



Elm Electronics recommends that you always test an OBD Interface 'on the bench' before connecting it to a vehicle. This way, wiring and component problems can be caught and corrected before they can damage a vehicle, or your circuit. If you are fortunate enough to have a test ECU, or an ECU simulator, then you can functionally test the unit on the bench as well.

Testing an electronic circuit involves monitoring currents, checking voltages, checking the leds, sending queries and receiving responses, etc. Most people are very good at this, so we will not try to cover the details here. (See Application Note AN02 if you want some help with the ELM327). What this Application Note will touch on is some of the more subtle issues and problems that we have been involved with over the past few years.

J1850 Issues

Generally, there are very few problems with J1850 interfaces, but if you are having problems, and have an ELM327 circuit that does not have an R36 resistor across OBD pins 2 and 10, you may want to add it. The J1850 circuit is a very high impedance one (when the output is passive), and that may result in excessive waveform distortion if the OBD wiring presents a large capacitance. By adding R36, you help to discharge the capacitance, and so reduce the waveform distortion.

ISO 9141 and 14230

A few people have had trouble testing the ISO9141/14230 portion of their ELM327 circuit when using a 9V battery for power. In these cases, it was found that the threshold levels at pin 12 (ISO In) caused the problems, and that by changing R21 to 33K all worked well. If you are having similar problems, you may want to look at changing R21.

The other issue that comes up fairly often is when people try to test the ELM circuit by monitoring another scan tool as it 'talks' to an ECU. Usually the ELM circuit sees receive errors, and often the ECU does not respond to commands while the ELM circuit is connected. Figure 1 shows the K line portion of such a connection.

It is a very common mistake to forget that both the ELM circuit, and the OBD scan tool circuit have internal 510 termination resistors. When both circuits are connected, the effective load resistance is 255 , and not 510 as per the spec. This is beyond the capabilities of some driver circuits.

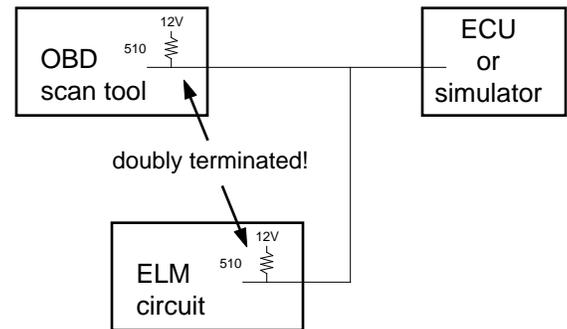


Figure 1. Monitoring a Scan Tool

If you want to monitor a scan tool that uses either the ISO9141 or the ISO14230 protocols, we recommend that you change the ELM circuit's pullup resistors from 510 to 10K .

One additional item concerns testing with ECU simulators. Recent versions of the ELM327 IC will try protocol 5 before protocols 3 and 4, as part of the automatic search procedure. This can confuse some simulators, and they will typically need about 2 seconds to recover. If you are having trouble with a 5 baud ('slow init'), this may be the cause. To allow enough time, try setting PP13 to a value of BD or greater, or upgrade to a v1.4 or newer chip.

CAN Testing

The following discusses the two areas of CAN testing that people have the most trouble with.

The first is from a misunderstanding of the purpose of the required network terminating resistors. The CAN standard specifies that every network is required to have two 120 terminating resistors, but many people think that they are only for impedance matching, in order to reduce reflections on the data bus. As such, they often feel that a functional test does not have to be perfect, and that the resistors can be eliminated. This is not correct - a resistance is always required.

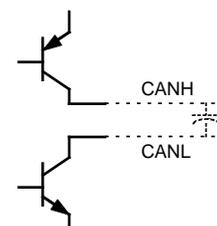


Figure 2. A CAN Output Driver Circuit



AN05 - Bench Testing OBD Interfaces

To understand why, consider the equivalent circuit for the CAN transceiver output, shown in Figure 2. When both transistors turn on, this circuit can very rapidly charge the wiring capacitance, but it has no mechanism to discharge the wiring (except for some very high value resistances associated with the internal circuitry, and that of the receivers). The 120 terminating resistors provide a low resistance path for the charged wiring to discharge through, allowing the voltage to quickly return to the passive level.

Normally, an existing CAN network has the two 120 resistors in place, and you do not even have to consider them. On the test bench, however, you may have to provide these yourself. Actually, there is no need to provide two as your wiring will be short. You could provide one resistor of 60 in order to meet the spec (but in practical terms, one resistor of 100 to 120 in value works just as well). Connect the resistor from CANH to CANL at any convenient place in the circuit.

The other main issue with CAN networks is the provision of an Acknowledge pulse (usually referred to as the ACK bit). All CAN transmitters will send a data message as a series of bits, then wait for an Acknowledge bit to be sent by at least one of the receivers. Preferably all will send this ACK bit, but as long as at least one does it means that the transmitter did its job. Figure 3 shows this pictorially.

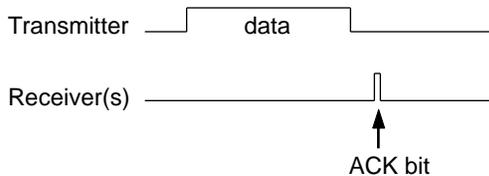


Figure 3. Typical CAN Message Transmission

In general, the ELM327 can not be relied on to send this ACK bit at all times. It will always send one when it receives a response after making an OBD request, but may not while monitoring, and usually not when idle and waiting for a command. It may help if you keep in mind that the ELM327 was designed to be connected to an active network, and not to be used to create a network with another device (so there is normally no need for it to generate these pulses while monitoring). It is only with the v1.4b firmware that we have added the ability to turn off silent monitoring and generate ACK pulses while monitoring.

If you have an ELM327 v1.4b, and wish to test your circuit with only one other CAN device, then turn off the silent monitoring with the CSM command:

```
>AT CSM 0
OK
```

This will cause the ACK bits to be generated by the ELM327 while monitoring.

If you have an earlier version of the ELM327, this convenience is not available to you, and you must take steps to create the ACK pulses for the test network. You will need a third device to generate the missing ACK bits for you. To understand why, consider the test setups that are shown in Figures 4 and 5.

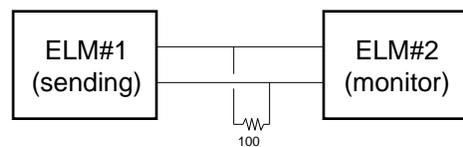


Figure 4. Connecting two ELM327s

In Figure 4, the ELM327s are both set to the same CAN protocol, with ELM#2 set to Monitor All. As a test, messages are sent by ELM#1, and the intention is to view them as received by ELM#2. Unfortunately, this will not work reliably, as ELM#2 will not generate ACK bits while monitoring. Each time you try to transmit with ELM#1, you will see a CAN ERROR message, since there are no ACK bits returned. The ELM#2 monitor may show some messages, and it may not (if you do receive anything, it will likely have <RX ERROR beside it).

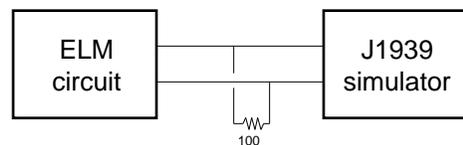


Figure 5. Testing a J1939 circuit

Figure 5 is a little more subtle. The ELM327 is connected to a J1939 simulator, and a terminating resistor is added. The J1939 protocol is predominantly one of 'broadcast messages' that are sent at regular intervals, which the simulator would begin sending immediately after power on. The ELM circuit will not generate any ACK bits for these messages, however, so none of the simulator's messages will be



acknowledged. The simulator will think that nothing is connected, and it is difficult to say exactly how it will respond. Likely, it will try to send the messages several times, realize there is a problem and then disable its transmitter, or possibly reset and try again. You may very well think that there is a problem with the simulator, when there is not.

In order to test most CAN circuits on the bench, with an ELM327 v1.4 or older IC, you must add a third device to generate the ACK bits for you! If you have a v1.4b or newer IC, then you do not need this extra device - simply turn off the silent monitoring, and the ELM327 will generate the bits for you.

ACK bits will be generated by any device that is not completely silent while monitoring. Fortunately, most commercial monitoring devices will generate the pulses for you. We use two devices for our testing - the Zanthic (www.zanthic.com) CAN4USBFX, and the Microchip (www.microchip.com) APGDT002 CAN BUS Analyzer. Both work very well, and both have their advantages. While the Microchip device does not provide as fine a resolution on the time-stamp, does not support high speed USB, and does not provide galvanic isolation, it does have a software selectable terminating resistor, and it is less expensive.

To test CAN circuits such as those shown in the previous examples, simply connect this 'monitor circuit' in parallel with the other two. Make sure that there is at least one terminating resistor at some point in the circuit, and make sure that all of the devices are set for

disconnect the USB cable, and reuse the PC elsewhere. The Microchip CAN BUS Analyzer is one device that supports this, as long as you power it from an external supply and not from the USB bus. The Zanthic device mentioned previously must remain connected to a PC for it to operate.

Several people have asked why the ELM327 does not generate these ACK pulses all the time, as this would simplify the bench testing. We have always maintained that the monitoring of data flow should be as unobtrusive as possible, and so have designed the ELM327 to always monitor silently (or at least do so by default). If there was a problem with one node on your system you might not detect it if the ELM327 always generated ACK pulses, possibly masking the fact that some were missing from the problem node. Version 1.4b of the ELM327 has added the ability to control this silent mode, which is great for bench testing and connecting to dedicated CAN ports on equipment, but silent monitoring remains on by default, keeping with our philosophy.

This Application Note has discussed some of the more common issues that we hear of when people begin to bench test their circuits. Some of the other problems and misunderstandings are discussed on our web site, on the 'help' pages, and of course, you can always send your questions to our technical support at techsupport@elmelectronics.com.

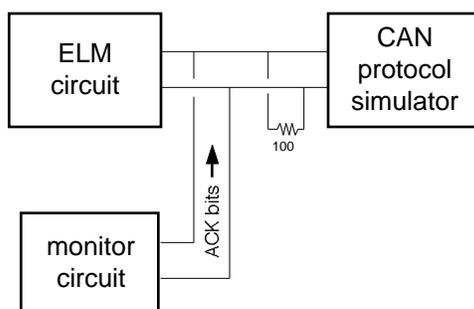


Figure 6. Typical Bench Setup for CAN Testing

the same baud rate. Figure 6 shows how the connections might typically be made.

One of the biggest problems with having to test this way, is that you may need an extra PC to control only the monitor circuit. If you choose your device carefully, you may be able to start up the monitor, then