

## Low cost

# FORTH DEVELOPMENT SYSTEM

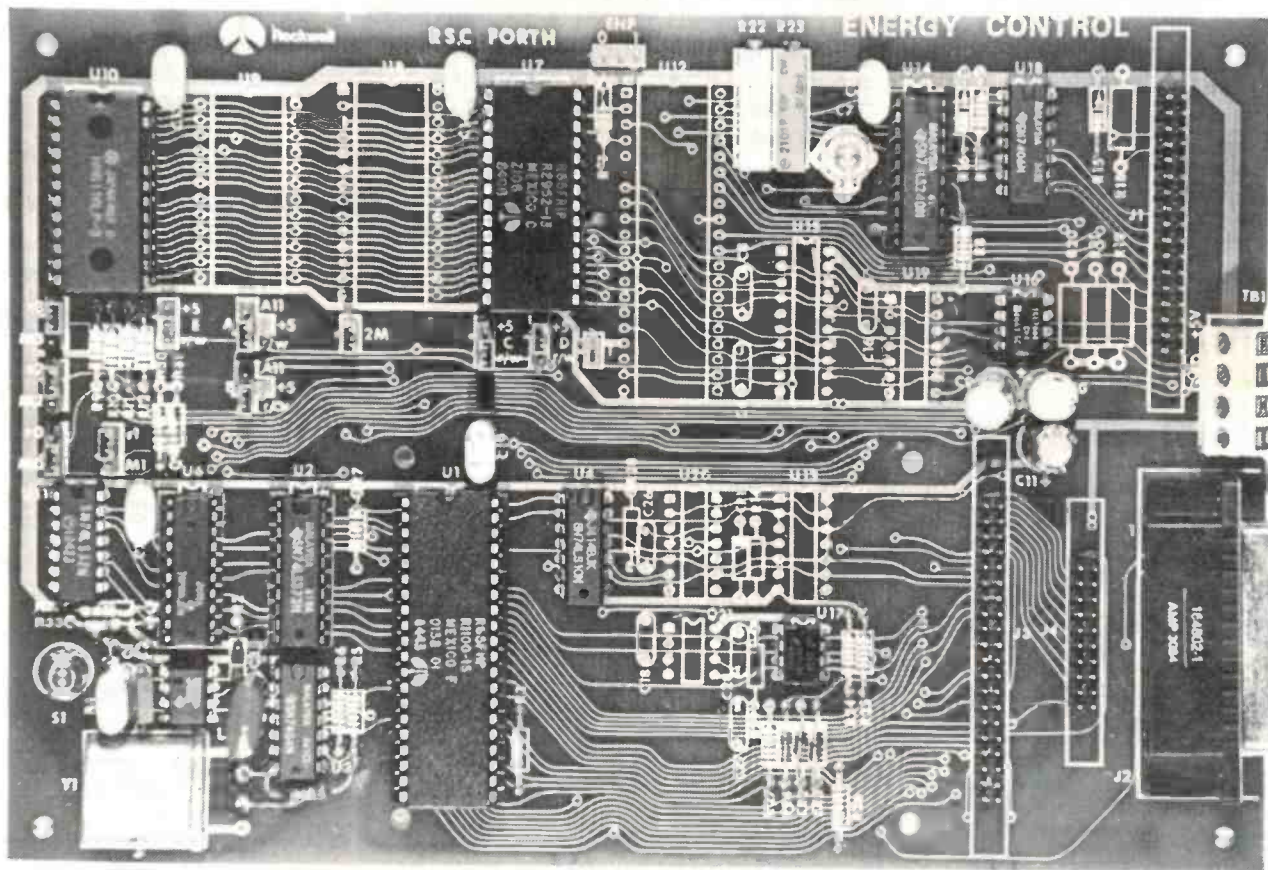
Peter Ihnat

Go FORTH and you will multiply — and add and divide and . . .

FORTH is one of those computer languages you may have heard about but kept away from. However, if you're into controlling

robots, mechanical arms, washing machines, train sets, etc, then it's about time you looked more seriously at Forth. It is a language which was developed with control in mind and is more flexible than either Fortran and Basic when it comes to whizzing stepping motors around or checking the status of sensors.

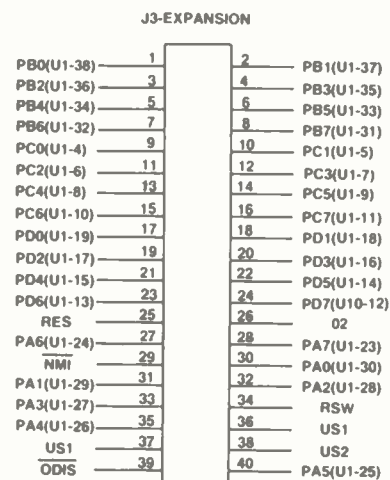
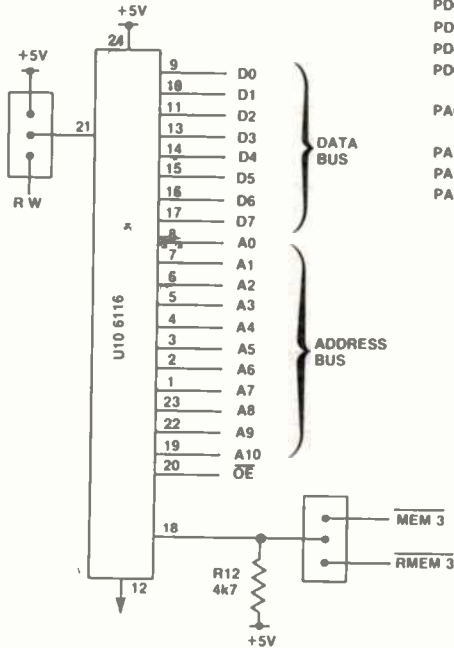
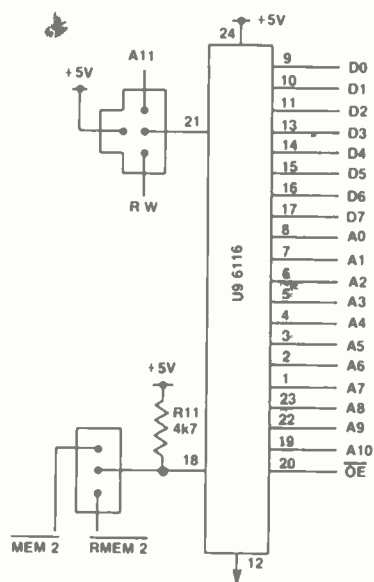
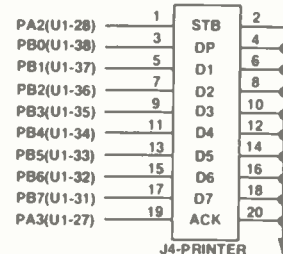
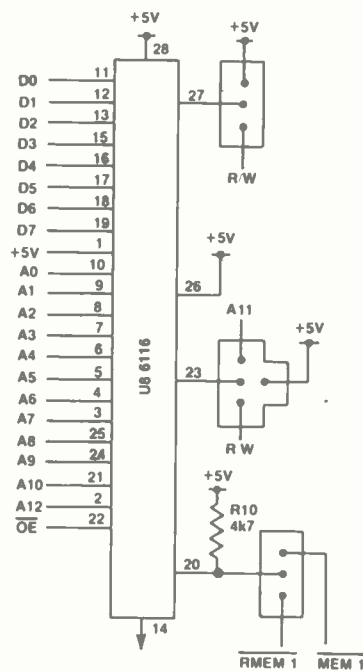
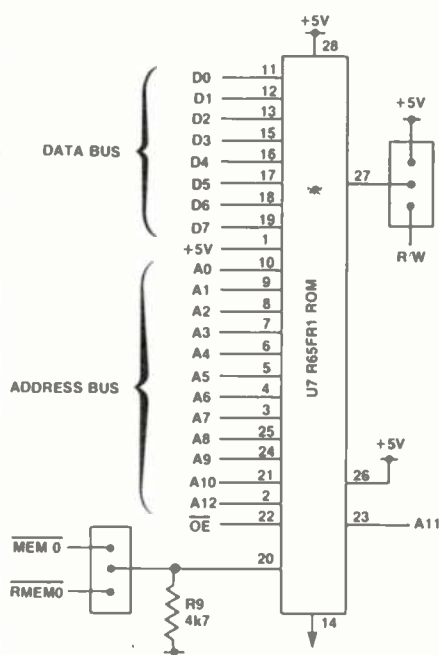
To briefly summarize its features, Forth is a language in which you add your own operations by defining them in terms of previously defined operations until a single word (or operation) represents your whole program. It is a structured language. It uses the concept of Reverse Polish Notation (RPN) and is quite easy to learn. Most peo-



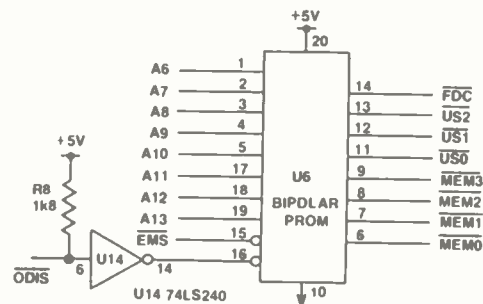




# Project 694



100n BYPASS CAPACITORS  
C13, C14, C15, C16, C17, C18, C19.







link for connection to a serial terminal. The data sheets state that this is a 1200 baud link which would be correct if a 1 MHz crystal was used. However, a higher speed CPU is supplied which requires a 2 MHz clock. The RS232 link becomes 2400 baud with seven data and two stop bits. Higher operating speed is always an advantage.

Finally, there are two 8-bit I/O ports, two 16-bit counter/timers and room to mount up to 16K of RAM, ROM or a combination of both — just about all you could want for a simple development system.

## R65F11 and R65F12 CPUs

The Rockwell R65F11 and R65F12 are complete, high performance 8-bit NMOS single chip microcomputers which are compatible with the whole 6500 family. What makes these devices so special is the fact that they are CPUs which also perform the task of a PIA, ACIA and counter-timer. The R65F11 has the following features —

- it is an enhanced 6502 CPU;
- it contains the kernel of RSC-Forth and an operating system built into 3K of ROM;
- it has 192 bytes of RAM on the chip, 32 bytes of which can be placed into standby mode when the power is off;
- it contains two 8-bit I/O ports including four edge sensitive lines and input latching on one 8-bit port;
- it contains two 16-bit counter/timers with three associated latches;
- it provides a full duplex serial I/O channel; and
- it can address up to 16K of external memory.

Not bad, you say, for a 40-pin package. Those of you familiar with the 6500 family of micros, will realize that the R65F11 is the equivalent of having a 6502 CPU, 6522 VIA and 6550 ACIA *all on the one chip* (as well as a Forth ROM and some RAM). The R65F12 contains all the above as well as three extra 8-bit ports and comes in a 64-pin

QUIP package (quad in-line). If you would like to use the R65F12 CPU, Energy Control sells a special daughter board which replaces the R65F11 and brings out the extra I/O pins.

## Construction

The circuit diagram illustrates the overall system. Construction shouldn't present any difficulties but care should be taken. Use a fine tipped iron and fine solder otherwise solder bridging may occur. Start with the resistors, diodes and capacitors (check orientation of all polarized components). The silk screening on the pc board will indicate the position of all components; remember the board can be **expanded** to include extra memory, a disk **controller** and a Centronics port, so there'll be parts marked on the board that are not needed for this project.

Next, insert and solder the IC sockets and capacitors. Note that IC sockets are recommended for IC1, IC6, IC7 and IC8. If you have doubts about your soldering ability simply use sockets for all ICs. Finally, mount the reset switch, 2 MHz crystal and DB-25 connector (for the RS232 link).

Insert the ICs. Note that they all point in the one direction. Plug the R65FR1 development ROM into position U7 on the board and RAM into positions U8, 9 and 10 (depending on which RAM chips you are using). The memory decoding PROM supplied by Energy Control is set up for the memory map shown in Table 1. Sockets U8, 9 and 10 require 2K RAM chips like the 6116. For other RAM or ROM ICs, you will have to blow your own PROM.

The only tricky part of setting up the system is fitting the on-board jumpers. These customize the memory sockets (U7, 8, 9 and 10) for the particular RAM or ROM IC which will be used. Figure 1 shows the location of the jumpers and Table 2 lists the combinations required. Note that an extra jumper at the position marked 2M is required for operation at 2 MHz.

Finally, the moment. Connect 5 volts to the board, an RS232 terminal and switch on. If all is well, the display will show —

## RSC-FORTH V1.7

Entering the command VLIST will list the entire Forth vocabulary. If not, check all your soldering and especially the on-board jumpers. The user's manual will take you through the whole Forth repertoire. ●

*\*Starting Forth* by Leo Brodie and *Forth Programming* by Leo Scanlon are available through ETI booksales. *Starting Forth* is worth \$25.00. *Forth Programming* will set you back \$24.25. For postage details see the book sales ad elsewhere in this issue.

**TABLE 1. Memory map decoded by PROM supplied by Energy Control.**

Function	Address space	Signal	Socket
FDC	\$0100-\$013F	FDC	U12
RAM1	\$0200-\$07FF	MEM3	U10
RAM2	\$0800-\$0FFF	MEM2	U9
RAM3	\$1000-\$17FF	MEM1	U8
ROM	\$2000-\$3FFF	MEM0	U7
Not used	\$1800-\$1FFF		
page zero	\$0000-\$00FF		
page one	\$0140-\$01FF		

**TABLE 2. Jumper Selection.**

Socket	Part Type	Jumper position		
		PIN 27	PIN 20	
U7	8K x 8 RAM	R/W	RMEM0	
	2764 PROM	+5	MEM0	
	2732 PROM	+5	MEM0	
U8	8K x 8 RAM	PIN 27	PIN 23	PIN 20
	2K x 8 RAM	R/W	All	RMEM1
	2764 PROM	+5	R/W	RMEM1
	2732 PROM	+5	All	MEM1
	2716 PROM	+5	+5	MEM1
U9	2K x 8 RAM	PIN 21	PIN 18	
	2732 PROM	R/W	RMEM2	
	2716 PROM	All	MEM2	
U10	2K x 8 RAM	PIN 21	PIN 18	
	1/2 2732 PROM	R/W	RMEM3	
	2716 PROM	+5	MEM3	

For 2 MHz operation, install 2M jumper. To use three 2K x 8 RAM chips and the R65FR1 development ROM, fit the jumpers highlighted in the above table.



# RSC-FORTH CARD EXTENSION

S.K. Hui

Rockwell's Forth microcomputer card offers the power of a high level language coupled with the speed you could expect at machine level. This, the second in our series on the Rockwell board, describes the implementation of a disk drive and house keeping commands.

EVERY COMPUTER enthusiast has a special regard for the computer manual. Full of jargon and special terms, even the formally trained can find it a little daunting.

Manuals are usually divided into sections, with each section covering a particular command or feature. While individual sections can be quite clear the links between the sections or features can be poorly covered. The other sort of manual of course is the type that is very easy to read but provides only shallow discussion of the hardware and software.

The problems of understanding the computer manual become even greater when you start to work at machine level. The commands needed to store files on a disk, send a file from the screen to the printer or transfer a file from the disk to the printer are no longer self-explanatory.

Fortunately the Rockwell Forth manual is of the type that will become more and more useful as you go along. Unfortunately it is the type that takes some effort to assimilate.

The main approach of this article will be to get you started on the important house-keeping operations so when you make that step into the manual, it will be much less daunting. This approach should suit beginners as well as the computer literate.

## The kit

This kit is somewhat different from most described in this magazine. It is based on a Rockwell single-chip computer that has a 6502 CPU, the heart or kernel of the operating system in ROM, RAM, a timer and input and output ports — all on one chip.

The CPU, pc board and any 'rare' parts are available in a kit from Energy Control in Brisbane. The double sided, solder masked and screen printed boards were developed by Rockwell and Energy Control and will greatly simplify construction. In the May issue this year Peter Ihnat described the construction of the basic computer board. A number of extension kits like the one for the disk drive controller described in this article are also available from Energy Control.

The sections of this article are presented in the same order that you will have to follow in setting up your Forth computer board. It will help if you follow this sequence faithfully.

## Construction and setting up

Whether you already have the basic computer board or are starting from scratch, construction of the kit is relatively simple. You are less likely to be plagued by forgotten feed through wires and solder bridging as the pc boards are plated through and sol-

der masked. A screen printed component overlay simplifies component placement. As a result these construction hints apply equally to the basic or extension kits.

As usual, insert the least sensitive parts first, placing components in the order of links, connectors, switches, IC sockets, resistors and capacitors before soldering in the ICs or placing ICs in sockets.

Be careful to correctly orientate all ICs and electros. The body of the 2 MHz crystal should be connected to ground by soldering a wire between the case and one of the holes in the board at the side of the crystal package.

There are a number of jumpers on the board marked A, B, M2 etc, which configure the board for different CPU and memory options. The arrangement of the jumpers for this project is shown on the component overlay.

You should now be ready to set up the board for initial testing. Connect 5 Vdc to the terminal block (TB1). Be careful of the polarity of the wires and check that the board draws about 400 mA from your power supply.

To communicate with the Forth board, you will need a terminal. If your terminal outputs true RS232 signals with  $\pm 12$  V swings, it can be directly hooked to the Forth board. For some other voltages, you could contact our technical service for help.

To connect the RS232 interface, only three wires are needed. J2 on the Forth board is a DB-25 connector intended for an RS232 serial interface. Pin 3 on the connector is the transmit pin and pin 2 accepts serial signals from your terminal. The ground line on pin 7 should join to the ground pin in your terminal.

Now set your terminal to 2400 baud and 7 bit data with two stop bits, no parity check, no echo and full duplex. If your terminal has the X-ON X-OFF option, leave it out. Turn on the 5 V power supply to the Forth card and your screen should display

RSC-FORTH V1.7

Pressing the reset button on the Forth card or typing COLD then pressing RETURN on your keyboard should cause the same message to be displayed again. A few simple interactive commands can be tried out immediately. Type

VLIST [RETURN]

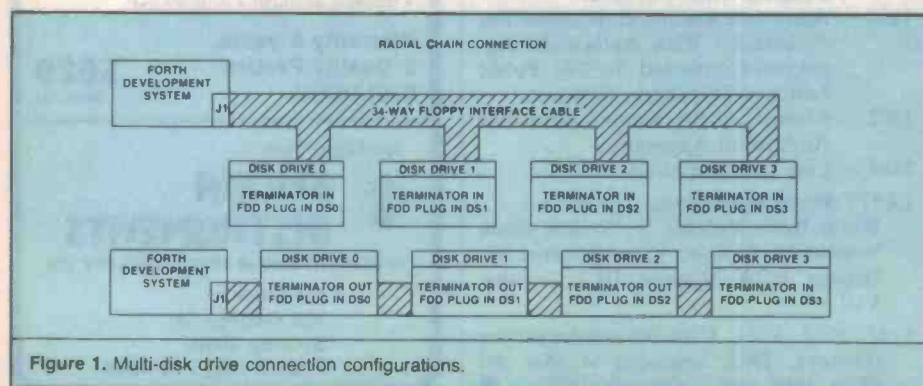


Figure 1. Multi-disk drive connection configurations.



The screen will start to display a list of all the possible Forth words (commands) that you can use. The listing will take a few seconds to finish. Pressing the control key and the S key together on your keyboard will terminate the listing action earlier.

One of the simpler commands you can try is multiplication. Type

5 8 \* [RETURN]

The Forth card will respond with OK immediately after RETURN is pressed. The numbers 5 and 8 are first pushed on to the stack with 8 on the top of the stack. Encountering the \*, the Forth compiler will compile and execute the multiplying function. It pulls the top two numbers from the stack, multiplies them and puts the answer back on the top of the stack. You can verify your answer by typing

[RETURN]

The response is  
40 OK.

The . command allows you to pull the value from top of the stack to be displayed on screen. Other simple commands like +, -, /, .S etc could be explored with the help of the RSC-Forth manual. If everything looks fine, the next step is to hook up the disk drive and the printer.

### Disk and printer set up

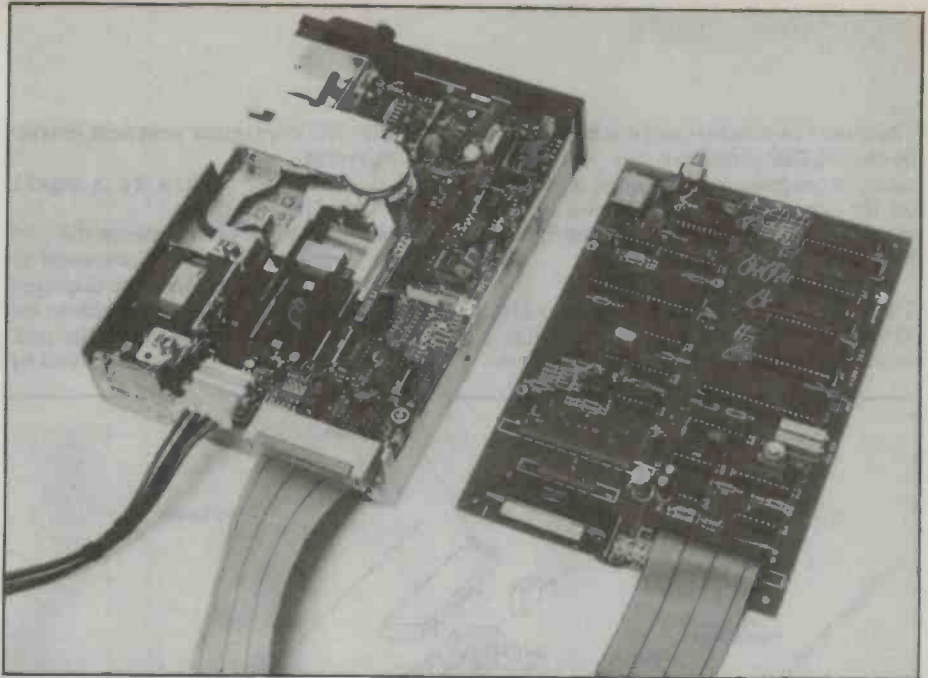
To save trouble and time, I suggest for your disk drive you use the TEC FB-503 40 track, which is available from Energy Control.

The Forth board is capable of controlling up to four disk drives. They can be connected in a daisy chain or radial chain manner as shown in Figure 1. Each drive has a terminator installed when shipped from the factory which looks rather like a 14-pin IC. Remove this terminator if necessary.

The numbering of each drive is determined by the FDD shorting plug. On the assumption that most people will use only one drive, the FDD plug is put in the DS0 position before shipment. DS0, DS1 etc have been marked on the pc board against the jumper socket. Refer to Figure 2.

The connector cables are your next problem. Every disk drive needs two cables; one connects the power and the other is for the I/O signalling. The power cable consists of four wires connecting the drive to +12 V, +5 V and two grounds (see photo). If the TEC disk drive is used, the J1 connector on the Forth card is pin-to-pin compatible with that on the disk drive. A 34-way ribbon cable is used to link up the two units.

The J4 connector on the Forth card is to drive a Centronics printer. To save room on the board, J4 is only a 20-way connector (the standard Centronics connector has 34 pins). On the card end a 20-way ribbon cable is cramped into a normal 20-way connector. At the other end the 20-way cable has to be placed against the pin 1 side of the 34-pin connector (see photo). To print



Disk drive (left) and Forth card (right) showing power cable combination.

something, you need to consult the accompanying printer driver program.

You should expect the disk drive to draw about 200 mA from your 12 V supply when idle. This could go up to 500 or 600 mA when the motors are running. Your 5 V power supply has to be able to supply at least 900 mA when the Forth card (drawing 400 mA) and the disk drive are hooked up to it.

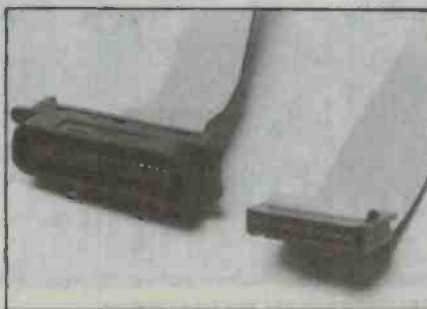
Now turn the power to the system back on. The same start up message RSC-FORTH V1.7 should be displayed. The system needs at least 6K memory to run a disk. So make sure you have 6116 RAMs in sockets 8, 9, 10.

Put a 5¼-inch floppy disk into the disk drive ready for formatting. The disk operating system has to know where the top of the RAM memory is. So on resetting you should type

HEX 1800 MEMTOP [RETURN]

The system will respond with OK. Hex tells the system that any number received will be in hexadecimal. The addresses of the RAMs range from 0200 to 17FF (in hex). The next address is 1800 and this is the one you should nominate with the command

Two ends of the 20-way printer cable, one cramped to a 34-way Centronics plug.



memory top (MEMTOP). Once this parameter is set, you format the disk by typing

DECIMAL 40 0 FORMAT

The disk drive should start to format the disk immediately. DECIMAL tells the system to change back to decimal mode. The 40 tells the system that the disk is 40 track and 0 tells the system to format disk drive 0 (remember that the system can control up to four drives). This job takes about a minute or so to do. When the OK sign is displayed, you are ready to create your program and store it in the disk.

### Disk file handling

A 40-track drive gives you 314 blocks. They are from block 0 to 157 and block 320 to 476. Blocks between 158 and 319 are used by an 80-track drive. Details can be found in appendix N of the Forth manual and the "Disk Interface" box. A block consists of 1024 bytes. Blocks 0 and 1 are reserved for storing system parameters. To create a file or program, type

5 LIST [RETURN]

The system responds by displaying  
SCR #5

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

OK

# Project 696

These are line numbers and can be modified by the >LINE command. Say you want to create a program for displaying a message on the screen when you execute it. You enter the instruction of the program in line 0 by typing

```
0 >LINE : TRY [RETURN]
```

There needs to be a space between 0 and >. The : is a Forth command for creating a word called, in our example, TRY. To de-

fine what TRY does, enter your next instruction by typing

```
1 >LINE '' This is a small
  testing program '';
```

A space must be allowed between the '' command and the This. The command allows the string of characters to be displayed inside the double quotes. To terminate the definition of TRY, the ; is put at the end. You can look at what you have entered by

typing 5 LIST [RETURN] again. The screen will display

```
SCR #5
0 : TRY
1 '' This is a small testing
  program '';
2
3
4
5
6
7
8
9
10
11
12
13
14
15
OK
```

If the listing shows a mistake in any line, it can be corrected by using the x >LINE command, where x is the number of the line carrying the mistake. You can also erase a whole line by typing

```
x >LINE
```

followed by two or more spaces and then [RETURN]

Line x will be totally erased. You can verify this with

```
0 >LINE [RETURN]
5 LIST [RETURN]
```

Line 0 should be a blank line. Now re-type your erased instruction in line 0 to complete your program. To store the program in the disk type

```
FLUSH [RETURN]
```

The system responds with OK. The command INDEX in the Forth dictionary allows you to check what is on the disk. An execution of this command will display the first line in every block in the disk you want to look at. Type

```
DECIMAL [RETURN]
0 10 INDEX [RETURN]
```

The DECIMAL command is just to make sure the 0 and the 10 are not in hexadecimal. The system will display

```
0
1
2
3
4
5 : TRY
6
7
8
9
10 OK
```

Indeed, the program is actually stored in the disk, block 5. Although blocks 0 and 1 seem to be empty, they are reserved for system use. So don't store any user programs there.

You can verify the storing action by turning off the power to the system. Leave it for a few seconds and turn it back on again.

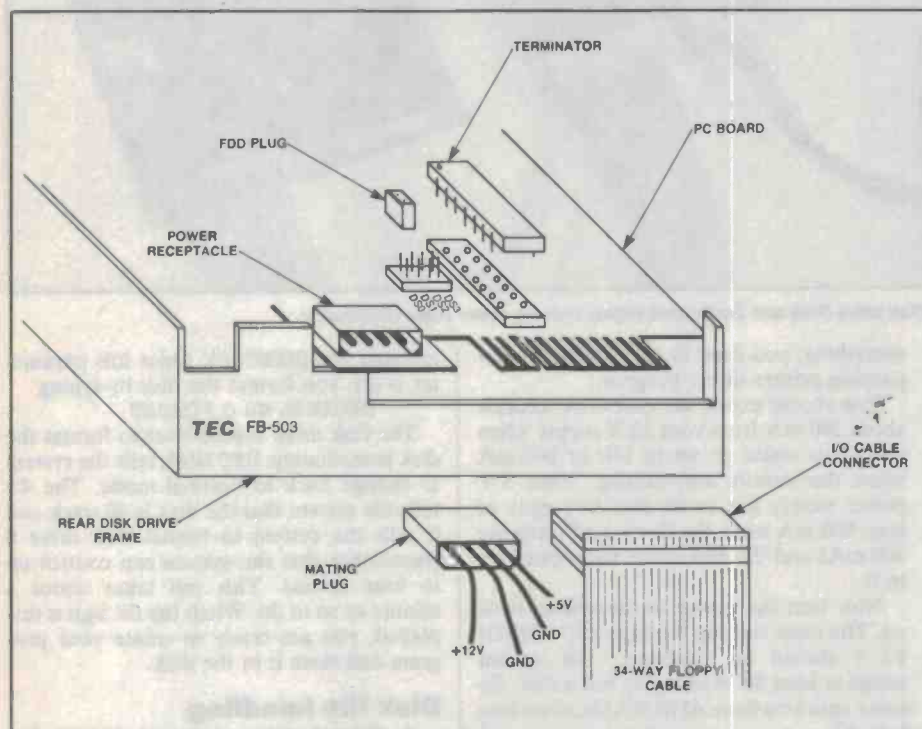
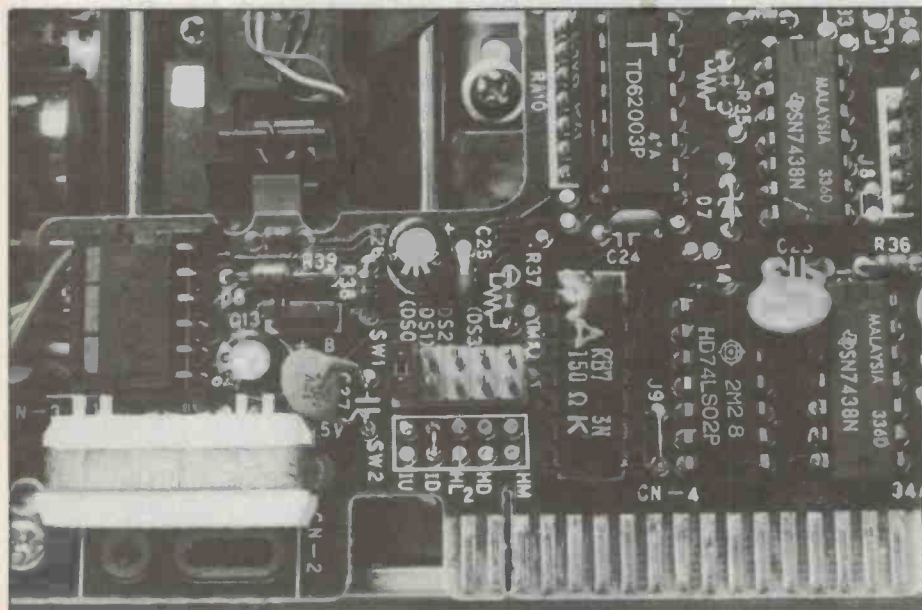


Figure 2. Disk drive hardware set up.



Close up of the disk drive to show the location of the FDD plug connectors and the terminator.



Type  
COLD [RETURN]  
to reset the system. Then initialise the memory for the disk drive again by typing  
HEX 1800 MEMTOP [RETURN]

Do a disk read by typing  
0 10 INDEX [RETURN]

You should get

```
0
1
2
3
4
5 : TRY
6
7
8
9
A
B
C
D
E
F
```

10 OK

Note that this time the 10 is in hexadecimal. The system stays in one number base until it is changed. On the cold start, the default base is decimal until it is changed by hex. To run your program, the easiest way is to type

5 LOAD [RETURN]

The LOAD command will put the program you created in block 5 on to the Forth dictionary. The word TRY you created becomes another Forth word. Calling the word will execute the function defined by the word.

Now try running your program by typing  
TRY [RETURN]

The system will display

This is a small testing  
program OK

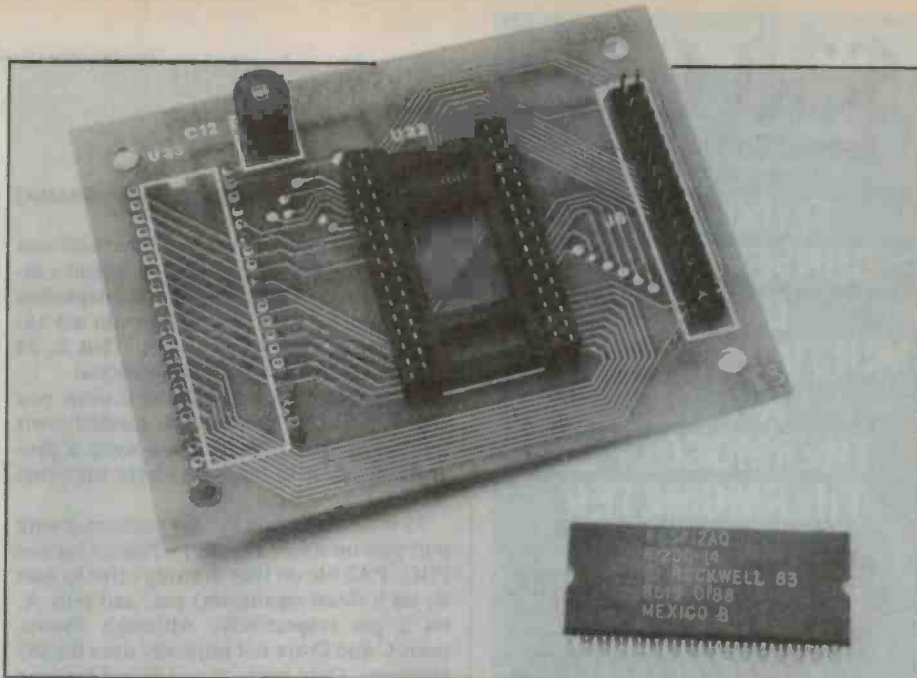
So far so good, but what about erasing files on the disk? You can, of course, list that file and erase line by line until the file is blank and store it back. This is a lengthy method. Unfortunately, the word WIPE which exists in the RSC-Forth manual does not exist in this system. There is no single Forth word to enable you to erase a file on the disk.

But, don't worry, ETI has solved the problem with a program to erase files on the disk. A listing of this program is shown here (as well as in the RSC-Forth manual). This program is created using the commands discussed above. If you want to store this program in block 12 (decimal), type

DECIMAL 12 LIST [RETURN]

Then enter your instructions using the x>LINE command. The finished program looks like the following

```
SCR #12
0 (Word to clear a screen)
1
2 (S - S WIPE BLANKS etc.)
3 : WIPE BLOCK 1024 BLANKS
```



Piggy-back board for the Forth card if 65F12 is used.

4 UPDATE FLUSH ;

5

6

7

8

9

10

11

12

13

14

15 ;S

Store the program in the disk using the FLUSH command. To run it, type

12 LOAD [RETURN]

The definition WIPE created by you is put into the Forth dictionary. Typing

5 WIPE [RETURN]

will erase the block (screen) 5 on the disk for you. You can verify this by typing

0 5 INDEX [RETURN]

Block 5 should be a blank block.

## Printing files

It is not always convenient if you can only check and correct your programs on the screen. The Forth operating system kernel and the development ROM do not allow you to print files from disk to a printer directly. A printer driver program has to be written and stored in the disk for future use.

To print any file, that program has to be called from the disk and compiled using the LOAD command. The listing of the printer program is as follows:

```
SCR #10
0 (Centronics Driver: P-On,
P-Off, Print)
1 (FILTER OUT FORM FEEDS)
BASE @ HEX
2 LATEST DP @ 100 + H/C
3 CODE STB-PRT 10 3 RMB, 0 2
RMB, 0 2 SMB, RTS, END-CODE
4 CODE ACK-WT BEGIN, 11 3
BITSET UNTIL, RTS, END-
CODE
```

```
5 CODE HOUT 1 STA, 0A # CMP, 0=
NOT IF, 'STB-PRT CFA @ JSR,
6 'ACK-WT CFA @ JSR, THEN,
RTS, END-CODE
7 0 HEADERLESS !
8 : P-ON [ ' HOUT CFA @ ]
LITERAL 0046 ! ;
9 : P-OFF F5EF 0046 ! ;
10 : FORM-FEED 0C EMIT
11 ' P-ON LFA ! BASE !
12 : PRINT 1+ OVER P-ON DO DUP
LIST 1+ LOOP P-OFF ;
13 : P-SCR P-ON LIST P-OFF ;
14 ;S
15
```

Store the printer driver program in the disk with FLUSH command. As usual type

10 LOAD [RETURN]

to put the word into the Forth dictionary. To print blocks 10 to 20 from the disk on to the printer, type

10 20 PRINT [RETURN]

If only one block is to be printed, try

10, P-SCR [RETURN]

Whatever is on block 10 will be printed.

There was a problem when I first tried this program. It does not output a line-feed to the printer and, consequently, the whole file gets printed on the same line. To avoid this, you have to choose the auto line-feed option in your printer. This facility is normally selectable by the rocker switches in your printer. For the one I was using, connecting a 0 V (ground) line to pin 14 on the Centronics plug has the same effect.

## Input/output ports

It would be a pretty useless computer if the Forth card could not control general purpose ports with software. A quick look at the specifications of the mother chip 65F11 will leave you stunned. Apart from the usual parallel ports, under your program's control, you can address two 16-bit counter latches and a full duplex serial communication channel.

Hardware interrupt signals applied to ▶



# \$30 OFF!

Special End of Year Offer

Offer closes 31.1.86

## FLUKE 80TK CONVERTS YOUR DIGITAL MULTIMETER INTO A K-TYPE THERMOCOUPLE THERMOMETER

### 80TK SPECIFICATIONS

Temperature Measurement Range  
-50 to 1000°C  
-58 to 1832°F  
Battery Life 1600 Hours (9V)



Treat yourself to a new Fluke Digital Multimeter and you can have the 80TK Thermocouple Adaptor for just \$69 (ex tax) or \$81.40 (inc tax) - a saving of \$30 on the recommended price!

The 80TK plugs into any Fluke DMM to give accurate temperature measurements from -50° to +1000°C using a K type bead thermocouple (included). Micro-connector or screw terminals accept a wide range of Fluke or other probes. Measures in °C and °F

See the 80TK and the complete Fluke Multimeter range at participating distributors.

Offer not available in States or Territories where it contravenes applicable laws.

## ELMEASCO

### Instruments Pty. Ltd.

NEW SOUTH WALES VICTORIA

15 McDonald Street,  
MORTLAKE  
P.O. Box 30, CONCORD  
NSW 2137  
Tel: (02) 736 2888  
Telex: AA25887

12 Maroonah Highway,  
RINGWOOD  
P.O. Box 623, RINGWOOD  
VIC 3134  
Tel: (03) 879 2322  
Telex: AA30418 ELTENT

QUEENSLAND

Tel: (07) 369 8688  
S. AUSTRALIA  
Tel: (08) 344 9000  
W. AUSTRALIA  
Tel: (09) 481 1500

## Project 696

some handshake pins can also be detected by software.

The input/output power of the card can be enhanced if you are willing to spend a little more money. A piggy-back adaptation board with a 65F12 chip will let you use another three 8-bit parallel ports. That is, 24 I/O lines which are fully bidirectional.

The following paragraphs will show you how to read and write to the parallel ports (port A, port B). Before you write a program, it's best to find out where the ports are.

J3 is an expansion header connector with port pins on it (see Figure 1). Names such as PBO, PA2 etc on that drawing refer to port B, bit 0 (least significant) pin, and port A, bit 2 pin respectively. Although shown, ports C and D are not normally used for I/O purposes. Only ports A and B can be used with the 65F11.

After locating which pins on J3 corres-

pond to what bit in the port, you can read the port by the following

COLD [RETURN]

COLD will reset the system. The default of ports A and B is input ports. Now type

HEX PB C@ [RETURN]

The system will respond with OK. HEX changes the number base of the system to hexadecimal. PB C@ causes the voltages on pins of port B to be read and stored in the stack. To read what they are, type

[RETURN]

The system will display

FF OK

meaning all the eight pins on port B are at logic 1.

Now try to earth some of the pins of port B with a wire connected to ground, say, connecting pin PBO for the sake of trying something simple. The logic pattern on port should be 1111110. Type

PB C@ . [RETURN]

### TEC DISK DRIVE ALIGNMENT

There are three adjustments to be made. As they require measurement of quite fast pulses you will need an oscilloscope of reasonable bandwidth to give good results. 15 MHz or better is desirable.

You will note there is a jumper at pin 1 of the 2793. Taking this jumper to +5 volts enables internal precompensation. Taking this pin to ground disables precompensation. The drives used were TEC 40-track drives from Energy Control. The precompensation adjustment was disabled as it was found to be unnecessary. If other drives are used, refer to the manufacturer's data to find what precompensation pulse-width is required. Write-precompensation provides a delay before a write-pulse is issued to the write-head of the drive. The actual delay is adjustable between 100 ns and 250 ns. If you choose to use precompensation carefully follow this procedure:

1. Set TEST (pin 22 U12) to a high.
2. Push the RESET button.
3. Set TEST (pin 22 U12) to a low; inserting a jumper at the link marked T will accomplish this.
4. Observe pulse-width (pin 31 U12).
5. Adjust R22 for the desired precompensation pulse-width.

### DATA SEPARATOR ADJUSTMENT

The following adjustments should be made carefully as misalignment will cause read/write errors:

1. Set TEST (pin 22 U12) high by removing link T.
2. Push the RESET button.
3. Set TEST low by inserting jumper at T.
4. Observe pulse-width (pin 29 U12).
5. Adjust R23 for a pulse-width of 500 ns.
6. Measure the frequency (pin 16 U12).
7. Adjust C7 for 250 kHz at pin 16 U12.
8. Set TEST (pin 22 U12) to a high (remove jumper from T).

### DISK STORAGE WITH FORTH

Disk-based systems have traditionally required a disk operating system (DOS) to handle such functions as program and data storage, file maintenance, loading and execution of programs, and so on. Forth, however, does not provide a DOS in the usual sense. Programs or 'words' as they are referred to in Forth are stored in blocks of 1024 bytes. The words 'block' and 'screen' are often used interchangeably in Forth but they represent the same thing. 'Screen' is perhaps more representative of how a block appears on the user's terminal. A screen is organised as 16 lines of 64 characters. The number of screens that can be stored on a disk is a function of the drive size. For example, an 80-track double density drive can provide 637 screens.

It should be noted that enough system RAM must be provided to allow memory allocation for screen buffers. The bipolar PROM decoder supplied by Energy Control allows three 2K by 8-bit RAM chips. It will be necessary to provide at least 6K to effectively use the disk system.

The heart of the disk interface is the 2793 floppy disk controller (FDC) chip (in socket U12). This chip provides all the functions necessary to step the head, seek tracks, and read/write sectors and tracks.

U21 is used as a monostable to delay the head load timing signal on U12 pin 40. It is triggered by the head load signal on U12 pin 28.

The FDC address is derived from the FDC signal from the bipolar decoding PROM (U6). Forth requires the FDC at \$100. Additional decoding is provided by U13 and U4 for drive status and control registers at \$106. Reading \$106 provides status information on D5 to D7 via U14. Writing to \$106 latches data into U15 to provide control of drive and side select, and also a signal to turn the drive motor on. U18 and U19 are open-collector buffers to interface the disk drive control lines.

The network C6, CR1, R14 attached to U12 pin 23 is a filter that controls the response of the internal VCO of the 2793 during data recovery.



The system should respond correctly with  
FE OK

To write something to port B, you have to send a byte of 00 (in HEX) to port B to turn it into an output (rather than input) port. Once this is done, port B will remain an output until it is changed. Perform this with

HEX 0 PB C! [RETURN]

Now any binary pattern can be loaded into port B by, for example, typing

A5 PB C! [RETURN]

A multimeter may be used to check the voltages appearing on port B pins.

This testing procedure is logical but a bit boring. The accompanying listed program is a lot more interesting. Before you try it, a simple hardware interface circuit has to be built. Use a breadboard if you wish to save time and components.

The schematic of the interfacing hardware is shown in Figure 4. Try this simple program interactively on the console. That is, the program will not be stored in the disk; it is executed directly from the keyboard to be thrown away later. Reset the system with

COLD [RETURN]

HEX 1800 MEMTOP [RETURN]

0 PB C! [RETURN]

DECIMAL [RETURN]

Create your program as usual with

6 LIST [RETURN]

Use x >LINE command to enter your instructions as before. Don't use the FLUSH command if the program is not to be stored in the disk. The listing of the program is shown below.

SCR #6

0 : ( Turn LEDs on/off in a binary pattern )

1 : FLASH CR 500 0 DO I PB C!  
2000 0 DO LOOP LOOP ;

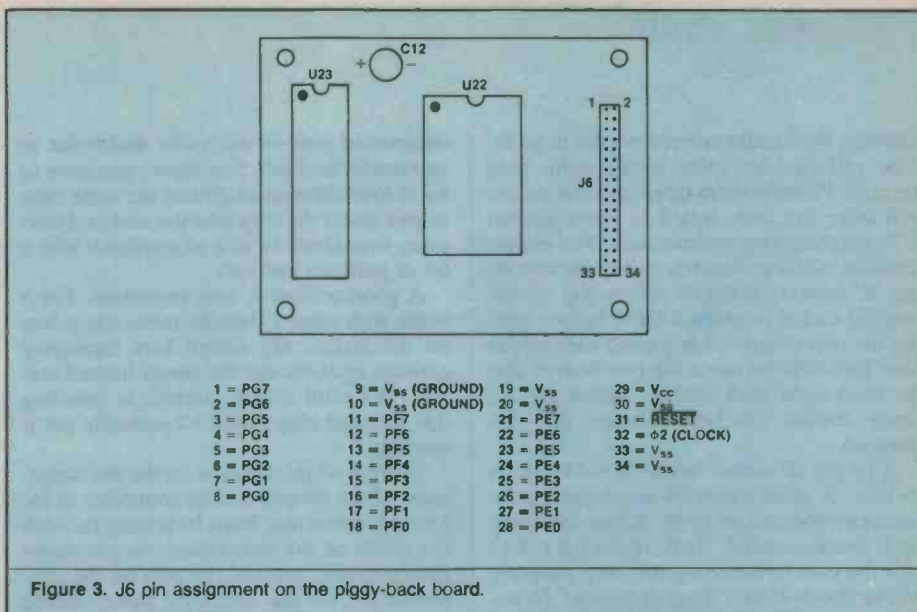


Figure 3. J6 pin assignment on the piggy-back board.

2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

Do a 6 LOAD [RETURN] command straight away. It compiles your program and the new Forth word FLASH is put in the Forth dictionary. To run your program, type  
FLASH [RETURN]

You should see the eight LEDs turn on and off like a counter counting up the binary scale. That is, the LED connecting to

PB0 will flash twice as fast as that to PB1 and four times as fast as the LED connected to PB2 etc. The counting keeps going on until the nested DO loop finishes itself. Changing the 500 to a larger number, say 1000, will double the time this flashing operation lasts before it stops. Changing to 2000 will, on the other hand, affect the flashing rate of the LEDs.

### Precautionary notes

Extra I/O capability of the Forth card can be obtained if 65F12 is used. A small piggy-back adaptation board allows for the different sizes of the 65F11 and 65F12. The white silkscreened position U23 is not for soldering a chip on to. Instead, two 20-way header connectors have to be soldered on the bottom side of the board.

The pins of the headers connect the two boards up electrically. Naturally, the 65F11 (U1) has to be unplugged from its socket, ►

## MICROMASTER—Simply Brilliant

MicroMaster is a new single board computer with more features and higher performance than previously thought possible in a board of this size. Based on the STD bus, it is possible to expand the MicroMaster into an extremely powerful multi-user system with hard disks. A friendly user interface and performance benefits are provided by a CP/M compatible operating system with MSDOS and UNIX—like features.

### Features

- High performance microcoded CPU with pipelining
- Z-80 compatible but faster
- up to 512K bytes RAM with MMU
- monitor/boot PROM in 2732-27256
- two DMA channels for fast data transfers
- two 16-bit counter-timer channels
- 12 level interrupt controller
- two RS232 serial ports with on-board drivers
- Centronics parallel port
- Floppy disk controller for 3½, 5¼, 8-inch drives
- Support for new dual speed 5¼-inch drives
- Compatible with CP/M, Z-System, CP/M3, TurboDOS
- Full 12 months warranty.

**\$750**

(plus sales tax)

For more information or to place orders(!) contact:

**Microtrix Pty. Ltd. 24 Bridge Street, Eltham, Vic., 3095. Ph. (03) 439 5155**



Distributor and OEM enquiries welcome.

# Project 696

allowing the header connector pins to go in. You will find the pins aren't quite long enough. Pushing them deep into the socket will force the little board to press against U1's neighbouring components. The easiest solution to this problem is to buy another 40 pin IC socket. Stack it up on top of the original socket to make it taller before joining the two boards. It is a good idea to put four standoffs between the two boards and fix them tight with screws. Holes are already drilled on both boards for this purpose.

A 64-pin IC socket has to be soldered on to U22. A double row 34-way header connector is soldered on to J6. A fine soldering tip is recommended. Next, insert the 65F12 into the socket. Inserting this chip properly will be much harder than you think! To ensure good contact between the IC and the socket, pins of the IC are flared outwards. The pins are forced back when you insert them.

The fact that this 'fat' monster has two

columns of pins on each side makes for an extremely hard job. Somehow, you have to bend four columns of pins at the same time as you insert the chip into the socket. However, you should be able to manage it with a bit of patience and care.

A good contact is very important. For a while with mine I thought there was a bug on the board. My screen kept displaying garbage symbols and the cursor buzzed crazily. Not until a fifth attempt at inserting this monster chip could I eventually get it working.

There is no information on the pin assignment of the 34-way header connector in J6. I have tracked this down by tracing through the tracks on the little board. As you might have guessed, most of the pins on the connector are for the additional ports offered by 65F12. The pin assignment for J6 is shown in Figure 3. This uses the convention of PEO, PF6 etc, where PE, PF are ports E and F respectively and 0 and 6 refer to the bit position of the specified ports.

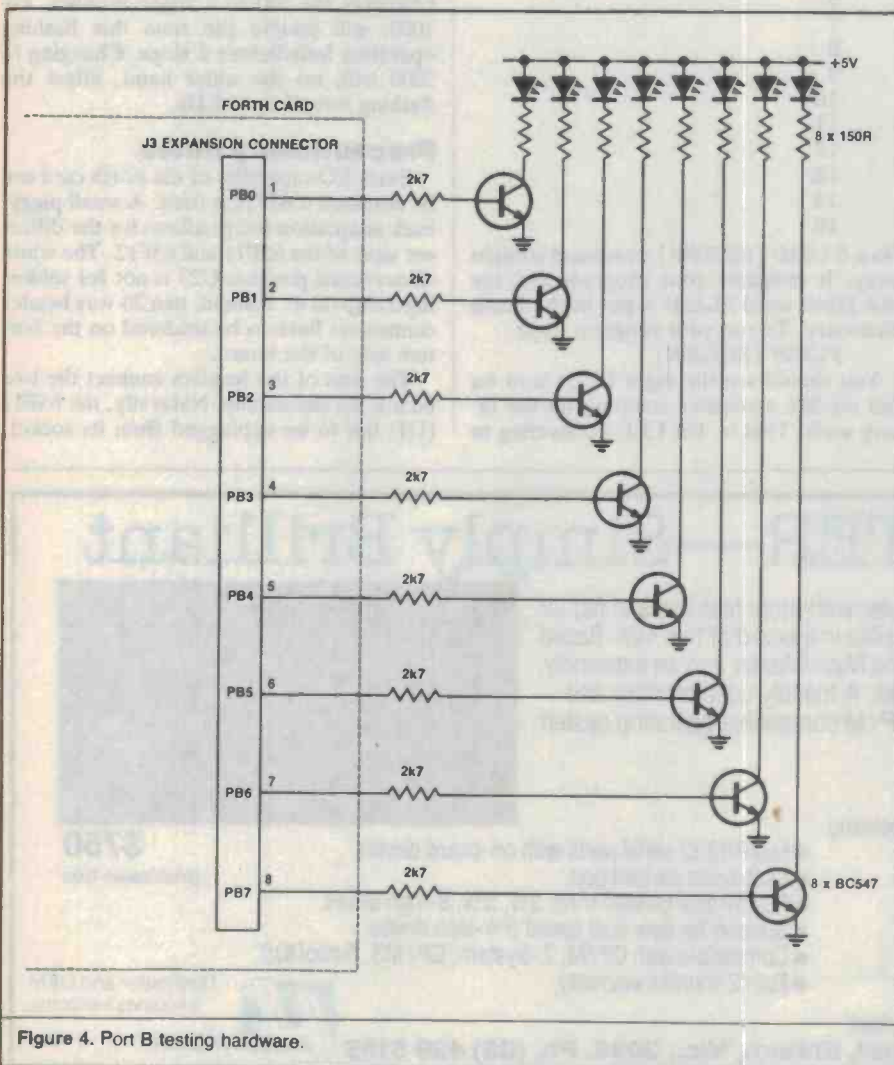


Figure 4. Port B testing hardware.

## PARTS LIST

The basic model consists of 2 MHz crystal, 65F11, bipolar PROM, R65FR1 ROM and pc board, and costs around \$159. You can enhance the I/O power by replacing the 65F11 with 65F12. The piggy-back adaptation board with the 65F12 and a few header connectors necessary for connecting to the mother board cost \$54. The Forth card cannot drive a disk drive without the 2793A disk controller chip. The chip itself costs \$27. A 40-track TEC-FB503 disk drive complete with connectors is available for around \$200. All the above parts are available from Energy Control. See below for full parts lists.

### PARTS FOR BASIC FORTH CARD

Resistors.....all 1/4 W, 5%

R1, 2.....	1M
R3, 9-12.....	4k7
R5, 6, 8, 25-27.....	1k8
R7.....	330R
R24.....	2k2
R30-33.....	4k7

#### Capacitors

C1, 13-19.....	100n ceramic bypass
C2.....	10n
C3.....	220n
C4, 5.....	10p
C9, 10.....	10µ 16 V RB electro
C11.....	100µ 16 V RB electro

#### Semiconductors

CR1-4.....	1N914
U2.....	74LS373
U3.....	74LS04 (must be LS)
U4.....	74LS10
U5.....	'555
U8-10.....	6116 2K RAM
U11.....	74LS32
U14.....	74LS240
U16.....	ICL7660
U17.....	LM1458
Y1.....	2 MHz crystal

#### Miscellaneous

1 x 40-pin IC socket; 2 x 24-pin IC sockets; 2 x 28-pin IC sockets; 1 x 20-pin IC socket; DB-25 female connector (right angle pc mount) normally open pushbutton; and the RSC-Forth development kit\* which includes:

U1.....	R65F11
U6.....	256 x 8 bipolar PROM
U7.....	R65FR1 development ROM

pc board and literature.

### PARTS FOR DISK CONTROLLER

#### Resistors

R13.....	4k7
R14.....	1k
R15, 16.....	10k
R17.....	1k8
R18-21.....	150R 1/2 W
R22.....	10k multi-turn trimpot
R23.....	50k multi-turn trimpot
R28.....	10k
R29.....	470k

#### Capacitors

C6.....	220n
C7.....	10-100p trimming capacitor
C26, 27.....	10n
C28.....	1µ5

#### Semiconductors

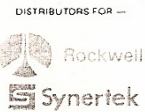
CR1.....	1N914 diode
U12.....	WD2793*
U14.....	74LS240
U15.....	74LS273
U18.....	7406
U19.....	7406
U21.....	'555

\* available from Energy Control.





ENERGY CONTROL Pty. Ltd. (Inc. in Queensland.)  
INDUSTRIAL CONTROL SYSTEMS ENGINEERS



P.O. Box 6502, Goodna, Qld. 4300.  
BRISBANE - AUSTRALIA  
Phone: 61-7-288 2455 - 288 2757  
Telex: AA43778 ENECON

P.O. Box 12153,  
Wellington North,  
WELLINGTON  
NEW ZEALAND  
Phone: 64-4-726462

### LOW COST FORTH DEVELOPMENT SYSTEM

Dear Customer,

The RSC Low Cost Development System has been designed for the computerist wishing to gain maximum benefit for both time and money spent. Standard features of the system firmware include

- (i) Disk interface
- (ii) Centronics printer port
- (iii) EPROM burning capability

The project consists of the Starter kit, with a number of additional upgrade modules available - pricing and details as follows:

#### STARTER KIT (EC-1F11): consisting of

- (i) Low Cost Development System Main PCB D.S. Plated through holes and silk screen
- (ii) The complete RSC Forth Manual and any ancillary notes
- (iii) R65F11 Forth Kernel and Microcomputer
- (iv) R65FR1 Development ROM
- (v) Programmed Bipolar PROM

Tax Free \$142-00  
Tax Included \$159-00

#### UPGRADE KIT 1 (EC-1F11/DD): consisting of

- (i) DSDD 40 track Disk Drive
- (ii) 2793 Disk Controller
- (iii) Flat cable and connectors
- (iv) In-line Power Supply +5V 1.5A; +12V 0.5A

Tax Free \$297-00  
Tax Included \$337-00

#### UPGRADE KIT 2 (EC-1F11/12): consisting of

- (i) Daughter board for R65F12 adaption to Low Cost Development Board
- (ii) R65F12 Processor for increased I/O capability

Tax Free \$54-00  
Tax Included \$62-00



P.O. Box 6502, Goodna. Qld. 4300.  
BRISBANE - AUSTRALIA  
Phone: 61-7-288 2455 - 288 2757  
Telex: AA43778 ENECON

P.O. Box 12153,  
Wellington North,  
WELLINGTON  
NEW ZEALAND  
Phone: 64-4-726462

## UPGRADE KIT 3 (EC-1F/CASE):

Metal case to accommodate 1 x disk drive and Forth Board

With the interest being shown in the Forth development systems, we presently have in the pipeline a number of "spin-offs" based around the Rockwell Forth microprocessor, kernel and development chips. These will be featured in "E.T.I." features over the next few months.

### June, 1985

R65F12 Minimum System - provision for 8K RAM, 8K ROM.  
Battery back-up. R65F12A (2MHz), 4MHz crystal, manual.

### July, 1985

R65F12 Daughter board with:

- (i) 2 channel D/A
- (ii) 8 channel A/D
- (iii) Watchdog timer
- (iv) Manual, PCB and program disk

### August, 1985

R65F12 Daughter board with:

- (i) EPROM Programmer for up to 27256 and fast programming algorithm and hardware
- (ii) Bipolar PROM Program
- (iii) Hardware capability for 24 pin PAL devices and software as a future project. PCB and manual and program diskette.

Prices correct at time of printing. The right is reserved to change pricing without prior warning.



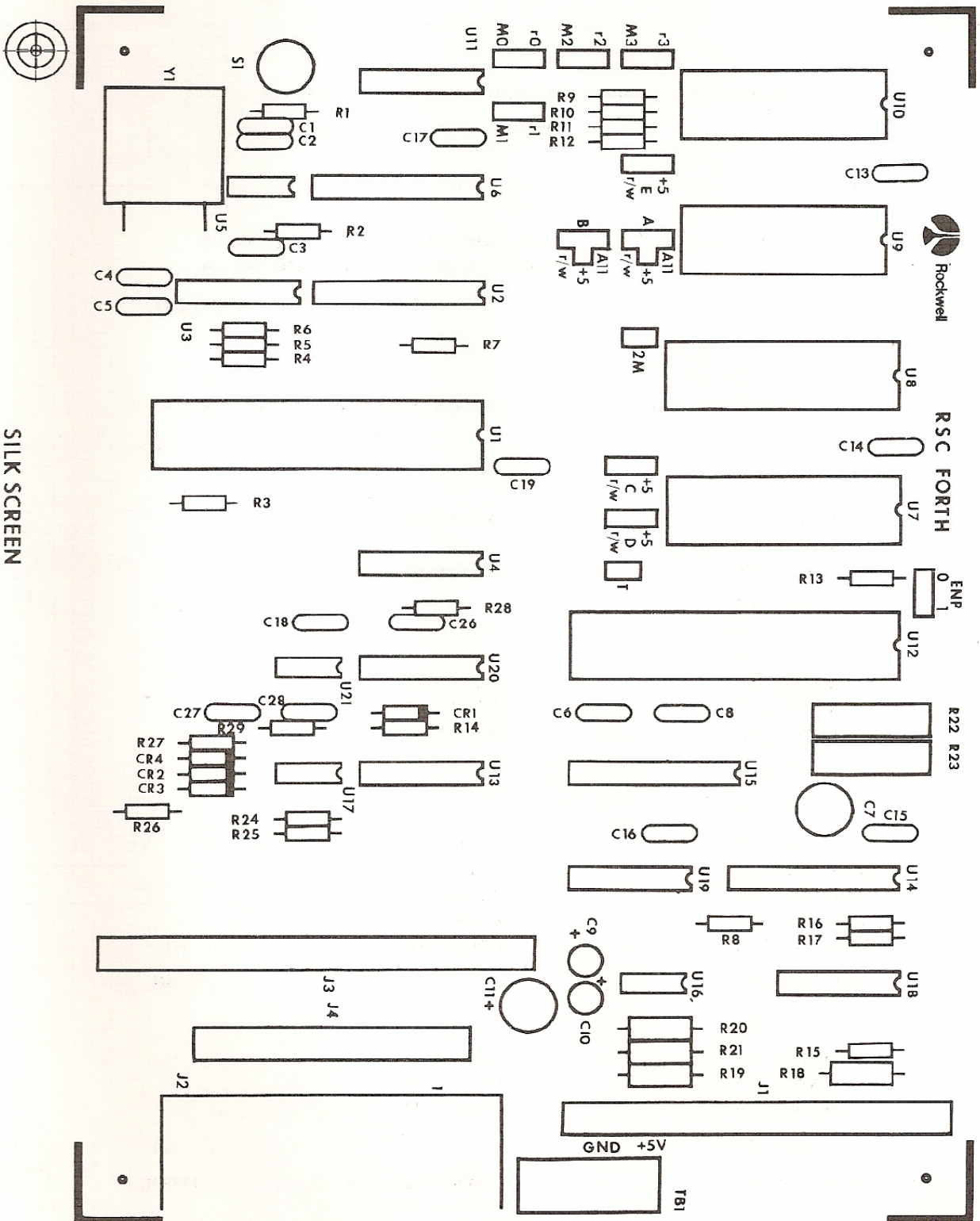
Table 1. Decode PROM (U6) Coding

The following configuration is assumed:

Function	Address Space	Code	Signal	Socket
FDC	= \$0100-\$013F	= 7F	= FDC	= U12
RAM1	= \$0200-\$07FF	= F7	= MEM3	= U10
RAM2	= \$0800-\$0FFF	= FB	= MEM2	= U9
RAM3	= \$1000-\$17FF	= FD	= MEM1	= U8
ROM	= \$2000-\$3FFF	= FE	= MEMO	= U7
Not Used	= \$1800-\$1FFF	= FF	= No Chip Selected	
Page Zero	= \$0000-\$00FF	= FF	= No Chip Selected	
Page One (ex FDC)	= \$0140-\$01FF	= FF	= No Chip Selected	

[illegible]

REDUCE TO 7.550 ± .005

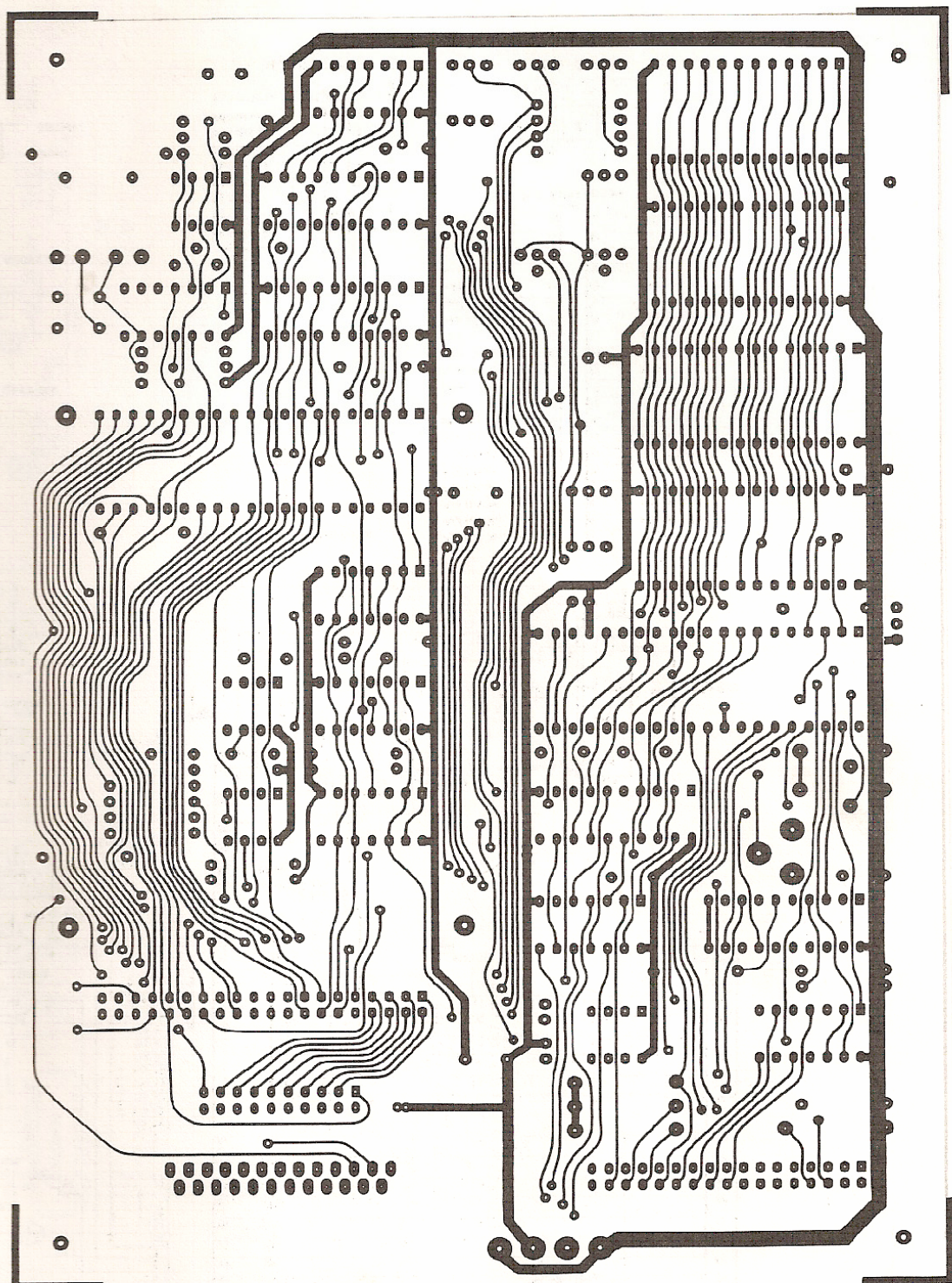


SILK SCREEN



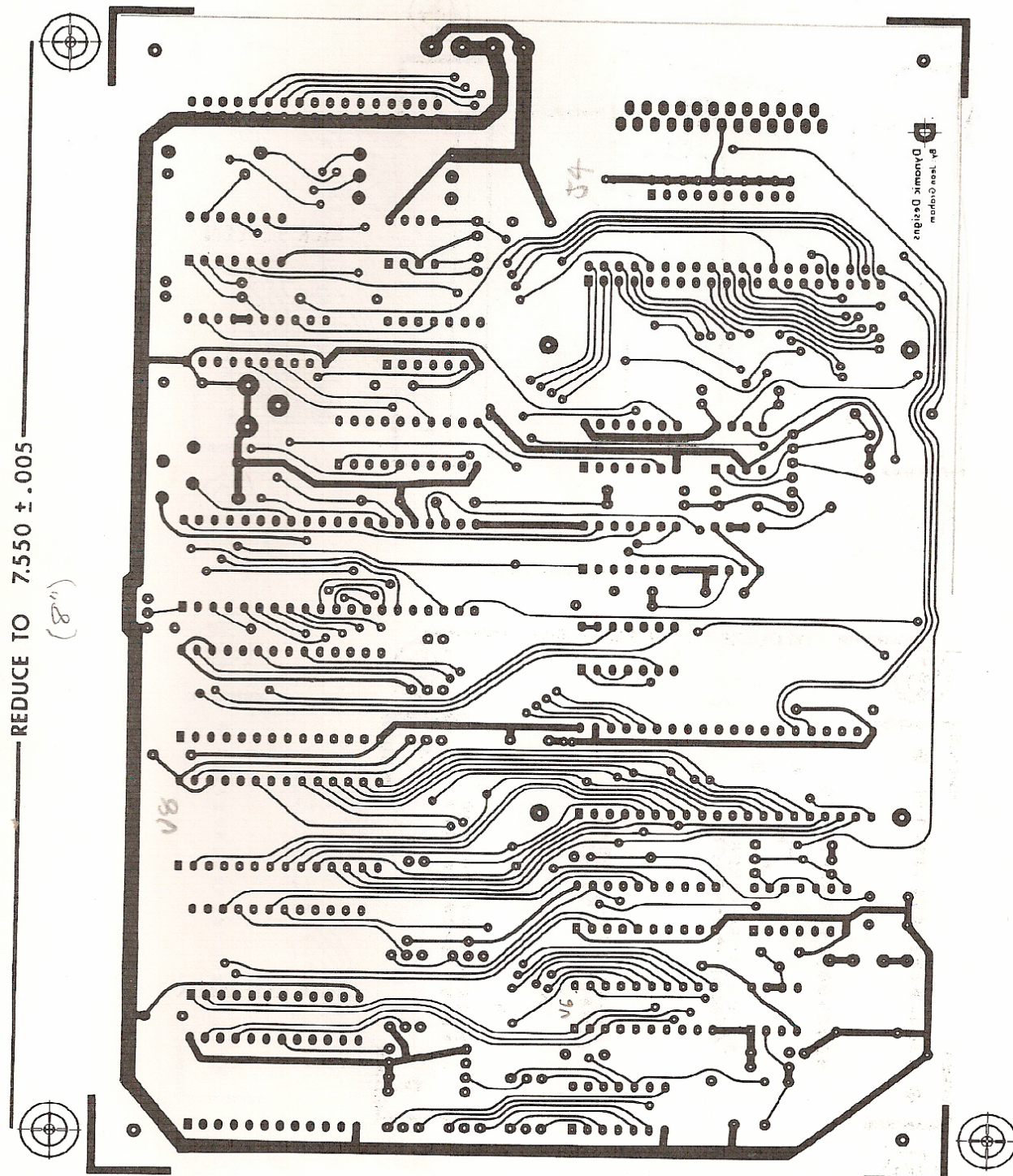
REDUCE TO 7.550 ± .005

(8")



COMPONENT SIDE





REDUCE TO 7.550 ± .005

(8")

DYNAMIC DESIGN

SOLDER SIDE  
(REVERSE)