MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Engineering 100-950

Electronics for Atmospheric and Space Measurements

Honors Capstone Final Report

Prepared by Cayetano Wagner, May 3rd, 2021

Table of Contents

1. Overview

Engineering 100 is an introductory engineering course required to complete a Bachelor of Science and Engineering degree at the University of Michigan. It is intended to be taken during students' first year at Michigan and aims to introduce students to the engineering design process, communication using written technical reports and oral presentations, and teamwork and team management. Roughly a dozen sections are offered each year, with each focusing on a specific design problem in various engineering fields. Engineering 100-950 was first introduced in Fall 2016 and leads students through the development of a sensor package to be used in high altitude weather balloons. Section 950 is among the most popular Engineering 100 sections and well received by a majority of its students. Developing and constructing a functioning sensor package requires students to work with upper-level electrical, computer, and climate engineering concepts and gives students valuable technical experience that can be applied in later coursework, project teams, or internships. Analyzing and communicating the impact of laboratory tests and real-world data prepares students for technical writing that they will be responsible for producing over the course of their engineering careers. And the opportunity to launch a high altitude weather balloon is an exciting and unique experience that can help define their first year in Michigan Engineering.

Although the class is well functioning and well received in its current state , it has been criticised for its perceived overemphasis on technical materials and lack of student oriented design work. In order to address this concern, we plan on shifting lab emphasis from field-specific technical experience to generalizable design experience. This will require increased integration between the engineering and technical communication aspects of the course. There are also unrelated concerns about the institutional knowledge held by current and previous instructors. Since its creation in 2016, there has been a continuous line of

Instructional Assistants that have been able to easily pass institutional knowledge down to maintain the course. Covid-19 has disrupted this flow of information and an additional aspect of my project will be to provide thorough documentation and instruction to future IAs to help the class succeed in future semesters.

2. Revision Process

The majority of design work that needed to be done for this project involved reassessing our original intentions with specific elements of the course and making sure they were meeting our needs. This also required us to integrate feedback and observations from past semesters. A framework that was used to help guide design was the use of *element, intent, side effect, alternative*. Listing elements and their intended goals for the student experience. Using feedback and observations to understand side effects that those elements have on the students. And, if there is a misalignment between intent and impact, developing alternative approaches that will produce a better experience for students. Listed below are three examples that show the framework at work.

Element	Skills labs
Intent	Teach foundational concepts Build interesting and confidence in engineering skills
Side effect	Overemphasis on EECS can alienate students with other interests Material isn't absorbed due to rapid pace – labs are instead being followed as tutorials
Alternative	Lean in to "tutorial" nature of complex circuitry and code Replace most advanced material with content centered on physical design

Element	Postlab reports
Intent	Reiterate important concepts from lab Build report writing skills
Side effect	Report quality is poor early in the semester because those skills aren't taught til later Students are frustrated with report length and quality expectations
Alternative	Replace lab reports with quizzes when concepts are important to lock in Require graphics and data to be used in major assignments to be postlab deliverables

Element	Build labs
Intent	Ensure student designs are flight ready Let students have fun with designs to make launch memorable
Side effect	Unstructured nature leaves students confused and without adjacency Majority of teams copy the “stock” design for simplicity
Alternative	Introduce physical design concepts earlier Formalize physical design with technical requirements Explicitly encourage fun designs with budget for orders

This analysis allowed us to find flaws in our previous approaches and attempt to improve the class by restructuring elements to better fit student needs.

3. Skills Labs

The most critical changes that resulted from this project are the changes to the lab plan. Labs that were previously ordered

1. Introduction to Arduino
2. Temperature Sensing
3. Humidity, Pressure, Acceleration
4. Power
5. GPS
6. Altium I
7. Altium II
8. Spring Break
9. Build/Test
10. Build/Test
11. Build/Test

have been altered to the following sequence.

1. Introduction to Arduino
2. Digital Sensors and Calibration
3. Openlog Guide / Analog Sensors
4. GPS Guide / Shock Experimentation
5. Power Guide / Thermal Modelling
6. Altium I
7. Altium II
8. Spring Break
9. Build/Test
10. Build/Test
11. Build/Test

Although this is largely a reformulation of previous labs, it does require additional content to be created. The two largest components being the shock experimentation and thermal modelling labs. These labs will be new to the course and attempt to give students necessary skills to create more detailed and sophisticated physical designs. It will also give them an improved ability to make substantial engineering decisions during their design process that will later test and fly.

Another large change is the shift to tutorials or guides for the more advanced electrical and programming components of the project. This was done in an attempt to allow students to focus of their design and free up space away from advanced technical content that they will encounter, with a better theoretical basis, later on in their college careers.

4. Postlab Reports

One of the most frequent pieces of feedback that instructors received in the past was that postlab reports were unnecessarily long, complex, and difficult. While there are a few important technical components that students do need to fully understand to be able to succeed in the course, the majority of technical content in taught for a specific component of a specific lab and can otherwise be disregarded. While we still plan on teaching this technical content in order to engage student curiosity and give them an understanding of coursework in the College of Engineering, testing their comprehension of the material is less important as it is not necessary to succeed in the course. To account for this, we have decided to shift postlab comprehension testing to a short quiz instead of a report. While it still tests basic understanding of the material, it does so in a manner that is much less time consuming for students to write and for instructors to grade. Below is an example of a question that is effective in testing understanding, but can be answered in less than 2 minutes and graded automatically.

5. Build Labs

Part of the changes being made to the skills labs are also targeting the build labs. Specifically the introduction of shock and thermal modelling. These concepts will allow students to have more self-guidance over their design and their introduction early in the semester will allow designs to be developed earlier in the semester. This will reduce the rushed nature of the physical designs and give students more time to prepare well-developed concepts.

Question 4

2 pts

Which of the following code blocks correctly compares cstrs and returns 1 if two strings are equivalent, or 0 if they are not.

- ○
```
int strcmp(char str1[], char str2[]){
    char* ptr1 = str1;
    char* ptr2 = str2;
    while(*ptr1 != '\0' && *ptr2 != '\0'){
        if(*ptr1 != *ptr2){
            return 0;
        }
        ptr1 += 1;
        ptr2 += 1;
    }

    if(*ptr1 == '\0' && *ptr2 == '\0'){
        return 1;
    }
    return 0;
}
```

- ○
```
int strcmp(char str1[], char str2[]){
    char* ptr1 = str1;
    char* ptr2 = str2;
    while(*ptr1 != '\0' && *ptr2 != '\0'){
        if(*ptr1 != *ptr2){
            return 0;
        }
        ptr1 += 1;
        ptr2 += 1;
    }

    return 1;
}
```

- ○
```
int strcmp(char str1[], char str2[]){
    if(str1.size() != str2.size()){
        return 0;
    }
    for(int i = 0; i < str1.size(); i++){
        if(str1[i] != str2[i]){
            return 0;
        }
    }
    return 1;
}
```