**490B Final Report**

**2 Submissions:**

* Team Report – One per team.
* Individual Report – 1 Page, Not Double Spaced, Detailing your contributions. Please mention if you worked with other team members on a specific aspect. Also include your perception of the contributions by other team members if you feel that the workload is not being distributed evenly or if members have been absent. This individual report is 100% private and will never be shared.

**Format Guidelines:** Not double-spaced, 12pt Calibri Font or similar. Should be mostly written (approx.. 60% or more), diagrams and images are necessary but should not comprise the majority of the report. Should be long enough to demonstrate a mastery of the topic and technologies used to implement your design. Length of the report is ultimately up to the team. Grading will be on completeness, thoroughness, and the overall attention given to the details.

**Team Name**

A name should be chosen for the team. It should be descriptive of the project and should be appropriate for all audiences. This team name must be included for all submissions, assignments, progress reports, sprints, etc… by you and your team.

**Team Members and Biography**

On one to two pages, include a picture of each team member and a short one paragraph biography for each. Biography should be professional and something that a prospective employer would be interested in (i.e. Engineering Interests and/or hobbies and other interests that make you unique).

**Project Overview**

Provide a two or more paragraph, non-technical, description of your project. This must be a clear, uncomplicated description that any lay person could understand. Include any diagrams or drawings to help the reader understand the project. Who is the intended end-user of this product? (Think about this section as how you would explain your project to a non-technical family member). This should take approximately 1 to 2 pages.

**Customer Needs**

The most important aspect in the early design process of an engineering design project is identifying the customer needs. From those needs we will then extract and define a precise and testable set of design specifications. The goal here is that if we design a product that can achieve the same or better in all specifications then the customer will have the best product possible and we will have completed our project to a satisfactory level that was agreed upon. Reference your notes from our lecture and activity we did regarding the customer needs we all did together during class.

Customer Needs can fall into 3 categories (Car Example)

* Minimum Needs - Standard Features that are Necessary: Wheels, Seating, Dashboard, etc...
* Assumed Needs - Basic Operation, Safety, Reliability: Ability to Drive
* Unrecognized Needs - Things the customer didn't know he needed: Compression Ratio of Engine, Voltage Levels

**Design Specifications**

 A design specification is a list of testable metrics that are used to validate the customer needs. This is a one-to-many relationship of the customer needs. One customer need usually results in many specifications. Example, a car needs to brake, but how fast must it be able to break and under what conditions. Try to think in terms of a specification as a testable metric (i.e. a number) that we can test and validate. If our product manages to satisfy each one of these specifications, then the customer has no choice but to accept delivery. If for some reason the customer doesn’t accept delivery and would have liked something to be different, then there was a failure in the customer needs and ultimately the design specifications that were extracted.

**Constraints**

Project constraints are limitations or restrictions that define the boundaries within which a project must operate. These constraints can impact the project’s scope, schedule, budget, quality, resources, and overall deliverables. The most common project constraints are:Time, Cost, Scope, Quality, Customer, Resources. But, there can be many more depending on the specifics of the project. Especially given the scale and resources we have as a senior design project being done by students during a fixed 2 semester time window, we may need to modify constraints as we progress in order to ensure successful completion of our project. The biggest tragedy we can have in this class is an unsuccessful project that failed to live up to a standard. We may have to adjust constraints and expectations if we encounter insurmountable hurdles. I will try to help each of the teams to detect these problem points early on so that we don’t hit a “wall” too late in our development. We want to make sure to get answers to the difficult questions as early as possible so that we have time to recover.

**Project Implementation**

 The same as the project overview, but now go into much more detail about your implantation of the solution using the technical “jargon” that accurately conveys how the entire project is to be implemented. All aspects of your project, the implementation, the problems your project are solving and the solutions should be detailed in an overview with appropriate technical engineering language.

**Subcomponent Descriptions**

Every project is a large complex device that is created from sub-components and/or sub-modules. All of these sub-components should be broken down into it’s own section that has a description of the subcomponent, the functionality, how it’s implemented, the theory of operation, and it’s applicability to your project. The goal here is to prove that you have a complete understanding and mastery of the fundamentals of these components. Appropriate and well documented proper engineering solutions should be used where applicable, i.e. if a closed loop control system is needed to solve a problem within your project, then I would expect a demonstration of a thorough understanding of PID controllers or similar and how it was applied. Keep in mind that it is not at all common for a project to consist of entirely novel devices and solutions, most of what we use and work with has a rich history of development and exists in a textbook or peer reviewed source like IEEE or ACM. If a communication scheme was used then I would expect a thorough explanation of an appropriate RS-232, I2C or SPI standard, etc. This would also be a good place where snapshots of the schematic or other data could be used to help explain the functionality. Any specialized tests relevant to using a sub-component in your specific application could also be placed in these sections where applicable. Keep in mind that this is one of the more important sections of the report from the perspective of where I look to see how much attention to detail and understanding of the project there was.

**Overall System Functional Block Diagram or Diagrams**

Many of our projects are being made up of other commercial (off-the-shelf) products or modules. These may be represented as boxes/blocks with all inputs/outputs/power connections clearly labeled. Every aspect of your project should exist in one of these modules and should have all inputs/outputs clearly labeled including power connections (in other words, if there is a wire involved it should be included).

A table must be written for each block/module after the Diagram and must include the functionality of every input or output, Signal Name, Signal Description, Signal Type (Analog/Digital/Power), Voltage Level, Current Requirements (if it is a power connection) and Operation. Refer to the class notes for how I specifically would like to see a block diagram implemented. On top of the usefulness for you and your team in having this block diagram for collaboration and verification, it is incredibly valuable for me to have this block diagram with connection tables as it allows me to double check consistency of voltage levels and signal types between sub-components. Feeding 5v into a 3.3v Input is a common mistake that may cause damage, this block diagram and signal tables are a good way to force us into verifying that we are not doing something like this.

**Complete Schematic**

* Designed in any proper schematic entry/capture program, i.e. Circuit Maker/ Altium, or Orcad. Kicad is ok too =), but my own personal preference is Altium (probably because I’ve used it for 20 years)
* A Detailed schematic of the above mentioned modules. If the module is an “off-the-shelf component” then a schematic does not need to be included if it doesn’t exist. However, every project is using a Microcontroller or FPGA. This “core” module of your project must have a schematic created by you. In other words, if you are using a Arduino/ARM/Xilinx development board, then **you must recreate this schematic**. We are attempting to design in a direction where this is a stand-alone device without reliance from development boards to do everything for us. Though I understand that the end project will probably still have these development boards.
* All schematics will include a BOM (Bill of Materials) that includes every part, part number, description, package used, manufacturer, distributer, cost, and quantity.

**Power Management**

All power requirements should be accounted for and shown in a schematic as well as an included power budget. If a battery is used, then an appropriate discharge curve should be included along with a description of the power components used, approximate efficiency and a demonstration of maximizing your utilization of the battery capacity.

**WBS (Work Break-Down Structure)**

A WBS should be included that provides a deliverable focused decomposition of your project/product and should always be updated to reflect the final design.

**Gantt Chart**

A Gantt Chart should also be created which includes each main task, all the subtasks and at a minimum be broken down into a week by week chart which includes all the tasks and the engineers in the team that have been assigned to the task. Use the Work Break-Down Structure in conjunction with the 5 Demo Contract to help guide the timeline and sequence of the tasks that must be completed.

**5 Demos Contract (“Agile Sprints”)**

List and describe 5 functional “milestones” of your project. In other words, what building blocks are required for the entire project to work. These “building blocks” should be demonstrable, prototyped, and testable. These will be the 5 demos throughout the 490B semester that lead up to the entire working project. Each demo may be an independent aspect of the project or build off the previous one. Each demo should have a measurable outcome that is stated, in other words, what will the instructor see that determines success and what is the acceptable error in the measured output. All three, the WBS, Gantt Chart, 5 Demo Contract must be included in the final report.

**Similar Products**

Research other similar products and try to find commercial products that are similar to your proposed project. Describe the commercial products and how yours is different. Maybe your project has different abilities or uses different technology for implementation, cost, etc…

**Societal, Environmental Impact and Sustainability**

How does this product benefit society and the environment? Try to consider the larger impact and importance of your project. Consider low power devices and how your project can make efficient use of materials, batteries, and resources. Consider the sustainability of the product from many perspectives. Can the theoretical organization producing the product maintain the product line? Sustain employees growth and well-being? Serve the community? This is your chance to tout the positive implications of your project and is important. We don’t just want to be creating a fad that ends up in the garbage but instead want to benefit people and society, locally and globally.

**Ethical Considerations**

This should be a conversation about the ethics and considerations behind your design choices in the project. Some considerations are but not limited to:

Safety: Ensuring the well-being of people and the environment by designing systems and products that minimize harm and risk.

Sustainability: Balancing economic, social, and environmental factors to create long-term solutions that do not compromise future generations’ quality of life.

User Autonomy: Respecting individuals’ rights and freedoms by designing systems and products that promote user control and agency.

Privacy: Protecting individuals’ personal information and data by designing systems and products that minimize data collection and ensure secure storage and transmission.

**Circuit Simulations, Verification & Modeling**

* All projects will include many aspects of verification, software or hardware verification.
* Simulated in a proper circuit modeling program, i.e. LTSpice or equivalent
* Although not required in every project, any circuit that the instructor sees is subject to inquiry as to the functionality and verification of this component. An LTSpice model of different aspects of our projects, give us the confidence that we aren’t wasting our time building something that doesn’t have a chance at working. A waveform and appropriate notes/annotations and description should demonstrate the functionality.

**Software Flowcharts**

A single flowchart or many smaller ones may be used to describe the functionality of the software aspect(s) of your project. It is your responsibility as the engineering team to decide what type of flowchart would best fit your needs. Dataflow Diagrams or Unified Modeling Language Diagrams are two appropriate and common types to be seen. Lucidchart, Visio and SmartDraw are common tools that can aid in this.

**Software Verification & Validation**

 Similar to a hardware verification, there should also be software components that are singled out and given extra scrutiny with verification. For example, if I was working on a Micromouse Maze Solving Autonomous robot, it would be common to isolate the maze solving algorithm and give this extra attention of verifying. Either through comparing different algorithms like Depth-First-Search or Bellman Flood etc… Or if I had some other type of process being controlled by a PID closed loop controller, then I might isolate the controller for extra tuning and create response curves to compare performance. The needs are different for every team and I will be working closely to help you identify what would be most useful for your specific application during progress reports or demos/sprints.

**Software Revisioning**

 Revisioning systems like subversion, git, and mercurial are invaluable and practically necessary these days for maintaining code while having multiple engineers collaborate. Demonstration of using Software Revisioning is necessary.

**A Note On Project/Report Changes**

It is quite common for the project to change and evolve as we are working on it. When aspects change that had been discussed in a previous version of the report, do not simply remove it from the report. You put effort and work into a potential solution that maybe didn’t make it. This is normal and I want to hear about it. Please feel free to take things that have changed as your report has evolved and move them to an appendix, then you can make reference to this appendix as you progress. This report is a living breathing document that gets modified, revised and updated as your project progresses.

**References**

Approximately 15 or more valuable sources should be used and cited in the bibliography. Although the internet, google and wikipedia is a convenient source, these should be limited. 3 or more sources must come from peer reviewed journals and periodicals. An attempt should be made to incorporate some aspect from these journals into your hardware or software, either an algorithm or hardware scheme.

**The report should not be limited to these sections nor should it be a replica of these suggestions. You should take an iterative approach to drafting this report and create an organization that best represents and suits your project.**

**Grading Guidelines**

I will always be using the previous report to compare to the newer one looking for improvement and changes.

**A -** All items are included and well defined. Multiple hardware and software verifications have been used and documented adequately. The entire project looks promising and has a good chance of success. There are no obvious unknowns in the design and terminology is used appropriate. Sufficient depth has been achieved, demonstrating a complete understanding of all aspects of the project.

**B -** All items are included and at least one major hardware and one major software verification has been undertaken successfully and demonstrated adequately.

**C -** Most items are included and some have not been sufficiently explained or expanded on.

**D-F -** Some items are included, verification has not been done or is lacking. Obvious questions, holes/unknowns still persist in the design.