



We are often asked how to connect our ELM327 to a Bluetooth wireless system. In addition to working at speeds of up to 700kbps, Bluetooth offers several other advantages over wired interfaces. For example, it provides galvanic isolation between the vehicle and the computer's wiring, and it also allows connections with devices that do not have RS232 or USB ports.

The following provides a few ideas on how you might add Bluetooth to your ELM327 circuit.

Using Serial to Bluetooth Adapters

There are a few RS232 to Bluetooth converters on the market that can simply connect to a 9 pin RS232 connector, and provide Bluetooth capability. Two of these devices are the Firefly from Roving Networks (www.rovingnetworks.com) and the GBS301 from IOGEAR (www.iogear.com).

Both the FireFly and the IOGEAR provide a connector for an external power source, but also allow DC to be provided through the RS232 connector (on pin 9). Your ELM327 circuit can be modified as shown in Figure 1 to provide this power, so that you do not need to connect the DC supply each time:

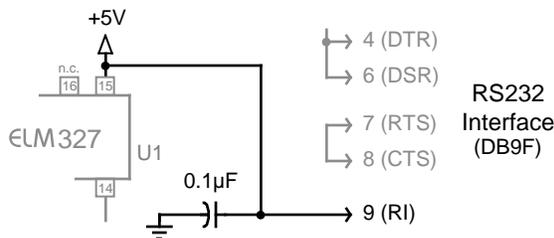


Figure 1. 5V for the Bluetooth Adapter

Note that these modules draw a significant current when operating (about 50mA), so you may need to modify your ELM327's 5V supply in order to provide the required power. If your circuit uses a small 78L05 regulator (as is shown in Figure 9 of the ELM327 data sheet), we recommend that you change it to a larger 7805 type regulator.

If you are using the RS232 interface from the ELM327 data sheet, you will find that it does not always work reliably at 115200 baud, so should only be used at speeds of 57600 or less. The IOGEAR GBS301 offers 9600, 19200, 38400, and 57600 baud rates from the dip switch, so can be easily set for 38400 or 57600 baud and connected to your ELM327 circuit. The FireFly adapter only offers 9600 or 115200 baud from the dip switch, however, which means that

you'll have to set it for 9600 baud (or learn to program it). Decide on which rate you prefer, and then make sure that you set the ELM327's pin 6 logic level to allow it.

Regardless of which Bluetooth adapter that you choose, the next step is to 'pair' the device with your computer. Follow the manufacturer's instructions on how to do this. Do not be surprised if there is no software needed, as many modern operating systems will have default Bluetooth drivers built in. Simply follow the procedure to add the device (as a 'Serial Port') and then use it as you would any physically connected serial device.

Using Serial to Bluetooth Modules

Connecting a Bluetooth adapter to the RS232 connector may be convenient, but your connection speeds are severely limited by the RS232 circuitry. If you could bypass the RS232 stage and connect the ELM327 directly to the Bluetooth transceiver, then you would be able to operate at much higher baud rates.

Several modules that contain a complete serial Bluetooth system (including an antenna) are now available in the marketplace. These provide a logic level RS232 interface for your circuitry and a standard Bluetooth protocol for 'connecting' to your controlling device. One such module is the LMX9838 from National Semiconductor (www.national.com).

The LMX9838 is a terrific little (1.0cm x 1.7cm) solution, but it needs a 3.3V supply to operate. Since the ELM327 uses 5V, it would seem that the two are not compatible. We often get questions concerning such an interface, so will discuss it here.

Interfacing to 3.3V

Connecting to a 3.3V system usually requires some form of voltage conversion. Some 3.3V ICs have inputs that are '5V tolerant' (but not all), and most 5V logic (the ELM327 included) will not tolerate 3.3V on an input, so in general a circuit is required.

You can buy 'level translator' integrated circuits to handle the voltage translation for you. Examples are the TXB0102 by Texas Instruments (www.ti.com), and the ST2129 by ST Microelectronics (www.st.com). Their connection is straight-forward as can be seen in the bubble of Figure 3.

If you wish to build your own interface circuit, it can be done with commonly available parts. Figure 2 shows one example that works well for most of the

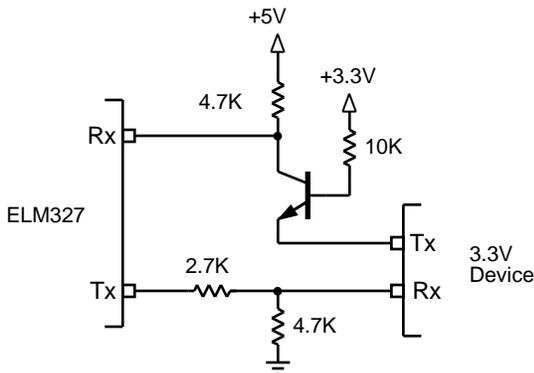


Figure 2. A 5V to 3.3V Interface

baud rates that you would encounter with an ELM327.

This circuit uses a simple resistor voltage divider to reduce the ELM327's 5V Tx output to approximately 3.3V. The 2.7K and 4.7k resistors shown work well over a wide range of frequencies while still presenting a reasonable load to the ELM327.

Interfacing a 3.3V output to the ELM327's 5V input requires a little more circuitry. Amplification is needed to increase the 3.3V level, so we add a transistor to the circuit, as shown. When the Tx output from the '3.3V Device' is above about 2.8V, the transistor will be turned off, and the ELM327's Rx input will have 5V applied to it through the 4.7k resistor. When the 3.3V output switches to a low level, the transistor will be driven into saturation and the Rx input will go to about 0V.

Since the transistor is operated in saturation, there will be a slight delay in turning off due to the charge storage. Because of this delay, the rising edge of the ELM327 Rx input will show a very slight pause in the voltage while the charge is depleted. This does not really affect the operation here, but if you wish to adapt this circuit for use at higher frequencies, it is something that you must be aware of.

Connecting to the Bluetooth module

The next page shows an LMX9838 Bluetooth module connected to an ELM327 through the circuit of Figure 2. (The ELM327 circuit is from Figure 9 of the ELM327 Data Sheet.)

The module requires a 3.3V power supply, so a voltage regulator has been added. We show an MIC5200-3.3 from Micrel (www.micrel.com), but there are many alternatives (such as Microchip's MCP1700). The regulator has been connected directly to the +5V supply, and filter capacitors have been connected to

its output. To meet the extra current requirements of the module (65mA maximum), be sure that your 5V supply uses a 7805 type regulator, and not a 78L05 type.

The LMX9838 module can be set for 9600 or 115200 baud by changing the logic levels on pins 25 and 26. Initially, set the module for 9600 baud, but make provisions for switching to 115200 baud later (as 9600 baud is too slow).

The ELM327 comes from the factory set for 9600 and 38400 baud - to provide a rate of 115200 requires changing the 38400 baud rate with Programmable Parameter 0C.

To change the ELM327's baud rate, first connect the circuit of Figure 3 for 9600 baud (see the chart for the connections). Power your circuit, and set up your PC to use Bluetooth for a serial port. Follow the Operating System's procedure to 'pair' to the LMX9838 (the code is 0000), and then connect to the ELM327 through the new serial port at 9600 baud. You should be able to perform simple commands such as AT I, even if no vehicle is connected:

```
>AT I
ELM327 v1.4b
```

Once you know that all is working well at 9600 baud, set the ELM327 for a power on default baud rate of 115200 (when pin 6 is high). To do this, assign the value 23 to PP 0C:

```
>AT PP 0C SV 23
```

and then enable the use of this new value:

```
>AT PP 0C ON
```

That's all that is required is to permanently change the baud rate. Now, change the wiring so that the ELM327 and the LMX9838 power up to the high speed baud rate. That is, connect pin 6 of the ELM327 to 5V, and connect the resistors on the LMX9838 module pins 25 and 26 to 0V and 3.3V, respectively.

Apply power to your circuit, and test the Bluetooth connection. You may have to change your software for 115200 baud instead of 9600. If all has gone well, you should now have a high speed wireless OBD interpreter.

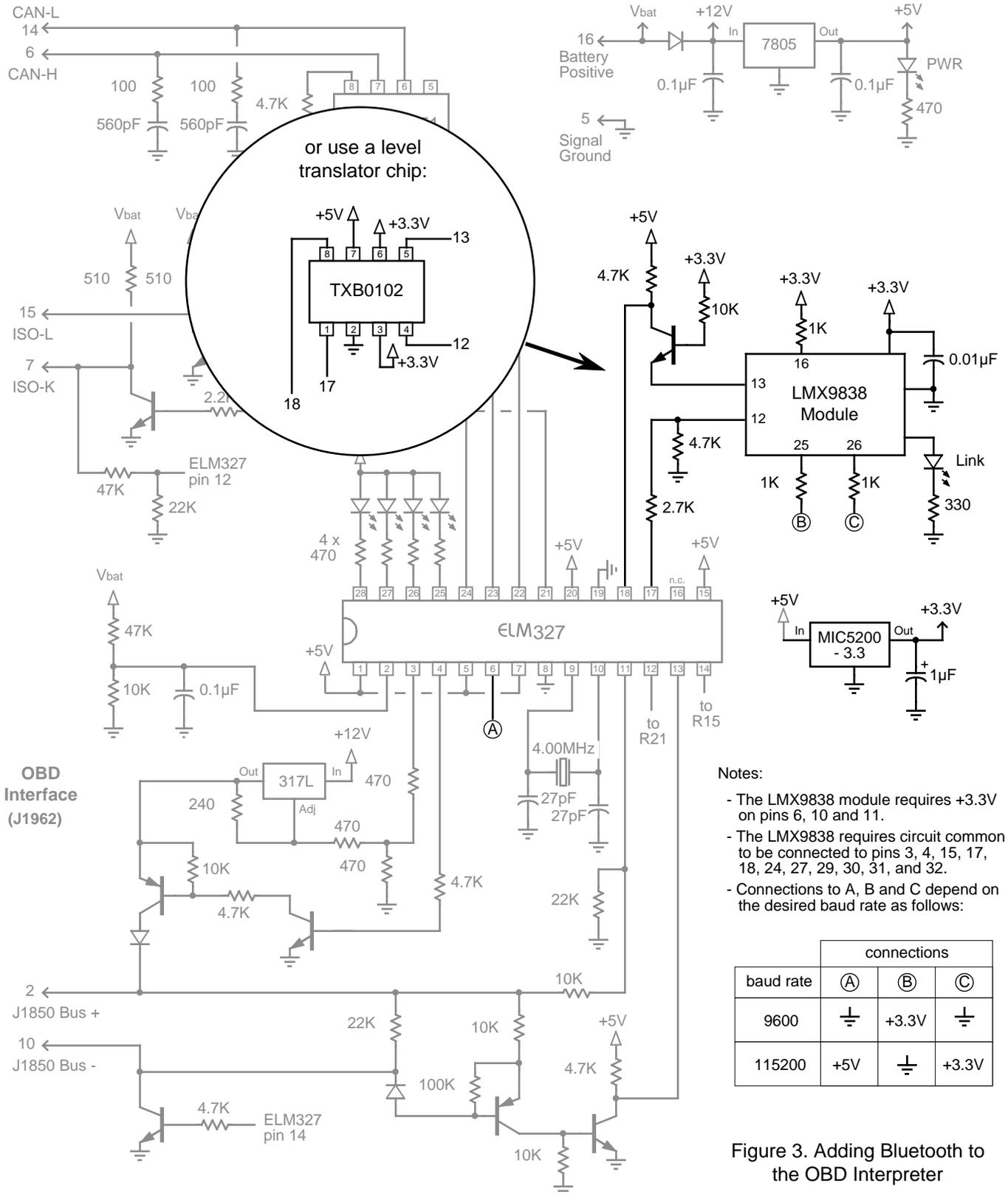


Figure 3. Adding Bluetooth to the OBD Interpreter