

# Experiment 1

## Introduction to Electronics Workbench

### 1.1 Preliminaries

You may be quite familiar with using a general-purpose computer, such as a Windows PC or a Mac, but the computer environment you will find yourself in in the Electronics lab is more specialized, and may present additional challenges. Be sure to attend the introductory lecture offered by the Brock Information Technology Services (ITS) early on in the semester. The computing environment around Brock evolves quickly; your lab instructor should have the most up-to-date information.

Currently, the ITS operates several computer labs within the Department of Physics (B203 and H300 are the two largest ones), all of them deploying identical Linux-based workstations. Logging on to any of them with your Brock assigned Campus username/password will bring you into the exact same environment, and all the files in your home directory will be available to you (they are stored on one central network server, accessible from all workstations equally).

In addition, this course makes a heavy use of a circuit simulation program called **Electronics Workbench** (EWB, also known as the National Instruments Multisim due to a recent corporate change). Unfortunately, there is no Linux-compatible version of this software, and it has to be run from a separate central Windows server. You access this server using the same Campus username/password.

- ❗ Log on to the Linux workstation using the Campus username/password.
- ❗ Click on the OpenSUSE icon at the bottom left of your screen, then click on *Science* followed by *Electronics Workbench* to open login window to the Windows server. Enter the same Campus/username/password and a Windows Desktop will appear.
- ❗ Click on the EWB5 icon to start the program.

Move around and examine the menus and controls. Pausing a cursor over an unknown item should bring up a bubble of a description of what the item is. The simulation is controlled by the *on/off* switch at the top right corner of the screen. If you are lost, quit and restart the program.

### 1.2 Virtual circuits in Electronics Workbench

*Electronics Workbench* is essentially “an electronics lab in a computer”, and the way it appears on your computer screen is shown in Fig. 1.1. The white field in the middle is the workspace into which you drag various components and devices found in the multiple parts bins divided into several categories, just above the workspace. When you then bring the mouse near the edges of

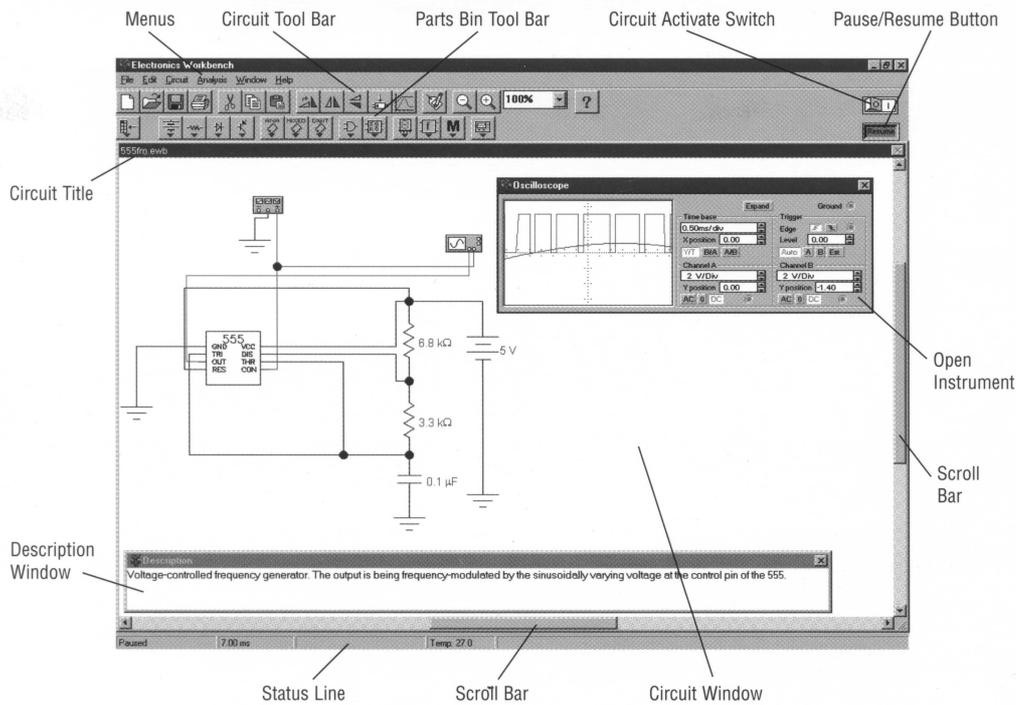


Figure 1.1: Electronics Workbench screen

each component, they turn into dark dots representing nodes of your future circuit. Click and drag until a line stretching out of a node reaches a node of another component, then release. You just connected a virtual “wire” between the components. The wires snap to a grid (which can be made explicitly visible through the **Circuit** menu), and as you move components around the wires stretch and follow as needed.

After a few mouse-clicks, you can assemble an entire virtual circuit that includes passive and active components, meters, oscilloscopes, and other virtual counterparts to the real devices and instruments found in an electronics lab.

There is one important difference to working with a virtual circuit. As you are putting it together, the program creates a set of mathematical equations that describe the circuit. As you then flick the virtual ON switch, the computer proceeds to solve these equations, quickly and with great precision, and reports and even plots the results. A variety of values can be swept through quickly and automatically, to discover the optimum ones; an entire frequency response curve can be obtained with a single click of a mouse.

What happens is that you are able to concentrate on the *physics* of the problem, and not on the sometimes tedious details of setting up and solving a fairly large system of coupled linear and differential equations. You do not need to be careful with the details of these calculations, and you concentrate instead on making sure you understood the behaviour of the circuit and how this behaviour relates to the underlying theory.

When you concentrate on the concepts and avoid applying by rote a memorized set of steps you are studying for mastery. When you understand what is going on behind the equations, you can apply that understanding to problems where the rote method is sure to fail. In our computer-assisted labs you will learn to test your understanding, to make up circuits and to predict the results mentally, then have the computer verify (or not!) your predictions. You will build up your intuition on the subject of Electronics. In some sense, your efforts will closely parallel what physicists do

every day in their research, something often called “the scientific method”: organize your knowledge, develop a theory, make predictions, test them by experiment.

The results of this exploration need to be described in a complete, coherent fashion to allow a reader (or marker) familiar with the subject the opportunity to understand your methodology and attempt to reproduce your data. **Do not** assume that the reader has intimate knowledge of your experimental setup and goals.

For each of the following exercises, explain fully the procedures undertaken, your observations, followed by a short summary of your results. Include screen captures of all circuits, simulation output, data tables and meaningfully scaled graphs. Do not refer to the lab manual; include *all* pertinent details, in your own words, as part of the report. Support *all* statements and conclusions with experimental evidence or reasoned arguments.

**Hint:** You can screen capture Electronics Workbench output by going to the OpenSUSE start menu then clicking on *Accessories* followed by *Screenshot*. In the window that opens, select to capture a screen region, outline the area of interest by pressing the left mouse and moving the cursor to outline a rectangular region, then release the button and save the output to a file. You can conveniently include the schematic, instrument settings and output, as well as descriptive text (A, textbox from the Miscellaneous submenu) as one capture that can be saved and imported to a document. **Do not** append printouts to the end of the lab report.

## 1.3 Real and ideal meters

- ❗ Consult your notes or the Web to determine the internal configuration of a real (non-ideal) voltmeter and ammeter.
- ❗ Pull down a battery (in the sources bin, move the mouse over the icons and a bubble with the name appears) and a multi-meter (in the instruments bin) into the worksheet. Double-click the multi-meter icon for a close-up view. Verify the multi-meter is in voltage mode, *i.e.* that V is highlighted. Practice connecting/disconnecting the wires and moving the components around the worksheet. Move the components to verify that the wires are properly connected to them and that the electrical circuit is complete. Connections display a black dot.
- ❓ When do you see a positive reading on the meter? a negative one?
- ❗ While the meter is connected to the battery, switch it into the current mode by pressing A.
- ❓ What happened? Why do you **never** do this to a real meter? Refer to the internal circuit for a real ammeter to describe what happened. Click the multimeter *Settings* button for further insight. What is the most common way used in real multi-meters to protect against errors like this?
- ❗ Switch the meter back to voltage mode. Pull down and insert a 1 k $\Omega$  resistor in series with the battery. (Position the resistor directly over an existing wire, and release; the resistor will insert itself.) Vary the resistance; is the voltage on the multi-meter changing? (*Hint:* you may have to go to pretty high  $R$  values!) Try to find the point where the meter reads exactly  $\frac{1}{2}$  of the nominal battery voltage.
- ❓ In EWB the batteries are always ideal sources; however, the meters are not. The point of  $\frac{1}{2}$ -voltage is where the internal resistance of the (non-ideal) meter is exactly equal to the external  $R$ . Explain your result and provide a calculation to support your observation.

## 1.4 Review of basic electronic components

While logic gates are all that is required to construct a digital circuit, you will find that to interact with your digital system some familiarity with the behaviour of basic electronic components such as resistors, diodes and transistors, is necessary.

- The resistor is a linear circuit element in that it obeys Ohm's Law:  $V=I \cdot R$ , where a voltage  $V$  across a resistance  $R$  causes a current  $I$  to flow through the resistor. The resistor is typically used in series with another component as part of a voltage divider or to limit the circuit current.
- The ideal diode can be visualized as a switch that is turned off ( $R \approx \infty$ ) until a positive voltage (forward bias)  $V \geq V_{on}$  is applied across the terminals. The negative diode terminal (cathode) is represented by the flat line. When turned on, a voltage  $V_{on}$  will be present across the diode and  $R \approx 0$ . The value of  $V_{on}$  is  $\approx 0.6V$  for a Silicon diode,  $\approx 0.3V$  for a Germanium diode and  $\approx 1.5V$  for a GaAs red light-emitting diode (LED). Continuous diode currents much greater than 10-20 mA will cause LEDs to burn out.
- The bipolar transistor is a three terminal device. The terminal with the arrow is the emitter, the opposite terminal is the collector and the centre terminal is the base. This device is a current amplifier in that the emitter-collector current  $i_{CE} = \beta * i_{BE}$ , the base-emitter current times the transistor current gain  $\beta$ . The base-emitter junction behaves like a diode so that  $i_{BE} \approx 0$  and  $R_{CE} \approx \infty$  until  $V_{BE} \geq V_{on}$ . As  $V_{BE}$  increases,  $R_{CE} \rightarrow 0$ .

In the digital domain, a transistor is typically used to switch on/off devices that require more current than a logic gate is capable of supplying, such as LEDs, lamps and relays.

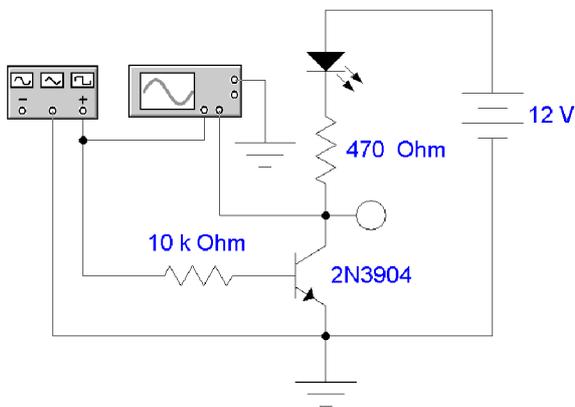


Figure 1.2: Transistor as a switch

You can incorporate these components in the circuit as shown. Use a scope (from the Instruments menu) to monitor the base and collector voltages. Drive it with a 10 V peak-to-peak square wave centered at zero volts with a frequency of less than 1 Hz. The round indicator probe turns on (solid color) at 2.5V. The LED is on when the arrows are filled in.

- ⓘ Start the simulation and observe the behaviour of the circuit. Note when the LED is on and when it is off (circle below) in terms of the voltage at the base,  $V_b$ , and at the collector,  $V_c$ , of the transistor and the corresponding voltages across the components:

$V_b$	$V_c$	LED	probe
low		on / off	on / off
high		on / off	on / off

- ❓ Explain the operation of the circuit for the two input states. What is the purpose of the  $470\ \Omega$  and  $10\ \text{K}\Omega$  resistors? What logic operation does the transistor perform? What is the maximum current that could flow through the LED?

## 1.5 Digital indicators and controls

- ❗ Clear the screen of all components, or simply open a new circuit page. Find and drag down into your circuit page the word generator (from the Instruments submenu) and several one-bit logic probes (from the Indicators submenu). You can enter sequences of 4-digit hexadecimal numbers in the data window or load a stored pattern. The sequencer can be single-stepped or made to cycle through this list at a certain predetermined pace. The sequencer can also be put into an infinite loop, repeating the steps you programmed and returning back to the beginning.

For each number, the corresponding pattern of high/low voltage levels is produced on its 16 outputs. Edit the pattern list to contain a few lines with different bit patterns; try to include FF, 00, and other interesting bit patterns in your sequence. Step through the sequence and observe how the pattern of indicators connected to the sequencer output lines follows.

- ❗ Logic probes are ideal sensors of the logical state of any wire in the circuit; a more realistic way is to use light-emitting diodes (LEDs). Drag several LEDs or an eight-LED bar graph indicator into your circuit, and connect the anodes of each LED diode to the outputs of the sequencer. Connect the cathode ends to ground through  $220\text{-}\Omega$  resistors. These serve as current-limiting resistors, without them the LEDs would “burn out” almost instantly. Enter a two-line sequence AA, 55 which will create an illusion of lights running up your LED bar graph.

Modify your sequence to create a single running light: 00000001, 00000010, 00000100 *etc.*

- ❗ Replace the LED bar graph with a 7-segment LED display. Assemble the circuit shown.

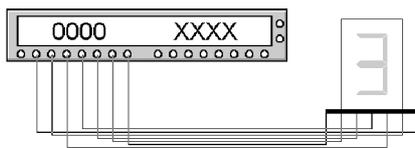


Figure 1.3: Controlling a 7-segment LED

The 7-segment LED display has seven inputs that each control the lighting up of a particular segment of the display. Thus for each pattern of bits produced by the pattern generator, there appears a pattern of lit LED segments. The labeling convention and pin assignment for each segment can be found in the help menu for the device.

- ❗ Map each segment to an output of the word generator, then generate a table of bit patterns that correspond to the 7-segment representations of the decimal digits 0-9. Program the list of patterns into the generator’s table to produce a sequence of seven-segment digits from 0 to 9. Set the pattern generator into a loop mode over the list you entered, and verify that the display appears to “count” from 0 to 9.
- ❗ For an extra challenge, expand the list to 16 entries, one for each of the hexadecimal digits.

## 1.6 Bit combinations control a 7-segment LED display

The previous section demonstrated a “brute-force” approach; usually the job of controlling the individual segments of a 7-segment display is given to an integrated circuit called BCD-to-7-segment encoder.

- ❗ From the DEC menu in Flip/Flop bin, place a “Generic BCD-to-seven-segment decoder” into your circuit, between the sequencer and the 7-segment LED display. The conventional order is that on the BCD side the bit A is the least significant bit, and that the seven-segment display inputs are A-to-G left-to-right. Make sure that the LT (lamp test), BI (blanking input) and RBI (ripple blanking input) are tied high, *i.e.* attached to a  $V_{cc} \equiv 5V$ . Modify the sequencer program to count up from 0 to 9, and then back down to 0, a total of 19 steps.

## Lab Report

Submit a lab report consisting of the work undertaken during this lab. waveforms observed, formula derivations, a description of the theoretical behaviour of the circuit and comparison with your actual observations, and answers to the pertinent questions. The presentation of your results should be organized and complete, your diagrams titled and referenced, so that someone who is not familiar with the experiments would have no difficulty understanding what was done.

At the end of the lab report, include a brief Conclusions section that summarizes and compares the results from the simulated and hands-on portions of the lab and a discussion of any problems encountered and insights gained.

**Note:** The lab report for this experiment is due before the beginning of the next scheduled lab session, one week later. Late labs receive a zero grade. To satisfy the lab component for the course, all lab reports must receive at least a passing grade. Be sure to have read and are familiar with the preface section of this lab manual.

**Note:** You need to prepare for the next experiment by designing several logic operations using NAND gates. Be sure to have these circuits ready to use.