AUTOMOTIVE ELECTRONICS

with MAJOR AL YOUNGER (USAR, Ret.)

Rolling your own analyser, with 'Visual Designer'

I am still getting letters on the December article, about using a PC to analyse and display ECU data, for speedy fault diagnosis. Here's the low-down on doing this using *Visual Designer*, a PC Data Acquisition/Analysis/Display/Control package which lets you achieve what's needed without having to get yourself a diploma in computer programming.

To use a PC for diagnosing faults in a modern car, the name of the game is *data acquisition*. You need a PC, the right interfacing hardware (data acquisition boards) and of course a program to let you change things (control), look at how the system responds (acquisition), and then work out what the car's system is telling you (analysis and presentation).

There are many companies in this general data acquisition business, mostly supporting the industrial or manufacturing sectors. Their products are all 'off the shelf' general purpose items, but in theory at least they should be capable of being adapted for automotive use. All that's really required is physical access to the auto's computer (the right interfacing cards and cables), plus a software package that lets you do what's needed without a lot of high-powered programming.

The interfacing cards are now fairly readily available, although they're not particularly cheap.

On the software side, perhaps the most promising developments are the recent appearance of new *Windows*-based or 'Windows like' packages for data acquisition, which let you 'program' by virtually drawing the system you want as a schematic, right on the computer screen.

To see what can be done using this approach, I decided to try out a package called *Visual Designer*, chosen at random from an advertisement in *EA*. So I simply called the company which advertised it (Kenelec), and they kindly sent me a review package containing three demo disks and a few kilos of manuals.

Visual Designer

Produced by a firm in Tucson, Arizona called Intelligent Instrumentation (part of Burr-Brown), *Visual Designer* works in the *Windows* environment, as a powerful, easy-to-use application generator for PC based data acquisition, test, measurement and control. It lets you 'create' application software customised for your own requirements. No other programs are required, nor do you have to master a programming language to make it work.

Basically it allows you to capture, record, manipulate, analyse, display, output data and control devices. Software-emulated 'display devices' available include chart recorders, panel meters, oscilloscopes, spectrum analysers and more.

To use *Visual Designer*, you need a PC with a mouse, a data acquisition board, an interface box and software (Fig.1).

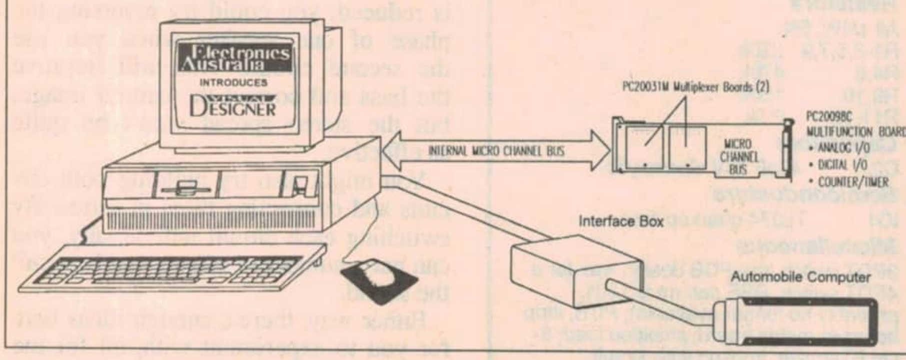

Fig.1: Visual Designer helps get your PC communicating with a car computer.

The basic system requirements to run the package are:

- IBM-compatible 80386DX/80486 PC
- 10MB free hard disk space
- VGA graphics
- Microsoft *Windows* 3.1
- 4MB minimum RAM (8MB preferred)
- A *Windows*-supported printer

Visual Designer is designed to work with a range of PC data acquisition boards which Intelligent Instrumentation also markets. For our kind of auto diagnostic work, where a total of 64 analog channels is required, you need a PC-20098C (PC98C) Multifunction Board and two 'piggyback' PC20031M (PC31M) Multiplexer Boards. The PC98C is inserted into a slot in the PC's Microchannel bus, while the PC31M's plug into the PC98C. You must write down the main board designation, in this case the PC98C, as the program will ask for board identification.

To facilitate making the physical connections between computer board's I/O channels and those of the car computer, you need an interface box. A traditional 'breakout box' (BOB) could probably be modified for this purpose, with a connector system which allows hooking-up the PC to many different car systems.

How it works

Now many of you *EA* readers may have built one of the magazine's construction projects which turn your PC into a waveform generator, digital storage scope or whatever. Well, this setup works the same way, only more so. Its combination of hardware and software gives you a precision voltmeter (analog or digital), a complex waveform generator, a CRO (cathode ray oscilloscope) or a spectrum analyser. You can even design your own custom type of tester.

When it's not in use testing cars, you

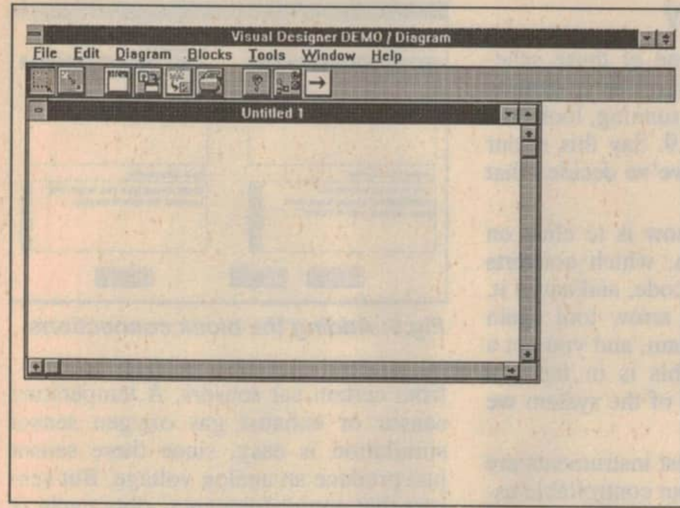

Fig.2: Visual Designer's opening screen.

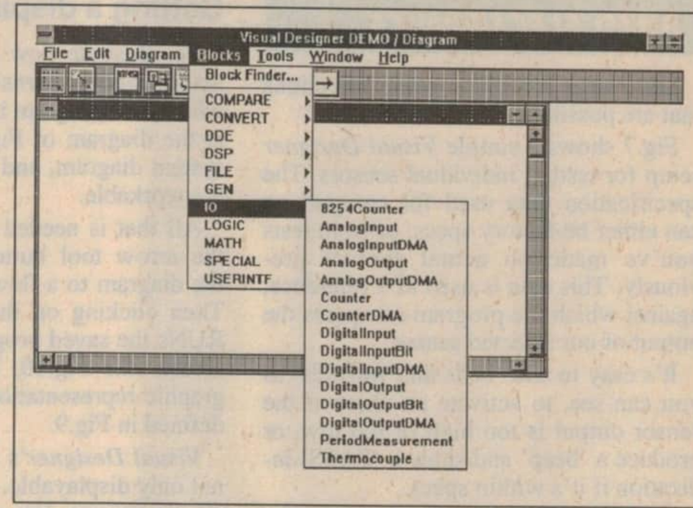

Fig.3: Starting to draw a schematic, by adding I/O blocks.

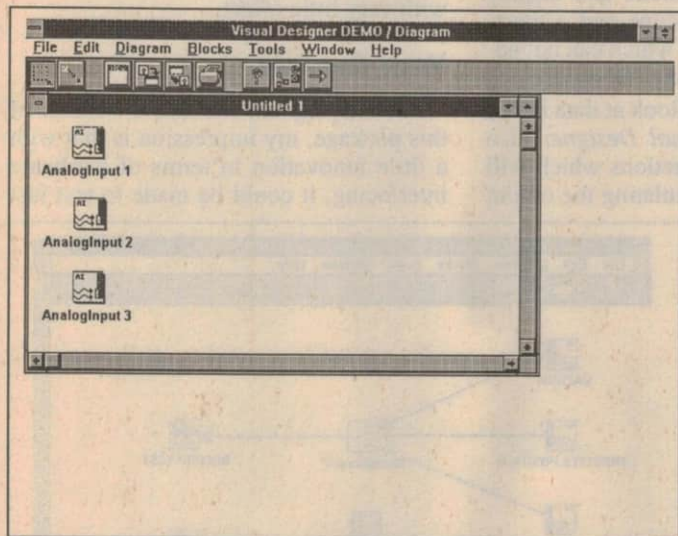

Fig.4: Three analog input blocks in place on the schematic.

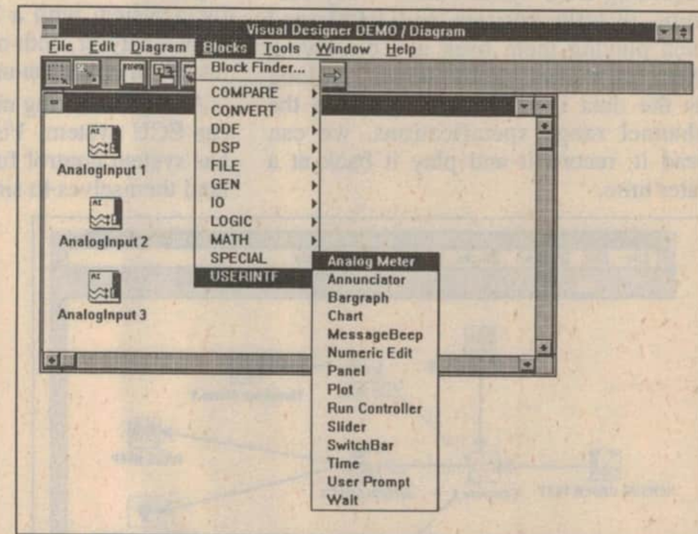

Fig.5: Now we begin to add readout device blocks...

can use it to control your home or workshop security alarm system.

But let's see how easy it is to use. The demo program requires just over six megabytes of space on the hard disk (HD) drive. The program is installed in the *Windows* environment, simply by putting disk 1 in your floppy drive A (or B), then clicking on 'RUN' in the Program Manager File menu, and typing '**A:** (or **B:**) **install.exe**'. The program is installed as a temporary directory on the HD drive. This allows you to completely delete the program without leaving files or corrupting your INI files. As the program is installed it's pretty dramatic, as all the component files are graphically displayed as they're installed.

Starting it up

After it's installed, you just double-click on the *Visual Designer Demo/Diagram* icon and you're off and running. When the program is ready to go, it looks like Fig.2.

To start building your diagnostic system, you just go to BLOCKS, on the top menu line, click on it with your mouse, and pull down to I/O. This gives you a second menu (Fig.3), from which you can select the type of block you want — say AnalogInput. If you do this three times, you'll have something like Fig.4.

Now let's provide some readout devices. Click on BLOCKS again, and this time select USERINTF. This gets you another secondary menu (Fig.5), from which you can select say an Analog Meter, a Bargraph and a Panel.

With this done, the next step is to tell *Visual Designer* how we want to interconnect each I/O channel to one of the readout devices. To do this, we must click on the 'arrow' button (second from the left) on the top tool bar (underneath the menu line, Fig.2), then drag a line from say the AnalogInput1 icon to the Analog Meter 1 icon. When you release the mouse button, the screen changes to a dialog box (Fig.6).

At this point you select the information which describes the input device, readout device and type of connection between them, and then click on the 'OK' button. The schematic will re-appear on the screen, with a solid arrow between the icons. This procedure is repeated to make all of your 'connections'.

To save your 'program' you click on the 'panel arrow' button, select a name, then save (to run the program it must first be saved). Then you click the arrow again. The program will promptly load and run. If there's an error the program will let you know by listing the error.

That's the basic way *Visual Designer* is used, to build a data acquisition/analysis system for this kind of work. So if you really want to know what the car computer is up to, you can go wild. For example we can read data from all the car computer pin-outs, display it and save it. Or we can generate our own testing signals, feed them out to the computer and watch how it reacts.

Let's take a look at some selections that are possible.

Fig.7 shows a simple *Visual Designer* setup for testing individual sensors. The specification data used for comparison can either be factory specs, or from tests you've made on actual sensors previously. This data is used as a reference, against which the program compares the output of our selected sensor.

It's easy to add 'bells and whistles' as you can see, to activate an alarm if the sensor output is too high or too low, or produce a 'beep' and show a 'PASS' indication if it's within specs.

Similarly Fig.8 illustrates a setup for recording ECU system data — say vacuum, throttle position and RPM, and then playing them back and displaying them on 'meters' on the screen. As long as the data in each case is within the channel range specifications, we can read it, record it and play it back at a later time.

Getting a display

To illustrate how one of these schematic diagrams turns into a panel display when your program is running, look first at the diagram of Fig.9. Say this is our system diagram, and we've decided that it's workable.

All that is needed now is to click on the arrow tool button, which converts the diagram to a flowcode, and saves it. Then clicking on the arrow tool again RUNs the saved program, and you get a screen like Fig.10. This is in fact the graphic representation of the system we defined in Fig.9.

Visual Designer's test instruments are not only displayable, but controllable using the mouse. For example Fig.11 and Fig.12 show the schematic and display for a system with a Scope and a Spectrum Analyser, both of which can be 'adjusted' in operation using our mouse.

As well as letting us look at data in our car ECU system, *Visual Designer* also has system control functions which will lend themselves to simulating the output

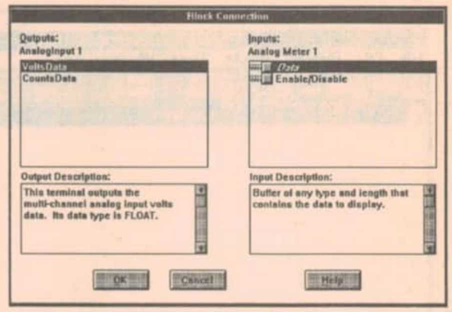

Fig.6: Adding the block connections.

from certain car sensors. A temperature sensor or exhaust gas oxygen sensor simulation is easy, since these sensor just produce an analog voltage. But sensors that output frequency, duty cycle or precise pulses could also be simulated, with very little effort.

Very flexible

After trying out the demo version of this package, my impression is that with a little innovation in terms of hardware interfacing, it could be made to test just

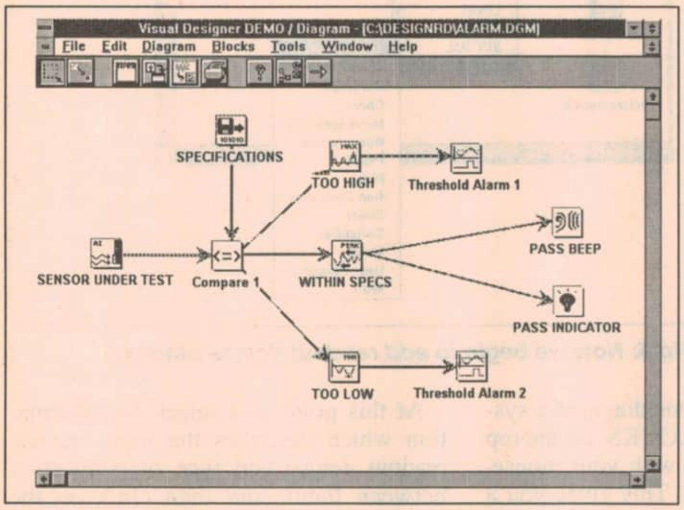

Fig.7: A simple Visual Designer setup for sensor testing.

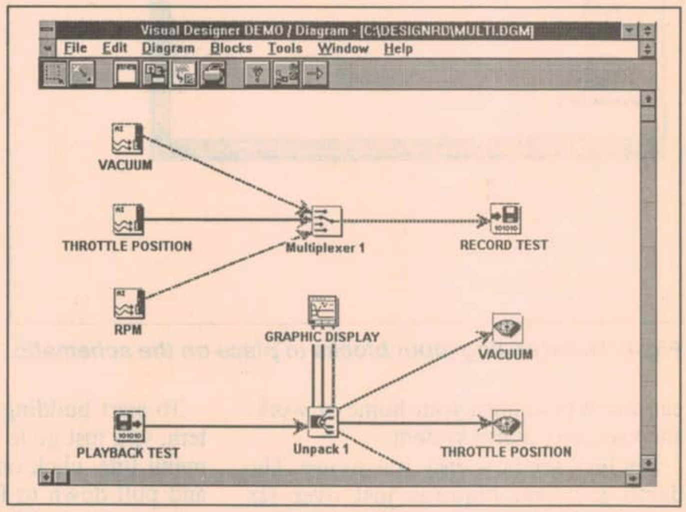

Fig.8: A setup for recording ECU data and replaying it.

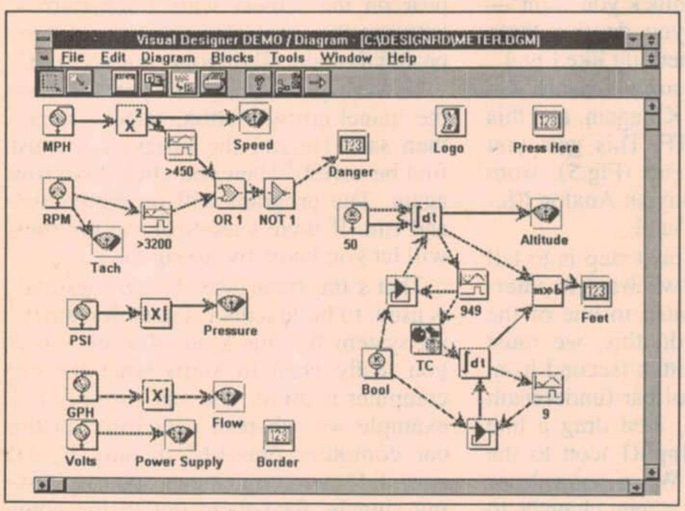

Fig.9: The schematic for a diagnostic system...

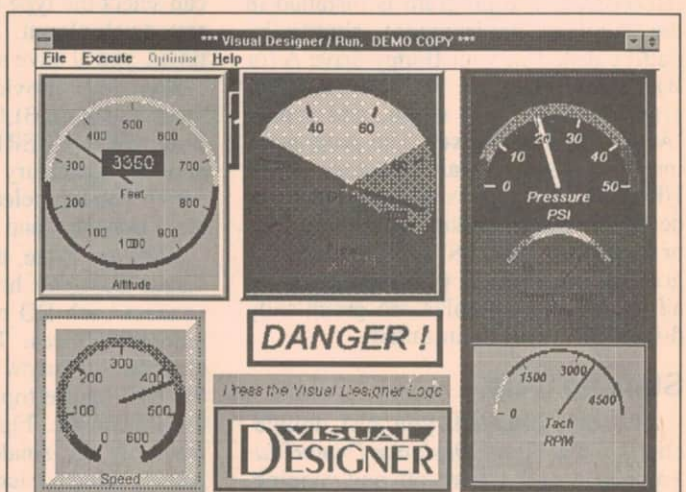

Fig.10: ...And the panel display it produces, when it runs.

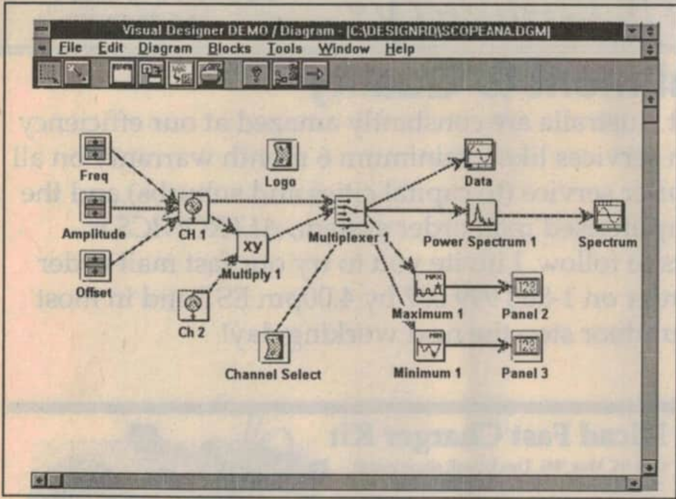

Fig.11: The schematic for a demo measuring system with a scope and a spectrum analyser.

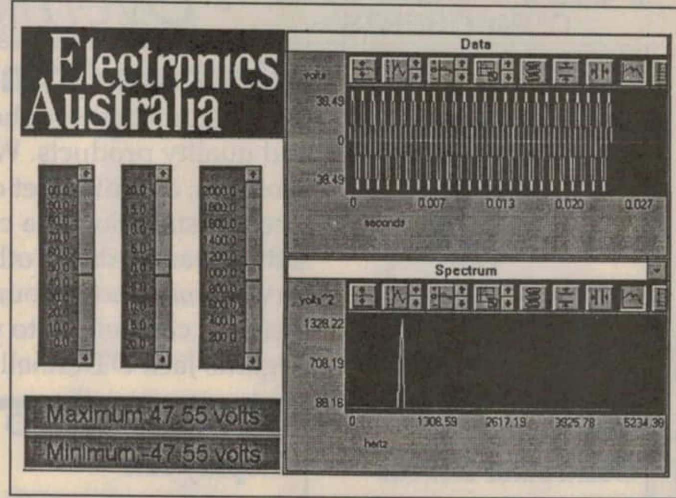

Fig.12: The display produced by the schematic of Fig.11, when it is running. Sorry about that outdated EA logo!

about everything on the car — including the ignition system. All you'd need to do is drop the signal voltages to a safe level, so as not to damage the boards.

Incidentally, some caution has to be taken when interfacing your PC to the car computer. Some circuits have high voltage components, spikes, etc., that may damage the PC's data acquisition boards. Take special care with any circuit with an inductor in it; these will almost always have spikes, and must be conditioned properly.

By the way, you'll notice that I haven't gone into the data acquisition board specifications, other than to state their capacity in terms of I/O lines. That's because in most cases their sampling rates, resolution, etc., are so high they're almost 'overkill' when you're working with an automobile system that operates in the milliseconds range.

Summary

I don't know about you readers, but I always read software books last — that is, when I'm in trouble. Mostly I just browse quickly through the books, noticing any particular cautions. With *Visual Designer* I just started in pulling down menues and having a go at it.

I did have some trouble generating suitable programs for testing. One problem, and it's not mentioned in the book as far as I can see, is if you want to label a schematic Block, you have to select and highlight it using the mouse, and just start typing. I had to call Kenelec's Sydney representative to find this out. But the technical support is on the superior side, as all my calls were answered within the hour, even when the rep was out of town.

Basically, though, with *Visual Designer* the path to designing a test setup involves just three easy steps: (1) Select, (2) Connect and (3) Manipulation (Acquire, Record, Analyse, Process, Display and Control). I'm sure a lot of you *EA* readers would therefore find it a practical way to use their PC for testing cars, given the right interfacing hardware.

What's it all cost, though? Well, at present the *Visual Designer* software package costs $1160. The matching Intelligent Instrumentation main board chosen, PC200098C, and the two 'piggyback' boards (PC200031M), will cost you just under $3000 more.

This setup covers 64 channels, which on many vehicles is an overkill. But of course many autotechs want a system that can check all the pins on a connector, even though all may not be used on all applications.

For more information on both *Visual Designer* and the Intelligent Instrumentation I/O cards, please contact Kenelec at Unit 1, 163-173 McEvoy Street, Alexandria 2015; phone (02) 550 5133 or fax (02) 550 1080. They also have a head office in Melbourne, at 2 Apollo Court, Blackburn 3130; call (03) 878 2700 or (008) 335 245, or fax (03) 878 0824.

Correction

In my review of the DEFIA handheld test instrument, in the April issue, I stated in the summary that the DEFIA would not read all IPWs (injector pulse widths) correctly.

This is not correct. I was told (after publication) this was only true of a few first units, which were returned for factory upgrade at no charge to the customer. I understand that *other* competitive units on the market may have trouble reading IPW, but *not* the DEFIA. Sorry for the mistake.

Incidentally I have also received letters asking about the power transformer I used for my Add-on Amplifier for Surround Sound construction project. I used an Antrim F2014 160VA 18VX2 transformer, from Harbuch Electronics. ❖