CAR COMPUTER

CAR COMPUTER PART TWO

Electronics Australia
AUGUST 1982

Although the operation of our microprocessor-based Car Computer is quite involved, most of the complexity is concealed within three major integrated circuits. Construction is therefore relatively simple. This month we give the circuit and software details and describe construction.

by JOHN CLARKE

By far the most important integrated circuit used in the Car Computer is IC1, a Motorola 6802 8-bit microprocessor (MPU). This microprocessor can perform all the instructions of the well-known 6800 microprocessor and contains 128 bytes of volatile Random Access Memory (RAM), which can be used for data storage when running a program. Additional to this is the advantage that the first 32 bytes (8-bits wide) of RAM is separately powered, enabling important information to be retained when power to the main processor is switched off.

IC2, the Peripheral Interface Adaptor (PIA), supports the external hardware devices. In our circuit, IC2 is used to drive (write) the LED display, read the function select switches and detect the distance, fuel and time pulses. Two 8-bit ports, PA0 to PA7 and PB0 to PB7, are available and can be programmed as inputs (read) or outputs (write). On each port are two extra lines, CA1 and CA2 and CB1 and CB2 respectively. CA1 and CB1 are inputs and CA2 and CB2 can be programmed as either inputs or outputs.

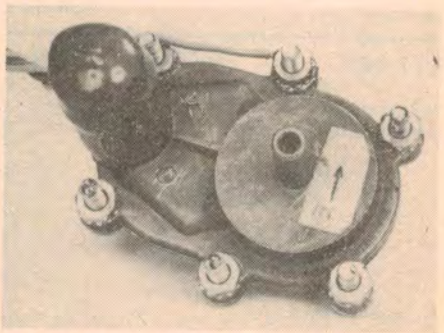
IC3 is a 2K byte Electrically Programmable Read Only Memory (EPROM), which holds the program for the Car Computer. This memory is non-volatile, which means that switching off the power to the IC will not erase the memory. The program remains stored indefinitely (unless erased by ultra-violet light).

A common 8-bit data bus interconnects IC1, IC2 and IC3. This provides two-way (Read/Write) communication between these devices. An address bus (A0 to A10) connects IC1 and IC3 and this is used to access all the EPROM locations.

Simple address decoding for IC2 and IC3 is performed by two NAND gates, IC6a and IC6b. IC3 is accessible to IC1 when both the VMA and A14 lines are high, and IC6a brings $\overline{CE}$ of IC3 low. The VMA line, or Valid Memory Address line, indicates that a valid address is on the bus. The memory locations for the EPROM are from hexadecimal number 6000 to 67FF.

All the registers of IC2 are accessible by high, low combinations of RS0 and RS1, which are connected to A0 and A2 respectively. However, these registers cannot be selected unless the Chip Select lines CS0, CS1 and $\overline{CS2}$ are true. CS1 is permanently held high and CS0 is connected to the $\overline{CE}$ if IC3. When A15 and the VMA are both high, IC6b brings $\overline{CS2}$ low. Providing that IC3 is not selected with A14 high, then IC2 is selected. We used addresses from 8004 to 8007 to access the PIA.

One point of interest here is why the $\overline{CE}$ of IC3 has been connected to CS0 of IC2 to prevent both ICs being selected at the same time. The only time that this conflict could occur is when both A14 and A15 are high. Why not simply avoid addresses at and above C000 in the program? To understand this, it is necessary to further discuss the operation of IC1.

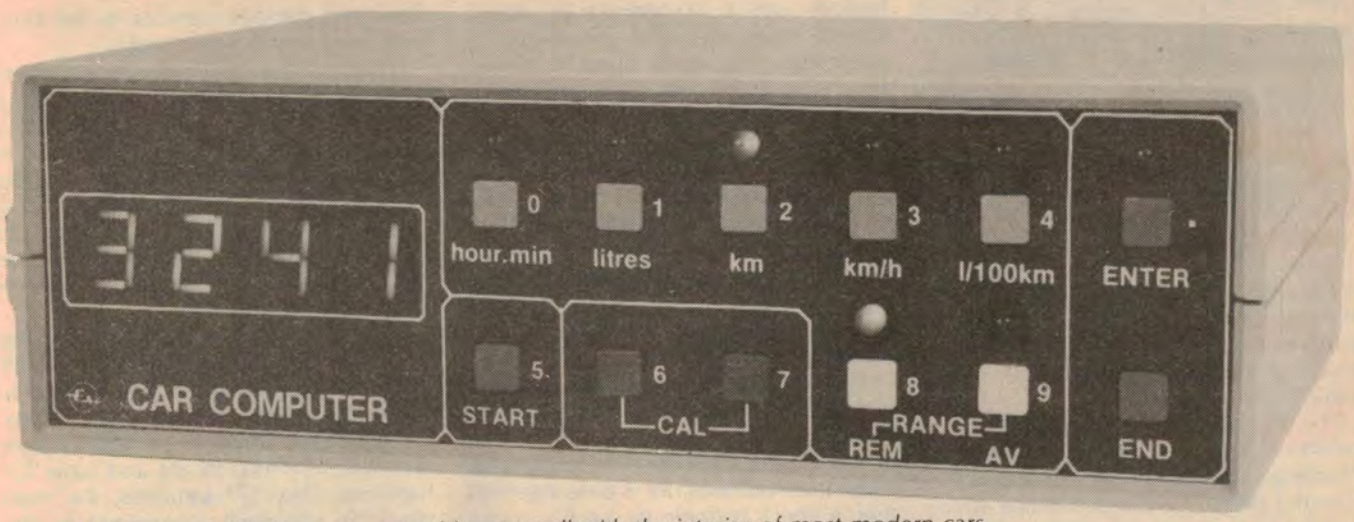


The Prince fuel flow sensor delivers 130 pulses per 0.1 litres of fuel flow.

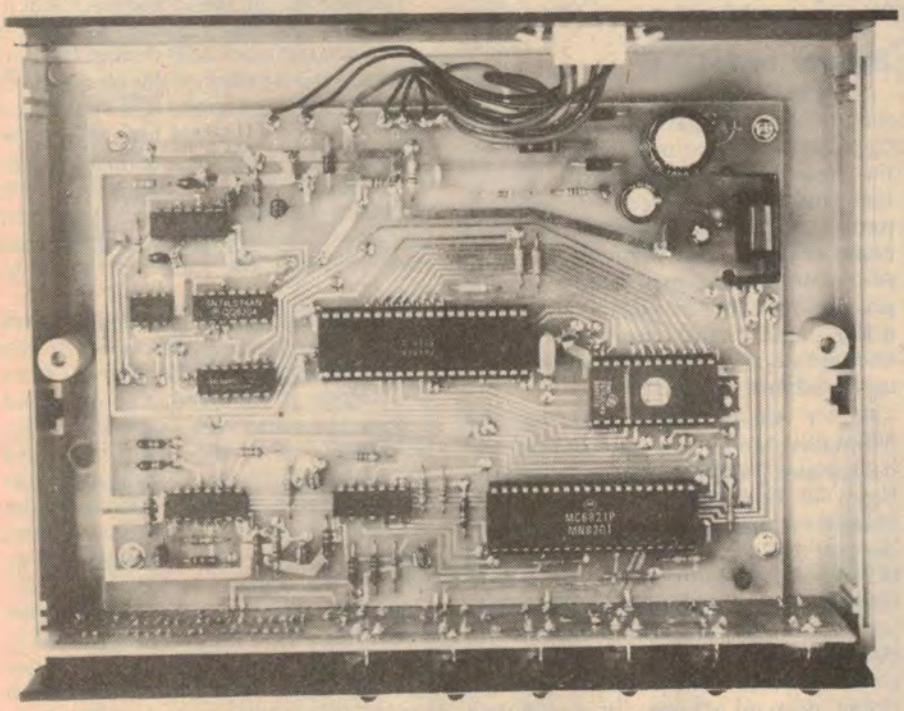
The alternative Moray fuel flow sensor delivers 1500 pulses per 0.1 litres, and is the unit we recommend.

Three programming levels are used in the Car Computer: RESET, Non Maskable Interrupt (NMI), and Interrupt Request (IRQ). The initiation of each of these programs is determined by the voltage levels (or edge triggering in the case of the NMI) on the respective hardware pins of IC1, pins 40, 6 and 4.

We shall discuss how the voltage levels at these pins are controlled and why we have used these levels of programs at a later stage. At present it is sufficient to say that to access the start address in the EPROM for these programs, IC1 looks at addresses between FFF8 and FFFF. This means that both A14 and A15 will be



Just 3241km to go! Car Computer should mate well with the interior of most modern cars.



View inside the assembled Car Computer. IC sockets are mandatory for the three main ICs, optional for others. Note clip-on heatsink fitted to the 7805 regulator.

high. Since we only want to access IC3 and not IC2, IC2 is disabled by the $\overline{CE}$ to CS0 connection.

The advantage of using these program levels is that each has its own priority. At first power on, the Car Computer runs the "background" program initiated by the power-on RESET. This program continues to run until interrupted by either the $\overline{NMI}$ or $\overline{IRQ}$. If the $\overline{IRQ}$ pin were to go low, the processor finishes the current instructions and begins the IRQ program routine. Once this program is complete, the processor continues with the background program as though no interrupt had occurred.

If an $\overline{NMI}$ occurs during an IRQ routine, the processor immediately carries out the NMI routine, then reverts to IRQ and finally resumes the background program. The converse is not true, however – ie, if an $\overline{IRQ}$ occurs during NMI, the IRQ program is not run until the completion of the NMI routine.

Display multiplexing

This first priority of the NMI is put to good use by using it to multiplex the display. The display consists of four common anode 7-segment display digits and eight individual LEDs. The cathode of each display digit segment is connected to the corresponding cathode on the adjacent display, while the anodes of the function LEDs are also common. In order to light the display segments and the LEDs, it is necessary to switch on driver transistors Q1-Q5 at the anodes.

Each display is lit in turn and this is repeated at such a fast rate that the eye perceives a continuous display free from flicker. PA0 to PA7 on the PIA are used to send the correct segment of the display low at the appropriate time, while PB3 to PB7 scan the common anodes of the display digits. Note that 7404 inverting buffers, IC8 and IC9, are used to drive the cathodes of the display and the bases of the anode driving transistors. These buffers are necessary because the PIA output lines are incapable of supplying the necessary current.

Note also that the key switches are tied in matrix form to the digit scanning PIA outputs. If any switch is closed, this is read as a high signal on the PB0 to PB2 lines which are programmed as inputs. The rate at which these keys are scanned, and consequently the scanning rate of the display, is determined by the Schmitt trigger oscillator IC5a, which runs at close to 600Hz.

IC6c gates IC5a, allowing $\overline{NMI}$ to only occur when both CB2 and the IC5a oscillator output are high. Normally CB2 is high; however, during the initial stages of the RESET program (background program), initialisation must be completed before the NMI is allowed to proceed. The initialisation includes setting up the PIA lines as inputs or outputs and for interrupt inputs. This can

be seen on the flowchart program beginning with RESET (Flowchart 1). Setting the CB2 line high occurs after the "B" connection point.

Flowchart 1 also shows what happens when the NMI routine begins. Firstly, leading zeros in the display memory are suppressed. Following this, a check is made to see if a switch is closed and the next display digit is lit.

The IRQ performs three functions, and is interrupted when either a time pulse, distance pulse or fuel pulse occurs. So that the processor can determine which of the three inputs actually caused the interrupt (after all the processor only has one $\overline{IRQ}$ pin), it is necessary to send the signals via the PIA. The PIA sets flag bits within the PIA registers which correspond to the input causing the interrupt. The hardware outputs, $\overline{IRQA}$ and $\overline{IRQB}$, then interrupt the $\overline{IRQ}$ pin of the processor.

CA1 and CA2 on the A half of the PIA are used to detect the pulses from the distance and fuel sensors respectively. CB1 on the B half detects the time interrupt.

When negative edges at CA1, CA2 or CB1 occur, the $\overline{IRQA}$ and $\overline{IRQB}$ signals, which are tied together, go low and trigger the $\overline{IRQ}$ input of IC1. The $\overline{IRQ}$ line only goes high when all interrupt flags within the PIA registers are cleared.

The time interrupt is derived from a 3.58MHz crystal connected directly to the crystal input pins (pins 38 and 39) of IC1. Inside IC1 is a divide by four circuit which provides the E clock (pin 37), and

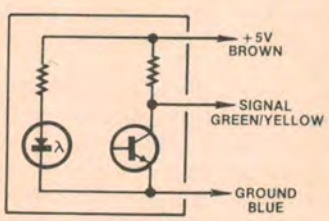


Fig. 1: MORAY FUEL SENSOR AND SPEEDOMETER CABLE DISTANCE SENSOR

this is used for timing the entire program. The resulting 894.9kHz E clock pulse is buffered by IC6d and fed to the input of IC7. IC7, an MM5369 divider, is designed to provide a 60Hz signal when operated from a 3.58MHz crystal. Since we are operating it from a crystal-controlled source of one quarter this frequency, the output at pin 1 has a frequency of 15Hz.

The 15Hz signal is connected to the data input (pin 12) of IC4b, a 74LS74 D flipflop, and is transferred to the Q output when the clock input, pin 11, goes high. This clock signal is derived from IC6c and is also connected to the $\overline{NMI}$ of IC1. The NMI routine occurs on the negative edge of the $\overline{NMI}$ clock and is completed well before the $\overline{NMI}$ clock goes high, which is the only time that the $\overline{IRQB}$ time signal can interrupt the $\overline{IRQ}$ of the processor. This arrangement is necessary since the NMI routine contains an instruction which will clear the $\overline{IRQB}$ interrupt; the very instruction which reads the key switches.

Flowchart 2 shows the logic operations which occur on an $\overline{IRQ}$. Firstly, a check is made to see if a time interrupt has occurred and, if so, the time is updated. If the interrupt was not a time interrupt then a check is made to see if it was a distance or fuel interrupt, or both. The corresponding distance and/or fuel reading is then incremented.

Sensor operation

As indicated last month, two different fuel flow sensors can be used with the Car Computer. The unit represented on the main circuit diagram is the "Prince" fuel flow sensor and consists of a ball running in a circular race to interrupt a beam of light from a small bulb to a phototransistor. Collector current for the phototransistor is derived via a 4.7kΩ resistor and the ouput signal filtered by a .01μF capacitor and squared up by Schmitt trigger IC5e.

Fig. 1 shows the circuit for the alternative Moray fuel flow sensor. This device uses multiple vanes to interrupt a beam of light between a LED and a phototransistor, an arrangement which delivers 11½ times more pulses per litre than the Prince sensor. The only change necessary to accommodate the Moray sensor is that the 4.7kΩ resistor be deleted from circuit.

The distance sensor shown on the circuit diagram consists of a coil and rotating magnet assembly. As the magnets rotate, they induce a voltage in the coil. This signal is half-wave rectified by a 1N4002 diode and filtered with a 0.1μF capacitor and 100kΩ resistor. A BC549 transistor provides the necessary gain and, after further filtering by a 0.1μF capacitor, the resulting waveform is squared up by Schmitt trigger IC5d.

The alternative speedometer cable sensor uses the same circuit configuration as the Moray fuel sensor (Fig. 1). In this case, however, the distance sensor signal is applied directly to pin 11 of IC5d and the diode, 0.1μF capacitor, and 100kΩ and 56kΩ resistors deleted. The .01μF capacitor should be left in circuit.

We'll have more to say about the fuel flow and distance sensors in the third (and final) article next month.

Power for the Car Computer is derived from the 12V car battery. A diode and 1000μF capacitor filter the battery voltage and a 7805 three-terminal regulator supplies +5V directly to the Vcc standby of IC1. Thus, power is permanently supplied to the first 32 bytes of RAM. The 10μF tantalum and 10μF electrolytic capacitors ensure stability of the regulator.

Transistor Q7 is used to switch the power to the main circuit on and off under the control of the ignition switch. When the ignition switch is turned on, current flows through an 82Ω resistor and series 1N4002 diode and turns Q7 hard on. Since Q7 saturates, the main circuit is effectively connected to the +5V output of the three terminal regulator.

With power on, the crystal oscillator in IC1 starts and the resulting $\overline{E}$ signal is applied to the clock of D flipflop IC4a. At the first positive edge of this clock, the $\overline{Q}$ output connected to the $\overline{RESET}$ and RAM Enable (RE) of IC1 is set low. When the 1MΩ resistor charges the 0.1uF capacitor at the input of Schmitt trigger IC5c, the output of IC5c goes low and, at the next positive clock transition, the $\overline{RESET}$ goes high, allowing the RESET program to begin.

When the ignition is turned off, power to the 82Ω resistor is disconnected and the current driving the base of Q7 from this source is removed. Q7 does not cease conduction immediately, however, due to the 100μF capacitor connected to its base. This capacitor can only discharge through Q7, since the associated 1N4002 diode is now reverse biased. During this discharge time, power is still applied to the circuit.

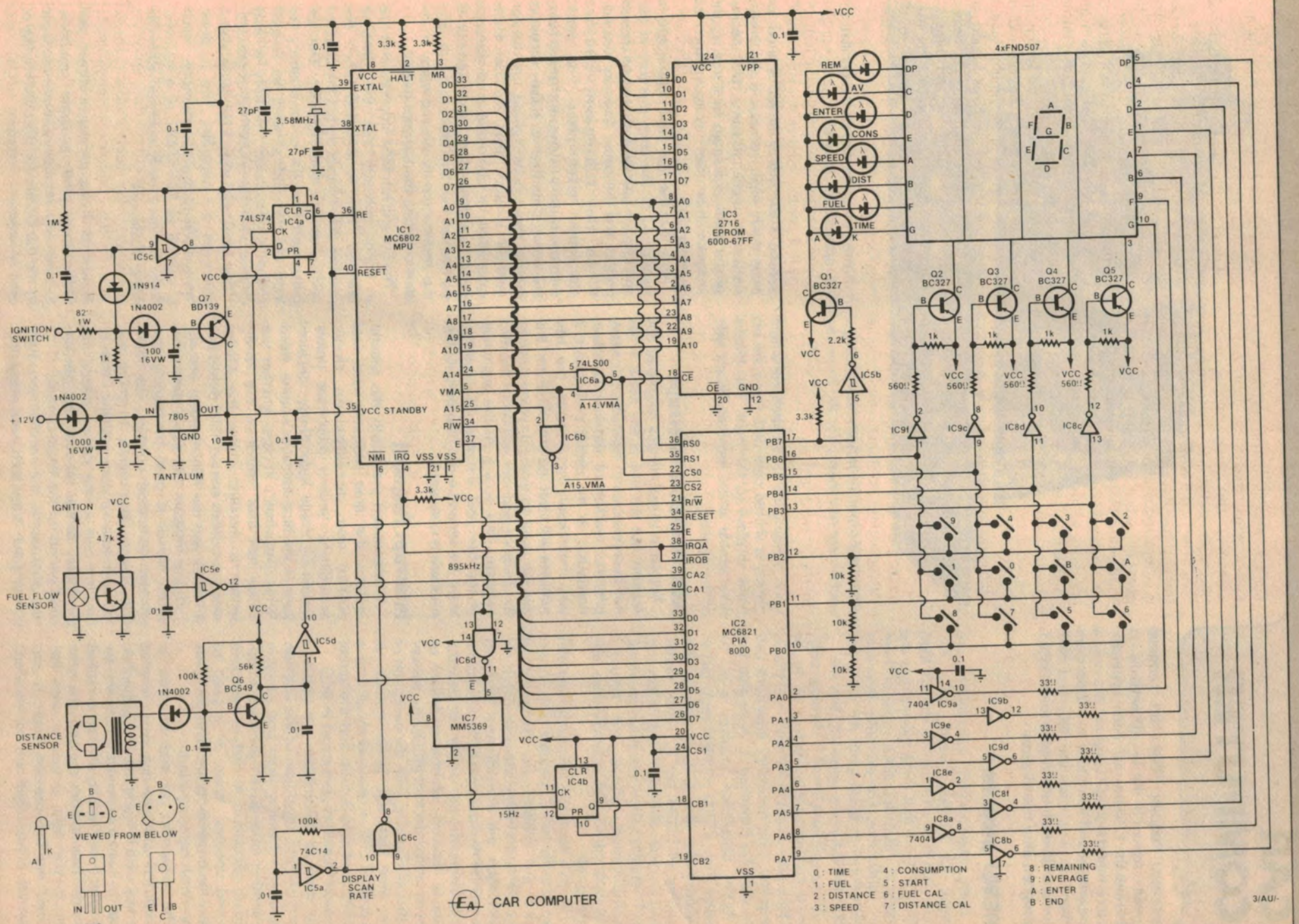
However, the 0.1μF capacitor at the input of IC5c is rapidly discharged at the moment of switch off via a series 1N4148 diode and 1kΩ resistor to ground. This sets the output of IC5c high and, at the next positive edge of the $\overline{E}$ pulse (IC1, pin 37), both RE and $\overline{RESET}$ go low.

This power down sequence ensures that memory in the first 32K bytes of RAM (which are permanently powered) is not corrupted at this critical stage.

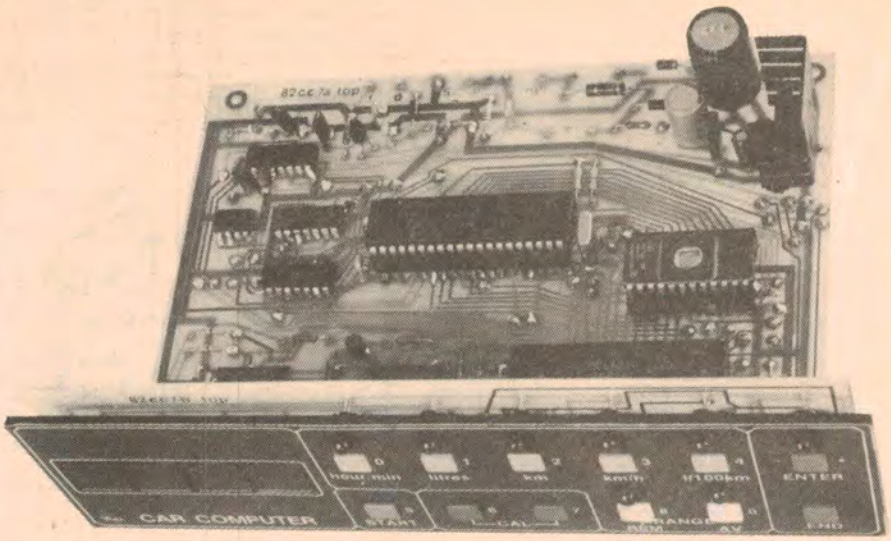
The software

Although we do not intend to completely describe the software, flowcharts have been included to explain the basic concepts of the program. A few clarifying points, however, will help in tracing through these flowcharts.

As already mentioned, there are three programs for the Car Computer: RESET, Non Maskable Interrupt (NMI) and


IC1 MC6802 MPU
IC2 MC6821 PIA 8000
IC3 2716 EPROM 6000-67FF
IC4a 74LS74
IC4b
IC5a 74C14
IC5b
IC5c
IC5d
IC5e
IC6a 74LS00
IC6b
IC6c
IC6d
IC7 MM5369
IC8a 7404
IC8b
IC8c
IC8d
IC8e
IC8f
IC9a 7404
IC9b
IC9c
IC9d
IC9e
IC9f
4xFND507
REM
AV
ENTER
CONS
SPEED
DIST
FUEL
TIME
Q1 BC327
Q2 BC327
Q3 BC327
Q4 BC327
Q5 BC327
Q6 BC549
Q7 BD139
3.58MHz
27pF
895kHz
15Hz
IGNITION SWITCH
IGNITION
+12V
1000 16VW
100 16VW
TANTALUM
FUEL FLOW SENSOR
DISTANCE SENSOR
VIEWED FROM BELOW
DISPLAY SCAN RATE
VCC STANDBY
A14.VMA
A15.VMA
0 : TIME
1 : FUEL
2 : DISTANCE
3 : SPEED
4 : CONSUMPTION
5 : START
6 : FUEL CAL
7 : DISTANCE CAL
8 : REMAINING
9 : AVERAGE
A : ENTER
B : END
CAR COMPUTER
3/AU/-

CAR COMPUTER



Repeated from last month, this photograph shows the completed PCB assembly. IC sockets are mandatory for the three main ICs, optional for others.

Interrupt Request (IRQ). The "terminal point", shown in the key of symbols on Flowchart 1, starts off each program. Note that the RESET program has only a beginning and continues in a loop from then on. It is only the IRQ and NMI programs which return back to the RESET program after completion.

RESET program

Every one second, calculations are made for l/100km, km/h, km/h AV, hour.min REM, l/100km AV and km REM RANGE. Note that these calculations are made in this order, since some of them are interactive. For example, hour.min REM depends upon km/h AV while km REM RANGE relies upon l/100km AV.

All calculations involve division only. Multiplications are in factors of 10, in which case a simple left shift is all that is required. The necessary equations can be seen on Flowchart 1. Note that the l/100km calculation involves a × 10 rather than × 100 multiplication factor because the litre CAL is actually the number of pulses per 0.1 litres rather than per litre. Although the l/100km and km/h functions show equations involving multiplication, these are actually manipulated so that only divisions are carried out.

The two equations involving hour.min are actually more complex than shown since the minutes are converted to decimal hours in the case of the km/h AV calculation and from decimal hours to minutes in the hour.min REM calculation.

For those who are wondering how the data is handled, whether in binary or binary coded decimal (BCD), the answer is that all counting, with the exception of the time interrupt pulse counts, is in packed BCD. Consequently all data is stored in the BCD form and the actual division routine is performed in BCD.

Because all calculations are updated every second, regardless of the function displayed, changing from one function to another immediately provides an up-to-date calculation. At each of these one second updates, "Mask 2" is used to allow the program to skip over the 20ms key switch debounce time.

The remainder of the program is concerned with entering data, reading the key switches and initialising the PIA. Note that when entering the data, the far left-hand (most significant) digit is loaded with the first number pressed, the next digit to the right is loaded with the second number pressed, and so on. However, if four digits are not entered, upon pressing the END switch the whole display is shifted right until there are no blank displays to the right. The blanks now appear at the most significant side of the display.

Non Maskable Interrupt

We have already briefly described this routine, but the use of the Mask 1 has not been explained. Basically, this Mask prevents reading of the key switches. Previously, in the circuit description, it was mentioned that reading the key switches will clear the time $\overline{IRQB}$ and it is this we want to avoid if, when running the distance and fuel section of the IRQ program, $\overline{NMI}$ occurs. At the beginning of the fuel and distance IRQ routine, Mask 1 is set and is not cleared until this program is complete.

Interrupt Request

As mentioned previously, this routine updates the time, distance and fuel pulses and operates on these accordingly. Several important points should be made here. Firstly, decision logic in the time interrupt routine, titled "litres CAL small", decides whether the Moray fuel flow sensor or the Prince sensor is installed. Since the Moray sensor produces around 1500 pulses per 0.1 litres and the Prince sensor around 130, the program can easily determine which sensor is used by reading the entered calibration number. The logic then directs the program to count the fuel and distance pulses for the l/100km function over a one second period for the Moray sensor or over an eight second period for the Prince sensor.

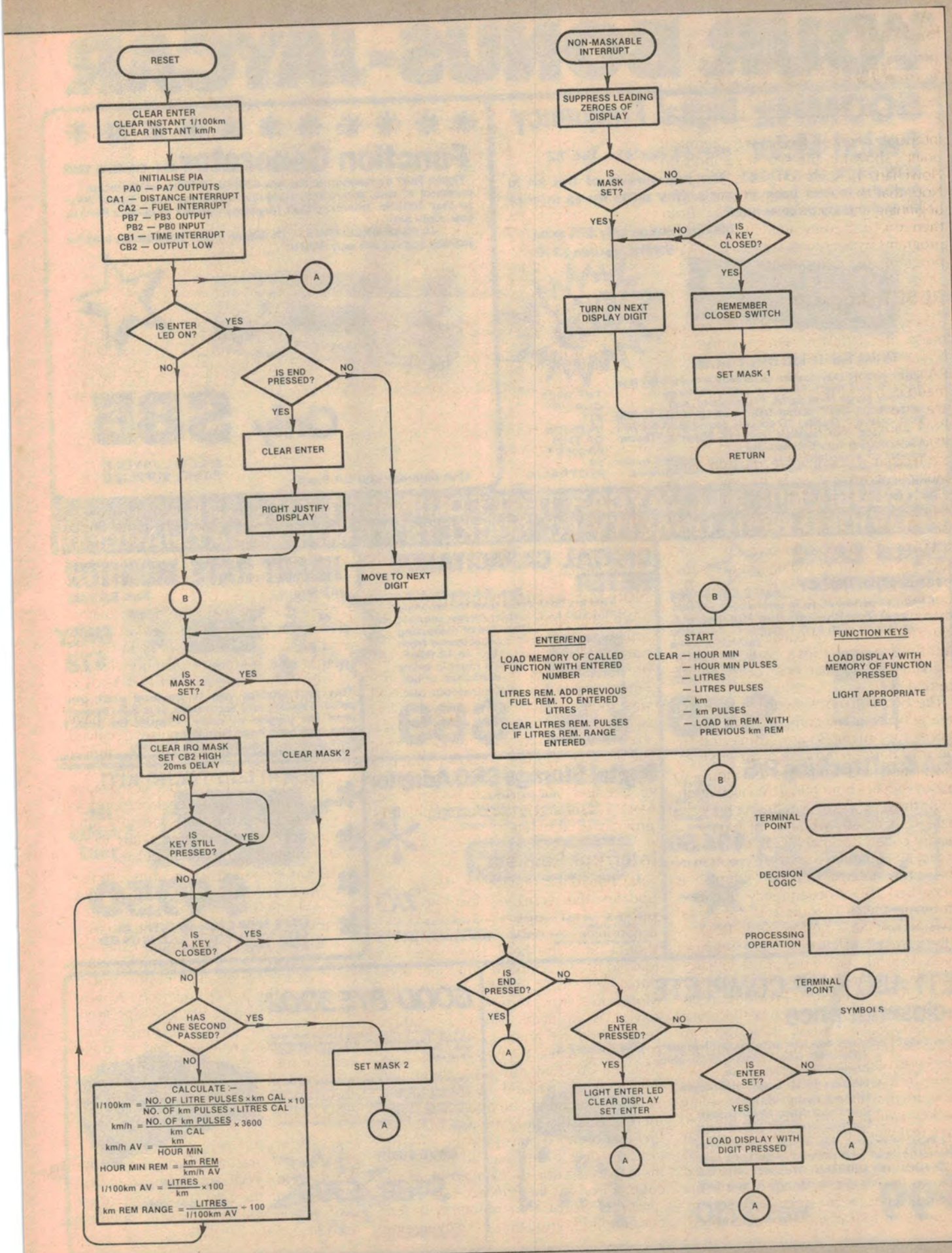
In other words, the instantaneous fuel consumption is updated once every one second if the Moray sensor is used, or once every eight seconds if the Prince sensor is used. (Clearly, the Moray sensor is the one to go for if you regard instantaneous fuel consumption as important.)

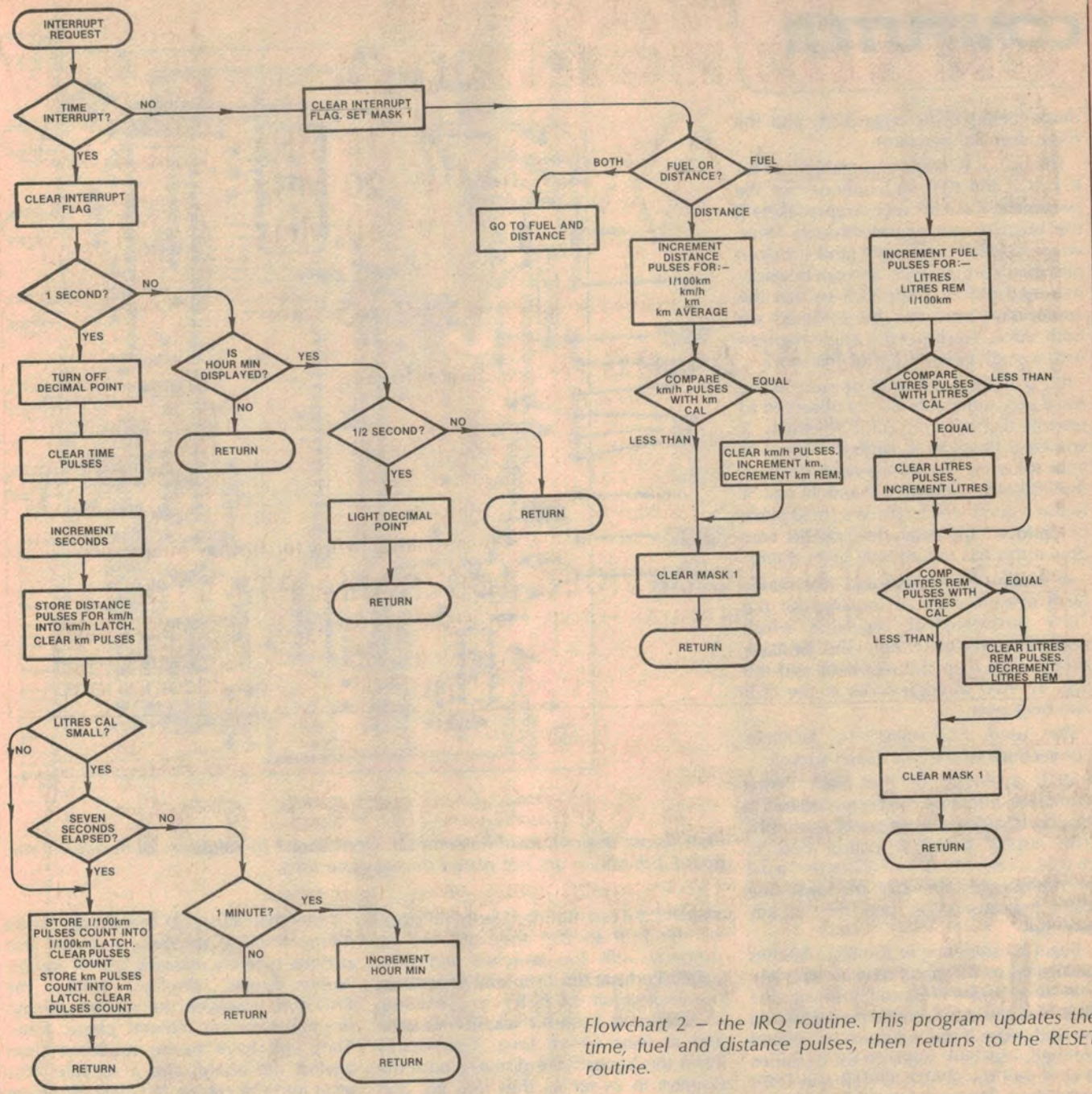
Note that the fuel flow pulse count is transferred to the latch memory at the end of each one second or eight second period, and the count memory cleared. When the subsequent l/100km calculation takes place, the latch memory is used to ensure that we have the correct number of pulses received during the count period. A similar method is also used for the km/h calculation; ie, the number of distance pulses counted over each second is transferred into the latch memory, and the count memory cleared ready to re-start counting the distance pulses.

The km and km REM functions also use a count memory for the distance covered. 1km is recorded when the number of km pulses equals the distance calibration (km CAL) number. The km pulses stored in the count memory are then cleared and allowed to re-start counting. The count memory is also cleared when the START key is pressed.

The litres and litres REM pulse count memories are separate. This is done because the litres pulse memory, which is cleared at the START of a journey, does not necessarily coincide with the filling of the tank.

Flowchart 1 (facing page) shows the RESET and NMI program routines, together with the calculations performed by the Car Computer. The NMI routine is used to update the display.


RESET
CLEAR ENTER
CLEAR INSTANT 1/100km
CLEAR INSTANT km/h
INITIALISE PIA
PA0 — PA7 OUTPUTS
CA1 — DISTANCE INTERRUPT
CA2 — FUEL INTERRUPT
PB7 — PB3 OUTPUT
PB2 — PB0 INPUT
CB1 — TIME INTERRUPT
CB2 — OUTPUT LOW
A
IS ENTER LED ON?
YES
NO
IS END PRESSED?
YES
NO
CLEAR ENTER
RIGHT JUSTIFY DISPLAY
MOVE TO NEXT DIGIT
B
IS MASK 2 SET?
YES
NO
CLEAR IRQ MASK
SET CB2 HIGH
20ms DELAY
CLEAR MASK 2
IS KEY STILL PRESSED?
YES
NO
IS A KEY CLOSED?
YES
NO
HAS ONE SECOND PASSED?
YES
NO
SET MASK 2
A
CALCULATE :—
l/100km = NO. OF LITRE PULSES × km CAL / NO. OF km PULSES × LITRES CAL × 10
km/h = NO. OF km PULSES / km CAL × 3600
km/h AV = km / HOUR MIN
HOUR MIN REM = km REM / km/h AV
l/100km AV = LITRES / km × 100
km REM RANGE = LITRES / l/100km AV ÷ 100
NON-MASKABLE INTERRUPT
SUPPRESS LEADING ZEROES OF DISPLAY
IS MASK SET?
YES
NO
IS A KEY CLOSED?
YES
NO
TURN ON NEXT DISPLAY DIGIT
REMEMBER CLOSED SWITCH
SET MASK 1
RETURN
B
ENTER/END
LOAD MEMORY OF CALLED FUNCTION WITH ENTERED NUMBER
LITRES REM. ADD PREVIOUS FUEL REM. TO ENTERED LITRES
CLEAR LITRES REM. PULSES IF LITRES REM. RANGE ENTERED
START
CLEAR — HOUR MIN
— HOUR MIN PULSES
— LITRES
— LITRES PULSES
— km
— km PULSES
— LOAD km REM. WITH PREVIOUS km REM
FUNCTION KEYS
LOAD DISPLAY WITH MEMORY OF FUNCTION PRESSED
LIGHT APPROPRIATE LED
B
IS END PRESSED?
YES
NO
A
IS ENTER PRESSED?
YES
NO
LIGHT ENTER LED
CLEAR DISPLAY
SET ENTER
A
IS ENTER SET?
YES
NO
LOAD DISPLAY WITH DIGIT PRESSED
A
A
TERMINAL POINT
DECISION LOGIC
PROCESSING OPERATION
TERMINAL POINT
SYMBOLS



Flowchart 2 – the IRQ routine. This program updates the time, fuel and distance pulses, then returns to the RESET routine.

Construction

Fortunately, construction of the EA Car Computer is a lot easier than understanding how it works. All the circuitry is accommodated on two double-sided printed circuit boards (PCBs) which are soldered together at right angles to virtually eliminate internal wiring. The completed PCB assembly is mounted in a standard Pac-tec case and fitted with a silver-on-black front panel that should mate well with the interior of most modern cars.

We understand that PCBs with plated-through holes will be available for this project, and these are well worthwhile as they simplify construction considerably. If the holes are not plated-through, you will have to solder the pads on the component sides of the PCBs as well as on the reverse sides. In this case, components such as IC sockets (wire-wrap type) and capacitors will have to sit slightly proud of the PCB so that you can gain access to the leads.

In addition, if the holes are not plated-through, you will have to insert and solder a large number of pin-throughs. These pin-throughs consist simply of a short length of tinned copper wire soldered in and then cropped close to the board. They must be inserted first, since some are beneath ICs.

Before starting construction, very carefully inspect the two PCBs for possible shorts between tracks or breaks in the copper pattern. A few minutes careful checking here could save a lot of frustration later on. Check also that the edge bus on the main PCB runs right up to the edge; if not, file the edge until it does.

The way in which it all goes together is fairly obvious from the photographs and diagrams. Start by assembling the main PCB (code 82cc7a, 171 x 123mm) according to the parts overlay diagram, making sure that all polarised components are correctly oriented. These include the ICs, transistors,

diodes, electrolytic capacitors, and the three terminal regulator.

The use of IC sockets is mandatory for IC1, IC2 and IC3, and optional for the remaining ICs. Use wire-wrap sockets if the board is not plated through. Wire-wrap sockets have longer (and stronger) pins than normal types, and can be easily mounted proud of the PCB so that the appropriate pins can be soldered on both sides. You'll need a soldering iron with a small pointed tip for this work.

IC5 (74C14) is a CMOS device, so the usual precautions should be observed to prevent damage from static electricity. If you elect to solder it, earth the barrel of your soldering iron to the earth track on the PCB and solder pins 7 and 14 first. It is also a good idea to place a small piece of opaque tape over the EPROM window if this has not already been done.

A small clip-on heatsink, Thermaloy 6038 or equivalent, is required for the 7805 three-terminal regulator which normally runs quite hot. The heatsink simply clips over the regulator and the lugs inserted through holes in the PCB and bent over.

We used PC stakes to facilitate connections to the rear panel socket.

With assembly of the main board complete, attention can be turned to the display board and front panel assembly. The display board is coded 82cc7b, measures 191 x 57mm and accommodates the LED readouts, the eight indicator LEDs, and the 12 key switches.

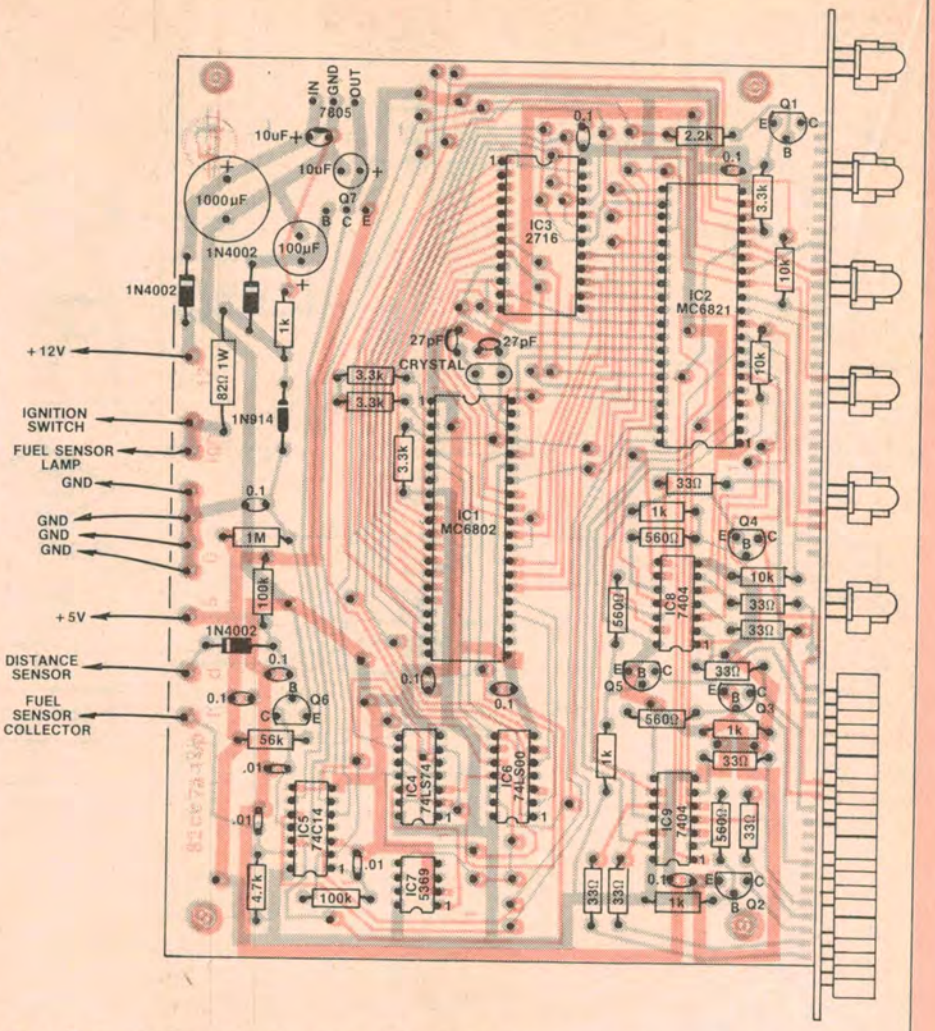
Begin by soldering in the key switches according to the parts overlay diagram. The switches are mounted flush with the PCB and the appropriate pins soldered on both sides if the holes are not plated through. Use blue switches for positions 0 to 4, red for START, ENTER and END, green for positions 6 and 7, and white for positions 8 and 9.

The four FND507 LED displays are next and must be oriented so that the ribbed edge of each display is at the top. The displays are not mounted flush but stood off the PCB so that they will line up properly with the switches and front panel. Perhaps the best way of locating the displays off the PCB is to use a strip of cardboard 1.5mm thick, 10mm wide and at least 65mm long. Temporarily insert this beneath the displays, push the displays in as far as they will go, and solder.

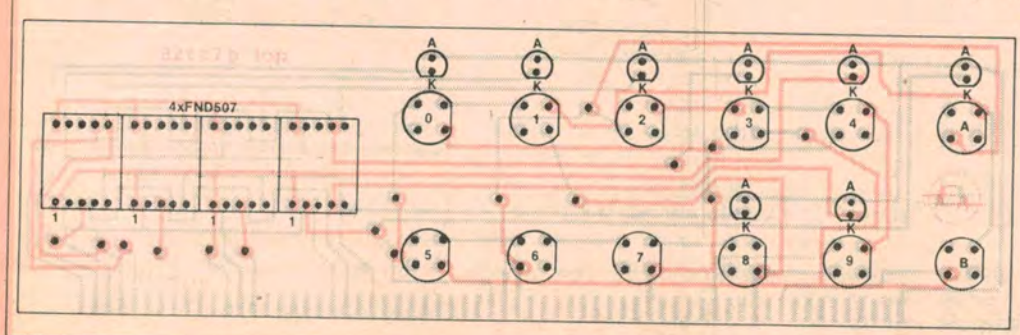
As with the key switches, some of the pins will have to be soldered on both sides of the PCB if the holes are not plated through.

Note that although the circuit shows 13mm FND507 displays, you can also use the recently released 15mm Stanley "super bright" displays. Unlike the FND507s, however, the Stanley displays do not have an integral plastic filter. They will have to be mounted flush against the board, and a suitable filter inset into the cutout. You will also have to make the cutout slightly larger.

Stanley displays are distributed by A&R Soanar and are available in three colours: NKR163 red, NKG163 green, and NKY163 yellow. Note that, in this

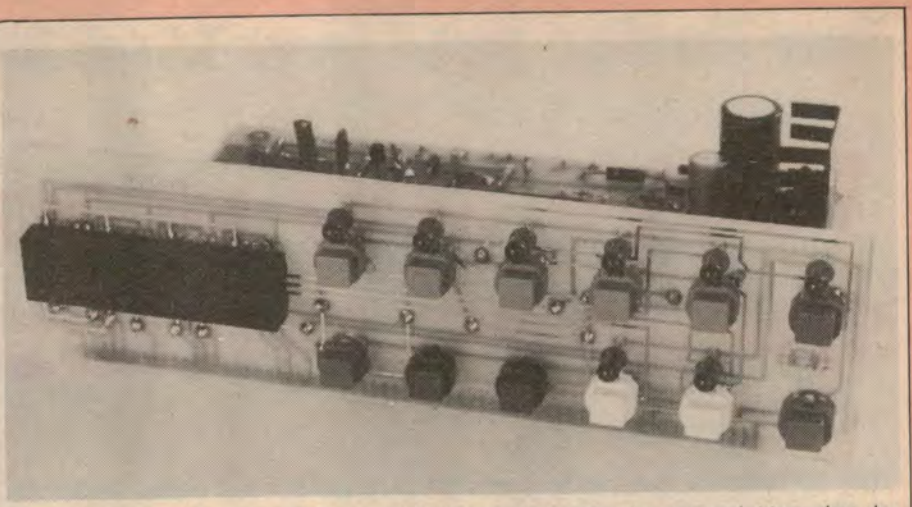


Parts layout diagram for the main PCB. Don't forget to solder on both sides of the board if the holes are not plated through (see text).



Left: parts overlay diagram for the display PCB. Make sure that you mount the four FND507 displays the right way up.

CAR COMPUTER



View showing the display PCB assembly before fitting the front panel. Note that the FND507 displays are mounted proud of the board (see text).

application, they can only be used with a plated-through PCB.

The Scotchcal label should now be carefully affixed to the smooth side of the front panel, and holes drilled and filed to shape to take the displays, LEDs and switches. Proceed carefully with this step, periodically offering the front panel to the display board so that you can judge how much progress has been made. The job is admittedly tedious, but requires care to ensure a neat finish.

In some kits, however, this work will not be necessary. At least one retailer will be supplying silk-screened and pre-punched front panels to ensure a "snazzy" job!

At this point, mount the LED bezels on the front panel and snap the eight LEDs into position. Orient the LEDs so that the anode leads are at the top, then insert all the LED leads through their respective mounting holes by carefully offering the front panel to the display PCB. Position the front panel so that it sits flush with the front surface of the seven-segment displays, then solder each LED in turn.

Check that all switches operate correctly and make any necessary adjustments before trimming the LED leads. The switches should sit about 1.5mm proud of the front panel.

As before, you will have to solder both sides of the PCB if the holes are not plated through. Install the LEDs as described above, then remove the front panel (by pushing the LEDs out of the bezels) to gain access to the two pads on the component side. Re-install the front panel when you have finished soldering.

The display PCB can now be soldered to the main PCB. To do this, slide the front panel/display PCB assembly into the retaining slot at the front of the case, and screw the main PCB to the four moulded standoffs on the base. Check that the two edge buses line up, then solder the six mating bus pads together. This done, remove the PCB assembly from the case and solder the bus pads on the reverse side.

With assembly of the PCBs complete, go over your work and check that all components are in the correct position and that all the pads have been soldered. Before actually inserting the three main ICs into their sockets, it is best to check voltages around the circuit.

Apply between 10 and 15 volts DC to the +12V and ignition terminals, and check the supply voltages on all ICs and IC sockets with a multimeter. If all is correct, disconnect power and insert the ICs. When power is reconnected the word "rEdY" should appear if the START switch is pressed. Pressing other buttons should turn on the appropriate indicator LED and bring up various numbers on the display.

Next month, we will tell you how to fit the sensors to the vehicle and describe how the Car Computer is operated.

PARTS LIST

1 Pac-tec case, 205 x 159 x 65mm
1 double-sided PCB, code 82cc7a, 171 x 123mm
1 double-sided PCB, code 82cc7b, 191 x 57mm
1 12-way Utilux line plug socket and panel plug
1 TO-220 clip-on heatsink, Thermaloy 6038 or equivalent
1 Scotchcal front panel, 192 x 59mm
12 Isostat key switches, 5 blue, 3 red, 2 green, 2 white
1 3.58MHz crystal
2 40-pin DIL sockets (see text)
1 24-pin DIL socket (see text)

SEMICONDUCTORS
1 MC6802 microprocessor
1 MC6821 PIA
1 74LS74 dual D flipflop
1 74LS00 quad NAND gate
2 7404 hex inverters
1 74C14 hex Schmitt trigger
1 MM5369 divider, 60Hz version
1 2716 2K EPROM with EA Car Computer program
1 7805, LM340T 5V regulator
5 BC327 PNP transistors
1 BC549 NPN transistor
1 BD139 NPN transistor
3 1N4002 1A silicon diodes
1 1N4148, 1N914 small signal diode
4 FND507 common anode displays, or equivalent
8 5mm red LEDs plus matching bezels

CAPACITORS
1 1000µF/16VW PC mounting electrolytic
1 100µF/16VW PC mounting electrolytic
1 10µF/16VW PC mounting electrolytic
1 10µF/16VW tantalum or low leakage electrolytic
8 0.1µF monolithic
3 .01µF metallised polyester
2 27pF miniature ceramic

RESISTORS (¼W, 5% unless stated)
1 × 1MΩ, 2 × 100kΩ, 1 × 56kΩ, 3 × 10kΩ, 1 × 4.7kΩ, 4 × 3.3kΩ, 1 × 2.2kΩ, 5 × 1kΩ, 4 × 560Ω, 8 × 33Ω, 1 × 82Ω 1W.

SENSORS (see text next month)
1 fuel flow sensor, Prince or Moray
1 distance sensor, Compucruise or Pimac
1 length of brass rod, 5mm diameter x 20mm long
1 T-junction piece to suit
1 length of fuel line hose plus clamps to suit

MISCELLANEOUS
Hook-up wire, solder, PC stakes, screws, nuts, etc

NOTE: Components specified are those used in the prototype. In general components with higher ratings can be used providing they are physically compatible.