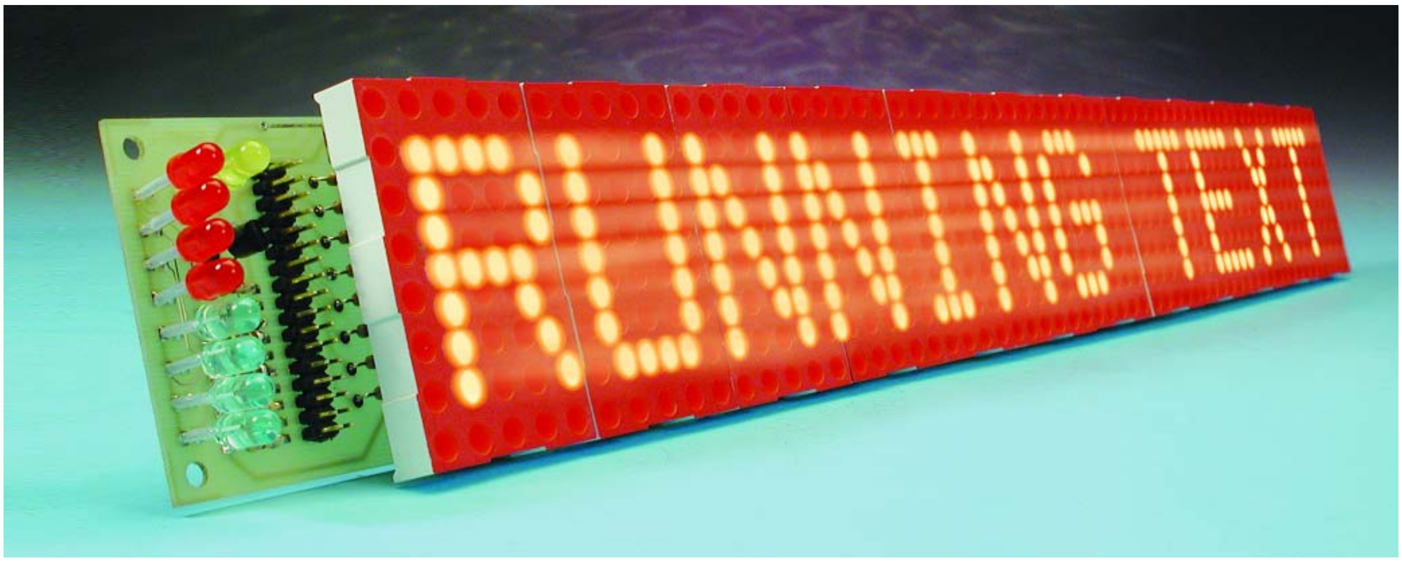
Running Text Display

A 10-character display for the Elektor Flash Micro board

Design by W. Wätzig

This peripheral circuit for the 89S8252 Flash Board provides a large-format, 10-character running-text display using dot-matrix modules.



The subject of this article may sound familiar. Haven't we already devoted enough attention to running-text displays? The answer is, yes and no! In the 'Modular Dot Matrix Display' (*Elektor Electronics*, June 2001), the text to be displayed is passed from a PC to an 89S8252 microcontroller using an RS232 interface. In that design, the task of the microcontroller is to convert the characters into bit patterns for the matrix modules and manage the multiplexed drive signals for the display modules.

At first glance, the basic idea of the circuit shown in **Figure 1** appears to be the same. However, in this case the only 'intelligence' is located in a separate microcontroller system, in the form of the AT89S8252 Flash Board. The actual running-text display, which can display ten characters on its twelve display modules, is fully passive. However, this design has a few unusual features.

The running-text functions can be controlled using a PC keyboard connected directly to the circuit, as well as via the serial interface of the Flash Board. In addition, a DS1302 real-time clock is included in the circuit, to allow the date and time to be output alternately.

Inputs

A serial PS/2 interface is used for entering characters and control codes. The PC keyboard is connected to K1, and the scan codes are stored in the 40105 first-in, first-out (FIFO) memory. The Flash Board reads the keyboard data from the FIFO memory at its own speed and converts the scan codes into ASCII characters. The character coding can be selected to match the German (DE) or English (EN) keyboard layout. The keyboard selection is indicated by the most significant bit of the status display (textsel); this LED is on if the EN layout is selected.

Alternatively, the circuit can be controlled via the serial interface of the microcontroller, using a maximum data transfer rate of 1200 baud. The baud rate can be set to 150, 300, 600 or 1200 using jumpers JP2 and JP3. Jumper JP1 selects either keyboard or serial interface control; the selection takes effect after the microcontroller is reset.

The texts to be displayed are stored in the microcontroller EEPROM, so they are protected in case of loss of power. The 2048 bytes of EEPROM memory are divided into eight 240-byte text blocks. Each block can hold twelve lines of text, with each line having up to twenty characters.

To show where the text is being stored during entry, the block number of the currently selected text

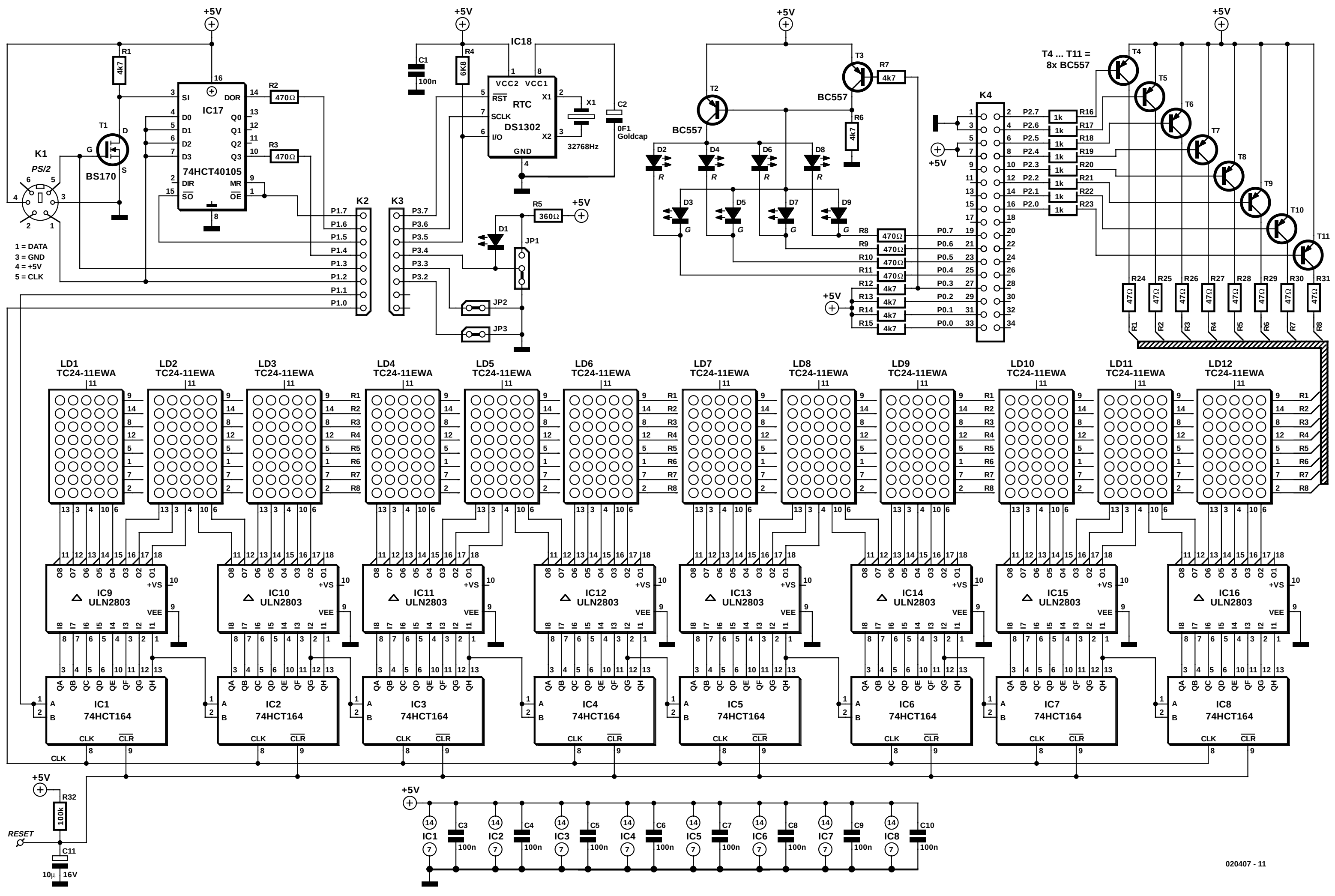


Figure 1. The running-text display has twelve display modules and a 60-stage shift register.

block is shown in binary form in the upper nibble of the eight-bit status display, and the current line number is shown in the lower nibble.

Outputs

The display consists of twelve dot-matrix display modules, each of which has a 5 × 8 LED matrix. This provides a total of 60 columns, each containing eight LEDs. Ten characters can be displayed at the same time, each in a 5 × 8 matrix, with one column of space between each pair of characters. With such a large number of LEDs, multiplex operation is the only reasonable choice.

A display buffer for a total of 20 characters is provided using 120 bytes of the internal RAM. The LED columns are multiplexed at a rate of 3600 Hz, which is driven by an interrupt. The refresh rate for the entire display is thus 60 Hz.

The column data is clocked out from the microcontroller board via P1.1 into a shift register formed by cascaded 74HCT164 ICs. The clock signal appears on P1.0 and is applied to all of the shift register ICs simultaneously. Once the column data has been clocked into the shift register, the row to be displayed is enabled using one of the driver transistors T4–T11. These driver transistors occupy all of port P2 of the microcontroller.

Text management

Control characters are used for text management. Function keys F1–F11 and other special characters, such as CR, Pause, Enter, Delete and so on, are used for this purpose.

When the serial interface is used, the control function codes are entered using the surrogate representation **#x**. The # character indicates a control function, and the following character specifies the specific function. For example, function key F1 is replaced by the character sequence **#1**. The control characters that can be entered are listed in the 'Control character input' table, along with their keyboard designations.

The circuit

The circuit consists of the microcontroller on the Flash Board, which provides the control functions, and the peripheral circuitry on the display board, which is connected to the Flash Board using three flat cables. The display board consists of four functional blocks:

- The 74HCT105 FIFO (IC17), for entering the scan codes from the keyboard. This memory decouples the keyboard data clock from the scan code read routine, in order to prevent any bits from being lost when the read routine is interrupted by the display routine.

- The DS1302 real-time clock (IC18) with the Goldcap storage capacitor (C2). When the running-text display is used for the first time, the actual date and time must be set in the real-time clock IC. We recommend doing this ‘top down’, starting with the year and finishing by setting the seconds. This is because the seconds register of the real-time clock is preset in the factory to ‘80’, which corresponds to a wait mode. The Goldcap capacitor provides backup power for the real-time clock, so it will continue running when the power is switched off.

- Two 4-bit LED status displays (D2–D9) for the text block number and line number.

- The dot-matrix display, consisting of the display modules (LD1–LD12) and 74HCT174 shift registers (IC1–IC8), the ULN2803 column drivers (IC9–IC16) and the BC557 row driver transistors (T4–T11).

The software

The microcontroller program can be divided into three parts, consisting of the initialisation routine, the display loop and the character input loop.

Initialisation

After the microcontroller is switched on, the initialisation routine is triggered by the reset interrupt and executes once. It first sets the baud rate for the serial interface (the microcontroller UART) according to the value selected by the levels on pins P3.2 and P3.3. It also sets the Keyboard/Serial flag (jumper JP1), creates the display buffer (120 bytes in RAM, starting at address 080h), and initialises the display counter.

Next, the keyboard is switched to Scan Code 3, in which mode only one character is sent for each keypress. This considerably simplifies further processing of the scan codes.

The real-time clock is also initialised, allowing the Goldcap storage capacitor to be recharged from the supply voltage. The initialisation phase ends with writing an initial text to the display buffer and enabling the interrupt, thus allowing Timer0 to run.

The display loop

The display loop is triggered by a Timer0 interrupt every 277 μs, which corresponds to

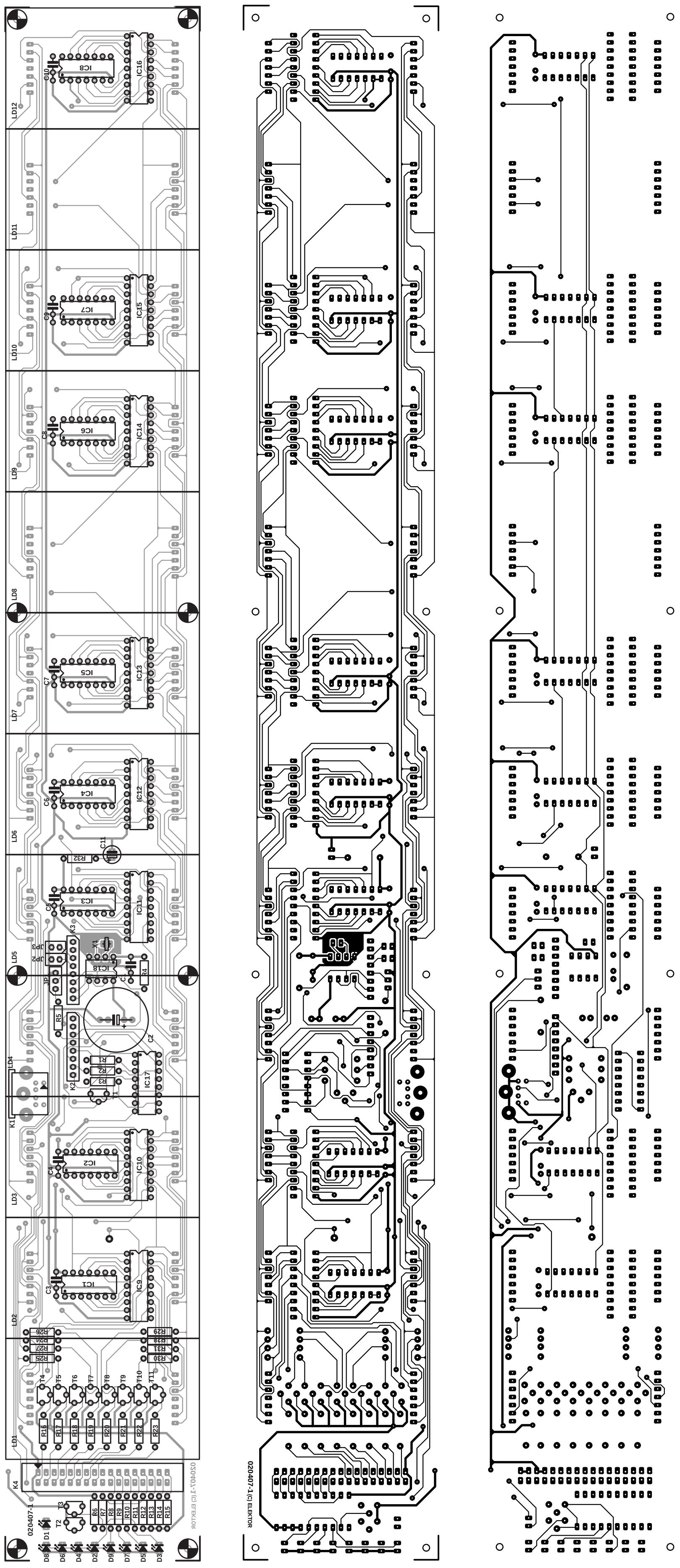


Figure 2. The printed circuit board layout (50% of actual size).

a rate of 3600 Hz. The entire dot-matrix display is thus refreshed 60 times per second. The display loop fetches the display bytes sequentially from the RAM and presents them to the shift register, generates clock pulses for the shift register, generates the data patterns for the LED displays, and outputs the control signals for switching the text, scrolling the text, and setting the display to inverted or blinking mode.

The character input loop

This loop waits for the next character from the keyboard or the serial interface. Each scan code from the keyboard is read via the FIFO and converted into a control character or an ASCII character using a code table (tastasc). Alternatively, ASCII characters are read from the serial interface. In this case, control characters are generated using a surrogate representation.

When a control character (such as Return, Shift, or Backspace) is encountered, the associated routine is called using a dispatcher. Each ASCII character is stored in the next free location in the EEPROM, and its character image (five bytes) is read from a table (charimg) and written to the display buffer in RAM.

Printed circuit board

It's not easy to design and produce a printed circuit board big enough to hold twelve large display modules and the control electronics (how many people have an etching tray that can hold a 50-cm PCB?). For this reason, we have designed a circuit board layout, which is shown at half its actual size in **Figure 2**. This double-sided, through-hole-plated circuit board is not available from Readers Services, but it can be obtained from The PCBShop via the link on the *Elektor Electronics* home page.

Fortunately, stuffing the board is quite easy. The SIL connectors, jumper pins and mini-DIN socket are fitted on the bottom of the board, while the remainder of the components are fitted on the top. Make sure that none of the components extends more than 8 mm above the surface of the board, since otherwise the display modules installed above these components will not fit into their sockets. The Goldcap capacitor specified in the components list is exactly 5.5 mm high. If you wish to use sockets for the ICs, only very low-profile types are suitable.

(020407-1)

COMPONENTS LIST

Resistors:
R1,R6,R7,R12-R15 = 4kΩ7
R2,R3,R8-R11 = 470Ω
R4 = 6kΩ8
R5 = 360Ω
R16-R23 = 1kΩ
R24-R31 = 47Ω
R32 = 100kΩ

Capacitors:
C1,C3-C10 = 100nF
C2 = 0.1F 5.5V Goldcap (Panasonic NF)
C11 = 10µF 63V radial

Semiconductors:
D1 = LED, yellow, low current, 5mm
D2,D4,D6,D8 = LED, red, low current, 5mm
D3,D5,D7,D9 = LED, green, low current, 5mm
IC1-IC8 = 74HCT164
IC9-IC16 = ULN2803A
IC17 = 74HCT40105
IC18 = DS1302 (Dallas)
T1 = BS170
T2-T11 = BC557B

Miscellaneous:
JP1 = 3-way SIL pinheader with jumper
JP2,JP3 = 2-way SIL pinheader with jumper
K1 = 6-way mini-DIN socket, PCB mount (PS/2)
K2,K3 = 8-way SIL pinheader
K4 = 34-way SIL pinheader
LD1-LD12 = 5x8 dot matrix display with common cathode, size 60.8×38 mm, Kingbright type TC24-11EWA
AT89S8252-24PC for 89S8252 Flash Micro board (010208), programmed
Disk, contains controller program (source and hex), order code **020407-11** or Free Download
PCB, order code **020407-1** (see Readers Services page)

Control character input

Function	Key	Serial Input	Function
Text selection	Pause Fx	#P #x	Selection of text block, Fx = F1-F8
	F1-F12	#1 to #9 #A #B #C	election of display lines #1 to #12 in current text block
Edit/Enter Texts	Cr	#+	Go to next line
	BackSpace/Delete	#–	Move back one character
	Shift left/right		Lower/Upper case letters
	Alt		Switch to special characters @ { [ ] } ~ | \
	Enter (num. keypad)	#Z	Clears current text line
	Insert	#*	Starts entry of a long text of up to 240 characters in current display line, extending over several lines. During text entry the display is switched to inverse characters. Text entry is ended by a pressing Insert again.
Display manipulation	Scroll Lock	#R	Scrolls running text
	Home	#0	Start of line and reset scrolling
	Print Screen Fx1 Fx2	#D #x1 #x2	Automatic text display #Fx(1) to #Fx(2)
	Num Lock	#N	Invert/blink display
Real-time Clock control	Esc Esc	#E #E	Display date and time
	Esc Fx yy	#E #x yy	F1: seconds (yy = 00-59)
			F2: minutes (yy = 00-59)
			F3: hours (yy = 00-23)
			F4: day of month (yy = 01-31)
			F5: month (yy = 01...12)
			F6: day of week (yy = 01-07)
			F7: year (yy = 00-99)
Diacriticals etc.	Direct entry on German keyboard	:s = ß :: = : :a = ä :A = Ä :o = ö :O = Ö :u = ü :U = Ü	
Keyboard encoding	Page Up		English keyboard
	Page Down		German keyboard