DESN2000 T2 2023 – COMP Stream
Project Brief: Lift Simulator

1. Project Description

There is a lift manufacturing company that has deployed thousands of new lifts across different venues. However, after a while, they realized that the lifts are buggy due to a design flaw in the lift controller firmware. The engineers overlooked real-world inputs that affected the firmware. The management has decided to redesign and develop the firmware rather than changing the existing one, as the documentation for the existing firmware is unclear, and the engineers who designed it are no longer with the company.

The plan is to implement the new firmware in three steps to ensure there is no room for error. The first step is to create an emulator on a development board with an ATmega2560 microcontroller, similar to the one used in the existing lift. Once the emulator is complete, the quality assurance team will use it to test the firmware with various combinations of user inputs to ensure its functionality before moving to the next step of flashing the firmware into a real prototype lift and testing it. Finally, the firmware will be deployed across all the lifts in a controlled fashion.

As a group of engineers tasked with designing and implementing the lift firmware, you are supposed to implement the emulator. Firstly, the group should observe existing real-world lifts in locations where lifts are commonly used in daily life (e.g., university buildings, apartments, shopping malls, etc.). This allows identification of potential issues that you may face. In addition, you may speak with colleagues or friends about their issues they have encountered while using lifts in their day-to-day lives.

The lift controller firmware is to be written in assembly language to allow optimisations in future for best response times and performance. The emulator will simulate different types of user inputs to test the firmware's functionality. You will be given a development board with an ATmega2560 microcontroller during the course. To build the emulator, the group must decide which components of the development board will map to which components in the lift. For example, the motor on the board can rotate to indicate whether the door is open or closed, and the LCD display can show the lift floor.

The group should consult with the quality assurance team (in this case your tutor will pretend to be that team) to determine the best mapping of components to simulate real-world user scenarios.

2. Teamwork

This is a group project, and you are required to work in a group of **four** students. Your tutors will facilitate weekly sessions, where your team will work on various aspects of the project. It is important to work efficiently together as a team to achieve timely deliverables and to successfully complete this project. Each student will have access to their own AVR board to ease collaboration and testing. Students are free to collaborate via any means they prefer but are expected to meet with tutors during face-to-face weekly labs/workshops.

3. Deliverables

Deliverable	*Weighting*	*Week due*
1. Design journal	20 %	Week 7
2. Lab exercises	20%	Weeks 1-5,7-9
3. Design Presentation	20%	Week 10
4. Final Product (including manuals and source code)	40%	Week 10

Design journal

Throughout the course, you will document your design process, including project management, in a design journal. The journal will be marked two times by your class demonstrator, Week 7 Monday 08:59 AM. Design journals can be used to draft ideas, record weekly meeting minutes and task assignments, draw sketches, write calculations and anything else related to your design process.

Lab exercises

Each week, students are expected to use provided lab sheets to progress their projects. These lab exercises are intended to help teams work through parts of the technical challenges you face in developing assembly code.

Design Presentation

Your design presentation should define the problem and describe what solution you propose based on the user research. You should do a live demonstration showing how your system can be used. The design presentations will take place in-class in Week 10.

Final Product

The final product is emulator of the firmware for the lift controller and includes the source code, user guide and developer guide. You tutor should be able to compile your source code and flash it onto a board and use it as instructed on the user manual. You tutor will also go through your code with the help of developer guide and attempt to understand the implementation.

User guide: The user guide is a critical component of your emulator project, as it provides instructions for the QA team to set up, operate, and troubleshoot the emulator independently, without assistance

from the development team. The guide should be comprehensive, clear, and easy to follow, and should include step-by-step instructions for all key functions, such as installing and configuring the emulator, selecting input/output components, and testing functionality. Your tutors will act as the QA team and will evaluate the guide based on their ability to successfully set up and operate the emulator using the instructions provided. The guide should be presented in PDF format, but you may also include additional resources, such as videos or diagrams, to enhance clarity and ease of use.

Source code: To ensure that your program is easily maintainable and extendable, it's crucial to structure it well and provide adequate comments. During development, you are expected to use a GitHub repository, which provides an excellent platform for version control and collaboration.

Developer guide: The developer guide is an essential document that describes the design of your code for future developers who may need to extend, optimize, or fix bugs in the codebase. Assume that you are no longer part of the company and therefore not available to offer any help to future engineers. As such, the design manual should be comprehensive and easy to understand. It should clearly indicate the mapping of components in the source code to the design components. You are expected to create the developer guide in markdown format and store it in the same GitHub repository as your code. During evaluation, your tutors will play the role of future engineers and assess the guide based on how easily they can understand and follow the instructions.