



## SCHEMATIC & PCB DESIGN SAMPLE FILES

The following is a list of the files related to Schematic Capture and PCB Layout contained in the SAMPLES directory along with brief descriptions.

CPU.DSN, CPU.LYT

CPUU.LYT

CPUR.LYT

CSMT.LYT

DBELL.DSN, DBELL.LYT

DBELLU.LYT

DBELLR.LYT

EPE.DSN

FEATURES.DSN

PPSU.DSN, PPSU.LYT

PSMT.LYT

SHIFTPCB Sample Files

SIGGEN.DSN, SIGGEN.LYT

SIGGENR.LYT

STYLE1.DSN, STYLE2.DSN, STYLE3.DSN

Schematic and PCB for Z80 CPU.

Unplaced version of above - try out the auto-placer.

Routed version of above - try out hard copy.

Same design with SMT components – shows power plane routing capability.

Schematic and PCB for Digital Doorbell.

Unplaced version of above - try out the autoplacer.

Routed version of above - try out hard copy.

Large hierarchical schematic.

Shows off some features of ISIS.

Very simple schematic/PCB - practice manual routing.

As above, using surface mount IC.

Simulation and PCB layout of large, hierarchical digital circuit.

Shows off gate-swap optimizer.

The circuit and PCB used in the self running demo.

Routed version of above.

These three samples show how you can create different drawing appearances without changing the schematic or library parts.

## SHIFTPCB Sample Files

This design consists of the schematic and PCB for a 16 bit shift register made entirely from TTL gate primitives. At the bottom level, an RS flip-flop is made from 2 NAND gates, then we make a JK flip-flop from 2 RS flip-flops and some other gates, then we make a 4 bit shift-register out of 4 JK flip-flops and at the top level we have four 4 bit registers coupled to make the 16 bit register.

The schematic is set up for both simulation and PCB design; the graph showing the propagation of a single pulse through the register.

The PCB presents a nice demonstration of the gate-swap optimizer. As given, the components have been placed but no optimization has been performed. If you attempt to auto-route this, it will fail. More quickly, you can just examine the total ratsnest length using the *System Information* command on the *System* menu.

On the other hand, if you run the Gate-swap Optimizer, you will find that the ratsnest length is dramatically reduced, and further that the board will then route to completion. It should take about 5 minutes to do both stages on a 300MHz Pentium II.

Click below to view the schematic and/or PCB.

[SHIFTPCB.DSN](#)

[SHIFTPCB.LYT](#)

## INTERACTIVE SIMULATION SAMPLES

These samples are provided both to demonstrate the capabilities of our Interactive Simulation technology, and also - free of charge - as an educational resource.

[Basic Electricity](#)  
[Measuring Voltage, Current and Resistance](#)  
[Potentiometers and Current/Voltage Characteristics](#)  
[Internal Resistance](#)  
[Capacitors](#)  
[Alternating Current](#)  
[Diodes and Rectification](#)  
[Inductors and Relays](#)  
[Transistor Circuits](#)  
[Operational Amplifier Circuits](#)  
[Oscillators](#)  
[Combinational Logic Circuits](#)  
[Sequential Logic Circuits](#)  
[Circuits for Fun](#)  
[Virtual Instruments](#)

## VIRTUAL INSTRUMENT SAMPLES

If you are interested in the capabilities of the Virtual Oscilloscope and/or the Virtual Logic Analyser, check out the following sample files:

[555 Oscillator](#)

Audio Frequency 555 oscillator circuit.

[Lissajous Figures](#)

Demonstration Lissajous Figures using the Virtual Oscilloscope. Extremely Pretty!

[8051 External Memory](#)

A demonstration of how Proteus VSM really does simulate the external hardware in a microprocessor design. Also compares and contrasts Interactive and Graph Based simulation.

*You must have installed the Virtual Instrument models to run these samples.*

## BASIC ELECTRICITY

### A collection of simple electrical circuits

<u>Circuit 1</u>	Battery and bulb with a switch.
<u>Circuit 2</u>	Battery and bulb with a variable resistor.
<u>Circuit 3</u>	Battery and two bulbs in series with a switch.
<u>Circuit 4</u>	Battery and two bulbs in parallel with two switches.
<u>Circuit 5</u>	Two way switching circuit.
<u>Circuit 6</u>	Electric Motor with speed and direction controls.
<u>Circuit 7</u>	Operation of a fuse.

## MEASURING VOLTAGE, CURRENT AND RESISTANCE

A set of circuits that demonstrate the measurement of voltage, current and resistance

<u>Circuit 1</u>	Battery and bulb with two voltmeters.
<u>Circuit 2</u>	Battery and two bulbs in series, with voltmeters across each component.
<u>Circuit 3</u>	Battery and bulb in series with an ammeter.
<u>Circuit 4</u>	Battery, bulb, voltmeter and an ammeter.
<u>Circuit 5</u>	Battery and two bulbs in parallel with series ammeters.
<u>Circuit 6</u>	Three batteries in series showing voltage summation.
<u>Circuit 7</u>	Battery, voltmeter, ammeter and a resistor.
<u>Circuit 8</u>	Demonstration of fixed potential dividers.
<u>Circuit 9</u>	Demonstration of potentiometer action.

## POTENTIOMETERS AND CURRENT/VOLTAGE CHARACTERISTICS

A set of circuits that demonstrate potentiometers and current/voltage characteristics

- |                  |  |
|------------------|--|
| <u>Circuit 1</u> | Battery and bulb with a variable resistor.                 |
| <u>Circuit 2</u> | Battery and resistor with a variable resistor.             |
| <u>Circuit 3</u> | Battery and reverse biased diode.                          |
| <u>Circuit 4</u> | Battery and forward biased diode with a variable resistor. |

## INTERNAL RESISTANCE

A couple of circuits that demonstrate internal resistance

Circuit 1

Battery connected to a variable resistor demonstrating the effect of heavy current drain on the battery output voltage.

Circuit 2

Battery short circuited.

## CAPACITORS

A set of circuits that demonstrate the nature of capacitors

Circuit 1

Battery and bulb in series with a capacitor.

Circuit 2

Battery, bulb, two switches, a resistor and a capacitor enabling the capacitor to be charged and then discharge through the bulb.

Circuit 3

As above, with voltmeters and ammeters added to the circuit.

## ALTERNATING CURRENT

### A list of circuits that demonstrate alternating currents

- |                  |   |
|------------------|---|
| <u>Circuit 1</u> | Alternator and light bulb.                    |
| <u>Circuit 2</u> | Alternator and two resistors with centre tap. |
| <u>Circuit 3</u> | Demonstration of transformer action.          |

## DIODES AND RECTIFICATION

### A set of circuits that demonstrate diodes and rectification

<u>Circuit 1</u>	Battery and resistor with a forward biased diode.
<u>Circuit 2</u>	Battery and resistor with a reverse biased diode.
<u>Circuit 3</u>	Alternator and resistor with a diode.
<u>Circuit 4</u>	Alternator, resistor and diode with an ammeter to show current variation.
<u>Circuit 5</u>	Half wave rectifier.
<u>Circuit 6</u>	Half wave rectifier with smoothing capacitor.
<u>Circuit 7</u>	Full wave rectifier using diode bridge.
<u>Circuit 8</u>	Full wave rectifier with smoothing capacitor.
<u>Circuit 9</u>	Full wave rectifier using two diodes and centre tapped source.
<u>Circuit 10</u>	Full wave rectifier as above with smoothing capacitor.

## INDUCTORS AND RELAYS

A set of circuits that demonstrate inductors and relays

- |                  |   |
|------------------|---|
| <u>Circuit 1</u> | Battery and bulb with an inductor in series.  |
| <u>Circuit 2</u> | Simple demonstration of a relay.  |
| <u>Circuit 3</u> | Relay circuit with variable resistor to demonstrate effect of latch and drop out voltages (hysteresis). |

## TRANSISTOR CIRCUITS

### A set of circuits that demonstrate simple transistor operation

- |                  |  |
|------------------|--|
| <u>Circuit 1</u> | Demonstration of biased and unbiased transistors.                              |
| <u>Circuit 2</u> | A transistor biased with a potential divider.                                  |
| <u>Circuit 3</u> | A transistor biased with a potentiometer.                                      |
| <u>Circuit 4</u> | A circuit that uses a transistor and a relay to control a light bulb.          |
| <u>Circuit 5</u> | A transistor circuit using a resistor and capacitor to create a turn on delay. |
| <u>Circuit 6</u> | A transistor circuit using a resistor and capacitor to give a turn off delay.  |
| <u>Circuit 7</u> | A circuit similar to (5) but with variable delay timing.                       |

## OPERATIONAL AMPLIFIER CIRCUITS

A set of circuits that demonstrate the use of an ideal operational amplifier

<u>Circuit 1</u>	Demonstration of open loop operation as comparator.
<u>Circuit 2</u>	Demonstration of open loop operation as amplifier.
<u>Circuit 3</u>	Non-inverting amplifier with 100% negative feedback.
<u>Circuit 4</u>	Non-inverting amplifier with partial negative feedback to give a gain of 3.
<u>Circuit 5</u>	Inverting amplifier with unity gain.
<u>Circuit 6</u>	Inverting amplifier with a gain of 3.

## OSCILLATORS

A set of circuits that demonstrate various oscillators

<u>Circuit 1</u>	Natural resonance oscillator.
<u>Circuit 2</u>	Transistor based astable multivibrator.
<u>Circuit 3</u>	Low Frequency 555 oscillator circuit.
<u>Circuit 4</u>	Audio Frequency 555 oscillator circuit.

## COMBINATIONAL LOGIC CIRCUITS

A set of circuits that demonstrate the operation of the basic logic gates

<u>Circuit 1</u>	AND gate.
<u>Circuit 2</u>	OR gate.
<u>Circuit 3</u>	NOT gate.
<u>Circuit 4</u>	NAND gate.
<u>Circuit 5</u>	NOR gate.
<u>Circuit 6</u>	XOR gate.

## SEQUENTIAL LOGIC CIRCUITS

A set of simple sequential logic circuits

<u>Circuit 1</u>	RS flip-flop.
<u>Circuit 2</u>	D type flip-flop
<u>Circuit 3</u>	J-K flip-flop.
<u>Circuit 4</u>	Shift register.
<u>Circuit 5</u>	3 bit binary counter.

## CIRCUITS FOR FUN

Here are few more designs that we have put together for your amusement!

[TRAFFIC.DSN](#)

A simple circuit that controls a set of Traffic Lights. Compare this with the [PIC Traffic Lights sample](#).

[COUNTER.DSN](#)

A four stage decade counter with 7 segment displays. Will operate in real time at seriously impressive speeds.

[TTLCLOCK.DSN](#)

A fully working digital clock based on standard TTL ICs.

[LISSAJOUS.DSN](#)

Demonstration Lissajous Figures using the Virtual Oscilloscope. Extremely Pretty!

## PIC Based Digital Clock

This sample implements a simple digital clock using a PIC 16F84 microcontroller. The time is displayed on a number of 7 segment displays and the time can be set using various mouse operated switches and buttons.

As with all the samples, pressing the PAUSE button stops the program execution and the commands/keys on the debug menu can be used to single step the code, display registers and so on.

[PICCLOCK.DSN](#)

## PIC Based Traffic Lights

This sample implements a trivial controller for a set of traffic lights using a PIC 16F84 microcontroller. It is a good starting point for studying how to set up a VSM simulation, and is also used as the basis for the VSM tutorial in the user manual.

As with all the samples, pressing the PAUSE button stops the program execution and the commands/keys on the debug menu can be used to single step the code, display registers and so on.

[TRAFFIC.DSN](#)

## PIC and Graphics LCD Display

This sample is taken from an article in [Everyday Practical Electronics](#) by John Becker and consists of a tutorial in controlling a PG12864F LCD display using a PIC16F877 processor.

To operate the sample, press the PLAY button, and then click the pushbutton switch on the circuit to step through the various demonstrations contained with John Becker's code.

As with other VSM samples, you may modify the source code even if you have not registered your copy of Proteus Lite or are using the Proteus Professional demonstration. But the sample will not simulate on an unregistered copy if you modify the electrical connectivity.

[GEPE456.DSN](#)

*You must have installed both the PIC and the VSM Peripheral models to simulate this sample.*

## PIC and Serial LCD Display

This sample demonstrates the use of serial LCD displays - as available from [Milford Instruments](#) - with a PIC1687X processor.

To operate the sample, press the PLAY button, and type text into the virtual terminal - the text you type will be output to the LCD display.

As with other VSM samples, you may modify the source code even if you have not registered your copy of Proteus Lite or are using the Proteus Professional demonstration. But the sample will not simulate on an unregistered copy if you modify the electrical connectivity.

[RS232LCD.DSN](#)

*You must have installed both the PIC and the VSM Peripheral models to simulate this sample.*

## PIC Musical Doorbell

Demonstrates a combination of the PIC 16F84 model with the digital sounder model to implement a novelty/musical doorbell. On a Pentium II or better PC, the simulated PIC will run fast enough to generate audio tones in real time!

With apologies to J.S Bach and W.A. Mozart....

[DOORBELL.DSN](#)

*You must have installed both the PIC and the VSM Peripheral models to simulate this sample.*

## PIC ADC Example

This sample demonstrates the use of the ADC module in the PIC16F877, and also proves the point that Proteus VSM can simulate analogue as well as digital electronics in conjunction with a CPU.

[PICADC.DSN](#)

## 8051 Calculator (Assembler Version)

The 8051 calculator uses an 8051, an LCD display and the universal keypad to implement a four function calculator. We borrowed the floating point code from Intel's 8052 basic (see the [8052 Basic Interpreter](#) sample).

The design uses external memory, and in this case, VSM is simulating the 6264 RAM chip, passing events to and from the 8051 just as in real life. When the simulation is paused, the contents of the RAM may be displayed from the Debug menu.

*Educationalists take note:* the code for this sample was produced by someone inexperienced in 8051 development and unfamiliar with 44780 based LCD displays. He is now considerably competent with both these devices.

[CALC.DSN](#)

## 8051 Calculator (Keil C51 Version)

This sample serves to demonstrate the link between Proteus VSM and Keil  $\mu$ Vision2. It consists of a re-working of the [8051 calculator](#) written in 'C' rather than assembler. Since the Keil Evaluation version allows only 2k of machine code, the functionality of the calculator is somewhat limited. In particular, only integer maths are provided. The full version of C51 does, of course, include floating point maths. However, this sample file serves well enough to illustrate the debugging capabilities offered by the combination of the two packages.

To experiment with this sample you will first need to install Keil  $\mu$ Vision 2, and then follow [these instructions](#). An evaluation copy of  $\mu$ Vision2 is provided on the Proteus Demo CD, or can be obtained from [www.keil.com](http://www.keil.com).

[CALC . UV2](#) - The  $\mu$ Vision 2 project.

[CALC . DSN](#) - The C51 Calculator Schematic.

[VDM51 . PDF](#) - Application note covering how to set up the VDM51 debug monitor driver for Keil UV2.

## 8051 External Memory

This sample demonstrates how Proteus VSM really can simulate the external hardware. An 8051 is wired up to an external RAM and the program repeatedly reads and writes data to the RAM chip.

The design also compares and contrasts interactive and graph based simulation, using both the Logic Analyser and a Digital Graph to display the memory cycles.

*Please note that graph based simulation is only available in Proteus VSM Professional, and is not supported in Proteus Lite.*

[EXTMEM.DSN](#)

*You must have installed both the MCS8051 model and the Virtual Instruments to simulate this sample.*

## 8051 LCD Driver

This design shows how to drive a standard LCD alphanumeric module using an 8051.

[LCDDemo.DSN](#)

*You must have installed both the MCS8051 model and the LCD Display models to simulate this sample.*

## 8052 Basic Interpreter

This sample comprises a complete design for a simple microcomputer.

At its heart is an 8052 running a BASIC interpreter originally written by Intel and now available as freeware on the Internet. The 8052 is wired to an external RAM and also to an EPROM. Additional circuitry is provided which allows the 8052 to program the EPROM in situ.

The 8052 BASIC is programmed through the serial port which is wired to a VSM Virtual Terminal. Press the PLAY button and (after a short pause whilst BASIC tests the external RAM) you should get:

```
*MCS-51(tm) BASIC V1.1*  
READY  
>
```

Brief Instructions on how to program in 8052 BASIC are to be found in the file BASIC-52.PDF (we assume that you have the Acrobat Viewer installed).

A quick example is as follows:

```
10 print "Hello World"  
20 goto 10
```

Type RUN to run it, and press CTRL-C to stop it.

If you then type PROG, you will store the program in the EPROM. Diode D2 will light up whilst programming is in progress, and the voltmeter will display the VPP being fed into the EPROM. If you then enter RROM1, you will run the program from the EPROM. And if you stop the simulation and re-start it, entering RROM1 will still run the program that you stored in the EPROM. Just like in real life.

We've turned on some of the VSM debugging aids on this one, including the pin logic state dots. If you locate the correct portion of the BASIC source code and set a breakpoint, it is possible to observe the programming of a byte of EPROM literally step by step.

[BASIC-52.DSN](#)  
[BASIC-52.PDF](#)

## HC11 Basic Interpreter

This sample comprises a complete design for a simple microcomputer running a slightly modified version of Gordon Doughman's BASIC11. This was original written for the HC11 evaluation board (EVB) and is widely available on the Internet.

BASIC11 is programmed through the serial port which is wired to a VSM Virtual Terminal. Press the PLAY button and (after a short pause whilst BASIC tests the external RAM) you should get:

```
BASIC11 v1.55
Copyright 1985-1990 by
Gordon Doughman

READY

>
```

Brief Instructions on how to program in BASIC11 are to be found in the file BASIC11.TXT.

A quick example is as follows:

```
10 print "Hello World"
20 goto 10
```

Type RUN to run it, and press CTRL-C to stop it.

The sample circuit also includes a potentiometer that is wired up to one of the HC11's analogue inputs. If you write

```
10 print adc(1)
20 goto 10
```

you will get a continuous printout of the values read by the on chip ADC module. This is a simple demonstration of how VSM can simulate the interaction of the analogue electronics with a microprocessor.

[BASIC11.DSN](#)  
[BASIC11.TXT](#)

## HC11 and Graphics LCD Display

This sample shows how to interface the HC11 with a Graphical LCD display. A range of other display models including various sizes of alphanumeric and graphical modules may be found in the DISPLAYS library.

[HC11DEMO.DSN](#)

*You must have installed both the HC11 and the VSM Peripheral models to simulate this sample.*

## Motorola EVB11 Board

This sample contains the complete circuitry for the Motorola EVB11 Development Board. All the major hardware elements are simulated, including the memory decode, 6824 PRU, 6850 ACIA, ROM, RAM and flip-flop.

The supplied code operates a simple terminal echo program, but you can also load your own S19 files and try out the same experiments as you would on a real EVB11, subject to the usual restrictions if you are using the Proteus Demo or Proteus Lite.

[EVB11.DSN](#)

*You must have installed both the HC11 and the VSM Peripheral models to simulate this sample.*

## DC Motor Samples

These samples demonstrate the various VSM motor models.

[HC11PWM.DSN](#)

PWM control of a DC motor using the HC11 timer subsystem. Requires installation of the HC11 model and VSM Peripheral models.

[PICSTEPR.DSN](#)

Stepper motor controller based around PIC1684 and ULN2003 darlington driver. Requires installation of the PIC1684 model and the VSM Peripheral models.

[SERVO.DSN](#)

Classic analogue servo motor controller. Requires installation of the VSM Peripheral models.

## COMPIM Sample

This design provides a very simple demonstration of the COMPIM model. The COMPIM is wired to a VSM Virtual Terminal, enabling characters generated by the terminal to be routed to the physical COM port of your PC and vice-versa.

[TERMECHO.DSN](#)

This sample is provided as a starting point for your own uses of the COMPIM model rather than as example of a practical application. It is hard to come up with anything sensible for this model, since most practical applications will involve specific pieces of physical hardware which we cannot assume that you will have!

## GRAPH BASED SIMULATION SAMPLES

As well as the interactive simulation mode, Proteus VSM Professional provides an extensive set of graph based analysis types based around the functionality of SPICE3F5. These files demonstrate this functionality and also provide a basis from which to set up your own simulations.

*Please note that these samples are not included with Proteus Lite as it does not support graph based simulation.*

### Analogue Simulation Examples

[741.DSN](#)

[CRYSTAL.DSN](#)

[FF.DSN](#)

[FOURIER.DSN](#)

[MUTUAL1.DSN](#)

[MUTUAL2.DSN](#)

[RA53.DSN](#)

[RESCAP.DSN](#)

[SPICE1.DSN](#)

[SPICE2.DSN](#)

[SWEEP.DSN](#)

[TRANSFER.DSN](#)

[TRFORMER.DSN](#)

[TRIANGLE.DSN](#)

[ZIN.DSN](#)

[ZOUT.DSN](#)

741 model using actual 741 circuit.

Shows modeling of a quartz crystal.

Simulating an astable multivibrator.

Example of Fourier Analysis.

Simple example of modeling mutual inductance.

Advanced example of modeling mutual inductance.

Simulation of Wien bridge sine wave oscillator

including self heating thermistor.

Assorted simulations of an RC network.

Example of how to use a 3rd party SPICE model for an IC.

Example of how to use a 3rd party SPICE model for a transistor.

Example of AC and DC sweep analyses.

Example of transfer curve analysis.

Shows how to use the transformer models in DEVICE.LIB.

Shows how to set up a pulse generator to make a triangle wave.

Shows how to measure input impedance.

Shows how to measure output impedance.

### Digital Simulation Examples

[16L8.DSN](#)

[393\\_154.DSN](#)

Example of PAL support in PROSPICE.

Shows how PROSPICE handles glitches.

### Mixed Mode Simulation Examples

[CMOSMONO.DSN](#)

[INVOSC.DSN](#)

[PLLTYPE1.DSN](#)

[PLLTYPE2.DSN](#)

[VCO.DSN](#)

[ADC0808.DSN](#)

[DAC0808.DSN](#)

Mixed mode simulation of CMOS monostable.

Mixed mode simulation of two inverter oscillator.

Mixed mode simulation of Type 1 Phase Locked Loop.

Mixed mode simulation of Type 2 Phase Locked Loop.

Mixed mode simulation model for a Voltage Controlled Oscillator.

Shows how to model an Analogue to Digital converter.

Shows how to model a Digital to Analogue converter.