, a routine or table would be required for each of input and output.

Two translation tables, each of 256 bytes, are required to handle outgoing and incoming data translations. The following program builds the translation tables and the machine code into memory starting at address AD. It needs at least 600 bytes to operate in so you need only change AD to the address you require. This version is for the C128 (Program Translate).

Program Translate

Line 5: AD=start address; BR=Control reg; CR=command reg.

Lines 10-100: Prepare tables
Lines 111-113: Setup name of channel (register images)
Lines 120-160: POKE receive table
Lines 170-200: POKE transmit table
Lines 210-230: POKE code to memory
Lines 240-270: Adjust table references in code
Lines 280-290: Adjust register image references in code
Lines 300-310: Select lowercase and call routine
Lines 330-370: Code data

Table 5 shows the disassembled machine code (addresses are offsets from the start address):

Basically, the operation is the same as before except:

1. The channel is opened from machine code.
2. Characters sent and received are translated.

Note that the backslash (\) character received will be displayed as a pound sign. The CBM Charset does not have a backslash.

Transferring files via RS232

This is probably the most useful application of RS232; transferring data between unalike machines. It is possible to transfer programs between two entirely different machines.

When using a 3-line interface, it is necessary to introduce some form of protocol into the software at both ends. This ensures that each machine understands exactly what the other intends to do.

A typical protocol exchange would look like a conversation between the two machines:

machine 1: Are you there?
machine 2: Yes.
machine 1: Requesting data transfer.
machine 2: OK.
machine 1: Transferring data (block of data sent)
machine 2: Received data.

and so on. The "are you there", "yes" etc. messages are actually 1 byte control codes. Normal ASCII has 32 control codes (codes 0-31). The actual usage varies from system to system and there are one or two protocol standards. For your own use, you don't have to follow any set protocol AS LONG AS BOTH MACHINES ARE USING

82

THE SAME ONE. Figure 9 gives a list of ASCII code 0-31 and their meanings.

Note the two marked XON and XOFF. These you may recognise. XON/XOFF protocol is useful where large amounts of continuous data are being transferred. The receiving machine can send an XOFF code when it wants the transmitting machine to hold off sending data, and send an XON when it wants to resume. This type of control is often used with dumb terminals to mainframes.

Another method of transferring data is to send it in blocks of 2-255 characters (usually 128 chars). This removes the need for XON/XOFF control because as each block is sent, the two machines exchange control codes. This method of transfer also allows error checking to be carried out on the data. One of the most popular error check methods is the Cyclic Redundancy Check (CRC). We won't go into this here as there is no real need for such a complex check with hard-wired RS232. CRC is really useful for Telecommunications via modems. Phone lines are noisy and data errors may easily result at high baud rates. The CRC error check allows the two machines at either end to check the data for errors, if an error is found, the receiving machine requests that the data is transmitted again.

When sending data using the block method, the software collects data bytes into "packets" of 128 bytes. Assuming the link is open, the transmitting machine (TM) sends a start-of-transmission code. The receiving machine (RM) replies with an acknowledge code. The TM then sends a start-of-block code followed by a data packet followed by an end-of-block code. The RM replies with an acknowledge code (assuming the data was received correctly) and the TM sends the next block. This continues until all data has been exchanged, whereupon the TM sends and end-of-transmission code and the link enters a wait-state.

The above description does not conform to any standard but will work. If error checks are included, the RM could send a data-error code if the data had been corrupted whereupon the TM would re-transmit the same data packet.

```
0000 LDA #$02       ;Logical channel 2
0002 LDX #$02       ;Device 2
0004 LDY #$00
0006 JSR $FEBA      ;Set lfn
0009 LDA #$02       ;Name length (2 chars)
000B LDX #<NAME     ;NAME is address of char string
000D LDY #>NAME
000F JSR $FFBD      ;Setnam
0012 JSR $FECO      ;Open
0015 LDX #$02       ;Make RS232 the input device
0017 JSR $FFC6
001A JSR $FFE4      ;Get a character
001D TAY            ;Save it
001E JSR $FFCC      ;Restore default device
0021 TYA
0022 BEQ 002A       ;If char=0 then no char rec'd
0024 LDA RXTABLE,Y  ;Get PETSCII char from TXTABLE
0027 JSR $FFD2      ;Send to screen
002A JSR $FFE4      ;Check keyboard
002D BEQ 0015       ;No char, loop to check RS232
002F TAY            ;Index
0030 LDA TXTABLE,Y  ;Get ASCII equivalent
0033 PHA            ;Save
0034 LDX #$02       ;Make RS232 the default output
0036 JSR $FFC9
0039 PLA
003A JSR $FFD2      ;Output character
003D JSR $FFCC      ;Restore normal output
0040 SEC            ;Forced loop
0014 BCS 0015
```

Table 5

The program given in Figure 10 should run on any of the CBM machines discussed and is written in Basic. Since the comms are being controlled by the software, the receive buffer will not overflow. It allows a user to transfer a data file from one machine to another. The link is 1200 baud, eight data bits and no parity. It should be run on both machines. The control codes are shown in Table 6.

Note that I'm not using some of these as they should be used but, as I said earlier, as long as you stick to the same protocol on both machines, you'll be OK.

Possible Developments

The example programs I've given here do not show all of what may be achieved using the RS232 interface. It is possible (using a special version of the circuit described earlier) to have more than two machines running on a single 3-line RS232 commlink. This would allow a group of users (with special software written in machine code for maximum speed) to set up a mini-network. The possiblities are endless. I hope this article has helped to fuel your imagination. If you have any comments or ideas, please write to me c/o Your Commodore or leave a COURIER on Compunet ID SC12.

Table 6

Code	ASCII	Meaning in this program
5	ENQ	Attent/enquire
6	ACK	Acknowledge
1	SOH	Start transfer (transfer filename)
4	EOT	End transfer
2	STX	Start block
3	ETX	End block
26	SUB	Enter terminal mode (special seq.)
10	DLE	Exit terminal mode (data link escape)

83

Figure 9 ASCII Control Codes.

Code	ASCII	Function				
0	NUL	Null	15	SI	Shift in	
1	SOH	Start Heading	16	DLE	Data link escape	
2	STX	Start text	17	DC1	Device control 1	XON
3	ETX	End text	18	DC2	Device control 2	
4	EOT	End transmission	19	DC3	Device control 3	XOFF
5	ENQ	Enquire	20	DC4	Device control 4	
6	ACK	Acknowledge	21	NAK	Negative acknowledge	
7	BEL	Rings terminal bell	22	SYN	Synchronous idle	
8	BS	Backspace	23	ETB	End transmission block	
9	HT	Horizontal tab	24	CAN	Cancel	
10	LF	Line feed	25	EM	End medium	
11	VT	Vertical tab	26	SUB	Special sequence	
12	FF	Form feed	27	ESC	Escape	
13	CR	Carriage return	28	FS	File separator	
14	SO	Shift out	29	GS	Group separator	
			30	RS	Record separator	
			31	US	Unit separator	

Fig 10

```
PROGRAM: TRANSLATE


READY.

88  5 AD=4864:NA=AD:BR=24:CR=0
88  6 REM PLUS4 USERS USE BR=24:
    CR=5
86  10 DIM F%(255),T%(255)
AF  20 FOR J=0 TO 64:T%(J)=J:NEX
    T
65  30 T%(10)=0:T%(20)=8
34  40 FOR J=65 TO 90:K=J+32:T%(
    J)=K:NEXT
23  50 FOR J=91 TO 95:T%(J)=J:NE
    XT
E5  60 FOR J=193 TO 218:K=J-128:
    T%(J)=K:NEXT
92  70 T%(146)=16:T%(133)=16
35  80 FOR J=0 TO 255
F7  90 K=T%(J)
EA  100 IF K<>0 THEN F%(K)=J:F%(
    K+128)=J
EC  110 NEXT
A1  111 POKE AD,BR
92  112 POKE AD+1,CR
DF  113 AD=AD+2
58  120 RB=AD
F5  130 FOR X=0 TO 255
BC  140 POKE AD+X,F%(X)
14  150 NEXT
FA  160 AD=AD+256:TB=AD
20  170 FOR X=0 TO 255
C2  180 POKE AD+X,T%(X)
3C  190 NEXT
72  200 AD=AD+256:RT=AD
5E  210 RESTORE
92  220 READ BY
D8  230 IF BY<>-1 THEN POKE AD,B
    Y:AD=AD+1:GOTO 220
A4  240 HR=INT(RB/256):LR=RB-HR*
    256
AC  250 HT=INT(TB/256):LT=TB-HT*
    256
BC  260 POKE RT+37,LR:POKE RT+38
    ,HR
23  270 POKE RT+49,LT:POKE RT+50
    ,HT
C5  280 NH=INT(NA/256):NL=NA-NH*
    256
C5  290 POKE RT+12,NL:POKE RT+14
    ,NH
17  300 PRINT CHR$(14)
EF  310 SYS RT
C1  320 END
E8  330 DATA 169,2,162,2,160,0,3
    2,186,255,169,2,162,0,160,0,
    32,189,255,32,192
78  340 DATA 255,162,2,32,198,25
    5,32,228,255,168,32,204,255,
    152,240,6,185,0
2A  350 DATA 0,32,210,255,32,228
    ,255,240,230,168,185,0,0,72,
    162,2,32,201,255,104
61  360 DATA 32,210,255,32,204,2
    55,56,176,210,-1

IOPROGRAM: FIGURE 10


READY.

2F  100 OPEN 2,2,0,CHR$(24)+CHR$
    (0)
C6  110 REM +4 OPEN 2,2,0,CHR$(2
    4)+CHR$(5)
42  120 :
BA  130 REM C64 MAY NEED TO RUN
    AT 600 BAUD
93  140 REM I.E. OPEN 2,2,0,CHR$
    (23)+CHR$(0)
AC  150 :
4E  160 GOSUB 2390:REM SETUP TAB
    LES
90  170 :
0A  180 PRINT"{CLR}";CHR$(14)
B4  190 :
3D  200 REM ********************
    ***********
55  210 REM MAIN LOOP STARTS HER
    E
49  220 REM ********************
    ***********
FD  230 REM WAITS FOR RS232 END
    CODE
6F  240 REM CTRL-F (FILE TRANSFE
    R)
0D  250 REM CTRL-Z (TERMINAL MOD
    E)
F2  260 REM CTRL-C (EXIT PROGRAM
    )
35  270 :
23  280 :
69  290 PRINT "{SA}WAITING {SE,S
    N,SO} OR KEYPRESS"
13  300 GOSUB 1210: REM GET FUNC
    TION
F9  310 ON C GOSUB 400,530,800,3
    60
56  320 REM  GOSUB MODE,FIX,TERM
    OD,EXIT
39  330 GOTO 290
6F  340 :
65  350 :
74  360 REM EXIT PROGRAM
8F  370 CLOSE2:END
47  380 :
8D  390 :
C8  400 REM MODE
79  410 REM END RECEIVED - WAIT
    FOR EITHER
98  420 REM SS  (TERMINAL MODE R
    EQUEST)
F7  430 REM SOH (FILE TRANSFER R
    EQUEST)
83  440 :
89  450 PRINT "{SE,SN,SO} RECEIV
    ED.."
37  460 GOSUB 900 : REM SEND ACK

7C  470 GOSUB 1310: REM GET RECE
    IVE MODE
52  480 ON C GOSUB 640,730
65  490 RETURN
CF  500 :
C5  510 :
6F  520 REM FIX
CF  530 REM INITIATE FILE TRANSF
    ER
B5  540 GOSUB 2300:REM FILENAME
    FROM USER
84  550 GOSUB 950 :REM SEND ENQ
AE  560 GOSUB 1150:REM WAIT FOR
    ACK
50  570 GOSUB 1050 :REM SEND SOH
     + FILENAME
D2  580 GOSUB 1150:REM WAIT FOR
    ACK
1E  590 PRINT "{SF}ILE TRANSFER
    BEGINS.."
A7  600 GOSUB 1760:REM RUN FILE
    TRANSMIT
B2  610 RETURN     :REM BACK TO M
    AIN LOOP
54  620 :
42  630 :
7D  640 REM RESPOND TO FILE TRAN
    SFER REQUEST
13  650 PRINT "{SF}ILE TRANSFER
    REQUEST ..."
BC  660 FR$=""
97  670 GOSUB 1380:REM GET NAME
    FROM RS232
3C  680 GOSUB 900 :REM SEND ACK
BB  690 GOSUB 1980:REM FILE RECE
```

```
    IVE
74  700 RETURN     :REM BACK TO M
    AIN LOOP
F2  710 :
EB  720 :
FE  730 REM RESPOND TO TERMINAL
    MODE REQ.
67  740 PRINT "CSI]ERMINAL MODE
    REQUEST..."
F2  750 GOSUB 900 :REM SEND ACK
B0  760 GOSUB 1520:REM RUN TERMI
    NAL MODE
2F  770 RETURN     :REM BACK TO M
    AIN LOOP
3S  780 :
23  790 :
B6  800 REM INITIATE TERMINAL MO
    DE
89  810 PRINT "CSI]NITIATING TER
    MINAL MODE"
3D  820 GOSUB 950 :REM SEND END
99  830 GOSUB 1150:REM WAIT FOR
    ACK
10  840 GOSUB 1000 :REM SEND SS
A5  850 GOSUB 1150:WAIT FOR ACK
89  860 GOSUB 1520:REM RUN TERMI
    NAL MODE
9B  870 RETURN     :REM BACK TO M
    AIN LOOP
49  880 :
47  890 :
BB  900 REM SEND ACK
40  910 PRINT#2,CHR$(6);
15  920 RETURN
9F  930 :
95  940 :
34  950 REM SEND END
97  960 PRINT#2,CHR$(5);
43  970 RETURN
E0  980 :
DB  990 :
5A  1000 REM SEND SS
E0  1010 PRINT#2,CHR$(26);
71  1020 RETURN
30  1030 :
2E  1040 :
C1  1050 REM SEND SOH + FILENAME

02  1060 PRINT#2,CHR$(1);
E9  1070 PRINT#2,CHR$(FL);
30  1080 FOR X=1 TO FL
7E  1090 PRINT#2,CHR$(I%(ASC(MID
    $(F$,X,1))));
D2  1100 NEXT
86  1110 PRINT#2,CHR$(I%(ASC(I$)
    ));
EA  1120 RETURN
54  1130 :
42  1140 :
27  1150 REM WAIT FOR ACK (WACK)

F3  1160 GET#2,A$
95  1170 IF A$<>CHR$(6) THEN 116
    0
2E  1180 RETURN
90  1190 :
8E  1200 :
39  1210 REM GETFUNCTION
CF  1220 GET#2,A$
14  1230 IF A$=CHR$(5) THEN C=1:
    RETURN
00  1240 GET A$
B6  1250 IF A$=CHR$(6) THEN C=2:
    RETURN
03  1260 IF A$=CHR$(26) THEN C=3
    :RETURN
A3  1270 IF A$=CHR$(3) THEN C=4:
    RETURN
2E  1280 GOTO 1210
35  1290 :
23  1300 :
AB  1310 REM GET END MODE
AC  1320 GET#2,A$
A5  1330 IF A$=CHR$(1) THEN C=1:
    RETURN
CF  1340 IF A$=CHR$(26) THEN C=2
    :RETURN
64  1350 GOTO 1320
6F  1360 :
65  1370 :
4D  1380 REM GET FILENAME FROM R
    S232 SOH
5E  1390 GET#2,A$:IF A$="" THEN
    1390
1C  1400 A=ASC(A$)
E4  1410 FOR X=1 TO A
B0  1420 GET#2,A$:IF A$=""THEN 1
    420
47  1430 FR$=FR$+CHR$(R%(ASC(A$)
    ))
27  1440 NEXT
C3  1450 GET#2,T$:IF T$="" THEN
    1450
3C  1460 E$=",S,W"
BB  1470 T$=CHR$(R%(ASC(T$)))
D5  1480 IF T$="CSP]" OR T$="P"
    THEN E$=",P,W"
59  1490 RETURN
DB  1500 :
D1  1510 :
B3  1520 REM TERMINAL MODE
94  1530 REM REMAIN IN TERMINAL
    MODE UNTIL
C3  1540 REM EITHER DLE FROM RS2
    32
9C  1550 REM OF CTRL-C KEYSTROKE

24  1560 :
CA  1570 PRINT "CSE]NTERING TERM
    INAL MODE."
15  1580 GET#2,A$
16  1590 IF A$<>"" THEN 1620
BA  1600 IF A$=CHR$(16) THEN GOT
    O 1710
06  1610 PRINT CHR$(R%(ASC(A$)))
    ;
18  1620 GET A$
E3  1630 IF A$="" THEN 1580
48  1640 IF A$=CHR$(3) THEN GOTO
     1680
84  1650 PRINT A$;:PRINT#2,CHR$(I
    %(ASC(A$)));
A7  1660 GOTO 1580
B6  1670 :
3F  1680 REM CTRL-C KEYSTROKE
B2  1690 PRINT#2,CHR$(16);
90  1700 :
A1  1710 REM EXIT REQUEST RECEIV
    ED
EA  1720 PRINT "CSI]ERMINAL MODE
     ENDS..."
46  1730 RETURN
E8  1740 :
E6  1750 :
01  1760 REM FILE TRANSMIT
22  1770 OPEN 3,8,3,F$
25  1780 BL=1:SS=0
3E  1790 :
2D  1800 C=0:CH$="":BL$="":PRINT
    "BLOCK";BL;
5A  1810 GET#3,CH$:IF T$="I" THE
    N CH$=CHR$(I%(ASC(CH$)))
AC  1820 IF ST THEN FOR X=CH TO
    128:BL$=BL$+CHR$(0):NEXT:SS=
    -1:GOTO 1850
8A  1830 BL$=BL$+CH$
63  1840 C=C+1:IF C<128 THEN 181
    0
0C  1850 PRINT#2,CHR$(2);
70  1860 PRINT#2,BL$;
07  1870 PRINT#2,CHR$(3);
2F  1880 PRINT ".128 BYTES";
83  1890 GOSUB 1150
6C  1900 PRINT ".CSO,SK]."
58  1910 IF NOT SS THEN BL=BL+1:
    GOTO 1800
BD  1920 :
3E  1930 PRINT#2,CHR$(4);
6B  1940 CLOSE 3
EB  1950 RETURN    : REM BACK TO
    MAIN LOOP
95  1960 :
83  1970 :
0B  1980 REM RECEIVE FILE
EC  1990 PRINT "CSF]ILE ";FR$;"
    OF TYPE ";T$
77  2000 FI$=FR$+E$
51  2010 OPEN 3,8,3,FI$
78  2020 BL=1
CF  2030 :
3B  2040 CH$="":BL$=""
D9  2050 GOSUB 2240:IF R$=CHR$(4
    ) THEN 2190
A5  2060 IF R$<>CHR$(2) THEN 205
    0
AU  2070 PRINT "CSB]LOCK ";BL;".
    ";
A1  2080 FOR X=1 TO 128
E0  2090 GET#2,CH$:IF LEN(CH$)=0
    THEN CH$=CH$+CHR$(0)
F5  2100 IF T$="CST]" OR T$="I"
    THEN CH$=CHR$(R%(ASC(CH$)))
E1  2110 BL$=BL$+CH$
D2  2120 NEXT
78  2130 GOSUB 2240:IF R$<>CHR$(
    3) THEN 2130
5C  2140 PRINT#3,BL$;" BYTES ";
DB  2150 BL=BL+1
B4  2160 GOSUB 900:PRINT ".(SO,S
    K).";LEN(BL$)
AD  2170 GOTO 2040
B6  2180 :
5B  2190 PRINT "CSI]RANSFER COMP
    LETE."
13  2200 CLOSE 3
1A  2210 RETURN  : REM BACK TO M
    AIN LOOP
BE  2220 :
B4  2230 :
C8  2240 REM WAIT FOR ETX OR EOT

A6  2250 GET#2,R$
CF  2260 IF R$=""THEN 2250
6B  2270 RETURN
CA  2280 :
C0  2290 :
1F  2300 REM GET FILENAME FROM U
    SER
74  2310 INPUT "CSF]ILENAME TO T
    RANSFER:";F$
57  2320 INPUT "(T)EXT ((SS,SE,S
    Q)) OR (P)ROGRAM ((SP,SR,SG]
    )";T$
15  2330 E$=",S,R"
0C  2340 IF T$="P" THEN E$=",P,R
    "
31  2350 FI$=F$+E$:FL=LEN(F$)
CF  2360 RETURN
71  2370 :
6F  2380 :
A9  2390 REM SETUP TRANSLATION T
    ABLES
FF  2400 PRINT "CSS]ETTING UP...
    "
D7  2410 DIM I%(255),R%(255)
F0  2420 FOR J=0 TO 64:I%(J)=J:N
    EXT
3A  2430 I%(10)=0:I%(20)=8
C3  2440 FOR J= 65 TO 90:K=J+32:
    I%(J)=K:NEXT
60  2450 FOR J=91 TO 95:I%(J)=J:
    NEXT
54  2460 FOR J=193 TO 218:K=J-12
    8:I%(J)=K:NEXT
05  2470 I%(146)=16:I%(133)=16
22  2480 FOR J=0 TO 255
2C  2490 K=I%(J)
CF  2500 IF K<>0 THEN R%(K)=J:R%
    (K+128)=J
55  2510 NEXT
6F  2520 RETURN
4F  2530 GET A$
5A  2540 IF A$="" THEN 2530
64  2550 PRINT ASC(A$)
25  2560 GOTO 2530
```

August/September/October & November 1991
Please send me Vol 3 Issue's 5/6/7 & 8 of 'C16/C116/+4 COMPUTING BI-MONTHLY'.

I enclose a PO/CHEQUE for the value of £4.00.

NAME & ADDRESS:

SIGNED: _____

Please Note, that if all monthly Subscribers should send there £2 for each Bi-Monthly issue by the 10th of every second month for the next issue, ie, £2 by 10th May for the June/July 1991 issue etc. This will improve effieciency (is that correct??) in collecting all funds, because last year saw a few members trying to dodge paying, once they received a mag, due to my trusting manner.