the correct places on the waveband.
Further remarks. Firstly, something that probably does not need mentioning. As the ferrite rod coil is in fact an aerial, it would be unwise to mount the completed receiver in a metal case! Secondly, the zener diode D1 must be either a 250 mW or a 400 mW type, as stated, as otherwise the input level for the voltage source T1 (3.9 V) will not be correct. This is because the current flowing through D1 is far lower than normal in order to keep the current consumption of the circuit to a minimum. Thirdly, as the output transistors do not require any quiescent current, the value of resistors R13 and R14 are fairly critical. If the stated values are not adhered to the chances are that the output transistors will start to draw current after all and, as there is no temperature compensation network, this could well have a detrimental effect on them. Using the values given in figure 4, transistors T4 and T5 will not have to be cooled. They can be ordinary types without any need for heatsinks.

## Results

In practice, the miniature medium waveband receiver was found to perform very satisfactorily. Being a single coil type, it may require constant retuning due to the set 'drifting' off frequency, especially where distant stations are concerned. Even so, it is eminently suitable as a 'stand-by' receiver for news bulletins etc. which is quite often all that is required anyway. It is only when the owner wishes to listen to a weak station in the neighbourhood of a powerful one that the MW receiver is going to have problems. This can often be remedied by turning the receiver towards the weaker station thereby eliminating the stronger one. Local stations can be received very well. In unfavourable circumstances, an external aerial may be experimented with. This should be connected to the top of the tuning coil via a small value capacitor (4p7). This, however, should hardly ever be necessary. If the input signal is clean enough, the sound quality of the receiver will be suprisingly good. In this respect it really stands out amongst similar commercial radios.
Finally, the receiver is remarkably inexpensive. If, like countless other constructors, you have a 'junk' box full of ferrite rods, tuning capacitors and transistors, it will only cost a few pence.



# text display on the Junior Computer

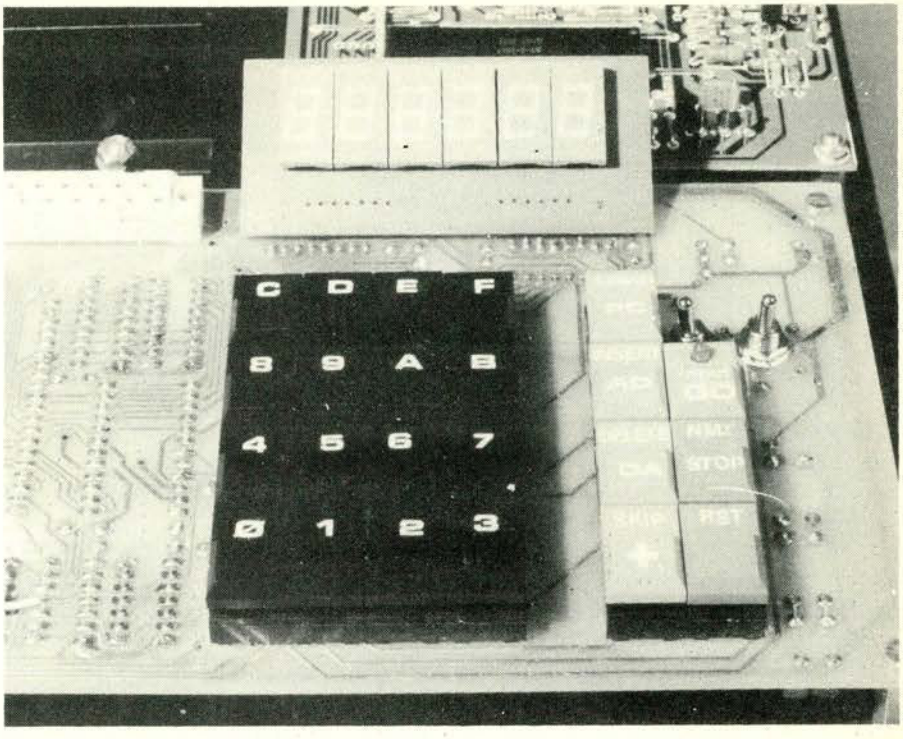
**As we know, the display of the Junior Computer is suitable for displaying both numerical and hexadecimal data. By utilising a seven segment alphabet it is also possible to display written texts. If the text is to be static, a total of six letters are available. If, however a longer message is required, this may 'run' along the display rather like the electronic news display at the top of tall buildings (dynamic text).**

**from an idea by U. Seyffert**

This particular topic receives full attention in Junior Computer Book 2 (to be available shortly), but there is no harm in whetting the appetites of our readers even if it is a little premature.
How can the Junior Computer display words? Normally speaking, data and address information is displayed with the aid of the monitor routine SCANDS. This involves one of the hexadecimal numbers, Ø . . . F, in each display. Where texts are concerned, however, the monitor routines are no good. What is needed is the subroutine SHOW with the addition of a special look-up table which contains the corresponding seven segment pattern for each individual letter.
Table 1 provides a survey of letters and figures together with the corresponding data which has to be entered into port A for them to be displayed. This table has been partly based on suggestions made to us from one of our readers. Obviously, letters which include diagonal lines (such as K, M, N, Q, V, W, X and Y) will have to be adapted to the horizontal and vertical set up of the display segments. Experience has shown, however, that the eye and the brain soon become accustomed to this.
Now for a short program that will allow a six letter word to appear on permanent display. A good example would be the word 'Junior' as indicated on the prototype of the Junior Computer in the front cover photograph of the May 1980 issue of Elektor and Book 1. The program, JUNIOR, is listed in table 2.
Here the modified SHOW routine will be called SHOWDS and the look-up table that holds the information relating to the display of any particular character is called TXT (text table). The Y index register acts as the display counter and text index. The value contained in the Y register increases from ØØ to Ø5 as an index for the particular character to be displayed. As soon as the value in the Y register becomes Ø6,

Table 1.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 40 | E | 06 | o | 23 |
| 1 | 79 | e | 04 | P | 0C |
| 2 | 24 | F | 0E | q | 18 |
| 3 | 30 | G | 42 | r | 2F |
| 4 | 19 | g (9) | 10 | S | 52 |
| 5 | 12 | H | 09 | S (5) | 12 |
| 6 | 02 | h | 0B | t | 07 |
| 7 | 78 | i | 7A | u | 63 |
| 8 | 00 | i | 6F | V | 41 |
| 9 | 10 | j | 72 | W | 01 |
| A | 08 | K | 0A | X | 36 |
| a | 20 | L | 47 | Y | 11 |
| b | 03 | l | 4F | Z | 64 |
| C | 46 | M | 48 | – | 3F |
| c | 27 | n | 2B | = | 37 |
| d | 21 | 0 (0) | 40 | sp | 7F |




after the instruction INY, it is reset to 00 (jump to DISMPX to begin another round). During the subroutine SHOWDS the Y register contains a delay value which determines the length of time that each display is actually lit. For this reason the previous value contained in the Y register (display counter/text index) must be saved in the address location TEMPY (0004) before the jump to the SHOWDS subroutine takes place.

The function of the X index register, on the other hand, is the same as it was for the SHOW routine: it acts as a display digit switch by way of port B. In other words, the information contained in the X register (08, 0A, 0C, 0E, 10 and 12 consecutively) is passed to port B data register to turn each of the displays on in turn.

## Text on the run . . .

A stationary text is all very well, but it does tend to get a little monotonous after a while. A much more interesting possibility would be to update the displayed text every few moments. In this manner whole sentences could be displayed instead of just single words. This can be accomplished with the aid of the program JUNTXT shown in table 3. The effect is very similar to that of an electronic news display. It is an expanded version of the earlier program JUNIOR (table 2). Page 03 is used to store the actual text which can, therefore, be up to 256 characters in length

Table 2.

| | | | | |
|---|---|---|---|---|
| JUNIOR | 0200 | A9 7F | LDA # 7F | |
| | 0202 | 8D 81 1A | STA-PADD | PA0 . . . PA6 are outputs |
| DISMPX | 0205 | A2 08 | LDX # 08 | start from Di1 |
| | 0207 | A0 00 | LDY # 00 | therefore display counter Y = 00 |
| ONEDIS | 0209 | 84 04 | STYZ-TEMPY | store display counter |
| | 020B | 20 17 02 | JSR-SHOWDS | display first/next character |
| | 020E | A4 04 | LDYZ-TEMPY | retrieve state of display counter |
| | 0210 | C8 | INY | increment display counter |
| | 0211 | C0 06 | CPY # 06 | have all 6 displays been accessed? |
| | 0213 | F0 F0 | BEQ DISMPX | if yes, start again |
| | 0215 | D0 F2 | BNE ONEDIS | if not, next display |
| SHOWDS | 0217 | B9 30 02 | LDA-TXT, Y | fetch seven segment code |
| | 021A | 8D 80 1A | STA-PAD | place segment code on port A |
| | 021D | 8E 82 1A | STX-PBD | turn on display digit |
| | 0220 | A0 7F | LDY # 7F | |
| DELAY | 0222 | 88 | DEY | |
| | 0223 | 10 FD | BPL DELAY | delay a short while |
| | 0225 | 8C 80 1A | STY-PAD | Y = FF (blanking) to port A |
| | 0228 | A0 06 | LDY # 06 | turn off display |
| | 022A | 8C 82 1A | STY-PBD | |
| | 022D | E8 | INX | prepare next display digit |
| | 022E | E8 | INX | |
| | 022F | 60 | RTS | |
| TXT | 0230 | 61 | "J" | look-up table |
| | 0231 | 63 | "u" | Y = text index |
| | 0232 | 2B | "n" | (Y = 00 . . . 05) |
| | 0233 | 6F | "i" | |
| | 0234 | 23 | "o" | |
| | 0235 | 2F | "r" | |

Table 3.

```
JUNTXT  0200   A9 7F        LDA # 7F
        0202   8D 81 1A     STA-PADD        PA0 . . . PA6 are outputs
        0205   A5 00        LDAZ-NUM        contents NUM (0000) to accumulator
        0207   38           SEC             C = 1
        0208   E9 05        SBC # 05
        020A   85 02        STAZ-NUMCOR     NUMCOR = NUM minus 05
BEGIN   020C   A9 00        LDA # 00
        020E   85 01        STAZ-NUMVAR     first display text
DSTIME  0210   A9 6F        LDA # 6F
        0212   85 03        STAZ-DISCNT     establish text display time
DISMPX  0214   A2 08        LDX # 08        start from Di1
        0216   A0 00        LDY # 00        display counter (Y) = 00
ONEDIS  0218   84 04        STYZ-TEMPY      store display counter
        021A   98           TYA             Y to accumulator
        021B   18           CLC             C = 0
        021C   65 01        ADCZ-NUMVAR     A ←Y + contents NUMVAR (0001)
        021E   A8           TAY             accumulator to Y
        021F   20 39 02     JSR-SHOWDS      display first/next character
        0222   A4 04        LDYZ-TEMPY      retrieve state of display counter
        0224   C8           INY             increment display counter
        0225   C0 06        CPY # 06        have all 6 display been accessed?
        0227   F0 02        BEQ TMECHK      if yes, move on to time check
        0229   D0 ED        BNE ONEDIS      if not, next display
TMECHK  022B   C6 03        DECZ-DISCNT     time up?
        022D   D0 E5        BNE DISMPX      if not, repeat present text
        022F   E6 01        INCZ-NUMVAR     if yes, update text
        0231   A5 02        LDAZ-NUMCOR
        0233   C5 01        CMPZ-NUMVAR     end of text?
        0235   B0 D9        BCS DSTIME      if not, show new text
        0237   90 D3        BCC BEGIN       if yes, start again
SHOWDS  0239   B9 00 03     LDA-TXT, Y  }
        023C   8D 80 1A     STA-PAD     }
        023F   8E 82 1A     STX-PBD     }
        0242   A0 7F        LDY # 7F    }
DELAY   0244   88           DEY         }   see 'JUNIOR' program
        0245   10 FD        BPL DELAY   }   TXT = 0300 (table 4)
        0247   8C 80 1A     STY-PAD     }   text index = Y + contents NUMVAR
        024A   A0 06        LDY # 06    }
        024C   8C 82 1A     STY-PBD     }
        024F   E8           INX         }
        0250   E8           INX         }
        0251   60           RTS         }
```

– enough for the average length paragraph!

Again, this program uses the subroutine SHOWDIS, only this time the text table (TXT) is located at address 0300 and although the Y register is still used as a display counter it is no longer used as a text index directly. Instead, the particular section of the text to be displayed is calculated by adding the instantaneous value in the Y register to the contents of address location NUMVAR (0001).

The value contained in NUMVAR will be constant for the period of time a certain text is on display (the actual duration can be adjusted by modifying the contents of location 0211). As soon as that period of time is over the contents of NUMVAR are incremented by one: the entire text shifts one location to the left and the right hand display shows a new character. When the contents of NUMVAR are greater than the contents of location NUMCOR, we will have arrived back at the beginning, as this means that the entire text will have been displayed. This is because the contents of NUMCOR are 05 less than those of location NUM. The latter (location 0000) is where the user must store the low order byte of the last memory location of the text table. In other words, if the last character of the text message is stored in location 0332, the value 32 is stored in location 0000 (NUM).

Table 4.

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0300  7F 7F 7F 7F 7F 7F 07 0B 2F 23 01 7F 20 7F 02 7F
0310  01 6F 07 0B 7F 07 0B 06 7F 61 63 2B 6F 23 2F 7F
0320  46 23 48 0C 63 07 06 2F 7F 3F 7F 03 63 11 7F 03
0330  23 23 0A 7F 24 xx xx xx xx xx xx xx xx xx xx xx

0000 (NUM) = 34
```

Table 4 provides a sample text which can be displayed on the Junior Computer with the aid of the program JUNTXT as given in table 3. The text contains a message for Junior Computer Book 1 owners. A text should always be preceded by at least six blank spaces (7F), so that the beginning and end of the message are clearly separated from each other.