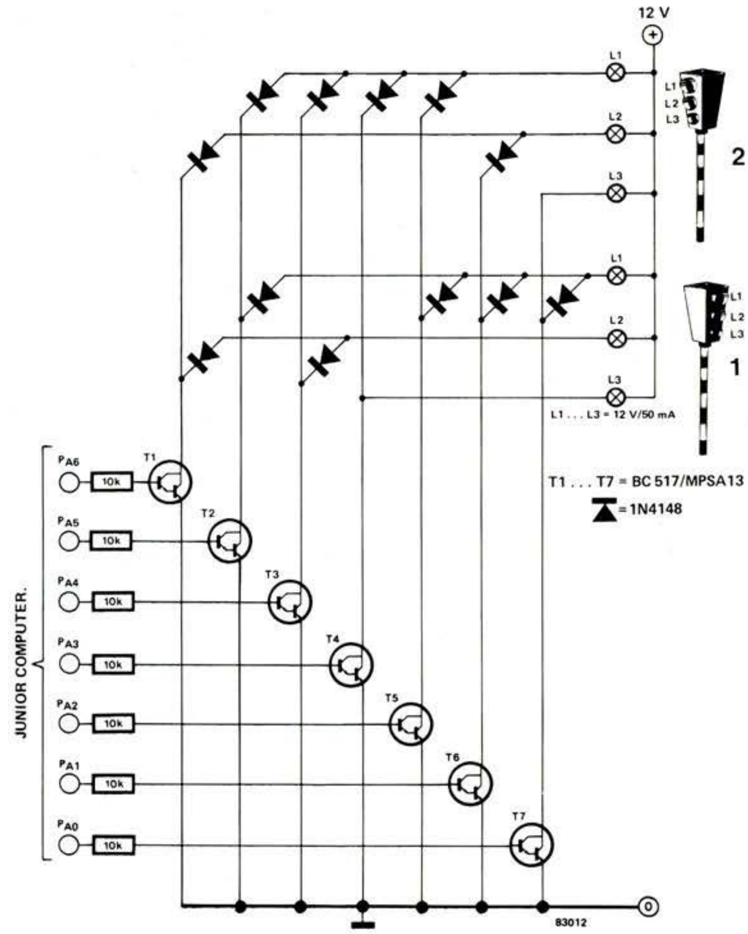

This classical task for a microcomputer, a control function which can otherwise only be implemented with a considerable amount of hardware, is executed by a program. Computers have been employed for fullscale versions of this technique for some time. This is now an opportunity to utilise the Junior Computer for the same application in miniature.

D. Herzberg

traffic-light control system...

. . . with the Junior Computer

The hardware can be easily constructed on a small perforated board, which can be positioned in the vicinity of the port connector. The two traffic lights are connected via two three-core control lines. The positive operating voltage can be taken from smoothing capacitor C5 of the basic power supply for the Junior Computer. Any other unregulated 12 V power supply is just as suitable. Bulbs with different voltage ratings will of course require a different operating voltage. Constructors who prefer to use LEDs for the traffic lights must join the anodes of the LEDs and connect them to the positive operating voltage via a (common) limiting resistor. At 12 V and an LED current of 60 mA (10 mA per LED), for example, the value of the resistor is 200 Ω/1 W. If all LEDs light up equally brightly, a limiting resistor of 1k2, 1/2 W must be connected in series with each LED.

As in the case of other circuits requiring a minimum amount of hardware, this article will merely provide some details regarding construction. Figure 1 shows the control stages for a total of six bulbs in two traffic lights, together with leads and a diode matrix which also provides a protective function. The two traffic lights are controlled by the program in table 1.
The program can best be described on the basis of the Assembler listing in table 1. Starting at address Ø2ØØ, the computer starts to 'shift' a logic 1 (which corresponds to a logic 0 at the collectors of the Darlington driver transistors) from PA1 via PA5 to PA0. The traffic light cycles are: traffic light 1 – red, traffic light 2 – amber (for 2 s); TL1 – red/amber, TL2 – red; TL1 – green, TL2 – red (for 10 s); TL1 – amber, TL2 – red; TL1 – red, TL2 – red/amber (for 2 s); TL1 – red, TL2 – green (for 10 s). The cycle then starts

```
          JUNIOR'S ASSEMBLER              PAGE Ø1
Ø2ØØ                       ORG    $Ø2ØØ


                 TRAFFIC LIGHT SIMULATION WITH THE JUNIOR COMPUTER


                 WRITTEN BY DIETER HERZBERG BERLIN

                 DEFINITION OF ADDRESSES

Ø2ØØ             PA     *      $1A8Ø  PORT A DATA REGISTER
Ø2ØØ             PADD   *      $1A81  PORT A DATA DIRECTION

Ø2ØØ             TIMERD *      $1A97
Ø2ØØ             END    *      $1A85


Ø2ØØ A9 7F       START  LDAIM $7F     INITIALIZE PORT A
Ø2Ø2 8D 81 1A           STA   PADD

Ø2Ø5 AØ Ø2       LOOP   LDYIM $Ø2     LOOP COUNTER
Ø2Ø7 8C 8Ø 1A           STY   PA      BEGIN WITH PHASE 1 =A1:RED/A2:AMBER

Ø2ØA 2Ø 2F Ø2    NEXT   JSR   SHORT   WAIT FOR 2 SECONDS(PHASE 1+2)
Ø2ØD ØE 8Ø 1A           ASL   PA      PHASE 2+3
Ø21Ø 88                 DEY
Ø211 DØ F7              BNE   NEXT
Ø213 2Ø 2C Ø2           JSR   LONG    WAIT FOR 1Ø SEC.AT PHASE 3
Ø216 AØ Ø2              LDYIM $Ø2     PHASE 4+5

Ø218 ØE 8Ø 1A    NEXTA  ASL   PA
Ø21B 2Ø 2F Ø2           JSR   SHORT   WAIT FOR 2 SEC. (PHASE 4+5)
Ø21E 88                 DEY
Ø21F DØ F7              BNE   NEXTA
Ø221 A9 Ø1              LDAIM $Ø1     PHASE 6=PHASE 1
Ø223 8D 8Ø 1A           STA   PA
Ø226 2Ø 2C Ø2           JSR   LONG    WAIT FOR 1Ø SEC. AT PHASE 1
Ø229 4C Ø5 Ø2           JMP   LOOP


Ø22C A2 28       LONG   LDXIM $28     DELAY FOR 1Ø SEC.
Ø22E 2C                 =     $2C

Ø22F A2 Ø8       SHORT  LDXIM $Ø8     DELAY FOR 2 SEC.

Ø231 A9 F4       LOAD   LDAIM $F4     25Ø MS
Ø233 8D 97 1A           STA   TIMERD  DIVISION FACTOR 1Ø24 MS

Ø236 2C 85 1A    TIMEND BIT   END     TIME OUT?
Ø239 1Ø FB              BPL   TIMEND  NO
Ø23B CA                 DEX           YES LOAD TIMER WITH 2ND DELAY
Ø23C 1Ø F3              BPL   LOAD    2ND TIME OUT?
Ø23E 6Ø                 RTS


Ø23F A9 7F       US     LDAIM $7F     ONLY IF LIGHTS ARE OUT OF ORDER
Ø241 8D 81 1A           STA   PADD
Ø244 A9 4Ø              LDAIM $4Ø

Ø246 8D 8Ø 1A    PORT   STA   PA      BOTH AMBER LIGHTS FLASHING
Ø249 A2 Ø4              LDXIM $Ø4     1 SEC. ON/OFF
Ø24B 2Ø 31 Ø2           JSR   LOAD
Ø24E AD 8Ø 1A           LDA   PA
Ø251 49 4Ø              EORIM $4Ø     INVERT PORT
Ø253 4C 46 Ø2           JMP   PORT



SYMBOL TABLE 3ØØØ 3Ø54
END    1A85     LOAD   Ø231     LONG   Ø22C     LOOP   Ø2Ø5
NEXT   Ø2ØA     NEXTA  Ø218     PA     1A8Ø     PADD   1A81
PORT   Ø246     SHORT  Ø22F     START  Ø2ØØ     TIMEND Ø236
TIMERD 1A97     US     Ø23F


JUNIOR

M
HEXDUMP: 2ØØ,255
       Ø  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
Ø2ØØ: A9 7F 8D 81 1A AØ Ø2 8C 8Ø 1A 2Ø 2F Ø2 ØE 8Ø 1A
Ø21Ø: 88 DØ F7 2Ø 2C Ø2 AØ Ø2 ØE 8Ø 1A 2Ø 2F Ø2 88 DØ
Ø22Ø: F7 A9 Ø1 8D 8Ø 1A 2Ø 2C Ø2 4C Ø5 Ø2 A2 28 2C A2
Ø23Ø: Ø8 A9 F4 8D 97 1A 2C 85 1A 1Ø FB CA 1Ø F3 6Ø A9
Ø24Ø: 7F 8D 81 1A A9 4Ø 8D 8Ø 1A A2 Ø4 2Ø 31 Ø2 AD 8Ø
Ø25Ø: 1A 49 4Ø 4C 46 Ø2

JUNIOR
```

again from the beginning. Another traffic light cycle is simulated with start address Ø23F. In this case the two amber bulbs flash at a 1 s rate. PA6 is utilized for this. The hex dump listing shows a summary of the data to be entered.

In the event of a computer fault, the diodes ensure that at least one traffic light is at red, thus preventing a traffic jam. Accidents are also almost ruled out.