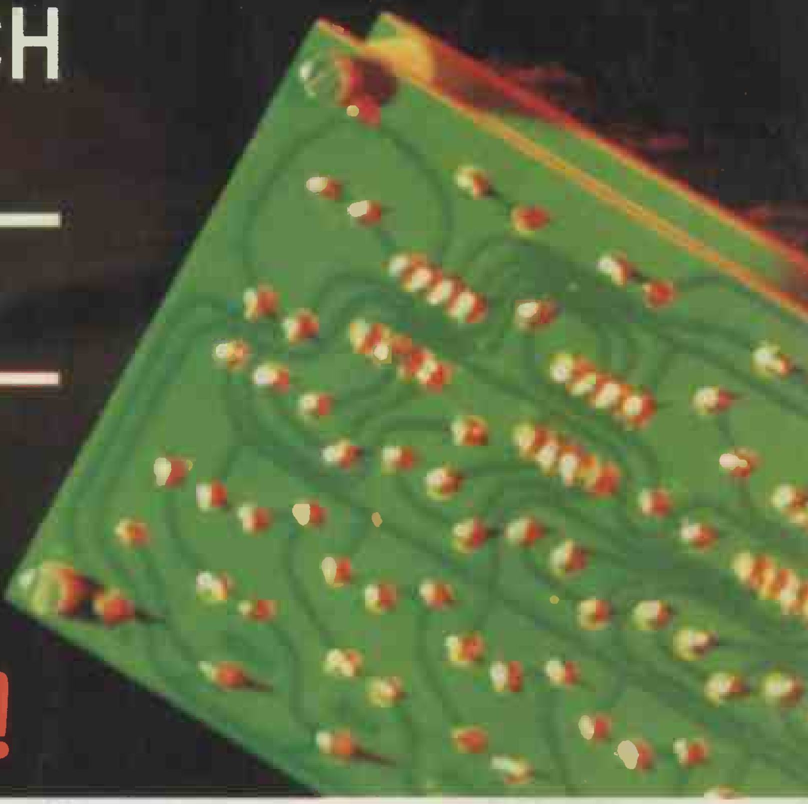
EVERYDAY
ELECTRONICS

MAY 1990

INCORPORATING ELECTRONICS MONTHLY

£1.40



AMSTRAD SPEECH SYNTHESISER

QUIZMASTER

ELECTRONIC BAROMETER

FREE INSIDE!

SPECIAL AMATEUR RADIO SUPPLEMENT

SHORT WAVE RECEIVER

DIY AERIALS

ISSN 0262-3617

05

9 770262 361003

The Magazine for Electronic & Computer Projects

VOL 19 No 5 MAY 1990

EVERYDAY ELECTRONICS

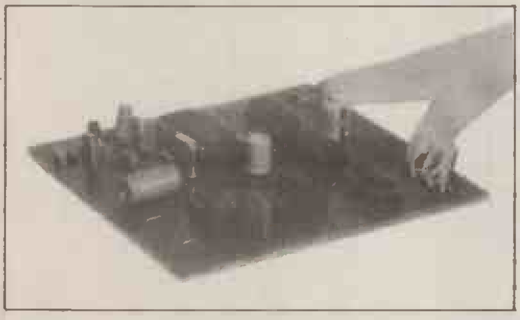
INCORPORATING ELECTRONICS MONTHLY

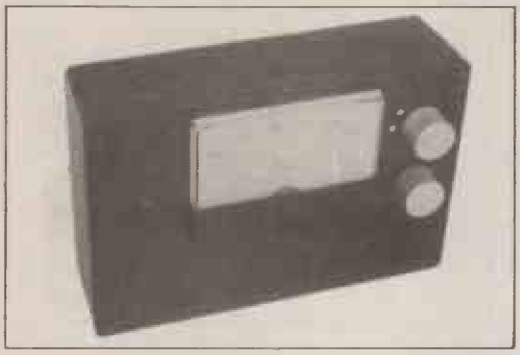
ABC

The Magazine for Electronic & Computer Projects

ISBN 0262-3617

PROJECTS . . . THEORY . . . NEWS . . .
COMMENT . . . POPULAR FEATURES . . .





Cover photograph by Reflections of Bournemouth

Projects

Series

Features

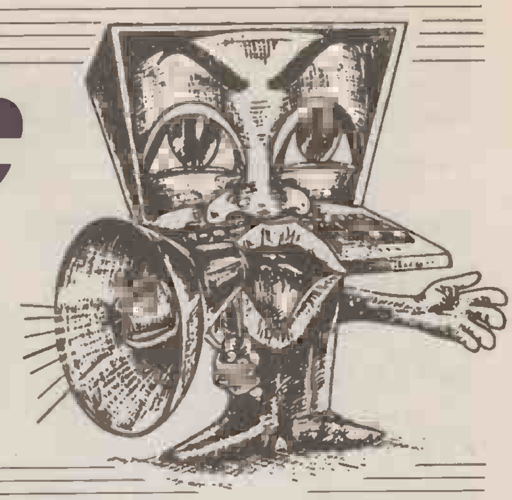
FREE **between pages 326 and 327**

AMATEUR RADIO SUPPLEMENT (12-pages)
Shortwave Reception — Crystal Set — DIY Aerials — T.R.F. Receiver

Our June '90 Issue will be published on Friday, 4 May 1990. See page 295 for details.

Constructional Project

AMSTRAD CPC SPEECH SYNTHESISER



ROBERT PENFOLD

Low-cost voicing for your Amstrad. Based on the popular SPO256 "speech" chip.

Although they rank as one of the most popular computer ranges, the number of published add-on projects for the Amstrad CPCs seems to be relatively small. Possibly this is due to the fact that Amstrad launched these computers after the main home computer boom, when interest in computer projects had waned somewhat. Anyway, this article describes an add-on speech synthesiser for any computer in the CPC range, and helps to rectify the lack of projects for these machines.

The unit fits onto the CPC expansion port, and provides its speech output via the computer's built-in loudspeaker. The circuit is based on the popular SPO256 speech synthesiser chip. This is a relatively cheap speech chip, but is one which gives an unlimited vocabulary.

Unlike some methods of speech synthesis, the one used by the SPO256 does not require large amounts of the computer's memory in order to store a few words. In fact a dozen long sentences would probably require no more than about 1K of memory. The speech quality is not as good as some systems, but is quite acceptable. Considering the simplicity of the system used in this method of speech synthesis, the speech quality is remarkably good.

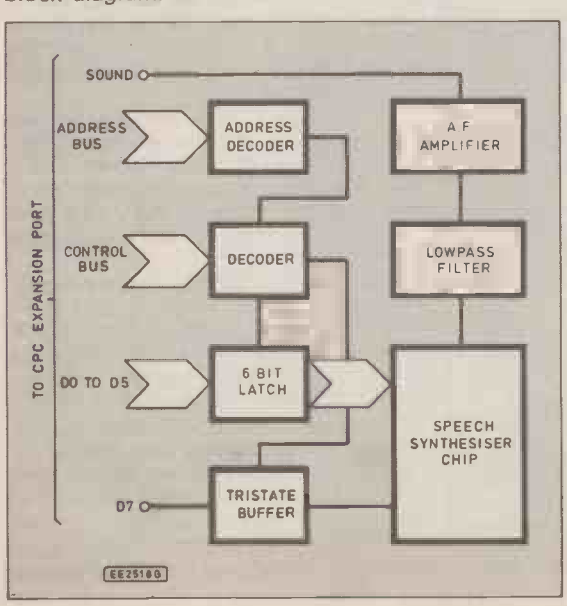
ALLOPHONES

There is a slight drawback in using the SPO256 speech chip, and this is simply that it is relatively difficult to program well. It produces words by stringing together a number of "allophones", which are the basic sounds of speech. For example, "shhhh", "arrr", and "ow" are the sort of simple sounds that can be fitted together to make up the more complex sounds of complete words. These words, separated by pauses where necessary, can then be put together to make complete sentences, or even a number of sentences if required.

For this system to be successful you must have at your disposal all the sounds needed to make up all the words in the language. Apparently the sounds required vary significantly from one language to another, and the sounds for a particular language are called "phonemes". The SPO256 contains sixty four phonemes (including pauses) that permit any English word to be produced.

For this system to work well the allophones need to be chosen very carefully, and the obvious ones are not necessarily the ones that give the best results. However, with a little experience it is possible to put together some intelligible sentences in a reasonably short space of time, perhaps with a little "fine tuning" being needed in order to get things just right.

Fig. 1. The Amstrad CPC Speech Synthesiser block diagram.



SYSTEM OPERATION

The block diagram of Fig.1 shows the basic arrangement used in this speech synthesiser. The speech synthesiser chip is at the heart of the unit, and this includes a 2K by 8 ROM that contains all the allophones. It has provision for operation with up to three external ROMs, giving a maximum capacity of 256 allophones. However, in most circuits (including this one) no external ROMs are used. There are 64 allophones available to the user, and these are selected via six address inputs.

Rather than a true audio output, the SPO256 provides a pulse width modulated output signal. In other words, the output signal is a high frequency pulse signal, with the pulse width being varied to give the required audio output voltage. A high mark-space ratio gives a high average output voltage, a low mark-space ratio gives a low average output potential, and a 1 to 1 ratio gives an intermediate average output level.

By varying the pulse width the average output voltage can therefore be made equal to the required audio output potential. In order to decode this type of signal it is merely necessary to pass it through a lowpass filter. This filters out the pulses and leaves a potential equal to the average voltage of the pulse signal. The output signal from the lowpass filter is too low to drive the audio input of the CPC computers properly, and so the signal is boosted by an amplifier before being fed to the computer.

INTERFACING

Interfacing the SPO256 to the buses of a computer is reasonably straightforward. In order to produce a series of allophones from the chip it must first be fed with a 6-bit allophone address. Then, in order to get the chip to produce the selected allophone, a pulse must be applied to the chip's "ALD" (allophone load) input.

It is important that the next allophone is not sent to the SPO256 until it has completed the current one. Otherwise, each allophone, apart from the final one, is cut short to make way for the subsequent one, giving what is just a brief burst of noise from the loudspeaker.

The SPO256 provides a sort of "handshake" output in the form of its "LRQ" (load request) output. This is normally low, but it goes high when the chip is producing an allophone.

The flow of data can be correctly controlled by using a software routine to send an allophone address to the SPO256, wait until "LRQ" goes low, send another allophone address, monitor "LRQ" again, and so on, until all the allophones have been produced. Timing loops are not really a viable alternative as the allophones have various durations from 10ms to 420ms. An accurately timed flow of data is essential if a good quality output is to be achieved.

In my experience at any rate, feeding the address inputs of the SPO256 direct from one of the computer's buses is not likely to be successful. More reliable results are usually obtained if it is fed from a latching output port. In this case it is driven from a 6-bit latching output derived from data lines D0 to D5.

The "ALD" input of the speech chip is driven from the "write" output of the address/control bus decoder circuit. Consequently, writing an allophone address to the SPO256 results in that allophone being automatically produced by the device.

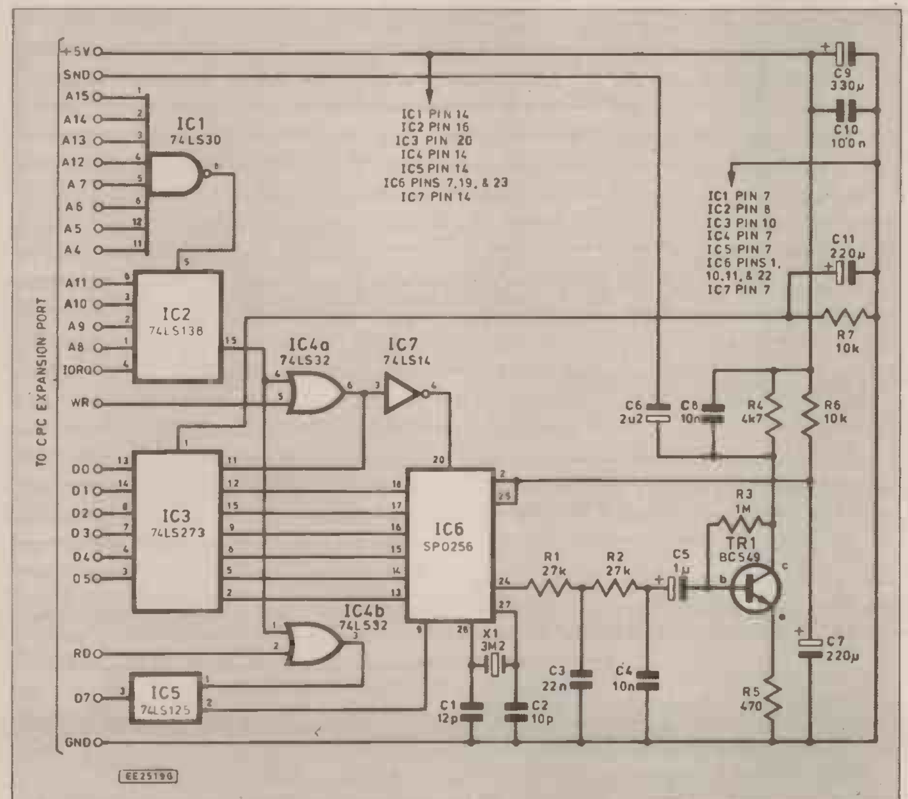


Fig. 2. Complete circuit diagram for the Amstrad CPC Speech Synthesiser.

The "LRQ" output of the SPO256 is fed to D7 of the data bus via a tristate buffer. Note that data line D6 is not required by the speech synthesiser, and that this line of the computer's expansion port is left unconnected.

DECODING

With an unexpanded CPC464 it is possible to utilize a very simple form of address and control bus decoding, which more or less consists of activating the interface when A10, IORQ, and either RD or WR are all low. With expanded or disk based CPCs the situation is a little more tricky, and much more precise decoding is required if interactions between the user add-on and other hardware (particularly the disk drivers) are to be avoided.

In this circuit all the address lines from A4 to A15 are decoded so that the speech synthesiser is placed in the input/output map at address &F8F0. In fact is appears at "echoes" from &F8F0 to &F8FF, and can be accessed by way of any address in this range.

This range of addresses is specified as being free for user add-ons in the Amstrad CPC manuals, and there should be no danger of any conflicts between the speech synthesiser and any official Amstrad add-ons or the computer's internal hardware. However, other add-ons might use the same address range, and might not be compatible with this unit.

The address/control bus decoder provides separate "read" and "write" outputs. Address &F8F0 is therefore used for sending allophone addresses to the SPO256, and for monitoring its "LRQ" output.

CIRCUIT OPERATION

Refer to Fig.2 for the full circuit diagram of the Amstrad CPC Speech Synthesiser. IC6 is the SPO256 speech chip, and this has a built-in clock oscillator. It requires a discrete crystal and two capacitors (X1, C1, and C2), and the clock frequency recommended by the speech chip manufacturer is 3.12MHz.

Crystals having this operating frequency do not seem to be very easily obtained, and I opted for a 3.2768MHz component. This is a readily available and inexpensive type intended for use in "quartz" clocks.

The increased clock frequency merely results in a slight increase in the pitch of the audio output, and a marginal increase in the speed with which words are "spoken" by the unit. Perfectly satisfactory results should be obtained using any crystal frequency from about 3MHz to 3.3MHz.

The SPO256 reqires a negative reset pulse at switch-on, and this is provided by R6 and C7. The pulse code modulated output is filtered by R1/C3 and R2/C4 to leave the decoded audio frequency signal which is fed to a common emitter amplifier based on TR1. This has a large amount of local negative feedback provided by R5, and this reduces its voltage gain to about 20dB (10 times). This gives an audio output of more than adequate strength to drive the CPC's audio input. C8 provides some additional lowpass filtering which helps to give a low ripple output signal.

ADDRESS DECODING

Address decoding is provided by IC1 and IC2. IC1 is an eight input NAND gate, and it decodes A4 to A7 and A12 to A15. The output of IC1 only goes low when all eight of these address lines are high, which is when an address which has "F" (hex) as the first and third digits is accessed.

IC2 decodes address lines A8 to A11. IC2 is a three-to-eight line decoder, but in this case only output "0" of the device is utilized. A8 is connected to its positive enable input, while A9 to A11 are coupled to its three address inputs.

Output 0 to IC2 goes low when an address having "8" (hex) as its second digit is accessed. However, the output of IC1 couples to one of the negative enable inputs of IC2, so that output 0 of IC2 only goes low when an address in the range &F8F0 to &F8FF is accessed.

This almost gives all the required decoding, but there are a couple of mild complications. One is simply that the Z80A microprocessor used in the CPC machines has separate input/output and memory maps.

The microprocessor indicates to the rest of the system that it is undertaking an input/output instruction by taking the IORQ line low. Thus, the speech synthesiser must ignore addresses in the relevant range unless IORQ is low. This is achieved by connecting IORQ to a second negative enable input of IC2.

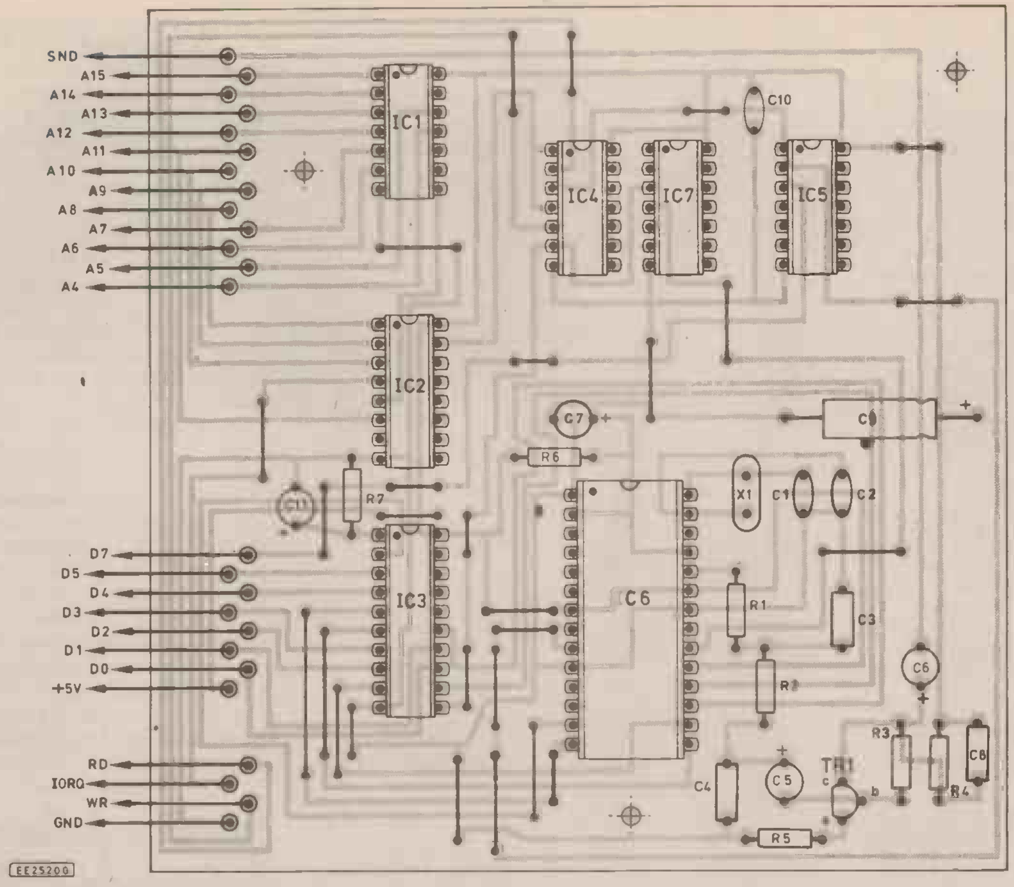


Fig. 3. (above) Printed circuit board component layout and wiring details. (below) Full size printed circuit copper foil master pattern.

The second complication is that we require separate outputs for read and write operations. This is achieved by gating the output of IC2 with the RD (read) and WR (write) lines of the CPC expansion bus, using a separate OR gate for each of these two lines. This gives a negative read pulse from IC4b, and a negative write pulse from IC4a.

A positive pulse is needed for the ALD input of the speech chip, and this is derived from the write output via inverter IC7. The six bit latching output for IC6 is provided by IC3, which is actually an octal D type flip/flop. In this case only six of the flip/flops are used, with the two "spares" just being ignored. IC4a provides a negative pulse to the "clock pulse" input of IC3 during write operations to the interface, and this latches the outputs of the flip/flops.

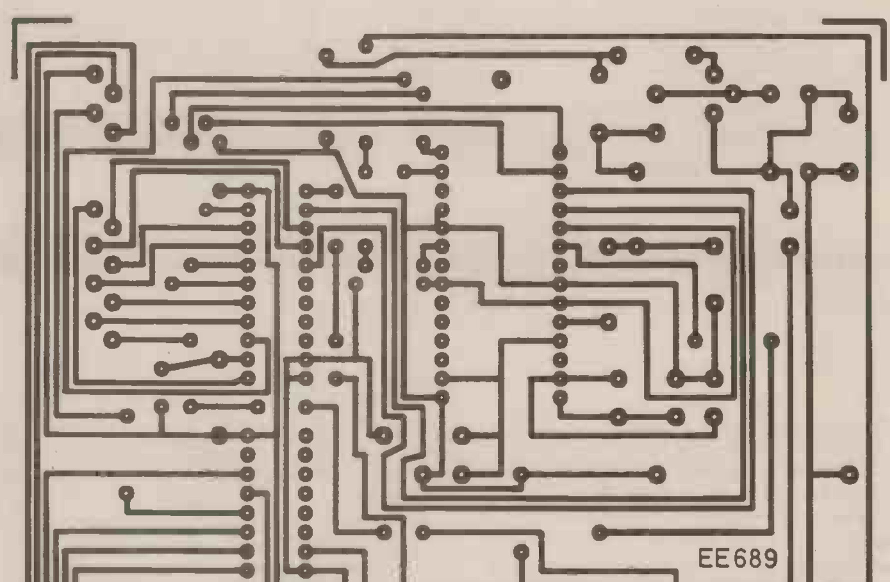
At switch-on a negative reset pulse is supplied to IC3 by C11. This ensures that IC3 commences with all its outputs low, and that the speech chip does not produce any output until the user activates it. R7 ensures that C11 is rapidly discharged when the computer is switched off, so that a fresh reset pulse is produced when it is switched on again.

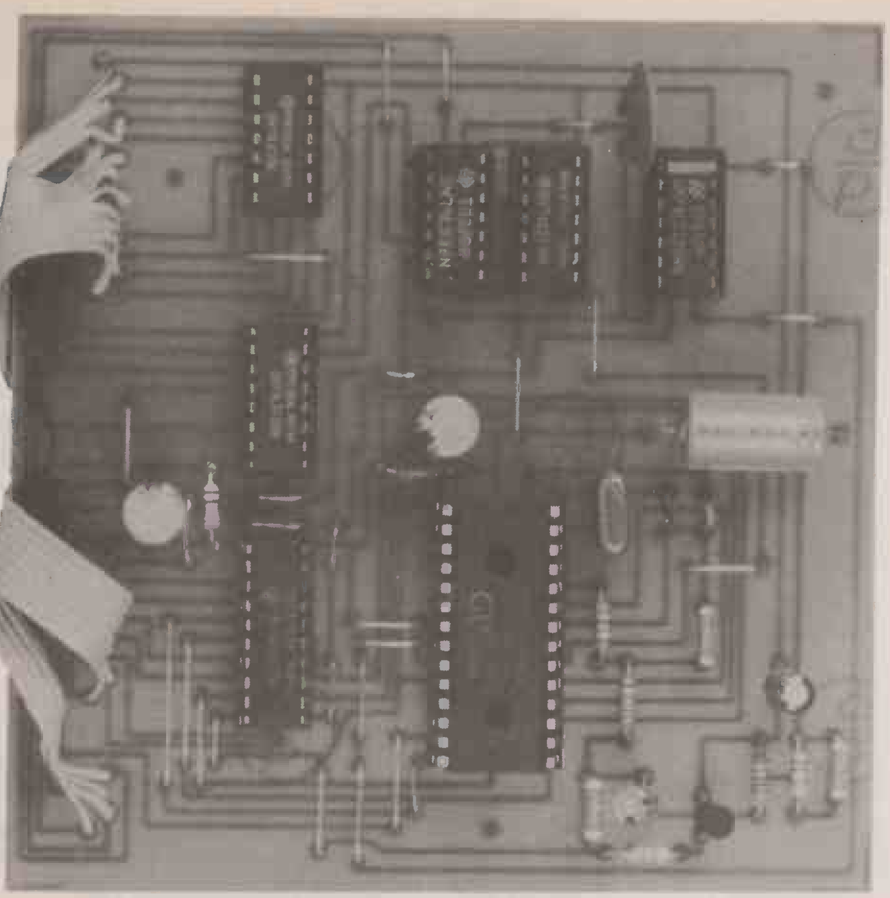
IC5 is a quad tri-state buffer, but in this circuit only one buffer is used and no connections are made to the other three. It interfaces the LRQ output of IC6 to D7 of the CPC expansion bus, and it is set to the active state by the output pulses from IC4b during read operations to the speech synthesiser.

Power for the speech synthesiser is obtained from the +5 volt supply output of the CPC expansion port. C9 provides smoothing which helps to give a low noise level on the audio output of the unit.

CONSTRUCTION

Details of the printed circuit board are provided in Fig.3. Crystal X1 must be a miniature wire-ended type with 0.2 inch lead spacing if it is to fit easily onto the board. Try to complete the soldered joints reasonably rapidy when fitting this component as some crystals are relatively easily





Completed circuit board showing all the i.c.s. mounted in i.c. holders.

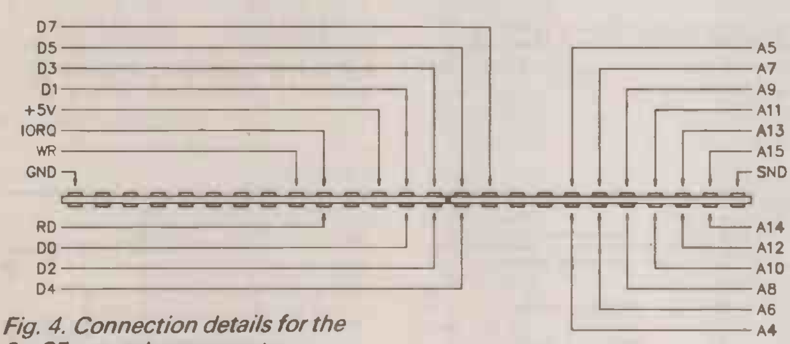


Fig. 4. Connection details for the 2 x 25 way edge connector.

damaged by heat. IC6 is a MOS integrated circuit, and as such it is vulnerable to damage by static charges.

Leave this component in its anti-static packaging until, in all other respects, the unit is finished. Then fit this component onto the board, handling it as little as possible, avoiding touching the pins and fitting it into a holder already fitted to the board so that there is no need to make any direct soldered connections to this chip.

With a fairly large integrated circuit of this type, IC6 can be a bit difficult to fit a into its socket. Usually the pins have to be carefully bent inwards slightly. Be careful not to buckle any of its pins when fitting IC6 into its socket. Although none of the other integrated circuits are static sensitive types, it is recommended that they should be fitted in holders.

There should be no difficulties in fitting the other components onto the board provided the correct types of capacitor are used. In particular, the polyester types should have a lead spacing of 7.5 millimetres (0.3 inches). Be careful to fit the electrolytic capacitors with the correct polarity. A number of link wires are required, and these can be made from 22s.w.g. tinned copper wire, or the leads trimmed from the resistors and capacitors might be sufficient.

CONNECTIONS

The board is connected to the computer via a 25-way ribbon cable which is fitted with a 2 by 25-way 0.1 inch pitch edge connector at the computer end. At the other end it is either connected directly to the board, or via pins if preferred.

Both ends of the cable should be prepared by separating all the wires, stripping a few millimetres of insulation from each one, and tinning them all with solder. The terminals of the edge connector (or the twenty five that are used anyway) should be tinned with solder, as should the pins on the board if this method of connection is adopted.

In my experience the CPC computers are not tolerant of long connecting leads on their expansion bus. It is therefore recommended that the connecting cable should be no more than about 0.6 metres long.

The edge connector connection details are given in Fig.4. The board has been designed so that the order of the connections on the board matches up well with that of those on the edge connector. There is no need for any crossing over of wires, but you still need to be very careful to get each lead connected to the right terminal of the edge connector.

The multi-coloured "rainbow" ribbon cable is better for this sort of thing, but grey ribbon cable is usable if you take extra care. If you are using a connector fitted with the appropriate polarising key, make quite sure that you have the connector the right way up before making the connectio ns to it.

If the connector is not fitted with a polarising key, clearly mark the top and bottom edges such as to minimise the risk of fitting it to the expansion port up-side-down. Once this wiring has been completed, check it very thoroughly at least once before connecting the unit to the computer and trying it out.

Projects of this type are often left as uncased boards, as was the prototype. However, it should not be difficult to fit the unit into a small plastic case if preferred. This does have the advantages of being a bit neater, and keeping dust etc. off the circuit board.

COMPONENTS

Shop Talk
See page 319

Resistors

R1, R2	27k (2 off)
R3	1M
R4	4k7
R5	470
R6, R7	10k (2 off)

All 0.25W 5% carbon

Capacitors

C1	12p ceramic plate
C2	10p ceramic plate
C3	22n polyester (7.5mm pitch)
C4, C8	10n polyester (7.5mm pitch) (2 off)
C5	1μ radial elect. 63V
C6	2μ2 radial elect. 63V
C7, C11	220μ radial elect. 10V (2 off)
C9	330μ axial elect. 10V
C10	100n disc ceramic

Semiconductors

IC1	74LS30 8-input NAND gate
IC2	74LS138 3-to-8 line decoder
IC3	74LS273 octal flip/flop
IC4	74LS32 quad 2-input OR gate
IC5	74LS125 quad tristate buffer
IC6	SP0256 speech synthesiser
IC7	74LS14 hex inverting trigger
TR1	BC549 silicon *npn*

Miscellaneous

X1	3.2768MHz miniature wire-ended crystal (see text)

Printed circuit board available from the *EE PCB Service*, order code EE689; case (see text); 2 x 25 way 0.1 inch pitch edge connector; 25 way ribbon cable; 14 pin d.i.l. i.c. holder (4 off); 16 pin d.i.l. i.c. holder; 20 pin d.i.l. i.c. holder; 28 pin d.i.l. i.c. holder; connecting wire; etc.

Approx cost. Guidance only **£20**

IN USE

With any add-on project that connects to a computer's buses it is essential to connect it to the computer *prior* to switch-on. Otherwise the computer is likely to crash when the add-on is connected, and the computer and (or) add-on could be damaged. Once switched on, the computer should go through its start-up routine in the normal way. Switch off at once and recheck all the wiring if there is any hint of anything out of the ordinary at switch-on.

To test the unit you can try this example program, which should result in the speech unit saying a couple of words that will be familiar to all readers!

```
5 REM SPEECH ROUTINE
10 READ a
20 OUT &F8F0,a
30 IF a = 64 THEN END
40 WHILE INP(&F8F0) > 127
50 WEND
60 GOTO 10
70 DATA 7, 7, 35, 52, 19, 0, 33, 20, 3, 19,
   45, 7, 42, 17, 39, 24, 11, 12, 41, 55, 64
```

Line 10 reads data from line 70, and at line 20 each read value is sent to the speech synthesiser. Line 30 is used to detect the end of the speech data, and terminate the routine.

This is achieved by using a number of 64 as the last allophone value. This value is sent to the speech unit, but as it does not respond to the two most significant bits, the value it is sent is effectively zero. This is important, because most of the allophones do not cut off after the appropriate length of time, but will continue indefinitely if allowed to do so. Using 0 as the last allophone address finishes the sequence of allophones with a short pause, and ensures that there is then silence from the speech unit until it is activated again.

A WHILE...WEND loop at lines 40 and 50 halts the progress of the program until the current allophone has been completed. The test used is whether or not the value returned from the speech unit is greater than 127. It must be more than 127 while an allophone is being produced, since D7 will be taken high by the status output of the SPO256, giving a returned value of at least 128. However, once an allophone has been completed, and D7 is taken low, the returned value can be no more than 127. Logic ANDing could be used to mask D0 to D6, but the method used in the demonstration program is probably faster and easier.

The allophone values used in the example program are the obvious ones, and they give reasonable results. In some cases the obvious allophones give something far removed from the desired result, but a little experimentation will often resolve matters.

Sometimes what may seem like a sound that requires just one allophone needs to be put together from two or even three allophones. The accompanying list of allophones, addresses (in hex and decimal), plus example words to clarify the sounds they produce, should aid the selection of likely allophones for your own phrases.

TABLE. 1: List of Allophones, their addresses and sample words

Dec Address	Hex Address	Allo-phone	Example Word	Dec Address	Hex Address	Allo-phone	Example Word
0	0	PA1*	10ms pause	32	20	AW	OUr
1	1	PA2*	30ms pause	33	21	DD2	Down
2	2	PA3*	50ms pause	34	22	GG3	fIG
3	3	PA4*	100ms pause	35	23	VV	Vertical
4	4	PA5*	200ms pause	36	24	EG1	GUest
5	5	OY	bOY	37	25	SH	SHone
6	6	AY	frY	38	26	ZH	aZure
7	7	EH*	End	39	27	RR2	tRain
8	8	KK3	Come	40	28	FF*	Find
9	9	PP	Poster	41	29	KK2	sKy
10	A	JH	dodGe	42	2A	KK1	Can
11	B	NN1	tiN	43	2B	ZZ	Zoo
12	C	IH*	pIt	44	2C	NG	bANk
13	D	TT2	Top	45	2D	LL	Love
14	E	RR1	Rattle	46	2E	WW	Wood
15	F	AX*	sUcceed	47	2F	XR	Rain
16	10	MM	Mint	48	30	WH	WHere
17	11	TT1	porT	49	31	YY1	Yell
18	12	DH1	THem	50	32	CH	CHoose
19	13	IY	sEA	51	33	ER1	fIR
20	14	EY	cAge	52	34	ER2	buRR
21	15	DD1	shoulD	53	35	OW	bOW
22	16	UW1	tO	54	36	DH2	THey
23	17	AO*	tAUght	55	37	SS*	veSt
24	18	AA*	lOt	56	38	NN2	Now
25	19	YY2	YEs	57	39	HH2	How
26	1A	AE*	tAp	58	3A	OR	sORe
27	1B	HH1	Hat	59	3B	AR	alARm
28	1C	BB	BUsy	60	3C	YR	reaR
29	1D	TH*	THen	61	3D	EG2	Got
30	1E	UH*	bOOk	62	3E	EL	saddLE
31	1F	UW2	bOOt	63	3F	BB2	Bottle

Where allophones are marked with a "*" it is possible to use the same one two or more times in a row in order to give an elongated version of the sound.

ALLOPHONE ALTERATION AND USE

Note that in most cases you cannot produce elongated versions of allophones simply by using the same one two or three times in succession. In most cases this just gives a sort of echo effect. It is possible with some though, as indicated in the list.

With a few of these it is acceptable to duplicate them at the beginning of a word, but not if they are used at the end or in the middle. However, I am only giving guidelines rather than rigid rules here. It does no harm to experiment a little, and the invalid use of allophones sometimes gives the best results.

In a number of instances there are what seems to be duplications of allophones. This is where the lengths of the sounds and the emphasis is different. In general, it is the longer and "harder" versions of sounds that are used at the beginnings of words, with shorter and "milder" versions being used at the end or in the middle.

The example words indicate the normal position in a word that each allophone should occupy, although I must again emphasise that there are no hard and fast rules here. Where there are alternative sounds, it is worthwhile trying them all to determine which one sounds best.

Some of the shorter and "sharper" sounds work best if they are preceded by silence. As many of these are often used at the beginnings of words, this preceding silence will be provided anyway. However, where necessary a strategically placed brief pause (allophone 0 or 1) will improve results. The sounds which benefit from this are the B, D, and G ones (which only need short pauses), and the P, T, and K ones (which require longer pauses, possibly even allophone 2 instead of 1).

Pauses in general tend to be problematical. There is a natural tendency to assume that there are short pauses between words, longer ones between sentences, and no pauses anywhere else. In fact there are often few pauses between words, and some in between syllables of the words that do not easily trip of the tongue. When selecting pauses, or any allophones come to that, you should think in terms of what the phrase sounds like, rather than how it appears in written form. □