Adrian’s Assignment Advice
MTRX3700 Major Assignment
Disclaimer

• This is not a complete list of everything that should and should not be done for the major project
• There are many ways to design a project. The idea here is to list traps, pitfalls and help you to avoid common mistakes that will be made by your peers 😊
• The main objective of this lecture is to prompt you to think about how the project will be designed and what you are actually doing
• I may jump around a lot – I tried to write down everything that I could think of and not forget anything
Group Management I

• Make sure that everyone knows everyone else and that everyone can contact each other somehow
• A communication medium that everyone can and WILL use will provide the most benefit for the group
  – Google hangouts, forums, etc
• Everyone has different strengths and weaknesses and these should be used to help determine who works on what
  – A person with the strongest background in an area will definitely work best for the group in that particular role, however it is your responsibility that you personally learn something to strengthen your weaknesses as an engineer.
  – Perhaps try budying people up in pairs as an example of work allocation
Group Management II

• The split of roles doesn't have to match the split based on modules (to come) or the assignment sheet
  – However optimality dictates that everyone *always* has something to work on. For instance assigning someone to integration will mean that they do nothing until the modules are ready to be integrated, and this may mean that there is not enough time to bring everything together properly

• Heavily document everything that you do, as you go about it
  – This includes writing documents about what you plan to do, how you plan to do it, comments and headers in code etc. Look at things like Doxygen.
  – Everyone is required to help with the writing of the report at the end, and writing as you go will make this significantly easier
General Design

• The final product that you will produce will be too complicated to tackle head on
  – Break it down into modules

• What are the requirements for each module
  – What hardware and/or software is required for each module
  – Each module should specify required inputs and outputs for both the hardware and software components
  – Each module should be as separate as possible from all other modules (both hardware and software)
  – Separate what needs to be done from the bells and whistles
    • Meet requirements before working on the extensions as functionality is key

• Should design for ease of (end) use
Software Management

• How is the software going to be managed
  – What type of backup systems will be implemented such as a SVN, cloud system etc
  – Need to make sure that it is possible to roll back the software for when mistakes are made
  – Really useful for overtired students near project deadlines

• Good module design will allow you to know the inputs and outputs of every module.
  – Generate module stubs that provide these inputs/outputs
  – This will allow for several modules to be developed at once without having to wait for someone else (parallel work at its finest)
Software Design

• Make module stubs
  – These are files with empty functions that take the correct inputs and output correct values
  – Also contain comments on what the code is supposed to do (What it does do too!) and what are acceptable ranges of inputs and outputs

• Determine a priority listing for which modules you complete first.
  – What modules are needed for others to work?
  – What modules can help you debug other modules if implemented first?
  – Get core modules fully functional and debugged before working on extras
Hardware Management

• Twisting Wires
  – Use a drill
  – Heatshrink / Cable Ties

• Crimping Headers
  – What will happen if you build/assemble it the wrong way
  – Block a header socket to restrict orientation when assembling
  – Mark pin 1 with (e.g.) liquid paper dot

• Wires – DIFFERENT COLOURS

• Additional Hardware Components
  – RTFM
  – What are the power / timing / other requirements of the module
Hardware Design

• For each module, determine the number of pins that are required

• Develop a pin map so that you can see where and how everything fits together
  – It helps to draw this, to show any errors such as multiple pin assignments

• Leave room for additions / extensions
  – Modular designs make this easy
  – Being strict on pin usage makes it easier to add bells and whistles later
Building on Stripboard (Vero)

• Plan the layout before building – 0.1 inch module
  – Adapt shape to suit space available
  – Minimise size of stripboard but provide proper space for components – e.g. resistor needs 0.4 inch between holes

• Put connectors towards the outside of boards

• Are you going to stack boards?

• Remember that a board has to be mounted. M3 screw needs 3.5mm hole, 7mm diameter clearance

• We will provide stripboard but only of a size to suit your design

• Discrete wires soldered to stripboard need strain relief
Supplied Components

• We have got reasonable stocks of
  – Hook up wire
  – Resistors and potentiometers
  – Bypass caps
  – LEDs
  – Stripboard and stripboard cutters
  – 74HCxxx ICs
  – Heatshrink and heatshrink gun
  – Etc, etc...

• See http://web.aeromech.usyd.edu.au/MTRXLAB
• Request by email to David before lab day
Power

• Make a power budget
  – What components are being used
  – How often are modules being activated (% duty cycle)
  – What assumptions are you making for determining this

• How is the power going to be distributed.

• Are there hardware components that need a large amount of power infrequently – how is this power going to be supplied and how does it affect the rest of the system.
Bonus Marks, Extensions and more Functionality

- It is extremely important that the extensions that you would like to incorporate are thought of at the start of the project during the design phase.
- Do extensions require integration with the rest of the project or are they something that can be worked on at a later date (time permitting etc)?
  - Figure out what extensions you realistically want to incorporate into the project so that you can plan accordingly.
- Make sure that the extensions actually improve the functionality or usability of the project to some extent.
- Extensions can be hardware, software or a combination of both.
Testing

• Hardware testing is very useful for quickly seeing whether modules are working and also for error catching
  – Keep your hardware test routines up to date
  – LEDs are particularly useful for showing test output.

• Remember that there are pins that are being used for the ICD puck.
  – If possible try to leave these pins free until necessary
  – Remember that this is the case for when you are debugging and wondering why things don't work
  – If these pins are used, ensure that there is a method to test the hardware using the pins that doesn't require the ICD
Testing - Integration

- Every module should be (unit) tested before it is integrated with everything else
- Test the modules with each new additional integrated module
  - Test A with B, then A with C rather than A,B,C all at once
    - This makes it possible to isolate errors when they arise
  - Although this may take more time initially, this is particularly useful at avoiding mistakes that may cost significantly more time
Fitting Everything Together I

- Where and how are the various components going to be positioned inside the enclosure?
  - Which components need to be near each other and which need to be far away from each other
  - Put components on box lids! Maximise the use of all interior surfaces. This is particularly useful for LCD screens
  - Design the wiring harness and make (and test) it once
  - Mount electronics modules properly (bolts, standoffs)

- Make sure that the enclosure you use is large enough to comfortably fit everything in
Fitting Everything Together II

- Ensure that wires are long enough to disassemble the enclosure while leaving everything attached
  - Make sure that you can work on (probe etc) the hardware once it is in the enclosure
- How are components fitting together on the PCB
  - Is it possible to stack PCB layers with spacers rather than having a single large PCB sheet (harder to probe though)
- Where will replaceable components go and how can they be reached when they need to be replaced
  - Batteries are a good example
  - These items should be easy to access and swap
- Some connectors may be needed on the wall of the enclosure