



# Scanning for Data

**D**o you remember when the UHF band was added to the television broadcast spectrum? As the old manual TV tuner in our photo illustrates, the new band brought roughly a seven-fold increase in the number of potential broadcast stations (up from the original 12). Later, as cable and then satellite TV gained a foothold in many American homes, the number of viewing choices multiplied yet again.

If you've ever spent a sleepless night flipping through all those channels, you'll probably agree the sheer *quantity* of TV broadcast information can be overwhelming. Views on the *quality* of that information is a discussion best left for another time. Anyway, unless you know when to tune to a particular channel for the programming you're looking for, it will take you a long time (if ever) to find it without a program guide.

The same quantity and quality problems apply when reading scandata parameters on a modern car. Although scandata have been around for at least 20 years in the automotive industry, OBD-II diagnostics has taken the quantity of available scandata to a new level. Sometimes there are so many parameters available it is hard to make sense of any of it!

That's the challenge, because interpreting scandata is a skill *every* technician should master. It's not like TV; you can't just turn it off and walk away. Although scandata can be misleading, you can't gather the same quantity of information in as short a time by any other means.

This article will give you an introduction to a few basic concepts, as well as a few tips on how to better utilize the information you collect. Call it the difference between channel surfing and data collection.

Let's start with the fundamental concept and the technology behind it. As we know, data are gathered by a control module, such as an engine control module (ECM). The ECM uses the data to calculate its command outputs — such as timing, fuel emission controls, etc. Scandata allow us to see what the ECM *thinks* it sees and then monitor what the ECM tries to do. It is important to have this concept quite clear before moving on. The scandata you see on your scanner's screen *may* reflect the vehicle's actual conditions, or they *may not*. All we know is that the ECM sent the data straight to the scan tool.

It is possible to connect a DSO to see what is happening in one circuit in much greater detail than a scanner can portray. In many cases this will be the *best* and possibly the *only* way to check that circuit. Scandata, on the other hand, allow us to view the circuit in a different way. If we check TPS voltage on a scanner, we may or may not see the actual voltage. But *we will* see what the computer *thinks* the voltage is.

For example, if the ECM sees a sharp change in the TPS voltage, it will probably increase injector pulsewidth to richen the air-fuel mixture properly for acceleration. Scandata show this as an increase in the pulsewidth command. If we checked the same two circuits, the sensor's and the command's, with a DSO and saw the TPS change, we would know the voltage had changed; but we wouldn't know whether the computer also saw the change. If we see the pulse width increase on the DSO, all we know is that the pulsewidth changed; but we don't know whether this happened because the computer *commanded* it to.

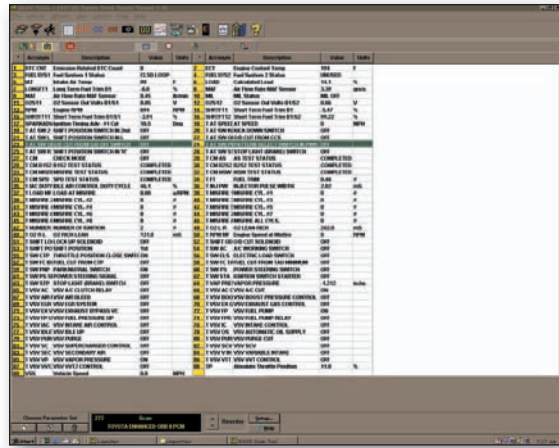
There is a very subtle difference here, but you need to understand the difference before you can rely on your diagnostic tools. When reviewing scandata, ask yourself "Did that actually happen?" If the scanner shows a misfire, did the engine really miss? If so, you have a basic engine problem — fuel, spark or mechanical. If not, you have an information problem — a software failure.

Software problems are very common, though many technicians overlook them. An example: An unlearned synchronization of cam and crank sensor signals can cause a false misfire count. Software problems are better diagnosed and solved with a scanner rather than with a DSO. The thought process required for solving scandata-detected problems are different from that for performing conventional circuit diagnosis.

### Scandata Examples

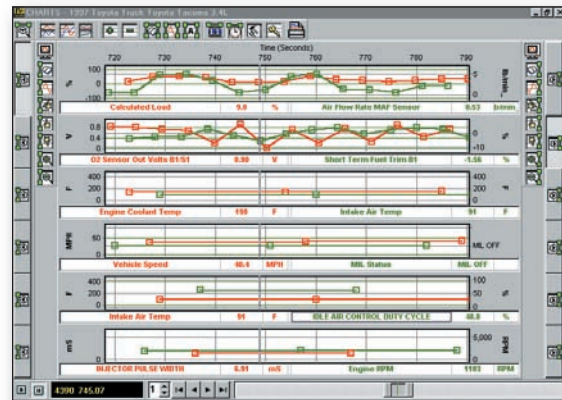
Let's take a look at some scandata and examine some tricks to locate the cause of the problem quickly. Our test vehicle is a 1997 Toyota Tacoma 3.4L automatic with just over 60,000 miles. Several manufacturers have elected to expand on generic OBD-II and offer more data parameters than just what the OBD-II regulations require. This is generally a good thing for technicians, but your first exposure to the additional data can be overwhelming. Some manufacturers have *hundreds* of parameters available, and monitoring even a small percentage would be a daunting task.

The number of parameters available depends on the vehicle and on the scanner used to communicate with the vehicle computer. For this article I used EASE software loaded on a PC, communicating with the vehicle through the computer's serial port and the vehicle's OBD-II port. I selected *Enhanced Toyota* on the scan software and entered the relevant vehicle information. I was able to read out 89 parameters for this vehicle.



**Figure 1**

The EASE software reveals more information than most people will need to analyze. The forest is so big, how can we pick out our Christmas tree? Elimination of parameters is one way to start. If you don't know what a parameter means or what it does, don't pursue that one. Concentrate on the ones you do understand first. Eliminating irrelevant parameters also speeds up the data transfer between the vehicle ECM and your PC or scanner.

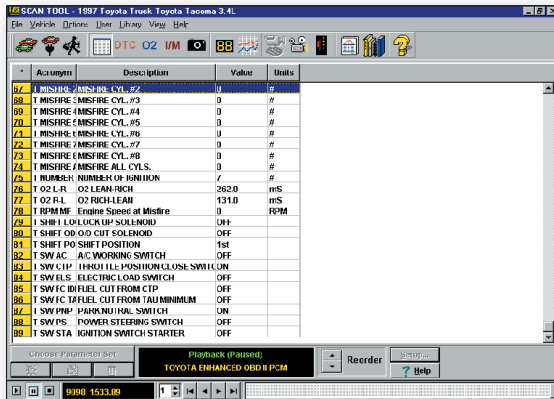


**Figure 2**

The maximum speed of data transfer depends on the vehicle's ECM. However, the scanner can optimize the data transfer rate, and it is usually beneficial to do so. The more data parameters you select through the OBD-II connector, the slower the vehicle computer can pass that information along. Most scan tools allow you to de-select parameters or select parameter groups instead of choosing the entire data set. EASE allows the selection of any one parameter or the entire set or any number in between. It also allows you to change to the time base to check your data rate.

I have changed the screen in Figure 2 to chart by time (seconds) and thus illustrate just how slow the update rate is if you select the full 89 available parameters. I also expanded the data to show how much time elapses between points. You can see not all parameters come across at the same speed. This can cause problems if you aren't careful.

Selecting just the one or two parameters relevant to your problem really speeds up the show, and this is the best trick for solving problems using scandata. A logical thought process will help you decide which parameters to select. The first step in eliminating parameters is to ignore the parameters you've never heard of. If this eliminates too many, learn more about that vehicle. Some data that does not pertain to the vehicle you are testing may also appear.

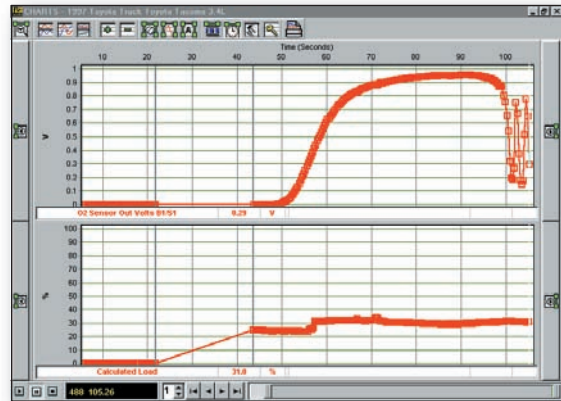


**Figure 3**

Looking at the available data for our Toyota reveals some interesting items. Number 73, 74, 86 and 89 stand out at first glance. 73 and 74 monitor cylinder misfires on a V8 and do not match the 6-cylinder engine we are testing. 86 is “FUEL CUT FROM TAU MINIMUM” and is OFF. If you don’t know what that means, turn it OFF. If you feel it has something to do with your problem, find out more about it. Number 89, ignition switch/ starter, is an example of a parameter that can be turned OFF or ignored, depending on the situation. If you are checking a problem during cranking, 89 is important. If you’re chasing a misfire or fuel-control problem, you can safely ignore parameter 89.

Using this procedure to evaluate the importance of individual pieces of scandata can quickly eliminate many irrelevant parameters and produce a set that fits your needs. You can use the opposite approach also. For example, a fuel control problem should include O<sub>2</sub> sensor scandata. You have to select at least one front O<sub>2</sub> sensor. Scrutinize each parameter carefully to determine its diagnostic relevance. Selecting the rear O<sub>2</sub> sensors may provide some needed information if your problem has to do with the catalytic converter, but if they slow down the transfer rate too much, you sacrifice utility of detail for mere quantity.

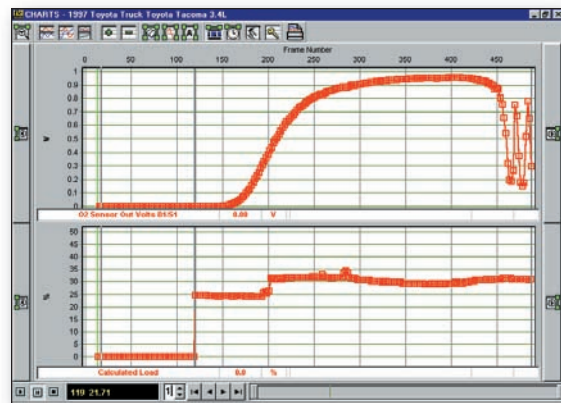
Here is a quick and dirty test of O<sub>2</sub> heater function. Start the vehicle in the morning and watch the data for O<sub>2</sub> B1S1. Compare this to load for an indication of how much the engine is contributing to sensor temperature. Now turn on the sample points and the chart-by-time option.



**Figure 4**

Notice the change in load from 0 to 25 percent and the lack of sample points between the two. Also notice that 22 seconds time have elapsed. What happened? The answer is simple, The gap in time was caused when the engine was cranked. This interrupted all communications between the ECU and the scan tool software. By the time it reconnected, 22 seconds had elapsed with no new data transferred to the scan tool.

This gap was noticeable because the EASE software allows you to plot the signal by time. When the scanner stopped communicating, there was a visible timelapse in the data. When viewing the data as text or in frames, it can be hard to see when the data is not ‘live.’

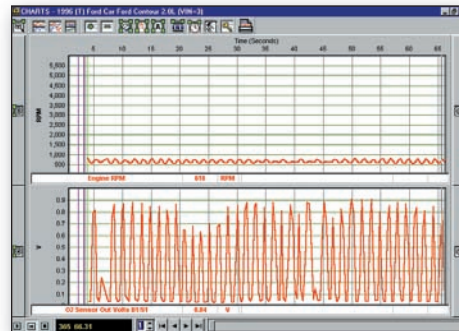


**Figure 5**

Here, the parameters are graphed in frames instead of time. We might totally overlook the missing 22 seconds, and it would appear the load changed instantly. Many scan tools won’t show this, but it is still a very important concept to understand when graphing or viewing text-based data. With this information, we can ascertain the heater circuit is functioning. The sensor began operating very quickly. The engine load was not high enough to heat the sensor immediately on its own, so the heater must be responsible.

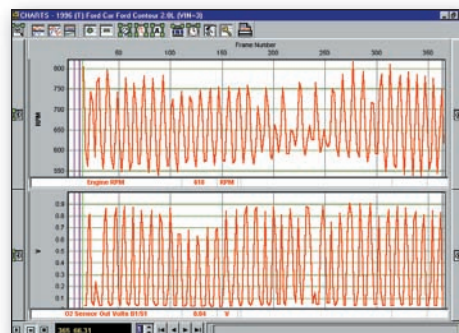
**Figure 6**

Now let's take a look at another diagnostic puzzle. I used a Mercury Mystique for this example because there's little difference between this and an import, as long as both are OBD-II vehicles. I wanted to isolate the cause of a surging idle. The first parameters I monitored were O<sub>2</sub> and RPM. Since the O<sub>2</sub> shows any fuel control problems, this was a logical place to start. Idle control and timing would also be good choices.



**Figure 7**

In this image we see the idle RPM changing right in time with the O<sub>2</sub>. The two patterns mirror each other so closely that it would be difficult to distinguish between their unlabeled shapes. The problem actually turned out to be a contaminated MAF sensor, verified using a DSO.



You might ask “How did we get from idle RPM and O<sub>2</sub> sensor voltage to replacing a contaminated MAF sensor?” Good question.

A slow O<sub>2</sub> could cause the same condition. In fact I have seen it happen on many Fords. However, the surge would be slower than what we found in our example. I knew the O<sub>2</sub> was fine on this vehicle, even though I didn't test it. Idle air control also could have caused this condition, but I chose to look at the O<sub>2</sub> reading first. I was sure it would lead me to fuel control.

If the O<sub>2</sub> hadn't mirrored RPM, I would have looked at idle control next. So I wagered first on the MAF since that is the primary fuel control input. A couple of tests confirmed this. When you know the system well enough, you can eliminate the impossibilities first. Then you're left with a small number of possibilities, and experience and your testing should lead you to the most likely cause from that group.

Scandata should be your first line of inquiry when diagnosing OBD-II vehicles. Narrowing the datastream and identifying the individual parameters you need is a skill to learn. Once you've mastered this, you'll find yourself fixing cars faster and with fewer problems than ever before.

For more information about EASE software, **Circle No. 120** on Reader Service Card. ■

— By Randy Bernklau