In an earlier article ('Chattering chips', Elektor September 1981), several speech synthesis systems were discussed. For various reasons, the Texas Instruments 'Solid State Speech' system seemed the best bet — certainly for microprocessor enthusiasts. In the first place, it can produce an output that is something like a human voice coming over a telephone line: not hi-fi, admittedly, but good enough to notice traces of an american accent coming through! Furthermore, the coding system used is fairly 'logical', which means that it is quite feasible to work out codes for new words — without having to resort to a huge computer.

talking board

a solid-state voice

In the early days of science fiction, robots could walk and talk like human beings. Later on, as authors learned of the possibilities and limitations of computers, it became more realistic to reserve the power of speech for huge, 'space-ship filling' electronic brains. Now, in this project, we can proceed to science fact: a single board that can provide a vocabulary of several hundred words for a microprocessor system!

Having decided to use the Texas Instruments system, the next step is to make a choice between the two versions: the older TMS 5100, intended for talking games and the like, or the new TMS 5200 that is intended for use in microprocessor systems. Surprisingly enough, we decided to use the 5100, for two good reasons: there is a much larger vocabulary available for this chip, as well as a good circuit in the TI application note! With only a few further modifications and additions, this system can be interfaced to almost any microprocessor system.

The basic principle of the actual speech synthesis process will be discussed later. For the moment, the only important thing to know is that a serial bit stream must be fed into the 'VSP' (Voice Synthesis Processor) in order to make it talk. For the word 'help', say, a total of 534 bits are required: just less than 67 bytes. Since this is a fairly short word, it will be obvious that a considerable memory range is required for a total vocabulary of several hundred words. To avoid wasting memory range in the 'host' microprocessor system, the 'speech memory' is included on the speech board — complete with a local address counter and associated control circuits.

The block diagram of the 'talking board' is given in figure 1. The lower half of this diagram shows the memory and control circuits. Initially, the first address for a given word must be loaded into the address buffer/counter. Since 16-bit addressing is used, the first address is loaded in two bytes (8 bit): first the low byte is placed on the data bus and $\overline{\text{LDA 1}}$ is toggled briefly, after which the high byte is loaded by pulsing $\overline{\text{LDA 0}}$. The 'bit counter' is reset when $\overline{\text{LDA 1}}$ is toggled.

Once the first address is loaded, the unit can be given the 'talk' command. Each I/O clock pulse from the VSP increments the bit counter, causing the 'parallel-to-serial bit stream converter' to select the next bit in the selected speech memory byte. The same I/O pulse clocks each bit in turn into a flip-flop, which passes the bit stream to the speech processor. When the bit counter has scanned all eight bits, it increments the address buffer/counter to select the next memory byte.

As illustrated in the block diagram, the connection between the bit stream converter and the following flip-flop can be interrupted, and both sides brought out to the 'host' processor. Data from the speech memory can be read into the host's RAM area via the Y output; after modification, to obtain a new word or sentence, it can be fed back in via the D input. Admittedly, this will often require a little interface — but we intend to publish a suitable circuit in the near future.

The upper part of the block diagram shows the word processor proper (the 'VSP'). Two control inputs, CØ and C1, come in at the left. These give the commands 'reset', 'talk' and 'test busy' as shown in table 1. The test busy command refers to the $\overline{\text{busy}}$ output: when enabled, this goes high at the end of a speech sequence.

The VSP chip contains a clock oscillator — among other things, this determines the pitch of the spoken output. To synchronise the external CCLK (control clock) input to this on-chip clock, the two signals are fed through a flip-flop. The result goes back into the PDC (processor data clock) input. The VSP indicates that it needs the next speech data bit by toggling its I/O output; as described earlier, this clocks the next bit into the flip-flop and updates the bit counter. When entering speech data from external RAM, the $\overline{\text{I/O}}$ output must be used for correct synchronisation. Finally, the two differential speech outputs are passed through a low-pass filter and power amplifier to the loudspeaker.

Timing

Obviously, the various control signals must be applied to the board in the correct sequence. This is illustrated in figure 2. After power-up, the circuit

Table 1

CØ, C1:

Command	CØ	C1
reset	1	1
talk	0	1
test busy	0	0
(invalid)	1	0

$\overline{\text{Busy}}$: When enabled , goes high at end of word.
CCLK: Control clock for word processor
$\overline{\text{LDA1}}$, $\overline{\text{LDAØ}}$: Enable input of low address byte and high address byte, respectively, on DØ ... D7
$\overline{\text{I/O}}$,D,Y: Control lines for external speech memory

Table 1. The three commands which are initiated via the control inputs CØ and C1.

1

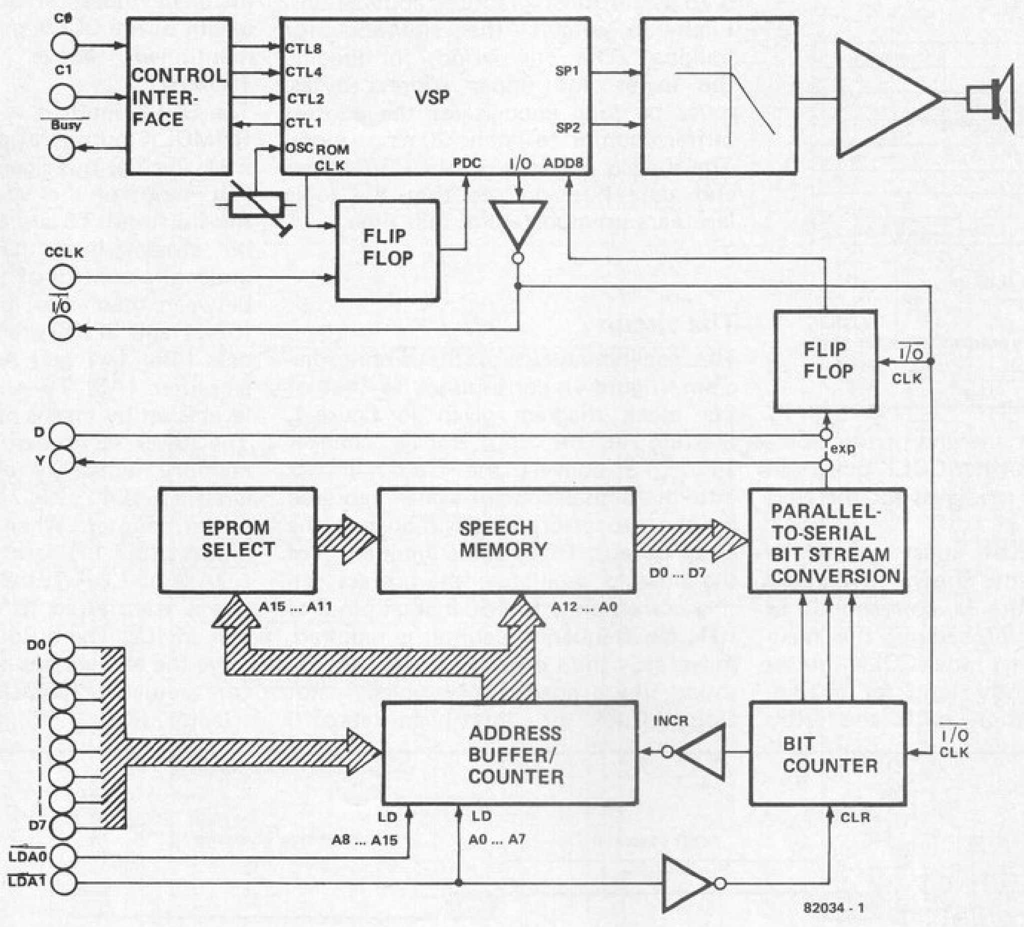

Figure 1. The block diagram of the talking board.

2

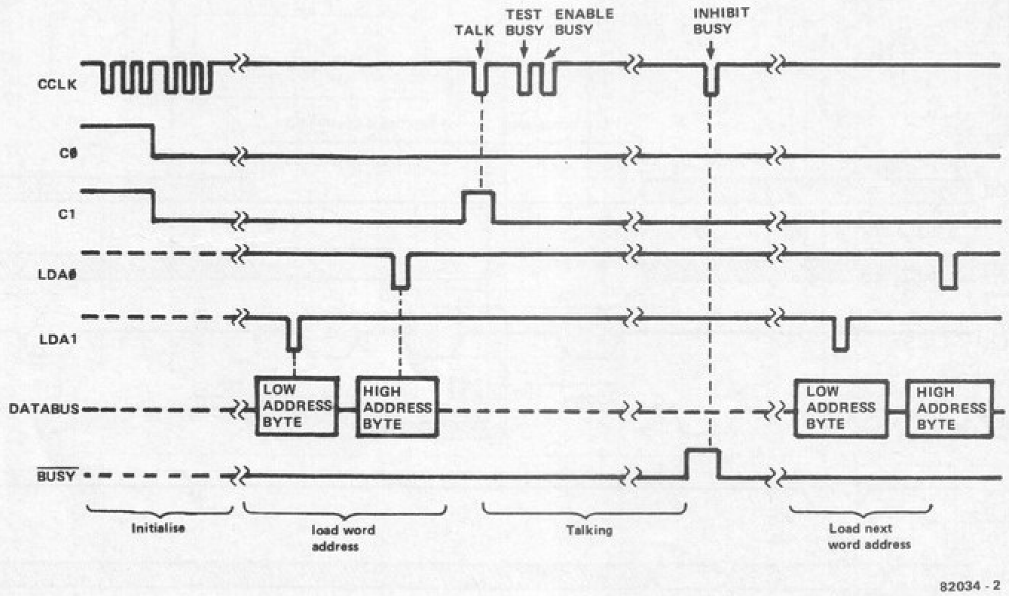

Figure 2. The various control signals must be applied to the talking board in the correct order.

must be initialised. This is done by applying a logic 1 level to CØ and C1 (corresponding to 'reset') and toggling the CCLK input three times; then, CØ and C1 are set to logic Ø (test busy) and CCLK is toggled a further three times. The unit is now 'ready to go'.

To output a word, the low address byte is put onto the data bus and $\overline{\text{LDA 1}}$ is pulled low briefly; then the high address byte is loaded from the data bus by toggling $\overline{\text{LDA Ø}}$. C1 is now set to logic 1 (CØ remains low), corresponding to the 'talk' command, and the CCLK input is toggled. This initiates the speech output. Meanwhile, C1 is returned to logic Ø and the CCLK input is toggled twice. This enables the 'busy' output, so that

Table 2

	min	max
T_S	0	–
T_{DOWN}	T = 6,25 µS	–
T_{UP}	T = 6,25 µS	–
T_H	1¾T = 10,9 µS	–
T_W	20 ns	–
T_{HO}	0	–
$T_{I/O}$	1¼T = 7,8 µs	8,1 µS

$T = T_{ROMCLK} = 6{,}25\ \mu s$

Table 2. The timing requirements for the various control signals.

it will go high at the end of the word. At that point, a further CCLK pulse will reset the VSP in readiness for the next word.

All control signals must meet the timing requirements shown in figure 3 and table 2. Figure 3a corresponds to the initialisation procedure; the main point here is that the CCLK pulses must be sufficiently long for guaranteed synchronisation with the VSPs 'ROMCLK' oscillator. This means that both T_{down} and T_{up} must be at least 6.25 µs, in most practical applications. Figure 3b shows the situation for 'talking'. The T_W period, for loading the lower and upper address bytes, must be long enough for the address buffer/counter to latch: 20 ns or more. The shaded portions on the CØ/C1 lines and data bus indicate that the logic levels are unimportant at that time.

The circuit

The general layout of the circuit diagram (figure 4) corresponds to that of the block diagram given in figure 1. Starting at the top, for a change: T1 . . . T3 convert the CØ/C1 inputs into the actual control signals required by the processor, and N2 buffers the Busy output. P1 sets the frequency of the on-chip oscillator: the correct setting corresponds to 160 kHz at pin 3 of IC1. No frequency counter is required, however: the output signal should sound like a normal male voice — not Donald Duck or 'infra-Iwan-Rebroff'! Normally, the mid-position of P1 should be fairly accurate. Note that this adjustment does effect the minimum length of the CCLK pulses — the 6.25 µs mentioned above corresponds to 160 kHz!

The CCLK input is synchronised to the ROMCLK output at pin 3 by means of FF1; via T4, this signal goes back to the PDC input of the VSP, IC1. The other flip-flop and T5 are used to clock the bit stream into the ADD8 of IC1, under the control of the I/O output. In between these two, the speech outputs (SPK1 and SPK2) are passed to the low-pass filter (A1 and A2) and the power amplifier (A3, T6 . . . T9). The output level is set by means of P2.

The lower section of the circuit is the memory with its associated control circuits. IC4 . . . IC7 are the address buffer/counter. When the parallel load inputs (pin 11) are pulled low, via $\overline{LDA\ Ø}$ or $\overline{LDA\ 1}$, the byte on the data bus is transferred to the corresponding pair of ICs. The outputs from these ICs drive the address inputs of IC12 . . . IC19 (the actual EPROMs) and the EPROM selector, IC9.

3

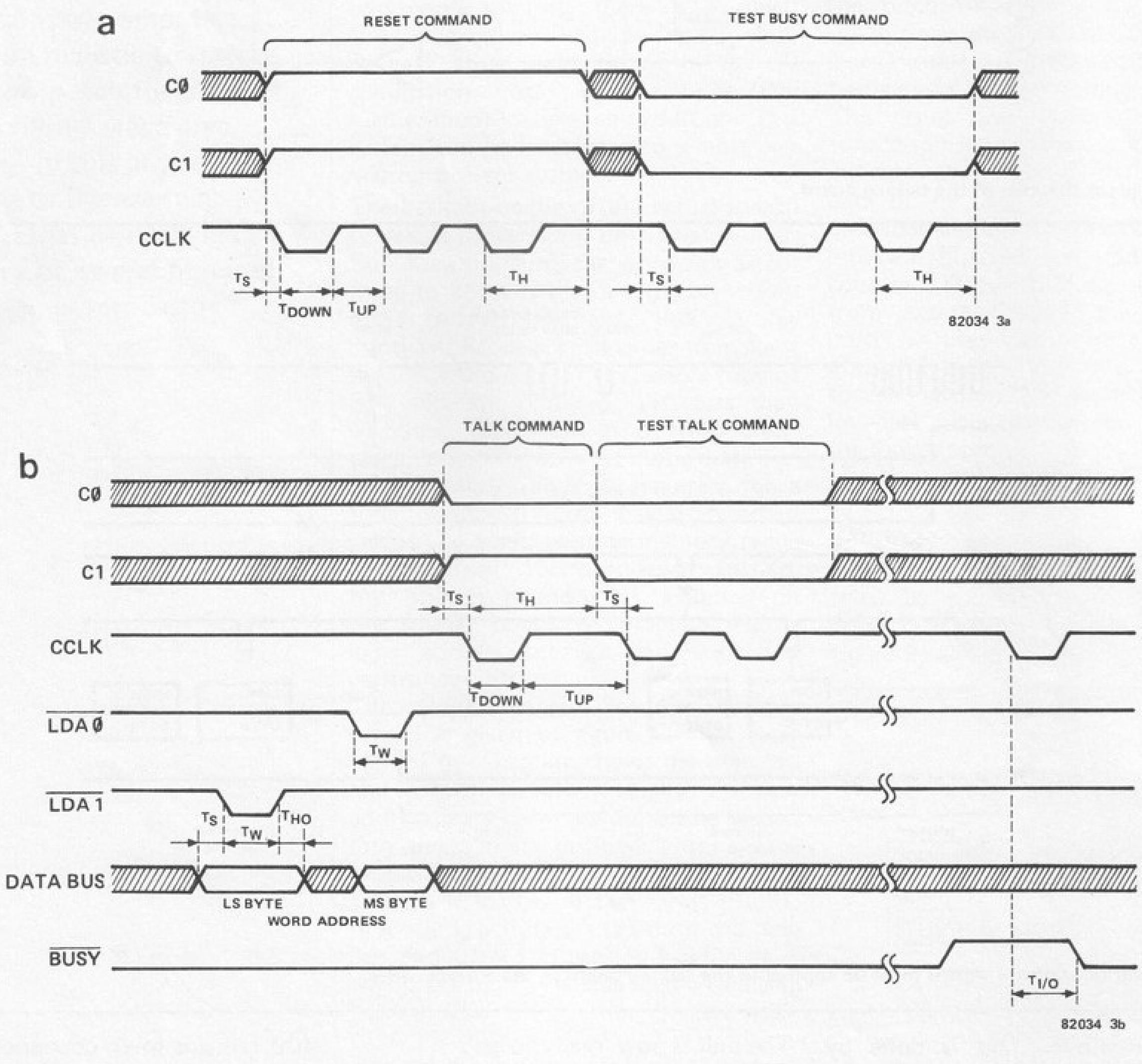

Figure 3. The control signals must meet certain timing requirements. Figure 3a shows the duration of the signals during the initialisation procedure while figure 3b illustrates the situation when the board is 'talking'.

4

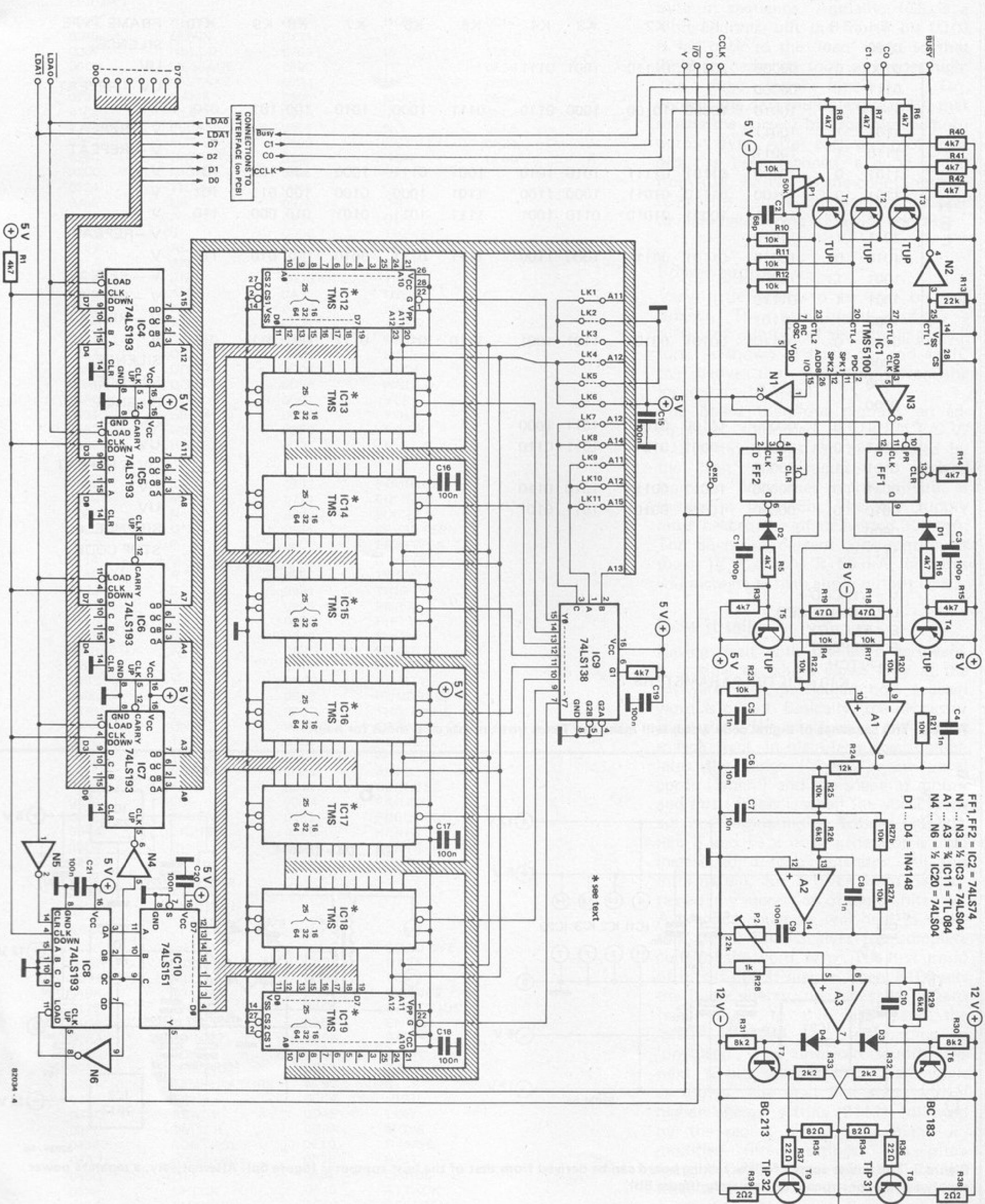

Figure 4. The complete circuit diagram of the talking board. The layout corresponds quite closely to that of the block diagram.

Table 3

	E	R	P	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	FRAME TYPE
	0000													SILENCE
HEL	0100	0	00000	10011	01110	1001	0111							UV
HEL	0111	1	00000											UV – REPEAT
HEL	1101	0	10010	10000	10100	1000	0110	0111	1000	1010	100	101	010	V
HEL	1101	1	10011											V – REPEAT
HEL	1110	1	10011											V – REPEAT
HEL	1101	0	10100	01101	01111	1010	1010	1001	0111	1000	100	101	101	V
HEL	1101	0	10100	01110	01011	1000	1100	1101	1000	0100	100	011	101	V
HEL	1101	0	10011	10001	01010	0110	1001	1111	1011	0101	010	000	110	V
HEL	1011	1	11010											V –REPEAT
HEL	1010	0	10010	01101	00111	1000	1100	1111	0111	0010	001	010	110	V
HEL	1001	1	10001											V – REPEAT
HEL	1001	1	01110											V – REPEAT
HEL	1000	1	01101											V – REPEAT
HEL	0010	0	01110	00101	00101	1101	1001	1110	0101	0111	001	011	011	V
	0000													SILENCE
	0000													SILENCE
	0000													SILENCE
P	0111	0	00000	10100	01011	1011	1000							UV
P	0111	0	00000	10001	01011	1011	0110							UV
P	0101	1	00000											UV – REPEAT
P	0011	0	00000	10011	00111	1010	0110							UV
P	0010	0	00000	10010	00101	1011	0101							UV
	0000													SILENCE
	1111													STOP CODE

V = VOICED
UV = UNVOICED
E = ENERGY
R = REPEAT
P = PITCH
K1 . . . K10 = FILTER PARAMETERS

Table 3 This sequence of digital code words will make the Texas Instruments chip shout for help!

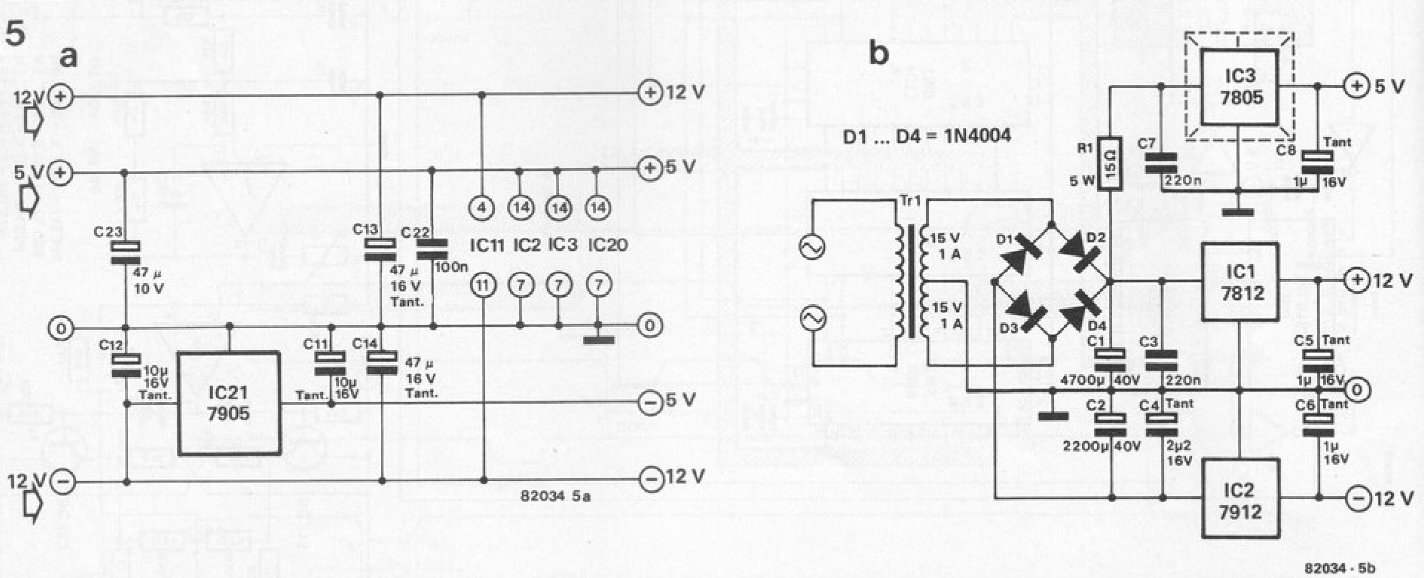

Figure 5. The power supply fot the talking board can be derived from that of the host computer (figure 5a). Alternatively, a separate power supply can be constructed quite simply (figure 5b).

The confusing array of wire links are included so that different types of EPROM can be used. For 2716s, links 2, 6, 7 and 9 should be used; the EPROMs are then addressed in the following sequence: IC12, IC13, IC16, IC17, IC14, IC15, IC18, IC19 — corresponding to the address range from ØØØØ to 3FFF in 2 kbyte chunks. For 2732s, as in the Talking Board kit, links 1, 6, 8 and 10 should be mounted, as shown. The EPROMs are now selected in sequence, from IC12 to IC19, to cover the address range from ØØØØ to 7FFF. Finally, links 1, 4, 8 and 11 are provided for 2764s; these cover the complete address range from ØØØØ to FFFF in the following sequence: IC12, IC14, IC16, IC18, IC13, IC15, IC17, IC19. It should be noted that the board layout and pinning is given for 2764s; the other types are slightly shorter, as indicated by dotted lines on the board. This means that pin 1 of a 2716 or 2732 is inserted in the pin 3 position, and so on down.

Finally, the lower right-hand corner of

Table 4.

ADDRESS (HEX)	WORD	ADDRESS (HEX)	WORD	ADDRESS (HEX)	WORD
EPROM 1					
0000	AGAIN	06DA	HOW	0D6C	B
0048	DOWN	0724	IN	0DA8	C
0084	HELLO	0760	IS	0DEC	D
00D0	MESSAGE	079C	IT	0E36	E
0138	MISTAKE	07B4	ME	0E60	F
0198	NAME	0800	MUCH	0E94	G
01CE	NEED	082A	MY	0EC4	H
0222	PLEASE	0856	NO	0EFE	I
0262	PUT	0890	NOT	0F34	J
028C	REPEAT	08C4	NOW	0F80	K
02CC	RIGHT	0906	OF		
0324	THANK	0946	ON		
036E	UP	0970	OR		
0388	WANT	099A	OUT		
03CE	'S	09D6	THE (E)		
03E4	ALL	0A08	THE		
041A	AN	0A44	THERE		
0446	AND	0A78	THIS		
0484	ANY	0A9E	USE		
04B4	ARE	0AF6	WHAT		
04D0	AT	0B20	WHEN		
04F2	CAN	0B6C	WHERE		
0522	DID	0BB4	WILL		
0566	DO	0C06	WITH		
05A0	DOES	0C5A	WOULD		
05FC	FOR	0C94	YES		
0634	FROM	0CC6	YOU		
0662	GOT	0CF8	YOUR		
069C	HAVE	0D2E	A		
EPROM 2					
0000	L	0732	FOURTEEN		
004C	M	0774	FIFTEEN		
008A	N	0800	SIXTEEN		
00C2	O	0864	SEVENTEEN		
00EA	P	08C2	EIGHTEEN		
0114	Q	08FC	NINETEEN		
014C	R	0952	TWENTY		
0178	S	0986	THIRTY		
01A0	T	09B6	FORTY		
01F0	U	09EC	FIFTY		
021E	V	0A46	SIXTY		
0250	W	0A7E	SEVENTY		
0298	X	0AC4	EIGHTY		
02BE	Y	0AF0	NINETY		
0300	ZED	0B58	HUNDRED		
0346	ZERO	0BC2	THOUSAND		
03A4	ONE	0C3E	EQUAL		
03F6	TWO	0C94	NUMBER		
0430	THREE	0D04	PERCENT		
0474	FOUR	0D54	AMPS		
04C2	FIVE	0D94	DEGREES		
0510	SIX	0DF4	FARAD		
054E	SEVEN	0E62	FREQUENCY		
05A4	EIGHT	0ECE	HENRY		
05D4	NINE	0F18	HERTZ		
061E	TEN	0F66	HOURS		
0652	ELEVEN				
069A	TWELVE				
06F6	THIRTEEN				
EPROM 3					
0000	MEGA	09DE	GOOD BYE		
005E	MICRO	0A36	DATE		
00D2	MILLI	0A6A	LEFT		
010A	MINUS	0A98	CHANGE		
0172	OHMS	0ADE	DIRECTION		
01C6	PLUS	0B5C	ENTER		
01FA	POINT	0BA6	FAST		
023C	POWER	0C00	SLOW		
0282	SECONDS	0C48	GO		
02EA	TEMPERATURE	0C9C	STOP		
0362	TIME	0CEA	HIGH		
03AC	READY	0D46	LOW		
03F0	SWITCH	0D9A	MOVE		
043E	CONTROL	0E10	RANGE		
04A2	WARNING	0E7E	EXIT		
04EC	OFF	0EBE	CARDS		
0530	CHECK	0F10	ATTACK		
0566	BUTTON	0F4C	DESTROY		
05B6	TELEPHONE				
0608	BUSY				
0656	INVALID				
06F6	MONDAY				
0752	TUESDAY				
0800	WEDNESDAY				
0872	THURSDAY				
08D2	FRIDAY				
0938	SATURDAY				
0984	SUNDAY				

Table 4. The vocabulary of the talking board is contained in EPROM. Note that the first digit in the word address must correspond to the position of the EPROM. If these are mounted in sequence in the IC12 . . . IC14 position, the first address in EPROM 2 will be 1000; EPROM 3 then starts at 2000.

figure 4. IC8 is the bit counter: the incoming $\overline{I/O}$ (clock) signal is divided by 8, to select the eight bits in each byte in sequence. Actually, IC8 is a 4-bit counter, but the fourth bit (Q_D) is fed back to the 'load' input so that ØØØØ is loaded as soon as it goes high. The three lower bits, $Q_A \ldots Q_C$, control the data multiplexer (IC10) that selects the correct bit from the memory output byte. After each group of eight bits has been scanned, a pulse is fed from IC8, via N4, to the count input of IC7. This causes the address counter to increment to the following address.

Power supply

Very little needs to be said on this subject. The main board contains a sufficient number of smoothing capacitors, as shown in figure 5a, and an IC that derives the −5 V supply from the incoming −12 V rail.
The board therefore requires an adequately smoothed +12 V/+5 V/0 V/−12 V input. This can be provided by the 'host' microprocessor, or derived from an additional supply circuit as shown in figure 5b. The 5 V supply must be capable of delivering 300 mA. The quiescent current consumption of the ± 12 V supply is 50 mA, but this will increase at high audio output levels.

How it talks

Having dealt with the basic hardware, it is time to take a closer look at the software — in particular, how a given word is coded. Basically, the processor is an electronic analagon of the human speech tract. In plain language, it simulates the lungs ('energy'), the vocal cords ('pitch') and the shape of mouth and lips ('filters'); when the vocal cords are not resonating ('unvoiced' sounds, like S and F) a noise generator is used instead of a tone generator. All this information, for a given word, is contained in a succession of digital bits.
A practical example will help to make this clear. Table 3 gives the complete code for the word 'help'. The first group of bits is ØØØØ: silence. Then, Ø1ØØ sets the initial energy; the repeat bit is zero (we'll come to this later) and the 'pitch' is ØØØØ — corresponding to 'unvoiced'. For unvoiced sounds, the next 18 bits set four filter parameters as shown. The next line starts with a higher 'energy' setting (Ø111), followed by the repeat bit at logic high: unmodified filter settings. The pitch remains ØØØØ, for unvoiced. Since the filter settings remain unchanged, we can proceed to the next line. A higher energy is defined, no repeat, and a non-zero pitch: 1ØØ1Ø, defining the desired tone generator frequency. For voiced sounds, more precise filtering is required. This results in a total of 39 bits to determine the settings of all ten filters. Fortunately, the filter settings can remain unaltered for the next two lines (repeat bit one), although

6

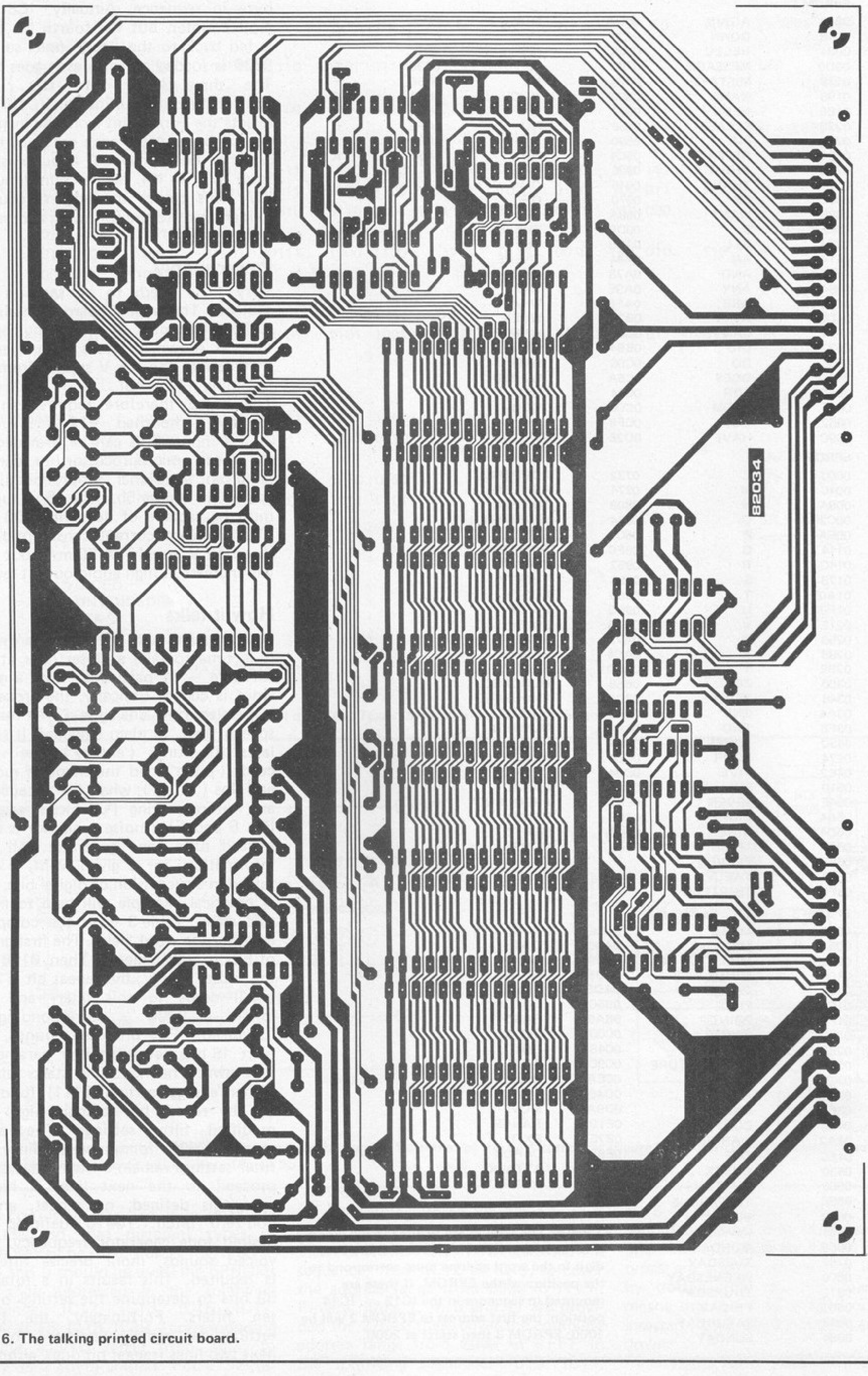

Figure 6. The talking printed circuit board.

7

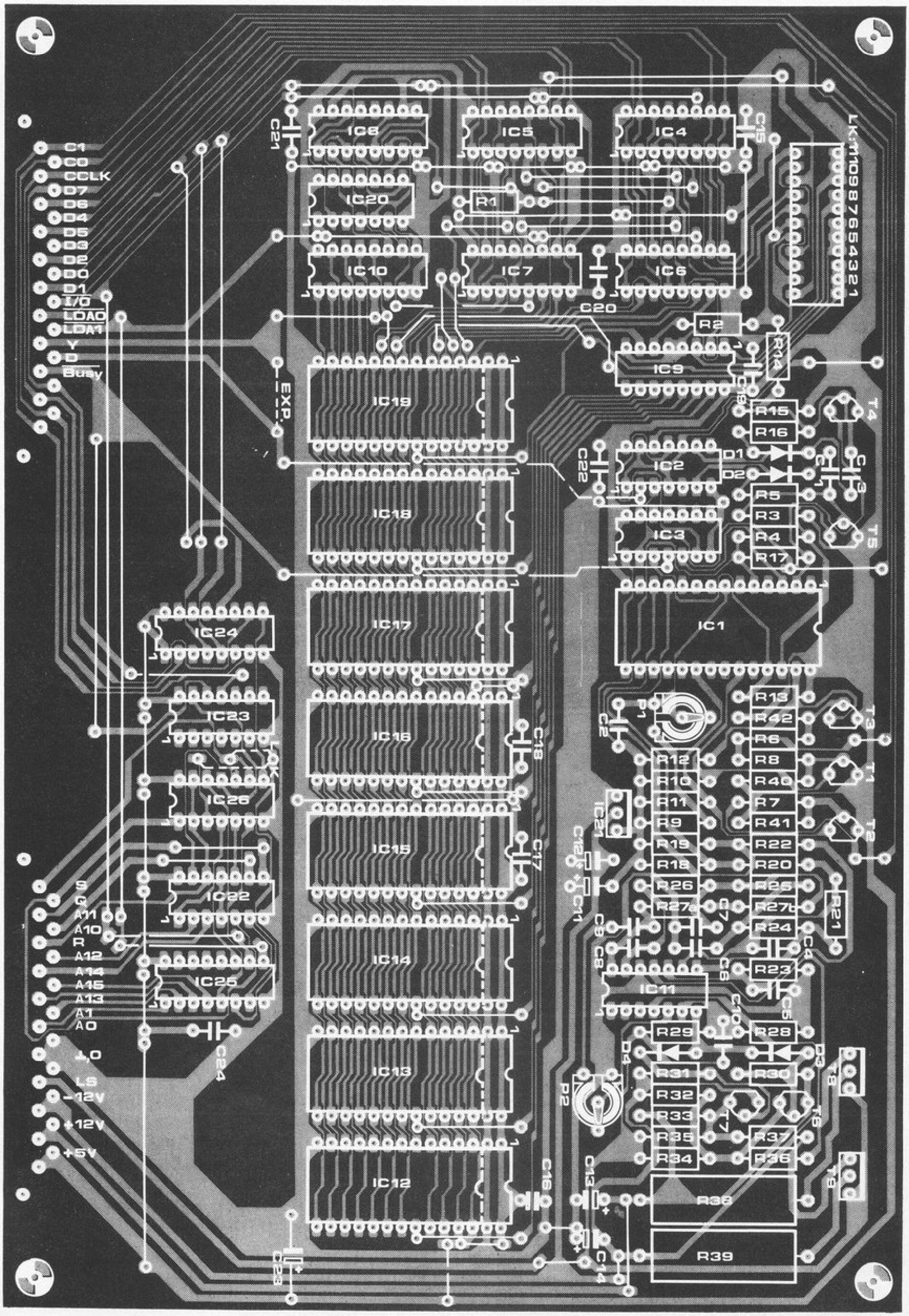

Figure 7. The component layout for the talking board.

Parts list:

Resistors:
R1 . . . R3,R5 . . . R8,R14 . . . R16, R40 . . . R42 = 4k7
R4,R9 . . . R12,R17,R20 . . . R23,R25, R27a,R27b = 10 k
R13 = 22 k
R18,R19 = 47 Ω
R24 = 12 k
R26,R29 = 6k8
R28 = 1 k
R30,R31 = 8k2
R32,R33 = 2k2
R34,R35 = 82 k
R36,R37 = 22 Ω
R38,R39 = 2Ω2/3 W
P1 = 50 k preset potentiometer
P2 = 22 k preset potentiometer

Capacitors:
C1,C3 = 100 p
C2 = 68 p
C4,C5,C8 = 1 n
C6,C7 = 10 n
C9,C15 . . . C22 = 100 n
C10 = 2n2
C11,C12 = 10 µ/16 V tantalum
C13,C14 = 47 µ/16 V tantalum
C23 = 47 µ/10 V

Semiconductors:
D1 . . . D4 = 1N4148
T1 . . . T5 = TUP
T6 = BC 183
T7 = BC 213
T8 = TIP 31
T9 = TIP 32
IC1 = TMS 5100
IC2 = 74LS74
IC3,IC20 = 74LS04
IC4 . . . IC8 = 74LS193
IC9 = 74LS138
IC10 = 74LS151
IC11 = TL 084
IC12 . . . IC19 = TMS 2532*
IC21 = 7905

* see text
A complete kit of parts is available from Crestway Electronics.

Parts list for the power supply (figure 5b) not included in the kit:

Resistors:
R1 = 15 Ω/5 W

Capacitors:
C1 = 4700 µ/40 V
C2 = 2200 µ/40 V
C3,C7 = 220 n
C4 = 2µ2/16 V tantalum
C5,C6,C8 = 1 µ/16 V tantalum

Semiconductors:
D1 . . . D4 = 1N4004
IC1 = 7812
IC2 = 7912
IC3 = 7805

Miscellaneous:
Tr1 = 2 x 15 V/1 A mains transformer
Heatsink for IC3

Parts list for the interface (figure 8) not included in the kit:

Capacitors:
C26 = 100 n

Semiconductors:
IC22 = 74LS02
IC23,IC24 = 74LS175
IC25 = 74LS138
IC26 = 74LS00

8

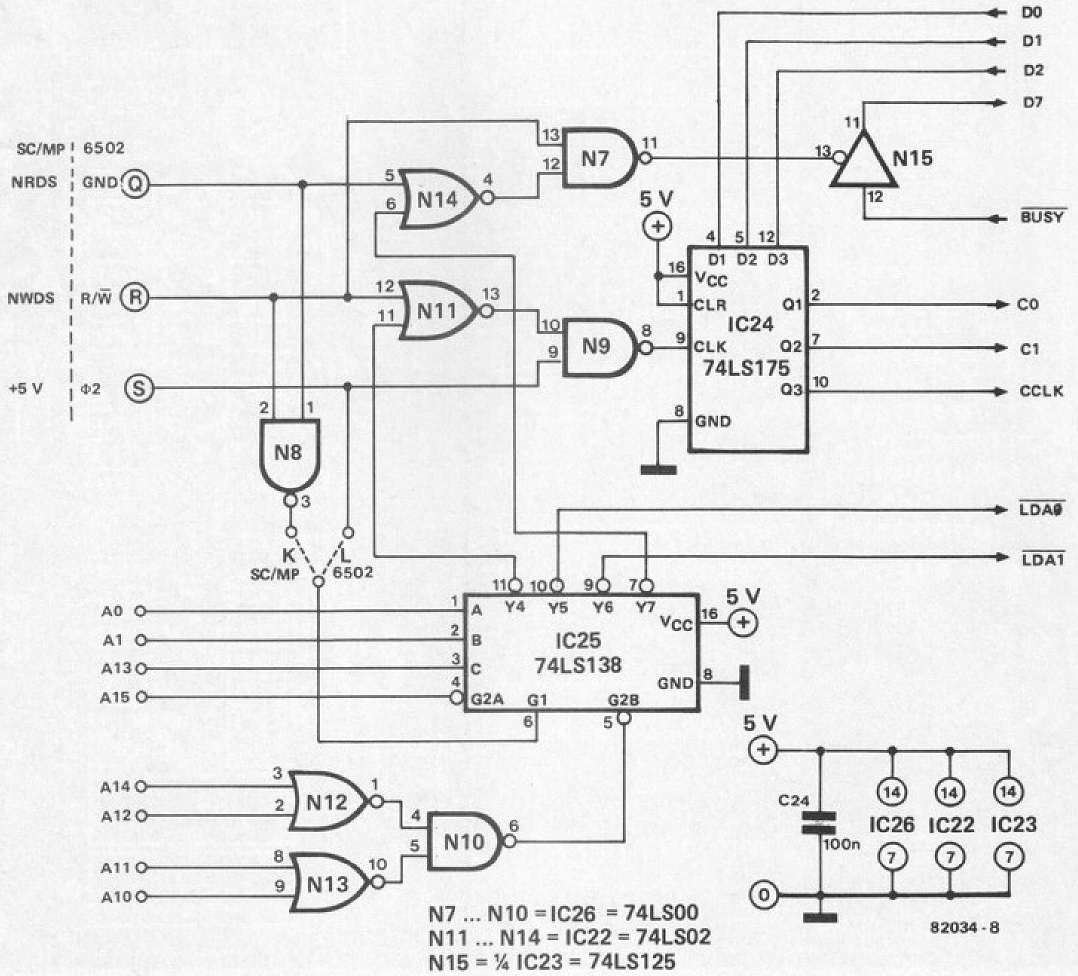

Figure 8. If there are no I/O lines available in the host computer, this simple interface will be required.

9

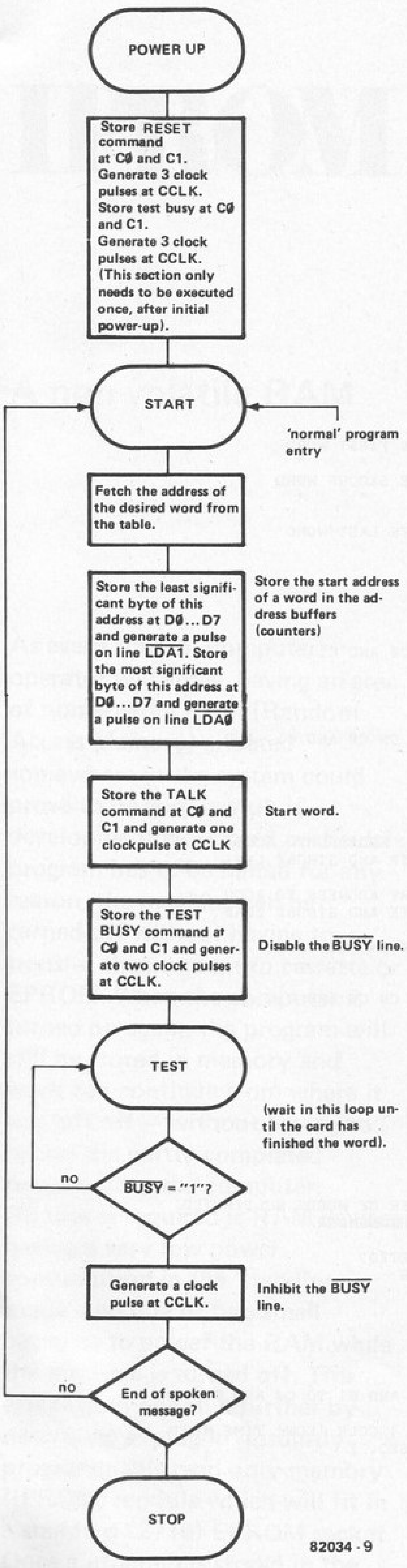

Figure 9. This flow chart explains steps required for the system to produce a speech output.

the energy and pitch increase slightly. And so it goes on.
The basic principle is fairly clear. When scanning a given word (with the intention to turn it into some other word?) the following rules apply

- If the first four bits on a line are zero, forget them: they specify 'silence'.
- Otherwise, look at the next (repeat) bit: if it is logic Ø, filter parameters will be specified; otherwise, the next five 'pitch' bits will complete the line.
- If the 'pitch' bits are ØØØØØ, an unvoiced sound is specified: the following 18 bits determine the filter parameters. For voiced sounds (pitch ≠ ØØØØØ), the following 39 bits determine the filter parameters.
- When the first four bits in a line are 1111, this signifies the end of the word.

Given this information, it is quite feasible to decode any given word. More importantly, it is possible to 'construct' new words by modifying existing codes. We had a crack at assembling the word 'Elektor', and the result was quite acceptable! A basic vocabulary is a great help, of course, and this is supplied in EPROMs with the kit. The words are listed, with the corresponding first addresses, in table 4.

Construction and operation

The printed circuit board and component layout are shown in figures 6 and 7. Construction is started by mounting all the wire links (including EXP) with the exception of link L or K which will be discussed later in the text. Note that T8 and T9 could do with a little heatsink if high output levels are required. As well as the basic circuit, room has been provided on the board for a general purpose microprocessor interface (IC22 . . . IC26 and C24). Connection can be made via a 21 way DIN 41617 male socket with 90° solder pins.
In principle, the board can be driven from any microprocessor system — provided 14 I/O lines are available. These are the 14 lines at the left of figure 4. Lines D, I/O and Y are not used initially. If necessary, they can be used for reading the code in and out. In some instances further interfacing may be required and a suitable circuit is shown in figure 8. It should be noted that, although this circuit can be mounted on the board, the components are not included in the basic 'Talking board' kit. Connection is carried out via the lines to the left of figure 8 which are linked to the corresponding microprocessor lines. In addition, lines D0 . . . D7 must be connected to the data bus in the microprocessor. Connections to the remaining lines at the left of figure 4 are obviously not then required.
Address decoding is rather rudimentary; the circuit shown utilises the complete address block from 2ØØØ to 23FF for only four addresses. Obviously, the address range can be moved or reduced by swapping lines and/or adding further address decoders. Basically, only four addresses are required:

- data for CØ, C1 and CCLK: in this circuit, address 2ØØØ is used. Bit Ø,1 = CØ, C1; bit 2 = CCLK.
- LDA 1 command: address 2ØØ2. (Data = lower address byte).
- LDA Ø command: address 2ØØ1. (Data = higher address byte).
- Busy output: address 2ØØ3, bit 7 (MSB).

The GI input to IC25 can be set according to the microprocessor system used. For the Junior Computer, it must be linked to Ø2 (link L); for the SC/MP it is derived from a combination of NRDS and NWDS (link K). In general, it indicates when the address and data are valid.
Given a suitable interface, it is a fairly simple matter to produce a 'speech' output. The basic flow chart is given in figure 9. After power-up, the first step is to initialise the word processor. This is accomplished by loading the data Ø7-Ø3-Ø7-Ø3-Ø7-Ø3-Ø7, alternately, to address 2ØØØ. This corresponds to a logic 1 for CØ and C1, while CCLK is toggled three times. Note that the CCLK pulse (bit 2 in this sequence) must remain low or high for at least 6.25 μs, which may involve adding a delay in this routine. The next step in the initialisation procedure consists of alternately loading 'ØØ' and 'Ø4' into address 2ØØØ — again, three times in all.
This brings us to 'start': the point at which an actual speech output is initiated. First, the lower address byte for the desired word is transferred to address 2ØØ2 (this automatically initiates the necessary LDA 1 pulse!); then, the higher address byte is transferred to address 2ØØ1. Now the 'Talk' command can be given (Ø2-Ø6 to address 2ØØØ). Finally, the data sequence 'ØØ-Ø4-ØØ-Ø4' is applied to address 2ØØØ, in a 6.25 μs rhythm as before. This corresponds to applying the test busy command and toggling the CCLK input twice.
A test loop is now run, waiting for the 'busy' output (the MSB at address 2ØØ3) to go high. When this occurs, a further 'ØØ-Ø4' sequence is loaded to address 2ØØØ to inhibit the 'busy' output. If further words are to be voiced, the whole procedure can now be repeated from Start.
As a further illustration, a complete program for the Junior Computer is given in table 5.

Component availability

For this project, we have found a very simple solution to the component availability problem: the 'Talking Board' kits are available from Crestway Electronics (among others). Details are given in an advertisement elsewhere in this issue. It

Table 5.

```
                JUNIOR'S ASSEMBLER             PAGE 01

0200                  ORG   $0200
                DATE :  29-9-'81
                SPEECH SYNTHESISER  TMS 5100
                INTERFACE ADDRESSING
0200            CMND    *     $2000  COMMAND ADDRESS
0200            LDZERO  *     $2001  LDA0 STROBE ADDRESS
0200            LDONE   *     $2002  LDA1 STROBE ADDRESS
0200            BUSY    *     $2003  BUSY READ OUT
                JUNIOR MONITOR START ADDRESS
0200            RESET   *     $1C1D
                SPEECH ADDRESS LOOK UP TABLE
0200            TABLE   *     $0400  LOWER ORDER ADDRESS BYTE FIRST WORD
                              $0401  HIGHER  "      "      "     "     "
                              $0402  LOWER ORDER ADDRESS BYTE SECOND WORD
                              $0403  HIGHER  "      "      "     "     "
                              .....
                              $04FF  HIGHER ORDER ADDRESS BYTE LAST WORD
                ***********
                MAINPROGRAM
                ***********
0200 A9 07      POWUP  LDAIM $07
0202 8D 00 20          STA   CMND   SET 'RESET' COMMAND ON C0 AND C1
0205 20 45 02          JSR   TOGGLE TOGGLE CCLK THREE TIMES
0208 20 45 02          JSR   TOGGLE
020B 20 45 02          JSR   TOGGLE
020E A9 00             LDAIM $00
0210 20 45 02          JSR   TOGGLE SET 'TEST BUSY' COMMAND ON C0 AND C1, AND
0213 20 45 02          JSR   TOGGLE TOGGLE CCLK THREE TIMES
0216 20 45 02          JSR   TOGGLE
0219 A2 00      START  LDXIM $00    CLEAR X-REGISTER
021B BD 00 04   STRT   LDAX  TABLE  LOWER ORDER SPEECH START ADDRESS TO ACCU
021E 8D 02 20          STA   LDONE  SET D0 TO D7 TO THIS BYTE AND STROBE LDA1
0221 E8                INX
0222 BD 00 04          LDAX  TABLE  HIGHER ORDER SPEECH START ADDRESS TO ACCU
0225 8D 01 20          STA   LDZERO SET D0 TO D7 TO THIS BYTE AND STROBE LDA0
0228 A9 02             LDAIM $02
022A 20 45 02          JSR   TOGGLE SET 'TALK' COMMAND ON C0 AND C1, AND
                                    TOGGLE CCLK ONCE
022D A9 00             LDAIM $00
022F 20 45 02          JSR   TOGGLE SET 'TEST BUSY' COMMAND ON C0 AND C1, AND
0232 20 45 02          JSR   TOGGLE TOGGLE CCLK TWICE

                JUNIOR'S ASSEMBLER             PAGE 02

0235 AD 03 20   TEST   LDA   BUSY   READ BUSY LINE
0238 10 FB             BPL   TEST   BUSY LINE NOT HIGH?
023A 20 45 02          JSR   TOGGLE TOGGLE CCLK ONCE
023D E8                INX
023E E0 08             CPXIM $08    COMPARE X WITH THE NUMBER OF WORDS MULTIPLIED
                                    BY TWO ( IN THIS CASE THE NUMBER
                                    OF WORDS IS FOUR )
0240 D0 D9             BNE   STRT   WORD SEQUENCE NOT COMPLETED?
0242 4C 1D 1C          JMP   RESET  RETURN TO JUNIOR MONITOR
                **********
                SUBROUTINE
                **********
0245 29 03      TOGGLE ANDIM $03    SET CCLK TO ZERO AND
0247 8D 00 20          STA   CMND   TRANSFER ACCUMULATOR B0 AND B1 TO C0 AND C1 LINE
024A 09 04             ORAIM $04
024C EA                NOP          DELAY TWO MICRO SECONDS ( CCLK 'LOW' TIME MUST
                                    BE AT LEAST 6.25 MICRO SEC. )
024D 8D 00 20          STA   CMND   SET CCLK TO ONE
0250 60                RTS
ID=
```

Table 5. This program provides the Junior Computer with the power of speech!

should be noted that the kit includes the Elektor p.c. board, the speech memory EPROMs and all other components for the basic circuit. It does not include the edge connectors, the loudspeaker or the components for the add-on interface and the power supply — although these are available separately.

At a later date, if there is sufficient demand, further speech memory EPROMs can be made available. For this reason, we will welcome any lists of 'desired words'! Meanwhile, it will prove quite feasible to code your own new words and store them in EPROM, with the aid of a little interface that will be published in the near future. ■