

“What?? An actual lab manual? You mean I have to pay extra for this? I thought computer labs were places where you went and did homework. What is this laboratory business?”

Computer science started out as a branch of mathematics. Indeed, at many major universities the Computer Science program has very close ties to the Mathematics program. Mathematics is unusual in the sciences insofar as its laboratory exists mainly in the minds of its practitioners. A mathematician doesn't walk into a lab and mix two equations together as a chemist would chemicals; nor would he take apart an equation with a particle accelerator as a physicist would an atom. Most mathematical “experiments” are little more than intellectual exercises.

Computer Science, itself, began as an intellectual exercise in the minds of various mathematicians. Since they created it, it is not surprising that the exploratory model found in most Computer Science programs is that of the mathematician. Despite its strong ties to mathematics, however, Computer Science is an *engineering* discipline. Engineers must be concerned not only with the theoretical aspects of their field, but the practical applications as well. Engineers *design* things. They begin with a theory, design a model or prototype, analyze the model, optimize the design, and construct a marketable final product. Alas, only recently have universities begun switching to this new instructional paradigm for Computer Science. By the year 2000, most students will have a hard time believing you could teach Computer Science without a laboratory component, much like today's students would have a hard time believing you could teach Biology, Chemistry, or Physics without a laboratory.

“What exactly is a computer lab about, anyway?” you're probably thinking. Well, it's like a lab in any other engineering science. It begins with a goal. You develop a hypothesis and run some experiments to determine if the hypothesis is correct. If not, you repeat the process; if so, you refine your hypothesis. In the design of computer software, you may start out asking if something is even possible. Once you determine that it's possible, the next step is to see if it is practical. Then the optimization phase becomes cost effective. And so on. Each of these steps requires various amounts of experimentation (e.g., testing) and refinement. A mathematician is often happy with an existence proof. An engineer must go well beyond this.

As the title implies, this supplement is a little more than just a lab manual. It's also a study guide, workbook, and a source of examples for “The Art of Assembly Language Programming.” A typical chapter consists of five components: a tutorial on the software you're about to use, the laboratory assignment, lots of examples, a series of questions for you to answer as you're reading the text and some accompanying software. You'll soon see that these components are related. Indeed, the examples and questions generally appear within the tutorial section.

To reap maximum benefit from this manual, it's important that you use it properly. As with any laboratory experiment, it's important that you properly prepare prior to actually running the experiment. This means you need to read up on the experiment *before* showing up to the lab. If there is a limitation on how much time you can spend in the lab (or how much time you're *willing* to spend in the lab), advance preparation is absolutely mandatory. If you show up to the lab without having read the lab manual, you'll waste a lot of time in the lab just catching up on your reading. To encourage you to properly prepare for the laboratory periods, this manual provides several preparatory exercises which you should perform prior to attending the lab section. In general, you will be expected to turn in a “pre-lab report” prior to beginning experiments in the laboratory. This pre-lab report includes answers to the workbook questions, a series of hypothesis, and set of expected results, and a procedure for carrying out the experiment(s). You should bring two copies of the pre-lab with you to the lab section: one to turn in (as proof you've properly prepared) and one to work from. The next chapter gives an example of a pre-lab report to serve as a guideline.

The workbook component of this manual is embedded in the tutorial. As you read through the material, you will encounter several questions. *Answer these questions immediately and in this workbook.* Any distraction will lessen the educational benefit of the question. Distractions include waiting until later to answer the question or writing the answers on a separate sheet of paper. This is a *workbook*. You are supposed to write in it; that's one of its major design features. Preserving it in pristine form will not help you learn the material any better!

The answer to each question appears in the margin two to three pages later. Answer the question as best you can without looking ahead; even if you have to make a wild guess. Then, before going on in the text, flip the page and look at the answer. If you answered the question incorrectly, figure out why, erase (or cross out) your incorrect answer and replace it with the correct one. Only after correcting the answer should you proceed. *Do not simply look at the answer, say, “oh yeah!” and then proceed with the tutorial.* Several educational studies have shown that you learn *much* more from your mistakes if you correct them *immediately* after you make them. The design of this workbook takes this into

account. If you do not take a few moments to follow this procedure, you will be cheating yourself. Considering the amount of time it will take you to read this material, it would be a shame not to take a few extra moments to guarantee that you retain the information.

Reading the tutorial section of this workbook is not a substitute for reading the associated chapter in the textbook. The textbook presents material in an orderly fashion, ensuring that it does not use new concepts until after it presents them. The tutorial makes no such guarantees. Although the Chapter Six tutorial section will not use material from Chapter Seven, section 6.2 may very well use material from section 6.8. If you've read the chapter prior to reading the tutorial and working through the exercises you won't have a problem. The tutorial will tie together the concepts presented in the textbook. On the other hand, if you attempt to work through the tutorial exercises without first reading the textbook, you may find yourself wondering what the tutorial is talking about.

Once you are done with a laboratory exercise, next comes the dreaded *lab report*. For some reason, students feel that a lab report is little more than an anticlimax to the laboratory. As a result, their lab reports look like afterthoughts. Keep in mind, however, that the lab report represents the majority of your grade associated with the laboratory. If you are at all concerned about your grade in this course, why would you spend several hours in a lab and then only spend a few minutes on the lab report? A hastily written and skimpy lab report is indicative of a hastily prepared and executed laboratory session. Don't let your instructor misjudge the quality of your work because of a sloppy lab report.

While on the subject of lab reports, a brief comment on presentation is in order. A handwritten *or even typewritten* lab report is no longer acceptable, especially for Computer Science students. As a student in a Computer Science course, you should use the computer, with an appropriate word processor or desktop publishing program, to prepare your lab reports. Likewise, with the plethora of graphics programs for personal computers, there is very little reason to submit hand drawn diagrams. Even if other students submit such diagrams, imagine how your instructor's impressions upon seeing your professional looking figures.

Although presentation will most certainly affect the grade you get on a lab report, by no means is it the lion's share. As usual, content has a lot to do with your grade on a laboratory report. A *minimally acceptable* lab report is one which includes a reasonable pre-lab section with appropriate hypotheses, conjectures, procedures, and expectations; a description of the laboratory session with observations, procedures followed (and/or modified from the pre-lab), results expected vs. results obtained, and time spent on each activity; and a conclusion supporting or rejecting original hypotheses.

If you have the above, you should expect the equivalent of a 'C' on your lab report. If you would like to produce a lab report which is better than minimally acceptable, it's time to start using your imagination. Certainly presentation is a big factor. If your report stands out because it looks professional, this can't help but improve your grade. This doesn't mean simply using a word processor and a laser printer. That's part of the minimally acceptable report. Using proper layout, appropriate fonts and diagrams, quality binding, and so on, **will** improve your grade. The next step up is to improve on the laboratory itself. The experiments and procedures appearing in this manual are for a basic lab. They, by no means, represent all of the possible experiments you could perform on the subject at hand. Devising your own experiments, along with supporting reasons for performing them, definitely shows initiative and is a big source of additional credit on lab reports. Another way to gain some extra credit is to expand on a given experiment. For example, if one experiment asks you to compare and contrast the behavior of two different forms of an assembly language instruction, contrasting three, four, or all forms of that instruction is even better.

The whole purpose of the laboratory is to get you to think, probe, experiment, and understand what is going on with the material at hand. It isn't busy work to give you something to do three hours a week. It is important work which will complement and cement the knowledge you're learning in class. If you do not take the laboratory seriously, you will be shortchanging your education.

Many of the laboratory experiments in this lab manual require you to wire up some simple circuits and control this circuitry via the computer's parallel (printer) port. These experiments use components that are readily available at any Radio Shack electronics store. Limiting the components to those available at Radio Shack does limit the possible experiments, since Radio Shack does not carry an astounding array of electronic components, but it does guarantee that the components are available world-wide. Note that Radio Shack is *not* the only source for these components. They are available from most electronic supply companies as well. If you cannot locate the parts, please contact the publisher. There are kits available containing pre-assembled components if you cannot locate the parts yourself.

The laboratory experiments in this manual require the following components:

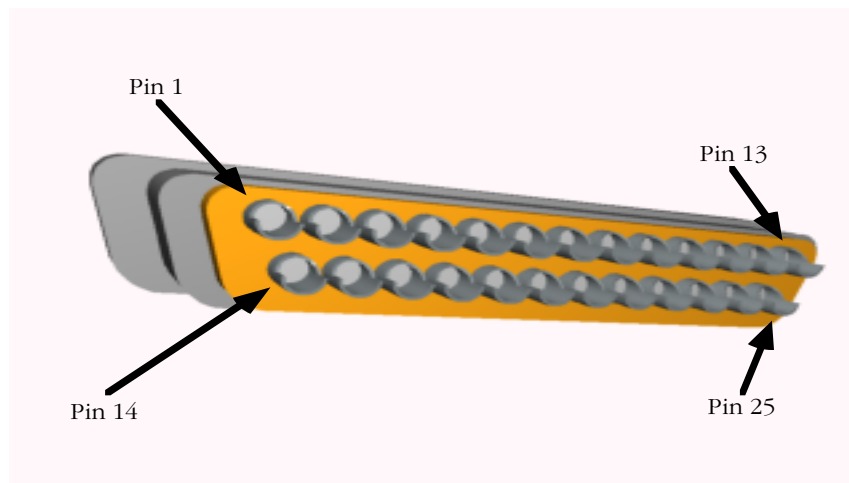
(8) Low-current LEDs (RS Part number 276-044). These are also known as high-output or high-brightness LEDs by other manufacturers.

- (1) Four-position dip switch (RS part number 275-1304c).
- (1) Momentary push button switch (RS part number 275-1556)
- (1) Solderless breadboard, experimenter socket, or protoboard (RS part number 276-175)
- (8) 2.2, 2.7, or 3.3 KOhm resistors (e.g., RS part number 271-1325)
- (5) 10KOhm resistors
- (1) DB25 female solder type connector (RS part number 276-1548b)
- (1) Spool 22AWG solid wire (e.g., RS part number 278-1221)
- (1) Pair wire strippers

In addition to these components, you will need access to a soldering iron and some solder to assemble the connector. You will also need a 6ft DB26 male to male cable, although the computer laboratory will probably supply this cable.

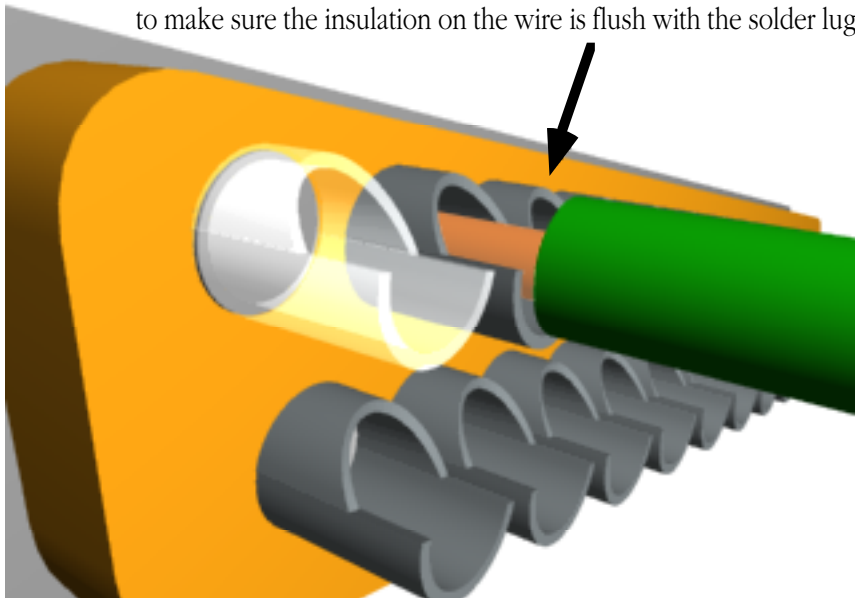
To assemble the connector for your kit, you will need to create fifteen strips of wire approximately one foot long. Strip about $\frac{1}{8}$ th inch insulation from one end of the wire and about $\frac{1}{4}$ inch insulation from the other. Eight of the wires should be one color, five of the wires should be a second color, and you should try to use red and black for the remaining two wires. If you have only three colors available, use the same color (preferably black) for the last two wires.

You should solder the first eight wires (all the same color) to pins two through nine on the DB25F connector. If you look at the back of the connector, containing the solder lugs, the pins have the following numbers:

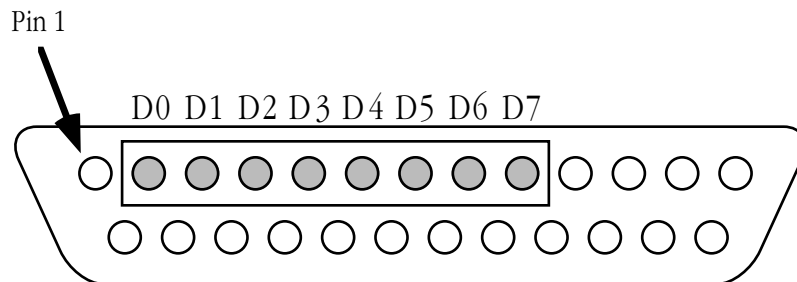


When soldering a wire to the DB25F connector, you must exercise considerable care. You want to apply sufficient solder to provide a good connection, but you do not want to apply so much solder that you form a *solder bridge* with an adjacent pin on the connector. Furthermore, you should not strip any excess insulation from the end of the wire that you solder to the DB25F connector. Ideally, the insulation should be flush against the solder lug on the DB25F after you solder the wire. Under no circumstances should you be able to bend the wire and have bare wire make contact with another wire or pin on the connector.

When soldering a wire to a solder lug on the DB25F connector, try to make sure the insulation on the wire is flush with the solder lug.

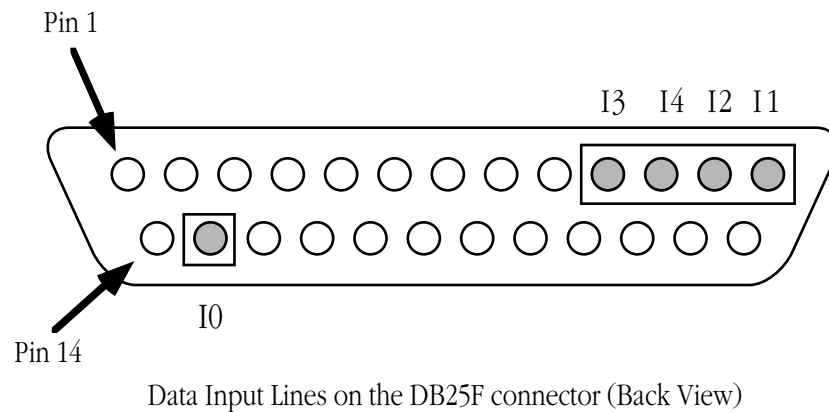


The wire connected to pin 2 is data line zero or D0, the wire connected to pin 3 is D1, the wire connected to pin 4 is D2, etc. These eight wires are the data *outputs* from the PC's parallel printer port.

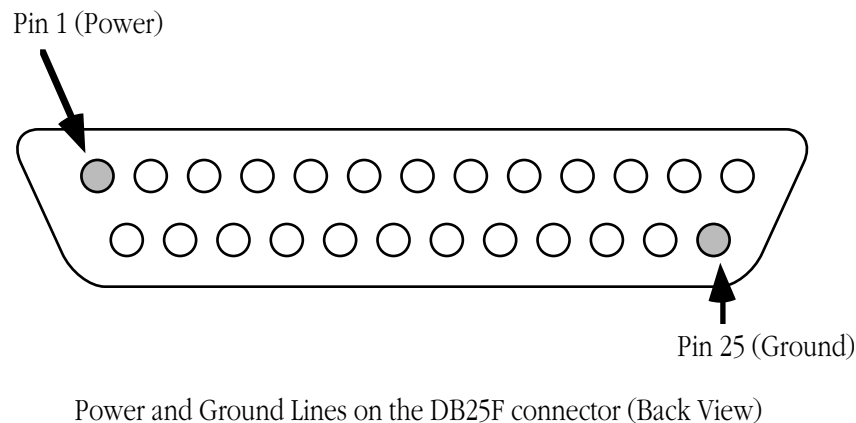


Data Output Lines on the DB25F connector (Back View)

The next five wires (preferably all the same color, and a different color from the eight wires above) handle *input* functions on the parallel port. You should solder these five wires onto pins 10, 11, 12, 13, and 15 on the DB25F connector. We will call these signals I5, I4, I3, I2, I1, and I0, respectively. Note that these inputs are not arranged in a logical order like the data output lines. Keep this in mind when labelling and using these signals.



The remaining two wires provide power and ground for the circuit. If you are using two colors (preferably red and black), solder the first (red) wire to pin one of the DB25F connector and the other wire (black) to pin 25 of the DB25F connector.



Note: If you do not have any soldering experience, you may want to get some help from someone who has soldered components together. Although soldering components together is not a difficult task to master, you can damage your components or even your computer system if your soldering job is really bad. Certainly have your lab instructor check over your job before you plug the connector into a computer.

Warning: Under no circumstances should you plug your DB25F (female) connector into the DB25M (male) connector that appears on the back panel of many different computer systems. The DB25M connector on the back of the computer is the *serial port*. You can damage your serial port hardware if you wire up a circuit described in this lab manual and connect it up to the serial port. You might also damage some of the components in your kit. To connect your circuits to the computer, use a 6 ft DB25M to DB25M cable. Plug one end of the cable into the DB25F connector on the back of the computer system (this is the parallel port), plug the other end of the cable into your DB25F connector.

