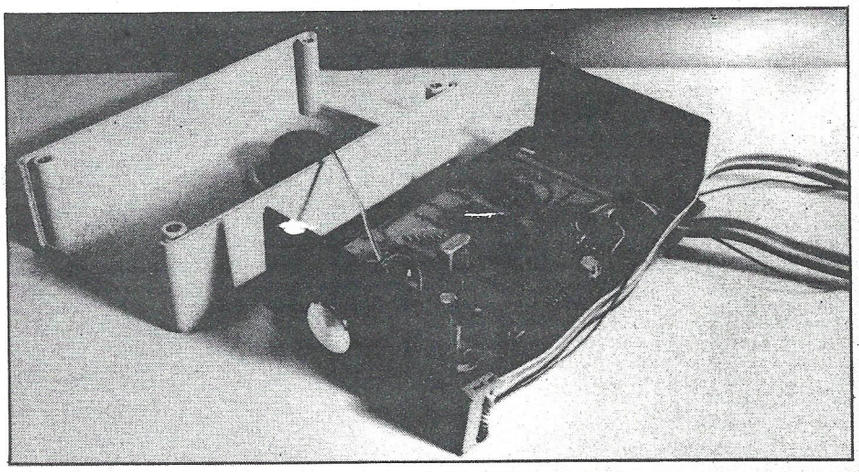
# Speech synthesiser for BBC micro

## And now a speech synthesiser for the BBC micro. Mark Stewart builds a low cost board with the same SPO256-AL2 allophone chip used in last month's project for the Spectrum, but with the added extra of EPROM storage.



Following on from the discussion of speech synthesis in the November and December '83 issues of *E&CM,* this article presents a practical speech synthesiser for use with the BBC Model B. Using the General Instruments allophone IC, the SPO256-AL2, this circuit allows two different modes of operation. In "direct" mode the computer controls the speech synthesiser IC directly, allowing total software control of individual allophones, which may be strung together to produce the required speech. In the second mode the computer selects a word or phrase from an on-board EPROM; 64 phrases, each up to 32 allophones long can be stored in a single 2716 2K EPROM. Alternative connections allow the format to be changed so that 32 phrases of 64 allophones, or 16 phrases of 128 allophones can be used.

It is also possible to use a 2732 4K EPROM and select between 2 banks of 2K using an on board switch. Switching between the two modes is done in software, so that a program can mix "standard phrases" from the EPROM with direct, software controlled words.

A particular advantage of the EPROM mode is that the computer is freed once it has initiated the phrase required. Thus fast action games can be combined with speech, using simple programming, without loss of speed. It is of course, necessary to construct a speech EPROM vocabulary for a particular set of programs. The "direct mode" allows the words or phrases to be set up experimentally in RAM before committing them to EPROM, using an EPROM programmer (but of course, the system can be used without an EPROM).

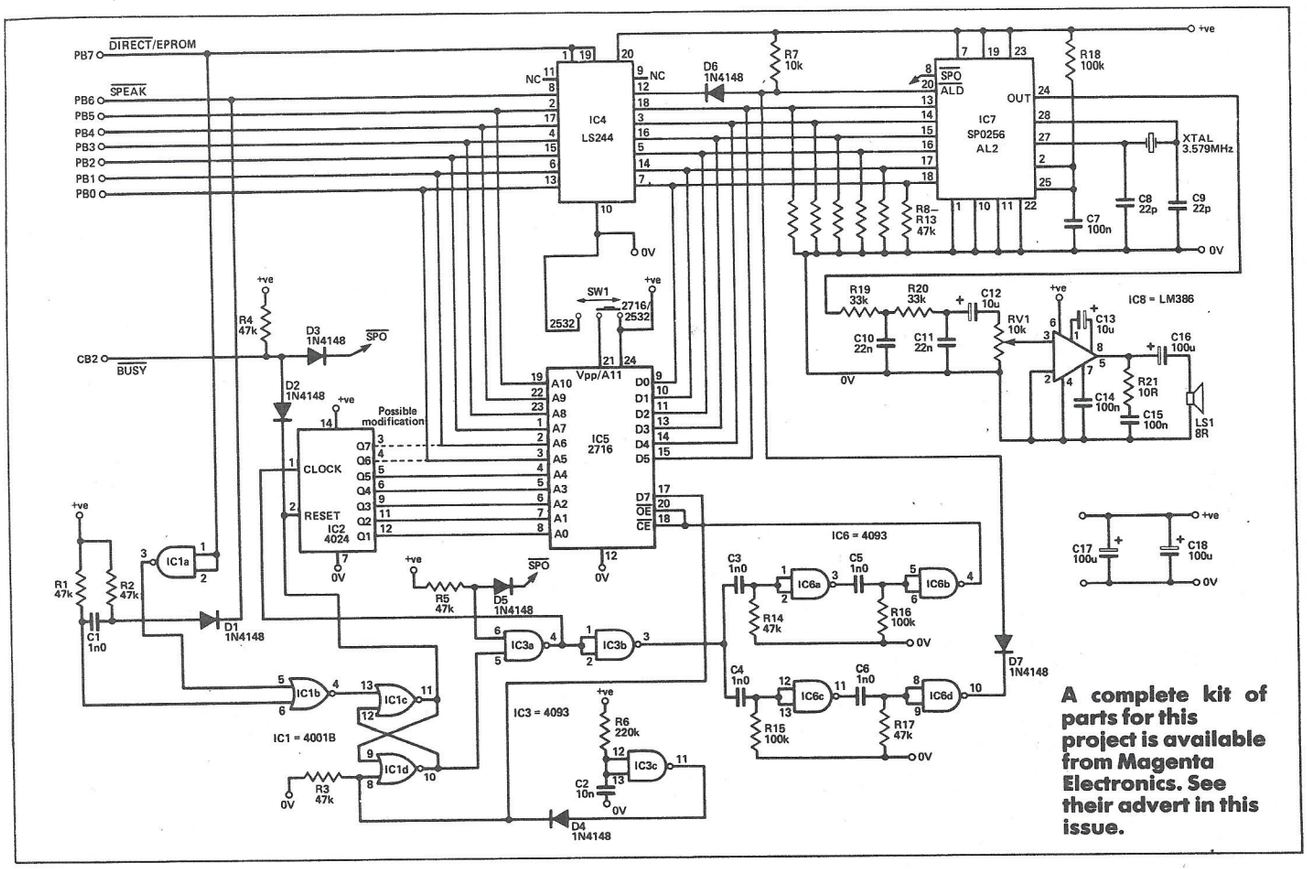
## The Circuit

A full circuit diagram is shown in **Figure 1.** The speech synthesiser IC7 has 6 data input lines which address the 64 allophones listed in **Figure 4.** The direct mode is selected by a logic low on the computer user port line, PB7. Tri state buffer, IC4, is enabled and the 6 data lines of IC7 are connected to PB0-5 of the computer user port. User port line PB6 is IC7. A signal from the active low output (SPO) of IC7 is coupled to the CB2 control line of the computer port. This line is pulled low by IC7 when it is busy. A rising edge indicates to the computer that the next allophone data can be loaded.

Selection of direct mode automatically disables the EPROM. Its outputs assume a high impedance state and are effectively disconnected from the system.

When PB7 is high EPROM speech is enabled. IC4 output lines are all set to their high impedance state. IC1a inverts the signal from PB7 and sets one input of IC1b low. The other input of NOR gate IC1b is fed from the SPEAK line PB6 via D1 and the pulse forming network R1, R2 and C1. A high to low transition of PB6 produces a short positive pulse at the output of IC1b. This pulse sets the bistable formed by cross coupled gates IC1c and d. The output of IC1c changes from high to low. This removes the reset condition from the 7 bit binary counter IC2, and pulls down the BUSY line via D2. Setting the bistable also changes the output of IC1d from low to high. The output from IC1d puts a high state on one input of IC3a, the other input of which is held high by the inactive SPO output of IC7. The output of IC3a therefore changes from high to low, and IC2 is clocked 1 count from zero. The output from IC3a is inverted by IC3b and coupled to the two delayed pulse forming circuits IC6a, b and IC6c, d. The low to high transition from IC3b is coupled, via C3, to the inputs of IC6a. The inputs of IC6a are pulled high via C3. C3 then begins to charge via R14, and the inputs are pulled back down again to zero, even though the output of IC3b may still be high. The result of this is that a short negative pulse is produced on the output of IC6a, starting when IC3b output changes from low to high. The length of the pulse is set by the values of C3 and R14.

IC6b is an identical stage which produces a negative pulse which starts on the rising edge at the end of the negative pulse from IC6a. A similar pattern occurs with IC6c and IC6d, but the relative pulse lengths differ because different value timing resistors are used. **Figure 2** shows these pulses graphically. The output pulse from IC6b enables the EPROM. The falling edge of the pulse on the output of IC6d is timed to occur during the time that the EPROM is enabled. This pulse is used by IC7 which loads the data present on the EPROM data lines, and starts to execute the first allophone routine. As soon as IC7 loads the data the 'busy' condition is signalled by the SPO output changing from high to low. This signal is coupled via D5 to IC3a. The inputs to NAND gate, IC3a, change from both being high to one being high and one being low. Its output changes from low to high. This change has no effect on the counter, which is clocked on falling edges. The change is inverted by IC3b, but has no effect on IC6 which is triggered only by rising edges.




**A complete kit of parts for this project is available from Magenta Electronics. See their advert in this issue.**

*Figure 1. Circuit diagram of the synthesiser.*

The circuit waits in a stable state until IC7 has completed its routine. At the end of the routine the SPO output changes from low to high, IC3a output also falls and clocks IC2 and IC6 repeats its pulse sequence. IC7 again loads the EPROM address, which is derived from IC2, and has increased by 1. IC7 therefore executes the allophones from the EPROM one by one. With IC2 and IC5 connected as shown, the lower 5 address lines would cycle indefinitely without some means of ending the sequence. To accomplish this, one of the two higher order data lines of the EPROM is used. The 64 available allophones require only 6 data lines from the 8 bit EPROM. The higher two lines are normally set to 0. By setting one of the high order lines to logic 1, it is simple to indicate the end of a sequence. EPROM data line D7 is connected ot IC1d. The control bistable is reset whenever a logic 1 appears on D7, IC2 is reset, and the BUSY signal returns from low to high. The circuit is then returned to its initial state.

**"... allows the computer to select words or phrases stored in EPROM".**

A 'power on reset' pulse is produced by IC3c to ensure that the control bistable is initially in the correct state.

IC7 has its own clock oscillator which uses C8, C9 and crystal X1. A 3.12MHz crystal is specified by the manufacturers, but a cheaper 3.57MHz USA colour TV crystal operates well. The speech output from IC7 passes through a simple low pass filter network and volume control, to a standard IC audio amplifier. A miniature speaker provides the final link.

## Construction

Construction is very straightforward. A double sided printed circuit board is used, see **Figures 3a** and **3b;** the only off board components are the volume control RV1 and the loudspeaker.

Follow the overlay drawing to be published next month when assembling the board. Note that through connections are made using PCB pins, and that some component leads are soldered on both sides of the board. IC sockets are recommended but are not essential except for the EPROM.

Connections to the speaker, RV1, and the ribbon cable from the computer are best made using PCB pins.

Although only 11 connections are made to the computer it is better to use a length of 20-way ribbon cable, and to cut back the unused leads at the synthesiser end.

## Programming

Direct mode allows the computer to operate the speech synthesiser IC directly. The mode is selected by setting the user port line PB7 to a logic 0. The user port must first be set up as an output port by using the instruction:

?&FE62=FF

PB7 is set low by any number less than 127. The production of an allophone is initiated by a high to low transition of PB6. The other 6 port lines select the required allophone. The allophone table, **Figure 4,**

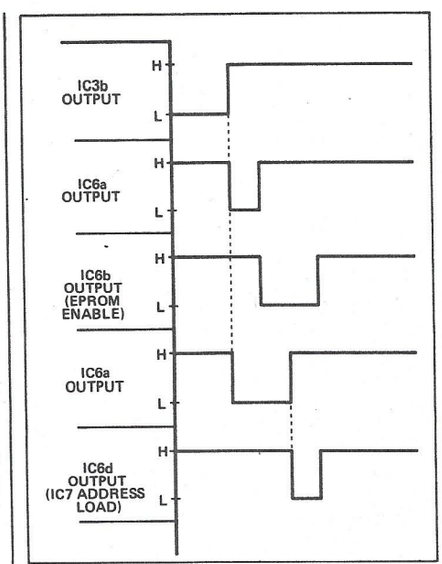



*Figure 2. The relationship between the various timing pulses.*

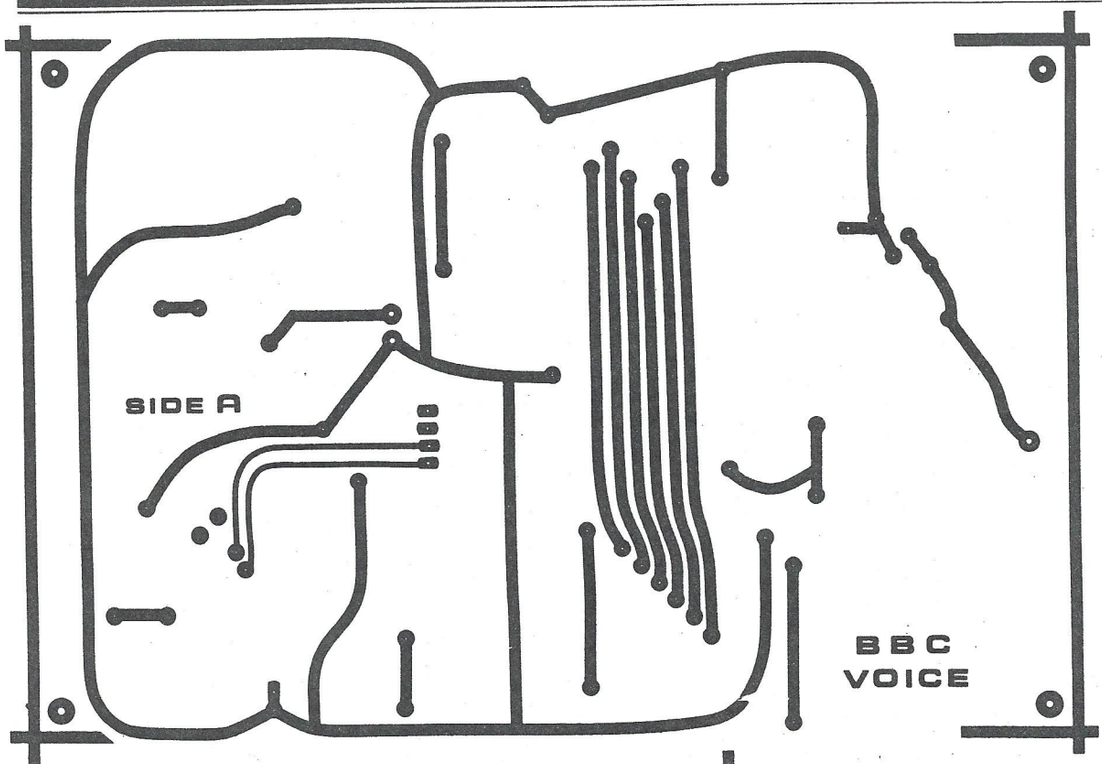



*Figure 3a. Top side foil of the double sided board.*

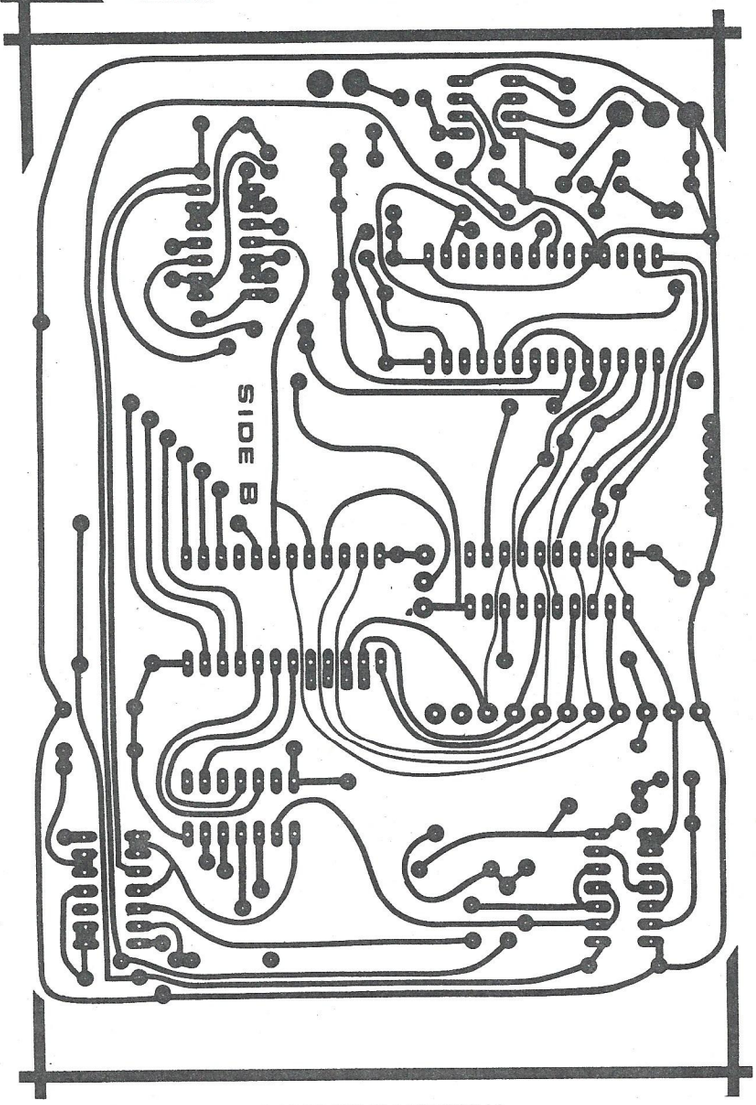



*Figure 3b. Lower foil pattern.*

lists the 64 sounds and pauses that can be produced.

The following short program allows each allophone to be selected and sounded:

```
10 ?&FE60=64
15 INPUT A
20 ?&FE60=A
30 GOTO 10
```

Line 10 sets all the port lines except PB6 to 0. The decimal number of the required allophone is entered in line 15. In line 20 the allophone number is put on port lines PB0-PB5, and PB6 is changed from high to low. The falling edge on PB6 tells IC7 to latch the allophone data into its register, and commence execution. There is a problem with this simple program because IC7 continues to produce the allophone until a new one is entered. A pause (data 0-4) must always be entered to finish a phrase or word.

It is necessary for the computer to know when one allophone has ended, so that the next one can be loaded inot IC7 without leaving an audible pause. This timing function is made possible by the SPO output of IC7 which goes low whilst each allophone is sounded. A low to high transition on the SPO line occurs at the end of each allophone. This BUSY signal is coupled to the computer port control line CB2.

Those familiar with the 6522 computer VIA chip will be aware of its many complexities. Without going into detail the relevant operation can be summarised as follows. Control line CB2 influences data Bit 3 of a register in the 6522 known as the interrupt flag register (IFR). The register can be read by the computer at FE6D Hex. Normally the register contains all '0s', so a simple test can be used to check on Bit 3.

The effect of CB2 on IFR Bit 3 is controlled by another 6522 register at FE6C called the peripheral control register (PCR). By setting Bit 6 of the PCR to a logic 1, a low to high transition on CB2 will set Bit 3 in the IFR to 1. Three of the bits in the PCR are already set (Bits 1, 2 and 3) and this must be allowed for when setting Bit 6.

When the computer sends an output to the user port PB lines Bit 3 in the IFR is automatically set to 0. After IC7 has executed the appropriate allophone, the BUSY line switches from logic 0 to logic 1 so instructing the computer to issue the next instruction.

The simple BASIC program that follows should clarify the method of operation:

```
 4 ?&FE62=&FF          Sets up the output port
 8 ?&FE6C=78           Sets up the PCR
10 ?&FE60=64           Sets PB6 high
20 READ A              Get allophone data
30 IF A=64 THEN GOTO 120   Tests for end condition
40 ?&FE60=A            Loads data into IC7
50 B=?&FE6D            Reads IFR
60 IF B=0 THEN GOTO 50 Loop while allophone is executed
70 GOTO 10             Repeat for next allophone
80 DATA 20,29,36,47...etc..64   List of allophone codes separated by commas
                                ending with 64
120 STOP
```

The program loops around lines 50 and 60 whilst the BUSY line is low. When the BUSY line is switched to high, B is no longer 0 so the program proceeds to line 70, and returns for the next.

## EPROM Programming

The EPROM mode of operation is set by a logic 1 on PB7 of the computer port. Speech is initiated by a high to low transition of PB6. The other 6 data lines determine which bank of 32 EPROM addresses is used. If PB 0-5 are set to zero the EPROM is addressed from 01 to 32. If PB0 is set to 1 the 32 EPROM addresses start at 33 and end at 64, etc.

Note that the sequence can be made shorter than 32 bytes by putting an end character in the EPROM (any number between decimal 128 and 255, or 80-FF Hex).

All of this translates into BBC code as follows:

First enter: ?&FE62=FF

This sets the user port lines PB 0-7 to output mode.

Then enter the following short program:

```
 5 INPUT A
10 ?&FE62=192+A
20 ?&FE60=128+A
30 GOTO 5
```

The value of 'A' determines the bank of addresses to be used, and can take a value from 0 to 63.

Adding 192 to A in line 10 sets PB6 and 7 high, and so sets up EPROM mode. Line 20 leaves the value of PB7 high whilst changing PB6 from high to low. The falling edge of PB6 initiates the speech cycle. Note that if an end character is not found in the EPROM the circuit will latch, and continuously cycle through the 32 addresses. Switching the power off and on will restore normal operation.

Alternative phrase lengths can be set by changing the hardware connections. If the Q6 output of IC2 is connected to A5 of IC5, and the computer PB0 line disconnected, there will be 32 banks of 64 addresses available. Linking Q7 of IC2 to A6 of IC6 as well, will give 16 banks of 128 addresses.

To use the speech in a program it is only necessary to run lines 10 and 20 with the required value of 'A' added to the 192 and 128. Remember though to initiate the port by entering ?&FE62=FF somewhere in the program, before the speech part.

Mixing EPROM speech with direct speech can be achieved simply by switching backwards and forwards using software. The busy condition may be read from the IFR in exactly the same way for both the direct and EPROM modes. ■

## PARTS LIST

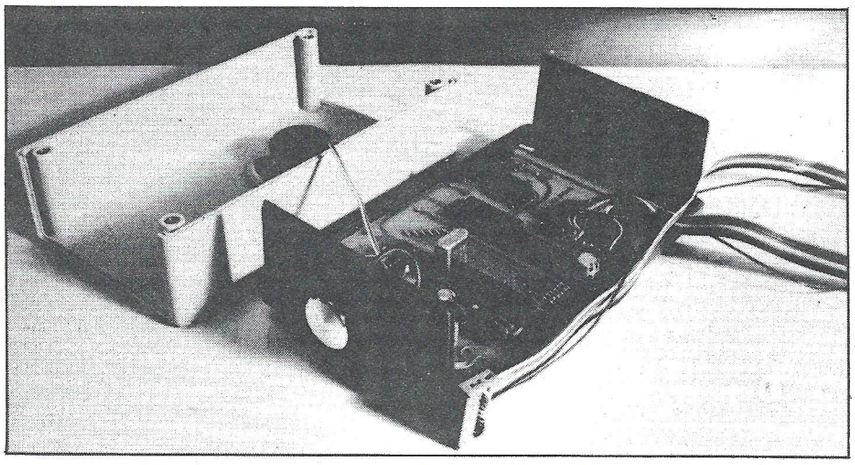
| | |
|---|---|
| **Resistors (¼Watt 5% Carbon Film)** | |
| R1,2,3,4,5,8,9,10,11,12,13,14,17 | 47K |
| R6 | 220K |
| R7 | 10K |
| R15,16,18 | 100K |
| R19,20 | 33K |
| R21 | 10R |
| RV1 | 10K log potentiometer |
| **Capacitors** | |
| C1,3,4,5,6 | 1nF ceramic plate |
| C2 | 10nF C280 |
| C7,14,15 | 100nF |
| C8,9 | 22pF ceramic plate |
| C10,11 | 22nF C280 |
| C12,13 | 10uF 10V radial electrolytic |
| C16,17,18 | 100uF 10V radial electrolytic |
| **Semiconductors** | |
| D1-7 | IN4148 |
| IC1 | 4001B |
| IC2 | 4024B |
| IC3,6 | 4093B |
| IC4 | 74LS244 |
| IC5 optional | 2716/2732 EPROM |
| IC7 | SPO256-AL2 |
| IC8 | LM386 |
| **Miscellaneous** | |
| Crystal - X1 3.579MHz or 3.12MHz; Speaker - Miniature 8 ohm; 11 way ribbon cable; IC sockets; Socket for micro port; Knob; Case; Fixings; PCB etc. | |

*Figure 4. The 64 allophones generated by the synthesiser.*

| ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD |
|---|---|---|---|---|---|---|---|---|
| 0 | 10ms PAUSE | | 22 | UW1 | tO | 44 | NG | aNchore |
| 1 | 30ms PAUSE | | 23 | A0 | AUght | 45 | LL | Lake |
| 2 | 50ms PAUSE | | 24 | AA | hOt | 46 | WW | Wool |
| 3 | 100ms PAUSE | | 25 | YY2 | Yes | 47 | XR | repaIR |
| 4 | 200ms PAUSE | | 26 | AE | hAt | 48 | WH | WHile |
| 5 | OY | bOY | 27 | HH1 | He | 49 | YY1 | Yes |
| 6 | AY | skY | 28 | BB1 | daB | 50 | CH | CHurch |
| 7 | EH | End | 29 | TH | THin | 51 | ER1 | summER |
| 8 | KK3 | Comb | 30 | UH | bOOk | 52 | ER2 | bURn |
| 9 | PP | Pit | 31 | UW2 | fOOd | 53 | OW | nOW |
| 10 | JH | dodGe | 32 | AW | OUt | 54 | DH2 | THey |
| 11 | NN1 | thiN | 33 | DD2 | Do | 55 | SS | veST |
| 12 | IH | sIt | 34 | GG3 | wiG | 56 | NN2 | No |
| 13 | TT2 | To | 35 | VV | Vest | 57 | HH2 | Noe |
| 14 | RR1 | Rural | 36 | EF1 | GUest | 58 | OR | stORe |
| 15 | AX | sUcceed | 37 | SH | SHip | 59 | AR | alArm |
| 16 | MM | Milk | 38 | ZH | aZure | 60 | YR | clearR |
| 17 | TT1 | parT | 39 | RR2 | bRain | 61 | EG2 | Got |
| 18 | DH1 | THey | 40 | FF | Food | 62 | EL | saddle |
| 19 | IY | sEE | 41 | KK2 | sKy | 63 | BB2 | Business |
| 20 | EY | bEIge | 42 | KK1 | Can't | | | |
| 21 | DD1 | coulD | 43 | ZZ | Zoo | | | |

# Speech synthesiser for BBC micro

## And now a speech synthesiser for the BBC micro. Mark Stewart builds a low cost board with the same SPO256-AL2 allophone chip used in last month's project for the Spectrum, but with the added extra of EPROM storage.



Following on from the discussion of speech synthesis in the November and December '83 issues of *E&CM,* this article presents a practical speech synthesiser for use with the BBC Model B. Using the General Instruments allophone IC, the SPO256-AL2, this circuit allows two different modes of operation. In "direct" mode the computer controls the speech synthesiser IC directly, allowing total software control of individual allophones, which may be strung together to produce the required speech. In the second mode the computer selects a word or phrase from an on-board EPROM; 64 phrases, each up to 32 allophones long can be stored in a single 2716 2K EPROM. Alternative connections allow the format to be changed so that 32 phrases of 64 allophones, or 16 phrases of 128 allophones can be used.

It is also possible to use a 2732 4K EPROM and select between 2 banks of 2K using an on board switch. Switching between the two modes is done in software, so that a program can mix "standard phrases" from the EPROM with direct, software controlled words.

A particular advantage of the EPROM mode is that the computer is freed once it has initiated the phrase required. Thus fast action games can be combined with speech, using simple programming, without loss of speed. It is of course, necessary to construct a speech EPROM vocabulary for a particular set of programs. The "direct mode" allows the words or phrases to be set up experimentally in RAM before committing them to EPROM, using an EPROM programmer (but of course, the system can be used without an EPROM).

## The Circuit

A full circuit diagram is shown in **Figure 1.** The speech synthesiser IC7 has 6 data input lines which address the 64 allophones listed in **Figure 4.** The direct mode is selected by a logic low on the computer user port line, PB7. Tri state buffer, IC4, is enabled and the 6 data lines of IC7 are connected to PB0-5 of the computer user port. User port line PB6 is IC7. A signal from the active low output (SPO) of IC7 is coupled to the CB2 control line of the computer port. This line is pulled low by IC7 when it is busy. A rising edge indicates to the computer that the next allophone data can be loaded.

Selection of direct mode automatically disables the EPROM. Its outputs assume a high impedance state and are effectively disconnected from the system.

When PB7 is high EPROM speech is enabled. IC4 output lines are all set to their high impedance state. IC1a inverts the signal from PB7 and sets one input of IC1b low. The other input of NOR gate IC1b is fed from the SPEAK line PB6 via D1 and the pulse forming network R1, R2 and C1. A high to low transition of PB6 produces a short positive pulse at the output of IC1b. This pulse sets the bistable formed by cross coupled gates IC1c and d. The output of IC1c changes from high to low. This removes the reset condition from the 7 bit binary counter IC2, and pulls down the BUSY line via D2. Setting the bistable also changes the output of IC1d from low to high. The output from IC1d puts a high state on one input of IC3a, the other input of which is held high by the inactive SPO output of IC7. The output of IC3a therefore changes from high to low, and IC2 is clocked 1 count from zero. The output from IC3a is inverted by IC3b and coupled to the two delayed pulse forming circuits IC6a, b and IC6c, d. The low to high transition from IC3b is coupled, via C3, to the inputs of IC6a. The inputs of IC6a are pulled high via C3. C3 then begins to charge via R14, and the inputs are pulled back down again to zero, even though the output of IC3b may still be high. The result of this is that a short negative pulse is produced on the output of IC6a, starting when IC3b output changes from low to high. The length of the pulse is set by the values of C3 and R14.

IC6b is an identical stage which produces a negative pulse which starts on the rising edge at the end of the negative pulse from IC6a. A similar pattern occurs with IC6c and IC6d, but the relative pulse lengths differ because different value timing resistors are used. **Figure 2** shows these pulses graphically. The output pulse from IC6b enables the EPROM. The falling edge of the pulse on the output of IC6d is timed to occur during the time that the EPROM is enabled. This pulse is used by IC7 which loads the data present on the EPROM data lines, and starts to execute the first allophone routine. As soon as IC7 loads the data the 'busy' condition is signalled by the SPO output changing from high to low. This signal is coupled via D5 to IC3a. The inputs to NAND gate, IC3a, change from both being high to one being high and one being low. Its output changes from low to high. This change has no effect on the counter, which is clocked on falling edges. The change is inverted by IC3b, but has no effect on IC6 which is triggered only by rising edges.

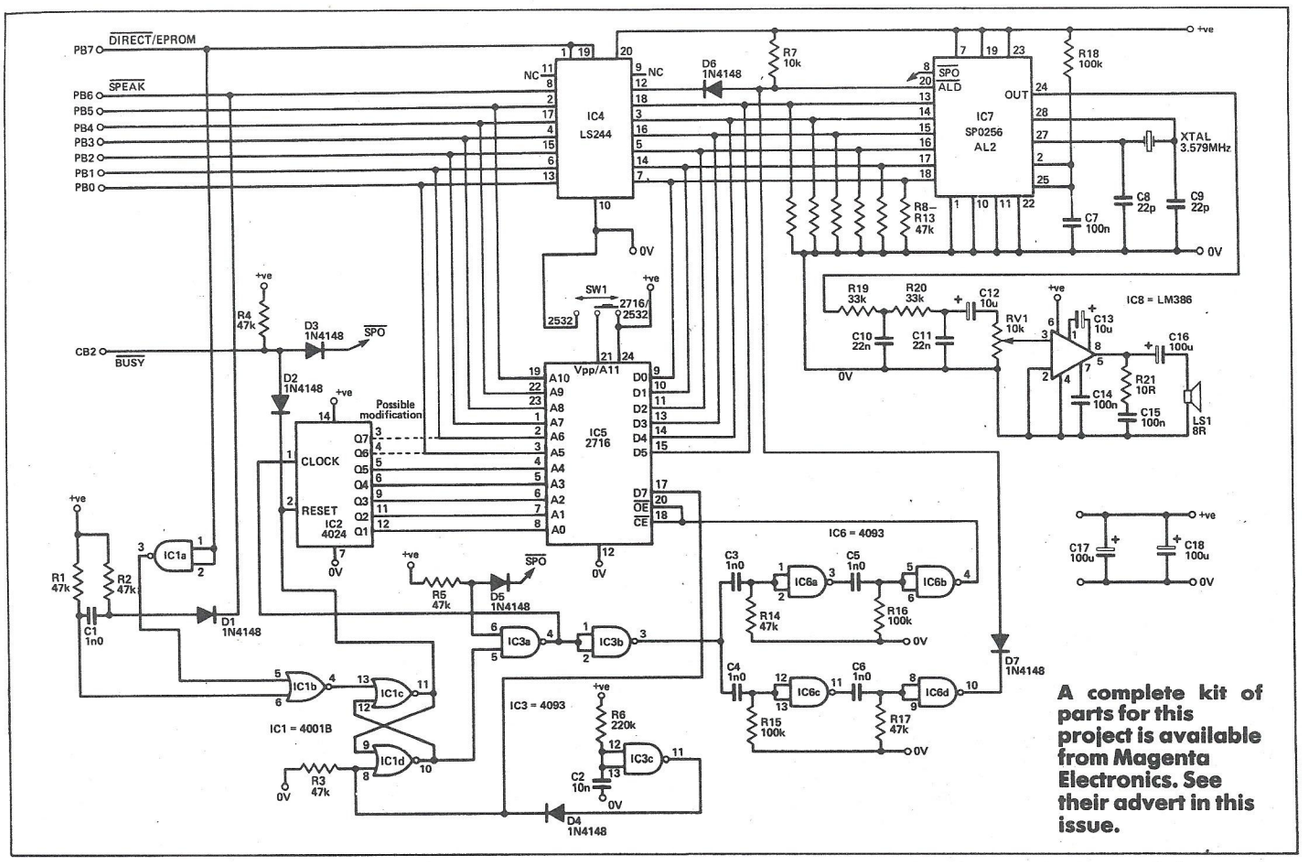



*Figure 1. Circuit diagram of the synthesiser.*

The circuit waits in a stable state until IC7 has completed its routine. At the end of the routine the SPO output changes from low to high, IC3a output also falls and clocks IC2 and IC6 repeats its pulse sequence. IC7 again loads the EPROM address, which is derived from IC2, and has increased by 1. IC7 therefore executes the allophones from the EPROM one by one. With IC2 and IC5 connected as shown, the lower 5 address lines would cycle indefinitely without some means of ending the sequence. To accomplish this, one of the two higher order data lines of the EPROM is used. The 64 available allophones require only 6 data lines from the 8 bit EPROM. The higher two lines are normally set to 0. By setting one of the high order lines to logic 1, it is simple to indicate the end of a sequence. EPROM data line

**"... allows the computer to select words or phrases stored in EPROM".**

D7 is connected ot IC1d. The control bistable is reset whenever a logic 1 appears on D7, IC2 is reset, and the BUSY signal returns from low to high. The circuit is then returned to its initial state.

A 'power on reset' pulse is produced by IC3c to ensure that the control bistable is initially in the correct state.

IC7 has its own clock oscillator which uses C8, C9 and crystal X1. A 3.12MHz crystal is specified by the manufacturers, but a cheaper 3.57MHz USA colour TV crystal operates well. The speech output from IC7 passes through a simple low pass filter network and volume control, to a standard IC audio amplifier. A miniature speaker provides the final link.

## Construction

Construction is very straightforward. A double sided printed circuit board is used, see **Figures 3a** and **3b;** the only off board components are the volume control RV1 and the loudspeaker.

Follow the overlay drawing to be published next month when assembling the board. Note that through connections are made using PCB pins, and that some component leads are soldered on both sides of the board. IC sockets are recommended but are not essential except for the EPROM.

Connections to the speaker, RV1, and the ribbon cable from the computer are best made using PCB pins.

Although only 11 connections are made to the computer it is better to use a length of 20-way ribbon cable, and to cut back the unused leads at the synthesiser end.

## Programming

Direct mode allows the computer to operate the speech synthesiser IC directly. The mode is selected by setting the user port line PB7 to a logic 0. The user port must first be set up as an output port by using the instruction:

?&FE62=FF

PB7 is set low by any number less than 127. The production of an allophone is initiated by a high to low transition of PB6. The other 6 port lines select the required allophone. The allophone table, **Figure 4,**

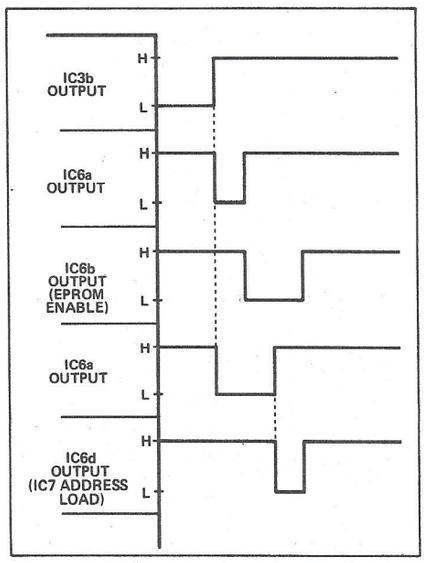



*Figure 2. The relationship between the various timing pulses.*

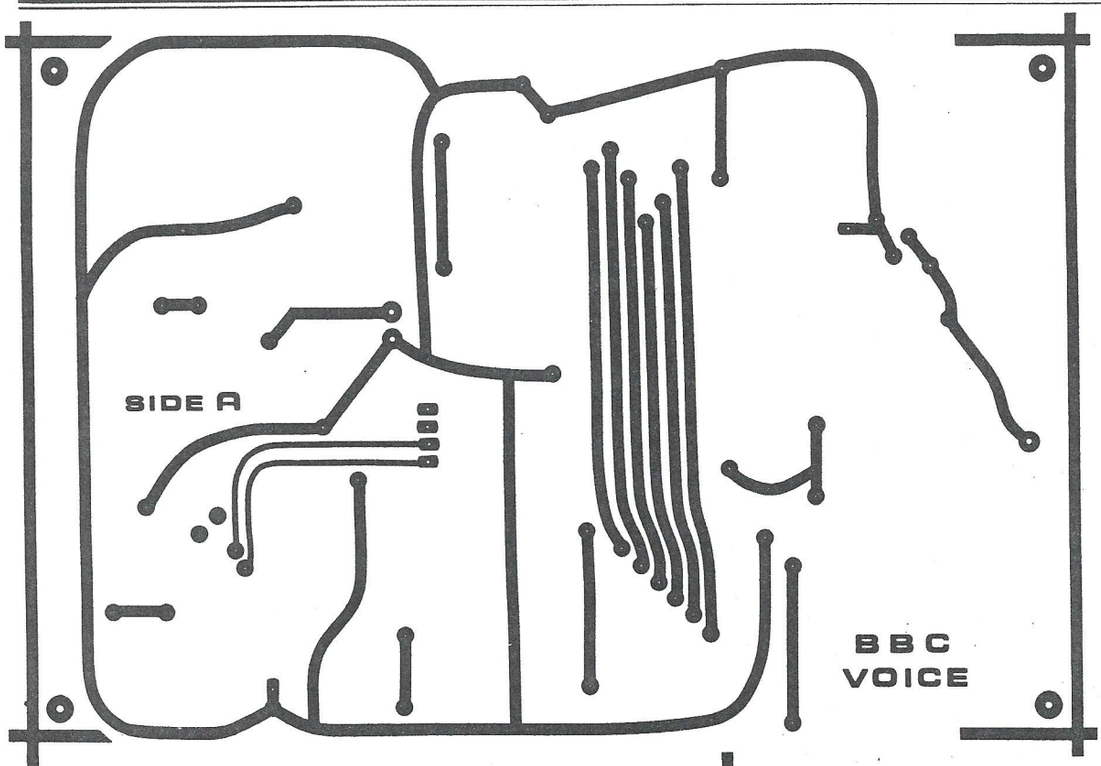



*Figure 3a. Top side foil of the double sided board.*

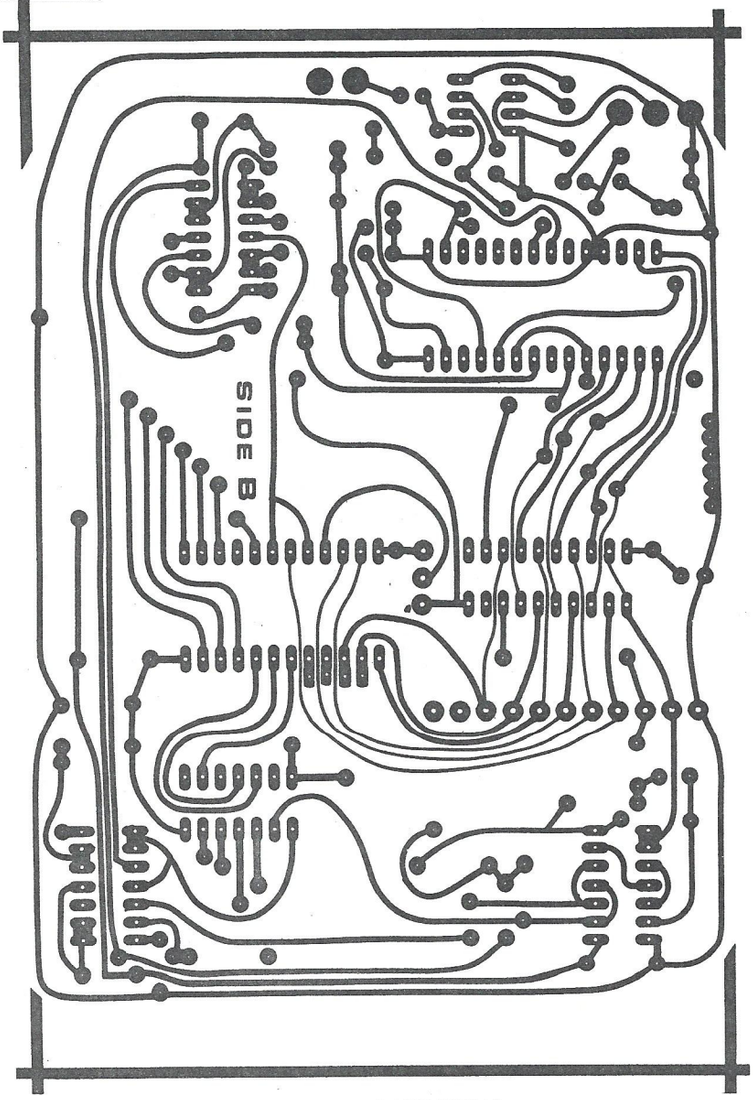



*Figure 3b. Lower foil pattern.*

lists the 64 sounds and pauses that can be produced.

The following short program allows each allophone to be selected and sounded:

```
10 ?&FE60=64
15 INPUT A
20 ?&FE60=A
30 GOTO 10
```

Line 10 sets all the port lines except PB6 to 0. The decimal number of the required allophone is entered in line 15. In line 20 the allophone number is put on port lines PB0-PB5, and PB6 is changed from high to low. The falling edge on PB6 tells IC7 to latch the allophone data into its register, and commence execution. There is a problem with this simple program because IC7 continues to produce the allophone until a new one is entered. A pause (data 0-4) must always be entered to finish a phrase or word.

It is necessary for the computer to know when one allophone has ended, so that the next one can be loaded inot IC7 without leaving an audible pause. This timing function is made possible by the SPO output of IC7 which goes low whilst each allophone is sounded. A low to high transition on the SPO line occurs at the end of each allophone. This BUSY signal is coupled to the computer port control line CB2.

Those familiar with the 6522 computer VIA chip will be aware of its many complexities. Without going into detail the relevant operation can be summarised as follows. Control line CB2 influences data Bit 3 of a register in the 6522 known as the interrupt flag register (IFR). The register can be read by the computer at FE6D Hex. Normally the register contains all '0s', so a simple test can be used to check on Bit 3.

The effect of CB2 on IFR Bit 3 is controlled by another 6522 register at FE6C called the peripheral control register (PCR). By setting Bit 6 of the PCR to a logic 1, a low to high transition on CB2 will set Bit 3 in the IFR to 1. Three of the bits in the PCR are already set (Bits 1, 2 and 3) and this must be allowed for when setting Bit 6.

When the computer sends an output to the user port PB lines Bit 3 in the IFR is automatically set to 0. After IC7 has executed the appropriate allophone, the BUSY line switches from logic 0 to logic 1 so instructing the computer to issue the next instruction.

The simple BASIC program that follows should clarify the method of operation:

```
  4 ?&FE62=&FF          Sets up the output port
  8 ?&FE6C=78           Sets up the PCR
 10 ?&FE60=64           Sets PB6 high
 20 READ A              Get allophone data
 30 IF A=64 THEN GOTO 120   Tests for end condition
 40 ?&FE60=A            Loads data into IC7
 50 B=?&FE6D            Reads IFR
 60 IF B=0 THEN GOTO 50 Loop while allophone is executed
 70 GOTO 10             Repeat for next allophone
 80 DATA 20,29,36,47...etc..64   List of allophone codes separated by commas
                                 ending with 64
120 STOP
```

The program loops around lines 50 and 60 whilst the BUSY line is low. When the BUSY line is switched to high, B is no longer 0 so the program proceeds to line 70, and returns for the next.

## EPROM Programming

The EPROM mode of operation is set by a logic 1 on PB7 of the computer port. Speech is initiated by a high to low transition of PB6. The other 6 data lines determine which bank of 32 EPROM addresses is used. If PB 0-5 are set to zero the EPROM is addressed from 01 to 32. If PB0 is set to 1 the 32 EPROM addresses start at 33 and end at 64, etc.

Note that the sequence can be made shorter than 32 bytes by putting an end character in the EPROM (any number between decimal 128 and 255, or 80-FF Hex).

All of this translates into BBC code as follows:

First enter: ?&FE62=FF

This sets the user port lines PB 0-7 to output mode.

Then enter the following short program:

```
 5 INPUT A
10 ?&FE62=192+A
20 ?&FE60=128+A
30 GOTO 5
```

The value of 'A' determines the bank of addresses to be used, and can take a value from 0 to 63.

Adding 192 to A in line 10 sets PB6 and 7 high, and so sets up EPROM mode. Line 20 leaves the value of PB7 high whilst changing PB6 from high to low. The falling edge of PB6 initiates the speech cycle. Note that if an end character is not found in the EPROM the circuit will latch, and continuously cycle through the 32 addresses. Switching the power off and on will restore normal operation.

Alternative phrase lengths can be set by changing the hardware connections. If the Q6 output of IC2 is connected to A5 of IC5, and the computer PB0 line disconnected, there will be 32 banks of 64 addresses available. Linking Q7 of IC2 to A6 of IC6 as well, will give 16 banks of 128 addresses.

To use the speech in a program it is only necessary to run lines 10 and 20 with the required value of 'A' added to the 192 and 128. Remember though to initiate the port by entering ?&FE62=FF somewhere in the program, before the speech part.

Mixing EPROM speech with direct speech can be achieved simply by switching backwards and forwards using software. The busy condition may be read from the IFR in exactly the same way for both the direct and EPROM modes. ■

## PARTS LIST

**Resistors (¼Watt 5% Carbon Film)**

| | |
|---|---|
| R1,2,3,4,5,8,9,10,11,12,13,14,17 | 47K |
| R6 | 220K |
| R7 | 10K |
| R15,16,18 | 100K |
| R19,20 | 33K |
| R21 | 10R |
| RV1 | 10K log potentiometer |

**Capacitors**

| | |
|---|---|
| C1,3,4,5,6 | 1nF ceramic plate |
| C2 | 10nF C280 |
| C7,14,15 | 100nF |
| C8,9 | 22pF ceramic plate |
| C10,11 | 22nF C280 |
| C12,13 | 10uF 10V radial electrolytic |
| C16,17,18 | 100uF 10V radial electrolytic |

**Semiconductors**

| | |
|---|---|
| D1-7 | IN4148 |
| IC1 | 4001B |
| IC2 | 4024B |
| IC3,6 | 4093B |
| IC4 | 74LS244 |
| IC5 optional | 2716/2732 EPROM |
| IC7 | SPO256-AL2 |
| IC8 | LM386 |

**Miscellaneous**

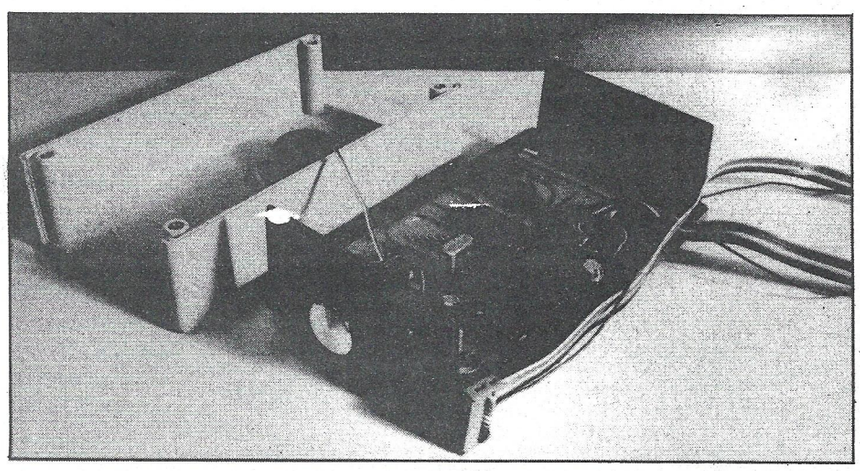
Crystal - X1 3.579MHz or 3.12MHz; Speaker - Miniature 8 ohm; 11 way ribbon cable; IC sockets; Socket for micro port; Knob; Case; Fixings; PCB etc.

*Figure 4. The 64 allophones generated by the synthesiser.*

| ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD |
|---|---|---|---|---|---|---|---|---|
| 0 | 10ms PAUSE | | 22 | UW1 | tO | 44 | NG | aNchore |
| 1 | 30ms PAUSE | | 23 | A0 | AUght | 45 | LL | Lake |
| 2 | 50ms PAUSE | | 24 | AA | hOt | 46 | WW | Wool |
| 3 | 100ms PAUSE | | 25 | YY2 | Yes | 47 | XR | repaiR |
| 4 | 200ms PAUSE | | 26 | AE | hAt | 48 | WH | WHile |
| 5 | OY | bOY | 27 | HH1 | He | 49 | YY1 | Yes |
| 6 | AY | skY | 28 | BB1 | daB | 50 | CH | CHurch |
| 7 | EH | End | 29 | TH | THin | 51 | ER1 | summER |
| 8 | KK3 | Comb | 30 | UH | bOOk | 52 | ER2 | bURn |
| 9 | PP | Pit | 31 | UW2 | fOOd | 53 | OW | nOW |
| 10 | JH | dodGe | 32 | AW | OUt | 54 | DH2 | THey |
| 11 | NN1 | thiN | 33 | DD2 | Do | 55 | SS | veST |
| 12 | IH | sIt | 34 | GG3 | wiG | 56 | NN2 | No |
| 13 | TT2 | To | 35 | VV | Vest | 57 | HH2 | Noe |
| 14 | RR1 | Rural | 36 | EF1 | GUest | 58 | OR | stORe |
| 15 | AX | sUcceed | 37 | SH | SHip | 59 | AR | alArm |
| 16 | MM | Milk | 38 | ZH | aZure | 60 | YR | clearR |
| 17 | TT1 | parT | 39 | RR2 | bRain | 61 | EG2 | Got |
| 18 | DH1 | THey | 40 | FF | Food | 62 | EL | saddle |
| 19 | IY | sEE | 41 | KK2 | sKy | 63 | BB2 | Business |
| 20 | EY | bElge | 42 | KK1 | Can't | | | |
| 21 | DD1 | coulD | 43 | ZZ | Zoo | | | |

# Speech synthesiser for BBC micro

## And now a speech synthesiser for the BBC micro. Mark Stewart builds a low cost board with the same SPO256-AL2 allophone chip used in last month's project for the Spectrum, but with the added extra of EPROM storage.



Following on from the discussion of speech synthesis in the November and December '83 issues of *E&CM*, this article presents a practical speech synthesiser for use with the BBC Model B. Using the General Instruments allophone IC, the SPO256-AL2, this circuit allows two different modes of operation. In "direct" mode the computer controls the speech synthesiser IC directly, allowing total software control of individual allophones, which may be strung together to produce the required speech. In the second mode the computer selects a word or phrase from an on-board EPROM; 64 phrases, each up to 32 allophones long can be stored in a single 2716 2K EPROM. Alternative connections allow the format to be changed so that 32 phrases of 64 allophones, or 16 phrases of 128 allophones can be used.

It is also possible to use a 2732 4K EPROM and select between 2 banks of 2K using an on board switch. Switching between the two modes is done in software, so that a program can mix "standard phrases" from the EPROM with direct, software controlled words.

A particular advantage of the EPROM mode is that the computer is freed once it has initiated the phrase required. Thus fast action games can be combined with speech, using simple programming, without loss of speed. It is of course, necessary to construct a speech EPROM vocabulary for a particular set of programs. The "direct mode" allows the words or phrases to be set up experimentally in RAM before committing them to EPROM, using an EPROM programmer (but of course, the system can be used without an EPROM).

## The Circuit

A full circuit diagram is shown in **Figure 1.** The speech synthesiser IC7 has 6 data input lines which address the 64 allophones listed in **Figure 4.** The direct mode is selected by a logic low on the computer user port line, PB7. Tri state buffer, IC4, is enabled and the 6 data lines of IC7 are connected to PB0-5 of the computer user port. User port line PB6 is IC7. A signal from the active low output (SPO) of IC7 is coupled to the CB2 control line of the computer port. This line is pulled low by IC7 when it is busy. A rising edge indicates to the computer that the next allophone data can be loaded.

Selection of direct mode automatically disables the EPROM. Its outputs assume a high impedance state and are effectively disconnected from the system.

When PB7 is high EPROM speech is enabled. IC4 output lines are all set to their high impedance state. IC1a inverts the signal from PB7 and sets one input of IC1b low. The other input of NOR gate IC1b is fed from the SPEAK line PB6 via D1 and the pulse forming network R1, R2 and C1. A high to low transition of PB6 produces a short positive pulse at the output of IC1b. This pulse sets the bistable formed by cross coupled gates IC1c and d. The output of IC1c changes from high to low. This removes the reset condition from the 7 bit binary counter IC2, and pulls down the BUSY line via D2. Setting the bistable also changes the output of IC1d from low to high. The output from IC1d puts a high state on one input of IC3a, the other input of which is held high by the inactive SPO output of IC7. The output of IC3a therefore changes from high to low, and IC2 is clocked 1 count from zero. The output from IC3a is inverted by IC3b and coupled to the two delayed pulse forming circuits IC6a, b and IC6c, d. The low to high transition from IC3b is coupled, via C3, to the inputs of IC6a. The inputs of IC6a are pulled high via C3. C3 then begins to charge via R14, and the inputs are pulled back down again to zero, even though the output of IC3b may still be high. The result of this is that a short negative pulse is produced on the output of IC6a, starting when IC3b output changes from low to high. The length of the pulse is set by the values of C3 and R14.

IC6b is an identical stage which produces a negative pulse which starts on the rising edge at the end of the negative pulse from IC6a. A similar pattern occurs with IC6c and IC6d, but the relative pulse lengths differ because different value timing resistors are used. **Figure 2** shows these pulses graphically. The output pulse from IC6b enables the EPROM. The falling edge of the pulse on the output of IC6d is timed to occur during the time that the EPROM is enabled. This pulse is used by IC7 which loads the data present on the EPROM data lines, and starts to execute the first allophone routine. As soon as IC7 loads the data the 'busy' condition is signalled by the SPO output changing from high to low. This signal is coupled via D5 to IC3a. The inputs to NAND gate, IC3a, change from both being high to one being high and one being low. Its output changes from low to high. This change has no effect on the counter, which is clocked on falling edges. The change is inverted by IC3b, but has no effect on IC6 which is triggered only by rising edges.

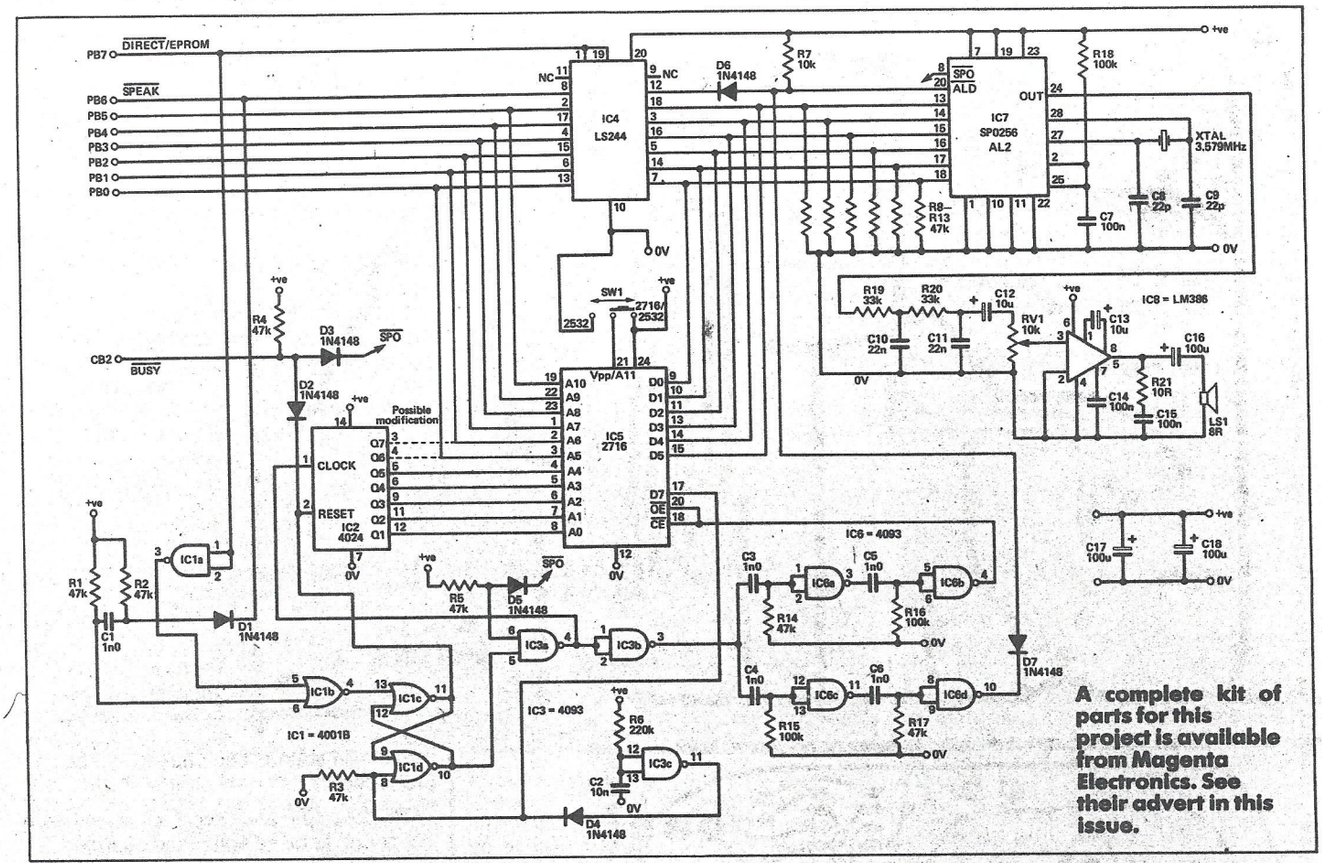



*Figure 1. Circuit diagram of the synthesiser.*

The circuit waits in a stable state until IC7 has completed its routine. At the end of the routine the SPO output changes from low to high, IC3a output also falls and clocks IC2 and IC6 repeats its pulse sequence. IC7 again loads the EPROM address, which is derived from IC2, and has increased by 1. IC7 therefore executes the allophones from the EPROM one by one. With IC2 and IC5 connected as shown, the lower 5 address lines would cycle indefinitely without some means of ending the sequence. To accomplish this, one of the two higher order data lines of the EPROM is used. The 64 available allophones require only 6 data lines from the 8 bit EPROM. The higher two lines are normally set to 0. By setting one of the high order lines to logic 1, it is simple to indicate the end of a sequence. EPROM data line D7 is connected ot IC1d. The control bistable is reset whenever a logic 1 appears on D7, IC2 is reset, and the BUSY signal returns from low to high. The circuit is then returned to its initial state.

A 'power on reset' pulse is produced by IC3c to ensure that the control bistable is initially in the correct state.

IC7 has its own clock oscillator which uses C8, C9 and crystal X1. A 3.12MHz crystal is specified by the manufacturers, but a cheaper 3.57MHz USA colour TV crystal operates well. The speech output from IC7 passes through a simple low pass filter network and volume control, to a standard IC audio amplifier. A miniature speaker provides the final link.

## Construction

Construction is very straightforward. A double sided printed circuit board is used, see **Figures 3a** and **3b**; the only off board components are the volume control RV1 and the loudspeaker.

Follow the overlay drawing to be published next month when assembling the board. Note that through connections are made using PCB pins, and that some component leads are soldered on both sides of the board. IC sockets are recommended but are not essential except for the EPROM.

Connections to the speaker, RV1, and the ribbon cable from the computer are best made using PCB pins.

Although only 11 connections are made to the computer it is better to use a length of 20-way ribbon cable, and to cut back the unused leads at the synthesiser end.

**"... allows the computer to select words or phrases stored in EPROM".**

## Programming

Direct mode allows the computer to operate the speech synthesiser IC directly. The mode is selected by setting the user port line PB7 to a logic 0. The user port must first be set up as an output port by using the instruction:

?&FE62=FF

PB7 is set low by any number less than 127. The production of an allophone is initiated by a high to low transition of PB6. The other 6 port lines select the required allophone. The allophone table, **Figure 4,**

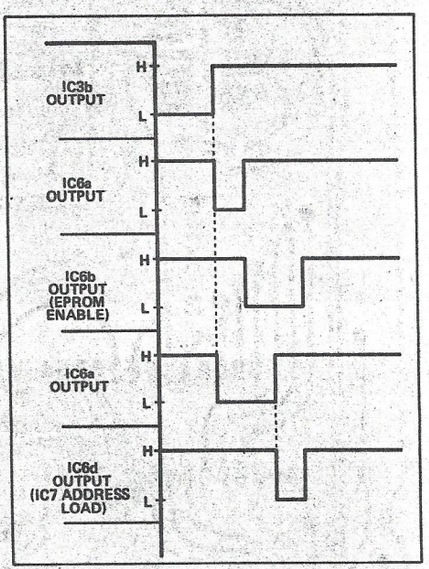



*Figure 2. The relationship between the various timing pulses.*

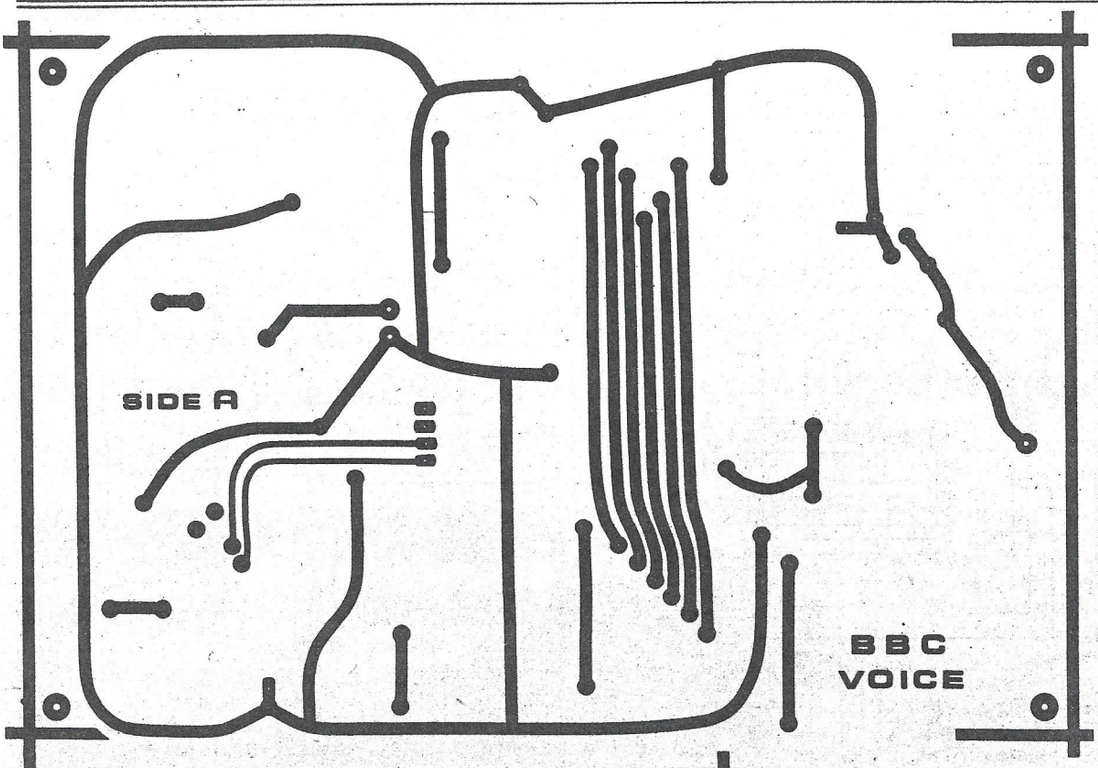



*Figure 3a. Top side foil of the double sided board.*

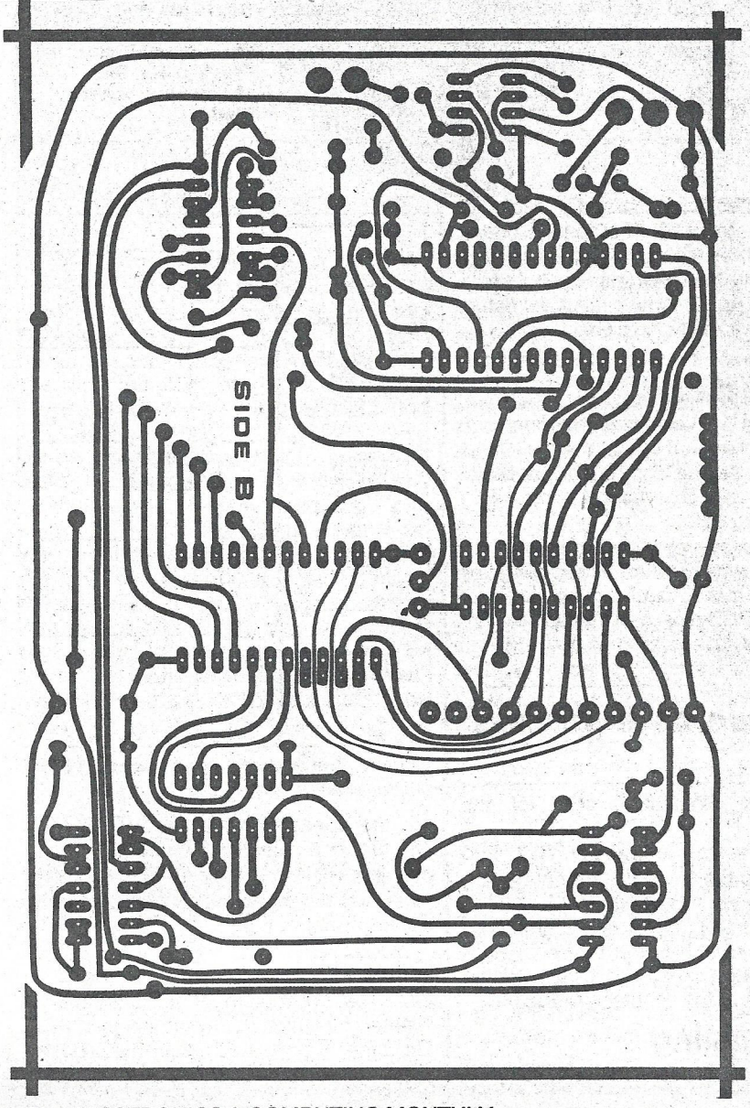



*Figure 3b. Lower foil pattern.*

lists the 64 sounds and pauses that can be produced.

The following short program allows each allophone to be selected and sounded:

```
10 ?&FE60=64
15 INPUT A
20 ?&FE60=A
30 GOTO 10
```

Line 10 sets all the port lines except PB6 to 0. The decimal number of the required allophone is entered in line 15. In line 20 the allophone number is put on port lines PB0-PB5, and PB6 is changed from high to low. The falling edge on PB6 tells IC7 to latch the allophone data into its register, and commence execution. There is a problem with this simple program because IC7 continues to produce the allophone until a new one is entered. A pause (data 0-4) must always be entered to finish a phrase or word.

It is necessary for the computer to know when one allophone has ended, so that the next one can be loaded inot IC7 without leaving an audible pause. This timing function is made possible by the SPO output of IC7 which goes low whilst each allophone is sounded. A low to high transition on the SPO line occurs at the end of each allophone. This BUSY signal is coupled to the computer port control line CB2.

Those familiar with the 6522 computer VIA chip will be aware of its many complexities. Without going into detail the relevant operation can be summarised as follows. Control line CB2 influences data Bit 3 of a register in the 6522 known as the interrupt flag register (IFR). The register can be read by the computer at FE6D Hex. Normally the register contains all '0s', so a simple test can be used to check on Bit 3.

The effect of CB2 on IFR Bit 3 is controlled by another 6522 register at FE6C called the peripheral control register (PCR). By setting Bit 6 of the PCR to a logic 1, a low to high transition on CB2 will set Bit 3 in the IFR to 1. Three of the bits in the PCR are already set (Bits 1, 2 and 3) and this must be allowed for when setting Bit 6.

When the computer sends an output to the user port PB lines Bit 3 in the IFR is automatically set to 0. After IC7 has executed the appropriate allophone, the BUSY line switches from logic 0 to logic 1 so instructing the computer to issue the next instruction.

The simple BASIC program that follows should clarify the method of operation:

```
  4 ?&FE62=&FF          Sets up the output port
  8 ?&FE6C=78           Sets up the PCR
 10 ?&FE60=64           Sets PB6 high
 20 READ A              Get allophone data
 30 IF A=64 THEN GOTO 120   Tests for end condition
 40 ?&FE60=A            Loads data into IC7
 50 B=?&FE6D            Reads IFR
 60 IF B=0 THEN GOTO 50 Loop while allophone is executed
 70 GOTO 10             Repeat for next allophone
 80 DATA 20,29,36,47 ..etc..64   List of allophone codes separated by commas
                                 ending with 64
120 STOP
```

The program loops around lines 50 and 60 whilst the BUSY line is low. When the BUSY line is switched to high, B is no longer 0 so the program proceeds to line 70, and returns for the next.

## EPROM Programming

The EPROM mode of operation is set by a logic 1 on PB7 of the computer port. Speech is initiated by a high to low transition of PB6. The other 6 data lines determine which bank of 32 EPROM addresses is used. If PB 0-5 are set to zero the EPROM is addressed from 01 to 32. If PB0 is set to 1 the 32 EPROM addresses start at 33 and end at 64, etc.

Note that the sequence can be made shorter than 32 bytes by putting an end character in the EPROM (any number between decimal 128 and 255, or 80-FF Hex).

All of this translates into BBC code as follows:

First enter: ?&FE62=FF

This sets the user port lines PB 0-7 to output mode.

Then enter the following short program:

```
 5 INPUT A
10 ?&FE62=192+A
20 ?&FE60=128+A
30 GOTO 5
```

The value of 'A' determines the bank of addresses to be used, and can take a value from 0 to 63.

Adding 192 to A in line 10 sets PB6 and 7 high, and so sets up EPROM mode. Line 20 leaves the value of PB7 high whilst changing PB6 from high to low. The falling edge of PB6 initiates the speech cycle. Note that if an end character is not found in the EPROM the circuit will latch, and continuously cycle through the 32 addresses. Switching the power off and on will restore normal operation.

Alternative phrase lengths can be set by changing the hardware connections. If the Q6 output of IC2 is connected to A5 of IC5, and the computer PB0 line disconnected, there will be 32 banks of 64 addresses available. Linking Q7 of IC2 to A6 of IC6 as well, will give 16 banks of 128 addresses.

To use the speech in a program it is only necessary to run lines 10 and 20 with the required value of 'A' added to the 192 and 128. Remember though to initiate the port by entering ?&FE62=FF somewhere in the program, before the speech part.

Mixing EPROM speech with direct speech can be achieved simply by switching backwards and forwards using software. The busy condition may be read from the IFR in exactly the same way for both the direct and EPROM modes. ■

## PARTS LIST

Resistors (all watt 5% Carbon Film)
R1,2,3,4,5,8,9,10,11,12,13,14,17 — 47K
R6 — 220K
R7 — 10K
R15,16,18 — 100K
R19,20 — 33K
R21 — 10R
RV1 — 10K log potentiometer

Capacitors
C1,3,4,5,6 — 1nF ceramic plate
C2 — 10nF C280
C7,14,15 — 100nF
C8,9 — 22pF ceramic plate
C10,11 — 22uF C280
C12,13 — 100uF 10V radial electrolytic
C16,17 — 100uF 10V radial electrolytic

Semiconductors
D1-7 — IN4148
IC1 — 4001B
IC2 — 4024B
IC3,6 — 4093B
IC4 — 74LS244
IC5 optional — 2716/2732 EPROM
IC7 — SPO256-AL2
IC8 — LM386

Miscellaneous
Crystal — X1 3.579MHz or X 3.12MHz
Speaker — Miniature 8 ohm (1 watt rating)
Iron cable, IC sockets, Socket for user port, Knob, Case, Tuning — PCB (ref. )

*Figure 4. The 64 allophones generated by the synthesiser.*

| ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD | ADDRESS | ALLOPHONE | EXAMPLE WORD |
|---|---|---|---|---|---|---|---|---|
| 0 | 10ms PAUSE | | 22 | UR2 | fUr | 44 | NG | aNchor |
| 1 | 30ms PAUSE | | 23 | AO | AUght | 45 | LL | Cake |
| 2 | 50ms PAUSE | | 24 | AR | fOr | 46 | WW | Wool |
| 3 | 100ms PAUSE | | 25 | YY2 | Yes | 47 | XR | mirrOr |
| 4 | 200ms PAUSE | | 26 | AE | hAt | 48 | WH | WHig |
| 5 | OY | bOY | 27 | HH1 | He | 49 | YY1 | Yes |
| 6 | AY | sKY | 28 | BB1 | daB | 50 | CH | CHurch |
| 7 | EH | End | 29 | TH | THin | 51 | ER1 | sUmmER |
| 8 | KK3 | Comb | 30 | UH | bOOk | 52 | ER2 | sUrfur |
| 9 | PP | Pin | 31 | UW2 | fOOd | 53 | OW | nOw |
| 10 | JH | doDge | 32 | AW | OUt | 54 | DH2 | THey |
| 11 | NN1 | thiN | 33 | DD2 | Do | 55 | SS | veST |
| 12 | IH | sIt | 34 | GG3 | wiG | 56 | NN2 | No |
| 13 | TT2 | To | 35 | VV | Vest | 57 | HH2 | No |
| 14 | RR1 | Rural | 36 | EF1 | fUest | 58 | OR | sHORe |
| 15 | AX | sUccEEd | 37 | SH | SHip | 59 | AR | alarm |
| 16 | MM | Milk | 38 | ZH | aZure | 60 | YR | clEAR |
| 17 | TT1 | parT | 39 | RR2 | bRain | 61 | EG2 | Get |
| 18 | DH1 | THey | 40 | FF | Food | 62 | EL | scrOLl |
| 19 | IY | sEE | 41 | KK2 | sKy | 63 | BB2 | Business |
| 20 | EY | bEIge | 42 | KK1 | Can't | | | |
| 21 | DD1 | coulD | 43 | ZZ | Zoo | | | |