4 Design

4.1 Design Content

Briefly describe what is the design content in your project.

We are designing many new interactive questions that will test students' knowledge in more interesting ways. These questions must be creative and test students in new ways beyond simple multiple choice or fill in the blank.

We will also design an emulator to simulate the CyBot in 288. The emulator must accurately represent the real world so that students can more easily simulate their code without needing a physical bot and test space.

4.2 Design Complexity

Provide evidence that your project is of sufficient technical complexity. Use the following metric or argue for one of your own. Justify your statements (e.g., list the components/subsystems and describe the applicable scientific, mathematical, or engineering principles)

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles
2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

For this project, there are a number of components that work together. One component is the PrairieLearn framework and course content. We need to create content that tests students' understanding of the course material. Along with that, we need to create an autograder that runs in Docker containers. The autograder needs to be able to compile the C code given by students, and determine the output. This will require an understanding of C and Docker to create robust software following many engineering standards. Another piece is implementing an emulator for the microcontroller used in class, and connecting it to a simulated CyBot. These simulations need to accurately represent what would happen if the code was run on physical hardware. The simulation should also allow the students to see a representation of each lab and the CyBot itself. This will require the combination of computer, electrical, and software engineering knowledge.

4.3 Modern Engineering Tools

What modern engineering tools were used for this design? Their roles.

The modern engineering tools we used for this design was IEEE standards, Git/Gitlab, and Ubuntu. The IEEE Standards helped us decide what tools to use and how to utilize them according to our project. We utilize git for our project in order to track changes and to have version control for all additions that are made. We will make use of Gitlab to track progress and keep a backlog and activity board to measure tasks and progress. VirtualBox and Ubuntu are tools used to create a virtual environment in order to run PrairieLearn.

4.4 Design Context

Describe the broader context in which your design problem is situated. What communities are you designing for? What communities are affected by your design? What societal needs does your project address?

List relevant considerations related to your project in each of the following areas:

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project does not have any impact on the public health, safety, or welfare of our stakeholders	Our project does not have any impact on the public health, safety, or welfare of our stakeholders
Global, cultural, and social	The project can reflect the values of the faculty creating coursework. It will also improve the learning experience of students, promoting participation.	The implementation allows the users on the faculty level to create questions with a high level of customization, allowing them to express themselves in many ways.
Environmental	Our project does not have any impact on environmental factors.	Our project does not have any impact on environmental factors.
Economic	This project has the ability to create jobs and provide a product to make a profit. It also trains new engineers who will join the workforce in the near future.	The product provides an opportunity for the sale of a product in order to generate revenue for the owners of the project. The development of this project also offers job opportunities for developers.

4.5 Prior Work/Solutions

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available. Thus, provide a list of pros and cons of your target solution compared to all other related products/systems.

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

A lot of similar products are in place for the autograder, such as Renaissance, MasteryConnect, and Gradescope, just to name a few of the bigger ones. These groups have assessment problems which teachers/professors can create questions and set the answers for students to fill out. This would include homeworks or exams, with the option for manually graded questions for long response. These competitors, however, do not have any form of emulator built in for microcontrollers, something that can set PrairieLearn apart.

We are following previous work, and the advantages we will gather are having a few items to work off of and using previous documentation to help us get started. The previous group had good documentation on the basic autograder function and made videos guiding us through setup. The shortcomings are that there wasn't as much done in the further development from the group last year, so past the initial autograder fixes we are doing, they have nothing we can build off of there.

A few pros for our solution would be help for the CPR E 288 class and an emulator that will allow students to have better access to the course labs and testing, which was a bulk of the work and learning in the class. This emulator will allow students to work from home without having to have a microcontroller with them physically. Some sections of CPR E 288 had an emulator in the past. The previous emulators used were not very user-friendly and are in need of a replacement, something we believe PrairieLearn has the ability to improve upon. This also gives students different ways to learn with more interactive questions to really help with understanding of different concepts. Doing more problems in different ways can make students succeed more and more.

A few cons for our solution come more from the autograder side, and the competition having UI that people may prefer more. We also know that these other products are more user-friendly and well-respected than the PrairieLearn framework. We can also see that the autograder must be independent for each question due to how the framework is structured, something that limits the ease of implementation. Another con is user friendliness in general, getting access to prairie learn is a little difficult and might be a hassle for some students.

4.6 Design Decisions

List key design decisions (at least three) that you have made or will need to make in relation to your proposed solution. These can include, but are not limited to, materials, subsystems, physical components, sensors/chips/devices, physical layout, features, etc.

- We are going to use PrairieLearn as the framework for distributing course content
- We will use Docker containers to compile and run code for the autograder
- We will use Git and GitLab to keep track of versioning
- We will use a campus server for the final production server

4.7 Proposed Design

Discuss what you have done so far – what have you tried/implemented/tested?

We have all set up the PrairieLearning environment which we will be using to have questions and the autograder working. In this we have implemented a few different forms of questions including multiple choice and fill in the blank, which would then be autograded for the correct response. These questions have randomized numbers in order for students to get slightly different questions, but still keep the concept the same.

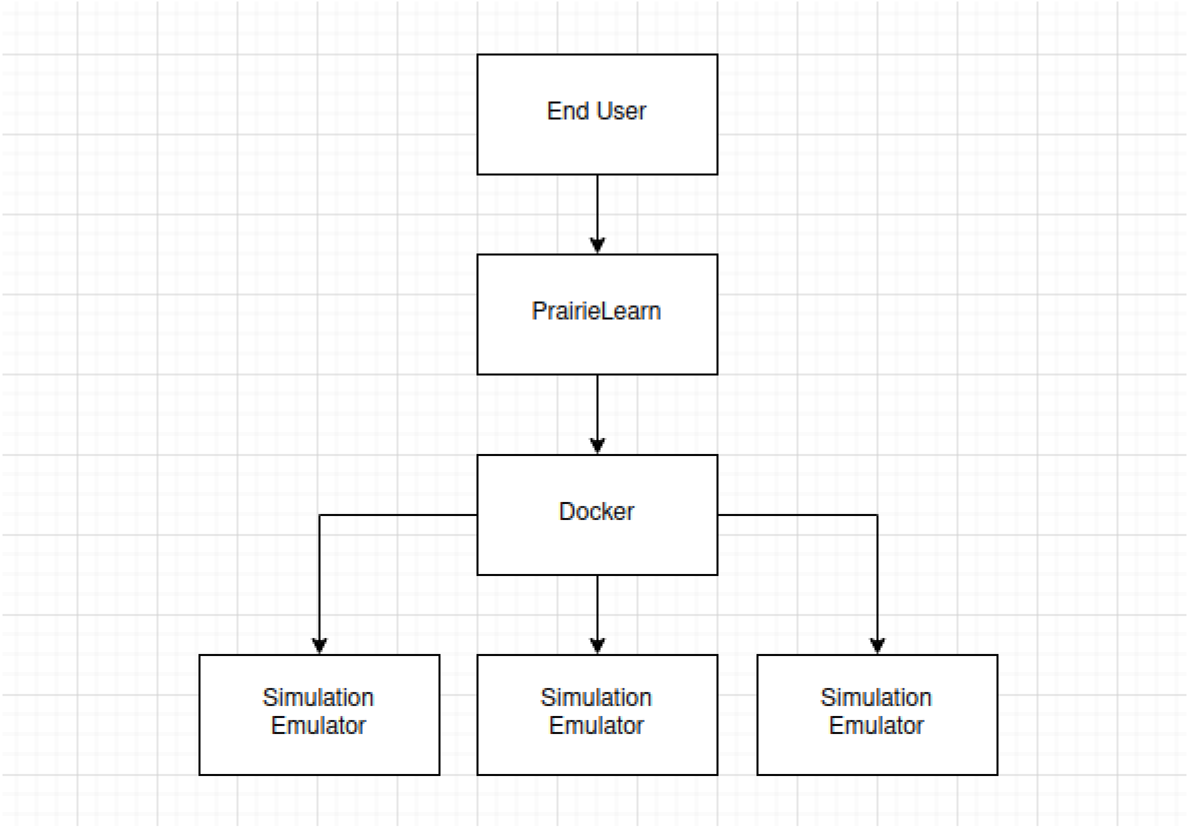
4.7.1 Design 0 (Initial Design)

Design Visual and Description

Include a visual depiction of your current design. Different visual types may be relevant to different types of projects. You may include: a block diagram of individual components or subsystems and their interconnections, a circuit diagram, a sketch of physical components and their operation, etc.

Describe your current design, referencing the visual. This design description should be in sufficient detail that another team of engineers can look through it and implement it.

Justify each component in the design with respect to requirements.

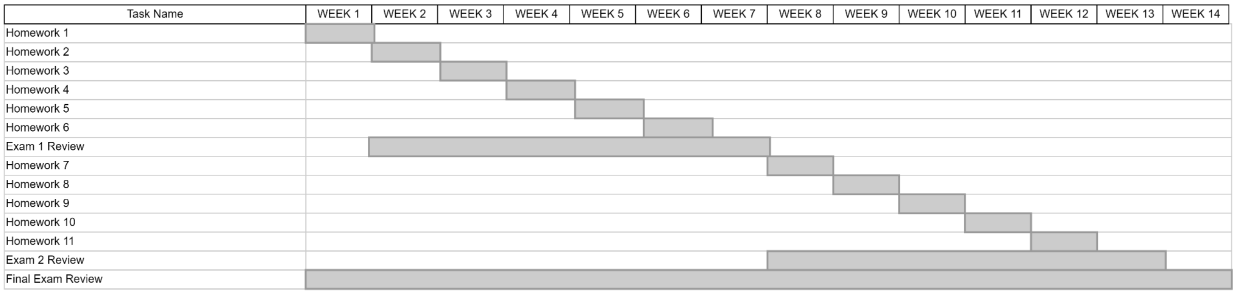


The current design uses PrairieLearn as the interface between the simulations and the end user. PrairieLearn uses Docker containers for each question and the code given by the user is put into the emulator / CyBot simulation. To implement this, we will utilize the existing PrairieLearn framework in order to create questions through the UI and built-in text editors using JavaScript and Python. These questions are then published to students through the framework's implementation. We will then source an emulator and make any necessary modifications in order to integrate it onto the PrairieLean site. This emulator will be used to simulate microcontrollers and allow questions to be made that utilize this. The emulator will aim to allow students to upload their code and run it on what would be the microcontroller and see its behavior.

Functionality

Describe how your design is intended to operate in its user and/or real-world context. This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

Our design is intended to create a better learning experience for students throughout CprE 288. The questions and auto grader create a more engaging learning experience that will help grasp lecture material. The emulator then helps students in the lab as it makes it easier and faster to test code without needing a physical CyBot and testing space. Both of these are used increasingly throught the course as the labs and lecture material becomes more complex



Task Name	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
Homework 1	▨													
Homework 2		▨												
Homework 3			▨											
Homework 4				▨										
Homework 5					▨									
Homework 6						▨								
Exam 1 Review		▨	▨	▨	▨	▨	▨							
Homework 7								▨						
Homework 8									▨					
Homework 9										▨				
Homework 10											▨			
Homework 11												▨		
Exam 2 Review								▨	▨	▨	▨	▨	▨	
Final Exam Review	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

How well does the current design satisfy functional and non-functional requirements?

The current design satisfies these functional requirements as it lays out a plan to finish the auto grader, create more questions for students, and then create an improved emulator to be used in the lab.

4.7.2 Design 1 (Design Iteration)

Include Include another most matured design iteration details. Describe what led to this iteration and what are the major changes that were needed in Design 0.

Design Visual and Description

Include a visual depiction of this design as well highlighting changes from Design 0. Describe these changes in detail. Justify them with respect to requirements.

NOTE: The following sections will be included in your final design document but do not need to be completed for the current assignment. They are included for your reference. If you have ideas for these sections, they can also be discussed with your TA and/or faculty adviser.

4.8 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

4.9 Design Analysis

- Did your proposed design from 4.7 work? Why or why not?
- What are your observations, thoughts, and ideas to modify or iterate further over the design?