different duty cycles, as indicated above.

IC2a simply acts as a comparator. Its pin 1 output switches low or high, depending on whether the voltage on pin 2 is higher or lower than the voltage on pin 3. The output from IC2a is then fed to RB0 via a 3.3kΩ limiting resistor. This is included to limit the current flow from IC2a when its output goes high; ie to +12V. The internal clamp diodes at RB0 then limit this voltage to 0.6V above IC1's 5V supply (ie, to +5.6V).

Note the 10kΩ pulldown resistor on RB0. This ensures that the signal on RB0 is detected as a low when pin 1 of IC2a goes low.

The A-D conversion process uses a "successive approximation" technique to zero in on the correct value. This all takes place inside the microcontroller, with the duty cycle for each successive approximation (and thus the valued stored in an internal 8-bit register) controlled by the software.

Initially, RA3 operates with a 50% duty cycle and the internal register in IC1 is set to 10000000. IC1 then checks the output of comparator IC2a to see whether it is high or low. It then adjusts the duty cycle at RA3 by a set amount, updates the register and checks the output of IC2a again.

This process continues for eight cycles, each step successively adding or subtracting smaller amounts of voltage at pin 3 of IC2a. During this process, the lower bits in the 8-bit register are successively set to either a 1 or a 0 to obtain an 8-bit A-D conversion.

Following the conversion, the binary number stored in the 8-bit register is processed (we'll look at this in more detail shortly) and converted to a decimal value so that it can be shown on the 3-digit LED display. Once again, this takes place inside the PIC microcontroller.

Note that the possible range of values for the 8-bit register is from 00000000 (0) to 11111111 (255) – ie, 256 possible values. However, in practice we are limited to a range of about 19-231. That's because the software must have time for internal processing to produce the waveform at the RA3 output and to monitor the RB0 input.

Processing the register data

OK, let's now take a closer look at how the PIC microcontroller processes the data in the 8-bit register following conversion. To do this, it requires several items of information.

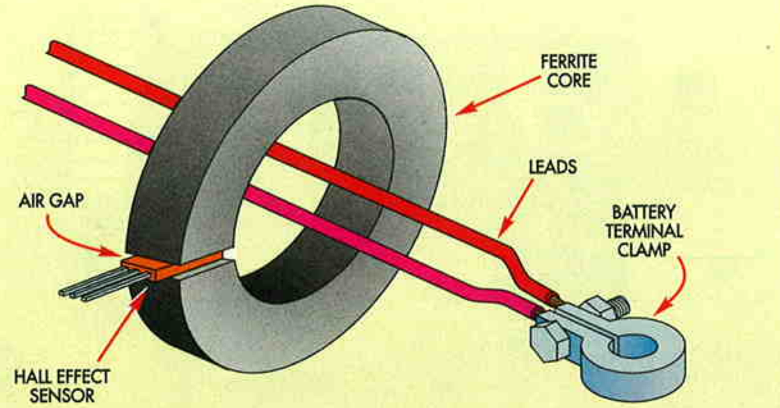


Fig.2: the current sensor consists of a ferrite core placed around the battery leads, with a Hall effect device positioned in the air-gap. A magnetic flux is induced in the ferrite when ever current flows through the leads and this flux passes through the Hall effect device which generates a proportional output voltage.

First, it needs to know the voltage produced by the Hall effect sensor when there is no current flow. This is nominally half the supply voltage (ie, 2.5V) but could be anywhere between 2.25V and 2.75V. This value is determined during the setting up procedure by installing Link 1 which pulls the RB1 line low via a 1.8kΩ resistor.

Second, the processor needs to know what the output voltage from the Hall effect sensor is for a known current. This is measured at either 17A, 25A or 30A by installing either Link 2, Link 3 or Link 4 on the RB2, RB3 and RB7 outputs.

The Hall effect device's quiescent output voltage is then subtracted from this measured value to derive a calibration number.

For example, let's say that the Hall effect sensor's output is 2.5V at 0A and 3.0V at 17A (ie, we are calibrating at 17A). In this case, the calibration factor would be 3 - 2.5 = 0.5 and this is stored by the processor along with the calibration amperage (17A in this case).

Once the processor knows this information it can calculate other currents, depending on the output from the Hall sensor. First, it subtracts the sensor's quiescent voltage from its new output voltage (note: this provides values that can be either positive or negative, depending on the current direction). The result is then multiplied by the calibration amperage and divided by the calibration factor to get the final result.

This is best illustrated by another example. Let's assume that the calibration factor is 0.5 and that the calibration amperage is 17A. Further, let's assume that the sensor output is at 3.4V. In this case, the current would be (3.4 - 2.5) x 17/0.5 or 30.6A.

This result (to the nearest amp) is shown on the LED displays and on the bargraph.

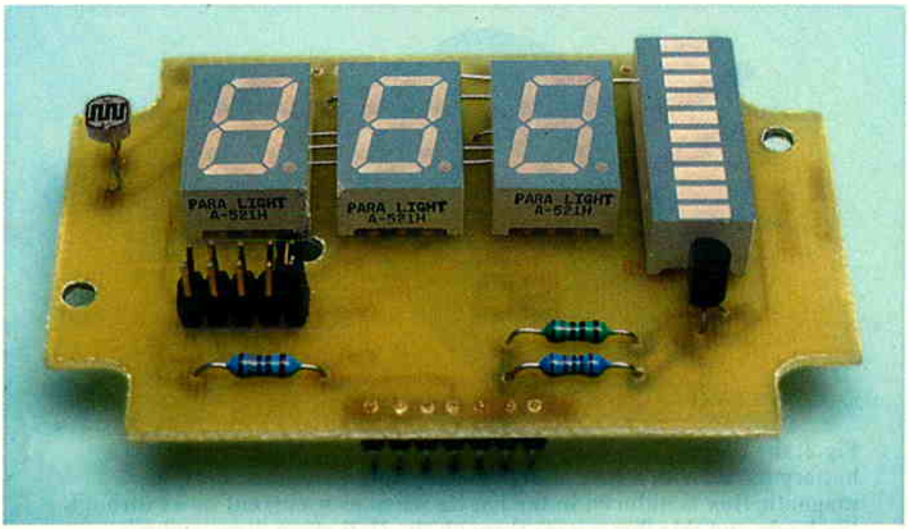
Driving the displays

The 7-segment display data from IC1 appears at outputs RB1-RB7. These directly drive the display segments via 150Ω current-limiting resistors, while the RA0, RA1, RA2 & RA4 outputs drive the individual displays in multiplex fashion via switching transistors Q1-Q4 (more on this shortly).

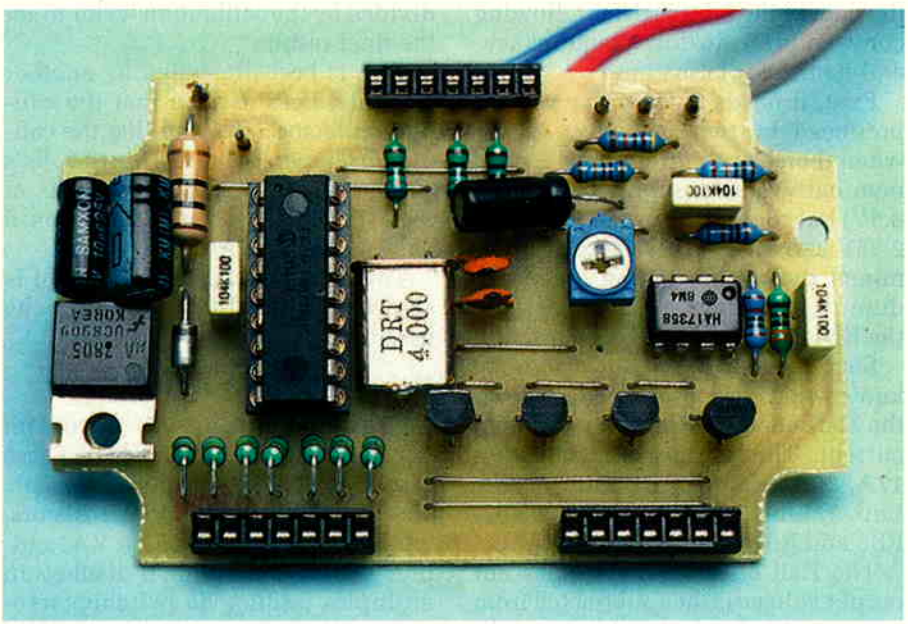
As shown, the corresponding display segments are all tied together, while the common anode terminals are driven by the switching transistors. Similarly, the cathodes of the LEDs in the bargraph display (LEDBAR1) are also connected to the display segments.

What happens is that IC1 switches its RA0, RA1, RA2 & RA4 lines low in sequence to control the switching transistors. For example, when RA0 goes low, transistor Q4 turns on and applies power to the common anode connection of DISP3. Any low outputs on RB1-RB7 will thus light the corresponding segments of that display.

After this display has been lit for a short time, RA0 is switched high and



This view shows the fully assembled display board. Note that the three 7-way pin headers are all mounted on the copper side of the board, with their leads just protruding through from the top.



The pin headers on the display board plug into matching in-line sockets on the microcontroller board. Note that the three electrolytic capacitors are mounted so that they lie horizontally across other components.

DISP3 turns off. The 7-segment display data on RB1-RB7 is then updated, after which RA1 is switched low to drive Q3 and display DISP2. RA2 is then switched low to drive DISP1 and finally, RA4 is switched low to give the LED bargraph its turn.

Note that IC1's RA4 output has a 1kΩ pullup resistor connected to the emitter supply rail for transistors Q1-Q4. This is necessary to ensure that Q1 switches off fully, since RA4 has an open-drain output.

Between driving DISP1 and the LED bargraph, the RB1-RB7 outputs are set as inputs. These have internal pullup resistors that hold them high unless pulled low via one of the links (ie, Links 1-4) and the associated 1.8kΩ resistor. By monitoring the state of these RB inputs, we can determine whether one of the links has been installed for calibration.

Link 1 tells the processor that the voltage from the Hall effect sensor is at the quiescent level (ie, when there is no current flow through the battery lead). The other three links set the current level used for calibration (you only have to choose one).

For example, if Link 2 is installed, the processor knows that the voltage output from the Hall sensor corresponds to a 17A current flow. Links 3 and 4 are respectively used for the alternative 25A and 30A current calibration levels.

Display dimming

Trimpot VR1, light dependent resistor LDR1 and op amp IC2b are used to control the display brightness. As shown, IC2b is wired as a voltage follower and drives buffer transistor Q5 to control the voltage applied to the

Table 1: Resistor Colour Codes

☐	No.	Value	4-Band Code (1%)	5-Band Code (1%)
☐	3	100kΩ	brown black yellow brown	brown black black orange brown
☐	1	47kΩ	yellow violet orange brown	yellow violet black red brown
☐	1	10kΩ	brown black orange brown	brown black black red brown
☐	1	3.3kΩ	orange orange red brown	orange orange black brown brown
☐	1	1.8kΩ	brown grey red brown	brown grey black brown brown
☐	1	1kΩ	brown black red brown	brown black black brown brown
☐	4	680Ω	blue grey brown brown	blue grey black black brown
☐	7	150Ω	brown green brown brown	brown green black black brown
☐	1	10Ω	brown black black gold (5%)	not applicable

Table 2: Capacitor Codes

	Value	IEC Code	EIA Code
❑	0.1µF	100n	104
❑	15pF	15p	15

emitters if the display driver transistors (Q1-Q4).

When the ambient light is high, LDR1 has low resistance and so the voltage on pin 5 of IC2b is close to the +5V supply rail delivered by REG1. This means that the voltage on Q5's emitter will also be close to +5V and so the displays operate at full brightness.

As the ambient light falls, the LDR's resistance increases and so the voltage at pin 5 of IC2b falls. As a result, Q5's emitter voltage also falls and so the displays operate with reduced brightness.

At low light levels, the LDR's resistance is very high and the voltage on pin 5 of IC2b is set by VR1. This trimpot sets the minimum brightness level and is simply adjusted to give a comfortable display brightness at night.

Clock signals

Clock signals for IC1 are provided by an internal oscillator which operates in conjunction with 4MHz crystal X1 and two 15pF capacitors. The two capacitors are there to provide the correct loading for the crystal, to ensure that the oscillator starts reliably.

The crystal frequency is divided down internally to produce separate clock signals for the microcontroller and for display multiplexing.

Power supply

Power for the circuit is derived from the vehicle's battery via a fuse and the ignition switch. This is fed in via a 10Ω resistor and decoupled using 0.1µF and 100µF capacitors. Zener diode ZD1 provides transient protection by limiting any spike voltages to 16V. It also provides reverse polarity protection – if the leads are reversed, ZD1 conducts heavily and blows the 10Ω resistor.

The decoupled supply is fed to 3-terminal regulator REG1 to derive a +5V rail. This rail is then further filtered using 0.1µF and 10µF capacitors and applied to IC1, Sensor 1 and the

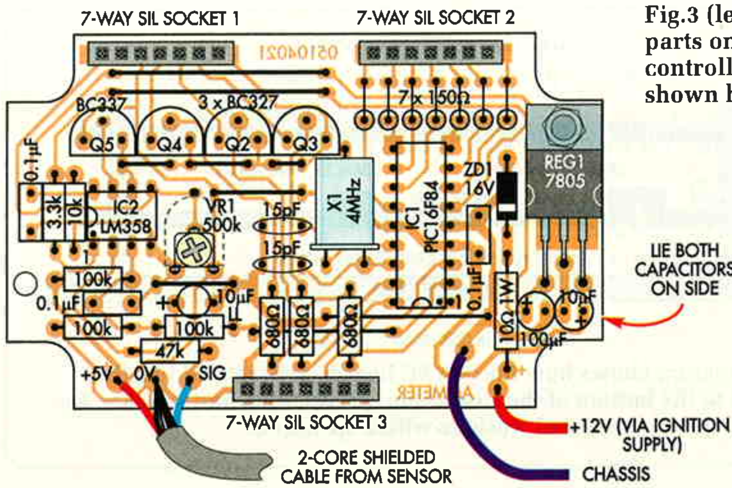
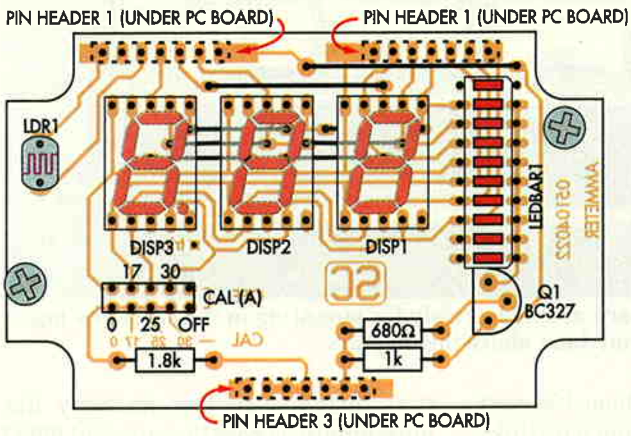


Fig.3 (left): install the parts on the microcontroller PC board as shown here.



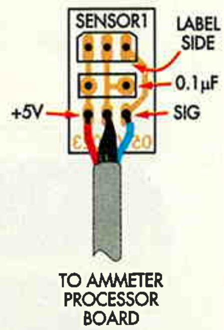


Fig.4: the parts layout on the sensor board is shown above, while at left is the display board.

collector of Q5. Op amp IC2 derives its power from the decoupled +12V rail.

Software

We don't have space to describe how the software works here but if you really must know, you'll find the source code posted on our website.

Of course, you really don't have to know how the software works to build this project. Instead, you just buy the preprogrammed PIC chip and plug it in, just like any other IC. So let's see how to put it all together.

Construction

Fig.3 shows the assembly details. Most of the work involves assembling three PC boards: a microcontroller board coded 05106021, a display board coded 05106022 and a sensor board coded 05106023. The latter carries just three parts: the Hall effect sensor (Sensor 1), a 0.1µF capacitor and three PC stakes and can be built in next to no time at all.

The assembled display and microcontroller boards are stacked together piggyback fashion using pin headers and cut down IC sockets to make all the interconnections. This completely eliminates the need to run wiring between the two boards.

Begin by inspecting the PC boards for shorts between tracks and for possible breaks and undrilled holes. Note that a "through-hole" is required on the display board to accommodate a screwdriver to adjust VR1 which mounts on the microcontroller board. This hole is just below the decimal point for DISP3 (see photo).

Note also that the two main boards need to have their corners removed, so that they clear the mounting pillars inside the case.

The sensor board can be assembled first. Install the capacitor and the three PC stakes first, then complete the assembly by mounting the Hall effect sensor. Mount the sensor with its leads at full length and be sure to mount it with the correct orientation.