

Random Thoughts on Electronic Project Management: Design, Construction and Debugging

Peter D. Hiscocks

Department of Electrical and Computer Engineering
Ryerson Polytechnic University

phiscock@ee.ryerson.ca

September 2, 2002

Contents

1	Introduction	2
2	Project Management	2
2.1	Project Planning	2
2.2	Research	3
2.2.1	Extrapolating the Circuit	5
2.3	The Specification	5
2.4	Estimating the Project	6
2.5	Documentation	7
2.6	The Project Binder	8
2.7	Freezing the Design	8
2.8	Prototypes that Lie	9
2.9	Design for Debugging	10
3	Debugging an Embedded System	10
3.1	The Emulator	11
3.2	The Monitor Program	11
3.3	Built-in Debugging Hardware	12

3.4	Tips and Tricks for Debugging an Embedded System	12
4	The Hidden Schematic	12
5	Power and Ground Wiring	12
5.1	Wiring Resistance	13
5.2	Shorted Power Supply Wiring	13
5.3	Wiring Inductance	14
5.4	Analog and Digital Power Supplies	16
5.5	A/D Converter Grounding	17
6	Logic Interfaces	17
6.1	Logic Speed Requirements	18
6.2	Unused Inputs, Spare Gates	18
6.3	Unused Inputs, Active Gate	18
7	And in Conclusion, Folks . . .	19
	Bibliography and References	20

1 Introduction

This paper is a distillation of 30 years experience in electronics consulting, teaching a wide variety of electronics courses in Electrical and Computer Engineering at Ryerson University, and supervising hundreds of fourth year Electrical Engineering undergraduate thesis projects. As a result of the consulting practice, I've had some experience managing, designing and building significant electronic projects. All of these projects, I am happy to say, worked and functioned to spec. Most of them, I am even happier to say, were profitable.

At Ryerson, it's not sufficient for a thesis to do a paper on some subject: the thesis must be a project that is somewhat original and, at the end of the day, functions: it absorbs energy and does something interesting or useful. As a result of the thesis supervision, I've seen legions of inexperienced EE students bloody their noses on projects ranging from legged robots to retinal scanners.

Out of all this work, angst and adventure, I have some idea of how to manage a successful electronic project. Only a small subset of these ideas is explicitly taught in the classroom. The thesis project at Ryerson is one of them – an attempt to expose the students to some of the ideas in a context where they are receptive. However, there are many useful ideas that never get taught and have to be learned, either from a battle-scarred veteran, or on one's own.

Hopefully, this presentation will allow some students and practitioners to avoid the battle altogether.

2 Project Management

Forsight and planning at the design stage can eliminate many potential problems and make the remaining ones easier to diagnose and fix. The opposite of *debugging* is something I call *enbugging* – the process of incorporating new bugs in a project. In this section, I discuss how enbugging can be kept to a minimum during the design phase, with a commensurate reduction in debugging at the commissioning, or *system integration* phase.

2.1 Project Planning

As a paid consultant, or an academic, or an engineering manager hoping to start a project, or a student proposing a thesis project, you begin by making a *proposal*. In some cases, the proposal is in response to an *RFP*: Request for Proposal. Or you may be initiating a project - you have an idea, and you need a sponsor. Either way, you have to tell the potential sponsors what you intend to build with their money¹.

Proposal-writing can be fun. On the one hand, it requires some blue-sky imagining of a technical device, so a vivid imagination and familiarity with the technology are strong assets. Engineers kvetch about lack of recognition, but the project proposal is one place where a clever design is likely to be rewarded by the best recognition of all – financial support.

Familiarity with the technology: where does that come from? After all, you can't be an expert in everything, and perhaps this project involves technology that is unfamiliar to you. Perhaps you don't even have a concept of a solution. You may have to do some research.

2.2 Research

The Internet

The Internet has made this much easier than it used to be. Armed with Internet access and the Google search engine, you can turn up scads of information in very short order. There are sites that list connections to sources of information, and nice people² have put tutorials and design information on the WEB for everyone to use.

The manufacturers of electronic components have huge amounts of useful information on their web pages. They provide datasheets and Application Notes that show how to use their products. The App Notes are a tremendous source of information, because they are intended to be read and understood – they are not written to impress other academics. And much of the information in App Notes is extendable over a range of products, not just the manufacturer's part. For examples, check out the web sites for National Semiconductor, Linear Technology, Maxim and Analog Devices.

The Applied Science and Technology Index

There will come a time when *every* written document is available on the WEB. However, the Internet took off as a source of information in the mid-ninety's, so it's only the last seven years of humankind's intellectual output. At the time this is being written - early 2002 - we're not quite there yet. Lots of knowledge is locked up in books, magazines and journals³.

The trick is to find it. It's a sobering experience to visit the periodical stack area of a library such as the basement of the Sigmund Samuel Library at the University of Toronto, or CISTI in Ottawa. The shelves stretch as far as the eye can see, in the X, Y and Z dimensions. Somewhere, there may be something useful.

Fortunately, the folks at Bowker have been cataloguing this information for us. Each year, they monitor all⁴ the articles that are published that year. They categorize these articles by subject and author, and then generate an index for the year. Lately, these indices have been on CDROM, so they're easy to search. In the BC era (Before

¹Or, in the case of student, with your own money.

²Like, ahem, me

³The term *magazine* is used here to refer to a commercial publication that generates a profit and carries advertisement. The articles are reviewed by the editor, but not by the author's peers. An author is paid for his/her work. The term *journal* refers to an academic document that contains peer-reviewed papers. Often, the author is expected to pay for the privilege of being published, and journals can be horrendously expensive.

⁴OK, not everything - the list of the periodicals they monitor is finite - but it's pretty comprehensive for our purposes.

CDROM), the index was published as a paper volume every two or three months, and then in a larger version for the entire year.

A typical entry looks like this:

To find some particular subject material, it's a matter of going through the The Applied Science and Technology Index, year by year, and identifying all the useful articles. The library may have these articles in the stacks, or they may be able to obtain the articles via Inter Library Loan.

Books

On the one hand, because books take a long time to produce, they tend to be somewhat behind the arc of technology. The absolute latest and greatest *Gismo Mk III* is being displayed at a conference or trade show. It's predecessor, *Gismo Mk II* is being displayed in a magazine. *The Original Gismo* is described in a text book. Books are generally not the place to look for current technology.

On the other hand, a well crafted book is the benefit of careful thought. It brings together a wide variety of ideas in a systematic organization that is easy to absorb and use. For example, the book *Astronomy on Your Personal Computer* by Peter Duffet-Smith, absolutely saved my bacon when I had to write a control program for the solar telescope at the McLaughlin Planetarium here in Toronto. Anyone who uses the Linux Operating System has a free operating system, yes, and hundreds of dollars worth of books showing how to use it. So books have their place.

Your Private Library

Every time you engage in one of these research efforts, you'll find some information which is useful for the current problem, and possibly for other design projects in the future. It's a good idea to collect a couple of filing cabinets and fill them up. Eventually, you will get to the point where trips to the library become infrequent, and the information is ready to hand in your own office.

Research is No Substitute For Thought

One of the best ways to learn a technology is to look at the work of other practitioners. The Open Source software movement has been a boon in that regard - we can learn how to write programs by looking at the work of others.

Equally, we can learn to design circuits by looking at other people's schematic diagrams. To someone who is an aficionado of electronic circuit designs, it is an religious experience to look at the schematic for a piece of Hewlett Packard or Tektronix test gear. There is a wealth of circuit ideas in the back issues of *Electronics World* (aka *Wireless World*), *Electronic Design*, *EDN*, and many others.

However, having studied how other people have done it is no substitute for a careful and thorough understanding of how a circuit works. After all, if (or *when*) it doesn't work, you'll have to understand it to troubleshoot it.

It was 1966 or so, in one of the U of T Electronics Labs. I was working on a fourth year electronics project: a tachometer circuit, which will eventually find its way into a buddy's racing motorcycle, but at this point, it clearly doesn't work.

The circuit was something I found in a book. It was based on a unijunction transistor configured as a monostable. Each ignition pulse triggered the unijunction, which then dumped a fixed amount of charge into an ammeter. The ammeter averaged these charge pulses, and so the average reading was proportional to the rate of the trigger pulses.

This is now, and I can tell you how it was supposed to work. That was then, and as the lab teaching assistant observed 'I think you have no idea how that circuit works, Mr. Hiscocks.' He was right, and the observation would have been more useful if *he* had some idea how it worked. Neither of us had any idea, and in that university at that time, professors did not often visit the labs and certainly never helped undergraduates with circuit problems. Eventually, with the combination of blind tinkering and blind luck, the circuit worked well enough to use.

Then the zener power supply in the tachometer flattened by friend's battery before a race, and he had to scramble to get his battery recharged.

Fortunately, other projects were more successful and the university released me as a junior engineer to prey upon an unsuspecting public.

That experience stuck with me, so I guess university had some value. I never built another circuit until I understood it completely.

2.2.1 Extrapolating the Circuit

You must understand the circuit completely, not only so you can troubleshoot it, but so you can modify it to suit what is available. It is a real test of engineering skill to design a circuit that uses readily available parts, or to repair a circuit with what is available.

For example, just because an existing circuit uses a certain type of transistor does not mean that your circuit has to use the same type. A competent engineer will determine the specifications of that transistor (and with experience, this can often be done by eye) and substitute whatever is suitable and available.

Better yet, you may be able to take the general principle - charge balancing, negative feedback, thermal balancing, whatever - and come up with a better alternative.

So, in summary, research and the reading of other people's circuits is not about slavishly copying other designs. It's about learning enough to do good design oneself.

2.3 The Specification

This doesn't seem like much fun - after all, wouldn't it be more entertaining just to start building? But no, a project proposal is a useful discipline, because it requires you, the designer, to create a *specification* document.

The specification describes what is going to get built. This is most critical because it prevents mis-communications between the client - the person or entity that is putting up the cash - and the designer. Nothing is more horrible than arriving at what you thought was the deliverable only to find that the client is expecting additional features, or has no interest in some feature that you slaved over, or was expecting something totally different: you thought you were developing an anti-gravity machine, and they expected a time machine. To prevent this, it is essential to describe (a) who does what and (b) what you are delivering. For example, if it breaks, I'll fix it on site, but the

client has to pay for my trip to Whitehorse to do so. For example, we are providing an installed display, but the client is providing the electrical conduit up to the installation.

Is this relevant to the EE student doing a project? Sure: the the spec ensures that both the student and the supervisor agree on the deliverable and the details of the deliverable. This is important for both parties – the student and supervisor. It protects the student from *The Creeping Feature Creature* – features that the supervisor decides would be nice to add. And from the point of view of the supervisor, it ensures that the student understands what they have to do to meet the course requirements.

You can change the spec. It happens all the time. The original plan doesn't work, and something has to change. OK, let's change the spec. But the changes have to be communicated and agreed to by both the engineer and client. In some cases, the changes are required because of technical difficulties that were unforeseen in the original design. In that case, (depending on the contract), the engineers may have to eat the extra costs and deal with the problem. Or the contract may have a contingency clause that allows the engineers to charge more. For me, it's a point of pride: if the cost goes up because of something I should have foreseen, I deal with it.

On the other hand, if the client adds features, then the price goes up. Consultants love this, because additions can be hugely profitable. Woe betide the client who overlooks something that will have to be negotiated and added later to the contract: it will be expensive.

2.4 Estimating the Project

Clients often ask '*Is it possible to build a machine that will do X?*' to which the proper answer is *We can build anything. How long are you willing to wait, and how much are you willing to pay?*

The issue is not whether something can be done: the issue is whether it can be done to a specific deadline and budget. Before embarking on the adventure of developing an electronic device, or a new computer program, it is essential to determine whether this is the case or not.

Putting it another way, we're into an issue of *risk management*, one of the primary activities of engineering. New projects, things that haven't been done before, involve risk. There's the risk of failing to complete the project, and looking silly. There's the risk of losing your shirt, if you're a consultant or entrepreneur. There's the risk of wasting time and money on a bone-headed idea, which may result in your demotion to the bottom of the food chain. Some risk is unavoidable – if we all did projects devoid of risk, I'd be writing this on a mud tablet. On the other hand, there is no need to behave like adolescent males on a small-town Saturday night. Some careful thought and planning can predict possible problems and prepare for them. This is mature behaviour. And it leads to successful projects, the ones that come in on time, under budget and look *easy* when they are done⁵.

Okay, we agree that it's a good idea to do an estimate. How do you do it?

If you're experienced, you may be able to look at a project and guesstimate cost and time. Maybe you did something like this before, and you know how long that one took, so you can ballpark the price and time without a great effort. This can be risky, but clients often put you on the spot and want some sort of estimate, so it has to be done. (Then, when you give them the final price, it's always higher because you think of all the things you

⁵A few years ago, I was hired to design and build a large *orrery* - an installation for a local museum. This was in a new gallery about astronomy and the universe, and there were a bunch of new exhibits beside mine. I and the other contractors beavered away. My most vivid memory of this thing is installing and wiring the 874 light emitting diodes that defined the motions of the planets. It took weeks of mind-numbing work, but eventually the whole thing was working. At the end, we - the museum technicians and I - had a nasty bug to find in a remote-controlled slide projector. (It turned out to be a problem the slide carousel!) In any case, as we approached the deadline, I was clear and done. The exhibit across the aisle - a multimedia database thing, when such things were novel - was not, however. The two developers laboured mightily and visibly to make the deadline. In the end, it didn't work completely correctly, but to everyone's relief, it did *something*.

Guess which exhibit got all the attention and praise: the exhibits that were complete and working days before the opening, or the one that was running late and required herculean efforts to finish? That episode taught me something: if you make it look difficult, they'll appreciate you more.

should have taken into account, and didn't, when you gave the 'back of the envelope' estimate. This can make the client unhappy, but that's the way it works. Obviously, if you first estimate high and they don't throw you into the street, you can make them very happy with a lower final price.)

A more scientific method, which is essential practice if you are beginning at this, is to *break the project into small pieces, estimate the time and price for each small piece, and add them up*. The key to this is the granularity of the decomposition: each part of the project has to be small enough that you can accurately estimate its time and cost.

Furthermore, each of these *project atoms* has various aspects associated with it. Each one will require design, documentation, implementation, debugging and testing. If it's hardware, it will have in addition a part cost and a time associated with tracking down the components.

Consequently, the project estimate is composed of two matrices (spreadsheets): one for time, and the other for cost. This could be visualized as a 3 dimensional matrix: each horizontal row is one of the project components (atoms). The vertical columns are the various activities required to develop that component to completion. The matrix is two layers deep: the top layer is hours, the second layer is material cost.

In some electronic projects where you are building one off, the cost of components may be irrelevant compared to the labour costs. (I found that the cost of the electronic components paled to insignificance compared to the cost of the hardware: the case, the connectors, the wire. A \$30 mil-spec connector rather dominates the 20 cent transistors.) On the other hand, if the product is to be produced in numbers, the component cost may be critical and you may be required to include every 2 cent resistor. Part of the estimate is to determine whether the device can be produced to a particular cost, and therefore can be profitable.

2.5 Documentation

Documentation is one of those motherhood issues that everyone agrees is a *good idea*, but most people don't practice. This subject reduces your otherwise amiable author to a raging maniac, so excuse me while I rant for a bit.

It is utterly astonishing to me how many students attempt to build and debug a complex electronic project, often with embedded hardware and software, but completely without documentation. The operative excuse is *I'm going to do the documentation once the project is working, but I have to debug it first*, an excellent case of placing the cart before the horse.

On being asked to intervene, when the prof asks for a schematic or code listing, he/she is presented with a pathetic collection of what looks like chicken tracks on scrap paper. The so-called schematic is often a marked-up version of a manufacturer's datasheet.

When you are doing an electronic project, you are generating intellectual property: ideas. The *thing* itself is the hardware embodiment of the ideas – it is proof that, in at least one instance, the ideas can be made to function. But the thing is not that important – it's the ideas. And the ideas are written down on paper, or coded into magnetic tracks on a computer disk.

Someone, either a client or whoever is funding the university education, is paying for you to generate those ideas. They are the output of this process, so they have to be complete, and they have to be safe.

Consider, for example, the situation of a movie shoot. We've all seen them: the row of trailers, the lights, the cameras, the people, the talent – even a food wagon! Dozens of talented people beaver away for hours. At the end of the shoot, what is there to show for all this effort and expense? Some images on a roll of film, or encoded into magnetic tape. Information. Intellectual Property. The generation of IP is Big Business, and it has to be done properly. Those images, on that precious roll of film, are worth a fortune⁶.

⁶At one point, I made electronic props for a movie. (*Kidnapping of the President*, with William Shatner, since you ask. The 'Bomb'). I

So, when a student directs me to look at their electronic hardware, they are focussing on the wrong thing: I need to see the *idea*, and the idea is the schematic, and the other drawings. Sometimes the idea is fine, and it is the implementation that is a problem. But the idea comes first.

2.6 The Project Binder

The ideas must be protected. By all means choose a system that works for you. Myself, I believe in binders. File folders don't impose a chronological order on things, and that can be important. A binder tends to sequence things. As well, there is a physical act in *removing something from the binder*. You're more likely to put it back, and in the right place. And binders can be fitted with dividers.

Projects have different parts to them, and these should be associated with different sections of the binder (or container, if you prefer.) Some of the project areas, corresponding to sections of the binder, might be:

Correspondence Negotiations with the client, or communications from the supervisor.

Design The *stream of consciousness* that documents progress and ideas towards a final design. In companies where such things are formalized, the design ideas may have to go in a lab notebook that is witnessed each day and locked away. In any case, the ideas should be dated and pages numbered, so they can be replaced in correct order. Yes, you will have to go back to it later.

Datasheets Collections of specifications for the various components.

Reference Material Helpful articles and papers by others.

Drawings The final version of each part of the design, both electrical and mechanical. For example, this would include computer-drawn versions of the schematics, printed circuit board layouts, connector pinouts and assembly diagrams.

Software Source code for the programs, software schematics, specifications for data structures.

Parts Lists This section lists the parts that have been selected, along with their suppliers and the cost. A spreadsheet is ideal for holding this data. Then a bill of materials and project cost can be generated easily.

The ultimate tests are these: If you got a better offer from another company, could someone else continue the project? If you had to leave this project for six months and then pick it up again and continue, could you do it?

2.7 Freezing the Design

I will have many suggestions to make about the project design - the details of the electronic circuit and the software. However, first some meta-ideas.

There is a great tendency to *buy parts and begin building* or, in the case of software development to *start coding*. After all, while you are still thinking about the project, there is nothing to show (except, of course, a progressively thicker project binder.)

But this is engineering: we are generating ideas. Moreover, the ideas have many aspects and implications – they all have to work harmoniously together. A change in the hardware could impact the choice of microprocessor

discovered that you can't invoice a film company. Once they've filmed your prop, they have no interest in paying. In fact, the original company may be dissolved. Fortunately, they inadvertently blew up my prop when they blew up the truck. They needed more shots of the bomb, so they had to get me to build another one. As you can imagine, it was a little more expensive the second time, and the terms were strictly Cash In Advance.

which would change the programming language – and so on. As the project progresses, decisions made at the start will have huge implications. So these decisions must be the correct ones.

This is not to say that the designers can dither forever. There comes a time when we have to decide: are we here to fish or cut bait? The decisions are made, and we have to live with them.

But it is easy, especially when fast-tracking a project, to try to do too many things at once. For example, it is a very common mistake to finalize the design of the electronics and then require that someone make a suitable power supply for the thing and package it in a nice enclosure. But the design of the power supply may have important implications for the electronics – it may need separate analog and digital supplies, for example. Or we might find that we could get a much cheaper power supply for a slightly different voltage. Or we might be able to save a bundle by operating from +5 volts rather than +5, +12 and -12.

In terms of the mechanical design, the box not only keeps people from putting their fingers into the thing, it has to extract heat from the circuit and direct it to the outside world. It has to support connectors, switches and indicators, and all these impact on the circuit design. Design the whole thing, including the mechanics, before you buy parts and begin building.

So:

Do not freeze the design too early.

Do it properly. Do the design, completely. Finish the drawings. Check the drawings. Then you can build the thing, confident that most of it is going to work, and fit in the box.

2.8 Prototypes that Lie

It is common practice to build a *proof of concept* circuit in order to establish that the basic idea is sound. The concept circuit is often a *lashup* – ugly as sin, resembling nothing as much as a piece of found art – resistors sticking out at odd angles, wires looping through the air, sections held together with duct tape.

All well and good: put a minimal investment into construction at first. Build it fast, and find out whether this clever concept will fly at all. But beware: concept prototypes can lie to you.

They lie because of the problem of *scaling*. Perhaps you are building an RF link, and testing the lashup circuit in the lab, it works fine over a space of a few meters. To work of a kilometer is a simple matter of antenna greater gain and increased transmitter power. Well, maybe, maybe not.

Or the new high-speed preamplifier works fine except for a bit of noise. It can be scaled to greater sensitivities, and putting it on a four-layer board will cure the noise problem. Well, maybe, maybe not⁷.

Either you need to do very careful calculations that indicate a high probability of success in scaling up or down, or you need to build the final scaled version of the circuit. For example, power dissipation scales as the *square* of the voltage or current. So a power supply that is doubled in size may not simply need more heatsink area – it may need fan cooling as well. You can predict that in careful calculations, but you need to be aware of the problem.

The other danger of a lashup is more insidious, especially if it works. It hasn't been tolerated properly. I once worked with an engineer who suffered from this problem⁸. He did his circuit design by building a lashup where the resistors were represented by decade boxes that could be adjusted. He would adjust these resistors until the circuit worked, and then copy their values onto the schematic drawing. Done.

⁷If you think the Printed Circuit Board (PCB) layout will improve things, you should do the layout and get a prototype PCB made. This is getting much easier to do: there are companies that specialize in fast turnaround prototypes, and there are good ways to get the boards made in house.

⁸Actually, I did much of the suffering, since I had to fix the production units.

This isn't engineering: this is tinkering. If you only have to build one device, then it's fine. Tidy it up and ship it. But if you build hundreds, the transistor gains are going to be all over the normal distribution, including ones at the extreme high and low limits of allowable gain. The resistors and capacitors are going to exhibit their tolerances too⁹. The result, unless it was designed to take it, is that a fraction of the completed units will simply not work properly. In these days of such fine circuit simulators, this is inexcusable.

Design the circuit. Lash it up. It works? Good, you now have exactly one working circuit. Now plug the design into the simulator¹⁰ and run the component values through their range of values. (If the combinatorial explosion makes this impossible, use a Monte Carlo statistically random simulation of device variations.) They all work? Great, you have a production design.

2.9 Design for Debugging

Whenever you design an electronic circuit, or write an embedded computer program, or design a mechanical enclosure, you should ask yourself: when this doesn't work (not *if*, but *when*), how am I going to diagnose the problem, and how am I going to fix it.

In the case of the electronic circuit, this means that you have to be able to get at the signals, and bring them to the outside world for measurement on a voltmeter, display on an oscilloscope or observation on a logic analyser. In the case of modern IC's, with their fantastically small pins and pin spacings, this can be a challenge.

- If the signal is a critical current, how can you measure it? Can you break the circuit and insert an ammeter?
- If the circuit is an important pulse, can it be directed to the oscilloscope without contaminating it with noise? You might need a dedicated connector to provide this feature.
- All circuit measurements required grounding instruments to the board. Design in ground points, close to the measurement, where you can attach the scope ground clip.
- Would this circuit be easier to troubleshoot if I can separate it into functional blocks? For example, should the power supply disconnect from the microcomputer section so it can be checked independently?
- In mechanical terms, can I get the case open and the circuit board accessible in a reasonable length of time? Is there some packaging scheme that makes this easier?
- Are there certain signals that can be indicated on an LED to make it easier to debug faults? For example, it's really handy to have a power-on LED, LED indicators on transmit and receive serial port lines, and an LED to indicate that the microprocessor is still alive. The cost of a few LEDs can really improve the useability and testability of a circuit. You don't need to hook up any equipment to diagnose certain problems. It is ridiculous to have to hook up an oscilloscope to discover that a communication cable has gone intermittent.

3 Debugging an Embedded System

An *embedded system* is an electronic system which includes a microprocessor (or *microcontroller*, if you prefer) as part of the system. An embedded microprocessor can eliminate boards of discrete logic and incorporate wonderful features that make modern devices possible and profitable. However, they can also be very, very nasty to debug.

⁹Electrolytic capacitors can exhibit tolerances of -20/+100%! That is, the actual value could be *double* the nominal value.

¹⁰A spreadsheet can be very useful for this, too.

The first embedded computer system that I worked on, a Digital Equipment PDP-8, had a console - lamps that read out the internal values, and switches that could be used to stop the machine, set a program address, and single step the machine¹¹. It also cost more than my annual salary at the time. A modern microprocessor costs less than a meal at your local diner, with at least 4K of memory thrown in. In quantity, the same device probably costs less than a chocolate bar.

However, the modern embedded microprocessor doesn't have a console. Worse, it may not even have a *connection* for a console. In order to have access to the inner workings of the device, the person who is doing system integration will need some sort of *user interface* to the microprocessor.

Specifically, you and the embedded microprocessor will need to interact in the following ways:

- Download test code from the host workstation into the microprocessor.
- Execute the test code, or sections of the code.
- Set one or more breakpoints¹² in the code.
- Upload memory and machine register contents to the host for display and diagnosis.
- Manually modify memory or register locations.

3.1 The Emulator

The emulator is a clever piece of hardware in the form of a box or circuit board with two connections. One connection plugs into the target system microprocessor socket and takes the place of (*emulates*) the target processor. The other connection goes back to a development station – usually a Unix workstation or PC – which provides the user interface. From the workstation, the developer can insert breakpoints into the code, do memory dumps, trace code execution, maybe even capture logic signals.

The emulator hardware is cleverly arranged so that it is essentially invisible to the target and can run at the full speed of the microprocessor in question. For complex systems, an emulator is almost a necessity. Unfortunately, an emulator may be extremely expensive and require a new investment every time the target processor changes. It may require a special version of the microprocessor, called a *bondout*, where internal signals are brought to the outside world. However, if it enables elusive hardware-software interaction problems to be identified and fixed, it will pay for itself.

Some emulators are affordable. For example, the *MPLAB* equipment from Microchip, used to debug a PIC microprocessor system, is very reasonably priced. Microchip also offer different versions of their microprocessors – development units which are easily programmable, and production units which are low cost.

3.2 The Monitor Program

If the emulator is the high-priced and heavy-duty solution to microprocessor system debugging, the monitor is the cheap and cheerful approach. In this arrangement, a small program is installed in the microprocessor memory. This program includes enough intelligence to communicate with an external host, set breakpoints, receive test code, dump memory and so on. Unlike an emulator, the monitor program takes over some of the resources of

¹¹The 4K magnetic core memory for this thing cost 1\$ per 12-bit word.

¹²A *breakpoint* is an address in the program. When the microprocessor reaches this address, it stops. This simple device enables one to determine that the machine is executing certain sections of code, or to determine what the result is after executing a certain section of code. Sometimes, it is possible to specify *watchpoints*. A watchpoint is a specific memory location. When it is changed by the program, the program halts.

the target processor and it is not transparent to the operation of the microprocessor. For example, it may require sacrificing an interrupt vector and certainly will require the use of on-board RAM for house-keeping. It will probably need a serial port of some sort (software or hardware) to communicate with the host PC. Some of the memory map must be dedicated to the monitor code. If this is acceptable, a monitor program is often sufficiently powerful to debug a complete application.

One problem with a monitor is this: the microprocessor must have certain facilities to support a monitor in the first place – a serial port, some scratchpad RAM and loadable memory for the program, for example. The ultimate design may require none of these facilities. Consequently, you have to be prepared to build two systems: one that supports a monitor program, and a second production system that takes out all the unnecessary hardware and software.

3.3 Built-in Debugging Hardware

Recognizing that debugging is An Issue for engineers, some companies are providing built-in facilities that facilitate debugging. For example, the Motorola MC68HC908 series of processors include a one-pin bidirectional interface that can talk to a host, together with a small built-in monitor program and a hardware breakpoint register.

3.4 Tips and Tricks for Debugging an Embedded System

4 The Hidden Schematic

Many of these design concepts relate to the concept of *the hidden schematic*. The hidden schematic is the wiring and the components that you didn't explicitly choose to be part of the design: they came along as *parasitic components* with the real components. Or they arise as a consequence of the wiring design. That innocent looking ground symbol has an entire schematic network associated with it, and you ignore the network at your peril.

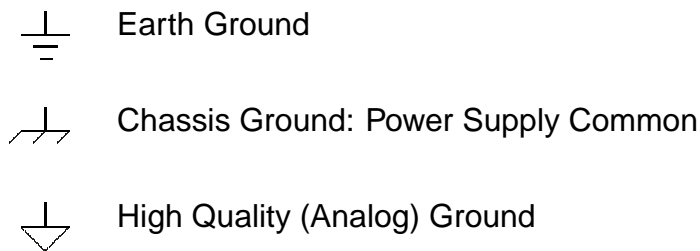


Figure 1: Grounds

5 Power and Ground Wiring

Any length of wire or printed circuit trace has three parasitic components associated with it:

- Resistance, which causes a voltage drop across the length of the wire as current flows through it
- Series Inductance, which causes a voltage spike to occur across it when the voltage changes and encourages rapid current changes to travel in such a path as to minimise inductance.

- Mutual Inductance, which couples the magnetic field of a wire into another (parallel) wire.
- Mutual Capacitance, which couples the electrostatic field of a wire into another wire or copper area.

Of course, you can't easily take into account every wiring parasitic in a circuit - it may be hand-wired as a prototype, and then changed later, laid out as a printed circuit board. Fortunately, it's not necessary if you recognize where the wiring parasitics are important and plan for those occasions.

5.1 Wiring Resistance

Wiring resistance becomes significant when small voltages are being measured (12 bit resolution in an A/D converter, for example) and high currents are involved. What is a *high current*? Well it depends: for small diameter wire, it may be as little as an amp or so. Whatever is sufficient current to set up a voltage across the length of wire.

A high current load should have its own direct connection to the power supply. Ideally, the circuitry that is sensitive to noise should have its own separate power supply (or regulator). When the high current load and the low power circuitry must share the same power supply, they should have independent supply and grounding wires that go directly back to the power supply terminals: the regulator output, if it's a regulated supply, or the filter capacitor terminals, if it's unregulated.

The power supply, one hopes, behaves as a pure voltage source, with insignificant internal resistance, and will hold its output voltage constant with changing current.

Provide a high-current load with its own power leads back to the power supply terminals.

One often-unexpected situation involving high currents is in a capacitor-filtered power supply. The charging current into the filter capacitor is a short, high-amplitude pulse that can create a significant voltage across a connecting wire. For full wave rectification, the pulse occurs at twice line frequency - 120Hz in North America - and may induce hum in an audio stage.

For the charging loop between a full-wave bridge rectifier and its associate filter capacitor(s), keep the wiring short and heavy. Connect loads *at the capacitor terminals*, not at the bridge rectifier.

Notice that adding filter capacitors to a wiring system does not fix the problem where a high-current load is creating a voltage drop across the resistance of some signal wire. Capacitors can stabilize the power supply lines against transients, which are signals of brief duration. If the interfering signals exist at low frequencies, the wiring must be re-routed.

Size wiring appropriately to the current being carried and the allowable voltage drop. If you are in doubt, consult a wire table and calculate the voltage drop for the chosen size of wire.

For example, suppose we are driving a small DC motor which requires 0.5 amps starting current. We happen to wire this with a 1 foot length of #32 solid wire (wire-wrap wire). What's the voltage drop across the wire? The wire table tells us that #32 has a resistance of 165Ω per 1000 feet, so the resistance of our one-foot length is 0.165Ω . The voltage drop would be $500\text{mA} \times 0.165\Omega = 82.5$ millivolts. An 8 bit A/D converter with a 5 volt range has a step size of $5/256 = 19.5$ millivolts. If the wiring is such that the motor current induces this error voltage as an error in the A/D reading, the 82.5 millivolt drop represents about 4 steps on the A/D converter.

5.2 Shorted Power Supply Wiring

It is not uncommon to find a dead short between power supply positive and the ground. For example, a particular component may have died as a short-circuit connection between V_{cc} and Gnd , or there may be a PCB defect, or

there may be a tiny solder bridge linking the two. (This is particularly a problem when every unused part of the board is a copper ground plane.) This is nasty, because there are *many* things connected to power supply and ground. Where is the short circuit?

This is where wiring resistance works to your advantage. Pump a substantial current through the power supply line, and the wiring resistance will create a *voltage gradient* that points to the short circuit.

A lab power supply with adjustable current limit is required. Connect the supply to the defective board, turn on the supply, and then increase the current control until you have something like an amp or so flowing through the short-circuit. Obtain a sensitive voltmeter – something that can indicate millivolts. Connect the negative lead of the voltmeter to the negative of the power supply. With the positive test lead of the voltmeter, start at the negative terminal of the power supply and move the test lead along the power supply ground wire *in the direction of increasing voltage*. You should be able to follow the voltage gradient back to the positive terminal of the power supply. In the process, you should encounter the shorted device or shorting conductor.

Incidentally, it's a good idea to check the power connections of a new circuit board, whether hand-wired or printed circuit, *before* loading it with components. It's a lot easier to find a short circuit on a bare board.

Sometimes, when the offending connection is a copper whisker between traces, the substantial current is enough to blow away the connection and fix the board.

Always check a new board for shorts between the various supply lines.

To find a short between a power supply line and ground, dump some current down the line and trace the voltage gradient.

5.3 Wiring Inductance

At first blush, the currents drawn by modern digital circuits such as the 74HC (High Speed CMOS) series, are negligible and therefore inconsequential. Indeed, the voltage drop due to IR losses - the resistance of the wiring - is not important. What *is* important is the power supply wiring *inductance*.

There are several factors that conspire to make this an issue:

1. The switching time of modern gates is in the order of nanoseconds: 10^{-9} seconds. Consequently, in the equation

$$v_l = L \frac{di_l}{dt}$$

the voltage v_l created across an inductor is proportional to the inductance L and the change in inductor current di_l , and inversely proportional to the span of time dt that the change occurs. As dt grows ever smaller (with increasing logic switching speeds), the induced voltage increases.

2. When a logic gate is in the high or low state, its current is negligible, and hence the static power consumption is negligible. When a gate switches and the output transitions through the active region half-way between V_{cc} and ground, the gate supply current increases significantly. So there is a momentary pulse of current from the power supply.

Reference [4], page 1-21 indicates that a typical current pulse in HC (High-Speed CMOS) logic is 20mA over a time-span of 5 to 10 nanoseconds. For AC (Advanced CMOS), a faster logic family than HC, the current pulse is given as 60mA over the same time-span.

- The loads driven by modern gates are largely *capacitive*. The driving gate does not have to supply significant current to drive some logic gate load, when in the quiescent state. But when the logic level changes, the gate must charge or discharge the load capacitance. The equation

$$i_c = C \frac{dv_c}{dt}$$

describes what happens: as the capacitor C is charged or discharged in a short time (dt again) the capacitor current i_c must increase. This current must be supplied via the inductance of the positive power supply line and returned via the inductance of the ground return line. In both cases, the changing current will generate voltages across those inductances.

- As a rule of thumb, wiring inductance is in the order of 5nH/cm (see reference [2], section 9.11, and reference [4], page 1-21).

Consider, for example, that a logic gate is switching 20mA into a capacitive load, in an interval of 5 nanoseconds. The power supply wire is 10cm in length. Then the induced voltage will be:

$$\begin{aligned} L &= 5\text{nH/cm} \times 10\text{cm} \\ &= 50\text{nH} \\ v_l &= L \frac{di}{dt} \\ &= (50 \times 10^{-9}) \times \frac{(20 \times 10^{-3})}{(5 \times 10^{-9})} \\ &= 0.2 \text{ volts} \end{aligned}$$

Each time the logic gate switches, it will generate a 200 millivolt spike on the supply line. Now multiply this by several gates switching simultaneously, and there may be real problems caused by this noise.

What are the solutions?

- The most important solution is to provide a reservoir of charge close to the integrated circuit, in a so-called *bypass* capacitor. This capacitor must be able to dump its charge quickly, so the leads of the capacitor must be short in order not to add inductance in series with the capacitor. A ceramic capacitor, with short leads, is ideal for this. A typical value is in the range of 10 to 100 nanofarads. With a faster, more complex logic device, the need for bypassing increases in importance.

This advice – always decouple logic circuits – applies not only to digital logic but many other devices as well. The ubiquitous 555 timer, in its original form, generated horrific power supply spikes that would trigger nearby logic. A 100nF capacitor had to be installed, without fail, next to the device. I spent one entire afternoon troubleshooting a problem of mysterious intermittent behaviour of some logic circuitry, which eventually turned out to be the 555 power-supply-noise-problem.

In another case, mysterious behaviour of a 16V8 GAL device turned out to be dependent on adequate bypassing at the device. (Students were putting the bypass capacitor at the far end of the protoboard, where it had no effect.)

- Use the slowest logic that is acceptable in the application. As indicated above, the switching current pulse of AC logic is 3 times the amplitude of HC logic. Furthermore, faster edges require larger currents to charge and discharge any load capacitance.

3. Ensure that the ground and supply line enclose the smallest possible area. The inductance of a loop of wire is proportional to its area, so reducing the enclosed area reduces the inductance.

Ideally, the ground return should be a continuous plane of copper on the opposite side from the supply wiring. If this is the case, the return path of the transient current will naturally choose the path of least inductance. In fact, it will choose a path that causes it travel in the ground plane directly underneath the source conductor, because this minimizes the enclosed area and the inductance. If possible, the ground plane should not be perforated with slits or open areas, because this forces the ground current to travel apart from the supply current, with a consequent increase in loop area.

For this reason, many digital logic boards are constructed in four layers. From top to bottom:

- Top side signal wiring
- Power supply positive
- Power supply return (digital ground)
- Bottom side signal wiring

Not only does this *minimizing of loop area* result in smaller voltage transients on the power supply lines, it significantly reduces EMI (Electro-magnetic Interference), radiated interference.

Use the slowest logic that is acceptable in the application.

Ensure that the ground and supply line enclose the smallest possible area. If possible, use a ground plane for power supply return.

5.4 Analog and Digital Power Supplies

Many systems contain a mix of analog circuitry, such as preamplifiers and A/D converters, driving digital circuitry. In some cases, you can get away with operating everything from the same ground and supply lines. But if the amplifiers are high gain, or the A/D converter must detect millivolt level signals, it's best to provide the analog circuitry with a separate power supply system.

Sure, op-amps are supposed to ignore noise on their power supply pins. But single-supply op-amps have poor positive power-supply rejection at the frequency of digital glitches. Some of them have *no* power supply rejection on the negative supply line – so-called ground. In other words, a pulse of N millivolts on the ground finds its way to the output of the amplifier as N millivolts!

If both the logic and analog devices operate from the same 5 volts, you may be able to provide separate power supply and ground leads for each, and 'star ground' the lines back at the common power supply. If digital noise contaminates the analog power supply, you can add a resistive-capacitive or inductive-capacitive filter in the analog supply line.

If you are laying out a circuit board with this scheme, and the analog power line and digital power line connect together back at the power supply, the autorouter program will think that they're the same net. It will then mindlessly intersperse digital and analog power supply connections, exactly what is not wanted. You need to keep the analog supply line and digital supply line separated and given distinct names, such as +5A, +5D. Get the autorouter to bring the leads back to the power supply area, and then put a jumper into the PCB to wire them together. This has other advantages: at a later date, you can separate analog and digital power in the event that they seem to be interfering and must be mutually filtered. The jumper can be used as a test point for board power supply voltage. Similarly, the grounds need to be separated and given distinct names, such as AGND, DGND, and brought back to jumpers.

I now do this as a matter of course, even on very simple boards. It's easy to do, and it makes it much easier to fix power supply crosstalk problems.

If the performance requirements are severe (a 12 bit A/D converter in the presence of high-speed digital logic, for example), the analog circuitry and the logic should each get their own power supply. In these days of inexpensive three-terminal regulators, this is easy to do and goes a long way towards quiet power to the analog system. Even with this measure, you may well have to add series inductances in the supply lines to isolate noise.

Use separate power supply and ground leads for the analog and digital sections of the circuit. If they use the same voltage, star connect them back at the power supply.

If the analog power supply and ground must be particularly clean, design in stringent separation with individual power supply regulators and isolating filters on the analog supply and ground lines.

5.5 A/D Converter Grounding

An A/D converter is both an analog circuit and a digital circuit. The front end has to resolve millivolt level signals while the back end produces digital level signals with nanosecond transitions. It's amazing technology, but it has to be wired up carefully to work correctly. Here are some hard-won pointers:

1. Keep the analog power-ground pair separate from the digital power-ground pair, as advised in section 5.4.
2. Do not extend a digital ground plane under the A/D converter.
3. Read the A/D datasheet *very, very* carefully. Pay special attention to the applications circuits and the way they are grounded. For example, it is common to find that the A/D pin labelled DGND is the ground return for the digital section of the A/D but (for best noise performance) should be returned to the analog ground!
4. Use special care in the connection of the A/D reference voltages. For reliable high-resolution operation, these need a clean source of power, which rules out the digital power supply (unless it's carefully filtered.)
5. In one case, I was designing a video wall system that inhaled standard video and delivered it to a computer-type bus. There were three 8 bit video A/D converters and D/A converters (one for each of the Red, Green and Blue signals) that shared this same bus, and they were multiplexed onto the bus along with various bits of shared RAM. I found that digital noise was contaminating the video signals. This eventually turned out to be capacitive feedthrough from the digital bus, back through the A/D converter to its input (even when the A/D converter was tri-stated off the digital bus), and from there into other analog video signals. The solution: an octal buffer between the A/D output and the digital bus.

Digital noise may leak backwards through the A/D converter, in which case you should buffer the output of the A/D.

If you are designing and debugging a high-speed A/D converter, read references [6],[7],[8],[9] and [10]. They contain valuable information.

6 Logic Interfaces

When one specific type of logic is used throughout a design, interface problems are avoided entirely. However, in many real-world designs, the logic has to accept inputs from the outside world – switches, or analog signals – and/or deliver it to the outside world, with high-current loads or non-standard voltages. Many potential problems occur at these interfaces.

6.1 Logic Speed Requirements

A logic inverter – the basis of much logic circuitry – is really an overdriven amplifier. The amplifier has fast response time and spends most of its life saturated or cutoff. However, it does briefly transition through the active region, and while there, it behaves as an amplifier.

If the input signal spends an appreciable length of time in the active region, and the conditions are right for a signal to find its way from the output of the gate back to the input, the gate may *oscillate*, generating a train of high-frequency pulses¹³. Depending on the frequency, these pulses can be hard to see on an oscilloscope – you have to have a fast scope, good probes and good measuring technique. Depending on the design of the logic, these pulses may wreak havoc with what follows.

A comprehensive table of maximum allowable rise times is given in reference [4], page 1-61. For example, assuming a 5 volt logic swing, the maximum transition times for various logic families would be:

Logic Family	Maximum Transition Time, nsec
74F	75
74HC	400
74AC	50

So, for example, if you have a photo-interrupter driving AC logic, you must ensure that the output of the photo-interrupter transitions through 5 volts in less than 50 nanoseconds. This can be difficult to achieve.

If there is any doubt that this is a problem, pass the signal through a Schmitt Trigger device, such as the 74HC14 Hex Schmitt Trigger. (In an FPGA design, you would specify Schmitt Trigger input configuration.) The Schmitt has enough logic hysteresis that a slow logic signal will not cause oscillation, and the output of the Schmitt always transitions in a few nanoseconds, suitable for high speed logic.

Logic gates require fast-transition signals. If in doubt, square up the signal with a schmitt trigger.

6.2 Unused Inputs, Spare Gates

Ground the inputs of any spare gates in a package. This isn't critically important when debugging the first prototype, but it's something that should be taken care of in the final design. Otherwise, the input will float anywhere between V_{dd} and ground. If the input strays into the active region, it will cause the gate to conduct significant current through its output stage. Multiplied by many gates, this sends up the power consumption unnecessarily (see section 5.3).

6.3 Unused Inputs, Active Gate

It is *extremely* important that spare inputs on a logic device that is used in the circuit be connected to V_{dd} or ground. This is particularly easy to miss on logic devices that have multiple select lines. It's an insidious problem with CMOS logic (the usual case these days), because the floating input may float in the correct position (high or low) during debugging¹⁴. However, as a corollary of the Law of Maximum Aggravation, you can be sure it will act up at the most inconvenient time, such as when it is being observed by an Important Client. In a complex

¹³This behaviour is sometimes put to use, when 4000 series CMOS inverters are used as linear amplifiers. In this case, the output is connected to the input through a resistor, which biases the input and output into the active region. 4000 series logic is slow enough that it doesn't tend to oscillate as much as higher frequency logic families – and if it does oscillate, it's easy to see on the scope.

¹⁴Grizzled Veteran Note: TTL logic, now obsolete but still found in much extant equipment, can tolerate unconnected inputs, because the device reads them as HIGH. Not true for CMOS.

circuit, this can be difficult to find, because it may be intermittent and depend on the ambient humidity, phase of the moon, mood of the circuit, and so on.

If you think you might have a floating pin problem, connect a 10K resistor to the positive supply and run it across all the possible package pins. Then connect the resistor to ground and repeat. When the circuit bursts into correct operation or stops operating as you do this, you have found the offending pin.

Incidentally, your PCB layout or schematic capture program may be able to warn you if you have an unconnected input pin.

7 And in Conclusion, Folks ...

There are many rewards in electronics and electrical engineering. There's the reward of getting something to work and seeing it do what you planned. There's the reward for being paid a significant amount of money to do something that not everyone can do. There's the reward of someone finding your work useful, and perhaps occasionally telling you so.

These rewards come only with some persistence and sweat. If it were easy, as I tell my students, anyone could do it. But we are the experts. The public trusts us, as professional engineers or the near-equivalent, to know what we are doing. So we must do the job properly or we betray that trust.

Doing the job properly means that we are *professionals*. A professional keeps on top of the technology by reading and learning, all their life. And in the fast-changing world of electronics, where the technology is totally different by the end of the week, there is no other way to survive for long.

A professional does careful design work and keeps complete documentation, whether or not the client demands it. That's just the way this job is done. And, not every pro does this, but it's nice to give back to your profession by sharing what you've learned with others. Presumably, we don't need to protect our livelihood by being a secretive guild.

Who can say why designing and building electronic gadgets is such fun? But it *is* fun (most of the time, anyway), and it can lead to an interesting life.

Okay, enough sermonizing. Go design something. I hope these notes help you get it to work.

References

- [1] Electronic Prototype Construction Techniques
Peter D. Hiscocks, 2001
<http://www.ee.ryerson.ca/~phiscock/papers/construction.ps>
- [2] The Art of Electronics, 2nd ed.
P.Horowitz and W.Hill
Cambridge University Press, 1989
- [3] High-Speed Digital Design: A Handbook of Black Magic
Howard Johnson, Martin Graham
Prentice Hall, 1993
- [4] Design Considerations for Logic Products
Application Book SDYA002
Texas Instruments 1997
- [5] Troubleshooting Analog Circuits
Robert A. Pease
Butterworth-Heinemann, 1993
- [6] Ask The Applications Engineer – 12
Grounding (Again)
Analog Dialog 26-2, 1992
pp26-27
- [7] Maintaining Signal Integrity Enhances
ADC Circuit Performance
Nicholas Gray
Electronic Design, May 1, 2000
(also www.elecdesign.com)
- [8] Pay Attention to the Clock and Output Bus
to Improve High-Speed ADC Designs
Nicholas Gray
Electronic Design, June 26, 2000
(also www.elecdesign.com)
- [9] Multiple ADC Grounding
Howard Johnson
EDN, February 1, 2001
- [10] Designing with High-Speed Analog-to-Digital Converters
Mark Sauerwald
National Semiconductor Application Note AD-01
May, 1988
(www.national.com)

- [11] The ARRL Handbook
American Radio Relay League
Published annually, always contains one chapter on construction techniques. Reasonable price, a must-have.

- [12] High Speed Amplifier Techniques
A Designer's Companion for Wideband Circuitry
Jim Williams
Linear Technology Application Note 47
1993 Linear Applications Handbook, Volume II, pp 47-1 to 47-132
Linear Technology Corporation
An invaluable reference on high frequency amplifier wiring techniques, theory and practice.

- [13] Layout and probing techniques ensure low-noise performance
Jim Williams
EDN Magazine
(issue unknown)
Construction and probing techniques for switching power supplies.

- [14] Handcrafting Design Ideas
Jeff Bachiochi
Issue #70, May 1996, Circuit Cellar INK, pp 62-67
Covers much the same ground as this paper.

- [15] A Surface-Mount Technology Primer – Part 2
Bryan P. Bergeron
QST Magazine, January 1991, pp 27-30
The use and soldering of surface mount passive components.

- [16] Prototyping with smds (Surface Mount Devices)
Nick Wheeler
Electronics World, January 1998, pp67-69
Prototyping with fine-pitch surface mount integrated circuits.

- [17] Analog Breadboarding
James M. Bryant
Chapter 9, The Art and Science of Analog Circuit Design
Jim Williams, Editor
Butterworth Heinemann, 1995

[18] Electrosonic Catalogue 991
Electrosonic Supply
1100 Gordon Baker Road, Toronto, M2H 3B3
416-494-1555