

DELTA 14/B HANDSET SYSTEM

INSTRUCTIONS FOR USE

Your DELTA 14b handset is guaranteed for a year against faulty workmanship or components. As with anything with moving parts it is possible to wear parts of it out inside that time if you use it an abnormal amount. An occasional wipe of the ball and socket joint with a clean duster is all that is required. The joint should not be lubricated nor should it be necessary to open the handset. If the handset is faulty, please return it to VOLTFACE Ltd., Park Drive, Baldock, stating the date and place of purchase and if possible your receipt if the item is less than a year old.

The DELTA 14b can be used at various levels from a single joystick up to two joysticks complete with keypads and software which integrates it into the operating system of the BBC micro. It requires a Model B, or a Model A with analogue and user ports fitted.

These instructions are not intended to teach you to program, nor to explain how the conversion from keyboard to joystick is performed, as on the driver cassette, which is sold separately. The intention is to provide the necessary information for people to

- (a) Use the handsets on programs which were written for joysticks.
- (b) Test the handset by typing in the demonstration programs
- (c) Provide the information necessary to write a program which uses the joysticks and keypads.

Some software is written with the joystick operating very near it's edges which makes it look insensitive. This quick test will show you how sensitive the joystick is.

```
10 MODE2:CLS:REPEAT
20 X=ADVAL1/50:X=1280-X:Y=ADVAL2/50
30 DRAW X,Y:UNTIL FALSE
```

1. USING A DELTA 14b ON SOFTWARE WRITTEN FOR ORDINARY JOYSTICKS.

(1a) A single joystick in the analogue port will run any program written for a single joystick in an ACORNSOFT compatible way. The top 5 buttons all act as one fire button and the 3 buttons below them work as the fire button on a second joystick; such as on systems which have two joysticks wired to the same plug. The bottom 6 buttons will have no effect. NO PROGRAMMING OF THE JOYSTICK IS NECESSARY. Run the program and select the joystick option.

(1b) A single joystick plugged into the adaptor box. It is obviously inconvenient and undesirable to keep plugging in and unplugging the adaptor box.

IF THE ADAPTOR BOX IS LEFT PLUGGED IN AND YOU WISH TO RUN A PROGRAM WRITTEN FOR A SINGLE JOYSTICK THEN THE DELTA 14B SHOULD NORMALLY BE PLUGGED INTO THE REAR SOCKET OF THE ADAPTOR BOX.

After switching on the computer or after running a program that used the adaptor box type in the following commands to ground the user port lines or the 'FIRE' button will not work.

For O.S 1.0 or later:-

*FX151,98,112 (RETURN) *FX151,96,0 (RETURN)

For O.S.0.1:- ?&FB62=112 (RETURN) ?&FB60=0 (RETURN)

Alternatively these commands can be put onto one of the red function keys. (See page 141 of the user guide)

(1c) Software written for two ordinary joysticks.
Plug a joystick into each socket of the adaptor box and type in the same commands as in (b) above if it has not already been typed in.

2. HOW TO PROGRAM A SINGLE HANDSET IN THE ANALOGUE SOCKET.

The most simple set up is one DELTA 14b plugged into the analogue port. The joystick will be read into channels 0 and 1 of the A/D converter. The buttons 9,10 & 11 and the two common fire buttons (see diagram on page 10) will be connected to Io0 and the buttons 6,7 and 8 to Io1. The co-ordinates of the joystick are read in BASIC by the commands X = ADVAL1 & Y = ADVAL2. The buttons can be read by B = ADVAL0 AND 3 which will give an answer of 0 if no button is pushed, 1 for one of the top 5 buttons, 2 for one of the next row down and 3 if a button in each of the top two rows of three are pressed together. Here is a demonstration program for a single joystick in the analogue socket. Called "SIMPLE PAINTER", it allows you to draw pictures using the joystick. You can move the flashing cross around the screen with the joystick. By pressing one of the top 5 buttons the joystick will draw a line. Next you can draw triangles. Move the cross to a point and press one of the second row of 3 buttons (i.e 6,7 or 8 see plan page 10). A number 1 will appear at the bottom of the screen. Now move to the next point and press the same button again. The number 1 will change to a 2. Move to the last point of the triangle and press the button again and the 2 will change to a 3 and a triangle will be drawn between the 3 points that you have plotted. Finally you can change the colour by pressing one button of the line 9,10,11 at the same time as one of line 6,7,8. Each time you press the buttons the colour will change and at the bottom of the screen will appear "COLOUR (No.)" in the selected colour. Be careful to press down the top row button first and to change the colours using the second row button whilst keeping the top one pressed or you will enter a triangle point.

SIMPLE PAINTER PROGRAM FOR ONE DELTA 14b JOYSTICK IN THE ANALOGUE PORT

```
10 MODE2:D%=7:C%=1
20 VDU 23;8202;0;0;0
40 GCOL0,3
50 MOVE 0,85
60 VDU 25,5,0;1023;25,5,1279;1023;25,5,1279;85;25,5,0;85;
70 VDU28,0,31,19,30,24,10;90;1270;1010;
80 REPEAT
90 VDU5
100 X=ADVAL1/65520*1500:IF X>1280 THEN X=1280
110 X%=1280-X
120 Y%=ADVAL2/65520*1300:IF Y%>1023 THEN Y%=1023
130 GCOL 0,D%
140 A%=ADVAL0AND3
150 IF A%=3 THEN PROCPCOL:GOTO 210
160 E=0
170 IF A%=1 THEN PROCDRAW:GOTO 210
180 IF A%=2 THEN PROCPOINT:GOTO 210
190 B=0
200 IF A%=0 THEN PROCMOVE
210 UNTIL FALSE
220 DEFPROCMOVE
```

```

230 GCOL 4,D%:MOVE X%,Y%:PRINT"+ "
240 FOR T=0 TO 200:NEXT
250 GCOL 4,D%:MOVE X%,Y%:PRINT"+ "
260 ENDPROC
270 DEFPROCDRAW
280 IF B=1 THEN 300
290 MOVE X%,Y%:B=1
300 DRAW X%,Y%
310 ENDPROC
320 DEFPROCPOINT
330 VDU4:COLOUR7
340 IF B=1 THEN 410
350 ON C% GOTO 360,370,380
360 X1=X%:Y1=Y%:C%=2:PRINT;C%-1:GOTO 400
370 X2=X%:Y2=Y%:C%=3:PRINT;C%-1:GOTO 400
380 X3=X%:Y3=Y%:PRINT;C%:C%=1
390 MOVE X1,Y1:MOVE X2,Y2:PLOT85,X3,Y3
400 B=1
410 ENDPROC
420 DEFPROCCOL
430 IFE=1 THEN 480
440 D%=D%+1:IFD%>7 THEN D%=0
450 E=1:VDU 4:COLOUR D%
460 IF D%=0 THEN COLOUR 7
470 PRINTTAB(10,0);"COLOUR ";D%
480 ENDPROC

```

NOTES ON "SIMPLE PAINTER" PROGRAM

The horizontal co-ordinate X% has to be inverted because to make the joystick ACORN compatible a decreasing A/D value is required for an increasing screen co-ordinate.

Variables used in the SIMPLE PAINTER program:-

A% = the value of the "fire" buttons 6,7,8 and 9,10,11.

X% = The horizontal co-ordinate.

Y% = The vertical co-ordinate.

D% = The current selected colour.

B & E are flags set when a button is pressed and cleared when it returns to the main program.

C% = The triangle point number counter.

Lines 10 to 70 turn off the cursor - set some of the variables - draw the border and set up the text and graphics windows.

Lines 80 to 210 is the main program which reads the joystick and push buttons and calls the various PROCedures according to the value of the pushbuttons held in A%.

Lines 220 to 260 - If no button is pressed moves the "+" around the screen.

Lines 270 to 310 Draws a line wherever the joystick moves if one of the top 5 buttons is pressed.

Lines 320 to 410 draw the triangle by entering three points if one of the second row buttons is pressed. Note that having entered a point, B is set to 1 which prevents any further action until the button is released and the program can run through line 180 to clear B in line 190.

Lines 420 to 480 change the colour if A%=3. Note that E performs a similar flag function to B in that it only allows one colour change per press of the button.

3. HOW TO PROGRAM THE ADAPTOR BOX AND 1 OR 2 HANDSETS

To make full use of the handset buttons an adaptor box is required. The adaptor box plugs into the analogue port and a ribbon cable connects it to the user port. If the plug on the ribbon cable has no polarising projection, then the small arrow on the plug should line up with the arrow on the case next to the words "user port". The analogue port gives the joysticks access to the A/D converters and the user port is needed to strobe and decode all of the handset buttons. The user port plug is removed by applying outwards pressure to the two small levers on the ends of the socket.

The joystick values are read in BASIC by the command ADVAL.

ADVAL 1 = horizontal value from the rear socket.

ADVAL 2 = vertical value from rear socket.

ADVAL 3 = horizontal value from side socket.

ADVAL 4 = vertical value from side socket.

The keypads are read in register B of the user port 6522 VIA (See pages 435 - 438 of the user guide). The handset buttons are arranged as a 3 x 4 matrix. This means that there are 3 vertical columns of 4 buttons commoned together and 4 horizontal rows of common buttons. The columns are strobed: i.e. each column is grounded in turn by a continuously running program. After each column is grounded the 4 rows are read as a number. Normally the 4 rows would be 4 1's; but if a button is pressed in a column that is grounded then that row will give a 0 instead of a 1. Thus by checking the number returned by the row every time a column is grounded then we know which button(s) is pressed.

The first step is to set the Data Direction Register (DDR) of port B of the user port, because we are going to write to columns (output) and read from rows (input). We also want line PB7 to be an output as we use that to select which handset socket we are reading from. Thus writing 240 to the DDR (binary 11110000) will set PB7 to PB4 as output and PB3 to PB0 as input. This is done with the command *FX151,98,240 (Write 240 to SHEILA +98) or ?&FE62 = &F0 (For O.S. 0.1).

Having set the DDR we can now write to the output register (ORB) the value for the column that we want to ground. PB7 selects which handset will be read. Set to 0, it will read from the rear socket or to 1, from the side socket. Lines PB6, PB5 and PB4 are connected to the columns which are common to both handsets. The following table shows the numbers needed to be written to ORB to read each column.

NUMBER	PB7	PB6	PB5	PB4	COLUMN TESTED
224 (&F0)	1	1	1	0	0, 3, 6, 9 SIDE SOCKET
208 (&D0)	1	1	0	1	1, 4, 7, 10 SIDE SOCKET
176 (&B0)	1	0	1	0	2, 5, 8, 11 SIDE SOCKET
96 (&60)	0	1	1	0	0, 3, 6, 9 REAR SOCKET
80 (&50)	0	1	0	1	1, 4, 7, 10 REAR SOCKET
48 (&30)	0	0	1	1	2, 5, 8, 11 REAR SOCKET

To use 1 column only you can write the number to ORB using the command *FX151,96,(value) or for O.S. 0.1 ?&FE60 = (value). If, as is more likely you need to scan the columns to read all of the buttons then it is best to make the column number a variable and go through a PROCedure for each value. We cannot use a variable with a

*FX call so we must substitute this for a BASIC CALL to the OSBYTE routine at &FFF4. (See page 445 of the user guide) This takes the form of:-

```

10 RESTORE 30:FOR N=0 TO 5
20 A%=&97:X%=&60:READ Y%:CALL &FFF4
30 DATA &E0,&D0,&B0,&60,&50;&30
40 A%=&96:X%=&60:R=USR (&FFF4)
50 NEXT N

```

Each time the columns are strobed with a different value from line 30 a value for R will be read from line 40. From R we can extract the number required by dividing by &10000. The following table gives the values of R obtained for each row of buttons.

R/&10000	PB3	PB2	PB1	PB0	BUTTON PRESSED
15	1	1	1	1	NONE
14	1	1	1	0	9, 10 or 11
13	1	1	0	1	6, 7, or 8
11	1	0	1	1	3, 4, or 5
7	0	1	1	1	0, 1, or 2

Therefore, as we know which column was grounded before we read R we can say exactly which button is pressed. More than 1 button in the same column, pressed at the same time can be detected from the value of R, but two buttons pressed in separate columns at the same time will require storing of the values or the program will react to the first column in which it finds a button pressed.

OTHER USES OF THE SIDE SOCKET ON THE ADAPTOR BOX

Apart from a second handset the side socket of the adaptor box can be used for other devices. From the circuit diagram you will see that the following is available from this socket.

- ADVAL3 and ADVAL4 analogue inputs
- I/O lines PB4, PB5 & PB6 are available as inputs or outputs.
- PB0 to PB3 are available as inputs only as they are passed through the select chip
- PB7 which switches the input from one socket to the other can be used as a signal to tell an external device that data from it is valid on that socket at that time.
- The light pen strobe line is taken to this socket so a light pen can be plugged in here without disconnecting a joystick in the other socket.

This second demonstration program is written for two players using ONE handset plugged into the REAR socket of the ADAPTOR BOX. The game is based on a simplified version of "Crown Green Bowling", with the winner being the person who has his ball closest to the randomly placed jack ball. The game is played by pointing the joystick in the direction that you want the ball to travel from the man in the centre of the green. The strength of the bowl is determined by which button is pressed from no's 1 to 11. Button 1 will bowl a short ball, up to button 11 which will just go off the edge of the green. After you have each bowled 4 balls; press the fire button to be told who was closest then press the fire button again to start another game.

ROUTING FOR TWO PLAYERS USING ONE DELTA 14B HANDSET IN THE REAR
 SOCKET OF A 14B/1 INTERFACE

```

10 DIM X(11),Y(11),NEAR(8)
20 *FX151,98,240
30 SFX=1500/65520:SFY=1300/65520
40 MODE2
50 VDU23,224,60,60,60,60,24,126,189,189,
23,226,189,60,102,102,102,102,102,102,
60 VDU 23,225,&00,&30,&7B,&7B,&7B,&30,&00,&00
70 BALL=0:WIN=9999999:PLAY=1:VDU5
80 GCOL 0,130:GCOL 0,4:CLS
90 MOVE 640,512:VDU 224:MOVE 640,480:VDU 226
100 X1=RND(1200):Y1=RND(1000)
110 GCOL 0,7:MOVE X1,Y1:VDU 225
120 REPEAT:BALL=BALL+1
130 PROC PB
140 PROC VAL
150 GCOL 3,2:MOVE 200,100:PRINT"PLAYER ";INT((PLAY+1)/2):FOR
T= 1 TO 200:NEXT
160 GCOL 3,2:MOVE 200,100:PRINT"PLAYER ";INT((PLAY+1)/2)
170 D4=1:D5=1:D3=1
180 D1=X%-640:D2=Y%-512
190 IF D1<0 THEN D4=-1
200 IF D2<0 THEN D5=-1
210 IF ABS(D1)<250 AND ABS(D2)<250 THEN 130
220 IF K%=0 THEN 130
230 D3=D3+1:D6=D1*((100+D3)/100)*D4:D7=D2*((100+D3)/100)*D5
240 IF D6<640 AND D7<512 THEN 230
250 FOR S= 1 TO 11:X(S)=640+(D6*D4*9*K%)/121:Y(S)=512
+(D7*D5*9*K%)/121:MOVE X(S),Y(S):GCOL 3,2:VDU 225
260 SOUND 0,-10,2,1
270 FOR T= 1 TO 200:NEXT:GCOL 3,2:MOVE X(S),Y(S):VDU 225:NEXT
280 GCOL 0,PLAY:MOVE X(11),Y(11):VDU 225
290 NEAR(BALL)=ABS(X1-X(11))^2+ABS(Y1-Y(11))^2
300 IF NEAR(BALL)<WIN THEN WIN=NEAR(BALL):WBALL=BALL
310 IF PLAY=1 THEN PLAY=3 ELSE PLAY=1
320 UNTIL BALL=8
330 PROC PB:IF K%=10 THEN PROC WIN ELSE 330
340 DEFPROC WIN:CLS
350 IF WBALL/2=INT(WBALL/2) THEN P=2 ELSE P=1
360 MOVE 200,600:GCOL 0,PLAY:PRINT"PLAYER ";P;" WINS"
370 MOVE 200,400:PRINT"WITH BALL ";INT((WBALL+1)/2)
380 PROC PB:IF K%=10 THEN 70 ELSE 380
390 ENDPROC
400 DEFPROC VAL
410 X%=(ADVAL1)*SFX:IF X%>1280 THEN X%=1280
420 X%=1280-X%
430 Y%=(ADVAL2)*SFY:IF Y%>1024 THEN Y%=1024
440 IF Y%<0 THEN Y%=0
450 ENDPROC
460 DEFPROC PB
470 K%=0:RESTORE 590
480 FOR COL%=1 TO 3
490 A%=&97:X%=&60:READ Y%:CALL
&FFF4:A%=&96:X%=&60:R=USR(&FFF4)
500 BUTTON=(R AND &F0000)/&10000
510 IF BUTTON=15 THEN 570
520 K%=COL%
530 IF BUTTON=14 K%=K%+8:GOTO 570

```

```

540 IF BUTTON =13 K%+K%+5:GOTO 570
550 IF BUTTON =11 K%-K%+2:GOTO 570
560 IF BUTTON =7 K%-K%-1:GOTO 570
570 NEXT
580 ENDPROC
590 DATA 96,80,48

```

BRIEF NOTES ON "BOWLING" PROGRAM

Line 20 sets up the user port, DDR for the adaptor box
Line 30 defines the 'scale factors' which ensure that the joystick covers the corners of the screen. The joystick reaches it's max. and min. values before it reaches the limits of it's travel, though it may still not reach a max. or min value in the corners of the screen because the joystick movement is circular and the screen oblong. It is undesirable to compress the potentiometers any further as this means that they will reach their max. and min values a long way short of the ends of travel, giving poorer resolution. It is better to expand the scale in software by use of the scale factor to ensure that the corners are reached.

Lines 50 and 60 define the shapes of the man and the balls.

Lines 90 & 100 prints the man and puts the jack ball at random on the screen.

Lines 120 to 320 are the main section of the program, calling three PROCedures.

Lines 230 & 240 effectively take the joystick readings as a triangle and then multiplies them out, making larger and larger similar triangles until one side reaches the edge of the screen.

Line 250 takes the provisional final position of the bowl, divides it by 11, then multiplies by the button number that has been pressed. It then plots the progress of the ball across the screen.

Lines 340 to 390 DEFINE PROCWIN. This waits for key 10 to be pressed before printing the winner on the screen. Then it waits again for button 10 to be pressed before starting a new game.

Lines 400 to 450 DEFINES PROCVAL which reads the joystick from A/D channels 0 and 1, then multiplies them by the scale factor. Line 420 inverts the X value, because to be ACORN compatible an increasing analogue value must give a decreasing screen co-ordinate value.

Lines 460 to 590 DEFINE PROCPPB which reads the keypad. Line 490 reads the data from line 590 and uses it to ground each column in turn. After it writes to each column it reads back the rows with R=USR(&FFF4) to get a value for BUTTON which it then converts into an actual number in lines 520 to 570.

4. USING THE DELTA DRIVER PROGRAMS TO CONVERT NON-JOYSTICK PROGRAMS TO RUN ON THE JOYSTICK AND/OR KEYPAD OR TO SET UP THE KEYPAD.

The DELTA DRIVER, on cassette or disc, consists of two programs written for VOLTFACE by C.V.P.

It must be appreciated that the joystick system was designed first and the software came along later, which has caused some confusion as to why certain necessary differences occur between using the joystick with programs written for DELTA 14b joysticks; with programs written for ordinary joysticks and with conversions for making programs, not written for joysticks, appear on the joystick and keypads.

The instructions to use the program will be displayed on-screen.

The JOYSTICK program will convert programs that were not written for

joysticks to run on joysticks providing they were written using INKEY(-ve) commands, or it's machine code equivalent, to test if a particular key has been pressed. JOYSTICK will compile a machine code driver, according to the instructions that you give it; i.e. which keys to replace; and this driver will be placed into the operating system memory area at &A00 to &AFF. If a program does not use the INKEY(-ve) command or if the program encroaches upon the operating system memory area then the driver will not work.

(4a.) One DELTA14b in the analogue socket will allow you to change 6 keys from the keyboard to the joystick. These will be 4 joystick directions and two button functions shown on the driver program as f1a, f1b, f2a and f2b. Note that it allows you to define two keys onto each button. This does mean that you can make f1a and f1b different keys but when any of the top 5 buttons is pressed then the effect will be the same as pushing both keyboard keys together and they cannot be separated. The same is true of the other fire button row using f2a and f2b.

KEYPAD driver cannot be used without an adaptor box fitted.

(4b.) A SINGLE JOYSTICK WITH AN ADAPTOR BOX.

Due to some programs using the user port timer, which affects the line used to switch between the two handsets, it is necessary for a single DELTA 14b in the adaptor box, when used with JOYSTICK program, to be PLUGGED INTO THE SIDE SOCKET OF THE ADAPTOR BOX.

As with a single joystick the 'FIRE' buttons which are the three red buttons and are always commoned together, can, if required, be made to act as if two buttons had been pushed together, though if the functions are required separately as well, then you would need to program some other buttons to take on the functions individually.

(4c) TWO JOYSTICKS AND AN ADAPTOR BOX

If you select two joysticks and an adaptor box in the JOYSTICK program then you are given the option to define both joysticks and keypads. Normally of course these will be the same for both players although one will be running off of ADVAL's 1 & 2 and the other off of ADVAL's 3 & 4. It may be possible to take the keyboard version of a two joystick game and convert it to run on one joystick.

The KEYPAD driver will set up one or both keypads as duplicates of whichever keyboard keys you select. This program works in the operating system so the keypads will work whilst writing programs. Also some BASIC programs will work with these keypads set up, such as the INKEY\$(time) command. When defining the keys in KEYPAD, the pads are distinguished by the words REAR or SIDE above the picture of the keys on the screen. If the flashing number does not change to another key when a keypad key is pressed it is because you are in the wrong socket and you should either plug into the other socket or do a CTRL A to select the other keypad.

CONVERTING THE DRIVER CASSETTE TO DISC

Although protection is incorporated in the "DELTA DRIVER" program, there is nothing to prevent you from transferring the software to disc if you upgrade your system. Follow this procedure for each side of the cassette.

CTRL BREAK

*TAPE

*OPT1,2

*LOAD "" 1900

(Screen displays file NAME, LENGTH, START & EXECUTION ADDRESS'S)

*DISC

*SAVE NEWNAME START ADDRESS LENGTH EXECUTION ADDRESS

USING THE DRIVERS COMPILED BY THE CASSETTE PROGRAMS

When you have defined the keys that you want to be converted to the handset, the DELTA DRIVER program has compiled a machine code driver for you. You can now save this machine code driver as instructed on the screen. The best way of using these drivers is to save them on a cassette along with the program that you will be using it with. First save a small basic program to run both programs, so that a single command will run a converted program. First save this program on tape or disc

```
:-  
10 *LOAD "(Driver Name)"  
20 CALL &A00  
30 CHAIN "(Program name)"
```

Now compile and *SAVE the driver program. Now save the program that you want to run. A single CHAIN command will now load and run the conversion then automatically load and run the main program. Note that line 30 may be *RUN "" for some programs. It will tell you this on the cassette or disc.

PROBLEMS WITH SOME DISC BASED SOFTWARE

Some disc based software resets the O.S vectors on starting up the disc. This cancels out the joystick and keypad conversion if you have run it already. This procedure should get around that problem. Firstly run the JOYSTICK or KEYPAD program and *SAVE (Driver name) A00 +100

Take a formatted disc and create this file called !BOOT as follows:-

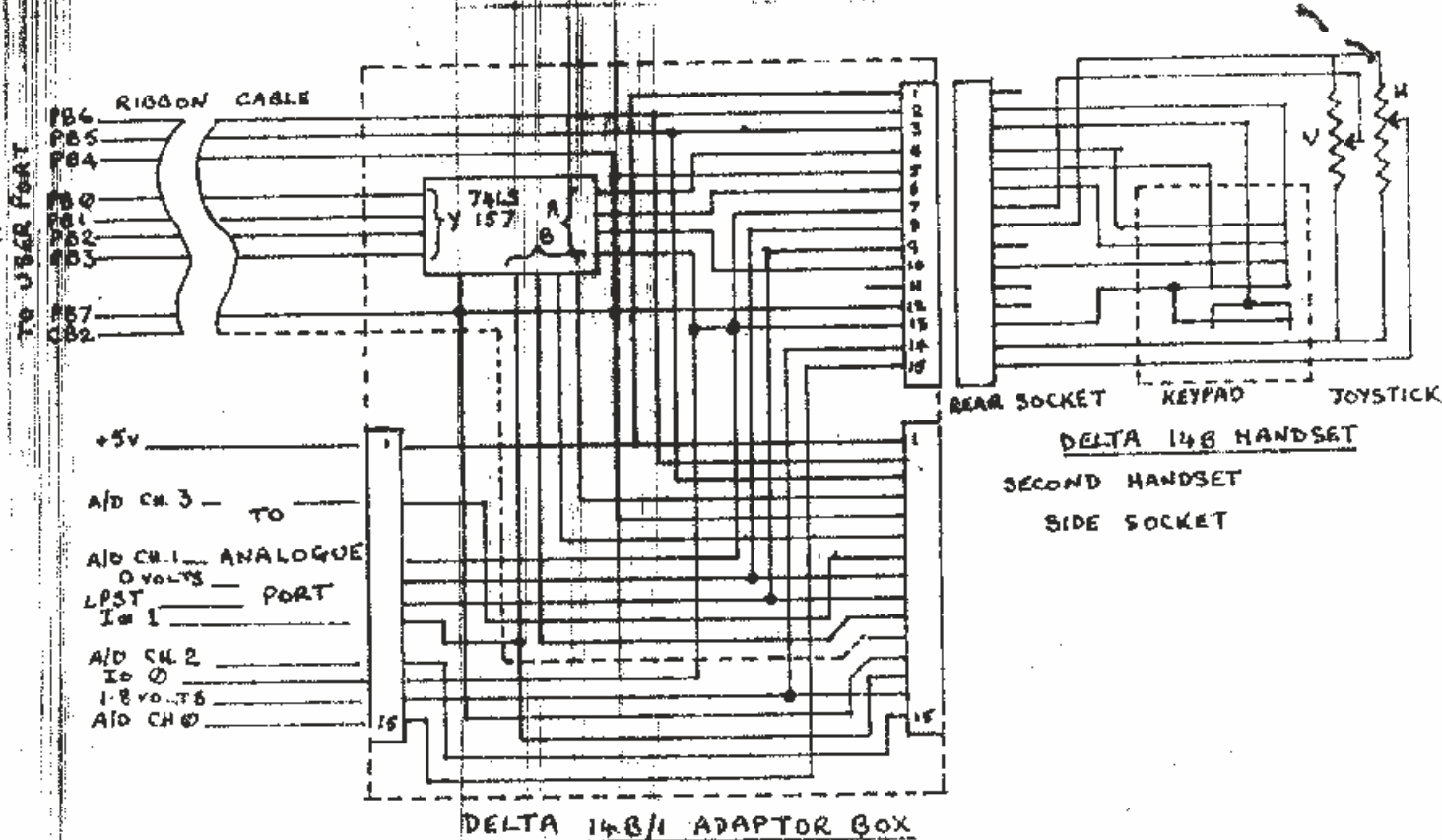
```
*OPT 4,3  
*BUILD !BOOT  
CALL &A00  
TIME = 0:REPEAT UNTIL TIME>1000  
*RUN !BOOT
```

NOW *RUN (Driver name) THEN DO A CTRL BREAK. Put the disc with the new !BOOT file, into the drive and do a SHIFT BREAK. The disc should run and then stop for 10 seconds. As soon as the disc has stopped remove the disc from the drive and put in the disc that you want to run. The disc should then restart without resetting the vectors.

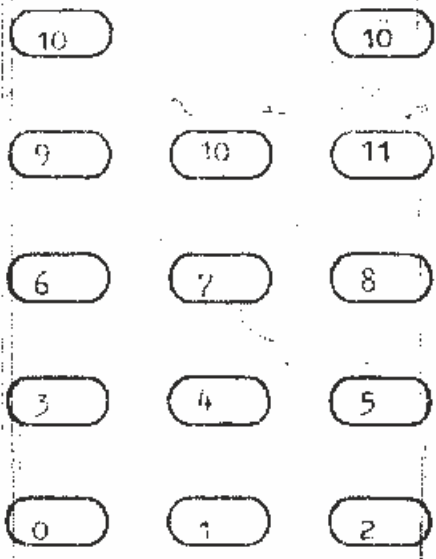
NOTE 1: The *RUN command on O.S.1.2 can cause error messages to be printed on the screen. If you *RUN a driver that you have saved and one of these error messages occur then in all cases that we have experienced if the error message is ignored then the program is found to be running alright. To avoid the spurious messages, *LOAD the program and then CALL &A00 to run it.

NOTE 2: Some programs, where they have run out of normal memory in which to store and run the program, may use &A00 to &AFF in which case the program will crash if a driver has been loaded first.

NOTE 3 Some plug in ROMS are known to cause problems with the driver routines. In particular with the Keypad Driver beware of graphics ROM's disabling the screen refresh interrupt.



DELTA 14B/1 ADAPTOR BOX



The buttons of the handset can, of course be programmed to do what you like. This is the notation used in these instructions to refer to the buttons. Note that the buttons marked 10 are always common and cannot be separated.

KEYPAD OVERLAYS.
The cream overlays supplied with the handset are blank to allow the user to write on their own legend. Additional overlays can either be bought from your stockist or they can be obtained direct from VOLTFACE at 35p per pair, plus 35p per order for postage and packing

VOLTFACE LTD., PARK DRIVE , BALDOCK, HERTS.
0462 894410