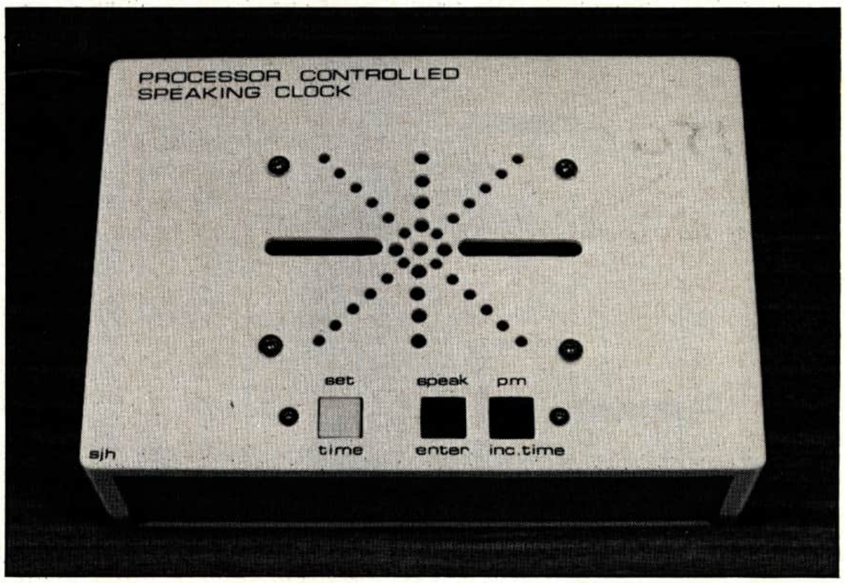
SPEAKING CLOCK

BY STEPHEN HUNT

I SPEAK YOUR DATA

In the beginning was the Word ... and time began. Seconding the SPO256AL speech chip to verbalise the output of a real-time clock may be a less momentous event, but it continues an ancient and honourable tradition!



Many articles have been designed around the SPO256AL speech synthesiser. However, the majority of them have been simply for experimentation in conjunction with a home computer and serve no real application. This article describes how the allophone speech system can be used to produce a speaking clock which may be invaluable to a blind person or simply a novelty clock demonstrating the use of current electronic devices.

The block and circuit diagrams are shown in Fig.1. The system forms a simple microprocessor controller which is used to process data from the real time clock. The speaking clock uses a 6802 micro-processor in conjunction with a 2716 eprom, 6821 pia and an ICM7170 real time clock. The heart of the system is the program stored in the rom.

MEMORY MAP

The eprom occupies locations 8000H to 87FFH (2K). The address decoding for IC3 uses A15 via an inverter IC6a when the eprom is enabled low. The pia, IC2, is selected by A14 and A2, corresponding to location 4004H. To prevent the pia and eprom being selected at the same time A15 is also connected to CS2. RS0 and RS1 are connected to A0 and A1 respectively and are used to select internal registers which control port operations. The pia places the speech synthesiser in the memory map.

It might appear at first that using a pia to interface with IC4 and a few switches is over complicating matters. Alternatively, octal buffers such as the 74LS244 type of devices could be used but would require extra decoding logic, take up more room and cost more. The 6821 has three select lines (CS0, CS1 and CS2) which can be connected to appropriate address lines to place it in the memory map, and allows for a much simpler approach. The 6821 also contains two user ports, A and B which can be set by software as inputs or outputs as required. The speaking clock program requires PA0 to PA6 to be outputs and PA7 to be an input line. Port B lines are all set as inputs. Port A controls the

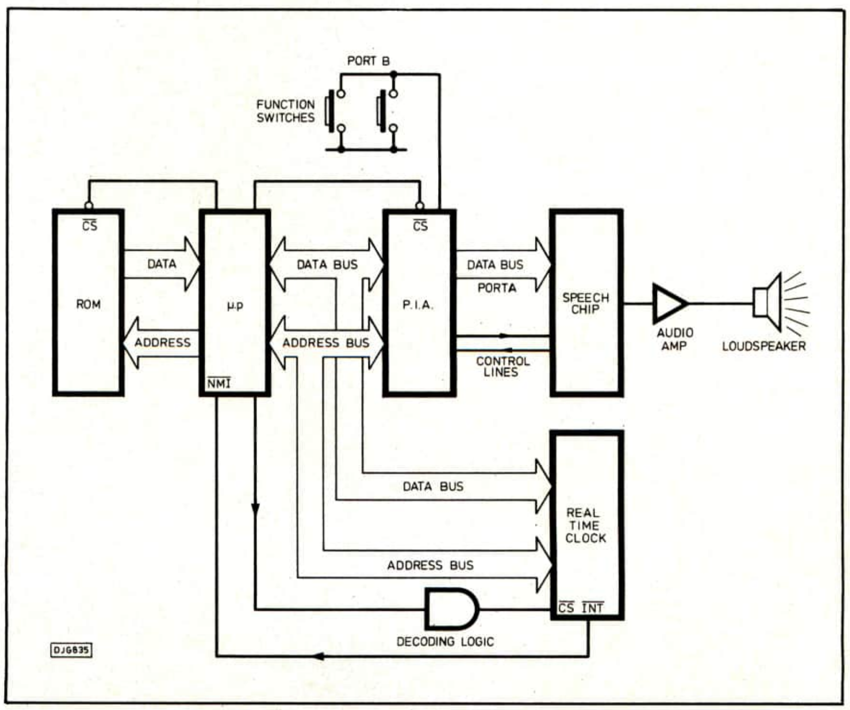


Fig.1. *Block diagram for the speaking clock*

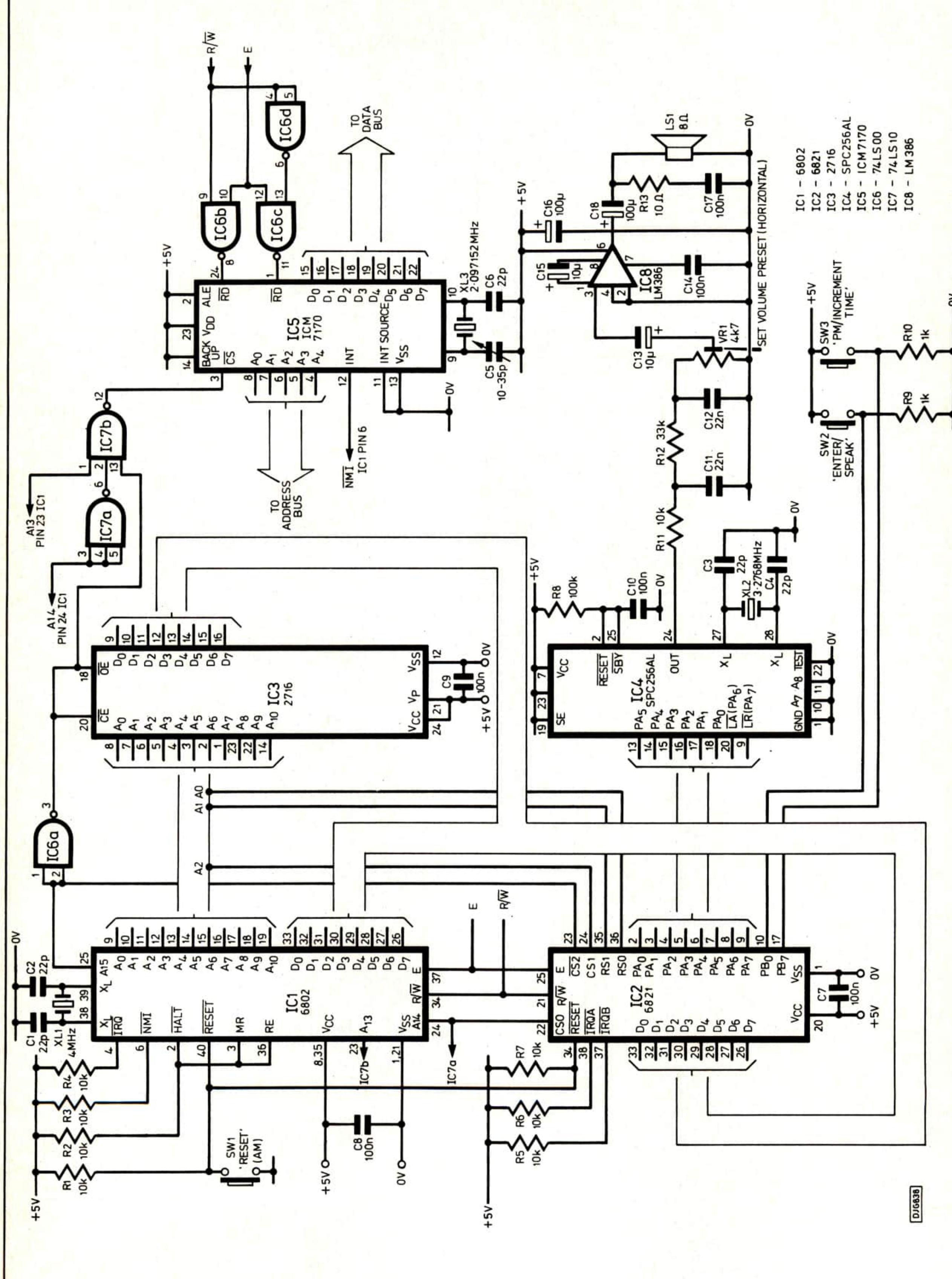


Fig.2. *Circuit diagram for the speaking clock*

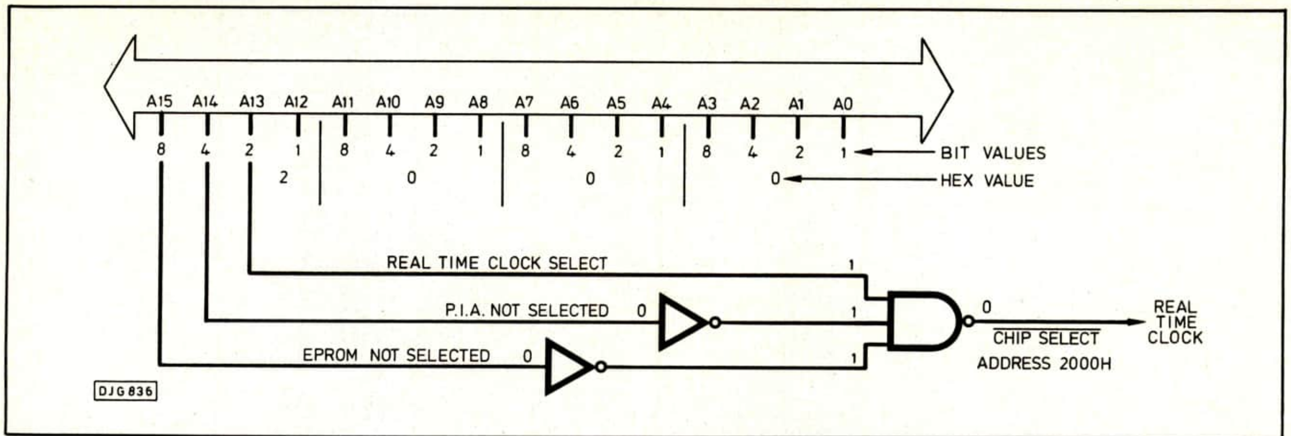


Fig.3. *Obtaining address decoding*

speech synthesiser while port B looks at the function switches, S2 and S3.

The real time clock, IC5, sits at location 2000H and is selected by IC7 from address lines A13, A14 and A15. Fig.3 shows how the decoding is obtained. All other decoding is done in a similar way.

The speaking clock uses a small amount of ram for data storage and as a stack for holding return addresses when jumping to subroutines. This ram is contained within the 6802 at locations 0000 to 007FH (128 bytes).

It should be realised that the devices are only partially decoded and therefore appear in several blocks in memory. However, as only three locations are decoded in the 64K area, this causes no problems of accidental enabling and is therefore perfectly acceptable.

SPEAK TO ME

Words are generated by stringing together allophones which make up the basic sounds of speech. By carefully selecting them a fairly reasonable quality of speech can be produced bearing in mind the limitations when compared to digitally encoded voice systems.

The data control (handshaking) is performed by two lines, load address and load request. When the synthesiser is ready for data the load request line is held low. Then the allophone data is placed onto the bus and by taking the load address low latches the data into the chip. While it is speaking (busy) the load request line is taken high and only when it goes low again can you send the next data. The processor looks at PA6 and PA7 of IC2 to control the data flow.

The speed or pitch at which it speaks is governed by XL2. The prototype used a 3.2768 MHz crystal but any value in the range of 3 to 3.5 MHz will do. The output from IC4 is pulse width modulated at 20kHZ and this is removed by a low pass filter formed by R11, C11, R12 and C12. The resulting audio is then amplified by a small power amplifier, IC8, to drive a loudspeaker.

No allophone data has been given here as many readers will have probably come across this device before. *(Allophone Data Principles were discussed in the MicroChat article in PE Sept–Oct 87. Ed.)*

KEEPING TIME

In order to keep track of the time a real-time clock is used. This device contains an onboard oscillator and a series of registers which contains data for hours, minutes, seconds etc. These registers are simply cascaded up-counters, ie, the seconds counter sequences from zero to sixty then the carry-out from this counter increments the next, which in this case is the minutes. The counters can be preloaded to any value before the oscillator is enabled. Each register has an address and by sending data to that address the counter is loaded. The hours counter is a programmable divider which can be set by software to divide by 12 or 24 (12 or 24 hour mode). The real-time clock is completely independent from the processor and is free running. The processor simply preloads the registers with time data and starts the clock. By reading the register addresses the processor can obtain the time which the program then sorts out to provide the speech synthesiser with word data.

When the processor has finished reading data from the real-time clock it scans the function switches. Not exactly doing too much, is it? This may seem rather a waste of processing power. Alternatively, you could use the processor to keep time by using its internal registers as programmable dividers. This however, requires several processor registers which are not being used for data manipulation and as the 6802 has only three working registers this is not feasible. The use of a real-time clock reduces the program length and complexity substantially.

The ICM7170 real-time clock was chosen mainly because it is easy to use. It is completely software controlled and has a total of 18 internal registers. The main register of interest is the command register which is used to set up operation modes. Table 1 shows the assigned bits.

Bits D0 and D1 set the internal clock frequency. In the prototype a 2.097MHz crystal was used but others can be accommodated by making the program changes shown in Table 2.

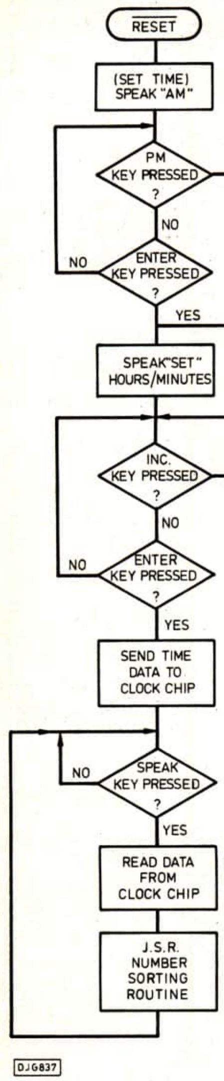
D2 is set to zero for 12 hours. Stopping and starting of the clock's counters is obtained by writing a 0 or 1 respectively into the bit location. D4 sets up interrupts for alarms and D5 is used as a test bit which clocks the counters at a faster rate. D5 and D7 are not used.

D7	D6	D5	D4	D3	D2	D1	D0
—	—	TEST	INT	RUN	12/24	FREQ	FREQ

Table 1

CRYSTAL FREQ	D0	D1	PROGRAM CHANGES
32.768KHZ	0	0	0010 00 04BC 08
1.048576MHz	0	1	0010 01 04BC 09
2.097152MHz	1	0	AS IN HEX LIST
4.194304MHz	1	1	0010 03 04BC 0B

Table 2



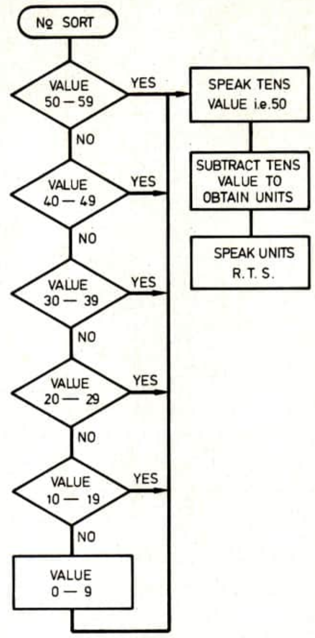


Fig.4. *Basic program flow*

```
0000 8E 00 10 CE 7F 04 FF 40
0008 04 CE 7E 04 FF 40 06 86
0010 02 B7 20 11 86 00 B7 20
0018 03 7E 84 00 20 02 B7 20
0020 03 86 0A B7 20 11 7E 83
0028 00 FF FF FF FF FF FF FF
0030 FF F6 40 06 C4 01 27 F9
0038 B6 20 00 CE 85 D7 BD 81
0040 9A B6 20 01 B7 00 00 BD
0048 80 6A B6 20 02 B7 00 00
0050 81 09 23 03 7E 80 5D CE
0058 85 CA BD 81 9A B6 00 00
0060 BD 80 6A 7E 83 00 FF FF
0068 FF FF B6 00 00 81 00 27
0070 03 7E 80 7E CE 85 CF B7
0078 00 00 BD 81 9A 39 81 31
0080 23 16 CE 85 C0 B7 00 00
0088 BD 81 9A B6 00 00 80 32
0090 81 00 27 03 7E 80 DE 39
0098 81 27 23 11 CE 85 B7 B7
00A0 00 00 BD 81 9A B6 00 00
00A8 80 28 7E 80 90 81 1D 23
00B0 11 CE 85 AE B7 00 00 BD
00B8 81 9A B6 00 00 80 1E 7E
00C0 80 90 81 13 23 11 CE 85
00C8 A2 B7 00 00 BD 81 9A B6
00D0 00 00 80 14 7E 80 90 81
00D8 09 23 03 7E 81 34 81 09
00E0 27 28 81 08 27 1E 81 07
00E8 27 26 81 06 27 28 81 05
00F0 27 2A 81 04 27 2C 81 03
00F8 27 2E 81 02 27 30 CE 85
0100 00 7E 81 9A CE 85 32 7E
0108 81 9A CE 85 38 7E 81 9A
0110 CE 85 29 7E 81 9A CE 85
0118 21 7E 81 9A CE 85 1A 7E
0120 81 9A CE 85 14 7E 81 9A
0128 CE 85 0D 7E 81 9A CE 85
0130 07 7E 81 9A 81 0A 27 26
0138 81 0B 27 28 81 0C 27 2A
0140 81 0D 27 2C 81 0E 27 2E
0148 81 0F 27 30 81 10 27 32
0150 81 11 27 34 81 12 27 36
0158 CE 85 97 7E 81 9A CE 85
0160 3F 7E 81 9A CE 85 46 7E
0168 81 9A CE 85 50 7E 81 9A
0170 CE 85 5A 7E 81 9A CE 85
0178 64 7E 81 9A CE 85 6D 7E
0180 81 9A CE 85 78 7E 81 9A
0188 CE 85 83 7E 81 9A CE 85
0190 8F 7E 81 9A FF FF FF FF
0198 FF FF 86 40 B7 40 04 A6
01A0 00 81 FF 26 01 39 A6 00
01A8 F6 40 04 C4 80 27 03 7E
01B0 81 A8 08 B7 40 04 7E 81
01B8 9A FF FF FF FF FF FF FF
0300 F6 40 06 C4 01 27 F9 B6
0308 20 00 B6 20 01 B7 00 00
0310 84 80 27 25 B6 00 00 80
0318 80 B7 00 00 81 0C 27 2E
0320 81 05 23 2A CE 86 10 BD
0328 81 9A CE 85 D7 BD 81 9A
0330 B6 00 00 BD 80 6A 7E 80
0338 4A CE 85 F0 BD 81 9A CE
0340 85 D7 BD 81 9A B6 00 00
0348 BD 80 6A 7E 80 4A B7 00
0350 00 CE 85 FE BD 81 9A CE
0358 85 D7 BD 81 9A B6 00 00
0360 BD 80 6A 7E 80 4A FF FF
0400 CE 86 B0 BD 81 9A F6 40
0408 06 C4 80 26 33 F6 40 06
0410 C4 01 27 F2 7F 00 06 CE
0418 86 C1 BD 81 9A 4F F6 40
0420 06 C4 01 26 29 F6 40 06
0428 C4 80 27 F2 81 0B 23 03
0430 7E 84 1D 4C B7 00 00 BD
0438 80 6A B6 00 00 7E 84 1E
0440 CE 86 B8 BD 81 9A 86 80
0448 B7 00 06 7E 84 17 B6 00
0450 00 F6 00 06 1B B7 20 01
0458 CE 86 CE BD 81 9A 4F B7
0460 00 00 BD 80 6A F6 40 06
0468 C4 01 27 03 7E 84 85 F6
0470 40 06 C4 80 27 EF B6 00
0478 00 81 28 23 03 7E 84 5E
0480 8B 0A 7E 84 5F B6 00 00
0488 B7 00 05 4F B7 00 00 BD
0490 80 6A F6 40 06 C4 01 27
0498 03 7E 84 B1 F6 40 06 C4
04A0 80 27 EF B6 00 00 81 08
04A8 23 03 7E 84 8B 4C 7E 84
04B0 8C F6 00 05 B6 00 00 1B
04B8 B7 20 02 86 0A B7 20 11
04C0 7E 83 07 FF FF FF FF FF
04C8 FF FF FF FF FF FF FF FF
04D0 FF FF FF FF FF FF FF FF
04D8 FF FF FF FF FF FF FF FF
04E0 FF FF FF FF FF FF FF FF
04E8 FF FF FF FF FF FF FF FF
04F0 FF FF FF FF FF FF FF FF
04F8 FF FF FF FF FF FF FF FF
0500 2E 0F 0B 04 04 04 FF 0D
0508 1F 04 04 04 FF 1D 0E 13
0510 04 04 04 FF 28 3A 04 04
0518 04 FF 28 06 28 04 04 04
0520 FF 37 0C 08 37 04 04 04
0528 FF 37 07 23 0F 0B 04 04
0530 04 FF 14 11 04 04 04 FF
0538 38 06 0B 04 04 04 FF 0D
0540 07 0B 04 04 04 FF 13 2D
0548 07 28 0B 04 04 04 FF FF
0550 0D 2E 07 3E 28 04 04 04
0558 FF FF 1D 33 0D 13 0B 04
0560 04 04 FF FF 28 3A 0D 13
0568 0B 04 04 04 FF 28 0C 28
0570 0D 13 0B 04 04 04 FF FF
0578 37 0C 08 37 0D 13 0B 04
0580 04 04 FF 37 07 23 07 0B
0588 0D 13 0B 04 04 04 FF 14
0590 0D 13 0B 04 04 04 FF 38
0598 06 0B 0D 13 0B 04 04 04
05A0 FF FF 0D 30 07 0B 0D 13
05A8 00 04 04 04 FF FF 1D 33
05B0 0D 13 00 04 04 04 FF 28
05B8 3A 0D 13 00 04 04 04 FF
05C0 28 0C 28 0D 13 00 04 04
05C8 04 FF 35 04 04 04 FF 2A
05D0 2D 18 02 29 04 04 FF 00
05D8 12 13 04 0D 06 10 04 0C
05E0 0C 37 04 04 04 FF FF FF
05E8 FF FF FF FF FF FF FF FF
05F0 24 1F 15 00 04 10 3A 0B
05F8 0C 2C 04 04 04 FF 24 1F
0600 00 15 04 3B 28 11 33 0B
0608 1F 0B 04 04 04 FF FF FF
0610 24 1F 00 15 04 13 28 0B
0618 0C 2C 04 04 04 FF FF FF
06B0 14 04 07 10 04 04 04 FF
06B8 09 13 04 07 10 04 04 04
06C0 FF 37 07 11 04 20 33 37
06C8 04 04 04 FF FF FF 37 07
06D0 0D 04 10 0C 0B 0C 01 0D
06D8 02 37 04 04 04 04 FF FF
07F8 FF FF FF FF FF FF 80 00
```

Fig.5. *Hex dump of the speaking clock program*

The ICM7170 was not specifically designed for the 6800 series of microprocessors. The device has both active-low read and write lines. IC6B to IC6D gate the processor's E line with the r/w line to provide IC5 with the required RD and WR signals. A simple inverter cannot be used between the two lines as this would not take into account the r/w timing cycle. In order to synchronise the r/w signals with the processors clock, 02 of the clock cycle ('E' line) is also gated with the r/w line.

PROGRAM OPERATION

After switching on you need to set up the time. By pressing 'SET TIME' the clock will say "AM". Pressing 'INC' sets to pm, or pressing 'ENTER' causes it to say "SET HOURS". By pressing 'INC' the hours are spoken in turn (1 to 12). When the required hour has been spoken pressing 'ENTER' stores this value in the hours register. It then says "SET MINUTES". Pressing 'INC' then advances the tens of minutes (10, 20, 30, 40, 50). When the required value has been spoken pressing 'ENTER' stores this value in a location in ram. Pressing 'INC' now advances the units (1 to 9). Pressing 'ENTER' stores this value in a second location in ram. Adding these two locations together gives the total minutes value which is then stored in the minutes register. The program now jumps to a routine which looks to see if the 'SPEAK' key has been pressed.

On pressing the 'SPEAK' key the 100ths second register is read. This latches the time data to prevent it rolling when reading the various time data registers. The hours register is read and the program jumps to another routine which determines the hour value. Having done this it now jumps to the next routine which sends the required word data to the speech synthesiser. Now the clock has to speak the minutes. To explain this it is best to look at an example.

Suppose the minutes value is 46, the routine determines the group value, which in this case lies between 40 and 49, and so "forty" is sent to be spoken. To obtain the unit value (six) forty is subtracted and "six" is sent to be spoken. Fig.4 shows the program flow.

A routine also sorts out the grammar. If the minute value is less than ten "OH" has to be spoken before the units value and if zero it has to say "O'CLOCK". *(Was Victor Borg consulted about the correct pronunciation of the apostrophe? Ed.)*

If the time lies between 1:00am to 11:59am it will say "GOODMORNING THE TIME IS.....". Between the times of 12;00noon to 5:00pm it says "GOOD-AFTERNOON THE TIME IS....." and from 6:00pm onwards it says "GOOD-EVENING THE TIME IS.....".

The machine code listing is shown in Fig.5 and occupies approximately 1K of code. If any readers might like to enhance or write their own programs, the addresses in Table 3 may be of use.

	REGISTERS	ADDRESS
REALTIME CLOCK	1/100THS	2000H
	HOURS	2001H
	MINUTES	2002H
	SECONDS	2003H
	MONTHS	2004H
	DATE	2005H
	YEAR	2006H
	DAYOFWEEK	2007H
	COMMAND REGISTER	2011H
SPEECH SYNTHESISER		4004H
FUNCTION SWITCHES		4006H

Table 3

FINDING THE STORED PROGRAM

When the processor is reset it looks at the reset vector addresses FFFEH and FFFFH and reads the data at each location. These two bytes are loaded into the processor's program counter and is used as a start location from which to run. As the eprom sits at memory location 8000H this must be the address held in the program counter.

On looking at the hex listing we see that the reset vector address is stored in locations 07FEH and 07FFH and not the required addresses FFFEH and FFFFH! This is easily explained. As the eprom is only 2K * 8 (16K) address lines A0 to A10 are all that are needed to access any location within it. Therefore the bit values A11 to A14 are not seen (A15 selects the eprom). When FFFEH is sent out by the processor A11 to A14 are 'lost' and the processor sees the contents of address 07FEH instead. This is of no consequence as long as the processor obtains the correct start location.

CONSTRUCTION

A suggested power supply is shown in Fig.6.

The majority of components fit onto a double sided pcb, the layout of which is shown in Fig.7. It is recommended that sockets are used for all ic positions, especially IC3 which may be required to be replaced with different versions of software. Also note the capacitor polarities around IC8. A soldering iron with a small bit is required for construction as many tracks run extremely close together and are very easily bridged by solder. Before fitting any components to the pcb make all the through hole connections using fine single strand wire. *(Better still, use pc link pins. Ed.)* Care should be taken to ensure all these connections are made as there are quite a few, many being under components. Construction can now continue in the usual manner.

The unit was housed in a Verobox type 75-1411D. The back panel is used as a chassis onto which the transformer is bolted. It also serves as a heatsink for the regulator (IC9). The mains lead passes through a grommet and is secured by a 'P' clip fixed to the base of the box. The earth wire is soldered to a tag bolted to the transformer fixings. A few 3mm holes are drilled in the base under the transformer to aid ventilation.

The pcb is fixed to the base by spacers. Make sure that the far side of the pcb sits flush against the back panel so that the regulator can be bolted to it. Terminal pins should be used where wires connect to the board. This makes the board easier to remove, and looks neater.

The speaker is attached using four M4*10 csk screws. The speaker should be as large as possible (at least 3in) to provide a good frequency response. Rub on transfer letters can be used to label switch functions and lacquer applied to fix them.

TESTING

After checking the board construction and wiring connect a voltmeter (set to 10V range) across C19. The rough dc should be around 7.5V (minimum). Check from 5V on the output of IC9. Press 'SETTIME'. It should respond by saying "A M". If nothing happens recheck your wiring. If you programmed your own eprom check it for errors. Also make sure that the rom is being addressed (selected) by monitoring IC3s chip-select pin, which should be low. (A15 should be constantly high).

Adjust C5 to obtain an accurate clock time. By trial and error an accuracy of few seconds a day is possible. An alternative method is to write a piece of software which causes the realtime clock to produce an interrupt every second. The interrupt signal can then be monitored with a frequency counter (negative side triggered) and C5 adjusted for a reading of 1.0000000 on the display. The first method is accurate enough although more time consuming.

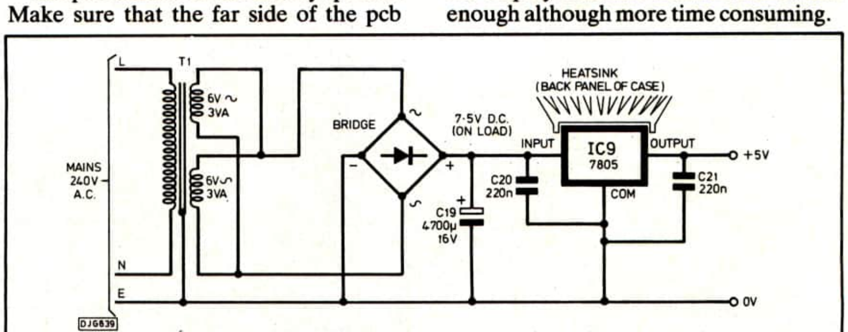


Fig.6. *Power supply circuit diagram*

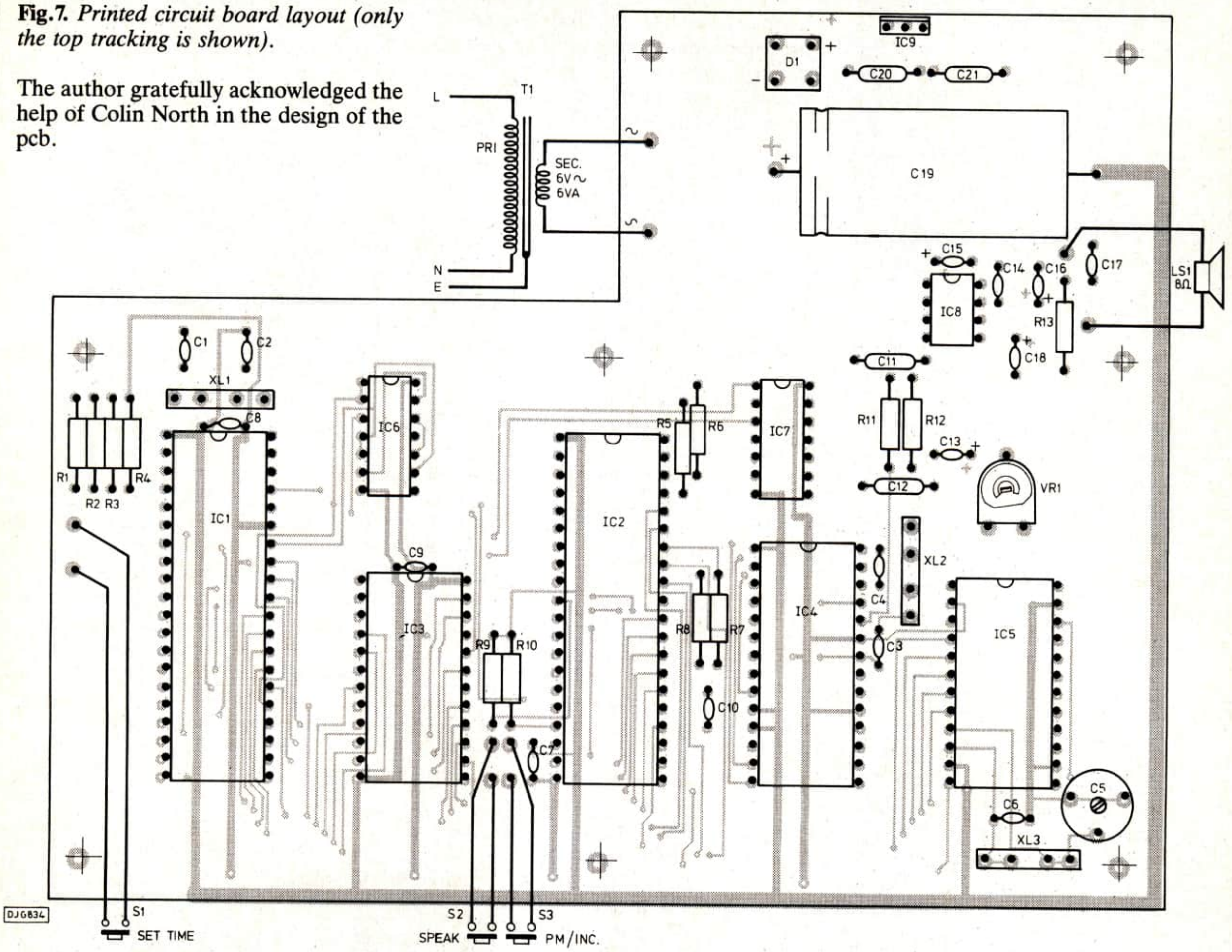


Fig.7. *Printed circuit board layout (only the top tracking is shown).*

The author gratefully acknowledged the help of Colin North in the design of the pcb.

COMPONENTS

RESISTORS

R1–R7	10k (8 off)
R9,R10	1k (2 off)
R8	100k
R12	33k
R13	10Ω

All resistors ⅓W 5%.

CAPACITORS

C1–C4,C6	22pF ceramic (5 off)
C5	10–35pF variable
C7–C10,C14,C17	100nF ceramic (6 off)
C11,C12	22nF polyester (2 off)
C13,C15	10µF 16V elect (2 off)
C16,C18	100µF 16V elect (2 off)
C19	4700µF 16V elect
C20,C21	220nF polyester (2 off)

SEMICONDUCTORS

D1	Bridge rectifier 2A 50V
IC1	6802
IC2	6821
IC3	2716 (5V)
IC4	SPO256AL
IC5	ICM7170
IC6	74LS00
IC7	74LS10
IC8	LM386 (6V version)
IC9	7805

POTENTIOMETER

VR1	4k7 horiz preset

CRYSTALS

XL1	4MHz
XL2	3 to 3.5MHz (see text)
XL3	(see text)

MISCELLANEOUS

T1	0–6V, 0–6V transformer 3VA per winding
S1,S2,S3	SP push make switches (3 off)
Case	Vero type 75-1411D (20.5 × 14 × 7.5cm)
LS1	8 ohm 3in speaker

Pcb, sockets, wire etc.

CONSTRUCTOR'S NOTE

One of Stephen's friends has very kindly offered to program any blank eproms sent to him in connection with this project. Send your eprom with a stamped addressed envelope (or equivalent return postage) to:

Charles Moore,
101 Heatherstone Avenue,
Dibden Purlieu, Hythe,
Hants. SO4 5LE.

(please state value of crystal being used in the real-time clock).

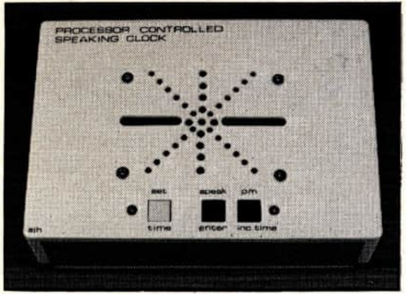
CONSTRUCTOR'S NOTE

The double sided printed circuit board is available through the PE PCB Service, see page 60.

STEPHEN HUNT

While preparing Stephen's article for publication, we were distressed to learn of his tragic death. It is his parent's wishes that we should still publish his article. We send our deepest sympathy to his family. Ed.

If you are unable to set up the clock fairly accurately then this is probably due to the crystal not being loaded correctly. Inserting the specified load capacitor (C6) for the crystal should cure the problem. Finally set VR1 to the volume required.



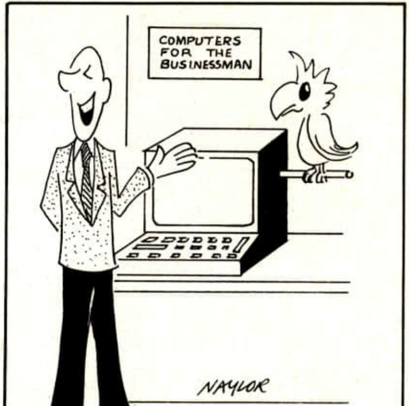
STATIC SENSITIVE

During bench testing of the prototype, it was discovered that the realtime clock's interrupt pin was prone to static charges. When "zapped", the chip latches up and draws excessive current.

This is due to the fact that the interrupt line does not have the normal protection diodes found in most cmos devices. As long as the relevent pcb tracks are not touched by "aerials" such as screwdrivers etc, no harm should result.

It was felt that adding components to suppress transients could prove to be more trouble than it's worth and as these components are not integrated on the silicon itself would only provide limited protection. However, once the board is housed in an enclosure the chance of latch up is reduced to zero. The prototype has been working faultlessly for many months.

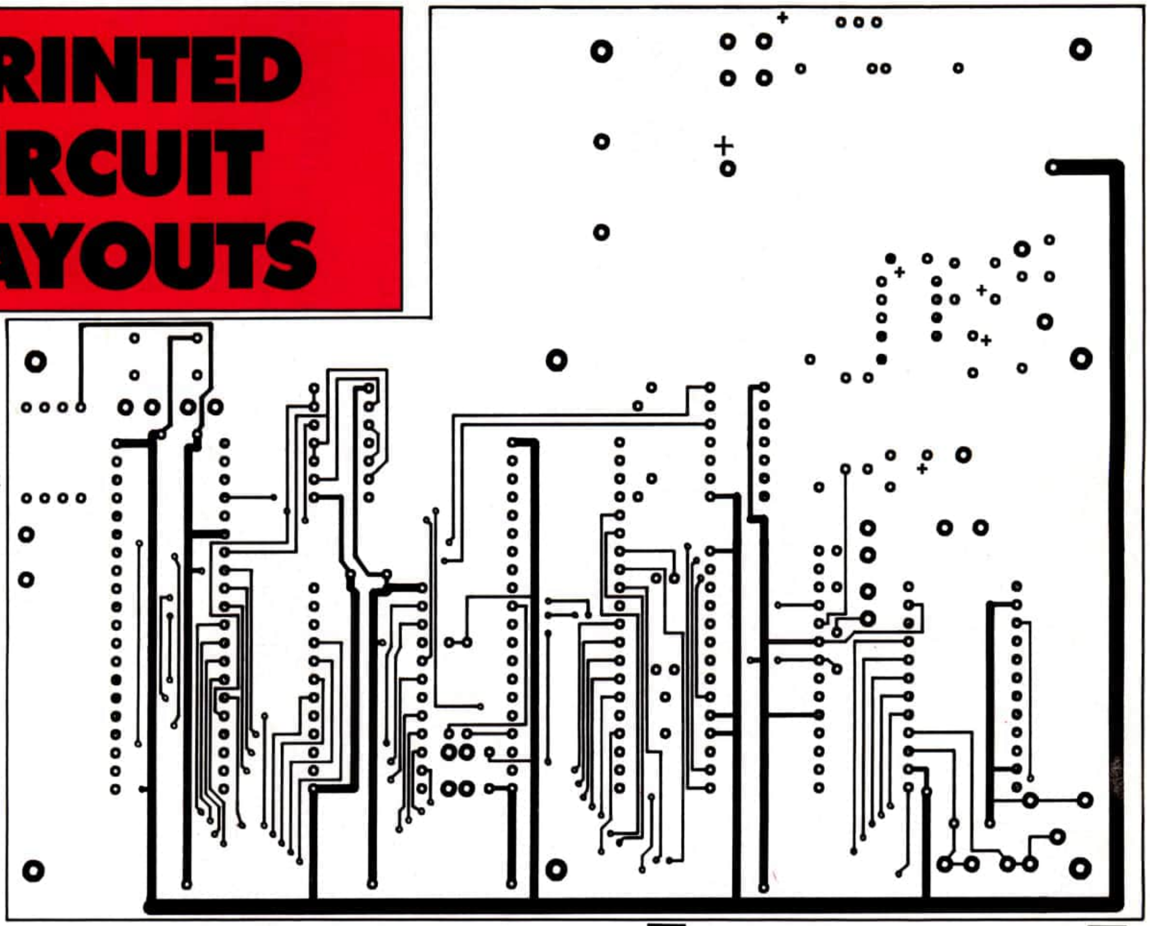
Please take care when bench testing! PE



"....And this model has a vocal output!"

PRINTED CIRCUIT LAYOUTS

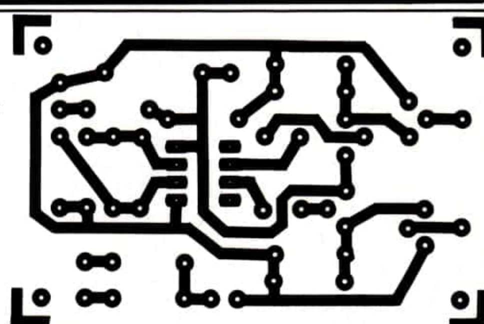
Speaking clocking component side tracking

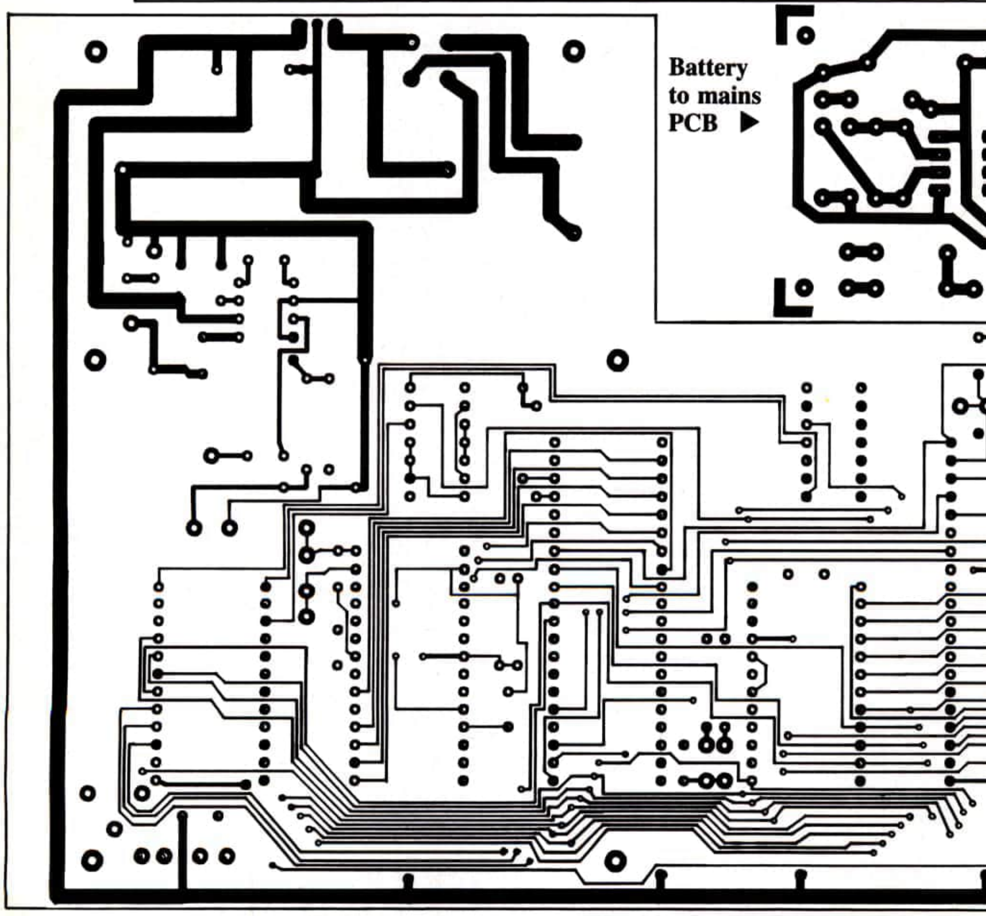


PE176 ▶

▼

Battery to mains PCB ▶





Speaking clock underside tracking