Senior Design Tips

I wrote the following shortly after completing CE 123B to identify the key problems in our project and formulate what I would tell another student beginning CE 123A; things that I may have not written at the beginning of CE 123A. My aim was to illustrate it with concrete examples that could be encountered in the 123A/B format.

Management: A well managed average group is better than a poorly managed smart group.

1) You should always leave time for unexpected delays, especially if you not exactly sure how something works. For instance, my group experienced some delays getting our microcontroller’s boot loader to work, but I anticipated this because I was not completely familiar with the process.

2) Having two people on a given part also increases the level of honesty about its progress. Everybody thought the motor control part of my project was under control, but the one person working on it was not coming completely clean about how it was not very far along. Consequently, other people started loading additional work on him. This lead to a downward spiral on his part.

   c. Every part of the project should have at least two people who understand it. Therefore, one person can catch the other person’s silly mistakes. This is especially important for PCB design because a faulty PCB design can cost money (and time) to correct. Software, on the other hand, just needs to be recompiled. My group had one person working on a motor control unit. He made several silly mistakes in his PCBs because he didn’t have anybody else to give him a sanity check.

4) If you have a sponsor, then get a signed written agreement on the requirements for the project and the terms for reimbursement early on. Get it done during 123A; you’ll be too busy during 123B.

5) Also, get a formal specification of what components are supposed to do exactly by the beginning of 123B. When writing software for my component, I would crank out a lot of code and be on my way to be done. However, I would then be stuck because I wasn’t exactly sure of the functionality our sponsor wanted.

6) Meet regularly with your sponsor: be wary of sponsors who want to leave the project alone for 3 months and expect to get exactly what they want at the end.

7) You should include a provision in your team charter to impeach your group leader. You need a mixture of democracy and a strong leader.

8) It’s very important to have a strong leader that understands all the different parts and can allocate resources efficiently. It might be hard to pick this person in 123A if you don’t know the people in the group before hand. Your guess at who will do good work and who will do bad work will likely be much different in 123A then at the end of 123B if you don’t already know your teammates. However, you should still pick a leader because having the leader position in limbo can be the worst thing of all.
i. Many people will need a leader than can assign parts of the project in homework assignment size chunks, otherwise they will be lost.

10) Don’t be afraid to ask someone with more experience in a given part to do it for you. In my project, the people who had experience with PCB layout off loaded the responsibility onto people who didn’t have layout experience; I think our project suffered as a result.

11) Don't make up work just because you have nothing else to do at the moment. You will be short on time at the end of 123B. One of our group members designed the button debouncers too fancy at the beginning since we had nothing else to do. He should have kept them simple and studied the other parts of the project.

Balance your expertise: Don't assign someone with expertise in A to spend most of their time on B.

12) Cut the fat out of the group in 123A. My project had four EEs and three CEs. We could have gotten by with one of our skilled EEs. If your project involves wireless, you might want more EEs.

13) You must have plenty people in the group who have taken CE 121 and CE 173/174. These classes can be very important in your senior project.

14) Since there aren’t any CS people in 123A/B, it is good to make sure you have at least one competent programmer.

Having the needed parts:

15) Never put a part in a final schematic until it has been ordered. My group put a power supply in a schematic that turned out to be unavailable. Building a custom circuit to emulate the behavior of the required power supply severely delayed our project. If we had known early on that the required power supply for our microcontroller was unavailable, we could have picked a different microcontroller. Even things like sockets should be ordered before they go on a schematic. You never know if the desired size is unavailable.

P. It is even better if you have the part and tested when you put it in the schematic.

17) Pay the extra money for full development kits if it will allow you to develop faster. Our group didn’t want to pay for a €99 full development board that pinned out all the microcontroller’s pins and was really good for prototyping ADC stuff. Therefore, we went with a $40 development kit, which limited our abilities to develop. I had to wait over 3 weeks after writing a driver for an LCD before testing on our actual board because we had the inferior development kit.

18) Always order double the amount of parts you actually need unless they are really expensive. It always good to have spares on hand in case you accidentally burn out some parts or it turns out you need more. If the parts become unavailable or have a long shipping delay it could delay your project. Also, if it is a cheap part then paying for extra is cheaper than paying for fast shipping when you need more. If you’re working on a high power application, it might be a good idea to
order even more spares. Our group had people designing an H-Bridge and they burned out many parts due to the high power nature of their component.

19) Don’t re-invent the wheel. Thoroughly look for components that do the job first. If you find them then buy it, unless it is too expensive, and then move on to bigger and better things. Otherwise, you can try to build it yourself. Commenting further on this subject would be too painful for me given that it is a sore subject in my group.

20) Likewise, always research the prior art before embarking on your project.

Productivity boosters:

21) JTAG interfaces on your processor are always a good thing. They can, at times, make debugging software problems that appear to be complex actually trivial. It might even be good to put a JTAG interface on your final board, not just your development board.

22) Don’t work longer hours and loose sleep as a result. You will actually complete less work because you’ll make stupid mistakes as a result of being tired. Some people think they can work effectively without much sleep. Studies have shown this is not true.

23) Keeping snacks and caffeine in the lab will increase the group’s productivity.

24) Use integrated parts as much as possible. In our group, we had RC filters and Schmidt triggers for button debouncers (total over-kill) that we implemented with op-amps, resistors, and capacitors. When it came time to assemble the board this required making approximately 150 solder joints. This was a complete waste of time. It would have been much better to go with integrated Schmidt triggers. Of course, in this example we could have just gotten rid of the Schmidt triggers. These were an artifact of having too many EE's in our group.

25) Keep the lab reasonably clean. Although cleaning the lab takes time, it also takes time when your looking for stuff you can’t find because there is too much clutter scattered around the lab. It is a trade off decision to take time to clean the lab. It is best to have a team leader that is capable of making this decision.

26) It is good to follow the guidelines with regard to laboratory notebooks. Keep them handy for reference. Keep pin outs and other stuff taped in your notebook for quick reference. If you don’t have you notebook next to you then you probably won’t write in it either. Even more important though is to have an organized binder for data sheets. Having to go on a treasure hunt in the lab for a data sheet can be really annoying.

Avoiding those silly mistakes: Once the magic smoke comes out of the chip, it is hard to get it back in!

27) Be careful with high voltage and reversing polarity. If you aren’t then you will likely hear a spark and then smell smoke. Always put a diode on your power supply to protect your circuit in case someone accidentally connects the power supply backwards.

   bb. Always make the orientation of the chip on the PCB. People often place the chip in at a 90 degree angle.
29) When testing parts like power supplies remember to test it with a load and not just its open circuit output. See how much current it can source. We didn’t test ours and realized that it couldn’t source all the current it was supposed to when we put LEDs on our board.

30) Be careful of things like putting TX and RX on backwards. Some people seem to screw this up more than 50% of the time, which is worse then what you’d expect from a monkey. Do loop-back tests on UARTs and such before connecting them to anything when stuffing a board.

31) Use color-coded wires; it easy to get things mixed up otherwise.

32) Watch out for surface mount parts. They are a pain to solder on and an even bigger pain to take off. However, surface mount is very common on anything but ancient parts.

33) Non-idealities should be noted with parts. My group built a power supply that required a certain size inductor. We simply put one in of the right size, but our power supply couldn’t put out the specified power. Turned out the inductor wasn’t rated for the current that the regulator was putting through it. Remember that \( v = L\frac{di}{dt} \) is only true for a certain range of \( i \) in real inductors. Petersen discovered this problem in our circuit.

34) You should reserve the final days of the quarter for practicing your final presentation. If you work on finishing your project at the very end, you’ll be in a complete rush and might do something stupid. Consequently, you could break part of your project at the worst possible time. Our group was in a rush to finish things the day of the presentation. We didn’t get it done and our presentation could have been better.

35) Bring up boards sequentially. First, bring up power, then any other parts necessary for the ICs, and so on. Test the board after each step. The worst thing to do is to build the entire board and then test. When it doesn’t work, you don’t know where to look for problems. Too many people learn this the hard way. In fact, it is even a good idea to check your boards for shorts and other defects before you even put any parts on to see if there was a manufacturing defect.

36) Practice your presentations; you can tell when people don’t. Also, the zipper on my pants always goes down during presentations. I don’t know if other people have this problem too.