

PICAXE VSM Tutorial Part 3

In the third part of our PICAXE VSM tutorial we look at how to use some of the ‘virtual instrumentation’ included with the software – from simple voltmeter to advanced i2c protocol debugger!

Virtual Instrumentation

After you have designed a circuit using PICAXE VSM, you will naturally want to test it! Fortunately VSM has a wide range of ‘virtual instruments’ available for testing purposes. This article assumes you are familiar with how to draw circuits in the VSM software, having worked through the tutorials in parts 1 and 2 of this series.

Voltmeter and Ammeter

Voltage and current can be measured in two separate ways. As an example, open the ‘high command.dsn’ sample file from the /samples/picaxe/commands folder. You will then need to use the ‘File>Save As’ menu to save with a different filename so you can experiment and save modifications to it.

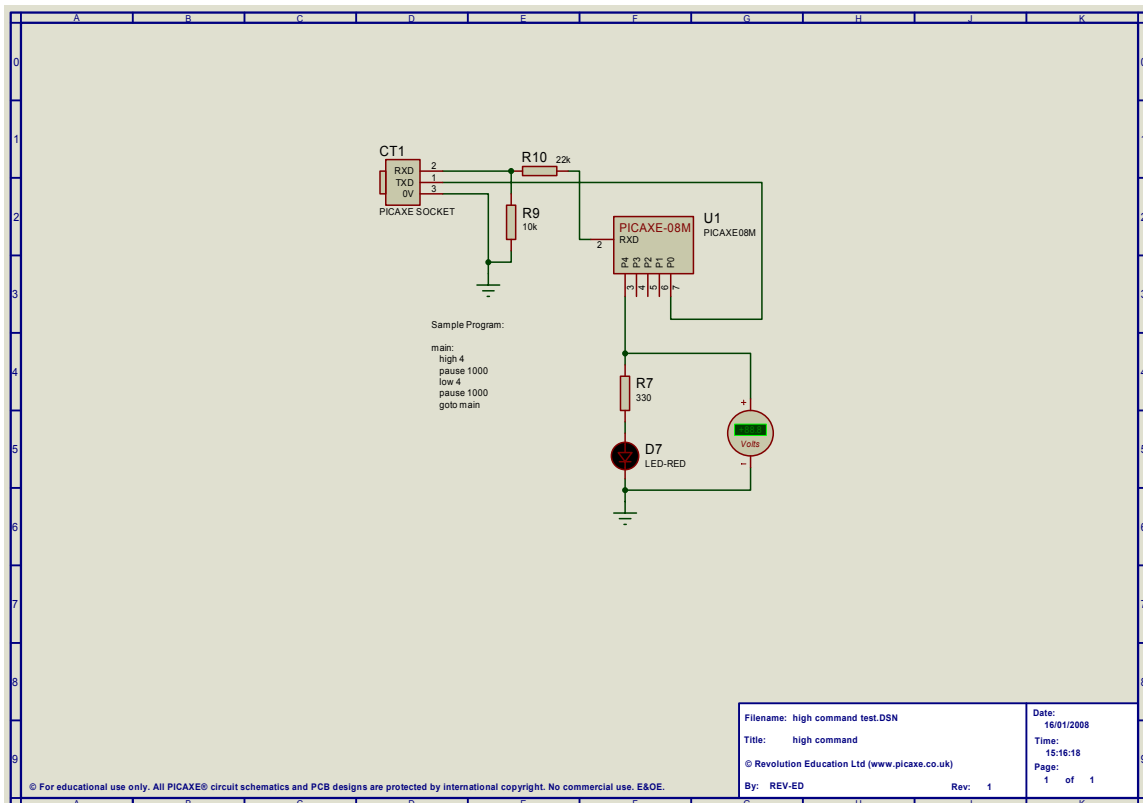


Figure 1. Simple LED circuit with voltmeter attached.

As shown in Figure 1 this circuit already has a traditional voltmeter connected across the LED. When run the voltmeter will show the voltage when the LED switches on and off.

An ammeter can be added by deleting the wire between the resistor and PICAXE pin. Right click and select Place>Virtual Instrument>DC Ammeter. Once dropped, right click on the ammeter symbol and edit the properties 'Display Range' from amps to milliamps. Then draw the two wires back in as shown in Fig. 2. When run you will now have a display of both current and voltage. If you change any of the other component properties – e.g. (changing the 'Forward Voltage' of the LED from 2V to 2.5V or the resistor from 330 to 270) you will see corresponding changes in the current.

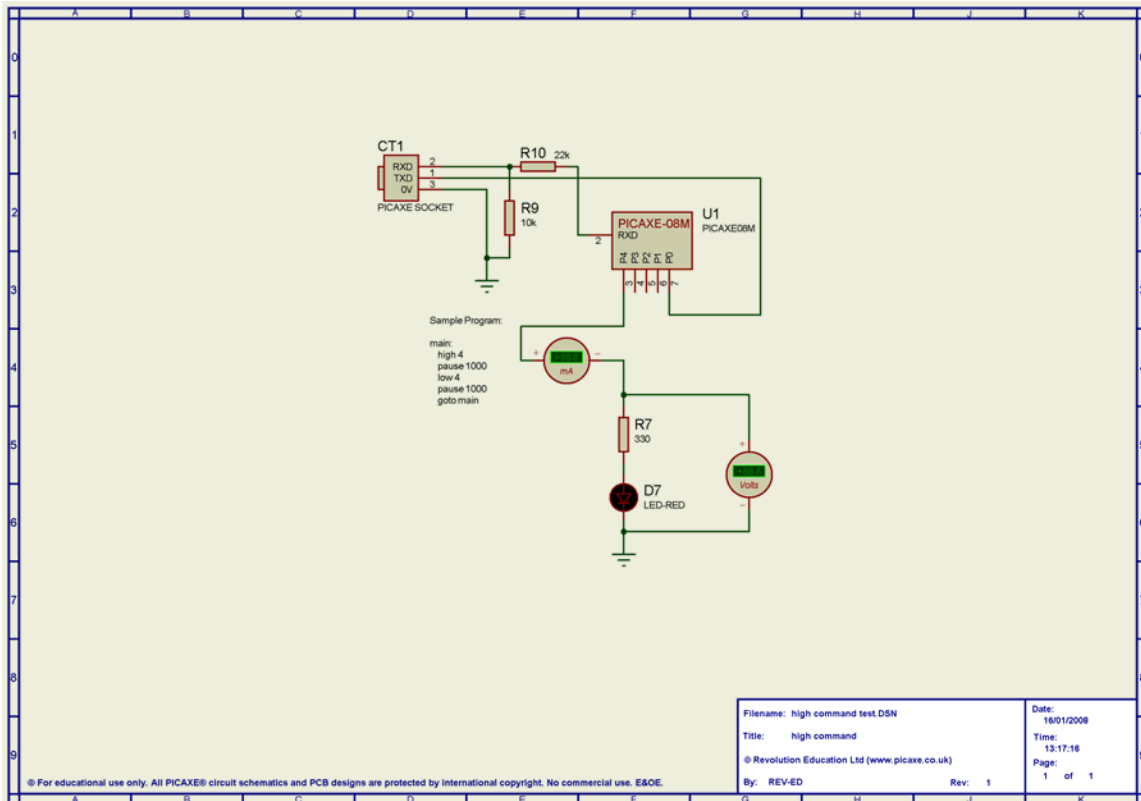


Figure 2. Virtual Ammeter added to the circuit. Remember to change to mA range!

The second, slightly simpler, way of measuring voltage and current is to just add probes onto any component pin or wire. To do this right click, select place and then either the voltage or current probes (see figure 3). Once the simulation is run the probes will show the current or voltage at the probe position.

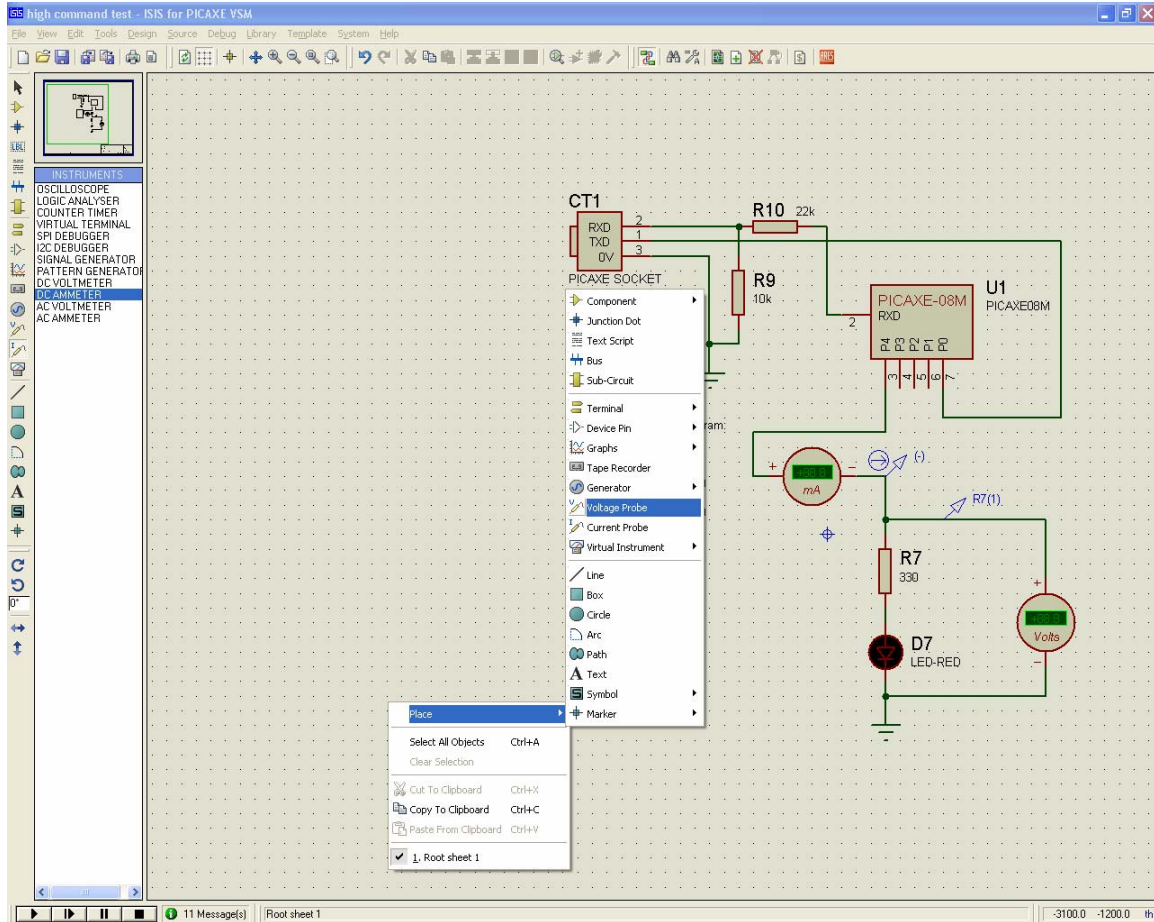


Figure 3. Right-click the mouse to add voltage and current probes.

Oscilloscope

On occasions you may wish to trace the voltage or a signal over a period of time. This is when the virtual oscilloscope is of use

This time open the ‘infracommand.dsn’ sample file and re-save with a different filename. This uses a ‘simulated’ IR LED and IR receiver (e.g. equivalent of the TSOP4840). Two channels of the oscilloscope are connected across the LED output and the receiver output.

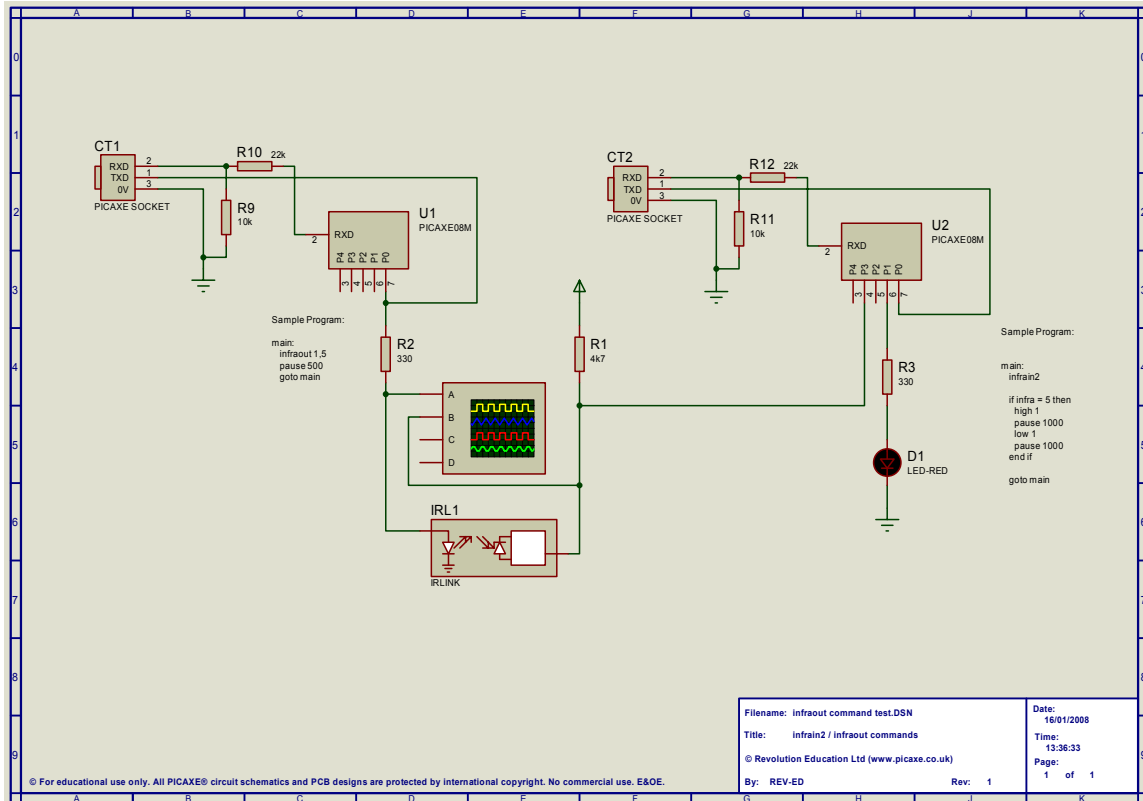


Figure 4. Two PICAXE chips communicating via a simulated infra-red link.

When run the oscilloscope display panel will appear on screen, showing the trace from the two channels. As can be seen from figure 5 the top LED trace clearly shows the modulating signal, whilst the receiver trace shows the de-modulated, inverted output! The virtual oscilloscope includes all controls that you would expect to find on a real device, and so the position, time base etc can all be adjusted as you require.

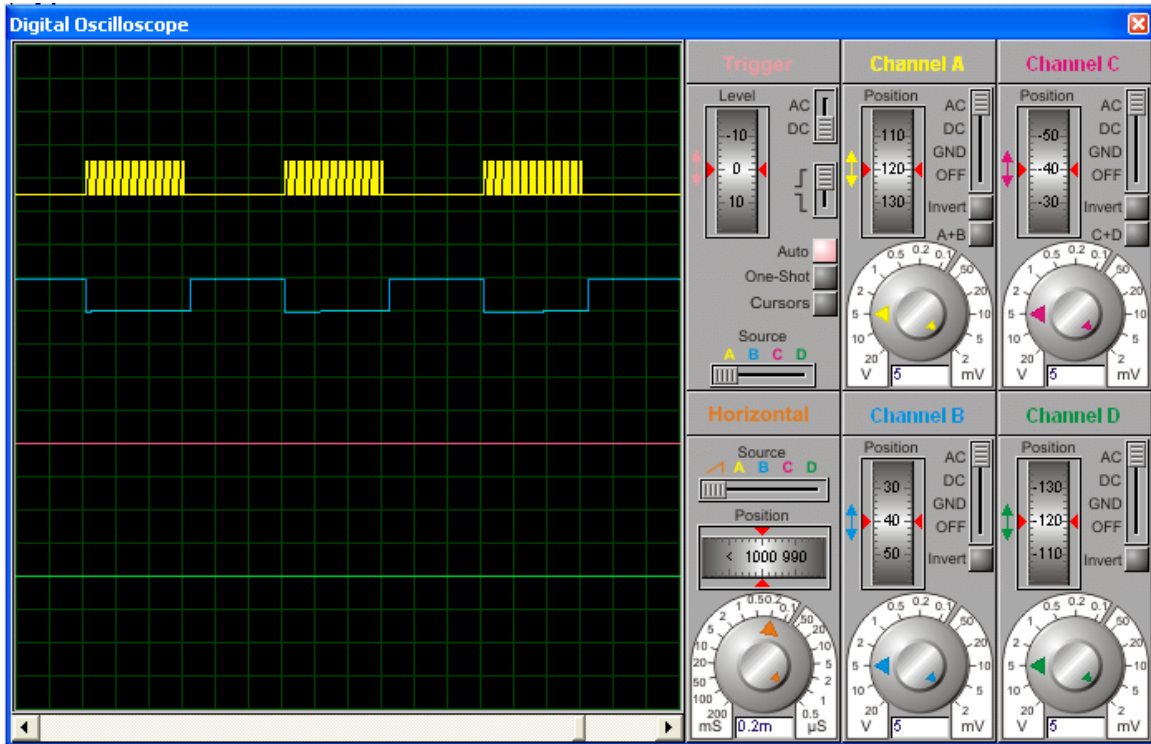


Figure 5. Virtual oscilloscope trace.

Serial Terminal

This time take a look at the ‘sertxd command.dsn’ sample file. This demonstrates the ‘Virtual Terminal’ which acts as an RS232 serial terminal for testing programs with serial data. When run, the serial data output from the PICAXE chip is displayed on screen within the terminal window.

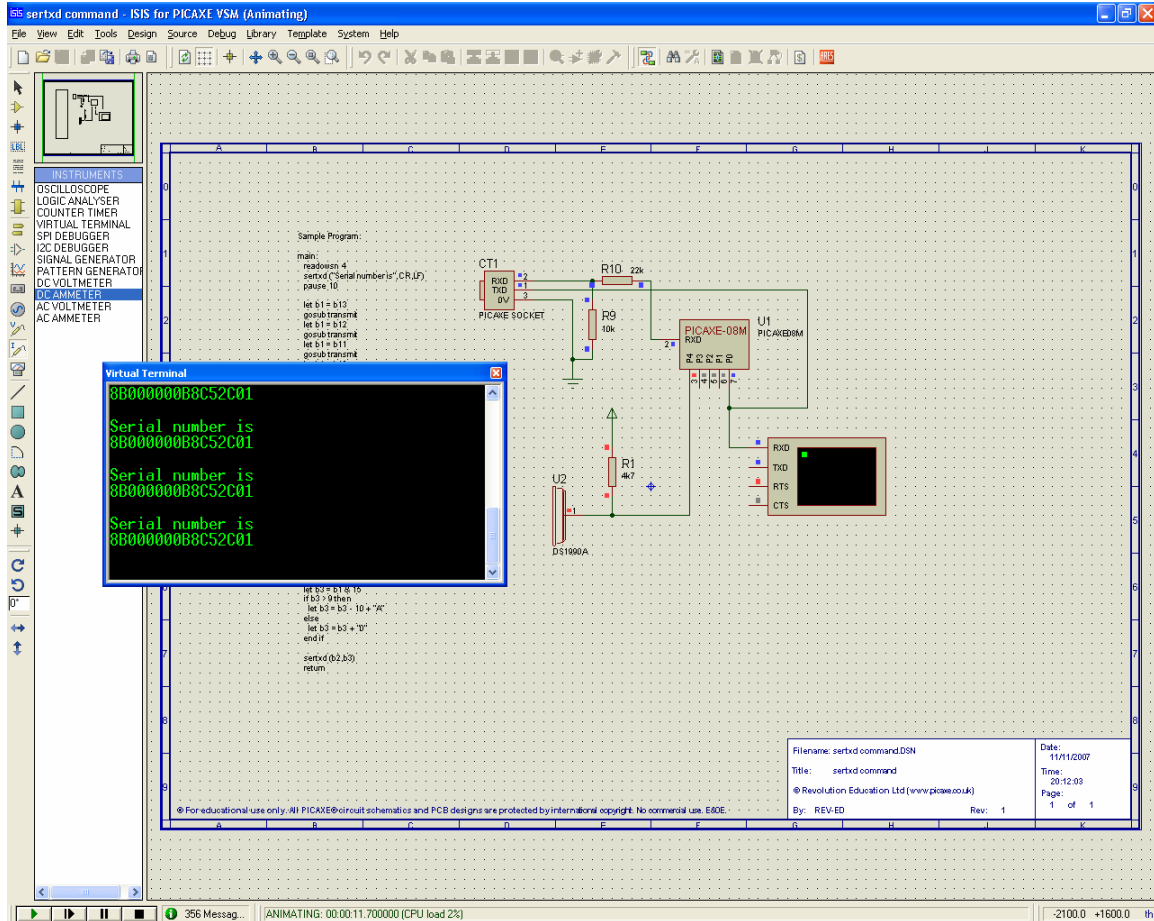


Figure 6. Serial Terminal Demonstration

As with all serial systems you do need to set the baud rate and polarity of the terminal to match the PICAXE settings (4800,n,8,1 (inverted polarity) in this case). This is carried out by right clicking on the terminal symbol and selecting 'Edit Properties'.

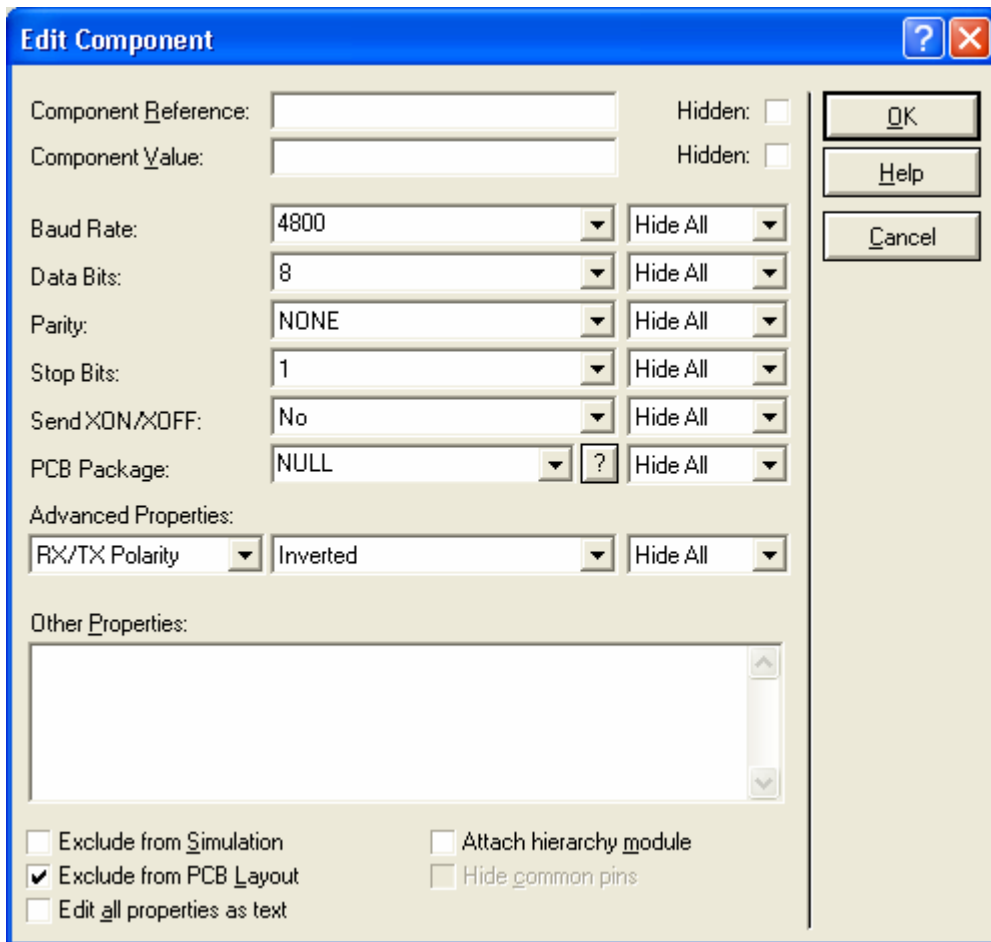


Figure 7. Remember to set the serial baud rate and other parameters, otherwise you will see corrupt characters on screen!

COMPIM

Whilst we are talking about serial communication it is also worth mentioning the COMPIM (COM Physical Interface Model). This ‘symbol’ acts as a link between the simulator and the real serial port of the computer. This, amazingly, allows connection of real serial devices to a simulated PICAXE chip!

As an example you could connect a serial GPS module to the serial port of the computer and then have the simulated PICAXE chip parse the NMEA serial data stream! The ‘COMPIM demo.dsn’ file demonstrates the COMPIM feature.

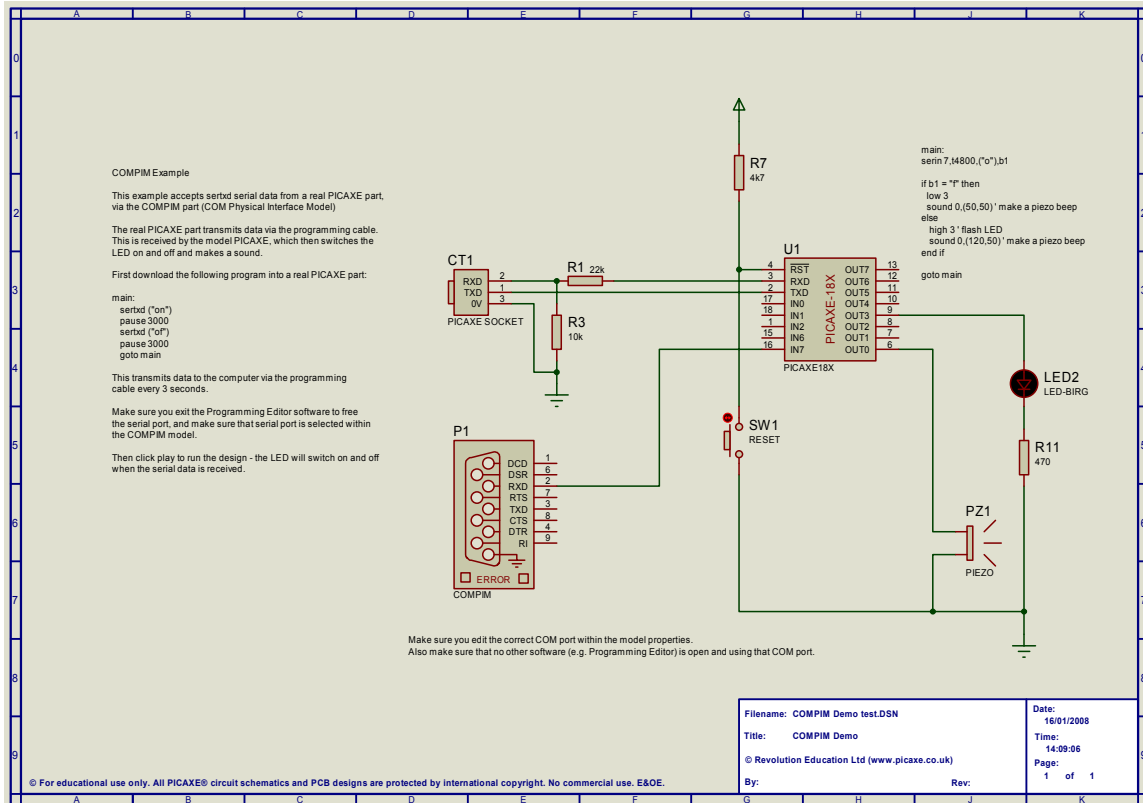


Figure 8. Demonstration of the COMPIM feature.

Signal Generator

The Signal Generator is demonstrated via the ‘count command.dsn’ file. When this simulation is run the Signal Generator control panel is displayed on screen. First make sure the generator is outputting a square wave (saw-tooth, triangular and sine are also available) of around 5V, and then try adjusting the frequency. The change in frequency should result in a different ‘count command’ value displayed on the serial terminal.

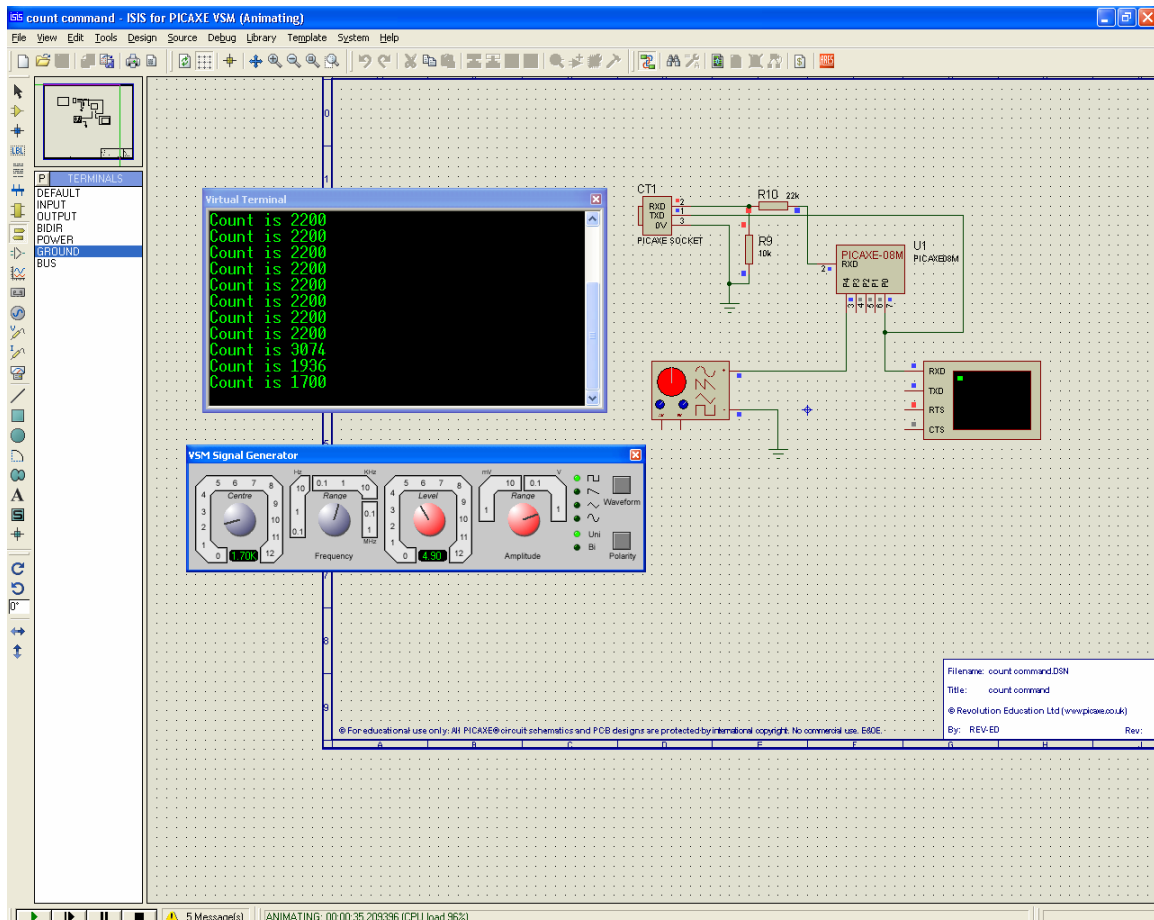


Figure 9. Virtual Signal Generator

An alternate, simpler, way of applying test signals to a wire is to just drop a generator probe onto the wire (as with the voltage and current probes). This is carried out by right-clicking and selecting Place>Generator.

SPI and I2C debuggers

SPI and I2C are two different communication protocols used to link ICs together. The protocols are quite complex, but fortunately the technicalities of the protocols are taken care of automatically in the PICAXE system, via use, as an example, of the writei2c and readi2c commands.

However those interested in studying these protocols further may be interested in the function of the i2c and spi debugger instruments. For instance, the writei2c command.dsn' file demonstrates how the 'i2c debugger' instrument is connected to the i2c bus.

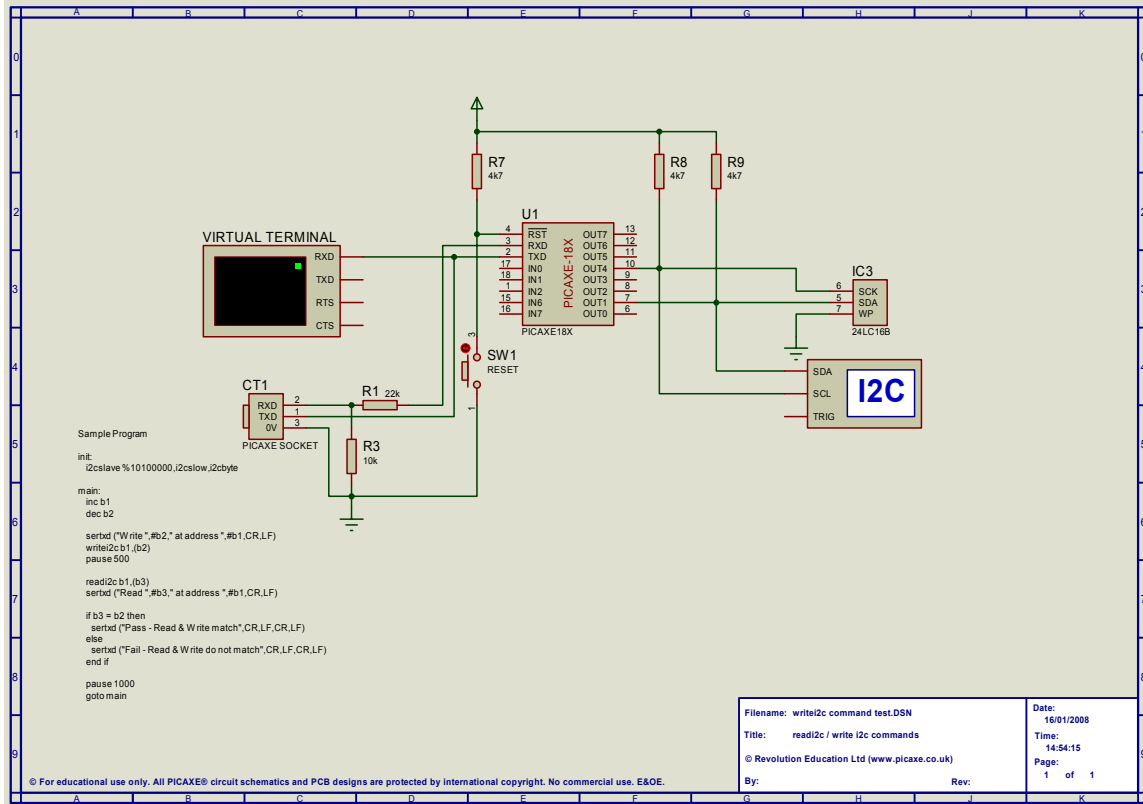


Figure 10. I2C debugger analysing data written to a 24LC16B EEPROM

When the simulation is run the details of each i2c transaction – start signal (S), address (AO), data pulses, ack pulses (A), nack pulses (N), stop signals (P) etc. are clearly listed in sequence on screen. This makes it an invaluable learning tool for those interested in understanding these protocols.

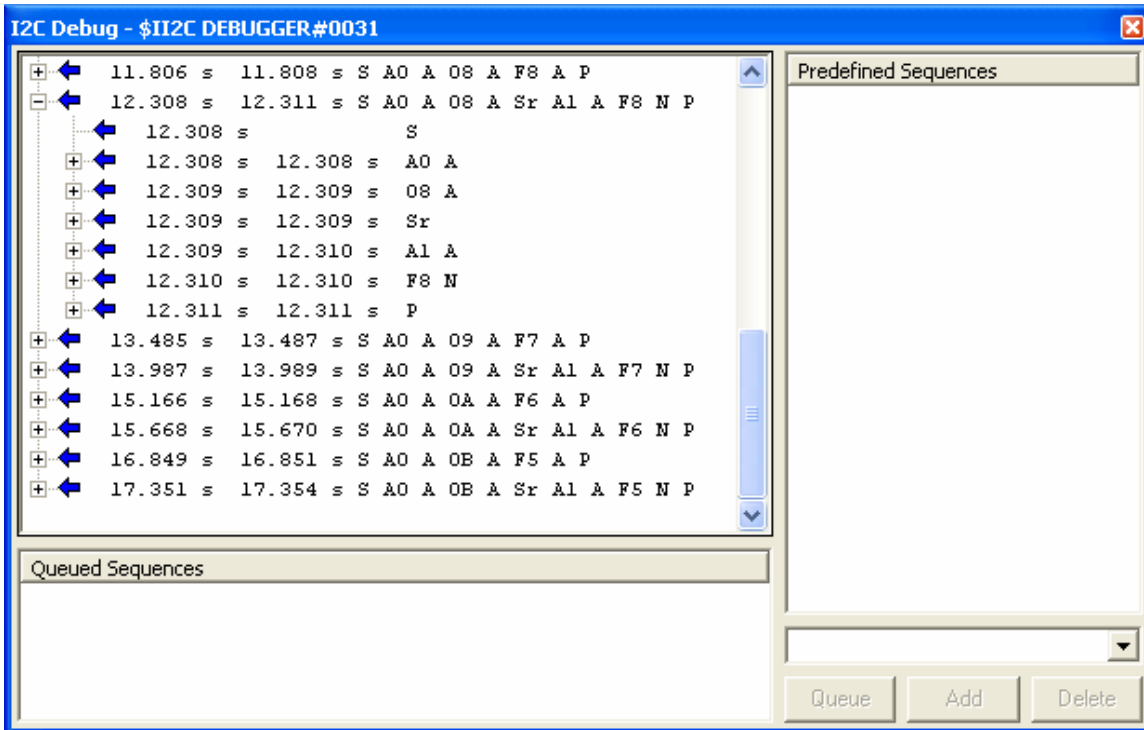


Figure 11. I2C debugger trace. This shows the time, and type, of all activity on the i2c bus

Summary

The software also contains a few other instruments such as a logic analyser – have a play with them! Using ‘virtual simulated’ instruments will never be quite the same experience as twiddling dials on real-life bench top models. However PICAXE VSM incorporates an impressive array of virtual instruments – and you certainly wouldn’t be able to buy a new oscilloscope, signal generator or i2c debugger for the same price as a VSM licence!